Learning to Interpret Historical Maps by Exploiting Polygon Metadata

by

Fandel Lin

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

May 2026

Dedication

To my father, Jer-Yann Lin, and my mother, Ding-Ying Guo, for their love and support.

Acknowledgments

First of all, I would like to thank my Ph.D. advisor, Prof. Craig A. Knoblock, for his guidance on my research and for providing the opportunity to work on interesting projects. I have learned many things that I did not expect before joining the USC Information Sciences Institute (ISI). Which, I believe, has been and will be a valuable experience and memory for my ongoing and long-term research journey.

I would like to thank all committee members for my qualifying exam, thesis proposal, and dissertation, Prof. Craig A. Knoblock (chair), Prof. Cyrus Shahabi, Prof. Yao-Yi Chiang, Prof. Yolanda Gil, Prof. John P. Wilson, and Prof. T. K. Satish Kumar, for their discussions and insightful feedback, which have helped me improve on how to formulate, shape, and present my research throughout the past years, not limited to my dissertation.

I would like to thank my colleagues at the USC Information Sciences Institute, Binh Vu, Basel Shbita, Minh Pham, and Abrar Jahin, for their supportive discussions on research and daily life in Los Angeles. In addition, I would like to thank Karen Rawlins for her stable administrative help during my years at USC ISI.

I would like to thank my long-term collaborator and M.S. advisor, Prof. Hsun-Ping Hsieh, for his ongoing support and discussions on our research. It is a pleasure working with him and his team on various research topics, especially in urban computing, for the past eight years almost without interruption.

I would like to thank my research collaborators and friends for their support and inspiring discussions on various research topics or projects: Zekun Li, Yijun Lin, Jina Kim, Leeje Jang, Sofia Kirsanova, Theresa Chen, Min Namgung, Jiyoon Pyo, and many others from Prof. Yao-Yi Chiang's group at UMN, Weiwei Duan, David Abbondanzio, Michael P. Gerlek, and Steven N. Minton from Inferlink, Min-Hsueh Chiu formerly at USC ISI, Ting-Rui Chiang and Han Zhang from USC, and Graham W. Lederer formerly at USGS.

I would like to thank Prof. Ning Wang, Prof. Saty Raghavachary, Prof. Filip Ilievski, Prof. Tracy Levin, Prof. Elizabeth Fife, Prof. T. K. Satish Kumar, and Prof. Cyrus Shahabi for their inspiring lectures and discussions related to teaching methodologies or organizing course flow, which have helped me become a better presenter and instructor.

I would also like to thank my friends here at USC or West Coast, Yu-Hsiu Hsieh, You-Ren Chen, Ming-Chang Chiu, Shih-Yao Huang, Ethan Chang, Andrew Lim, Brian Kuo, Janet Wu, and Yan Wen, and friends back in Taiwan, Tien-Yuan Chen, Chung-Kai Zheng, Sheng-Ying Pan, Yi-Ting Hsieh, Lei-En Chen, and Nai-Yu Chen, for chatting, hanging out, or dining out when available.

I am grateful to my parents, Jer-Yann Lin and Ding-Ying Guo, for their understanding, support, and discussions on research. It was a dream come true for me to contribute and work with them on some interesting research topics. I would like to thank my brother, Handel Lin, for his support and helpful suggestions, especially on the engineering aspects.

In addition, I appreciate the 4-year fellowship from the Ministry of Education of Taiwan under the Pilot Project on Scholarships for Taiwanese Studying in the Focused Fields at Top Foreign University, as well as the 3-year fellowship top-off from the USC Viterbi School of Engineering.

For some sections of this dissertation, the material is based upon works partially supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112390132 and Contract No. 140D0423C0093. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA) or its Contracting Agent, the U.S. Department of the Interior, Interior Business Center, Acquisition Services Directorate, Division V.

Table of Contents

List of Tables

List of Figures

Abstract

Historical maps contain valuable information for applications such as critical mineral assessment. However, most maps are available only in raster format, hindering automated analysis. We target the problem of learning to interpret historical maps by colorizing, recoloring, and digitizing polygonal features.

We present a metadata-driven machine-learning framework that leverages polygon metadata. The polygon metadata includes map keys indicating the visual appearance or semantic attributes of polygonal features, as well as lines delineating polygon boundaries. Building on this information, our framework jointly addresses colorization, recoloring, digitization, and generalization. First, for draft maps with dense handwritten overlays on monochromatic basemaps, our approach learns to colorize maps by interpreting monochromatic keys and annotations. Second, for maps with color inconsistencies caused by production or scanning artifacts, it learns to recolor maps by detecting and correcting mismatches between polygonal features and their corresponding map keys. For digitization, it encodes map content and keys into bitmaps and employs a convolutional model to learn to recognize polygonal features. To generalize digitization across diverse map styles, it learns region representations from map legend items and consensus-based pseudo-labels via contrastive objectives, adaptively reweighting solutions from complementary modules.

We evaluate our framework on six public map series, spanning diverse printing techniques, color schemas, and pattern degradation. For out-of-domain polygon digitization without target-style annotations, our approach achieves statistically significant improvements over state-of-the-art methods, including pre-trained large vision-language models, by 6.09% in F1 score and by 5.88% in estimated reductions in post-editing effort. For the in-domain historical map benchmark with predefined training data, our polygon-digitization approach outperforms comparative methods by 4.52% in F1 score. Controlled experiments on the

in-domain benchmark show that recoloring and colorization improve downstream polygon-digitization precision by more than 8.55% and outperform comparative methods by more than 7.51%, demonstrating their effectiveness as supporting modules.

Chapter 1

Introduction

1.1 Motivation

Historical maps preserve long-term geographic, environmental, and geological information that is often unavailable from modern surveys [16]. These maps document land cover, geological formations, infrastructure development, and environmental changes across centuries. Consequently, digitizing historical maps into structured vector representations enables a wide range of downstream analyses, including land-use reconstruction, environmental monitoring, geological assessment, and urban growth modeling.

A large portion of historical map archives exists only as scanned raster images. Transforming these raster archives into analysis-ready vector data requires identifying and delineating geographic entities, including polygonal regions, linear boundaries, and point features. Among these, we target polygonal features that represent spatial extents of semantic regions such as geological units, land cover classes, and administrative zones. However, converting raster maps into polygonal vector layers typically requires substantial manual interpretation by domain experts.

For instance, finding potential sites for undiscovered critical mineral deposits, such as lithium and cobalt, is pivotal to securing the global supply chain for technologies and national security. Accurate geological data is essential for these assessments, and the United States

Geological Survey (USGS) has a collection of over 100,000 historical maps. However, a significant portion of these historical archives exists only in scanned raster formats and hinders downstream analysis.

The difficulty in digitizing geological features represented as polygons from raster maps is four-fold. First, the cartographic styles of map keys (legend items) and their content are highly diverse, often incorporating complex markings, textures, and textual labels for specific regions. Second, existing color maps may suffer from significant color inconsistencies caused by scanning artifacts, such as paper creases and shadows, or production artifacts, including shaded relief overlays or digital elevation models, leading to mismatches between polygon features and their intended appearances defined in the legend. Third, geological data of some regions exists only as monochromatic draft maps produced during fieldwork. These draft maps contain dense handwritten annotations and interwoven boundaries that are difficult to distinguish without color encoding, yet manual colorization is prohibitively labor-intensive. Finally, automated models typically suffer from severe domain shift. A polygon-digitization model trained on one map series may fail when deployed on maps from different archives due to variations in printing techniques, ink diffusion, and pattern degradation.

1.2 Problem Statement

To address these challenges, given the historical maps with identified polygon map keys and map content area, we target the problem of automatically interpreting and digitizing polygonal features from historical maps across diverse cartographic styles.

1.3 Thesis Statement

We can build a system that automatically learns to interpret historical maps by colorizing, recoloring, and digitizing polygonal features using polygon metadata.

1.4 Approach

We define the term ***Polygon Metadata*** as *the external data on maps about polygonal features*. The polygon metadata can be any data that is on the map, describing or about the polygon feature, but cannot be the geometric boundaries of the polygon feature ('a set of line segments') itself.

An example of polygon metadata is the map legend, which has a set of map keys that describe the visual representation (e.g., colors and text patterns) of the corresponding polygon features on the map. In addition to the visual representation, the map keys may reveal the semantic meaning of the corresponding polygon features. For instance, the colors used for each map key indicate the rock type and geological age of that polygonal feature, or geological unit [71, 75].

Our work consists of four frameworks to address the various stages of exploiting polygon metadata for facilitating historical map interpretation.

First, we present ***LOAM*** [49] (**L**egend-**O**riented **A**utomated polygon digitization from **M**aps), a polygon-metadata-driven machine-learning approach to extract polygonal features from raster maps. It encodes the raster map and its corresponding map keys into a series of intermediate bitmaps that mimic how humans read maps. These representations are then processed by a convolutional neural network to learn to adaptively recognize polygon features across different map styles.

Second, to correct significant color inconsistencies between map content and polygon map keys, we present ***REPOLISH*** [50] (**RE**coloring via **P**olygon-**O**riented **L**earning with **I**nterpretative **S**pectra in **H**istorical maps). It reformulates map recoloring as a constrained image-to-image correction problem, utilizing a generative adversarial network (GAN) to adjust saturation and value while preserving the hue identity defined by map keys. By extracting interpretative spectra and structural sketches, it learns to correct anomalies and maintain semantic consistency for downstream analysis.

Then, to address the colorization of monochromatic draft maps, we propose ***SHADING*** [51] (**S**emantic–**H**armonic **A**chromatic **D**raft **I**nterpretation and **N**arration for **G**eological maps). This serves as a semantic bridge to turn draft maps into a particular cartographic style that our polygon digitization model is familiar with. We combine instance segmentation results derived from the map content with the semantics of the polygon legend items. By employing a conditional generative model with a cross-attention mechanism, the model learns to colorize draft maps guided by implicit coloring conventions and spatial constraints.

Finally, we present ***GLYPH*** [52] (**G**eneralization via **L**egend-guided **Y**oked **P**olygon extraction in **H**istorical maps), which addresses cross-domain generalization across diverse historical map styles. This enables automated polygon digitization without requiring annotations for the unseen cartographic domains. It integrates outputs from multiple complementary expert models at the regional level. By leveraging region-level representations and optimizing fusion weights via test-time adaptation, it learns to dynamically reconcile model outputs and produce polygon masks that are geometrically coherent and semantically aligned with the map keys.

1.5 Contribution of the Research

We summarize the main contributions of this dissertation as follows:

- A metadata-driven approach that generates multiple representations capturing different aspects of map interpretation and learns adaptive recognition of polygon features.
- A machine-learning model that corrects color inconsistencies between polygon features in map content and their corresponding map keys, improving semantic alignment.
- A conditional generative framework that integrates structural understanding of sketches with semantic reasoning of polygon map keys to colorize achromatic draft maps.
- A legend-guided mixture-of-experts framework that performs test-time adaptation to generalize polygon digitization across map archives with diverse cartographic styles.

1.6 Outline of the Dissertation

The thesis is organized as follows. Chapter 2 introduces LOAM for digitizing polygonal features from raster maps. Chapter 3 presents REPOLISH for correcting polygon coloring inconsistencies in historical maps. Chapter 4 describes SHADING for colorizing achromatic draft geological maps. Chapter 5 details GLYPH for cross-domain generalization of polygon digitization. Finally, Chapter 6 concludes the dissertation and discusses future research directions.

REPOLISH (Chapter 3) and SHADING (Chapter 4) serve as semantic restoration means that regularize significantly noisy or incomplete maps into a standardized cartographic format. LOAM (Chapter 2) acts as the core for the in-domain polygon digitization to handle arbitrary unseen legend items with known cartographical map styles. Based upon LOAM's outputs, GLYPH (Chapter 5) supports out-of-domain polygon digitization for unseen map styles.

In addition, we introduce automated map segmentation in Appendix A to turn the entire digitization pipeline into a fully automated process, with the only human input being the raster map.

Chapter 2

Exploiting Polygon Metadata to Digitize Polygonal Features

Historical maps preserve critical geographic, environmental, and geological information that is often unavailable from modern surveys. Transforming these raster archives into structured vector representations, specifically polygonal layers, is essential for longitudinal studies in land-use reconstruction, urban growth, and natural resource management. For instance, locating undiscovered deposits of critical minerals requires accurate geological data. However, most of the 100,000 historical geological maps of the United States Geological Survey (USGS) remain in raster format. This hinders modern critical mineral assessment. We target the problem of extracting semantic polygonal features from raster maps. We exploit polygon metadata, which provides information on polygon features, such as map keys indicating how the features are represented in the map content, to extract polygon features. We present a metadata-driven machine-learning approach that encodes the raster map and map key into a series of bitmaps and uses a convolutional model to learn to recognize the polygon features. We evaluated our approach on USGS geological maps; our approach achieves a median F1 score of 0.809 and outperforms state-of-the-art methods by 4.52%.

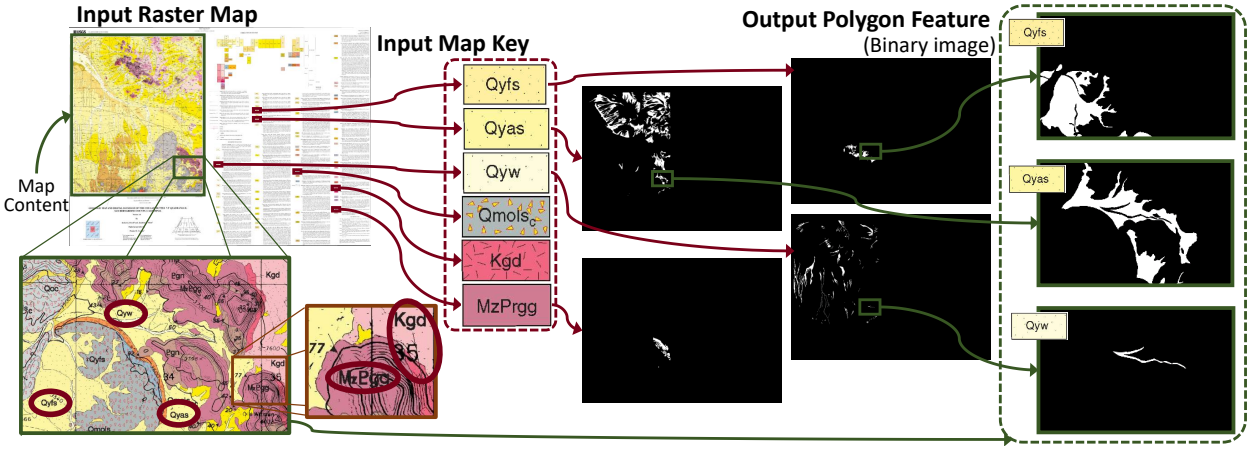


Figure 2.1: An example of extracting polygonal features represented as polygons from a raster map. The input is a raster map (top left), and the bounding boxes of the map key (red squares in the top-left map). The desired output is a binary image for each polygon feature on the raster map. There are several challenges in map key extraction: 1) the map key for *MzPrgg* has a solid background color, and the map key for $Qmols_c$ has triangular markings; 2) color shift for feature *Qyfs* between the map key and its corresponding polygon feature in the targeted map content; 3) similar colors between map keys, e.g., *Qyfs* and *Qyas* in the map content (top left) despite being distinguishable from the map keys (center). The only means to differentiate these two polygon features in the map content is the text label; 4) the text labels of *Qyw* and *Kgd* located nearby the correct polygon feature, and the map uses a line pointing text label of *Kgd* to the correct polygon feature (bottom left, brown block).

2.1 Motivation

Historical maps preserve long-term geographic, environmental, and urban information that is often unavailable from modern surveys [16, 40]. Digitizing such raster archives into structured, linked polygon layers enables downstream analyses such as land-cover reconstruction, urban growth studies, and infrastructure planning [67]. Among the various cartographic elements, we target polygonal features as they represent the spatial extent of thematic regions, such as forest types, soil classifications, urban property boundaries, and geological units.

Figure 2.1 shows an example of extracting polygon features from a raster map. Each polygon feature is identified by a map key specifying the feature's color and name. For each key, the task is to extract a binary mask image, capturing the corresponding feature to convert raster maps into analysis-ready formats.

The challenge of extracting polygon features from raster maps is four-fold (see Figure 2.1). First, the map keys come in different styles (e.g., colors or markings). Second, for a certain polygon feature, the color used in the map key may differ from the color used in the corresponding polygon feature in the map content. This is due to the scanning process or the fact that polygon features overlap with translucent symbols. Third, multiple keys on one map can have the same color and are differentiated only by text labels. Fourth, a text label can be located nearby, instead of inside, the correct polygon feature. Sometimes, the map uses a line to point the text label to the correct polygon. This is due to the size of the polygon features.

Most previous research focuses on extracting only one specific characteristic type (e.g., color) for extracting polygons from raster maps [4]. These single-feature extraction approaches consist of a series of image-processing techniques and require a tailored model for each feature. Thus, they do not apply to processing a large number of maps and polygon features. On the other hand, some previous studies apply foreground detection [79] and instance segmentation [27, 36] to segment polygon features from raster images. However, these approaches do not match the identified polygon features with the map keys. An additional procedure is needed to either link the identified features to the targeted map key (post-process) or fuse the map content with the map keys (pre-process).

To address these challenges, we propose a novel method that exploits the metadata that provides information on polygonal features. Our approach first encodes the raster map and map keys into a series of bitmaps. These bitmaps represent various aspects that people rely on when recognizing polygon features from raster maps. These aspects include matching and detecting colors, texts, and polygonal feature boundaries. Meanwhile, we embed the raster map and map keys that represent their complexity. The embedding includes color variety, color distances, and content coverage in the raster map or among keys. Next, our approach applies a convolutional neural network model that treats the series of bitmaps as multiple input channels. The model exploits the input bitmaps and the embedding to learn to adapt

to different styles of raster maps and map keys to return the extracted polygon features. We provide an overview of our approach in Figure 2.2.

We evaluate our approach using USGS geological maps [25] provided in the DARPA Critical Mineral Assessment Competition[1]. This dataset serves as the primary public benchmark for the polygon digitization framework due to its cartographic complexity, rigorous protocol, and data integrity. The dataset provides explicit training, validation, and testing sets, with no overlap in maps or polygon map keys across the three sets. This ensures that the tested model learns the relationship between legend and content rather than memorizing specific categories. With an average of more than 30 polygon map keys per map, this dataset serves as a difficult benchmark for evaluating supervised-learning-based methods with available in-domain training data (in-domain cartographic styles). Our approach outperforms the state-of-the-art method that placed first in the competition in October 2022 by 4.52%. Moreover, we present a case study that provides a qualitative analysis of the polygon-recognition model in our approach. While we use the USGS dataset to validate the core architecture given unseen polygon map keys with similar cartographic styles in this chapter, we present the cross-domain evaluation, in which map keys are unseen, and maps are in distinct cartographic styles, to demonstrate the robustness across broader map archives in Chapter 5.

The contribution of this chapter is a novel approach that exploits metadata to extract polygonal features from raster maps with arbitrary polygon map keys in the map legend. Our method generates multiple representations of the map for each map key. These representations capture different aspects of map understanding, such as extracting polygon features based on colors, textual descriptions, and boundaries. Our method then leverages these representations to adaptively extract polygon features from maps with polygon map keys that were not seen during training.

[1] https://criticalminerals.darpa.mil/The-Competition

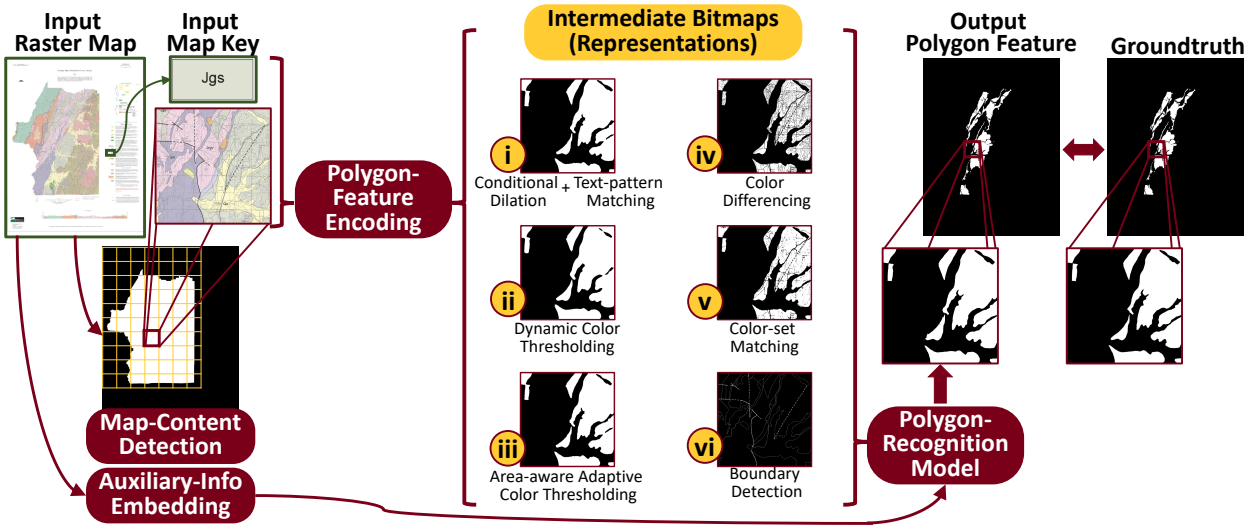


Figure 2.2: An overview of our approach to polygon feature extraction. There are five types of intermediate bitmaps when encoding the map content with a map key. These bitmaps and the embedding support the polygon-recognition model.

2.2 Approach to Polygon Extraction

2.2.1 Problem Definition

The input includes (1) a raster map and (2) a raster image of the map key (the geological feature represented as polygons that we want to extract) from the map legend in the raster map.

The output is a binary image for each polygonal feature in the raster map. We use binary representation for the extracted polygon feature. We show an example of extracting polygon features from a raster map based on keys in Figure 2.1.

2.2.2 Approach Overview

Our goal is to extract geological features from a raster map into a binary image by using the map keys as background knowledge.

We summarize the process of people reading a map and recognizing polygon features from the map as follows: (r1) comprehending the colors, texts, or symbols used by map keys in

the map legend; (r2) identifying the region of interest in the map; (r3) finding the areas with colors similar to the color of our targeted map key; (r4) distinguishing symbols and textures overlapped with or nearby the found areas; and (r5) using texts or boundaries to further differentiate the polygon features if multiple keys have the same colors.

Consequently, our approach mimics this process for polygon extraction. We illustrate our approach in Figures 2.2 and 2.3. In the first stage, our approach detects the map content of the scanned image (corresponding to r2) and embeds auxiliary information representing the color variety and complexity of the map and map keys (corresponding to r1). Next, we generate five types of (intermediate) bitmaps representing different aspects of understanding the map for extracting the polygon feature of each map key. We introduce their intuitions (the labeled Roman numerals corresponding to the ones in Figure 2.2 and Figure 2.3): (i) identifying the polygon feature based on the color of the map key, and excluding polygons that are labeled with texts different from the targeted one (corresponding to r5); (ii) identifying the polygon feature based on the color of the map key, and including surrounding areas that have different colors due to translucent symbols or textures (corresponding to r4); (iii) directly finding the areas that have colors similar to the color of our targeted map key (corresponding to r3); (iv) assuming that each pixel in raster maps can only belong to one map key, finding the areas that have the most-similar colors to our targeted map key (corresponding to r1 and r3); and (v) using boundaries to differentiate the polygon features if nearby polygons have the same colors but belong to other map keys (corresponding to r5).

The above five types of bitmaps (map-understanding aspects) correspond to (i) the polygon feature generated based on adaptive color thresholding with conditional dilation (extracting areas based on colors) and text-pattern matching; (ii) the polygon feature generated based on dynamic color thresholding (considering translucent symbols); (iii) the color differencing showing the distances between the key and each pixel (finding areas with similar colors); (iv) the polygon feature generated based on color-set matching; and (v) the output of boundary detection.

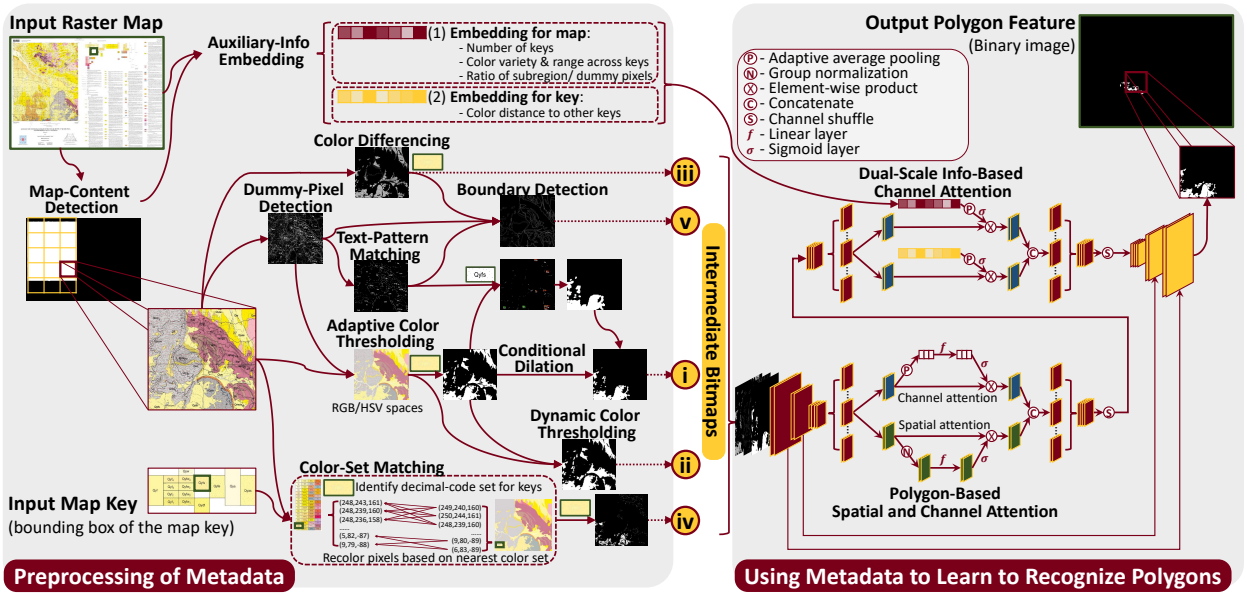


Figure 2.3: The workflow of polygon feature extraction. Our approach uses the metadata to encode the map content and each map key into a series of bitmaps. It then applies a convolutional model to learn to recognize the polygon feature. The Roman numerals in the intermediate bitmaps (center, yellow block) correspond to the ones in Figure 2.2.

In the second stage, our approach treats the series of bitmaps as the input of a convolutional neural network to learn to recognize the polygon feature of each key. Meanwhile, our polygon-recognition model exploits auxiliary-information embedding representing the latent styles of raster maps and map keys. Based on the identified latent styles, our polygon-recognition model leverages different types of bitmaps to adapt to maps and keys with different styles. The idea is that certain types of map-understanding aspects are more reliable than others, given certain map styles.

2.2.3 Preprocessing of Polygon Metadata

We exploit *metadata (polygon metadata)* to (1) encode the map and map keys into a series of bitmaps based on different aspects of understanding the map and (2) embed the map and map keys.

There are three sub-tasks for preprocessing the metadata: (1) detecting the map content in scanned images; (2) encoding the polygon features in the map content based on map

keys into a series of bitmaps to support the polygon-recognition model; and (3) embedding auxiliary information from the raster map and keys to support the polygon-recognition model.

2.2.3.1 Map-Content Detection

The map content is the specific area or region that has the geological features and information we are interested in. Our approach identifies the map content from the input raster map based on a connected-component analysis that considers the color variety.

The detection of map content from scanned images has two challenges: (1) there may be inconsistency in the shape and relative location of the various components on the scanned image, such as the map, the legend, and the auxiliary labels, photos, and text around the scanned images; (2) because these are scanned images, there may be artifacts in the image from the scanning process, such as creases in the paper, that cast a shadow over areas that are not our targets and generate false positives.

To address the first challenge, we identify background colors to determine the foreground image and then employ a connected-component analysis of the foreground image. In particular, our approach finds the background colors of the image based on the four corners of the image and separates the foreground image into multiple connected components. Next, by calculating the number of distinct colors for each connected component, our approach selects the largest connected component with its color variety higher than a lower bound that is set empirically. The reason for setting the lower bound of the color variety is to prevent selecting the auxiliary texts as our targeted map content; selecting the largest connected component is to avoid selecting the legend or auxiliary photos. For instance, in Figure 2.1, the connected component for the auxiliary texts occupies more than half of the raster image.

For the second challenge that involves the artifacts from the scanning process, our approach performs a series of dilation and erosion of the foreground image. This allows us to better determine the background and foreground of the scanned images. Figure 2.4 shows an input raster map having a non-rectangular map content with auxiliary labels or having creases and

Figure 2.4: Two examples of an input raster map (left) with its corresponding map content (right, white). One input raster map has a non-rectangular map content (top); while the other has creases with auxiliary labels and photos (bottom).

non-white background color with auxiliary labels and photos.

2.2.3.2 Polygon-Feature Encoding

Polygon-feature encoding provides a series of bitmaps (candidates) representing different map-understanding aspects. Our approach generates the candidates using the *Polygon Metadata* (e.g., colors and text patterns in the map keys).

The challenge of extracting polygon features from scanned geological maps is four-fold: (1) the keys from the legend on maps come in different colors, texts, or textures; (2) the polygon features in the map content often overlap with symbols or textures. This leads to a color shift

between the map key and the corresponding polygon feature, making it difficult to identify them from their surroundings. For instance, geological maps often overlap with contours, fault lines, or terrain features; (3) multiple keys from one map come in the same color; the only difference is the labeled text. This is usually due to the hierarchical relationships in the map legend; (4) the text is not always labeled in the corresponding polygon features due to the limited space of the features. These four factors make it difficult to develop a model that can accurately extract all polygon features without retraining the model for each feature.

To address the challenges, our proposed polygon-feature encoding turns the input raster maps and the map key into a series of bitmaps representing five map-understanding aspects. We illustrate the workflow of polygon-feature encoding in Figure 2.3 and introduce the components of our workflow as follows.

Dummy-Pixel Detection. The dummy pixels indicate opaque texts, contours, fault lines, or terrain features in geological maps. During polygon feature extraction, we can ignore these dummy pixels when they overlap with or are surrounded by our targeted polygon features.

The dummy-pixel detection takes the map content as input. Since most USGS geological maps label text, contours, fault lines, and boundaries in black, our approach applies color thresholding to identify these dummy pixels. We set the threshold based on observations from the training datasets. Besides, since most geological maps use solid lines or dashed lines to represent the boundaries between two polygon features, these dummy pixels can also support text-pattern matching and boundary detection.

Color Differencing. To address (1) the challenge of keys in different colors and (2) the slight color shift between map keys and corresponding polygon feature in the map content due to the scanning process (e.g., the feature *Qyfs* depicted in Figure 2.1 has different colors in the map key and map content). Our approach evaluates the color distance between the input map key and each pixel in the map content. We adopt the pixel-wise (1) Euclidean distance in RGB color space and (2) the difference in H space (Hue) from the HSV color

space to retrieve two results. We store the results into bitmaps ranging from 0 (black) to 255 (white), respectively. We assign a higher value to a pixel if it has a shorter distance to the target map key.

In addition, we notice that the boundaries among the polygon features can induce a large color gradient between the pixels located around the boundaries. Our approach calculates the color gradient among pixels to help identify the boundaries among polygon features in the map content.

Adaptive Color Thresholding with Conditional Dilation. Adaptive color thresholding exploits the idea that (1) there may be a slight color shift between the polygon feature and map key due to the scanning process or overlapping with translucent symbols, and (2) the range of fault tolerance of this color shift depends on whether other keys are using a similar color. If no map keys use a similar color, one can enlarge the range of fault tolerance due to the color shift for the map key. Despite the slight color shifts of a polygon feature, areas with shifted colors, such as translucent symbols, shall still be located by areas with no color shifts.

Conditional dilation exploits the idea that one can ignore dummy pixels that (1) represent opaque symbols and (2) are located by the areas extracted based on colors. We illustrate an example of the workflow for adaptive color thresholding with conditional dilation in Figure 2.5.

Due to the scanning process and the overlap with translucent symbols or textures, simply applying color thresholding based on the RGB or HSV color space could yield false-negative extraction. Accordingly, our approach first applies the mean shift for image smoothing [18]. Next, our approach automatically and iteratively relaxes the tolerance in the RGB and HSV thresholds for each key while preventing overlapping with the thresholds of other keys.

We depict the adaptive relaxation of the color space in Equation 2.1, in which h_{th}^{up} represents the upper bound of the threshold in the H color space of a map key, h_{th}^{lw} represents the lower bound of the threshold; h_{adapt}^{up} and h_{adapt}^{lw} indicate the relaxed threshold. h_{th}^{lw} refers

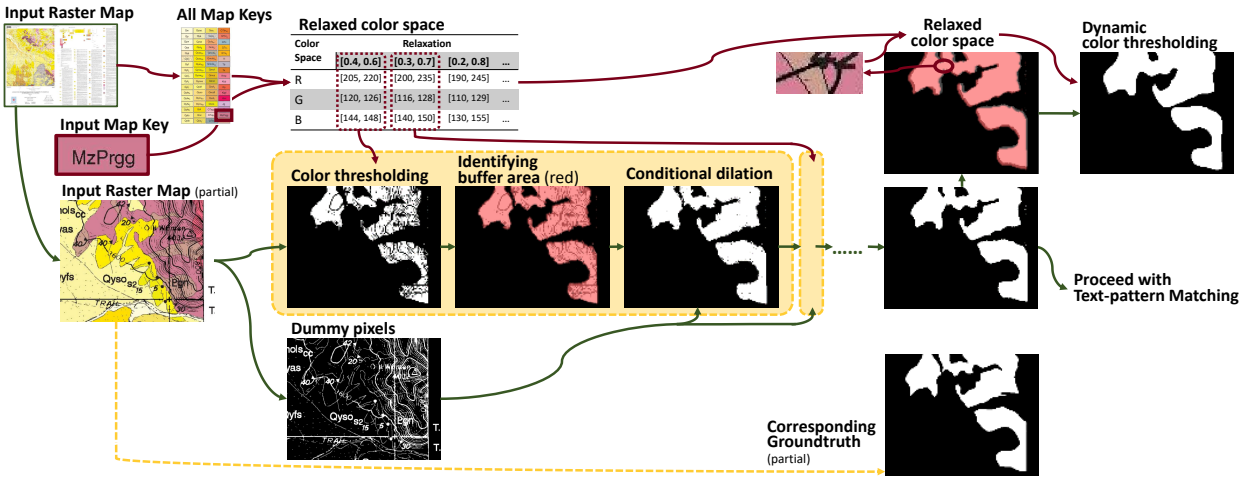


Figure 2.5: An example of incorporating the dummy pixels and the dilated buffer area from the previous iteration in adaptive color thresholding with conditional dilation.

to the minimum lower bound threshold for all keys that are larger than h_{th}^{up}, while $h_{th}^{\prime up}$ is the maximum upper bound threshold for all keys that are smaller than h_{th}^{lw}. There are two adjustable parameters: α is the minimum relaxation value, and β is the relaxation rate. We set α at 2 and β to 0.25 in the evaluation based on observations of the training dataset.

$$
\begin{aligned}
h_{adapt}^{up} &= h_{th}^{up} + min(\alpha, \beta(h_{th}^{\prime lw} - h_{th}^{up})) \\
h_{adapt}^{lw} &= h_{th}^{lw} - min(\alpha, \beta(h_{th}^{lw} - h_{th}^{\prime up}))
\end{aligned}
\tag{2.1}
$$

Dynamic Color Thresholding. Dynamic color thresholding deals with pixels surrounded by the extracted polygon but not extracted by adaptive color thresholding due to an overlap with translucent symbols. Our approach defines the relaxed color threshold based on the pixels in the buffer area of adaptive color thresholding that do not belong to any other keys.

Text-Pattern Matching. Text-pattern matching addresses the challenges of (1) multiple keys on one map having the same color and (2) texts not labeled in the corresponding polygon feature (e.g., *Qyw* in Figure 2.1). The input base map of text-pattern matching is the dilation and overlap of adaptive color thresholding and dummy-pixel detection. We apply pattern matching on this base map to find the target map key's text pattern. We depict its workflow

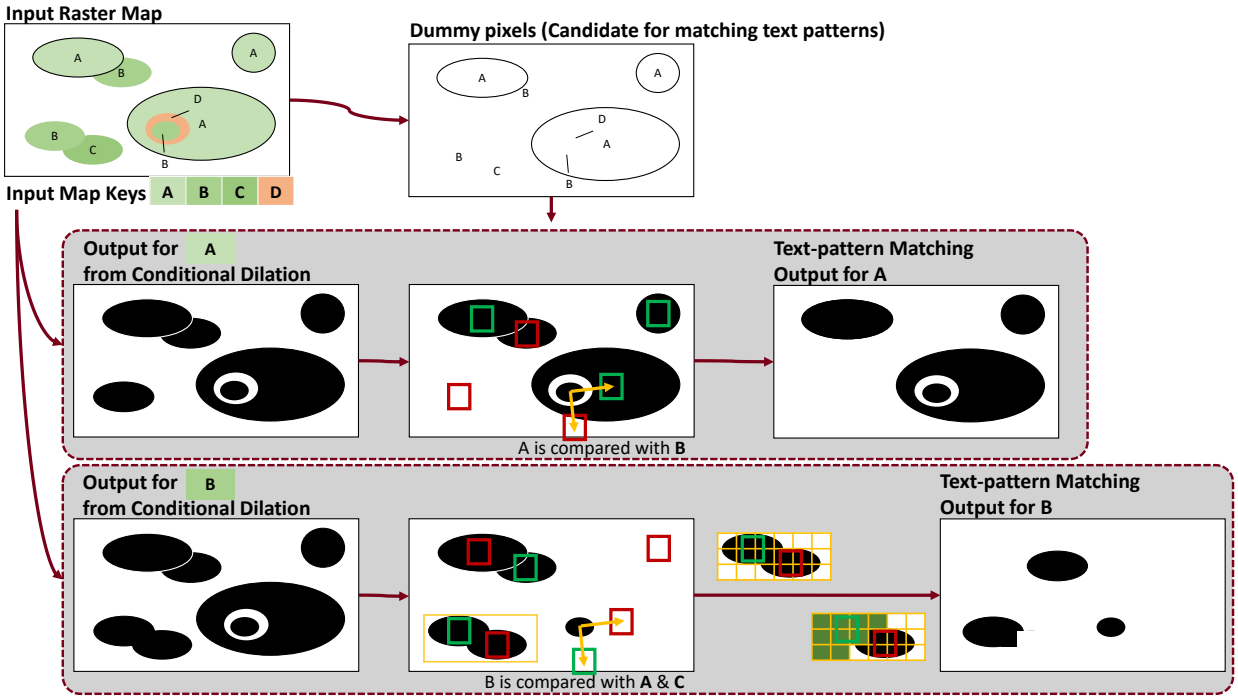


Figure 2.6: The workflow of text-pattern matching. We apply pattern matching with connected component analysis based on the map keys, the polygon outputs from adaptive color thresholding with conditional dilation, and dummy-pixel detection. The text-pattern matching handles cases in which multiple keys on one map have the same or similar colors.

in Figure 2.6.

Our approach excludes the connected components labeled with (or significantly closer to) text labels that correspond to a map key other than the targeted one but have a similar color. Our approach does not adopt optical character recognition due to the inconsistency in text representation between the key and the map content (e.g., *MzPrgg* in Figure 2.1).

Boundary Detection. The goal of boundary detection is to provide the polygon-recognition model with reference to denoising and smoothing the boundaries of polygon features.

Our approach incorporates the results from dummy-pixel detection and color differencing as the preliminary boundaries in the raster map. Next, our approach removes the pixels corresponding to the text labels from the preliminary boundaries and treats the rest of the pixels as the output.

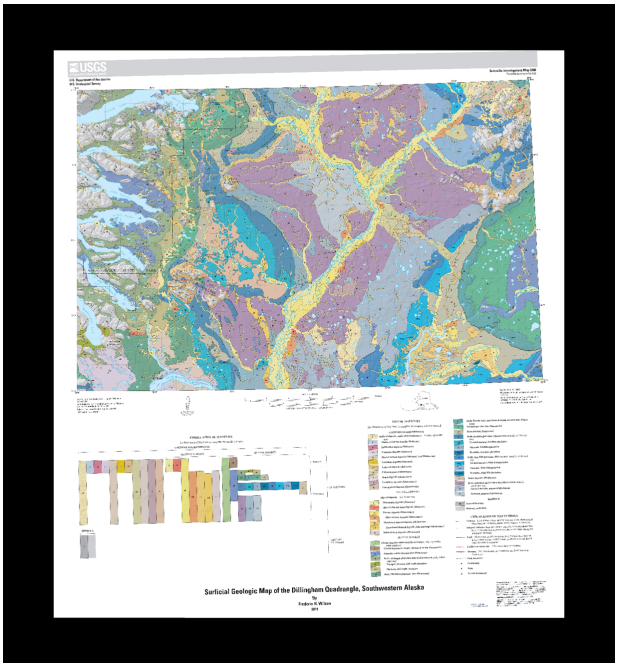
Color-Set Matching. The concept of color-set matching is that a pixel in the raster map can only belong to one map key. Our approach first identifies the color set for each map key per ten percentile. Next, we use a 5x5 kernel to represent the color set of each pixel in the map content. Our approach then calculates the distance between each pair of color sets from the map key and the pixel and assigns a pixel to the map key with the shortest distance.

Our approach considers the Euclidean distance in RGB color space and the difference between the R, G, and B spaces to evaluate the color-set distance. The idea of considering the difference between two spaces is to provide a reference when all keys in a raster map have similar values in two of the three spaces.

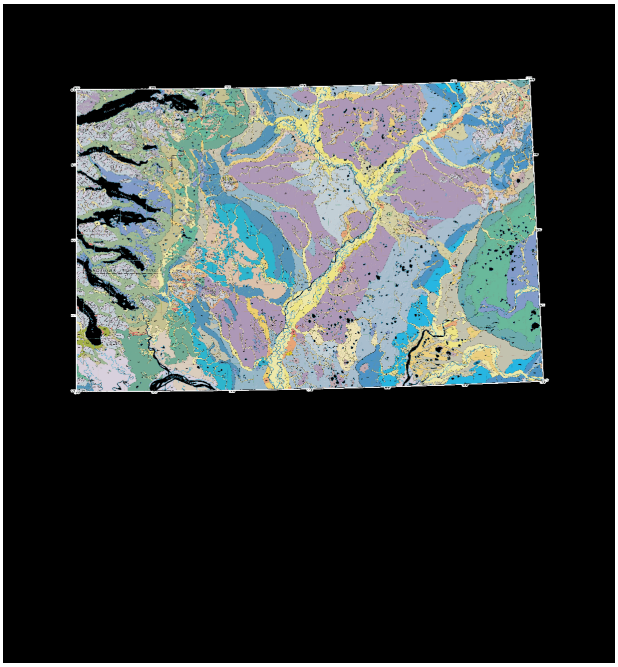
We depict the color-set distance $Dist_{K,P}$ in Equation 2.2, in which K stands for an input map key, P is a pixel in the map content; K^s is the identified color set for key K, P^n is the kernel of 5x5 pixels centered at P. R_x refers to a value in the R space of the color set x, in which x belongs to the key $k \in K^s$ or the pixel $p \in P^n$. On the contrary, $\overline{RG}_x$ is a value of the difference between the R and G spaces in the color set x. An adjustable parameter γ controls the weights between the distance based on the RGB color space and the distance based on the difference between the three spaces. We set γ to 0.95 in the evaluation based on the observation of the training dataset. After calculating the distance of all pairs of color sets for each pixel P in the map content, we assign the map key K with the smallest $Dist_{K,P}$ among all input map keys.

$$Dist_{K,P} = \sum_{k}^{K^s} \sum_{p}^{P^n} \left(\gamma \sqrt{\sum_{c}^{R,G,B} (c_k - c_p)^2} + (1-\gamma) \sqrt{\sum_{c}^{\overline{RG},\overline{GB},\overline{BR}} (c_k - c_p)^2} \right) \tag{2.2}$$

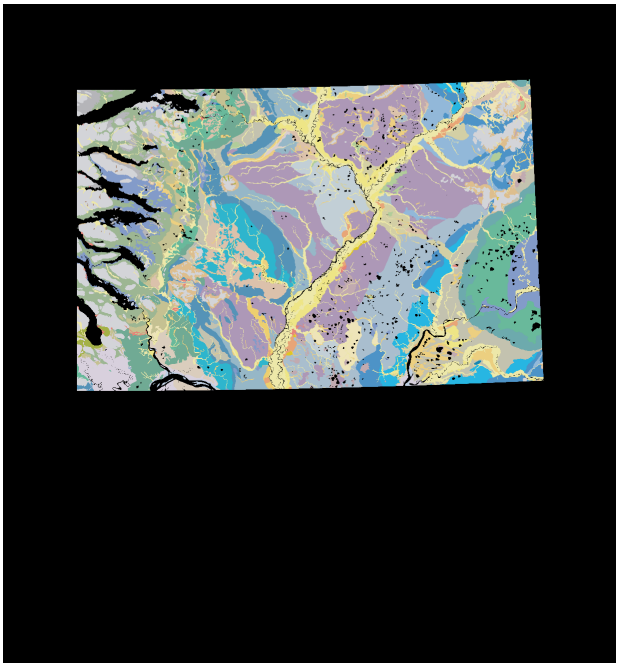
We demonstrate an example of color-set matching in Figure 2.7, in which we generate the reference image using the median color of each polygon map key and its corresponding ground truth polygonal feature, illustrating the color mismatches due to overlapping artifacts.



(a) Input raster map.



(b) The preliminary output of our color-set matching.



(c) An image generated as a reference for color-set matching.

Figure 2.7: An example of color-set matching. We generate the reference by recoloring the raster map based on the median color of each key and its corresponding ground truth polygonal feature.

2.2.3.3 Auxiliary-Information Embedding

Auxiliary-information embedding assesses and quantifies the styles of the raster maps and the map keys in two series of embeddings. The embeddings represent the latent characteristics of the maps and keys, respectively. Our approach then uses the embedding to let the polygon-recognition model learn which map-understanding aspects (the output of polygon-feature encoding) are more reliable, given raster maps and map keys with different styles.

Embedding for Map. For map embedding, the concept is to retrieve general information that demonstrates the style, complexity of a raster map and the variety of colors used by all map keys.

We include the number of keys in a map, the color variety across keys in terms of H space, the color range in RGB color space across keys, and the ratio of content and dummy pixels for map embedding. For instance, given a raster map with all keys using a similar color (indicating a small color variety regarding the H space), the candidate generated based on color differencing is more reliable than the candidate based on adaptive color thresholding.

Embedding for Key. The motivation for key embedding is to evaluate the color distances between the target map key and other keys. This helps the model know whether other keys use a similar color in the raster map.

We calculate the color distances between the targeted key and other keys with a similar color in terms of the Euclidean distance of the RGB color space, the minimum difference among three spaces, and the difference in the H space, respectively. Our approach sorts the values in ascending order and embeds the first ten values. For instance, if a key has a large value for the H-space distance to the nearest color, this indicates that there is no other key using a similar color. The candidate of color differencing will have a larger tolerance, since our approach highlights the pixels according to the color distance to the map key.

2.2.4 Using Metadata to Learn to Recognize Polygons

We feed the series of bitmaps (different map-understanding aspects) as multiple channels into a convolutional model to learn to recognize the polygon feature of each key. We use the embedding (latent characteristics of the map and keys) to teach the model to rely on different channels given raster maps and map keys with distinct characteristics. We illustrate the model in Figure 2.3.

For each map key, the input of our polygon-recognition model is the series of bitmaps (split with a size of 1024x1024 pixels) with the auxiliary-information embedding. The output is the binary image showing the corresponding polygon feature in the raster map.

2.2.4.1 U-Net-Based Convolutional Model

We feed the series of bitmaps as multiple channels into a convolutional model. Our model follows the architecture of U-Net [64].

The adopted U-Net is a neural-network architecture for biomedical image segmentation [12]. The idea of U-Net is to exploit feature maps generated during the downsampling process in the upsampling process. This enables precise localization and increases the resolution of segmentation.

The architecture of U-Net consists of two parts: a series of downsampling and upsampling. For the first half of the architecture, we apply 4 downsampling components. Each downsampling component doubles the number of feature channels and has 2 series of 3x3 unpadded convolution, rectified linear unit (ReLU), and a 2x2 max pooling operation with stride 2. For the second half of the architecture, we apply 4 upsampling components. Each upsampling component halves the number of feature channels, constructs a 2x2 up-convolution, concatenates the correspondingly cropped feature map during the downsampling process, and has 2 series of 3x3 convolution with ReLU.

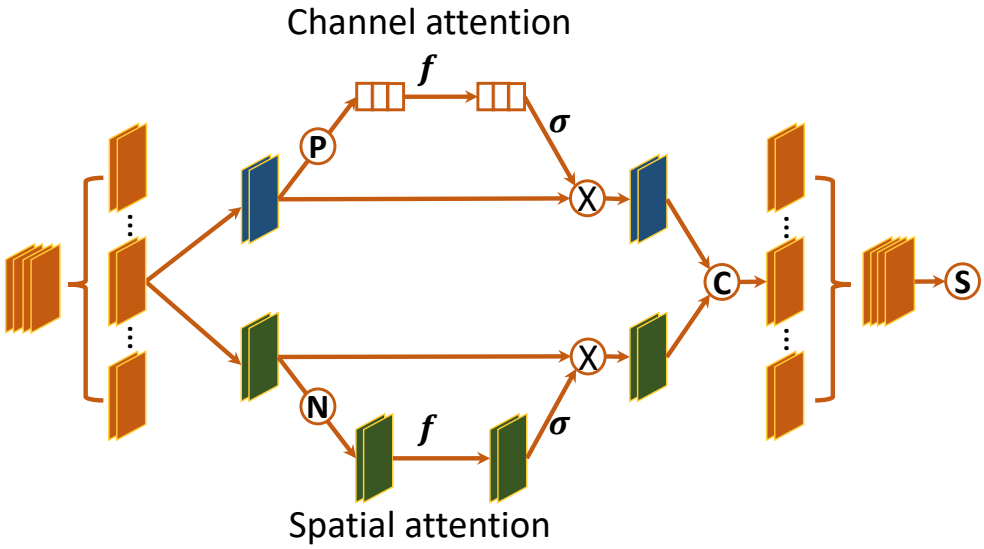


Figure 2.8: The adopted dual-attention mechanism applied in our two-phase shuffle attention. The first phase (polygon-based spatial and channel attention) follows this structure; we treat the results of auxiliary-information embedding as attention input in the second phase (dual-scale information-based channel attention) instead. In this workflow, "f" is a linear layer, "σ" is a sigmoid layer, "P" indicates adaptive average pooling, "N" refers to group normalization, "X" denotes element-wise product, "C" indicates concatenation, and "S" refers to channel shuffle.

2.2.4.2 Two-Phase Shuffle Attention

We then apply two phases of the shuffle attention (SA-Net) structure [93] between the downsampling and upsampling of the U-Net architecture, as illustrated in Figure 2.3. The concept is to let the model learn to put different weights across channels (bitmaps) and areas on the map to recognize the polygon feature.

The first phase of our model exploits the information from the series of bitmaps itself and has spatial and channel attention. In contrast, the second phase leverages the auxiliary-information embedding and turns out to be dual-scale channel attention.

Polygon-Based Spatial and Channel Attention. The first-phase attention in our model follows the SA-Net structure [93], which is a lightweight implementation of channel and spatial attention, as depicted in Figure 2.8. The goal is to let the model know which channels of bitmaps and what areas in the bitmaps are meaningful by exploiting the inter-relationship

of channels and the inter-spatial relationship of features.

SA-Net splits feature channels into groups and splits each into two branches, X_1 and X_2, to capture channel dependency and pairwise relationship at the pixel level, respectively. For channel attention, it employs 2-dimensional adaptive average pooling; on the other hand, spatial attention applies group normalization. Finally, it concatenates the results from the two attentions for each group and shuffles the results from each group back into the feature channels.

We list the output of channel attention in Equation 2.3, in which X_1 is one of the two branches of feature channels, and X_1' is the corresponding output after applying the channel attention. F_P indicates the adaptive average pooling, and $F_P(X_1)$ refers to the channel-wise statistics of X_1. σ is a sigmoid function that limits the output between 0 and 1. W_1 and b_1 are two parameters that the model learns to scale and shift $F_P(X_1)$; while g_h and g_w are the sizes of the input feature map.

$$\begin{aligned} X_1' &= \sigma\left(W_1 \cdot F_P(X_1) + b_1\right) \cdot X_1 \\ F_P(X_1) &= \frac{1}{g_h \times g_w} \sum_{i=1}^{g_h} \sum_{j=1}^{g_w} X_1(i,j) \end{aligned} \tag{2.3}$$

On the other hand, we list the spatial attention output in Equation 2.4, where X_2 is the other branch of feature channels and X_2' is the corresponding output after applying spatial attention. F_N indicates group normalization, with W_2 and b_2 being the parameters the model learns to scale and shift $F_N(X_2)$.

$$X_2' = \sigma\left(W_2 \cdot F_N(X_2) + b_2\right) \cdot X_2 \tag{2.4}$$

Dual-Scale Information-Based Channel Attention. For the second-phase attention in our model, our approach leverages the embedding of auxiliary information under two scales (map and key) to let the model know which channels of bitmaps (map-understanding aspects) are more reliable given different styles of input raster maps and map keys.

Our approach follows the SA-Net structure to split the feature channels into groups and halve the group into two branches. Next, we apply 1-dimensional adaptive average pooling on the map embedding and key embedding from the auxiliary information. We treat the results as the channel attention for each branch, respectively. Finally, our approach concatenates the results from the two attentions for each group and shuffles back into the feature channels.

2.3 Evaluation

To demonstrate our approach, we implement the ideas in a system named ***LOAM*** (***L***egend-***O***riented ***A***utomated polygon digitization from ***M***aps). We use the USGS geological maps released in a competition named *DARPA AI for Critical Mineral Assessment Competition - Feature Extraction Challenge*[1] to conduct the evaluation.

2.3.1 Dataset

We use the USGS geological maps [25] released in the competition to evaluate our approach. Each USGS geological map is a raster image and has a corresponding JSON file that records the bounding box of each map key in the geological map.

This series of geological maps has training, validation, and testing datasets. We use part of the training and validation datasets to train our model and use the testing dataset to evaluate the performance of our approach. We list the statistics of the dataset used for training and testing in Table 2.1.

Note that following the competition setting, we do not look into the testing dataset to tune our model.

Table 2.1: Statistics of the USGS geological map dataset used for training and testing.

Attribute / Usage	Training	Testing
Number of raster maps	14	24
Number of map keys	536	849
Maximum number of map keys per map	100	103
Median number of map keys per map	31	31
Minimum number of map keys per map	13	1
Maximum size of a raster map (pixel, width x height)	17,572 x 15,950	13,200 x 18,450
Minimum size of a raster map (pixel, width x height)	6,479 x 13,614	7,200 x 11,113

2.3.2 Evaluation Metric

Following the competition setting for polygon feature extraction[2], the evaluation metric is the median value of the weighted F1 score across map keys in the testing dataset.

For each map key, pixels in the ground truth that can be extracted based on a color-matching baseline are *easy pixels*. The competition set a weight of 0.3 for the *easy pixels* and a weight of 0.7 for the rest of the pixels in ground truth to calculate precision and recall. The F1 score is the harmonic mean of precision and recall, as stated in Equation 2.5.

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP} \\ F1 &= \frac{2 \times Precision \times Recall}{Precision + Recall} \end{aligned} \tag{2.5}$$

in which TP indicates true-positive weighted pixels that exist in both extraction output and the ground truth, FN is false-negative weighted pixels that exist only in the ground truth, while FP is false-positive pixels that exist only in extraction output.

[2] https://criticalminerals.darpa.mil/Files/Map_Feature_Extraction_Challenge_Details.pdf

2.3.3 Evaluation Setting

We adopt the PyTorch implementation of U-Net[3]. Following the implementation, we use the dice coefficient for the loss function, a widely used metric to evaluate the segmentation output. We set a batch size of 1, a learning rate of 1e-05, a weight decay of 1e-08, a momentum of 0.999, and split 20% of the training dataset for validation. We train the model for 40 epochs and select the model with the best performance on the validation set to infer the test dataset. Since there is no overlap between testing and training (including the validation set) datasets, this prevents contamination of the datasets or further tuning based on the testing dataset.

To address the data imbalance at the pixel level, during training, we only include bitmaps (after splitting into 1024×1024 pixels) with a corresponding ground truth that contains more than 25% ones (polygon features).

We implement all methods in Python on an ASUS desktop computer equipped with an AMD Ryzen 9 5900X CPU at 3.70 GHz, 128 GB RAM at 3200 MHz, and an NVIDIA GeForce RTX 3090 GPU. We release the source code on Github [4].

2.3.4 Comparative Method

Because we follow the evaluation setting in the competition[1], we use the performance listed on the leaderboard[5] of the competition held in October 2022 as the state-of-the-art methods. Performance on the competition leaderboard is reproducible.

The approach that won first place in polygon extraction [59] (team "ICM") integrates optical character recognition, adaptive histogram equalization, and a modified U-Net[6].

The approach that won second place in the polygon extraction (team "uncharted") first extracts pixels in the map content that can highly likely match a unique map key with a

[3] https://github.com/milesial/Pytorch-UNet

[4] https://github.com/Fandel-Lin/LOAM

[5] https://web.archive.org/web/20221202080740/https://criticalminerals.darpa.mil/Leaderboard

[6] https://www.ncsa.illinois.edu/nfi-ncsa-win-second-place-in-ai-competition-for-critical-mineral-assessment/

convolutional model. Next, it applies spatial high-pass filtering for noise removal and expands polygonal regions without overlapping each other[7].

The approach that won third place in polygon extraction (team "ISI-UMN", our submission to the competition[8]) integrates dummy-pixel detection, adaptive color thresholding, and text-pattern matching, as introduced in the previous section.

In addition, we use the Segment Anything Model (SAM) [36] as a comparative method. SAM is a zero-shot generalization-based approach to identifying objects from images. It is a state-of-the-art method for solving instance-segmentation tasks.

The baseline method is the color matching released by the competition authority. First, it identifies the median values in the map key regarding the RGB color space. Next, it applies a fixed value for color thresholding to extract the polygon feature. The evaluation metric uses this color-matching baseline to identify *easy pixels*.

2.3.5 Evaluation Result

2.3.5.1 Overall Performance

We show the overall performance in terms of the median weighted F1 score in the testing dataset for our proposed LOAM against comparative methods in Table 2.2. Our approach achieves a performance of 0.809 and outperforms state-of-the-art methods by 4.52%.

The performance of the color-matching baseline shows the limitation of using a fixed color threshold to extract polygon features from scanned images. This color-matching approach fails to deal with cases such as overlap with translucent symbols, contours, fault lines, and terrain features, or the color shift between the map key and the map content due to the scanning or map-generating process. Also, the baseline can produce false positives without identifying the map content. Thus, with map-content detection and adaptive color thresholding, LOAM and the team "ISI-UMN" can retrieve a performance significantly better than the baseline.

[7] https://uncharted.software/blog/uncharted-earns-another-spot-on-the-leaderboard-in-ai-competition-by-darpa-and-usgs/

[8] https://github.com/Fandel-Lin/mineral_assessment_competition

Table 2.2: Overall performance in terms of median weighted F1 score on the testing dataset.

Method	F1 Score
LOAM	**0.809**
Rank of polygon extraction on leaderboard[5]	
1st-place: Team "ICM"[6] [59]	0.774
2nd-place: Team "uncharted"[7]	0.632
3rd-place: Team "ISI-UMN"[8]	0.629
Instance-segmentation method	
SAM [36] with Color Matching	0.282
Baseline	
Color Matching	0.046

By applying a convolutional model to denoise and integrate candidates from metadata preprocessing, our LOAM and the team "ICM" perform better than the other comparative methods. Moreover, our LOAM exploits auxiliary-information embedding in the convolutional model to consider the latent styles of input raster maps and map keys. This allows LOAM to outperform team "ICM".

Besides, SAM has a limited understanding of geological symbols that overlap in the characteristics of the target polygon features [31]. Furthermore, the use of text patterns as a reference for SAM to identify polygons is not always accurate (Figure 2.1). SAM results in segmenting small objects and has limited performance on our task.

2.3.5.2 Ablation Study

Table 2.3 shows our ablation study regarding the weighted precision, recall, and F1 score at the 10th, 25th, 50th (median), 75th, and 90th percentiles on the testing dataset for LOAM. The ablation study has three parts: removing structures from the polygon-recognition model, removing input channels of the polygon-recognition model, and composing components in the metadata preprocessing.

We first focus on the ablation for the model structure. *LOAM -DC* and *LOAM -SA* have similar F1 scores, but *LOAM -SA* has better recall, and *LOAM -DC* ends up with better precision. When comparing these two with *LOAM -DC/SA*, we notice that exploiting only

Table 2.3: Ablation study for performance in terms of weighted precision, recall, and F1 score at the 10th, 25th, 50th (median), 75th, and 90th percentiles on the testing dataset. For each percentile, the bold value is the best performance, while the underlined value refers to the second-best performance. The abbreviations of LOAM components are listed as follows. DC: dual-scale information-based channel attention; SA: polygon-based spatial and channel attention; DT: dynamic color thresholding; CD: color differencing; CM: color-set matching; BD: boundary detection; AT: adaptive color thresholding with conditional dilation; TM: text-pattern matching.

Method	Weighted Performance														
Indicator	Precision					Recall					F1 Score				
Percentile	10	25	50	75	90	10	25	50	75	90	10	25	50	75	90
LOAM	0.032	0.280	**0.891**	**0.973**	0.986	0.422	0.762	0.915	0.966	0.988	0.054	**0.330**	**0.809**	**0.944**	**0.974**
Model Structure															
LOAM -DC/SA	0.035	0.262	0.870	0.970	0.984	0.339	0.695	0.897	0.962	0.985	**0.057**	0.307	0.771	0.938	0.974
LOAM -DC	0.030	0.246	0.838	0.966	0.983	0.409	0.741	0.906	0.966	0.988	0.048	0.304	0.763	0.935	0.972
LOAM -SA	0.031	0.220	0.814	0.961	0.980	0.470	0.792	0.927	0.972	0.988	0.051	0.289	0.760	0.938	0.973
Model Input															
LOAM -DT	0.030	0.216	0.795	0.963	0.981	**0.502**	**0.801**	0.928	0.970	0.988	0.053	0.271	0.769	0.937	0.973
LOAM -CD	**0.040**	**0.281**	0.825	0.964	0.983	0.241	0.661	0.903	0.967	0.987	0.054	0.316	0.765	0.937	0.973
LOAM -CM	0.024	0.156	0.780	0.961	0.984	0.458	0.770	0.921	0.977	0.992	0.034	0.220	0.691	0.918	0.966
LOAM -BD	0.026	0.185	0.784	0.960	0.987	0.348	0.623	0.839	0.943	0.983	0.043	0.251	0.689	0.895	0.952
Component															
AT+TM	0.029	0.224	0.740	0.892	0.946	0.337	0.702	0.892	0.982	0.997	0.049	0.272	0.682	0.876	0.940
DT	0.028	0.196	0.678	0.838	0.915	0.458	0.770	**0.930**	0.990	0.998	0.049	0.260	0.672	0.846	0.914
AT	0.024	0.139	0.650	0.868	0.936	0.451	0.760	0.925	**0.991**	**0.998**	0.042	0.210	0.630	0.861	0.928
CM	0.016	0.109	0.451	0.880	**0.989**	0.084	0.286	0.519	0.739	0.861	0.024	0.112	0.377	0.675	0.836

the information from the series of bitmaps itself (SA) or only the auxiliary information from the raster maps (DC) for attention has a limited contribution to performance. Without integrating both auxiliary-information embedding and polygon-based attention, it is difficult for the model to learn which channels (across bitmaps) and areas (within bitmaps) are more reliable. The results between LOAM and $LOAM$ -DC/SA show the efficacy of using two types of shuffle attention to help the model learn.

We notice that AT+TM has a higher precision with lower recall than AT. This demonstrates our design of using text-pattern matching to rule out false-positive connected components (pixels). On the other hand, DT obtains the best recall in the ablation study. This demonstrates that in the map content, a decent number of pixels belong to a certain map key, but have a different color from the key. Finally, although CM gets the lowest F1 score among the preprocessing components, $LOAM$ -CM shows that this aspect can still contribute to the model. The model can identify the latent styles of input raster maps and map keys with auxiliary-info embedding. Depending on the latent styles, the model exploits the candidate based on color-set matching (CM) to compensate for false positives or false negatives of other candidates. We show this in the first two cases in Figure 2.9.

We then focus on comparing the performance at different percentiles. Regarding precision, LOAM gets the better performance among the ablation study for most cases. $LOAM$ -CD gets the best performance for lower percentiles. This can be attributed to the limited efficacy of including color differencing (CD) when multiple map keys use similar colors in a raster map. It is worth mentioning that color-set matching (CM) has the best precision with the worst recall at the 90th percentile. This shows that CM can contribute extraction results with fewer false positives in some particular cases.

Regarding recall, we notice that metadata preprocessing components such as AT and DT tend to perform better than those based on the polygon-recognition model for higher percentiles. The main reason is that our polygon-recognition model cannot compensate for false negatives, as depicted in the third and fourth case in Figure 2.9.

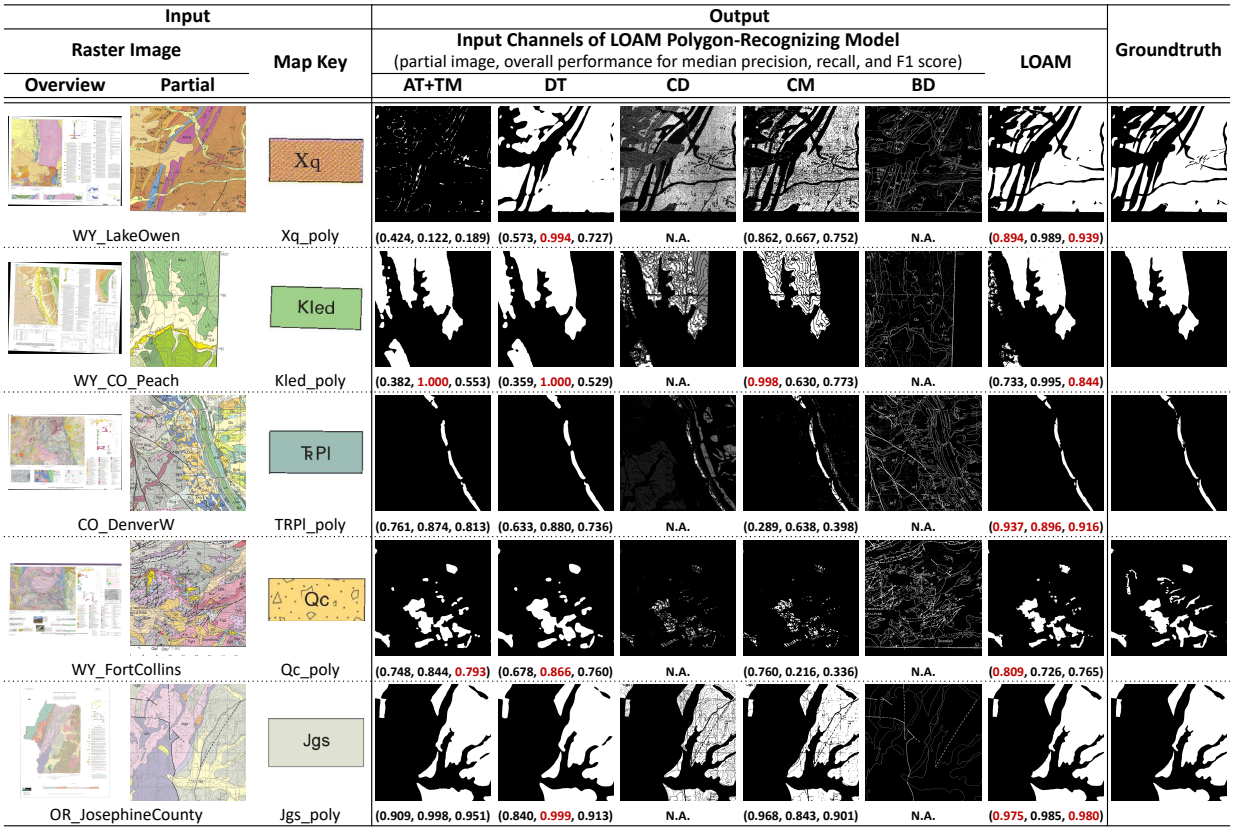


Figure 2.9: Case study for our LOAM and its input channels (corresponding to the outputs from metadata preprocessing) on the testing dataset. The abbreviations of LOAM components are listed as follows. AT: adaptive color thresholding with conditional dilation; TM: text-pattern matching; DT: dynamic color thresholding; CD: color differencing; CM: color-set matching; BD: boundary detection. We highlight the best precision, recall, and F1 score for each case in red. The overall performance for CD and BD is not applicable, as CD is not a binary image and BD does not directly correspond to the targeted polygon feature.

For the F1 score, LOAM performs best at almost all percentiles. The ablation study shows that each component of our approach (including polygon-recognizing model structures, input channels of the model, and metadata preprocessing components) contributes to the final performance.

2.3.5.3 Case Study

We show the case study of our proposed LOAM and its input channels on the testing dataset in Figure 2.9. We demonstrate five cases to provide a qualitative analysis of our design for

the input channels of the polygon-recognition model.

For the first case (map: *WY_LakeOwen*, key: *Xq_poly*), we demonstrate a scenario with a color shift between the map key and the map content. Due to the color shift, adaptive color thresholding (*AT+TM*) cannot fully capture the polygon feature in the map content. However, dynamic color thresholding (*DT*) successfully updates the threshold to include the colors belonging to the targeted polygon and obtains the best recall among input channels. On the other hand, color-set matching (*CM*) manages to identify color-shifted pixels and retains the best F1 score among input channels. Consequently, LOAM observes such a characteristic of color shift based on the two-phase shuffle attention and therefore ignores the output of *AT+TM*. It then focuses on refining the polygon feature of *DT* based on *CM* and *BD*. LOAM retains a precision better than *CM* and a recall slightly worse than *DT*.

For the second case (map: *WY_CO_Peach*, key: *Kled_poly*), we demonstrate a scenario in which the text pattern detection fails. In this case, *AT* overrelaxes the color threshold, and *TM* does not exclude false positive connected components. On the contrary, *CM* retrieves a conservative result with the best precision among the input channels. With the embedding of auxiliary information showing the color distance to other map keys, LOAM suggests refining *AT+TM* and *DT* based on *CM* and *CD*. It compensates for the failure of *TM* and ends up with an F1 score better than all input channels.

For the third and fourth cases (map: *CO_DenverW*, key: *TRPl_poly*; map: *WY_FortCollins*, key: *Qc_poly*), we demonstrate a scenario with a slight color shift due to complex and translucent symbols. All approaches retrieve several false positives in the third case and false negatives in the fourth case. LOAM manages to denoise and remove some false positives in both cases. However, we observe that the improvement regarding false negatives is limited. Since LOAM cannot use *BD* to identify possible boundaries of polygons due to the complex background, it is difficult to recover false-negative pixels if none of the other channels highlights these pixels.

For the fifth case (map: *OR_JosephineCounty*, key: *Jgs_poly*), we demonstrate a scenario

in which there are opaque contours and symbols in the map content. In this case, with dummy-pixel detection, all input channels obtain decent performance, and LOAM retains the precision and F1 score better than all input channels.

2.3.5.4 Running Time Performance

We list the running time performance of each component in LOAM per map on the testing dataset in Table 2.4. The running time is based on the hardware we use to implement and evaluate our approach, as described in Section 2.3.3.

Extracting polygon features for all map keys from the testing dataset takes around 35 hours, since the color-set matching and the text-pattern matching are not on the critical path of each other. By applying parallel processing for the model inference and metadata preprocessing across input raster maps, our approach can execute the whole procedure within 24 hours, the competition time limit.

Nonetheless, our proposed LOAM can extract polygon features from raster maps in a timely manner compared to manual labeling.

2.4 Related Work

Understanding Raster Maps. Maps are often the only source of information about the Earth surveyed using geodetic techniques [16]. Geographic information extraction from maps helps to understand several fields of geography-related research [55].

Extracting different types (e.g., point, line, or polygon) of features from raster maps requires distinct technologies to tackle significantly different technical challenges, as evidenced by the fact that most publications focus on one feature type.

The challenges of extracting line features include: (1) the map keys only demonstrate parts of the line feature, making it difficult to extract a continuous feature from maps; (2) marginal differences due to the scanning process make it difficult to distinguish the targeted

Table 2.4: The running time performance of each component in our approach per map on the testing dataset. Since the model inference is based on the input of a series of bitmaps that are split into a size of 1024x1024 pixels (instead of the original size of the raster map), the corresponding running time performance per map is not applicable.

Stage	Component	Running Time (format: *hh:mm:ss*)			
		Median	Average	Maximum	Overall
Preprocessing of Metadata	Map-Content Detection	0:03:36	0:03:58	0:08:31	1:35:14
	Color Differencing (**CD**)	0:03:16	0:04:53	0:18:22	1:57:14
	Adaptive Color Thresholding (**AT**)	0:01:05	0:01:28	0:04:43	0:35:15
	Conditional Dilation (**AT**)	0:07:41	0:10:26	0:35:41	4:10:25
	Dynamic Color Thresholding (**DT**)	0:01:52	0:02:48	0:10:47	1:07:15
	Text-Pattern Matching (**TM**)	0:07:44	0:30:02	3:25:31	12:00:50
	Boundary Detection (**BD**)	0:00:42	0:00:55	0:03:07	0:21:51
	Color-Set Matching (**CM**)	0:12:23	0:25:08	2:22:09	10:03:19
	Auxiliary-Info Embedding	0:00:18	0:00:20	0:00:43	0:07:52
	Generating Bitmaps for Model	0:01:40	0:02:28	0:08:04	0:59:11
Learning to Recognize Polygons	Model (Training on training dataset)		N.A.		2:53:17
	Model (Inference on testing dataset)		N.A.		11:45:16

lines from other similar lines in the maps (e.g., topographic contours in USGS geological maps also use solid lines). To address the challenges, Chiang and Knoblock [14] apply mean shift to reduce noise in colors and use Hough transform to identify road centerlines based on the colors. Duan et al. [21] adopt a convolutional model to align contemporary vector data with the corresponding line features on historical raster maps. Xia et al. [85] apply transformer with contrastive learning. Despite their accuracy, both approaches require a tailored model for each type of line feature.

The challenges of extracting point features include: (1) maps usually contain much information, making it difficult to distinguish a single point feature from the noisy backgrounds; (2) the symbols representing point features come in different shapes, colors, and sizes. Previous works on point-feature extraction often address single-feature extraction. For instance, Chiang and Knoblock [15] apply image processing and graphics recognition methods to extract road intersections from raster maps. Saeedimoghaddam and Stepinski [65] use a convolutional model to extract road intersections.

Although there is a plethora of work on point or line extraction, one future direction is to exploit the ability of our method to handle input raster maps and keys of diverse styles. With some further research in metadata preprocessing for encoding and embedding maps along with map keys, our approach could synergize with state-of-the-art methods to generalize to multiple point or line features from maps of diverse styles.

Polygon Extraction from Raster Maps. For polygon extraction from raster maps, most research focuses on extracting only one type of feature from a map. For instance, Arteaga [4] applies a series of image-processing techniques in extracting buildings represented as polygons from historical maps. Since there is only one thing to extract from the map, this is similar to a boundary or foreground detection problem.

On the other hand, the U-Net architecture [64] has demonstrated its efficacy in binary segmentation tasks for medical images [69] and single-feature segmentation from historical maps. We et al. [81] integrate U-Net with a transformer to extract water bodies from

historical maps. Since water bodies have fixed representations (e.g., colored in blue) across different maps, their approach does not consider the map key and can only extract one type of polygonal feature (the water bodies). Similarly, some previous research applied U-Net-based model to segment polygon features for buildings [28], roads [32], hydrological features [82], or archaeological features [24] from historical maps. However, all the above approaches to single-feature extraction do not apply to our targeted problem, since they require a tailored model for each feature. Instead, we aim to provide a unified model that can handle all input map keys, which come in different colors, texts, or textures, without retraining the model for each map key.

Geographical Feature Extraction from Images. Some previous research aims to identify polygonal features from sensing data instead of maps. For example, the work of Song and Jung incorporates morphological and filtering techniques to extract buildings from LiDAR data [70]. Xu suggests using terrain ruggedness index and vector ruggedness measurement to help identify flat surface areas [89], while Chen et al. generate a domain knowledge-informed feature based on input LiDAR images and apply a deep-learning model to identify buildings [13]. For remote sensing data, Wang et al. [77] integrate convolutional neural networks with various matching and spectral segmentation models to generate a series of intermediate outputs. They then fuse the intermediate outputs to get the segmentation result of a certain object type.

For multi-feature extraction, Lee et al. suggest using 3D geospatial data with a convolutional model to classify the terrain features [37]. Some previous work formulates polygon extraction into boundary detection and proposes a graph neural network to identify the boundaries of buildings in satellite images [45, 95].

Although the above works are not applicable to our targeted problem due to the nature of single-feature extraction or the usage of external data, some approaches inspire parts of the design of our method. Specifically, we apply morphological analysis during conditional dilation in metadata preprocessing. Meanwhile, we generate an auxiliary-information embedding

based on the input raster map and use this embedding to help recognize the polygons.

Instance Segmentation from Images. From another perspective, we can formulate our target problem into a variation of combining foreground detection with instance segmentation. However, most approaches to these research problems do not apply to our target problem, since they do not conduct extraction based on specified keys and are not fully automated. For example, FreeSOLO [79] is a weak supervised approach based on a bounding box for instance segmentation, and SAM [36] is a method based on zero-shot generalization to identify unfamiliar objects from images. Nevertheless, these approaches inspire our design regarding two-phase attention, informing the model which channels or areas are reliable.

Critical Mineral Assessment Competition. DARPA and USGS held the AI for Critical Mineral Assessment Competition[1] with two challenges: map georeferencing and map feature extraction, in October 2022. The map feature extraction challenge aims to find solutions to extract multiple geological features (including polygons, lines, and points) from raster maps.

The approach that won second place in polygon extraction (team "uncharted") integrates entropy analysis and convolutional neural networks[7]. However, since they have not published their method yet, we do not have enough information to discuss the theoretical differences between our method and theirs.

The approach that won first place in polygon extraction [59] (team "ICM") integrates optical character recognition (OCR), adaptive histogram equalization, and a modified U-Net[6]. The main differences between the team "ICM" [59] and our approach are three-fold. First, they apply OCR instead of text-pattern matching. This allows their approach to retrieve more accurate results in text detection, but their performance may be limited regarding maps with a lower resolution. Second, their approach treats the map key as a prompt and feeds it into a 6-channel U-Net. On the contrary, we treat the map key as a reference to turn the original map into multiple representations. This allows our approach to exploit information from maps in addition to colors. Third, they apply data augmentation such as

re-scaling, channel shuffle, and RGB shift to enhance the generalizability. Instead, we propose the auxiliary-information embedding that implies latent characteristics of the map and keys to enhance the adaptability of our model.

2.5 Summary

This chapter addresses the research problem of extracting geological features from raster maps. We leverage the metadata of the maps that describe the geological features represented as polygons. We presented a metadata-driven approach to use map keys as background knowledge in polygon extraction. Our approach can handle raster maps and map keys that come in different styles. The comprehensive evaluation shows the efficacy of our approach in exploiting background knowledge in raster maps for polygon recognition. Our approach can facilitate raster map vectorization and the investigation of geography-related topics.

However, there are three important problems to address in future work. First, we believe that a more accurate boundary extraction can help determine text labels and prevent text-pattern matching from producing incorrect results. Second, the parameter setting in color-set matching can be adaptive to the latent styles of input raster maps and map keys to improve its performance. Third, we aim to extend the polygon-recognition model to the geological features of other representations (lines and points). Extracting different types of features from raster maps requires distinct technologies to tackle significantly different technical challenges. Therefore, we need some modifications to metadata pre-processing to capture the corresponding information from the raster map.

Chapter 3

Exploiting Polygon Metadata to Recolor Historical Maps

Historical maps often suffer from coloring errors caused by artifacts during map production or scanning. These errors result in color mismatches between important map features (e.g., polygon layers) and their corresponding map keys, which hinder both human interpretation and automated feature extraction. This chapter targets the problem of automatically correcting polygon coloring errors in historical maps using only in-map information, such as the map keys. The challenge lies in the diverse visual representations of map keys and variations in coloring errors, which differ significantly both within and across maps. We propose a machine-learning model that automatically identifies and corrects color inconsistencies between map polygon layers and their visual appearances defined by the map keys on the same map. Our approach leverages polygon metadata, such as map keys describing the visual and semantic properties of each polygon on maps, to detect mismatches in color histograms and representations and recolor the incorrect areas in the map content accordingly. We evaluate our approach on USGS geological maps; it outperforms comparative methods by at least 7.51%. In addition, our approach improves the downstream automated polygon-extraction task by 18.00% in precision.

3.1 Motivation

Historical maps serve as invaluable resources for understanding geography, historical human activities, monitoring environments, and preserving cultural heritages [16, 55]. These maps are often the only available records for certain regions and time periods. However, most historical maps exist only as scanned raster images. The quality and usability of these images are often degraded by coloring errors introduced during map production or scanning. These coloring errors lead to mismatches between polygon features in the map content and their expected appearances defined in the map legend, hindering both human interpretation and automated analysis tasks such as feature extraction and vectorization.

The challenge of recoloring historical maps is three-fold (see Figure 3.1). First, the types of color mismatch vary significantly both within and across maps. Coloring errors and mismatches may arise from scanning artifacts (e.g., creases, shadows), production artifacts (e.g., overlays with shaded relief or elevation models) (see Figure 3.2), or systematic color shifts between the map keys and map content. Second, map keys come in diverse visual styles, including solid colors, textures, or markings. This makes it difficult to precisely detect color mismatches and recolor those identified regions while preserving their original textures or markings in the map content. Third, historical maps often have polygon features intertwined with lines, points, text labels, and basemap elements (e.g, contour lines).

We present ***REPOLISH*** (**RE**coloring via **P**olygon-**O**riented **L**earning with **I**nterpretative **S**pectra in **H**istorical maps), a machine-learning approach that leverages polygon metadata, including map keys and color spectra, to recolor existing historical maps while preserving semantic consistency for the polygons in the map content.

To the best of our knowledge, we are the first to target the problem of automated recoloring of historical maps using only in-map information. Previous works on map feature extraction [49, 59] assume color consistency and do not explicitly address coloring errors or mismatches. Previous works on synthetic map generation [39, 42] target style transformation

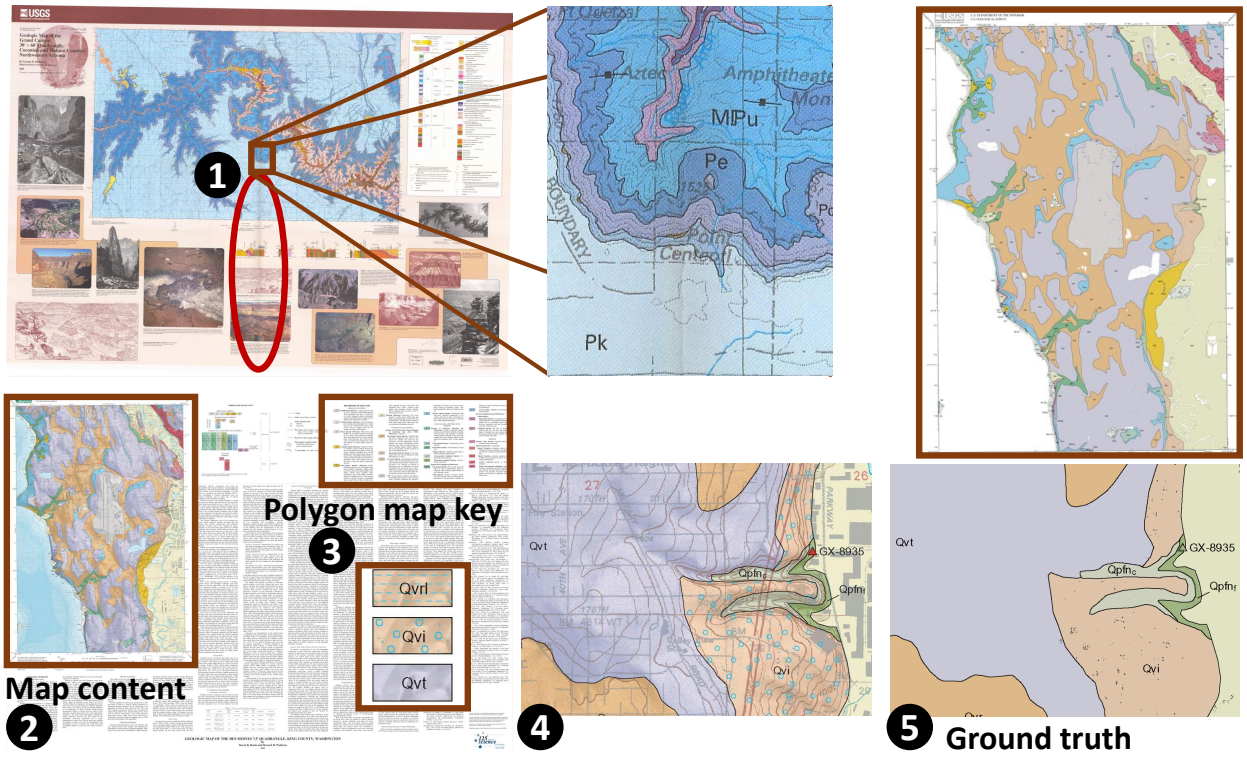


Figure 3.1: Illustration of the targeted map recoloring problem. ① Scanning artifacts. ② Map content with production artifacts. ③ The input includes the polygon map key describing the visual appearances. ④ Existing maps have overlap with shaded relief and other features, leading to color mismatches with the map keys. ⑤ The expected output of recoloring is derived based on colors corresponding to each map key.

at the map level rather than correcting color inconsistencies in existing historical maps. In related areas, line-art recoloring [56, 68] takes structural guidance such as reference images or color scribbles to propagate color to line drawings. While natural image recoloring [9, 92] adjusts image colors based on palette or region guidance. However, these methods from related areas can not handle the complex visual composition of historical maps, where polygon features entangle with lines, points, and text.

Our approach leverages polygon metadata, particularly the map keys that describe the visual and semantic properties of each polygon in the map content, to guide the recoloring process. We reformulate the map recoloring as a constrained HSV-color-space correction problem. REPOLISH combines metadata-driven feature engineering with a generative adversarial network (GAN) to learn to adjust saturation and value while preserving the hue

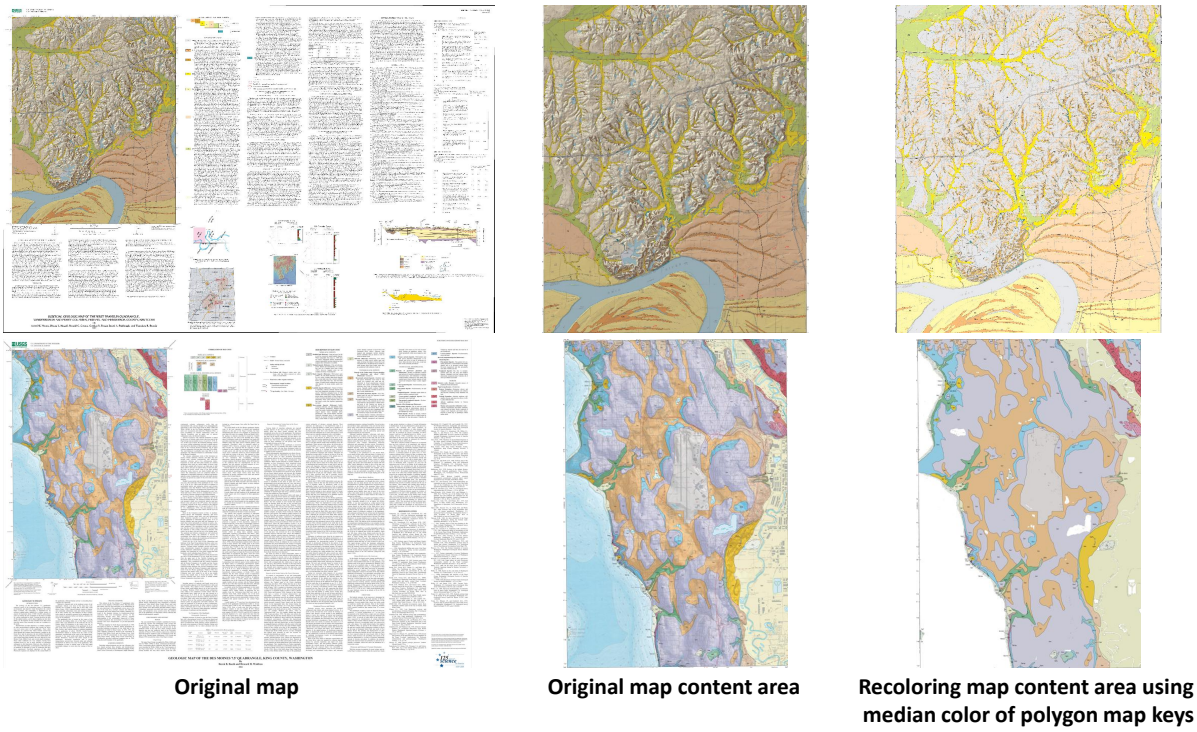


Figure 3.2: Two example cases of maps with significant color mismatches between polygonal features in the map content and the polygon map keys. We present the side-by-side images of the original map content area and the one with its polygon features assigned the median colors of the polygon map keys.

identity defined by the map keys.

We evaluate our approach using historical USGS geological maps [25]. REPOLISH outperforms comparative methods by at least 7.51% in recoloring accuracy, and improves downstream polygon extraction precision by 18.00%. This demonstrates both visual and functional benefits of our approach. Moreover, we present a case study that provides a qualitative analysis of REPOLISH.

To summarize, this chapter presents a novel approach that exploits in-map information about polygon features to automatically identify and correct color mismatches while preserving polygon semantics in the map content.

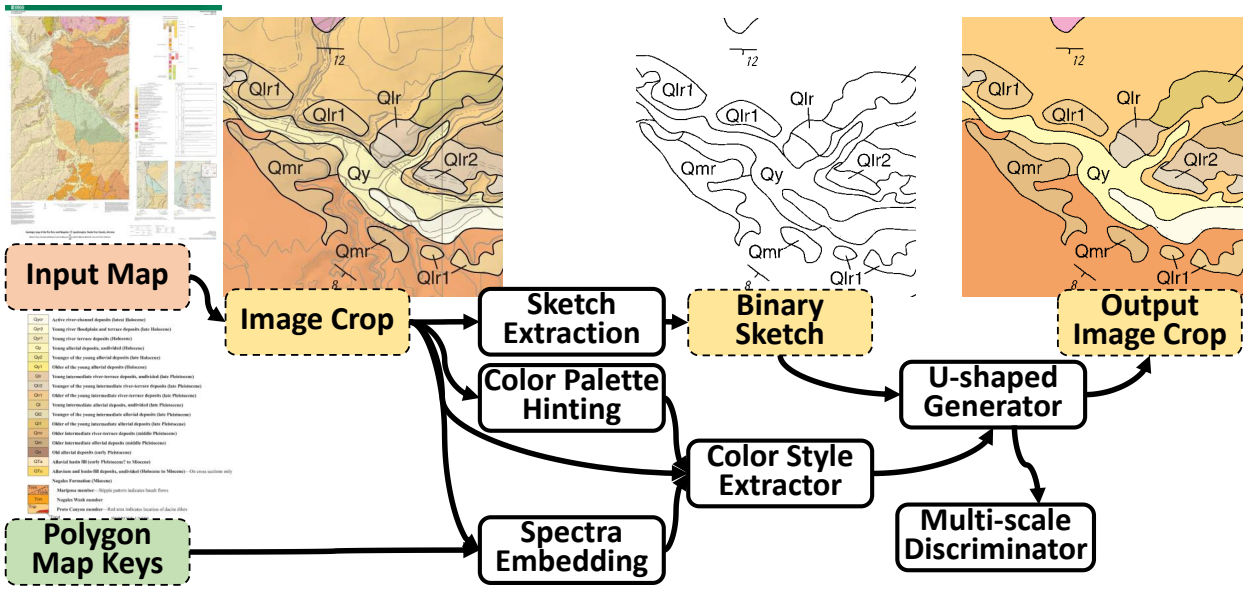


Figure 3.3: The workflow of our approach REPOLISH.

3.2 Approach to Map Recoloring

3.2.1 Problem Definition

Given (1) a raster map with possible coloring mismatches and (2) a list of pixel coordinates indicating the location of map keys in the raster map, the task is to produce a recolored raster map where polygon regions match the intended appearance defined by the key.

3.2.2 Approach Overview

We illustrate the workflow of our proposed REPOLISH in Figure 3.3. REPOLISH is a two-stage approach. First, it exploits polygon metadata to extract structural, color, and statistical representations of both the map content and map keys. Then, it uses these multimodal representations with a GAN-based model to learn color corrections and derive the recolored image.

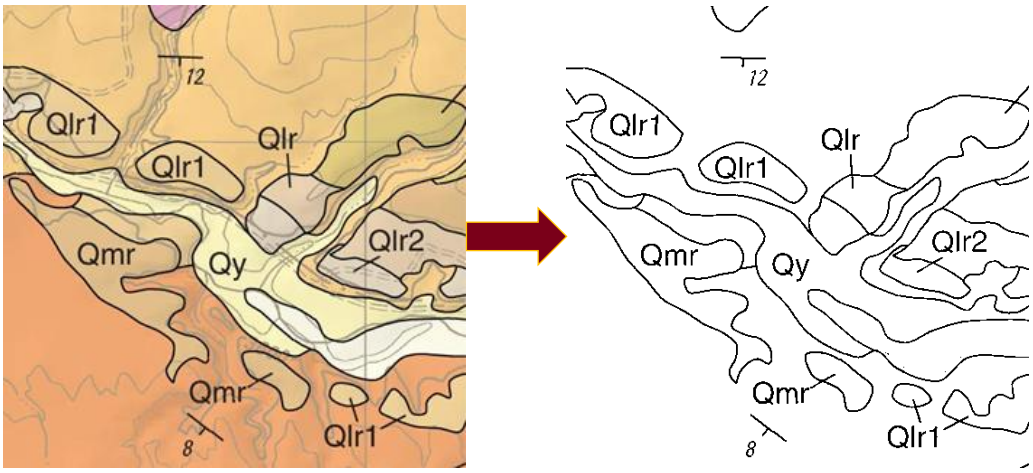


Figure 3.4: An example of sketch extraction in REPOLISH.

3.2.3 Preprocessing of Polygon Metadata

We exploit polygon metadata to extract sketch, color palette, and spectra embeddings from the map content and polygon map keys.

3.2.3.1 Sketch Extraction

We apply heuristics to generate a binary sketch image M_{sketch} by extracting dark lines from the map crop M_{ref}. This sketch serves as the structural backbone for recoloring, supporting the segmentation of polygon features in the map content. We convert the map to CIELAB color space and threshold the L^* channel to isolate dark lines [23]. In addition, we apply Canny-edge [6] and color-gradient detection to identify lines that separate polygons using statistically significantly different colors.

We show an example of sketch extraction in Figure 3.4.

3.2.3.2 Color Palette Hinting

We derive a colorgram embedding E_{cg} from the polygons associated with each map key. We apply K-Means clustering to store up to 16 colors per image crop. This captures the dominant target color distribution and serves as an explicit palette hint to the model.

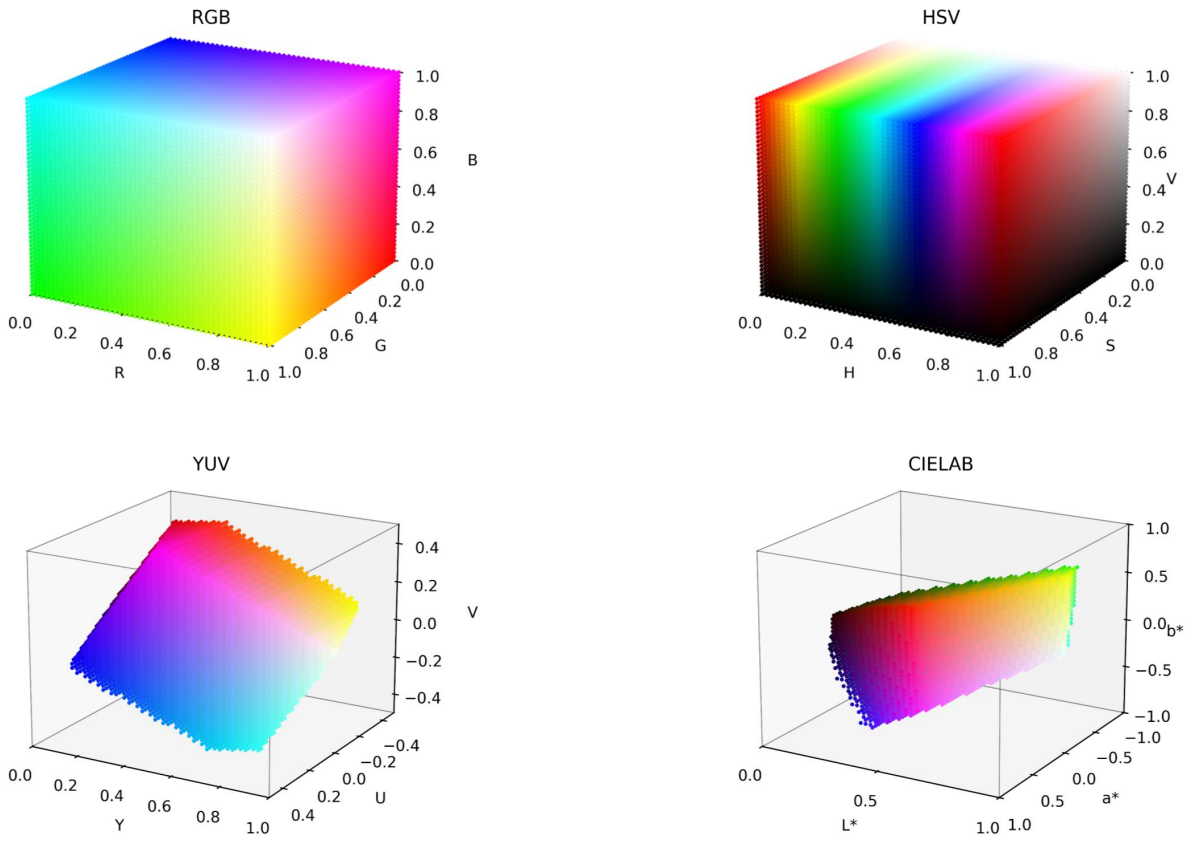


Figure 3.5: The adopted color spaces for palette hinting and spectra embedding in REPOLISH.

3.2.3.3 Interpretative Spectra Embedding

The color spectra embedding E_{sp} captures the color histograms for the overall map content and the map keys for polygon features, respectively. This spectra embedding encodes the broader semantic context of polygon features in the map, highlighting the coloring difference between map content and map keys. We use binned histograms for Hue (from HSV), L^* (from CIELAB), and U/V (from YUV) channels, building a perceptual signature of color distributions.

We show the range and corresponding visual representations of the adopted color spaces, including RGB, HSV, CIELAB, and YUV, for color palette hinting and interpretative spectra embedding in Figure 3.5.

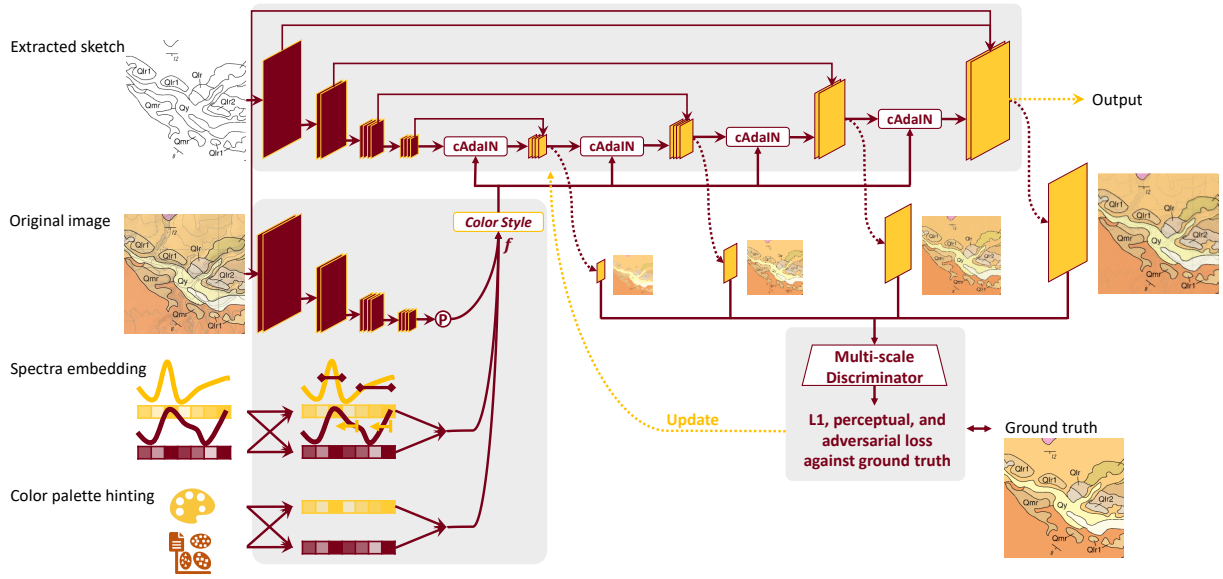


Figure 3.6: The multi-scale recoloring model structure in REPOLISH.

3.2.4 Using Metadata to Learn to Recolor Maps

We feed the processed polygon metadata into a GAN-based model to learn to recolor the maps. We depict the multi-scale recoloring model structure in Figure 3.6.

3.2.4.1 Color Style Extractor

We apply a ResNet-based encoder to process M_{ref}, with two branches handling the E_{cg} and E_{sp}, respectively, capturing visual and metadata-driven map-style cues.

3.2.4.2 Learning-based Recoloring Generator

As scanning or production artifacts often lead to color shifts only in saturation and value rather than hue, we feed E_{cg} and E_{sp} to the generator to adaptively adjust the saturation and value in M_{ref}. So that the U-Net-based generator can learn to update M_{ref} with M_{sketch} treated as a mask for color corrections.

3.2.4.3 Multi-Scale Discriminator and Loss Functions

We use a multi-scale PatchGAN discriminator [11, 29, 56] to ensure that the generated images are both visually plausible and consistent with the style of historical maps. The discriminator operates at multiple image resolutions to capture both fine-grained texture details and global color consistency. This is crucial for downstream tasks that require distinguishable and consistent polygon topology, such as polygon extraction. Precisely, we downsample the input image to multiple scales and apply the same PatchGAN discriminator to each scale, encouraging the generator to produce realistic outputs across different spatial resolutions.

We use a combination of adversarial loss and reconstruction loss for the generator. The adversarial loss encourages the generator to produce outputs indistinguishable from real recolored maps. On the other hand, reconstruction loss consists of L1 loss and perceptual loss, enforcing fidelity to the ground truth. The generator objective is a weighted sum of the adversarial and reconstruction losses:

$$\mathcal{L}_{total} = \mathcal{L}_G + \lambda_{rec}(\lambda_{content}\mathcal{L}_{content} + \lambda_{perc}\mathcal{L}_{perc}) \tag{3.1}$$

where $\mathcal{L}_G$ is the adversarial loss, $\mathcal{L}_{content}$ refers to the L1 loss, and $\mathcal{L}_{perc}$ indicates the perceptual loss. λ are their weights.

This multi-scale discriminator design ensures that both local characteristics, such as textures or markings within polygon features, and global color histogram, particularly polygon-level color consistency, are preserved in the recolored output.

3.3 Evaluation

3.3.1 Dataset

We use USGS geological maps [25], as in Chapter 2, for evaluation. Each map is a raster image and has a corresponding JSON file that records the bounding box of each map key in

the map content. This series of geological maps has training, validation, and testing datasets. We use 70 maps from the training and validation datasets to train our model and use all 32 maps from the testing dataset to evaluate the performance of our approach. Among the 32 testing maps, at least half have noticeable or significant coloring errors and mismatches.

3.3.2 Evaluation Metric

To quantitatively evaluate recoloring accuracy, we adopt two metrics: peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

PSNR measures the pixel-wise reconstruction quality between the generated image and ground truth based on the mean squared error (MSE) between the two images. PSNR is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{\text{MSE}} \right) \tag{3.2}$$

where MAX is the maximum possible pixel value (255 for 8-bit images).

On the other hand, SSIM evaluates the perceptual similarity between two images by considering luminance, contrast, and structural information. SSIM is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.3}$$

where μ_x and μ_y are the means of images x and y, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance between x and y, and C_1, C_2 are small constants to stabilize the division.

For both PSNR and SSIM, a higher value indicates better similarity between the generated image and the ground truth. However, SSIM is bounded to 1.

In addition, we evaluate the effect of recoloring on downstream tasks, particularly polygon extraction. We adopt the weighted F1 score (precision and recall) [25] to evaluate the accuracy of extracted polygons. By using the same polygon-extraction model [49] with different input images (before/after recoloring), we compare the polygon-extraction accuracy accordingly.

3.3.3 Evaluation Setting

We implement REPOLISH in PyTorch. We set a batch size of 6, Adam optimizer, a learning rate of 1e-04, $\lambda_{content} = 10$, $\lambda_{perc} = 1$, $\lambda_{rec} = 100$, and split 20% of the training dataset for validation. We train the model for 60 epochs and select the model with the best performance on the validation set to proceed to the test dataset. Since there is no overlap between testing, training, and validation sets in the public USGS dataset (benchmark), this prevents contamination of the datasets or further tuning based on the testing dataset.

To address data imbalance, we only include image crops (after splitting into 512×512 pixels with 32-pixel overlaps) with a corresponding ground truth that contains more than 50% non-background pixels and 5 distinct polygon features in the training data.

We implement all methods in Python on a Gigabyte workstation equipped with an Intel Xeon w9-3595X CPU at 2.00 GHz, 512 GB RAM at 4800 MT/s, and two NVIDIA A6000 GPUs.

3.3.4 Comparative Method

We select two state-of-the-art methods for solving problems with similar settings as comparative methods. (a) Reference-based Recoloring [56] is a GAN-based approach that exploits structure-oriented color styles from the original image to support recoloring; and (b) Color-set-based Recoloring (a variant of [49]) is a heuristic approach that recolors a pixel based on the lowest color-set distance across all the map keys in RGB and HSV color spaces.

3.3.5 Evaluation Result

3.3.5.1 Overall Performance

We show the overall performance in terms of the PSNR and SSIM on the testing dataset for our proposed REPOLISH against comparative methods in Table 3.1. Our approach achieves a performance of 21.652 in PSNR and 0.930 in SSIM, outperforming state-of-the-art methods

Table 3.1: Overall performance in terms of PSNR and SSIM. We report the average performance with standard deviation.

Method	PSNR (↑)	SSIM (→1)
REPOLISH (Ours)	**21.652** ± 3.508	**0.930** ± 0.048
Reference-based Recoloring	20.015 ± 2.821	0.865 ± 0.061
Color-set-based Recoloring	16.451 ± 2.200	0.459 ± 0.043

by 7.51%. In addition, by treating the images recolored based on our approach as the input to the downstream polygon-extraction model, we observe an improvement of 18.00% in precision and 3.60% in F1 score for maps with significant coloring errors and mismatches.

We notice that reference-based recoloring is able to recognize the dominant color usage with its spatial correlation in the image crop. However, it fails to ensure consistency and tends to recolor a polygon into multiple color patches due to translucent symbols. Color-set-based recoloring guarantees the color consistency between map content and map keys. However, it is unable to maintain polygon topology, which results in noisy pixels within polygons.

3.3.5.2 Case Study

We present a case study showing input, REPOLISH output, and polygon extraction results before/after REPOLISH's correction in Figure 3.7. With recoloring, downstream polygon extraction tends to have better precision for maps with significant color mismatches due to overlap with shaded relief.

The quantitative and qualitative results demonstrate that REPOLISH can address color inconsistency in historical maps with sufficient training data from similar cartographic styles.

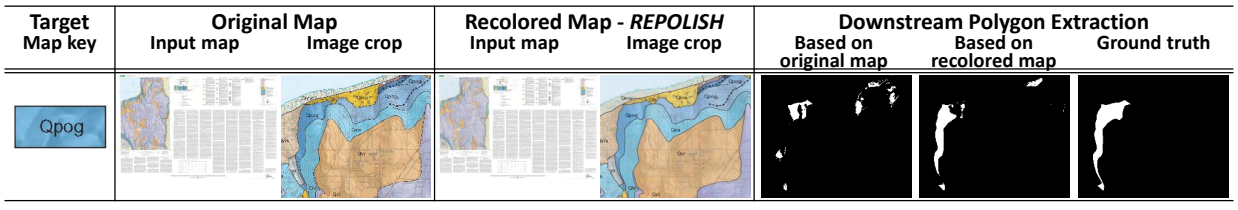


Figure 3.7: Case study for our REPOLISH. We provide the polygon features extracted based on the map before/after REPOLISH. The adopted extraction model is presented in Chapter 2.

3.4 Related Work

Historical map understanding. Historical maps are often the only source of information for studying historical geography and supporting modern applications [16, 55]. Previous research on map feature extraction [49, 59] assumes color consistency and does not explicitly address coloring errors. Some previous works generate synthetic historical maps [39, 42] to simulate particular map styles or augment training data, but they do not address color inconsistencies in existing maps. In contrast, our approach targets color transformation guided by polygon metadata and structural cues.

Line-art recoloring. Previous research on line-art recoloring exploits various guidance to propagate colors to particular areas. Image-based approaches [56, 68] transfer color styles from reference images, while scribble-based approaches [8, 17, 26] use spontaneous markings combined with diffusion or adversarial models. These methods rely on explicit and strong alignment between the input image and guidance, whereas our method leverages polygon metadata to learn to recolor maps with complex overlapping symbols.

Natural image recoloring. Previous research on natural image recoloring adjusts colors by using palette [9, 92] or region guidance [62, 94]. These works apply multimodal frameworks or optimization techniques to preserve perceptual quality, and target visual enhancement rather than enforcing strict correspondence to the keys. Moreover, the perceptual segmentation used in these methods is unreliable for historical maps due to overlapping features [49]. Our method uses palette and statistical cues derived from polygon metadata to achieve semantically consistent recoloring.

3.5 Summary

We target the problem of correcting color mismatches in historical maps using only in-map information. We present REPOLISH, a metadata-driven learning approach that leverages

polygon-oriented color guidance to recolor maps while preserving the semantic consistency of polygons in the maps. Our approach combines structural, palette, and statistical cues to correct complex coloring mismatches in existing historical maps. The evaluation shows that REPOLISH outperforms state-of-the-art methods in visual quality of recolored maps and improves accuracy in downstream polygon extraction. Our approach can facilitate map correction and support applications in map digitization.

The future work lies in generating synthetic data to broaden to a wider variety of historical map styles.

Chapter 4

Exploiting Polygon Metadata to Colorize Draft Maps

Black-and-white draft maps and field sketches, produced during fieldwork or surveys, often contain dense handwritten annotations overlaid on monochromatic basemaps. Although interpretable in grayscale, the lack of color makes it difficult to visually distinguish overlapping or adjacent thematic regions, especially when boundaries are unclear and annotation styles vary. However, colorizing these draft maps is labor-intensive but essential, as they may be the only source of detailed geographic or environmental information for certain regions and time periods. This hinders both human interpretation and downstream tasks such as map digitization and critical mineral resource assessment. We target the problem of automated colorization of draft maps. The challenge lies in interpreting noisy visual cues from uncolored sketches and assigning appropriate colors according to their semantic categories. We propose a novel machine learning approach that exploits polygon metadata, including map keys that explicitly define thematic features and implicitly suggest their intended colors, along with the semantic interpretation of the sketch content in the maps. We evaluate our method on USGS draft geological maps; it outperforms comparative methods by 15.66%. In addition, our approach improves downstream polygon-extraction performance by 8.55% in F1 score.

4.1 Motivation

Historical draft maps preserve a unique record of environmental, geographic, and geological observations that often predate and are unavailable in modern digital surveys. While many published historical map series are available in color-coded formats, a portion of archival collections still exists only as monochromatic draft versions. For instance, some of the USGS geological maps in the training set introduced in Chapter 2 are monochromatic draft maps, and no colorized versions of USGS geological maps covering neighboring regions and times have ever been published. These drafts typically contain hand-drawn boundary lines for the polygon features (thematic regions) and handwritten annotations overlaid on contour basemaps. Without color encoding to separate regions, adjacent polygon features are difficult to differentiate, especially in areas where boundary lines are faint, incomplete, or interwoven with topographic details. This lack of visual separation creates a bottleneck for expert interpretation and automated vectorization workflows.

The challenge of colorizing these draft maps is threefold. As shown in Figure 4.1, the draft maps, or map sketches, are inherently noisy and may lack closed boundaries, making traditional segmentation-based methods unreliable (① and ② in Figure 4.1, and Figure 4.2). Furthermore, the choice of colors in thematic mapping often follows established conventions. For instance, the color assignment for geological maps is often based on agency convention, such as rock type and geological time period [71] (③ and ④ in Figure 4.1). Although these conventions are non-mandatory and flexible, they are often respected to ensure interpretability and consistency. In addition, most draft maps do not explicitly label every region with its corresponding polygon features, and visual variations in handwriting and boundary style add further ambiguity. As a result, automatic colorization requires both structural understanding of the sketch and semantic reasoning about the intended map content.

To address these challenges, we present a metadata-driven machine-learning approach named ***SHADING*** (**S**emantic–**H**armonic **A**chromatic **D**raft **I**nterpretation and **N**arration

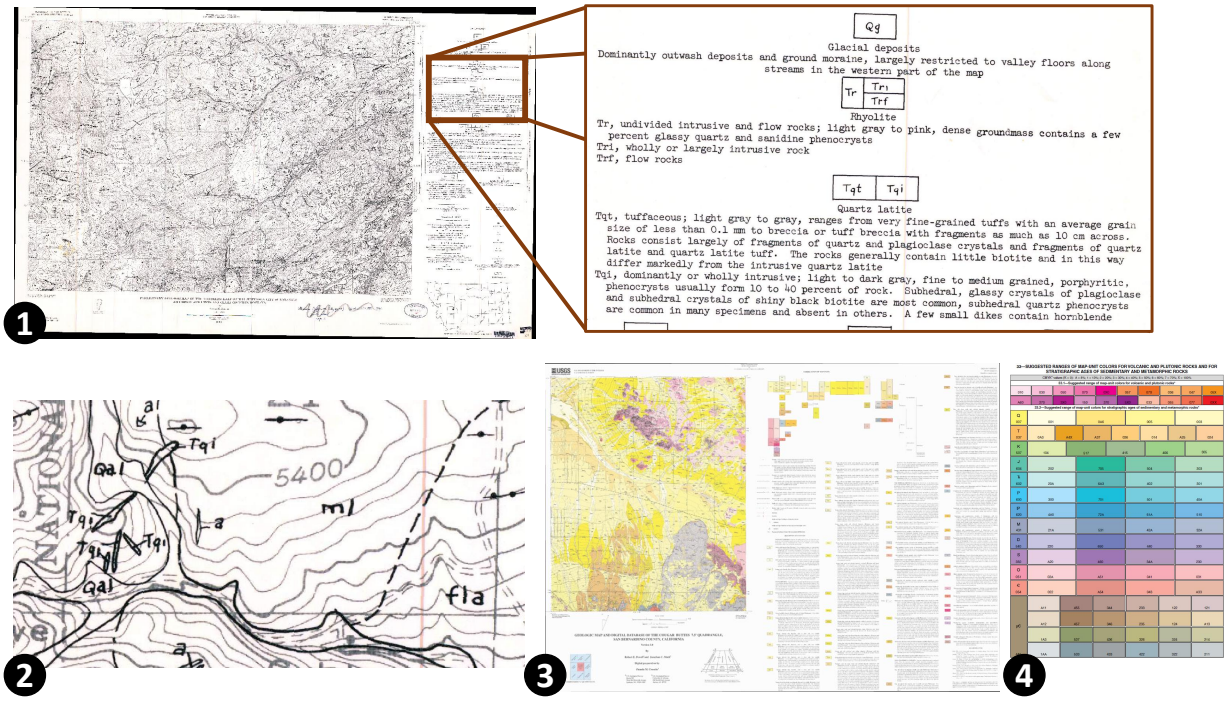


Figure 4.1: Illustration of the targeted map colorization problem. ① A draft geological map. ② The annotations for polygon boundaries, text labels, and contour lines in the draft map can be interwoven. ③ An example of a colorized geological map, in which there is a dominant color of yellow for the polygon features, representing sedimentary rock groups. ④ Agencies such as USGS often have a guideline for color encoding for the polygon features in geological maps.

for **G**eological maps). SHADING leverages polygon metadata derived from the input sketch, including hierarchical superpixel-based region maps and semantic tag embeddings extracted from the visual appearance of map keys. It uses a conditional generative model to learn to map the monochromatic sketch to a fully colorized output, guided by spatial constraints and semantic information.

To the best of our knowledge, this is the first work to target the problem of automated colorization of monochromatic draft maps with an implicit color schema. Previous research in line-art colorization [34, 56, 78] and natural image colorization [80] do not address this task, as they lack the ability to enforce the strict region-based semantics and spatial consistency required for thematic maps.

Evaluation results show that SHADING outperforms comparative methods in colorization

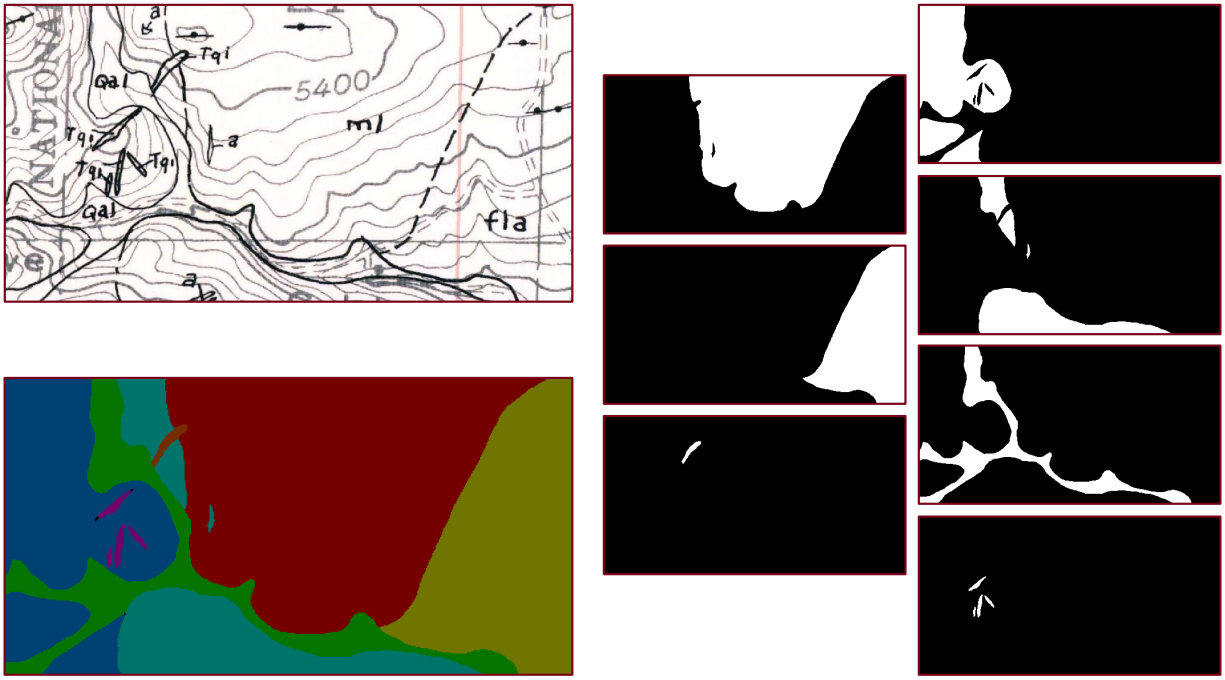


Figure 4.2: An example case of part of a draft map and its corresponding polygon features (thematic regions), including binary masks and a color-coded image.

accuracy by 15.66% and improves downstream polygon extraction performance by more than 8.55%.

To summarize, this chapter presents a novel metadata-driven approach that learns to effectively integrate the spatial and semantic properties for draft map colorization.

4.2 Approach to Map Colorization

4.2.1 Problem Definition

Given (1) a monochromatic draft map and (2) a list of pixel coordinates indicating the location of map keys in the draft map, the goal is to generate a colorized map where each polygonal feature is assigned an appropriate and spatially consistent color, harmonized with its corresponding map key. The output must preserve the flat-color cartographic style of thematic maps while respecting the semantic relationships between polygon features (thematic regions) and their visual representations.

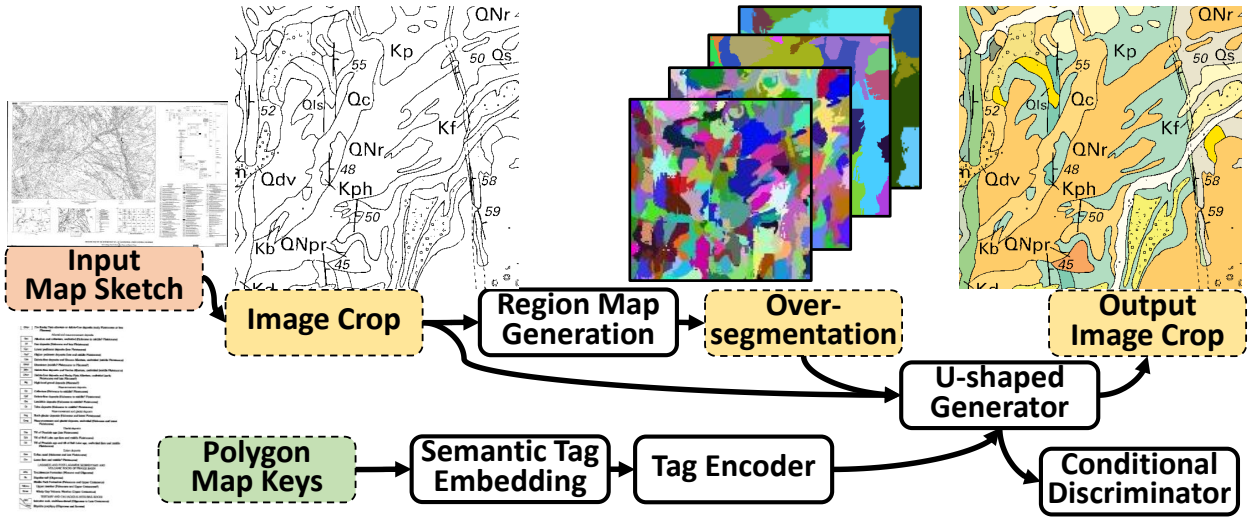


Figure 4.3: The workflow of our approach SHADING.

4.2.2 Approach Overview

We formulate the map colorization problem as a conditional image generation task guided by polygon metadata. The core idea of SHADING is to leverage both high-level semantic cues (e.g., polygonal thematic categories) and low-level spatial constraints (e.g., polygon boundaries) to produce colorized maps faithful to thematic conventions [75] or agency guidelines [71].

As illustrated in Figure 4.3, we design a conditional Generative Adversarial Network (cGAN) framework where a U-Net-based generator is conditioned on polygon metadata embeddings. In addition, we exploit a cross-attention mechanism to align encoded sketch features with semantic tag information. For the discriminator, we design a region consistency loss that encourages the colorized maps to maintain consistent flat coloring within each polygon, conforming to the polygon topology in thematic maps.

4.2.3 Preprocessing of Polygon Metadata

To support guiding the generator, we exploit polygon metadata from the input sketch (draft map) via both map content and map keys. For the map content, we generate a region map

using hierarchical superpixel segmentation based on the sketch lines. For map keys, we extract semantic tag embeddings from their visual appearance.

4.2.3.1 Region Map Generation

We generate the region map from the map content through a hierarchical integration of the SLIC superpixel [1]. The intuition is that the thematic regions in draft maps exhibit varying degrees of visual and spatial coherence. We may recognize some units primarily by boundaries in the sketch, but rely more on overall shape or local textures to identify other units.

Accordingly, we apply fast-SLIC segmentation [33] at four hierarchical levels, varying two key parameters: target number of superpixels and shape compactness. For higher levels, we aim to derive fewer superpixels to encourage broader, color-oriented segmentation. In contrast, lower levels target a larger number of superpixels to capture finer shape details that follow boundaries of the input sketch.

By varying these parameters, we obtain both color-oriented and shape-oriented segmentation across levels. We then reconcile these segmentations by fusing them into a unified oversegmentation, in which each pixel is assigned to a region that is consistent across levels. The result is an integer-labeled region map, used as a hard spatial constraint during learning, where pixels within the same region are expected to receive the same color in the output. This enforces the flat-color style of thematic maps and improves the robustness against sketch noise.

While reconciling segmentations into a single region map for oversegmentation, we preserve the original segment identifier in the region map to support efficient indexing in subsequent steps. We illustrate the schematic diagram of this region map generation process in Figure 4.4.

4.2.3.2 Semantic Tag Embedding

We embed map keys' visual appearance and semantic identity, crucial to determining their colors. We present a schematic diagram of this semantic tag embedding in Figure 4.5.

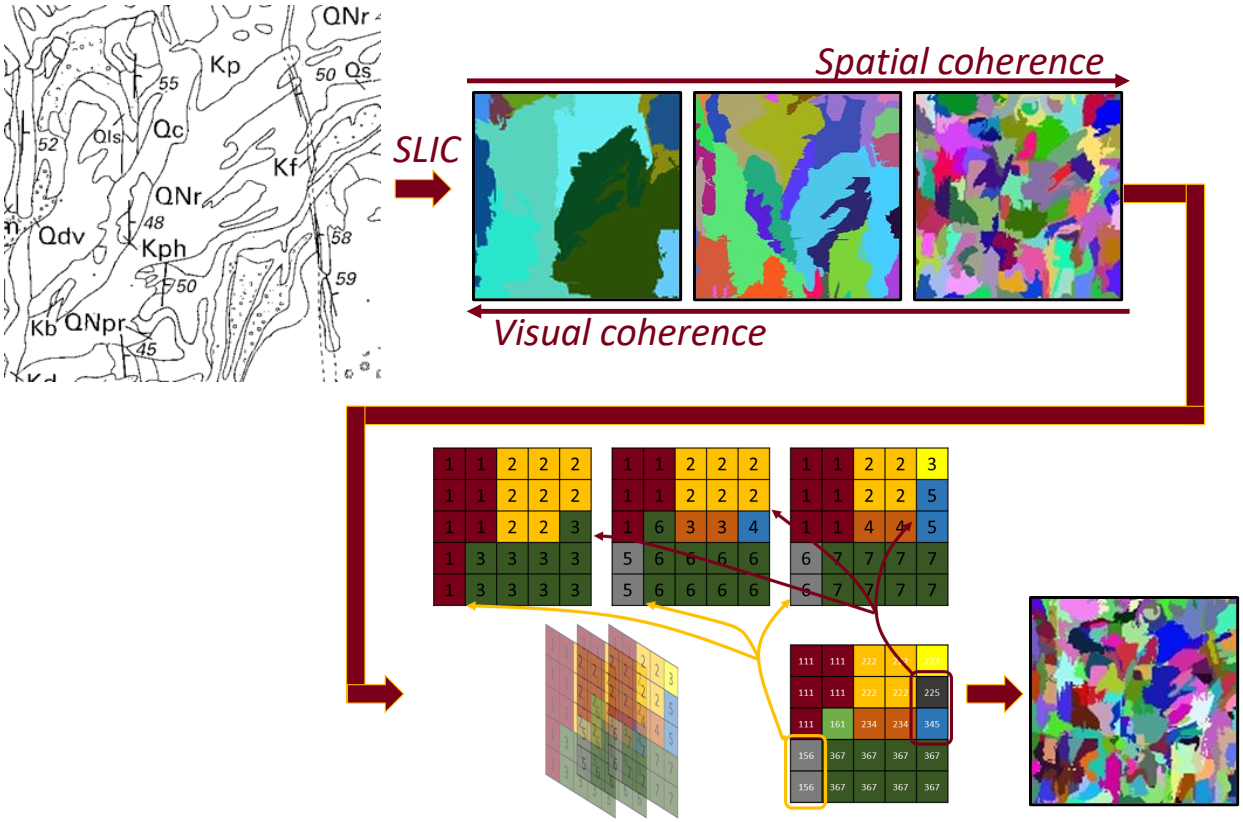


Figure 4.4: A schematic diagram of the region map generation in SHADING. We apply fast-SLIC at hierarchical levels to address spatial or visual coherence. The segmentations are then reconciled into a region map for oversegmentation, with the indexing back to each SLIC segmentation preserved.

For each map key, we identify its centroid of visual pattern and apply pattern filtering with 64 convolution kernels of various sizes and patterns (lines, diagonals, etc.). From each filtered output, we extract the first four magnitudes of the Fast Fourier Transform (FFT) components, yielding an 8-dimensional vector per kernel.

Concatenating across all kernels produces a 512-dimensional semantic tag embedding per map key. This embedding captures multi-scale structural and textural patterns and serves as a compact representation of the map key's visual appearance. In addition, we apply a boolean mask to indicate which tags are active so that the model can handle an arbitrary number of map keys in a map.

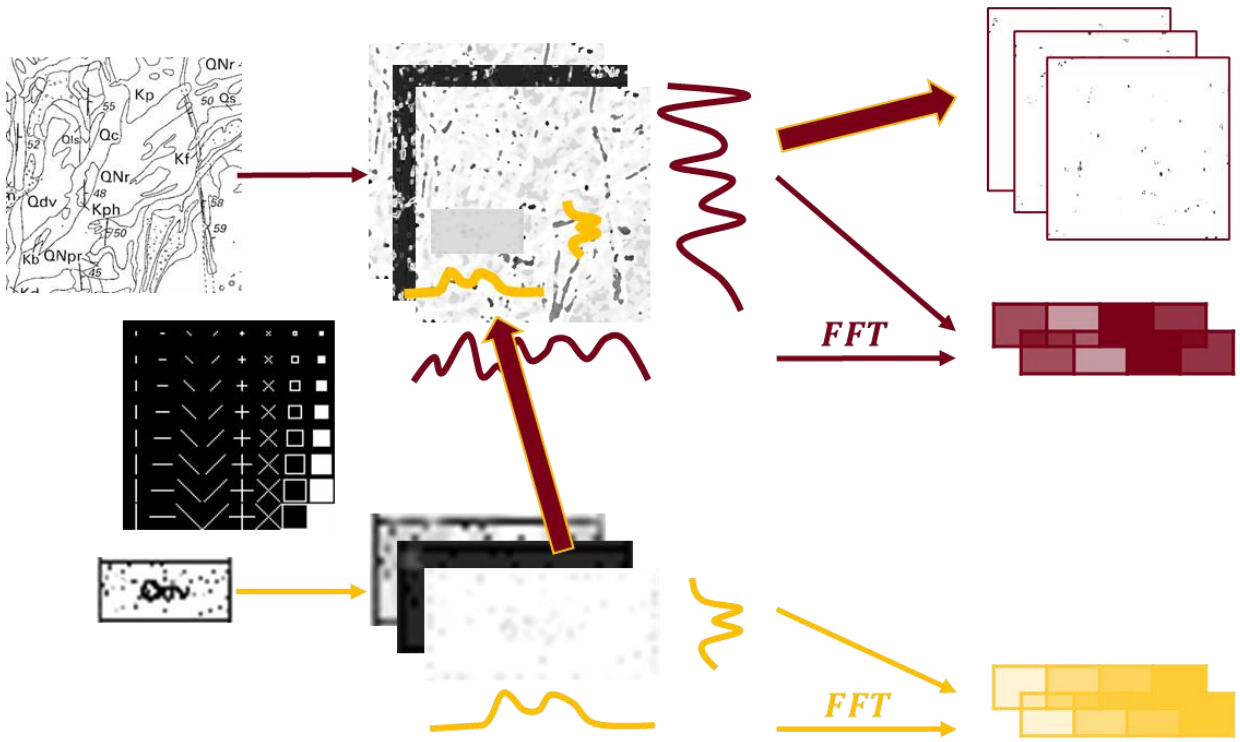


Figure 4.5: A schematic diagram of the semantic tag embedding in SHADING.

4.2.4 Using Metadata to Learn to Colorize Maps

We feed the processed polygon metadata into a cGAN framework composed of three components: a tag encoder, a conditional generator, and a conditional discriminator. We depict the conditioned colorization model structure in Figure 4.6.

4.2.4.1 Tag Encoder

The tag encoder is a multi-layer perceptron that processes the semantic tag embeddings of the active map keys to produce (a) style vectors representing global color style and (b) attention features representing spatial alignment between sketch features and semantic intent.

4.2.4.2 Conditional Colorization Generator

The generator is a U-Net architecture with skip connections. The encoder downsamples the sketch into hierarchical feature maps. The decoder upsamples these maps to produce

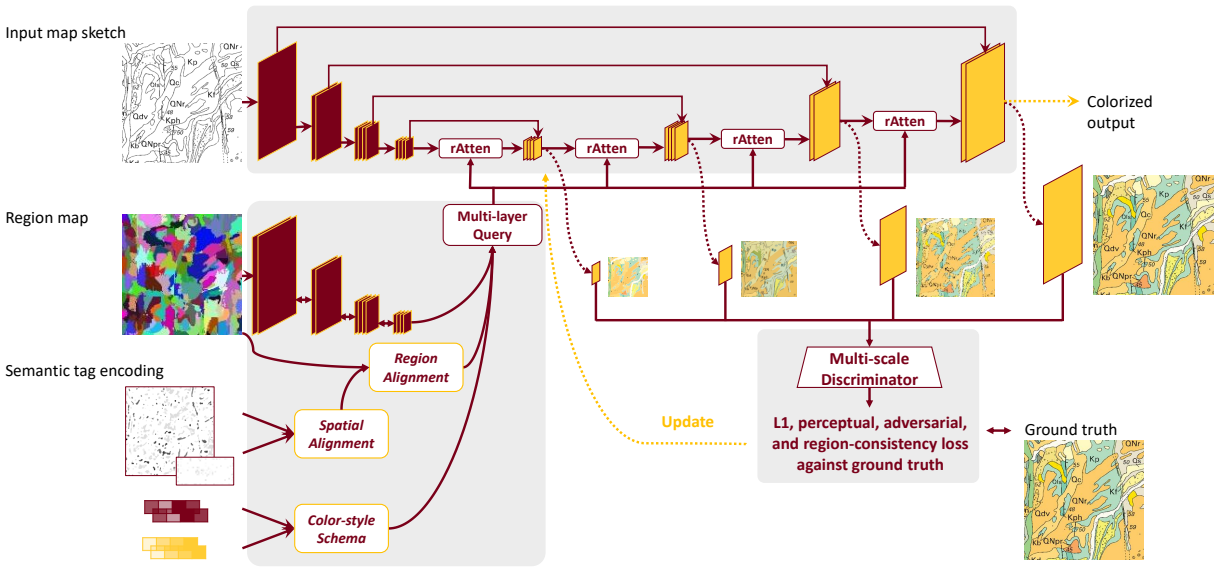


Figure 4.6: The conditioned colorization model structure in SHADING.

a colorized output. This architecture supports both global semantic guidance and local structural fidelity, allowing SHADING to colorize maps from coarse to fine scales.

We employ a cross-attention mechanism to enhance spatial correspondences between the regions in the sketch and the map keys. The sketch feature maps act as queries to direct attention towards relevant information belonging to the tag attention features. These tag features then serve as both keys (identified semantic concepts) and values (estimated colors). This allows the model to learn to identify where each semantic concept is located within the sketch.

4.2.4.3 Conditional Discriminator and Loss Functions

We adopt a multi-scale PatchGAN-based Conditional Discriminator [56] that receives both the image (real or generated) and the corresponding semantic tag embeddings. This encourages the generator to produce outputs that are not only visually realistic but also semantically consistent. The total loss function is:

$$\mathcal{L}_{total} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{region}\mathcal{L}_{region} \tag{4.1}$$

where $\mathcal{L}_{adv}$ is the adversarial hinge loss from the discriminator. $\mathcal{L}_{rec}$ combines L1 and perceptual losses to ensure fidelity to ground truth. $\mathcal{L}_{region}$ is the region consistency loss, which penalizes color variance within each region of the region map, explicitly enforcing flat-color styling.

4.3 Evaluation

4.3.1 Dataset

We use USGS geological maps [25] for evaluation. In the dataset, each map is a raster image and has a corresponding JSON file that records the bounding box of each map key in the map content. We remove colors from the map to derive the monochromatic sketch, with the original colorized image treated as the ground truth. This series of geological maps has training, validation, and testing datasets. After removing maps in which no patterns are used in map keys, we take 30 maps from the training and validation datasets to train our model and use 16 maps from the testing dataset to evaluate the performance of our approach.

4.3.2 Evaluation Metric

We select two evaluation metrics: peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

PSNR measures the pixel-wise reconstruction quality between the generated image and ground truth according to the inverse mean squared error (MSE) between the two images.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{\text{MSE}} \right) \tag{4.2}$$

where MAX is the maximum possible pixel value (255 for 8-bit images).

SSIM is the perceptual similarity between two images based on luminance, contrast, and structural information.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.3}$$

where μ_x and μ_y are the means of images x and y, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance between x and y, and C_1, C_2 are small constants to stabilize the division.

For both PSNR and SSIM, a higher value indicates better similarity between the generated image and the ground truth.

In addition, we evaluate downstream performance by feeding colorized maps to a polygon extraction model [49] and examining the improvement in F1 score. This evaluation emphasizes accurate and consistent color assignment within map content rather than deriving distinguishable colors among polygon map keys.

4.3.3 Evaluation Setting

For SHADING, we set a batch size of 6, Adam optimizer, a learning rate of 1e-05, and split 20% of the training dataset for validation. Loss weights are empirically set as $\lambda_{adv} = 2.0$, $\lambda_{rec} = 4.0$, $\lambda_{region} = 1.0$. The model is trained for up to 80 epochs, with early stopping based on the validation PSNR. Since there is no overlap between testing and training (including the validation set) datasets, this prevents contamination of the datasets or further tuning based on the testing dataset.

We implement all methods in Python on a Gigabyte workstation equipped with an Intel Xeon w9-3595X CPU at 2.00 GHz, 512 GB RAM at 4800 MT/s, and two NVIDIA A6000 GPUs.

4.3.4 Comparative Method

We compare SHADING against an existing method targeting line-art colorization, the modified Tag2Pix [34], and a heuristic method based on pattern matching.

Table 4.1: Overall performance in terms of PSNR and SSIM. We report the average performance with standard deviation.

Method	PSNR (↑)	SSIM (→1)
SHADING (Ours)	**19.339**	**0.798**
Modified Tag2Pix	10.997	0.255
Pattern Filtering	16.721	0.356

4.3.5 Evaluation Result

4.3.5.1 Overall Performance

We show the overall performance in terms of PSNR and SSIM on the testing dataset for our proposed SHADING against comparative methods in Table 4.1. Our approach achieves a performance of 19.339 in PSNR and 0.798 in SSIM, outperforming state-of-the-art methods by 15.66%. In addition, by treating the colorized images as the input to the downstream polygon-extraction model, we observe an improvement of more than 8.55% in F1 score.

Due to the inconsistent amount of textual information that can be extracted from the map legend in many of the draft maps, the performance of text-oriented approaches such as Tag2Pix is limited. On the other hand, although pattern matching can partially identify the textual representation from the map content, it struggles to preserve the polygon topology.

4.4 Related Work

Historical map understanding. Historical maps provide essential geographic and geological information for supporting modern applications such as critical mineral assessment [75]. Previous works on polygon feature extraction from raster maps [49, 59] exploit polygon metadata but require fully colorized maps and do not target colorization. Other previous works [39, 42] generate synthetic maps to augment training data or transform map styles, but do not address the colorization of existing monochromatic draft maps. In contrast, this work targets automatic colorization of draft geological maps guided by polygon metadata

and structural cues.

Line-art colorization. Line-art colorization focuses on assigning colors to black-and-white line drawings, typically guided by reference images or auxiliary input. Image-based methods [56] extract color styles from reference images, while region-based approaches [78] propagate colors within predefined regions. Tag-based frameworks [34] use sparse text tags to guide colorization. These methods rely on strong alignment between input sketches and the guidance. However, they do not address the semi-mandatory semantic interpretation of keys or the flat-color consistency within thematic regions required for thematic map colorization.

Natural image colorization. Natural image colorization methods aim to restore colors to grayscale images based on statistical priors or user input. Palette-based approaches [80] use learned color palettes to guide colorization, while other methods employ global or local color hints. These techniques focus on perceptual realism but lack the ability to enforce strict region-based consistency or adhere to predefined color semantics, which is critical for thematic maps where each polygon must be mapped to a specific key-defined color.

Image colorization and recoloring. Colorization and recoloring target different tasks. Recoloring aims to correct color inconsistencies or errors in existing color images [9], typically restoring intended colors based on reference palettes or map keys. In contrast, colorization targets the transformation of monochromatic or grayscale images into colorized outputs. This work addresses the colorization problem, where the input draft maps lack any initial color and must be transformed to a semantically consistent color map guided by polygon metadata.

4.5 Summary

We target the problem of colorizing monochromatic thematic maps using only information derived from the map itself. We present a metadata-driven learning approach that exploits polygon metadata, including boundary lines and map key semantics, to guide the colorization process. Our approach combines hierarchical superpixel-based spatial constraints with seman-

tic tag embeddings to enable consistent and semantically accurate colorization. Evaluation results demonstrate that our proposed method outperforms comparative approaches in both visual quality and downstream polygon extraction accuracy.

Future work lies in generating synthetic training data to broaden the range of map styles supported and in adapting the framework to handle more diverse thematic map designs or map legend conventions.

Chapter 5

Exploiting Polygon Metadata to Generalize Digitization across Styles

Historical maps are critical for understanding long-term environmental, geographical, and urban systems. Previous research on automated polygonal feature extraction has shown promising results for map series with uniform styles. However, generalizability across diverse printing techniques and color schemas with insufficient labeled data remains limited. This hinders scalable digitization of historical archives. We target the problem of generalizing cross-domain polygon extraction from historical maps. The challenge lies in domain shift across various printing techniques, color schemas, and pattern degradation. To address these unseen styles without target-domain polygon annotations, we propose a legend-guided, test-time adaptive mixture-of-experts framework that fuses solutions from complementary modules. Our approach learns region representations from polygon map key (legend item) cues and consensus-based pseudo-labels via contrastive objectives, adaptively reweighting expert solutions to produce region-consistent polygon masks. It adapts per map at inference time while keeping the underlying expert models fixed. We evaluated five diverse historical map datasets. Our approach statistically significantly outperforms state-of-the-art methods, including pre-trained large vision-language models, by 43.89% in instance-based accuracy, by 5.45% in pixel-based F1 score, and by 2.05% for estimated reductions in post-editing effort.

5.1 Motivation

Historical maps preserve long-term geographic, environmental, and urban information that is often unavailable from modern surveys. Digitizing such raster archives into structured, linked polygon layers enables downstream analyses such as land-cover reconstruction, urban growth studies, infrastructure planning, and others that rely on linked historical geographical information [16, 40, 57]. However, automating polygon extraction from historical maps remains challenging because map collections vary widely in printing technologies (e.g., chromolithography, offset printing, hand-drawn styles), color schemas, and degradation artifacts including fading, ink diffusion, scanning noise, and textual overlays [49, 66]. These variations induce severe domain shift, causing extraction models trained on one map series to fail when deployed on unseen collections.

Previous research only partially addresses this gap. Some learning-based pipelines [81, 84] are designed for a fixed and small set of polygon categories (e.g., water bodies) and do not incorporate polygon map keys (legend items) during inference. Hence, supporting new polygon map keys requires retraining parts of the model. Other legend-oriented approaches [49, 59] can incorporate arbitrary polygon map keys at inference time, but are typically trained on a single map series (e.g., geological maps with consistent cartographic conventions) and have limited ability to generalize to unseen styles with different symbolization, color drift, and boundary rendering. Generic segmentation foundation models [36, 63] can delineate coherent regions, yet they are not map-specialized and usually require nontrivial post-processing to link segments to polygon map keys. In addition, pre-trained large vision-language models [2, 72] can condition on polygon map keys and provide semantically informed predictions, but their pixel-level localization and accuracy are still limited.

We target *cross-domain polygon extraction from historical maps without target-domain polygon annotations*. Following the setting in modern digitization systems, the input includes (i) the raster map, (ii) a region-of-interest (ROI) mask for the map content area, and (iii) a

set of pre-parsed legend keys in JSON format. The output is a series of binary image for polygon map keys in the map, as depicted in Figure 5.1.

To address the challenges, we leverage three complementary expert models that provide class-wise segmentation masks for the legend-defined polygon categories. The expert models include a learning-based dedicated historical-map polygon-extraction model (LOAM [22, 49]) trained on a disjoint map collection, a pre-trained segmentation model (SAM2 [63]) with post-processing entity linking to polygon map keys, and a pre-trained large vision-language model (Gemini 3 Flash [72]). Then, we propose ***GLYPH*** (**G**eneralization via **L**egend-guided **Y**oked **P**olygon extraction in **H**istorical maps), a legend-guided semantic fusion framework that generalizes polygon extraction by integrating these solutions at the instance level (region level). GLYPH constructs region partitions from cross-solution boundary consensus, learns lightweight region representations and fusion weights at test time using polygon map keys and agreement-based pseudo labels. It then outputs region-consistent per-legend masks without fine-tuning the expert models or using any target-domain polygon supervision.

GLYPH learns to adaptively leverage the dedicated model to distinguish polygon map keys with similar colors and patterns, producing geometrically coherent outputs within limited map styles. For the VLM, GLYPH aims for semantic generalization across diverse map appearances despite its coarse, geometrically imprecise predictions. For the segmentation model, GLYPH focuses on sharp polygon boundaries, despite its limited semantic grounding.

We collect five public historical map datasets that span distinct cartographic conventions, printing technologies, and spatio-temporal ranges, and manually annotate their polygon ground truth to support quantitative evaluation of cross-domain generalization. Our approach achieves statistically significant improvement over state-of-the-art methods, including pre-trained large vision-language models (VLMs), by 43.89% in the instance-based evaluation metric, 5.45% in the pixel-based evaluation metric, and 2.05% estimated reduction in post-editing effort on modern map digitization systems. In addition, we present the trade-offs among accuracy, runtime, and monetary cost for each method for deployment.

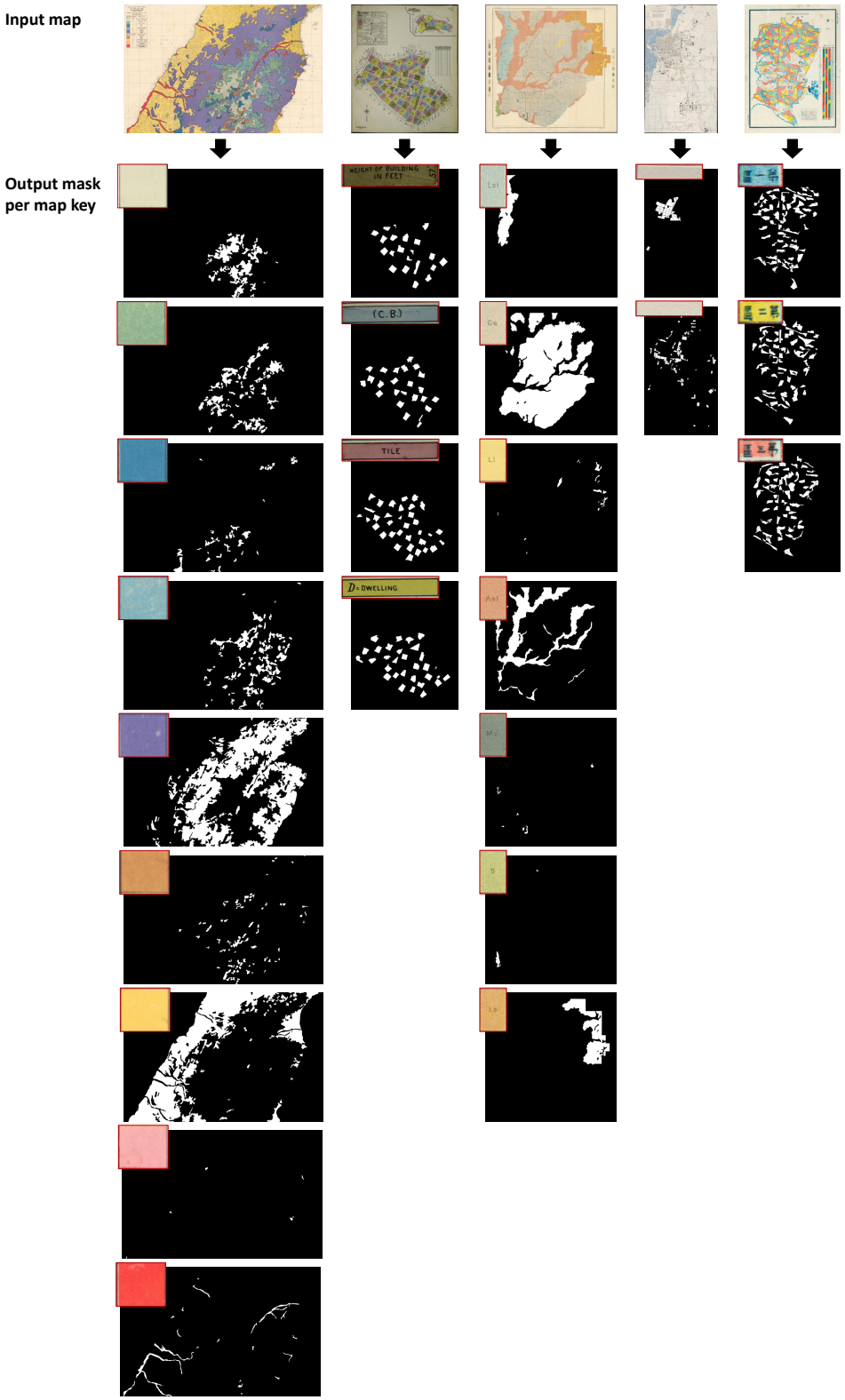


Figure 5.1: The input and output examples, with the JSON-indicated polygon map keys (legend items) attached to each of the output masks. None of the exact same maps or polygon map keys exist in the training dataset of the employed expert model LOAM (Chapter 2).

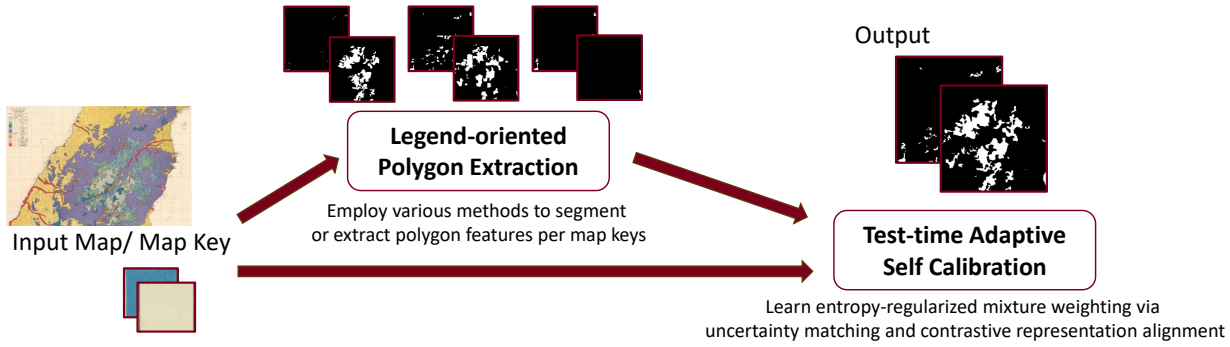


Figure 5.2: The schematic diagram of how we exploit polygon metadata in GLYPH.

5.2 Approach to Generalize Polygon Digitization

5.2.1 Problem Definition

We target the extraction of polygonal features from a raster map image I, along with a binary mask defining the map content area Ω and a set of geometries $\mathcal{L}$ specifying the pixel coordinates of the polygon map keys in the map. The $\mathcal{L}$ only includes polygon map keys that have their corresponding polygon features that exist in Ω.

Our framework leverages three expert-model solutions representing distinct paradigms: a dedicated historical-map polygon-extraction model (LOAM), a pre-trained large vision-language model (VLM) (Gemini), and a segmentation model (SAM2).

Given these inputs, our goal is to produce refined polygon masks without target-domain pixel supervision. The framework must reconcile the semantic and geometric strengths of these distinct experts while adapting to map-specific color drift and printing degradation at inference time.

5.2.2 Approach Overview

We propose GLYPH, a polygon metadata-driven semantic fusion framework to generalize polygon digitization across diverse cartographic styles without requiring additional supervision. GLYPH exploits polygon metadata in two complementary ways, as illustrated in Figure 5.2.

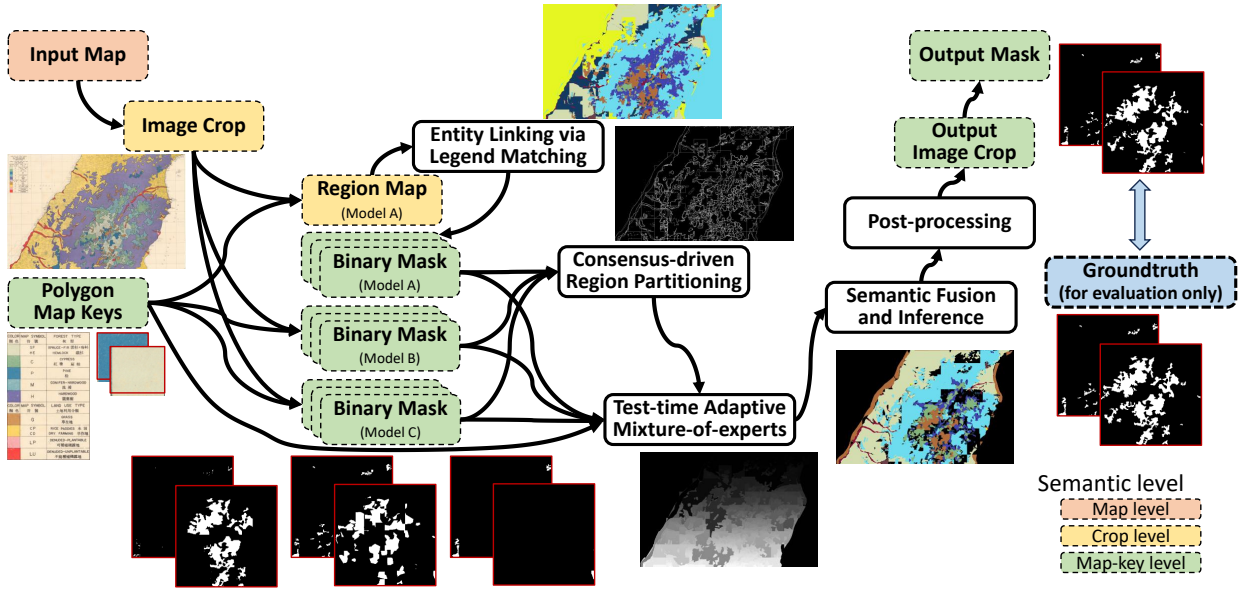


Figure 5.3: The workflow of our approach GLYPH.

We first use polygon map keys to extract preliminary solutions of various expert models. Then, we standardize expert inputs, partition the map into semantic-homogeneous regions, and learn to gate among solutions based on their uncertainty matching and contrastive representation alignment at test time, to produce geometrically regularized outputs. We illustrate the workflow of the proposed GLYPH in Figure 5.3.

5.2.3 Processing of Polygon Metadata

We first process polygon metadata via an optional polygonal entity linking and a consensus-driven region partitioning. We exploit the polygon map keys, along with their corresponding polygon digitization results from expert models, including legend-oriented polygon digitization, legend-based VLM retrieval, and general segmentation results. By reconciling the polygon metadata and its relevant preliminary solutions, we aim to derive minimal polygon instances with distinct confidence levels to support further self-calibration.

5.2.3.1 Entity Linking via Legend Matching

Expert models such as SAM2 [63] may generate class-agnostic segments. To transform these into a class-aware expert, we implement an entity-linking module that maps individual segments to polygon map keys.

We first identify a reference median color for each polygon map key using its localized geometry. For each class-agnostic segment, we compute its median color in the CIELAB color space [19] (as illustrated in Figure 3.5) and assign the segment to the map key with the minimum Euclidean color distance. Segments exceeding a specified distance threshold are discarded as background noise. Finally, we merge all segments assigned to the same label to form class-aware expert masks. This grounding process ensures that segmentation-based experts are semantically aligned with the specific map keys.

5.2.3.2 Consensus-driven Region Partitioning

To reconcile conflicting expert predictions and ensure geometric stability, GLYPH partitions the map into regions through a consensus-driven polygon-boundary extraction process. The intuition is to leverage the expert models to identify high-confidence structural boundaries.

From each expert's segmentation, we derive a binary boundary map and construct a union boundary map by identifying pixels where the density of expert boundaries meets a voting threshold. To eliminate fragmentation, tiny regions below an area threshold are merged into their most similar neighbor based on expert label enclosure and color closeness. We then assign a unique ID to each connected component separated by these reconciled polygon boundaries and propagate the IDs back to the boundary pixels to ensure a dense partition and efficient indexing [51]. We depict an example in Figure 5.4.

This process yields a stable region map in which each region serves as the minimal instance for polygon extraction. We assume these resulting instances to be semantically homogeneous, providing a geometrically consistent basis for subsequent fusion and inference.

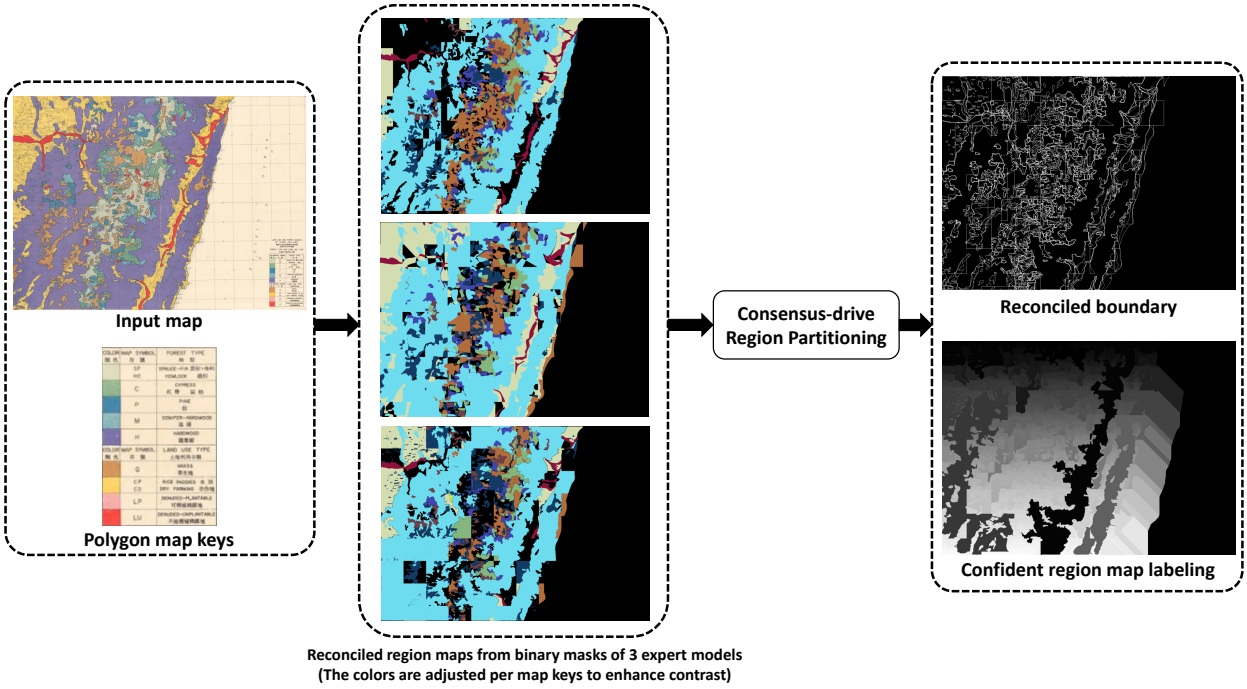


Figure 5.4: An example of consensus-driven region partitioning in GLYPH.

5.2.4 Using Metadata to Learn to Generalize Digitization

With minimal instances of polygons and their corresponding confidence labels, we apply a test-time adaptive mixture-of-experts approach to self-calibrate the polygon features. We illustrate a schematic diagram in Figure 5.5. GLYPH treats the map keys as anchors to identify reliable positive samples of polygon instances, uses the discrepancies among three preliminary solutions to learn the representations of map keys under contrastive objectives, and adaptively updates the weights among them with respect to the map and map keys in it to determine the final output for each polygon instance. We provide the details in Appendix B.1.

5.2.4.1 Test-time Adaptive Mixture-of-experts

GLYPH handles domain shift by adapting to the specific cartographic style and color schema of each map at inference time. The intuition is to leverage the map legend and expert consensus as localized weak supervision to calibrate the extraction process. We represent

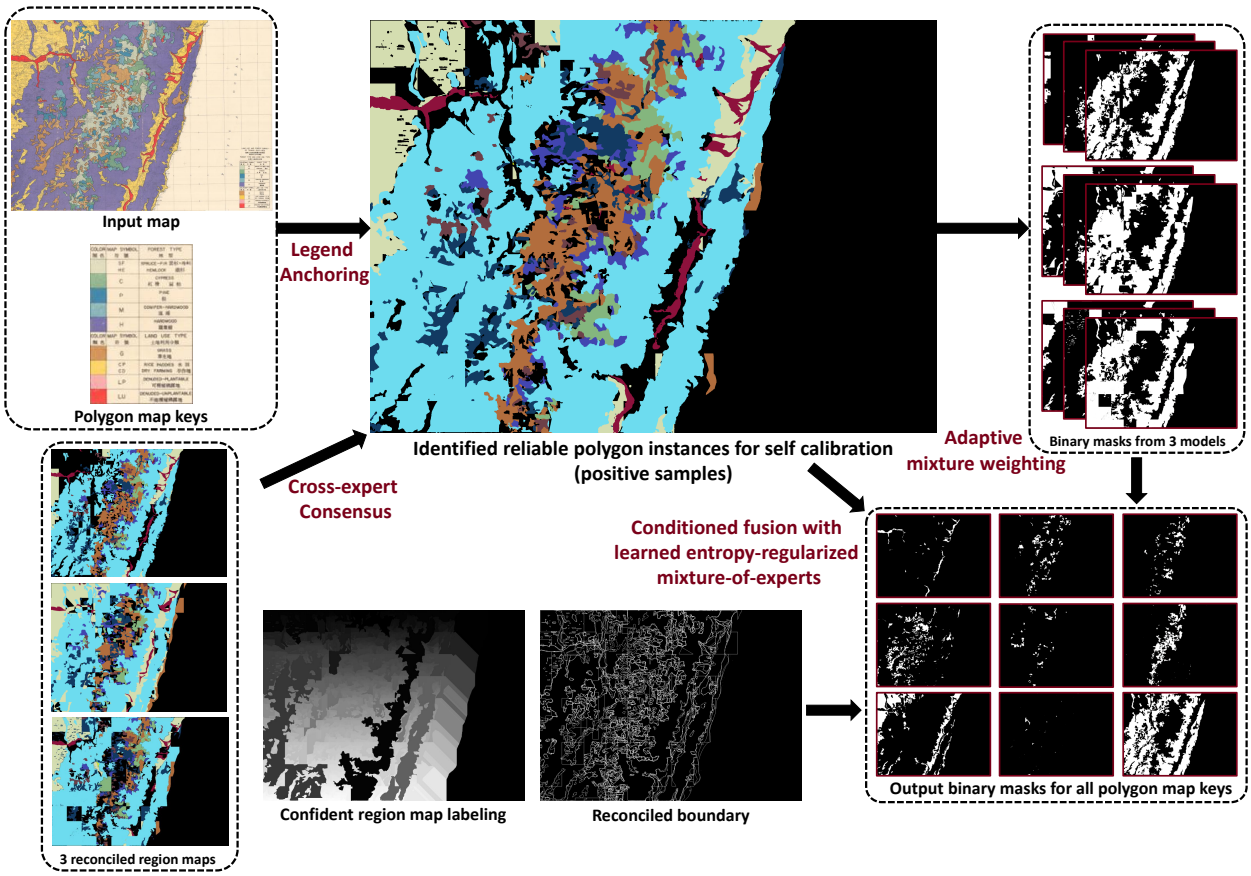


Figure 5.5: A schematic diagram of the test-time adaptive mixture-of-experts with semantic fusion and inference in GLYPH.

each reconciled region using its perceptual color statistics and a global context vector that captures the map's overall visual complexity and expert reliability.

The adaptation workflow employs a lightweight representation learning strategy. We optimize a color embedding network f_θ and an expert gating module g_ϕ using a contrastive objective. This process aligns regional features with learned legend prototypes by maximizing the similarity between embeddings sharing the same semantic label while separating embeddings with dissimilar labels. It utilizes training signals from two complementary sources: *Legend Anchors* and *Cross-Expert Consensus*. While polygon map keys serve as the canonical semantic targets, they often deviate from their corresponding polygon features in the map content area due to fading, overlapping context, or printing artifacts. To bridge this semantic gap, we identify regions where at least two experts reach a majority consensus and utilize these

as agreement-based pseudo-labels. This dual-source logic allows the model to learn a mapping that reconciles the idealized legend-item metadata with the degraded reality of polygon features in the map content area. By utilizing consensus-driven regions as positive samples, the embedding space is regularized to recognize the "on-site" polygon-feature appearance of map keys. Simultaneously, the gating module adapts the mixture sharpness of the expert ensemble based on global style statistics and inter-expert reliability signals.

Unlike classical mixture-of-experts models trained with explicit gating supervision, the gating mechanism here operates at test time via self-regulation based on ensemble-uncertainty signals. The gating module does not receive explicit supervision on per-expert correctness; instead, it takes (i) global color-style features extracted from the map image and (ii) summary statistics of expert agreement and majority confidence. The gating objective regularizes the entropy of the mixture weights to match the observed level of inter-expert disagreement. Thus, the module controls how decisive or conservative the ensemble should be under varying style and uncertainty conditions, rather than directly optimizing expert selection accuracy.

This allows GLYPH to formulate the original zero-shot extraction task as a self-calibrating optimization problem. Rather than relying on static global priors, the per-map adaptation loop bootstraps a map-specific classifier. By iteratively refining the latent representation against localized weak signals, GLYPH compensates for expert hallucinations and domain shift using only the internal evidence provided by the map itself. This ensures that the final fusion is not a simple or heuristic averaging of experts, but an automatically and adaptively modulated ensemble and semantically grounded selection tailored to specific artifacts and uncertainty characteristics of the target archive.

5.2.4.2 Semantic Fusion and Inference

The semantic assignment for each region is performed by integrating gated expert evidence with learned visual similarity. The intuition is to reconcile the high-level semantic knowledge of expert models with the low-level chromatic grounding provided by the polygon map keys.

We utilize the learned gating module to determine the relative weights of the experts based on the map's global style and agreement statistics, producing a weighted consensus for each reconciled region. The gating weights are adaptively determined by the gating module and conditioned on style and agreement features, rather than supervised expert ranking.

This consensus is then combined with a similarity score between the region's embedding and the learned legend prototypes. The final label is determined by the joint maximization of the expert vote and the legend similarity. To ensure robustness, we incorporate a background rejection mechanism that filters out regions with low confidence or weak similarity to the legend. This fusion ensures that the final labels are both regionally consistent and semantically grounded in the localized metadata.

By weighting experts according to the global context, favoring certain models for sharp hand-drawn maps and others for complex geological prints, GLYPH ensures the final polygon extraction is tailored to the specific artifacts across various archives. To support the real-world map digitization process, GLYPH tends to be conservative and prioritizes precision by rejecting ambiguous regions that do not strongly align with the expert consensus or the learned legend prototypes.

5.2.4.3 Structural and Geometric Post-processing

To produce vectorized outputs that conform to the modern map digitization systems, GLYPH performs a multi-stage structural refinement. We apply a series of morphological filters to eliminate isolated noise and small fragments potentially due to scanning or printing artifacts. The refined polygon entities can then be treated as the final output to support efficient indexing on the map digitization systems and to support downstream analysis.

5.3 Dataset

We evaluate on five historical map datasets that span distinct geographic regions, temporal ranges, cartographic conventions, and printing technologies. The selected five datasets are intentionally diverse in both visual appearance and structural complexity, enabling a rigorous assessment of cross-domain generalization. We show representative examples in Figure 5.6.

The dedicated learning-based approach, LOAM (Chapter 2), was trained on a separate geological map collection from the USGS [25], referred to as GE in the following subsections. This GE dataset is excluded from our main evaluation for GLYPH and has no overlap with the five datasets at the map or polygon map key levels. Moreover, the printing techniques, cartographic agencies, and publishers of GE differ from those of the five selected datasets. This allows us to better examine the out-of-domain performance across all methods.

5.3.1 Dataset Overview

We summarize the five datasets used for our main evaluation for this chapter in Table 5.1.

Forest Type Maps (FT). The FT dataset consists of thematic forest maps released by Academia Sinica between 1954 and 1955. These maps contain irregular polygon features around mountainous areas, with thick polygon boundaries to indicate gradual changes and separations among forest types. In addition to the overlaid topographic lines and text, the color within each polygon in the map content area is uneven, with creases and fading. This dataset can support long-term environmental monitoring and landscape change research.

Sanborn Maps (SA). The SA dataset is published by Sanborn Map Company and available from the Library of Congress. These maps contain information about properties and individual buildings spanning 1920 to 1960, supporting fire insurance assessments or studies on historical architecture and settlement. The polygon features tend to be regular in shape, with overflows to roads. They suffer from dense overlapping text and color mismatches between polygon features in the map content area and their corresponding map keys.

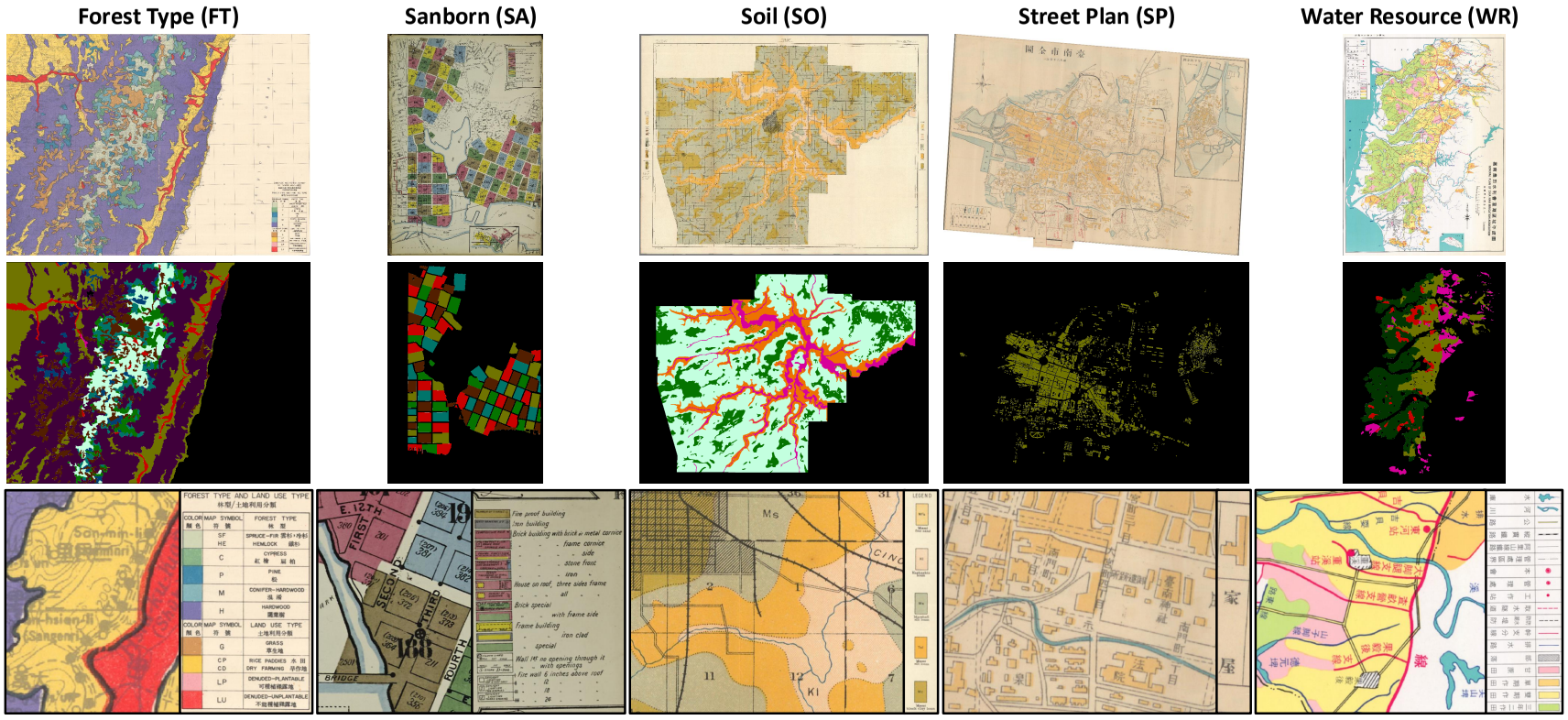


Figure 5.6: Representative examples for each dataset. Each column shows a map (top), corresponding polygon ground truth annotation with enhanced color differences (middle), and an enlarged image snippet of the map with the map legend (bottom).

Table 5.1: Summary of historical map datasets used for evaluation. Raster resolution is reported in pixel space with mean ± standard deviation. HMC stands for "Historical Map Collection". Links to the resources are provided if available.

	Forest Type (FT)	Sanborn (SA)	Soil (SO)	Street Plan (SP)	Water Resource (WR)
Data Source	Academia Sinica	Library of Congress	David Rumsey HMC	Multiple	Academia Sinica
Year	1954–1955	1920–1960	1903–1908	1911–1950	1921–1995
# Maps	6	6	6	4	4
# Map Keys (avg ± std)	38 (6.33 ± 3.01)	27 (4.50 ± 0.55)	37 (6.17 ± 1.33)	7 (1.75 ± 0.50)	16 (4.00 ± 0.82)
# Polygons (avg ± std)	1,031 (171.83 ± 182.81)	402 (67.00 ± 30.30)	569 (94.83 ± 105.90)	2,091 (522.75 ± 884.32)	690 (172.50 ± 60.83)
Image Width (px)	12,729.50 ± 1,438.32	6,139.83 ± 553.71	10,033.83 ± 2,121.56	5,034.50 ± 1,436.79	9,069.00 ± 2,597.17
Image Height (px)	9,005.17 ± 990.25	7,983.00 ± 184.77	8,914.17 ± 1,487.39	4,843.50 ± 2,361.14	10,628.50 ± 165.46
Total Pixels	$(1.16 \pm 0.26) \times 10^8$	$(4.89 \pm 0.37) \times 10^7$	$(9.16 \pm 3.31) \times 10^7$	$(2.50 \pm 1.40) \times 10^7$	$(9.65 \pm 2.85) \times 10^7$
Printing Technique	Chromolithography	Lithography	Chromolithography	Hand-drawn, CMYK	Offset printing
Language	Chinese, English	English	English	Japanese, English	Chinese, Japanese
Geographic Coverage	Taiwan	United States	United States	Japan, Taiwan	Taiwan
Data Availability	Public[a]	Public[b]	Public[c]	Partially Public[d,e]	Public[e]

[a] https://gis.sinica.edu.tw/showwmts/index.php?s=tileserver&l=1956_Landuse_250K_1
[b] https://www.loc.gov/collections/sanborn-maps/about-this-collection/
[c] https://www.davidrumsey.com/luna/servlet/view/search/who/U.S.+Department+of+Agriculture.+Field+Operations+of+the+Bureau+of+Soils.
[d] https://www.mediapal.co.jp/book/3374/
[e] https://gis.sinica.edu.tw/tileserver/

Soil Maps (SO). The SO dataset is published by the United States Department of Agriculture (USDA) and available from the David Rumsey Historical Map Collection. These maps record soil types and range from 1903 to 1908. Similar to USGS geological maps, adjacent soil classes often differ subtly in color and can only be distinguished by their associated text or patterns. In addition, we notice ink overflows across dashed polygon boundaries separating adjacent irregular polygon features of distinct map keys. Rather than supporting fine-grained analysis, this dataset is helpful for large-scale agriculture modeling, infrastructure siting, and hydrologic modeling.

Street Plan Maps (SP). The SP dataset aggregates historical urban maps from Academia Sinica and a historical map collection in Tokyo. These maps record the purpose of properties and the density of individual buildings, with only one or two polygon map keys per map and polygon features in regular shapes. Most of the maps were originally produced from 1911 to 1950 and later printed on paper. We then scan the paper maps at 1200 dpi, resulting in noticeable CMYK halftone dot patterns on the raster images. In addition, these maps suffer from significant ink overflows and severe fading. This dataset can support studies on historical settlement and urban planning across time and locations.

Water Resource Maps (WR). The WR dataset is published by domestic irrigation associations and available from Academia Sinica. These maps document irrigation plans for the farmland of two regions spanning 1921 to 1995. While the polygon features tend to be in regular shapes, there is a significant amount of overlaid or adjacent transportation networks, water channels, and text. In addition, the color within each polygon feature is uneven, and there are noticeable ink overflows across adjacent polygon features with no explicit solid or dashed polygon boundary lines. This dataset can support studies on historical settlement as well as agricultural planning across time and locations.

5.3.2 Dataset Annotation

To support the evaluation of polygon extraction and quantification when comparing the post-launch performance, we manually annotate the polygon ground truth from scratch for the five datasets and record the corresponding annotation time.

5.3.2.1 Polygon Ground Truth Annotation

The polygon ground truth is manually annotated by three independent voluntary annotators, including one Ph.D. candidate (myself) and two retired professors. One annotator has received formal training in GIS-related subjects at the university level. All three annotators are familiar with English, Chinese, and Japanese, enabling accurate interpretation of multilingual legends and textual elements on historical maps.

A unified guideline is agreed upon prior to annotation, covering polygon inclusion criteria, boundary interpretation under degradation, nested or overlapping regions, and ambiguous areas. Annotations are conducted independently following these shared instructions. Each annotator uses GIMP 3.0.4 with a drawing pad to trace the centerline of polygon boundaries with a 1-pixel width for each map and to fill each region with a color (ID) that links to a polygon map key the annotator believes to be corresponding to.

The polygon ground truth follows the same setup as our targeted polygon extraction task: we only target polygon map keys that have corresponding polygon features that exist in the map content area, and we only annotate polygon features in the map content area that can correspond to a polygon map key.

5.3.2.2 Inter-Annotator Agreement

To assess annotation reliability, at least one map from each dataset is annotated by all three annotators. We present the inter-annotator agreement in Fleiss' κ in Table 5.2 along with annotation statistics, including annotation time and the number of annotated vertices.

The inter-annotator agreement is above 0.96 across all five datasets, indicating near-perfect agreement among the three annotators. This demonstrates the clarity of the annotation protocol and the consistency of polygon interpretation despite substantial stylistic and linguistic variation. For maps with multiple annotators, the final ground truth polygons are obtained by consolidating annotations via majority agreement, with remaining ambiguities resolved through joint review.

5.4 Evaluation

We evaluate out-of-domain performance on the five datasets introduced in Section 5.3. This allows us to better assess the cross-domain generalizability in the polygon digitization task across methods. Still, we include a subsection for in-domain evaluation using the GE dataset.

5.4.1 Evaluation Metric

We evaluate extraction quality from three quantitative aspects: (1) whether polygon instances are correctly identified and grouped, (2) whether polygon boundaries align with ground truth at the pixel level, and (3) how much manual geometric correction would be required for post-extraction? Accordingly, we report three primary metrics, each emphasizing a distinct evaluation perspective: **MMPQ** for instance-level structural correctness, **F1@8** for pixel-level polygon boundary alignment, and **NBDR** for estimated post-editing effort.

Let I denote a map image with N pixels. Let $\mathcal{G} = \{G_i\}$ and $\mathcal{P} = \{P_j\}$ denote the sets of ground truth and extracted polygon instances, respectively, where each instance corresponds to a binary pixel mask. For a pixel set $S \subseteq I$, $|S|$ denotes its cardinality. For two pixel sets A and B, the Intersection over Union (IoU) is defined as

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Many-to-Many Panoptic Quality (MMPQ)

Table 5.2: Statistics for the annotation of historical map datasets used for evaluation. Annotation time is presented in minutes.

	Forest Type (FT)	Sanborn (SA)	Soil (SO)	Street Plan (SP)	Water Resource (WR)
# Maps / Map Keys	6 / 38	6 / 27	6 / 37	4 / 7	4 / 16
# Polygons (avg ± std)	1,031 (171.83 ± 182.81)	402 (67.00 ± 30.30)	569 (94.83 ± 105.90)	2,091 (522.75 ± 884.32)	690 (172.50 ± 60.83)
# Annotated Vertices	81,924	5,917	42,603	101,853	41,476
(avg ± std)	(13,654.00 ± 15,124.31)	(986.17 ± 743.48)	(7,100.50 ± 8,667.66)	(25,463.25 ± 39,480.41)	(10,369.00 ± 4,511.22)
Annotation Time (min.)	1,925	840	1,660	1,405	2,655
(avg ± std)	(320.83 ± 372.97)	(140.00 ± 124.10)	(276.67 ± 251.61)	(351.25 ± 338.83)	(663.75 ± 46.44)
Fleiss' κ	0.9758	0.9874	0.9950	0.9684	0.9828

Since most downstream tasks of polygon extraction from historical maps rely on their vectorized polygon instances, MMPQ evaluates the polygon extraction quality at the *instance level*. It assesses whether the extracted polygons form structurally correct instances with appropriate spatial extent.

We construct a bipartite graph between ground truth polygon instances $\mathcal{G}$ and extracted polygon instances $\mathcal{P}$. We create an edge in the graph between G_i and P_j if

$$\mathrm{IoU}(G_i, P_j) \geq \tau,$$

where τ is an overlap threshold and set to 0.1 for our evaluation.

Each connected component c in this graph defines a many-to-many matched group with union masks

$$G_c = \bigcup_{G_i \in c} G_i, \qquad P_c = \bigcup_{P_j \in c} P_j.$$

For each matched component, we compute a group-level IoU

$$\mathrm{IoU}_c = \frac{|G_c \cap P_c|}{|G_c \cup P_c|}.$$

Following the Panoptic Quality (PQ) similar to [54], we define the *Segmentation Quality (SQ_w)* as the area-weighted average IoU over matched components and the *Recognition Quality (RQ_w)* as an area-weighted component-level F-score,

$$\mathrm{SQ}_w = \frac{\sum_{c \in TP} w_c \cdot \mathrm{IoU}_c}{\sum_{c \in TP} w_c}, \quad w_c = |G_c \cup P_c|;$$

$$\mathrm{RQ}_w = \frac{\sum_{c \in TP} w_c}{\sum_{c \in TP} w_c + \frac{1}{2}\sum_{c \in FP} w_c + \frac{1}{2}\sum_{c \in FN} w_c}.$$

Then we can derive MMPQ accordingly

$$\text{MMPQ} = \text{SQ}_w \cdot \text{RQ}_w.$$

MMPQ jointly captures region coverage accuracy and instance recognition quality, while weighting errors by the affected area. It ranges from 0 to 1, and the higher the better.

Pixel F1 with Tolerant Radius (F1@8)

As an evaluation metric suggested in [25], the F1 score evaluates extraction quality at the *pixel level*, focusing on polygon boundary alignment. We relax the definition of true positives to account for minor spatial inconsistency caused by thick polygon boundaries that can span up to 20 pixels in the historical maps.

A predicted pixel p is considered correct if its Euclidean distance to the nearest ground truth pixel is within a tolerance radius r:

$$\min_{q \in G_i} \|p - q\|_2 \leq r.$$

Let TP_r, FP_r, and FN_r denote the number of true positives, false positives, and false negatives under tolerance r. Pixel-level precision, recall, and F1 score are then defined as

$$\text{Precision}_r = \frac{\text{TP}_r}{\text{TP}_r + \text{FP}_r},$$

$$\text{Recall}_r = \frac{\text{TP}_r}{\text{TP}_r + \text{FN}_r},$$

$$\text{F1}_r = \frac{2 \cdot \text{Precision}_r \cdot \text{Recall}_r}{\text{Precision}_r + \text{Recall}_r}.$$

We report F1_8 with $r = 8$ pixels, named F1@8, to emphasize local geometric alignment with tolerance to small boundary shifts. It ranges from 0 to 1, and the higher the better.

Normalized Boundary Displacement Ratio (NBDR)

NBDR estimates the *post-editing effort* required for humans to fix the extracted polygons. It measures the average symmetric polygon boundary displacement normalized by the size:

$$\text{NBDR}_i = \frac{\text{ASSD}_i}{\sqrt{|G_i|}},$$

where ASSD_i is the Average Symmetric Surface Distance between the polygon boundaries of G_i and P_i. Let ∂G_i and ∂P_i denote the boundary pixel sets of G_i and P_i, respectively,

$$\text{ASSD}_i = \frac{1}{|\partial G_i| + |\partial P_i|} \left(\sum_{x \in \partial G_i} \min_{y \in \partial P_i} \|x - y\|_2 + \sum_{y \in \partial P_i} \min_{x \in \partial G_i} \|y - x\|_2 \right).$$

Here, polygon boundaries are extracted as 1-pixel-wide contours from binary masks. A lower NBDR indicates less estimated post-editing effort per unit of polygon scale.

Reminder of the Evaluation Metrics

The three metrics capture complementary aspects of polygon extraction quality and may exhibit divergent trends. NBDR is used to estimate post-editing effort, but the actual effort is highly dependent on the user interface (UI). One must still consider MMPQ and F1@8 when assessing results, especially in terms of universal accuracy. For instance, a method may achieve high F1@8 by closely following boundary pixels but split into smaller polygon instances, resulting in lower MMPQ. In contrast, a method may preserve the structure of the polygon instances (high MMPQ) but yield misaligned boundaries, resulting in lower F1@8. Similarly, a method with high MMPQ and F1@8 may still achieve high (worse) NBDR if the polygon geometry is overly complicated, even though the polygon boundaries are close to the ground truth and their errors are small. In this case, the NBDR-identified potentially increased post-editing cost may not be accurate.

5.4.2 Evaluation Setting

We implement all methods in Python on a Gigabyte workstation equipped with an Intel Xeon w9-3595X CPU at 2.00 GHz, 512 GB RAM at 4800 MT/s, and two NVIDIA A6000 GPUs.

All evaluations are conducted under a strict zero-shot setting, with no pixel-level supervision, polygon ground truth annotations, or dataset-specific calibration provided to the models. The legend JSON and ROI are used solely to localize semantic cues and restrict extraction to the map content area, reflecting realistic historical map digitization scenarios.

We compute evaluation metrics per map and report the mean and standard deviation across maps for each dataset.

5.4.3 Comparative Method

We compare GLYPH against individual experts: LOAM (Chapter 2), SAM2 [63] with legend-based entity linking, and Gemini 3 Flash. In addition, we include the image segmentation model with image exemplars as inputs (SAM3) [7], as well as other proprietary pre-trained large visual-language models, including Gemini [72] (Gemini 3.1 Pro and Gemini 2.5 Pro), GPT [2] (GPT 4o and GPT 5.2 Pro), and Claude (Claude Sonnet 4.5 and Claude Opus 4.6).

To assess whether tiling affects extraction quality for methods that operate on cropped inputs, we vary the tile size for comparative methods and report performance accordingly, including 4096×4096, 2048×2048, 1024×1024, 512×512, and 256×256, with 128×128 if applicable. This analysis is only applied to comparative methods. GLYPH is not tuned by tile size and always fuses expert masks generated under the fixed 1024×1024 tiling setup.

Prompt to VLMs

We provide the prompt to VLMs for polygon extraction in Figure 5.7. For each prompt, we attach the cropped map tile and a key-value set containing the name and image of each polygon map key in the map legend, which is similar to the structure shown in Figure 5.1. VLMs are expected to return the polygon feature geometries of all indicated map keys.

Prompt to SAM3

We use SAM3 [7] with an instance segmentation setup using positive and negative image exemplars as parts of its inputs. We show an example of the prompt to SAM3 in Figure 5.8. Similarly, we crop the original map into smaller tiles. Then, we integrate all polygon map

```
**GOAL**
Identify fine-grained polygonal regions in this specific MAP TILE
that match colors and textures for each of the MULTIPLE provided
legend item crops.

**Context**
- This image is a TILE (sub-section) of a larger map.
- Tile Dimensions: {tile_w}px width x {tile_h}px height.
- **Pixel Coordinates**: Return pixel coordinates normalized to a
   {COORD_SCALE}x{COORD_SCALE} scale
relative to the TOP-LEFT of this tile.

**Input Description**
1. Map Tile image. Irrelevant/Out-of-bound areas are black.
2. A list of Legend item crops, each with a unique Name.

**Instructions**
1. For EACH legend item provided:
    a. Understand its major colors, textures, markings, and text
       patterns.
    b. Identify fine-grained regions (pixels) in the map tile that
       match or are similar.
    c. Build fine-grained multi-polygon boundaries for the
       identified regions (can have holes if needed).
2. Organize the results into a JSON object where keys are the
   Legend Names.

**Output JSON Format**
JSON Format: {{
    'LegendName_A': [ [[x1, y1], [x2, y2], ...], ... ],
    'LegendName_B': [ ... ]
}}

**Constraints**
- Do not hallucinate points.
- Ignore masked-out (black) areas.
- If the feature is cut off by the tile edge, trace the edge to
   enclose the multi-polygon boundaries.

**Input Data**
- Map Tile: Provided below.
- Legend Items: Provided below.
```

Figure 5.7: Prompt used for large visual-language model polygon-feature extraction.

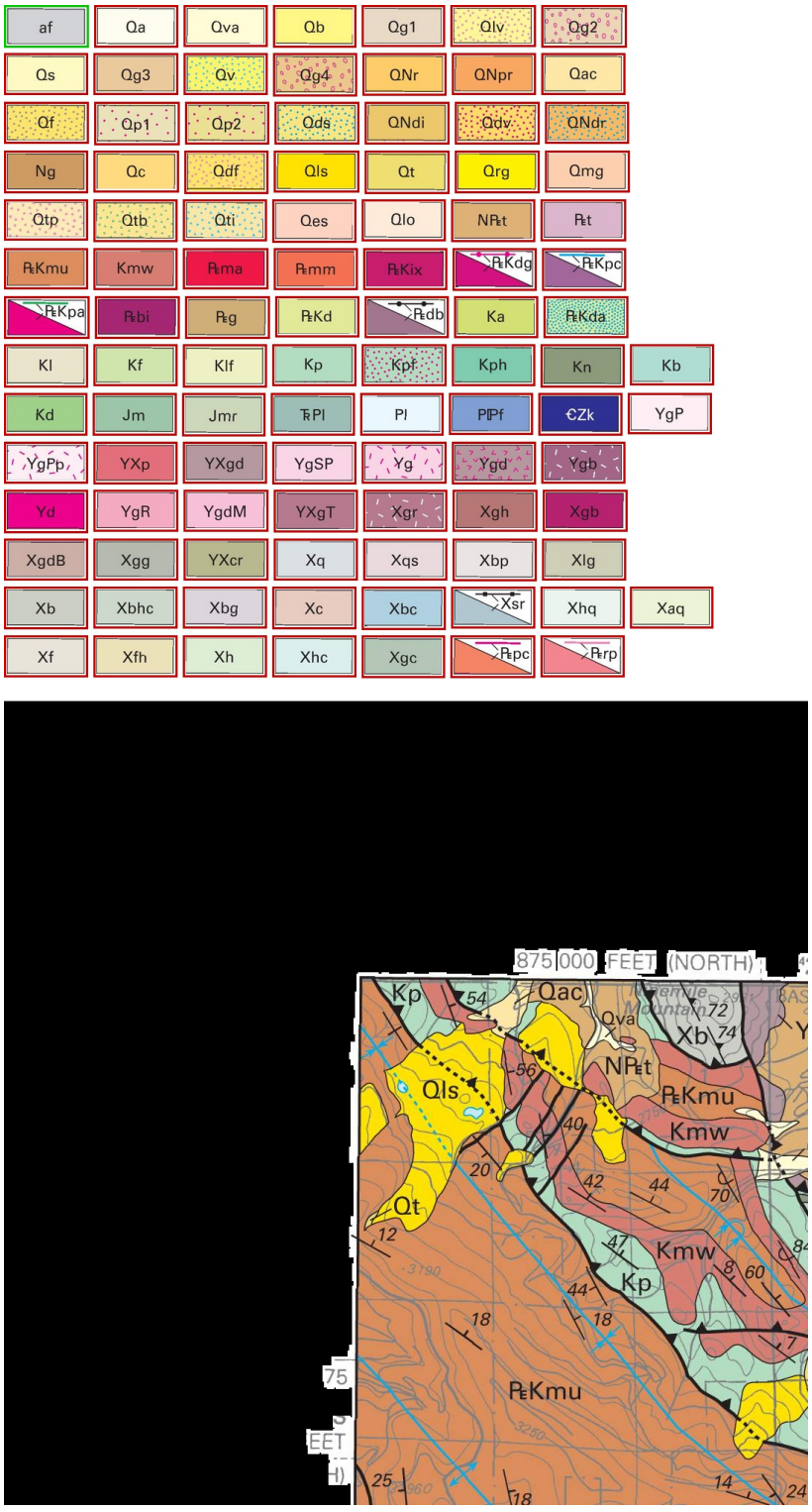


Figure 5.8: Prompt used for SAM3. For visualization purposes, the positive image exemplar is highlighted with a green bounding box, and negative image exemplars are highlighted with red bounding boxes.

keys in the map into a legend patch, as shown in the upper portion of Figure 5.8, and append them to each map tile, as shown in the lower portion. The prompt includes the appended image with the bounding box coordinates for the positive and negative image exemplars. For each map tile, we iterate through all polygon map keys, treating each as the positive image exemplar once to derive the corresponding polygon feature geometries. Finally, for each map key, we merge the polygon features back.

5.4.4 Evaluation Result

Our evaluation results aim to answer the following questions:

- **EQ 1**: Does GLYPH outperform comparators across evaluation metrics and datasets?
- **EQ 2**: How does tiling size affect the performance of comparative methods?
- **EQ 3**: Is there a statistical significance between GLYPH and comparative methods?
- **EQ 4**: How is the trade-off of accuracy, runtime, and monetary cost in deployment?
- **EQ 5**: What is the potential improvement to post-editing effort by GLYPH?
- **EQ 6**: How does each component in GLYPH contribute to the final outputs?
- **EQ 7**: How sensitive is parameter setting to the final outputs of GLYPH?
- **EQ 8**: How is the trade-off of integrating the expert models at various tile sizes?
- **EQ 9**: What are the qualitative results of GLYPH against comparative methods?
- **EQ 10**: What is the quantitative performance of GLYPH on in-domain dataset?

5.4.4.1 Overall Performance

For **EQ 1**, we present the main evaluation results of GLYPH against comparative methods in Table 5.3, covering dataset-level performance under three complementary metrics: instance-level structural correctness (MMPQ), pixel-level boundary alignment (F1@8), and estimated post-editing effort (NBDR).

Table 5.3: Summary of evaluation performance across datasets. For each evaluation metric, the best performance within a method family is in bold, and the best performance overall is in red. Values are presented in mean±std unless otherwise noted. "N.A." indicates NBDR is not applicable when a method returns empty results across all cases.

Dataset / Metric	FT			SA			SO			SP			WR		
Method - Tile Size	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓
GLYPH-1024 (Ours)	**0.90**	**0.90±0.16**	**0.48±1.27**	**0.69**	**0.87±0.16**	**0.12±0.16**	**0.81**	**0.91±0.18**	**0.36±1.12**	**0.30**	**0.59±0.21**	**0.79±0.82**	**0.65**	0.82±0.13	0.15±0.16
LOAM-1024	0.71	0.83±0.25	0.51±2.26	0.38	0.57±0.28	0.39±0.55	0.76	0.88±0.26	0.38±1.01	0.13	0.32±0.27	1.33±1.53	0.33	**0.89±0.08**	**0.08±0.06**
SAM2-4096	0.33	0.53±0.32	2.97±9.89	0.47	0.61±0.35	0.45±0.50	0.41	0.64±0.29	0.72±1.57	0.06	0.39±0.29	1.20±1.12	0.18	0.49±0.27	0.32±0.34
SAM2-2048	**0.56**	**0.65±0.28**	1.20±2.54	0.42	0.60±0.33	0.29±0.26	0.48	0.73±0.24	0.67±1.44	0.10	0.45±0.28	1.19±1.35	0.27	0.56±0.24	0.26±0.21
SAM2-1024	0.54	0.59±0.29	1.29±2.74	0.46	0.61±0.32	**0.26±0.22**	0.50	0.70±0.25	0.69±1.46	0.13	0.50±0.28	**1.06±1.36**	0.08	0.53±0.23	0.22±0.20
SAM2-0512	0.54	0.60±0.26	1.08±1.99	0.42	0.60±0.31	0.31±0.33	0.49	0.71±0.25	0.70±1.53	**0.16**	0.51±0.27	1.15±1.33	0.23	0.53±0.22	0.20±0.21
SAM2-0256	0.53	0.60±0.25	**0.81±1.41**	**0.47**	**0.63±0.32**	0.27±0.24	**0.57**	**0.76±0.28**	**0.59±1.22**	0.14	**0.55±0.26**	1.10±1.25	**0.34**	**0.63±0.25**	**0.20±0.22**
SAM3-4096	0.00	0.00±0.00	15.98±16.69	0.03	0.06±0.11	11.91±7.33	0.00	0.00±0.00	11.63±13.02	**0.09**	0.18±0.24	3.93±4.18	0.00	0.00±0.01	10.10±6.47
SAM3-2048	0.02	0.01±0.05	20.14±20.47	**0.09**	**0.14±0.13**	6.62±7.63	0.00	0.00±0.00	10.25±9.49	0.03	**0.30±0.31**	2.18±1.56	0.00	0.00±0.01	8.16±6.70
SAM3-1024	0.02	0.02±0.06	22.22±25.10	0.06	0.10±0.13	6.35±7.82	0.00	0.01±0.02	7.41±9.57	0.06	0.25±0.16	1.21±0.85	0.03	0.03±0.08	4.60±4.92
SAM3-0512	**0.09**	**0.07±0.14**	**8.99±11.47**	0.08	0.10±0.14	8.07±8.59	**0.05**	**0.08±0.13**	**2.15±2.95**	0.04	0.19±0.14	1.20±0.94	**0.04**	**0.04±0.10**	**4.42±5.56**
SAM3-0256	0.00	0.00±0.00	15.84±20.42	0.04	0.09±0.12	**6.23±8.24**	0.01	0.03±0.06	2.27±3.07	0.05	0.15±0.12	**1.16±0.88**	0.00	0.01±0.02	5.16±6.17
Gemini-3-flash-4096	0.37	0.40±0.22	0.91±1.63	0.30	0.50±0.16	0.42±0.36	0.28	0.41±0.18	**0.53±0.97**	0.05	0.24±0.13	1.12±0.93	0.05	0.19±0.11	0.81±0.50
Gemini-3-flash-2048	0.46	0.49±0.21	0.89±1.96	0.29	0.49±0.15	0.41±0.30	0.37	0.46±0.24	0.76±1.42	0.07	0.32±0.19	1.01±0.91	0.13	0.29±0.14	0.59±0.52
Gemini-3-flash-1024	0.64	0.68±0.20	0.67±1.55	**0.47**	**0.73±0.16**	**0.27±0.22**	0.50	0.59±0.23	0.63±1.12	0.08	**0.41±0.25**	0.97±1.00	0.23	0.47±0.20	0.51±0.43
Gemini-3-flash-0512	0.69	0.71±0.21	**0.41±1.01**	0.37	0.63±0.18	0.30±0.22	0.54	0.62±0.24	0.64±1.15	0.09	0.37±0.17	1.09±1.11	**0.29**	**0.59±0.21**	**0.44±0.41**
Gemini-3-flash-0256	**0.74**	**0.79±0.26**	0.84±1.86	0.41	0.65±0.23	0.34±0.24	**0.54**	**0.67±0.26**	0.79±1.41	**0.10**	0.36±0.22	**0.90±0.89**	0.29	0.58±0.27	0.51±0.43
Gemini-3.1-pro-4096	0.09	0.07±0.12	3.19±4.66	0.09	0.16±0.18	1.37±0.71	0.13	0.16±0.19	1.46±2.13	0.01	0.07±0.09	4.16±3.64	0.01	0.06±0.07	1.21±0.64
Gemini-3.1-pro-2048	0.13	0.22±0.17	1.05±1.69	0.18	0.37±0.14	0.43±0.15	0.20	0.29±0.19	0.74±0.97	0.00	0.05±0.04	2.38±1.92	0.06	0.16±0.10	0.73±0.51
Gemini-3.1-pro-1024	0.49	0.57±0.21	0.83±2.07	**0.52**	**0.75±0.15**	**0.19±0.17**	0.41	0.58±0.21	**0.61±1.29**	0.11	0.39±0.26	1.12±1.25	0.22	0.45±0.17	0.39±0.42
Gemini-3.1-pro-0512	0.59	0.64±0.19	**0.46±1.12**	0.39	0.59±0.17	0.26±0.18	0.43	0.58±0.21	0.62±1.01	**0.11**	0.39±0.20	0.95±1.02	0.31	0.59±0.22	**0.33±0.27**
Gemini-3.1-pro-0256	**0.65**	**0.70±0.24**	0.78±1.99	0.32	0.52±0.22	0.29±0.20	**0.45**	**0.61±0.25**	0.68±1.11	0.10	**0.40±0.18**	**0.92±1.01**	**0.35**	**0.65±0.24**	0.39±0.32
Gemini-2.5-pro-4096	0.21	0.14±0.19	4.25±10.91	0.00	0.08±0.06	0.91±0.48	0.07	0.12±0.12	1.42±2.00	0.02	0.07±0.12	2.31±1.61	0.04	0.11±0.09	1.14±0.81
Gemini-2.5-pro-2048	0.26	0.28±0.20	0.97±2.15	0.01	0.12±0.07	0.72±0.37	0.23	0.28±0.18	0.77±1.22	0.05	0.17±0.12	1.31±1.26	0.10	0.23±0.10	0.62±0.46
Gemini-2.5-pro-1024	0.34	0.40±0.19	1.00±3.22	0.23	0.46±0.19	0.38±0.31	0.34	0.45±0.21	0.64±1.00	0.07	0.26±0.15	1.00±0.92	0.16	0.35±0.16	0.48±0.35
Gemini-2.5-pro-0512	0.51	0.57±0.20	**0.43±0.90**	0.26	0.55±0.23	0.33±0.29	0.44	0.53±0.23	0.65±1.18	0.06	0.30±0.20	0.94±1.00	0.25	0.49±0.19	**0.34±0.29**
Gemini-2.5-pro-0256	**0.65**	**0.68±0.26**	3.92±14.29	**0.32**	**0.62±0.26**	**0.31±0.27**	**0.48**	**0.58±0.23**	**0.59±1.09**	**0.08**	**0.37±0.23**	**0.89±0.93**	**0.27**	**0.54±0.23**	0.55±0.44
GPT-4o-4096	0.11	0.03±0.10	9.20±13.35	0.00	0.00±0.00	9.69±7.21	0.01	0.02±0.07	4.62±6.27	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.
GPT-4o-2048	0.21	0.11±0.18	7.30±17.17	0.00	0.01±0.02	3.90±4.46	0.00	0.04±0.04	1.74±2.09	0.00	0.01±0.02	6.08±4.06	0.09	0.09±0.15	1.23±0.67
GPT-4o-1024	0.33	0.24±0.22	1.53±2.82	0.00	0.10±0.04	0.88±0.29	0.02	0.13±0.09	1.02±1.41	0.05	0.05±0.10	12.89±15.16	0.00	0.00±0.00	N.A.
GPT-4o-0512	0.34	0.24±0.23	1.63±3.41	0.06	0.18±0.11	0.66±0.37	0.04	0.18±0.14	0.89±1.30	0.02	0.12±0.12	1.93±1.84	0.16	0.24±0.18	0.59±0.40
GPT-4o-0256	**0.58**	**0.55±0.24**	1.42±4.86	**0.24**	**0.38±0.23**	**0.38±0.25**	**0.16**	**0.31±0.24**	**0.84±1.38**	0.05	0.27±0.17	1.06±1.14	**0.38**	**0.53±0.20**	**0.35±0.35**
GPT-4o-0128	0.17	0.38±0.32	**1.25±2.28**	0.18	0.31±0.28	0.42±0.19	0.13	0.27±0.28	0.99±1.43	**0.13**	**0.41±0.22**	**0.85±0.98**	0.20	0.38±0.25	0.54±0.61
GPT-5.2-pro	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.
Claude-sonnet-4.5	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.
Claude-opus-4.6	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.	0.00	0.00±0.00	N.A.

For GPT 5.2 Pro, Claude Sonnet 4.5, and Claude Opus 4.6, they fail to return any polygon geometry across various tiling sizes, which may be due to their focus on code processing. In addition, we fail to complete the Gemini and SAM series with a tile size of 128×128 due to their high monetary cost and runtime, respectively.

In terms of instance-level quality, our proposed CLYPH achieves the best MMPQ scores across all five datasets compared to other methods. This shows that legend-guided semantic fusion effectively preserves the topological structure of polygons compared to individual experts. For pixel-level boundary alignment, GLYPH achieves high F1@8 scores and the best quantitative results across four out of five datasets, demonstrating that the fusion mechanism does not sacrifice local geometric precision for global consistency. GLYPH slightly falls short of LOAM on the WR dataset in F1@8, but it significantly outperforms LOAM in MMPQ.

Regarding the estimated manual post-editing effort, GLYPH achieves the lowest NBDR for three out of five datasets. Compared to most other methods, GLYPH can still achieve a decent reduction in manual correction in a modern digitization pipeline. When assessing NBDR results, the complexity of the polygon geometry can significantly affect the results. Therefore, the NBDR results across all methods for the FT and SP datasets tend to be higher than those for the other three datasets.

In addition, by combining pixel-based (LOAM) and instance-based (SAM2 and Gemini 3 Flash) results, GLYPH aims to produce fine-grained polygonal geometry, resulting in a large number of vertices. Therefore, although individual experts, including LOAM or Gemini 3 Flash, demonstrate competitive performance on WR or FT datasets in NBDR, their NBDR performance fluctuates significantly across multiple datasets, and they often have worse instance-level (MMPQ) or pixel-level (F1@8) accuracy compared to GLYPH. For instance, LOAM outperforms GLYPH in NBDR but performs worse in MMPQ for the WR dataset, whereas Gemini 3 Flash surpasses GLYPH and LOAM in NBDR for the FT datasets but performs worse in MMPQ and F1@8 compared to both GLYPH and LOAM. Most importantly, comparative methods such as Gemini 3 Flash and Gemini 3.1 Pro do not

always outperform GLYPH and LOAM in NBDR across all their tiling sizes. The best setting for these comparative methods varies across datasets and is difficult to estimate before using them.

For segmentation-based methods, while SAM2 consistently achieves competitive quantitative accuracy in MMPQ and F1@8 across all five datasets, sometimes outperforming the Gemini series at the same tiling sizes, it still fails to achieve the best performance in NBDR. On the other hand, SAM3 surprisingly performs significantly worse across all five datasets.

Although SAM3 can directly take image exemplars as input, its accuracy under this setup is statistically significantly worse than that of SAM2 with entity linking across all tiling sizes and five datasets. We notice that SAM3 seems to fail to effectively leverage the input ontology to correctly perform polygonal entity linking to the targeted map keys. This results in either no polygon features being identified or, in most cases, polygon features being assigned to multiple map keys. We attribute the SAM3's failure to three reasons. First, the image exemplars (polygon map keys) are always in a rectangular shape, as shown in the upper portion of Figure 5.8 and in Figure 5.1. However, the actual polygon features in the map content area are not in a rectangular or similar shape, as shown in the lower portion of Figure 5.8. This nature may make the image exemplars too implicit as a concept reference for SAM3. As a comparison, our LOAM (Chapter 2) explicitly applies text pattern matching and combines it with color-based results. The second reason is the over-complicated ontology indicated by the input image exemplars. As we provide all remaining polygon map keys in the map as negative image exemplars, SAM3 might be overwhelmed by the ontology of potential instances and the negative weighting for most of its identified instances. Third, there are significantly fewer adjustable parameters for SAM3 with this image-exemplar setup than for SAM2 with the segment-anything setup we exploited. This might limit the granularity of the instances that SAM3 can find. However, this third hypothesis might not be the primary reason for SAM3's significant failure, since we already tried SAM3 with all tiling sizes, including 256×256, in which each image tile includes only a small portion and a

limited number of potential polygon features.

Most importantly, this observation supports our design choice regarding SAM2 to use its segment-anything setup and then link entities to support GLYPH. Moreover, the significant difference shows that the general research problem of this dissertation, legend-item-based polygonal feature digitization from historical maps, remains challenging for the state-of-the-art general segmentation model that can directly incorporate image (concept) prompts. This comparison with SAM3, using their dedicated setup of positive and negative image exemplars, demonstrates the efficacy of our proposed GLYPH under a zero-shot scenario.

It should be noted that the three metrics (MMPQ, F1@8, and NBDR) capture complementary aspects of polygon extraction quality and can exhibit distinct trends. The NBDR is an estimated manual post-editing effort adapted based on the current post-editing workflow and domain expert evaluations [22], rather than an accuracy assessment at the instance- or pixel-level. The actual post-editing effort might vary significantly depending on the post-editing system or user interface used, and NBDR might no longer be applicable or meaningful in those scenarios. However, MMPQ and F1@8 themselves are universal metrics for assessing quantitative accuracy and may not change significantly across different scenarios or downstream purposes.

The proposed GLYPH achieves the best instance-level accuracy (MMPQ) across five out-of-domain datasets, surpassing the dedicated learning-based approach LOAM, VLMs, and segmentation-based methods. In addition, GLYPH also achieves the best pixel-level accuracy (F1@8) for four out of five datasets. This demonstrates GLYPH's ability to generalize across cartographic styles not encountered in the training dataset.

5.4.4.2 Discussion on Comparative Methods

For **EQ2** (*How does tiling size affect the performance of comparative methods?*), regarding individual expert solutions and other comparative methods, the trends vary across three evaluation metrics. For the instance-level accuracy (MMPQ), VLMs such as Gemini 3 Flash

and Gemini 3.1 Pro, under some ideal settings, can outperform a dedicated learning-based approach LOAM on the FT and SA datasets, where polygon features in the map content area are either distinguishable from one another or more regular in shape. However, for pixel-level boundary alignment (F1@8), many VLMs and SAM2, under some ideal settings, can surpass LOAM in SA and SP datasets, in which most polygon features are of regular shape. This is also supported by the lower NBDR of Gemini 3 Flash, Gemini 3.1 Pro, Gemini 2.5 Pro, and GPT 4o in SA and SP datasets compared to LOAM. For the segmentation-based method, SAM2 outperforms the dedicated learning-based approach when the number of polygon map keys per map is limited. For SP datasets with fewer than 2 polygon map keys per map, SAM2 consistently outperforms LOAM across various tile sizes across most evaluation metrics. For the SA and WR datasets, where there are approximately 4 polygon map keys per map, SAM2 achieves a competitive MMPQ compared to LOAM.

For VLMs, while GPT 5.2 Pro and Claude series failed, Gemini 3 Flash achieves competitive accuracy for most datasets compared to the dedicated LOAM. Followed by Gemini 3.1 Pro, Gemini 2.5 Pro, and GPT 4o. For the SP and WR datasets, GPT 4o tends to achieve higher accuracy than the Gemini series. This may be attributed to CMYK and offset printing noise, as well as the significant degradation of the maps caused by these printing techniques.

For comparative methods that operate on cropped inputs, we evaluate multiple tile sizes (e.g., 4096, 2048, 1024, 512, etc.) to test whether smaller crops mitigate domain shift or reduce boundary ambiguity. We observe that tiling can affect performance for both VLM-based and segmentation-based methods, but the optimal tile size varies across datasets. With a smaller tile size, although methods can examine detailed colors and patterns in the image, they lose the overview, resulting in a binary decision on whether to include a full image tile and often yielding bitmap-like results. For VLMs, this is supported by the increase in NBDR observed across four datasets as tile size decreases. The SP dataset is the only case in which decreasing tile size consistently improves NBDR, due to its simpler polygon geometry and fewer polygon map keys per map. In contrast, SAM2 achieves consistent performance

across three evaluation metrics but shows fluctuations with tile size, with no clear trend across datasets. Similarly, while SAM3 fails in most cases with the prompt of Figure 5.8, its performance fluctuates with tile size, with no clear trend across datasets.

As indicated in the previous subsection, the optimal settings for these comparative methods vary significantly across datasets and tiling sizes, and it is extremely difficult to estimate them before use.

5.4.4.3 Statistical Analysis

For **EQ 3** (*Is there a statistical significance between GLYPH and comparative methods?*), to evaluate the statistical significance of our results, we conduct a one-way ANOVA followed by Fisher's Least Significant Difference (LSD) test at $\alpha = 0.05$. One-way ANOVA can show whether there is a significant difference between solutions derived by different methods. Applying Fisher's LSD can then cluster all methods into tiers based on their performance. We summarize the grouping results in Table 5.4, where "A" denotes the statistically best group. These statistical groupings reflect distributional overlap across maps rather than a simple ordering of dataset-level means; thus, multiple methods may share the same group if their performance differences are not statistically significant. Similarly, methods may have better average performance at the dataset level in Table 5.3 but worse grouping results here in Table 5.4. For instance, a method performs decently across all maps, while the other method performs statistically significantly better on most maps but fails drastically on a small subset. The first method may have better average performance, but the second method may be clustered to a better group in Fisher's LSD test.

The ANOVA p-values reported in the final rows of Table 5.4 are consistently extremely small, lower than e^{-7} across metrics and datasets. This indicates that there exist statistically significant differences between solutions from all methods across metrics and datasets. Our proposed GLYPH is the only approach that consistently achieves the "A" group across all metrics (MMPQ, P@8, R@8, F1@8, and NBDR) and all five datasets.

The dedicated learning-based approach LOAM frequently ranks in the "A" group for P@8, F1@8, and NBDR across some datasets; it falls into lower groups for MMPQ and R@8. Similarly, SAM2 and Gemini series can reach the "A" group on specific metrics when configured with optimal tile sizes, but they fail to maintain this standing across all evaluation categories. These results show that our proposed fusion framework provides a statistically significant improvement and a more stable solution for cross-domain polygon extraction than any single expert model or baseline configuration.

5.4.4.4 Complexity and Cost Analysis

For **EQ 4** (*How is the trade-off of accuracy, runtime, and monetary cost in deployment?*), we present the average runtime and API cost per map for our proposed GLYPH and the comparative methods in Table 5.5. The runtime and cost for GLYPH in the table already include the three incorporated expert models. For API-based methods, e.g., VLMs, the runtime may not reflect algorithmic efficiency as it is dominated by network and external service latency or paid-tier limitations. Still, we tried to maintain a fair comparison among methods; both the Gemini API and the OpenAI API are called at their highest tiers.

Combined with the quantitative results presented in Table 5.3, we observe a trade-off between extraction granularity, computational overhead, and financial cost. For large VLMs, reducing the tile size to 256×256 or 128×128, if applicable, often improves local extraction accuracy but leads to an exponential increase in API costs. Gemini 3.1 Pro and Gemini 2.5 Pro with 256×256 tile size and GPT 4o with 128×128 tile size both reach a skyrocketing cost of around $60 per map, making it economically unfeasible for large-scale map archive digitization. Similarly, while smaller tile sizes improve the boundary alignment of SAM2, the computational cost becomes prohibitive. SAM2 with 256×256 tile size requires over 100 minutes to process a single map. Although SAM3 is more efficient in terms of time spent than SAM2, its accuracy on MMPQ, F1@8, and NBDR is significantly worse than that of SAM2.

Table 5.4: Fisher's LSD grouping results across datasets and evaluation metrics. Each cell shows the Fisher-LSD grouping letter by one-way ANOVA and Fisher's LSD test at α=0.05 over map-level quantitative results; "A" denotes the statistically best group. For each metric, the best group within a method family is in bold, and the best group overall is in red. The last two rows report the one-way ANOVA p-value and the LSD threshold. We use $log(NBDR)$ to stabilize variance across methods in the NBDR metric, and p-values are shown in a compact scientific form e_m^n to denote $m \times 10^n$. P@8, R@8, and F1@8 refer to precision, recall, and F1 score, respectively, with a tolerance radius of 8 pixels. "N.A." indicates NBDR is not applicable when a method returns empty results across all cases. The "$\rightarrow 0$" indicates that the p-value is smaller than $5e^{-324}$ and underflows to zero in double precision.

Dataset / Metric	FT					SA					SO					SP					WR				
Method - Tile Size	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR
GLYPH-1024 (Ours)	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
LOAM-1024	**A**	**A**	**B**	**A**	**A**	**C**	**A**	**C**	**C**	**C**	**A**	**A**	**A**	**A**	**A**	**B**	**B**	**A**	**B**	**A**	**B**	**A**	**A**	**A**	**A**
SAM2-4096	D	**B**	E	D	**C**	**B**	**B**	**B**	**C**	**B**	C	**A**	D	C	**B**	**A**	**A**	B	**A**	**A**	C	**B**	D	C	**B**
SAM2-2048	**C**	**B**	**D**	**C**	**C**	**B**	**B**	**B**	**C**	**B**	**B**	**A**	C	**B**	**B**	**A**	**A**	**A**	**A**	**A**	**B**	**B**	C	**B**	**B**
SAM2-1024	**C**	**B**	**D**	**C**	**C**	**B**	**B**	**B**	**C**	**B**	**B**	B	C	**B**	**B**	**A**	**A**	**A**	**A**	**A**	D	**B**	C	**B**	**B**
SAM2-0512	**C**	**B**	**D**	**C**	**C**	**B**	**B**	C	**C**	**B**	**B**	B	C	**B**	**B**	**A**	**A**	**A**	**A**	**A**	C	**B**	D	C	**B**
SAM2-0256	D	**B**	**D**	**C**	**C**	**B**	**B**	**B**	**C**	**B**	**B**	**A**	**B**	**B**	**B**	**A**	**A**	**A**	**A**	**A**	**B**	**B**	**B**	**B**	**B**
SAM3-4096	**H**	H	I	H	G	**F**	F	F	G	G	**G**	G	H	**H**	G	B	**B**	B	**B**	B	**E**	**F**	**G**	**F**	G
SAM3-2048	**H**	H	I	H	G	**F**	F	**E**	**F**	**F**	**G**	G	H	**H**	G	**A**	**B**	B	**B**	**A**	**E**	**F**	**G**	**F**	G
SAM3-1024	**H**	H	I	H	G	**F**	**E**	F	**F**	**F**	**G**	G	H	**H**	F	B	**B**	**A**	**B**	**A**	**E**	G	**G**	**F**	**F**
SAM3-0512	**H**	**G**	**H**	**G**	**F**	**F**	**E**	F	**F**	**F**	**G**	G	**G**	**H**	**E**	B	**B**	**A**	**B**	**A**	**E**	G	**G**	**F**	**F**
SAM3-0256	**H**	H	I	H	G	**F**	**E**	F	**F**	**F**	**G**	**F**	H	**H**	**E**	B	**B**	B	**B**	**A**	**E**	G	**G**	**F**	**F**
Gemini-3-flash-4096	E	E	E	E	C	D	C	D	D	C	E	D	E	E	C	B	B	B	B	**A**	E	E	F	E	D
Gemini-3-flash-2048	E	D	D	D	C	D	C	D	D	C	E	D	E	D	C	B	B	B	B	**A**	D	E	E	D	**C**
Gemini-3-flash-1024	C	C	B	C	**B**	**B**	**B**	**A**	**B**	**B**	D	**C**	C	C	**B**	**A**	**A**	B	**A**	**A**	C	D	C	C	**C**
Gemini-3-flash-0512	C	C	B	**B**	**B**	C	C	B	C	C	**C**	**C**	C	C	**B**	**A**	B	**A**	B	**A**	**B**	**C**	**A**	**B**	**C**
Gemini-3-flash-0256	**B**	**B**	**A**	**B**	**B**	**B**	C	**A**	**B**	C	**C**	**C**	**B**	**B**	C	**A**	B	**A**	B	**A**	**B**	**C**	**A**	**B**	**C**
Gemini-3.1-pro-4096	H	G	H	G	E	F	E	F	F	E	F	F	G	G	D	B	B	C	C	B	E	F	G	F	E
Gemini-3.1-pro-2048	G	E	F	F	C	E	C	E	E	D	F	E	F	F	C	C	B	D	C	B	E	F	F	E	D
Gemini-3.1-pro-1024	D	D	C	D	**B**	**B**	**B**	**A**	**B**	**B**	D	**C**	D	**C**	**B**	**A**	**A**	**B**	**A**	**A**	C	D	C	C	C
Gemini-3.1-pro-0512	**C**	C	C	C	**B**	C	**B**	C	C	**B**	D	**C**	D	**C**	**B**	**A**	**A**	**B**	**A**	**A**	**B**	**C**	B	**B**	**B**
Gemini-3.1-pro-0256	**C**	**B**	**B**	**B**	**B**	D	C	C	D	**B**	**C**	**C**	**C**	**C**	**B**	**A**	**A**	**B**	**A**	**A**	**B**	**C**	**A**	**B**	C
Gemini-2.5-pro-4096	G	G	G	G	E	F	F	F	F	E	G	F	G	G	D	B	**B**	C	C	B	E	F	F	F	D
Gemini-2.5-pro-2048	G	F	F	F	C	F	F	F	F	D	F	E	F	F	C	B	**B**	C	**B**	**A**	E	E	E	E	D
Gemini-2.5-pro-1024	E	E	E	E	C	D	D	C	D	C	E	D	E	D	C	B	**B**	B	**B**	**A**	D	E	D	D	**C**
Gemini-2.5-pro-0512	D	**C**	D	D	**B**	**C**	**C**	B	**C**	C	**D**	D	D	D	**B**	B	**B**	B	**B**	**A**	**C**	D	C	C	**C**
Gemini-2.5-pro-0256	**C**	**C**	**C**	**C**	**B**	**C**	**C**	**A**	**C**	**B**	**D**	**C**	**C**	**C**	**B**	**A**	**B**	**A**	**B**	**A**	**C**	**C**	**B**	**B**	**C**
GPT-4o-4096	H	G	H	H	F	F	G	G	G	G	G	F	H	H	F	C	C	D	C	N.A.	E	G	G	F	N.A.
GPT-4o-2048	G	G	H	G	E	F	F	G	G	F	G	F	H	H	D	C	C	D	C	B	E	F	G	F	E
GPT-4o-1024	G	F	F	F	D	F	D	F	F	E	G	F	G	G	D	**B**	C	C	C	B	E	G	G	F	N.A.
GPT-4o-0512	G	E	G	F	C	F	**C**	F	F	D	G	E	G	G	**C**	**B**	B	C	C	**A**	E	D	F	E	D
GPT-4o-0256	**D**	**D**	**D**	**D**	**B**	**E**	**C**	**E**	**E**	**C**	**F**	**D**	**F**	**F**	**C**	**B**	B	**B**	B	**A**	**C**	**C**	**C**	**C**	**C**
GPT-4o-0128	F	E	E	E	C	**E**	D	**E**	**E**	**C**	**F**	E	**F**	**F**	**C**	**B**	**A**	**B**	**A**	**A**	D	D	**C**	D	**C**
GPT-5.2-pro	H	H	I	H	N.A.	F	G	G	G	N.A.	G	G	H	H	N.A.	C	C	D	C	N.A.	E	G	G	F	N.A.
Claude-sonnet-4.5	H	H	I	H	N.A.	F	G	G	G	N.A.	G	G	H	H	N.A.	C	C	D	C	N.A.	E	G	G	F	N.A.
Claude-opus-4.6	H	H	I	H	N.A.	F	G	G	G	N.A.	G	G	H	H	N.A.	C	C	D	C	N.A.	E	G	G	F	N.A.
ANOVA p-value	e_2^{-242}	e_9^{-227}	$\rightarrow 0$	e_3^{-291}	e_1^{-194}	e_5^{-157}	e_8^{-152}	e_3^{-198}	e_6^{-190}	e_4^{-232}	e_1^{-252}	e_9^{-214}	$\rightarrow 0$	e_6^{-306}	e_4^{-171}	e_8^{-7}	e_1^{-9}	e_3^{-29}	e_4^{-14}	e_1^{-16}	e_3^{-76}	e_2^{-121}	e_9^{-145}	e_4^{-127}	e_4^{-143}
LSD (α=0.05)	0.08	0.11	0.09	0.09	0.77	0.10	0.12	0.11	0.11	0.51	0.08	0.11	0.08	0.09	0.71	0.17	0.22	0.25	0.22	1.47	0.11	0.13	0.12	0.12	0.65

GLYPH provides a highly effective balance for practical applications. With parallel processing, the runtime for GLYPH is bounded by the longest one among three experts: LOAM-1024, Gemini-3-flash-1024, and SAM-1024. By utilizing expert masks generated at a fixed 1024×1024 tile size, it achieves the state-of-the-art accuracy with a moderate runtime of approximately 11 minutes per map and a significantly lower API cost of around $0.38 per map. This efficacy makes GLYPH a scalable solution for processing vast historical map collections where both time and budget are limiting factors. Additionally, when time or budget are not limiting factors, GLYPH's scalability and modularity enable integration with expert models at fine-grained tile sizes when appropriate.

5.4.4.5 Estimated Benefits to Post-editing Effort

For **EQ 5** (*What is the potential improvement to post-editing effort by GLYPH?*), we can estimate potential labor savings by comparing GLYPH's NBDR with the baseline NBDR of models that return nearly nothing. As shown in Table 5.2, full manual digitization typically requires several hours per map.

Based on Table 5.3, the reduction in NBDR for GLYPH indicates that the vast majority of polygon vertices and structures are correctly placed automatically. In the SA and WR datasets, the "first draft" of GLYPH is geometrically nearly complete, leaving only minor topological refinements for manual post-editing. Even in the complex SP dataset, with an over 90% reduction in NBDR, the labor-intensive process of identifying and tracing irregular boundaries is significantly mitigated.

As mentioned in the previous subsection, the exact post-editing time may not be strictly linear with geometric distance for vertex correction and may depend heavily on UI efficiency. However, based on the results presented in DIGMAPPER [22], which includes polygon extraction accuracy similar to LOAM, the post-editing time can be reduced by more than 30-fold. Accordingly, GLYPH's performance still suggests a transformative shift in productivity regarding out-of-domain polygon digitization from historical archives.

Table 5.5: Average runtime and API cost per map. For API-based methods, their runtime is dominated by external service latency or limitations and may not be algorithmically meaningful. N.A. indicates that no API request is required. The best performance within a method family is in bold.

Method - Tile Size	Avg. Time / Map (min.)	Avg. API Cost / Map (USD)
GLYPH-1024 (Ours)	10.68	0.38
LOAM-1024	**10.12**	N.A.
SAM2-4096	7.08	N.A.
SAM2-2048	**5.37**	N.A.
SAM2-1024	7.55	N.A.
SAM2-0512	28.78	N.A.
SAM2-0256	101.23	N.A.
SAM3-4096	**0.52**	N.A.
SAM3-2048	0.75	N.A.
SAM3-1024	1.68	N.A.
SAM3-0512	5.55	N.A.
SAM3-0256	27.37	N.A.
Gemini-3-flash-4096	**1.93**	**0.04**
Gemini-3-flash-2048	2.95	0.12
Gemini-3-flash-1024	3.58	0.38
Gemini-3-flash-0512	4.90	0.77
Gemini-3-flash-0256	28.04	6.92
Gemini-3.1-pro-4096	**4.34**	**0.34**
Gemini-3.1-pro-2048	14.77	1.13
Gemini-3.1-pro-1024	8.50	6.98
Gemini-3.1-pro-0512	13.02	17.96
Gemini-3.1-pro-0256	28.12	65.76
Gemini-2.5-pro-4096	17.82	**0.39**
Gemini-2.5-pro-2048	9.37	0.86
Gemini-2.5-pro-1024	**5.37**	3.14
Gemini-2.5-pro-0512	11.52	6.36
Gemini-2.5-pro-0256	36.66	57.18
GPT-4o-4096	5.42	0.31
GPT-4o-2048	2.48	**0.19**
GPT-4o-1024	2.82	0.58
GPT-4o-0512	**1.20**	1.54
GPT-4o-0256	2.17	10.38
GPT-4o-0128	15.52	58.46

By providing a high-quality first draft that achieves decent structural correctness (MMPQ) and boundary alignment (F1@8) across nearly all datasets, GLYPH can drastically shorten the path from raw archival image to digitized linked data.

5.4.4.6 Ablation Study for GLYPH

For **EQ 6** (*How does each component in GLYPH contribute to the final outputs?*), we conduct an ablation study to remove or replace components in our proposed GLYPH to assess whether each component contributes to the final output. We present the results of our ablation study in terms of MMPQ, F1@8, and NBDR in Table 5.6. Similar to our main evaluation, we conduct a one-way ANOVA followed by Fisher's Least Significant Difference (LSD) test at $\alpha = 0.05$ to assess the statistical significance in the ablation study and summarize the grouping results in Table 5.7.

We include the following ablation setups:

- **GLYPH**: The standard GLYPH pipeline.
- ⇒ **RGB space**: Use RGB color space instead of CIELAB color space, while keeping the remaining components the same.
- ⇒ **Pixel majority**: Apply only pixel-wise majority voting among three expert models. (keep only Section 5.2.3.1)
- ⇒ **Region majority**: Apply only region-wise majority voting among the expert models. (keep only Section 5.2.3.1 + Section 5.2.3.2)
- **- Psuedo**: The training signal of contrastive learning for color embedding does not consider cross-expert consensus pseudo-labels; it relies only on input map keys. (change in Section 5.2.4.1)
- **- Anchor**: The training signal of contrastive learning for color embedding does not consider input map keys; it relies only on cross-expert consensus pseudo-labels. (change in Section 5.2.4.1)

- **- Gating**: The gating in test-time adaptation is not optimized from global style or reliability features; instead, it directly applies uniform weighting for semantic fusion. change in Section 5.2.4.1)
- **- Evidence**: The semantic fusion does not consider expert evidence; it relies only on similarity between region embedding and legend prototypes.
  (change in Section 5.2.4.2)
- **- Similarity**: The semantic fusion does not consider similarity between region embedding and legend prototypes; it relies only on expert evidence.
  (change in Section 5.2.4.2)
- **- Post**: The structural and geometric post-processing is skipped.
  (removal of Section 5.2.4.3)

Regarding the quantitative results, integrating all components tends to yield better results, or at most 0.03 variance, in MMPQ and F1@8. The two exceptions appear at the MMPQ of the SA dataset and the F1@8 of the SP dataset. However, the decrease is due to a different component in GLYPH. In the SA dataset case, uniform weighting for semantic fusion ("- Gating") yields better MMPQ than test-time adaptation. This may be due to the significant color shift and insufficient cues from the polygon map keys; these potentially misleading cues limit the ability to learn meaningful geometric information from experts' solutions and refine the final solutions accordingly. This is supported by the similar F1@8 and NBDR between GLYPH and "- Gating". Another case is the SP dataset; conducting contrastive learning for color embedding solely on cross-expert consensus pseudo-labels ("- Anchor") seems to outperform learning with the integration of input map keys. This is due to the extremely small number of polygon map keys per map (e.g., 2 or fewer), which naturally limits the efficacy of contrastive learning.

The ANOVA p-values reported in the final rows of Table 5.7 are lower than 0.05 for R@8 except in the SP dataset. While p-values are lower than 0.05 for MMPQ, F1@8, and NBDR

Table 5.6: Ablation study of GLYPH across datasets. For each evaluation metric, the best performance is in red, and the second-best performance is in bold. Values are presented in mean±std unless otherwise noted.

Dataset / Metric	FT			SA			SO			SP			WR		
Ablation Setting	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓
GLYPH	**0.90**	**0.90±0.16**	0.48±1.27	0.69	0.87±0.16	**0.12±0.16**	0.81	**0.91±0.18**	**0.36±1.12**	**0.30**	**0.59±0.21**	0.79±0.82	**0.65**	**0.82±0.13**	0.15±0.16
⇒RGB space	**0.84**	0.86±0.15	0.53±1.32	0.63	0.81±0.14	0.14±0.09	0.76	0.88±0.25	0.36±1.08	0.13	0.36±0.23	2.62±2.98	0.41	0.73±0.24	0.32±0.30
⇒Pixel majority	0.76	0.82±0.21	0.35±0.92	0.58	0.76±0.22	0.17±0.15	0.72	0.87±0.24	0.37±1.02	0.18	0.49±0.22	0.82±0.85	0.61	**0.82±0.18**	**0.10±0.18**
⇒Region majority	0.72	0.77±0.21	0.46±1.01	0.63	0.78±0.20	0.16±0.16	0.75	0.84±0.21	0.36±0.87	0.18	0.52±0.23	0.72±0.84	0.51	0.75±0.17	0.15±0.20
-Psuedo	0.83	**0.93±0.12**	**0.30±0.79**	0.64	0.85±0.18	0.14±0.17	**0.81**	0.88±0.22	0.39±1.19	0.15	0.53±0.29	0.89±0.98	0.52	0.73±0.20	0.21±0.22
-Anchor	0.81	0.88±0.18	0.47±1.06	0.66	0.82±0.20	0.16±0.19	0.79	0.89±0.22	**0.35±0.96**	**0.36**	**0.65±0.23**	**0.65±0.99**	0.49	0.71±0.24	0.21±0.23
-Gating	0.81	0.89±0.17	0.33±0.82	**0.78**	**0.88±0.17**	**0.11±0.17**	0.77	0.88±0.24	0.38±1.10	0.16	0.53±0.24	1.21±1.04	**0.66**	0.81±0.16	**0.14±0.17**
-Evidence	0.82	0.87±0.22	**0.32±0.85**	0.71	0.79±0.33	0.29±0.54	0.63	0.81±0.27	0.42±1.26	0.15	0.40±0.34	4.29±6.53	0.60	0.74±0.25	0.16±0.20
-Similarity	0.60	0.68±0.25	1.98±9.61	0.46	0.65±0.24	0.18±0.13	0.74	0.79±0.20	0.48±1.01	0.25	0.55±0.26	**0.68±1.05**	0.21	0.47±0.22	0.29±0.25
-Post	0.82	0.89±0.17	0.43±0.96	**0.73**	**0.87±0.15**	0.14±0.17	**0.82**	**0.89±0.22**	0.39±1.15	0.26	0.56±0.26	0.95±1.10	0.54	0.75±0.21	0.19±0.22

Table 5.7: Fisher's LSD grouping results across datasets and evaluation metrics under various ablation setups of GLYPH. Each cell shows the Fisher-LSD grouping letter from one-way ANOVA and Fisher's LSD test at α=0.05 over map-level quantitative results; "A" denotes the statistically best group. For each metric, the best group is in red. The last two rows report the one-way ANOVA p-value and the LSD threshold. We use log(NBDR) to stabilize variance across methods in the NBDR metric, and p-values are shown in a compact scientific form e_m^n to denote $m \times 10^n$. P@8, R@8, and F1@8 refer to precision, recall, and F1 score, respectively, with a tolerance radius of 8 pixels.

Dataset / Metric	FT					SA					SO					SP					WR				
Ablation Setting	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR
GLYPH	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
⇒RGB space	A	B	A	A	A	A	A	A	A	B	A	A	A	A	A	A	B	A	B	A	A	B	A	A	B
⇒Pixel majority	B	A	B	B	A	B	A	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	B	A	A
⇒Region majority	B	A	B	B	B	B	A	B	A	B	A	A	B	A	A	A	A	A	A	A	A	A	B	A	A
-Psuedo	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	A	B
-Anchor	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	A	A
-Gating	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	A	A
-Evidence	A	A	A	A	A	A	A	A	A	A	A	A	C	A	A	A	A	B	A	A	A	A	B	A	A
-Similarity	C	B	C	C	B	C	B	C	B	B	B	A	C	B	B	A	A	A	A	A	B	A	C	B	B
-Post	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	A	A
ANOVA p-value	e_2^{-9}	e_9^{-3}	e_4^{-12}	e_1^{-7}	e_3^{-2}	e_2^{-4}	e_4^{-1}	e_7^{-5}	e_4^{-3}	e_1^{-2}	e_2^{-1}	e_9^{-1}	e_5^{-9}	e_5^{-1}	e_2^{-1}	e_8^{-1}	e_5^{-1}	e_6^{-1}	e_7^{-1}	e_8^{-1}	e_3^{-4}	e_2^{-1}	e_3^{-11}	e_4^{-4}	e_9^{-2}
LSD (α=0.05)	0.10	0.07	0.09	0.09	0.95	0.13	0.09	0.13	0.11	0.63	0.13	0.11	0.06	0.10	1.00	0.27	0.32	0.31	0.29	1.87	0.15	0.16	0.13	0.15	0.86

in the FT, SA, and WR datasets, FT dataset is the only one that has its p-values for P@8 lower than 0.05. It shows that ablation of components in GLYPH mostly affects extraction recall and has no statistically significant effect on extraction precision. We believe that the FT, SA, and WR may be viewed as out-of-domain datasets with moderate difficulty, leading to statistically different performance across ablation setups. SO can be interpreted a "simpler" out-of-domain dataset as it has a similar characteristics compared to the in-domain GE dataset. In contrast, SP can be considered a "difficult" out-of-domain dataset, as the printing techniques, degradation, and the number of polygon map keys per map are significantly different from those in the GE dataset.

Although integrating all components does not yield the best quantitative results among all ablation setups, it is the only setup to achieve the statistically best-performing group across all evaluation metrics and datasets. All the remaining ablation setups have at least one metric or dataset not achieving the "A" rank based on the Fisher's LSD test. The results of Fisher's LSD test support the design of our proposed GLYPH for generalizing polygon extraction across historical maps with diverse styles.

5.4.4.7 Parameter Setting for GLYPH

For **EQ 7** (*How sensitive is parameter setting to the final outputs of GLYPH?*), the main adjustable parameter in GLYPH is its learning rate. We present the quantitative evaluation results on the parameter setting of learning rate in GLYPH in Table 5.8. To assess the statistical significance among various settings, we conduct a one-way ANOVA followed by Fisher's LSD test at $\alpha = 0.05$ and summarize the results in Table 5.9.

We notice that there is no significant difference nor clear trend in terms of performance across different learning rate settings for GLYPH. The best performance appears at different learning rate values across both evaluation metrics and datasets. This is supported by the high ANOVA p-values and the fact that, almost all setups achieve the statistically best group. However, the worst performance seems to appear at the highest or the lowest learning rate.

Table 5.8: Evaluation on parameter setting of GLYPH across datasets. For each evaluation metric, the best performance is in red. Values are presented in mean±std unless otherwise noted.

Dataset / Metric	FT			SA			SO			SP			WR		
Learning Rate	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓
1e-2	0.64	0.71±0.24	0.64±1.38	0.66	0.80±0.19	0.15±0.13	**0.81**	0.89±0.23	0.39±1.18	0.16	0.44±0.18	1.11±0.90	0.38	0.76±0.26	0.16±0.19
5e-3	0.85	0.89±0.16	0.46±1.31	0.77	0.89±0.16	0.11±0.16	0.74	0.86±0.24	0.46±1.19	0.27	0.57±0.23	0.82±0.89	0.40	0.71±0.25	0.19±0.23
1e-3	0.86	**0.91±0.14**	0.46±1.33	**0.79**	**0.90±0.15**	0.11±0.16	0.74	0.86±0.23	0.39±1.13	0.23	0.57±0.25	0.84±0.82	0.55	0.81±0.21	0.17±0.21
5e-4	0.85	0.86±0.19	0.64±1.62	0.77	0.88±0.16	**0.09±0.09**	0.75	0.88±0.24	0.36±1.10	0.27	0.56±0.16	**0.72±0.76**	0.48	0.76±0.25	0.18±0.22
1e-4	0.87	0.87±0.20	0.59±1.56	0.70	0.83±0.19	0.11±0.13	0.75	0.87±0.21	0.37±1.04	0.23	0.52±0.25	0.94±0.94	0.52	0.73±0.24	0.19±0.22
5e-5	**0.90**	0.90±0.16	0.48±1.27	0.69	0.87±0.16	0.12±0.16	0.81	**0.91±0.18**	0.36±1.12	**0.30**	**0.59±0.21**	0.79±0.82	0.65	0.82±0.13	0.15±0.16
1e-5	0.80	0.80±0.23	0.51±1.06	0.56	0.76±0.24	0.19±0.20	0.66	0.87±0.25	0.39±0.97	0.13	0.57±0.30	0.96±1.23	0.60	0.82±0.20	0.19±0.23
5e-6	0.85	0.88±0.14	**0.36±0.83**	0.55	0.76±0.19	0.20±0.15	0.72	0.86±0.26	0.39±1.11	0.12	0.51±0.28	1.18±1.33	0.57	0.83±0.21	0.20±0.23
1e-6	0.71	0.83±0.25	1.74±9.62	0.49	0.67±0.26	0.24±0.17	0.65	0.85±0.27	**0.34±0.99**	0.08	0.34±0.21	1.50±1.16	**0.73**	**0.86±0.11**	**0.10±0.10**

Table 5.9: Fisher's LSD grouping results across datasets and evaluation metrics under various parameter settings of GLYPH. Each cell shows the Fisher-LSD grouping letter by one-way ANOVA and Fisher's LSD test at α=0.05 over map-level quantitative results; "A" denotes the statistically best group. For each metric, the best group overall is in red. The last two rows report the one-way ANOVA p-value and the LSD threshold. We use log(NBDR) to stabilize variance across methods in the NBDR metric, and p-values are shown in a compact scientific form e_m^n to denote $m \times 10^n$. P@8, R@8, and F1@8 refer to precision, recall, and F1 score, respectively, with a tolerance radius of 8 pixels.

Dataset / Metric	FT					SA					SO					SP					WR				
Learning Rate	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR
1e-2	C	**A**	C	C	B	B	**A**	B	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
5e-3	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	B	**A**	**A**
1e-3	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
5e-4	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	B	**A**	**A**
1e-4	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	B	**A**	**A**
5e-5	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
1e-5	B	B	B	B	**A**	B	B	**A**	B	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
5e-6	**A**	**A**	**A**	**A**	**A**	B	B	**A**	B	B	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
1e-6	**A**	**A**	B	**A**	**A**	C	B	B	B	B	**A**	**A**	**A**	**A**	**A**	**A**	B	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
ANOVA p-value	e_6^{-5}	e_2^{-1}	e_1^{-10}	e_1^{-4}	e_8^{-2}	e_2^{-4}	e_6^{-7}	e_1^{-1}	e_2^{-4}	e_1^{-3}	e_1^{0}	e_1^{0}	e_8^{-1}	e_1^{0}	e_1^{0}	e_7^{-1}	e_4^{-1}	e_8^{-1}	e_6^{-1}	e_1^{0}	e_6^{-1}	e_1^{0}	e_4^{-3}	e_6^{-1}	e_1^{0}
LSD (α=0.05)	0.10	0.09	0.08	0.09	0.99	0.13	0.10	0.12	0.11	0.63	0.13	0.12	0.05	0.11	1.03	0.23	0.30	0.27	0.27	1.73	0.17	0.17	0.11	0.15	0.94

We assume that a low learning rate may not be able to address the significant color shift in SA dataset, and a high learning rate may not be able to stabilize the uneven coloring within polygon features in FT and WR datasets.

For simplicity, we set a learning rate of 5e-5 for GLYPH for the remainder of evaluation.

5.4.4.8 Evaluation on GLYPH with Other Experts

For **EQ 8** (*How is the trade-off of integrating the expert models at various tile sizes?*), we combine LOAM with Gemini 3 Flash and SAM2 at various tile sizes and present the results in Table 5.10 and Table 5.11, with some additional results left in Appendix B.2. There is no statistically significant difference in accuracy. However, we notice that the 4096×4096 setup sometimes falls out from the "A" rank in Fisher's LSD test. This degraded R@8 can be attributed to the increased number of polygon features to extract per image tile.

Considering the cost and runtime of Gemini 3 Flash and SAM2 at smaller tile sizes, the results here support the decision to deploy a 1024×1024 tile size for all incorporated models.

5.4.4.9 Case Study

For **EQ9** (*What are the qualitative results of GLYPH against comparative methods?*), we present the case study in Figure 5.9 to Figure 5.13. For each dataset, we demonstrate six cases to provide a qualitative analysis of our design for adaptively leveraging expert models of cross-domain polygon digitization.

FT Dataset

The FT dataset is characterized by mountainous terrain, large contiguous thematic regions, and irregular polygon boundaries with uneven intra-polygon color caused by printing artifacts and scanning degradation. As illustrated in Figure 5.9, the dedicated learning-based method, LOAM, occasionally achieves high precision but suffers from low recall due to uneven color distributions within polygon features. Similarly, SAM2 achieves high precision when the color of a particular polygon map key is significantly different compared to the others from the map

Table 5.10: Evaluation performance on expert tile size across datasets. The indicated tile size only applies to Gemini and SAM2. For each evaluation metric, the best performance is in red. Values are presented in mean±std unless otherwise noted.

Dataset / Metric	FT			SA			SO			SP			WR		
Tile Size	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓	MMPQ ↑	F1@8 ↑	NBDR ↓
4096	0.80	0.88±0.19	**0.36±1.04**	0.58	0.77±0.21	0.17±0.19	0.80	0.89±0.20	0.33±1.01	0.08	0.32±0.19	1.34±1.30	0.44	0.69±0.21	0.19±0.18
2048	0.83	0.88±0.14	0.44±1.28	**0.69**	0.83±0.22	0.15±0.17	0.77	0.89±0.22	**0.33±1.02**	0.26	0.45±0.22	1.05±1.06	0.48	0.75±0.22	0.18±0.20
1024	**0.90**	0.90±0.16	0.48±1.27	0.69	**0.87±0.16**	**0.12±0.16**	**0.81**	**0.91±0.18**	0.36±1.12	**0.30**	**0.59±0.21**	**0.79±0.82**	**0.65**	**0.82±0.13**	**0.15±0.16**
0512	0.86	0.89±0.17	0.44±1.21	0.65	0.83±0.18	0.15±0.17	0.74	0.88±0.18	0.43±1.06	0.22	0.58±0.21	0.87±0.95	0.62	0.81±0.14	0.16±0.16
0256	0.88	**0.91±0.17**	0.52±1.40	0.68	0.84±0.24	0.14±0.18	0.72	0.88±0.21	0.37±0.97	0.16	0.55±0.26	0.89±1.02	0.53	0.79±0.20	0.22±0.24

Table 5.11: Fisher's LSD grouping results across datasets and evaluation metrics on expert tile size. The indicated tile size only applies to Gemini and SAM2. Each cell shows the Fisher-LSD grouping letter by one-way ANOVA and Fisher's LSD test at α=0.05 over map-level quantitative results; "A" denotes the statistically best group. For each metric, the best group overall is in red. The last two rows report the one-way ANOVA p-value and the LSD threshold. We use log(NBDR) to stabilize variance across methods in the NBDR metric, and p-values are shown in a compact scientific form e_m^n to denote $m \times 10^n$. P@8, R@8, and F1@8 refer to precision, recall, and F1 score, respectively, with a tolerance radius of 8 pixels.

Dataset / Metric	FT					SA					SO					SP					WR				
Tile Size	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR	MMPQ	P@8	R@8	F1@8	NBDR
4096	**A**	**A**	B	**A**	**A**	**A**	**A**	B	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	B	B	**A**	**A**	**A**	B	**A**	**A**
2048	**A**	**A**	B	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
1024	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
0512	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
0256	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
ANOVA p-value	e_9^{-1}	e_1^0	e_3^{-2}	e_1^0	e_1^0	e_4^{-1}	e_1^0	e_1^{-1}	e_5^{-1}	e_6^{-1}	e_1^0	e_1^0	e_8^{-1}	e_1^0	e_6^{-1}	e_2^{-1}	e_5^{-1}	e_2^{-1}	e_2^{-1}	e_9^{-1}	e_4^{-1}	e_1^0	e_1^{-1}	e_3^{-1}	e_9^{-1}
LSD (α=0.05)	0.09	0.09	0.05	0.08	0.98	0.13	0.11	0.11	0.11	0.63	0.12	0.11	0.04	0.09	1.02	0.21	0.27	0.29	0.26	1.79	0.15	0.16	0.12	0.13	0.80

legend. However, it may entirely miss low-contrast classes, such as the 1_*poly* for spruce-fir and hemlock in $FT\_Taiwan1956r2$, resulting in extremely low performance. On the other hand, Gemini 3 Flash produces semantically plausible but geometrically fragmented and sometimes over-simplified masks, resulting in mediocre pixel-based performance compared to other methods.

Our proposed GLYPH consistently reconciles large-scale structural coherence with improved recall. An exceptional case is the map of $FT\_Taiwan1956r5$, with the cases of 2_*poly*, 3_*poly*, and 7_*poly* depicted in the Figure. GLYPH corrects under-segmentation or fragmented geometries introduced by LOAM, and it reduces false positives or over-simplified geometries produced by SAM2 and Gemini 3 Flash. In addition, as GLYPH and all three employed expert models consider all polygon map keys, the mutual exclusiveness of polygon features across map keys is preserved, despite not being a hard constraint, enabling the contrastive learning process to affect the final polygon feature outputs. Thus, GLYPH can preserve polygonal instance integrity while mitigating hallucinated fragments, yoking results from distinct models and perspectives to yield balanced precision–recall trade-offs for the polygon features.

SA Dataset

The SA dataset consists of densely structured maps with regular, rectangular building footprints, and heavy textual overlays. As illustrated in Figure 5.10, LOAM often fails in recall due to the strong text interference. SAM2 performs well in geometric delineation but exhibits unstable semantic grounding across legend categories. Gemini 3 Flash captures semantic intent but produces polygon boundary irregularities and occasional merging of adjacent structures. GLYPH substantially improves both color recognition and polygon boundary alignment, especially in dense grids such as *Alameda*_1948, where it suppresses false-positive fragments and restores compact rectangular footprints. The gains are evident in cases where GLYPH combines two or more expert models with low accuracy into a final output above 0.8, leveraging cross-expert agreement and region-level semantic embedding.

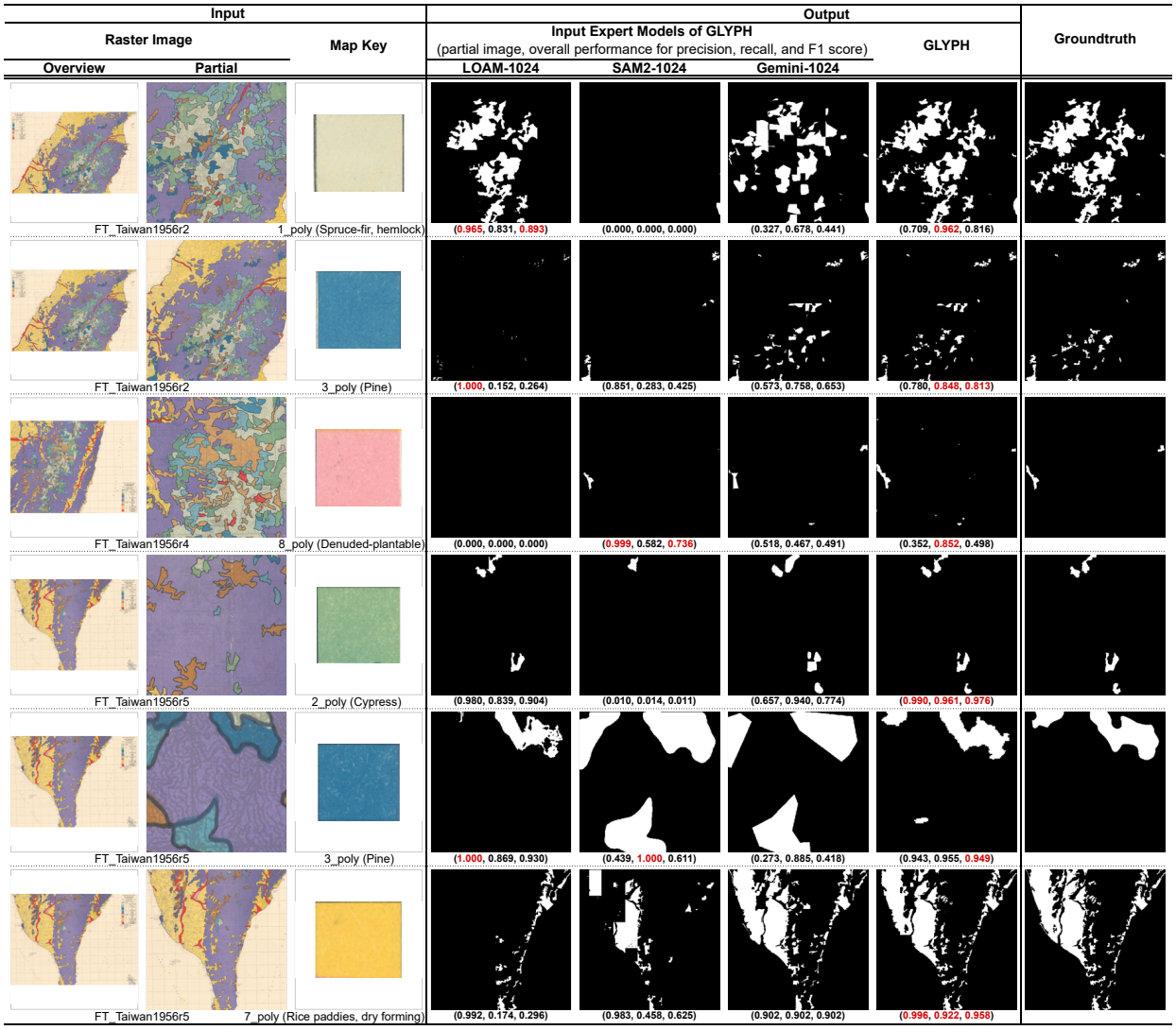


Figure 5.9: Case study for our GLYPH and its input expert models on the FT dataset.

SO Dataset

Although the SO dataset seems to have similar characteristics compared to the in-domain GE dataset, the challenge of this dataset lies in distinguishing adjacent soil types whose chromatic distributions overlap and whose borders are thin and meandering. As illustrated in Figure 5.11, LOAM frequently achieves near-perfect precision in high-contrast map keys but over-extends boundaries or misses narrow regions in a few cases. SAM2 captures boundary continuity well but may oversegment polygon features, leading to a large number of small segments. Gemini 3 Flash exhibits fragmented masks with moderate recall but poor geometric coherence.

GLYPH demonstrates substantial qualitative improvement by recovering boundary continuity while suppressing interior noise. For the 2_*poly* in *SO_Illinois*1905 and 3_*poly* in *SO_Utah*1905, GLYPH preserves contiguous region shapes and achieves a decent precision–recall balance. The fusion mechanism appears particularly effective when at least two experts agree on boundary placement, enabling smooth region-wise consolidation. Overall, GLYPH exhibits strong structural consistency on geological-style maps, where boundary precision and instance completeness are both critical for downstream digitization.

SP Dataset

Among the five datasets, the SP dataset is the most difficult and includes small polygonal features, overlapping graphical elements, and CMYK printing and scanning artifacts. As illustrated in Figure 5.12, LOAM fails to address this dataset and tends to achieve extremely low precision, demonstrating the significant difference between this SP dataset and the GE dataset on which LOAM is trained. In addition, this result shows the limitations of the pixel-based method for CMYK-printed maps with extremely high resolution, leading to less reliable color-set information from polygon map keys. Meanwhile, similar trends can be observed in the previous subsections, where the optimal tile size for this SP dataset often differs from that of the other datasets and can fluctuate within or across methods.

Since there are at most two polygon map keys per map, SAM2 performs relatively well

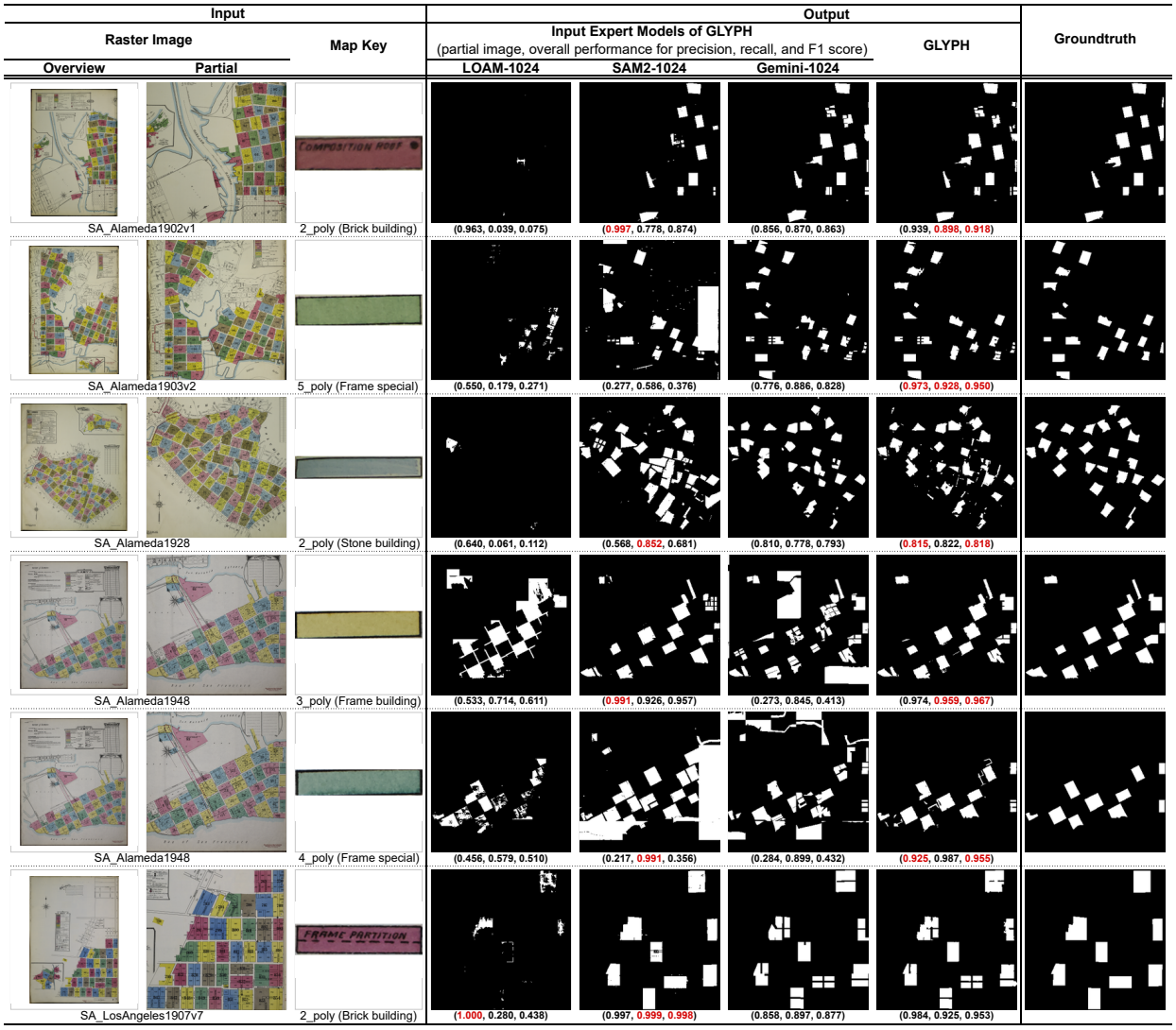


Figure 5.10: Case study for our GLYPH and its input expert models on the SA dataset.

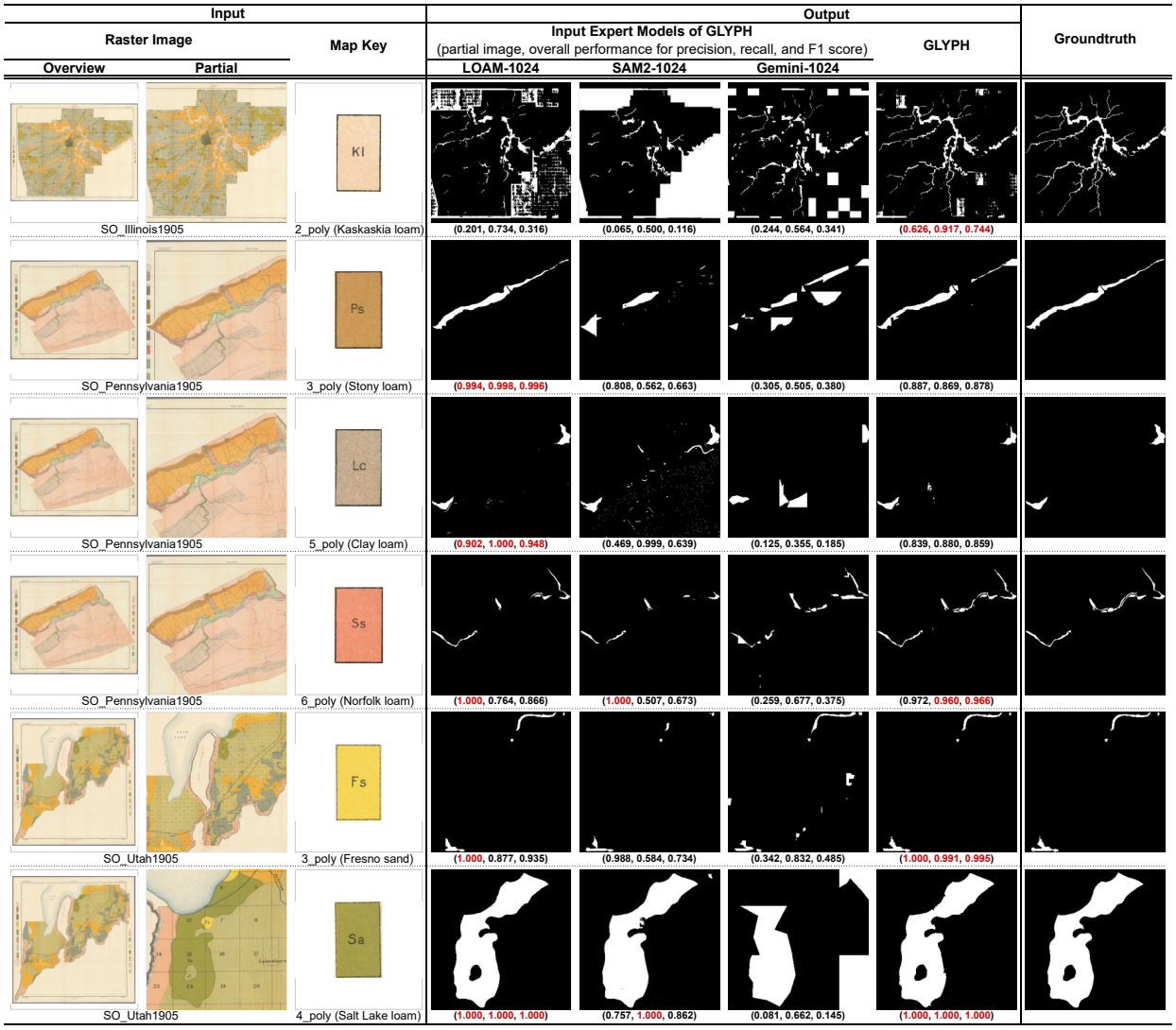


Figure 5.11: Case study for our GLYPH and its input expert models on the SO dataset.

at the task of segmenting polygons. Similarly, although Gemini 3 Flash achieves better accuracy than LOAM, it still produces some fragmented masks with lower precision than SAM2. GLYPH significantly stabilizes extractions across these heterogeneous cases. For $SP\_Tokyo1926v1$, GLYPH combines the boundary sharpness of SAM2 with the semantic discrimination of Gemini 3 Flash, improving F1 to around 0.8 while removing scattered false positives. This is a significant improvement over the results of the dedicated learning-based method for an out-of-domain dataset. In more degraded cases, such as $SP\_Tokyo1945v1$, GLYPH suppresses isolated noise clusters and restores contiguous urban blocks. The qualitative improvements suggest that region-consensus partitioning is beneficial in high-density urban cartography, where instance adjacency and line-art interference are common.

WR Dataset

The WR dataset comprises extensive color overlays, textual annotations, and dashed or partially visible boundaries. Polygons are large but visually cluttered by water channels and annotation marks. As illustrated in Figure 5.13, LOAM maintains structural coherence and achieves the most balanced results among expert models. SAM2 tends to have limited recall with cluttered polygon features, whereas Gemini 3 Flash produces mediocre results.

For maps with unclear polygon map keys such as 2_*poly* in $WR\_Irrigation1930$, GLYPH improves both precision and recall, correcting under-segmentation while removing spurious fragments. However, frequent overlapping polygon features result in mixed colors that correspond to an incorrect polygon map key. Along with the increased emphasis on segmentation, this contributes to an increase in false positives (3_*poly* in $WR\_Irrigation1978$ and 2_*poly* in $WR\_Irrigation1995$) or false negatives (2_*poly* in $WR\_Irrigation1976$ and 5_*poly* in $WR\_Irrigation1996$).

5.4.4.10 In-domain Evaluation of Polygon Generalization

For **EQ10** (*What is the quantitative performance of GLYPH on in-domain dataset?*), we evaluate our proposed cross-domain generalization approach, GLYPH, on the in-domain

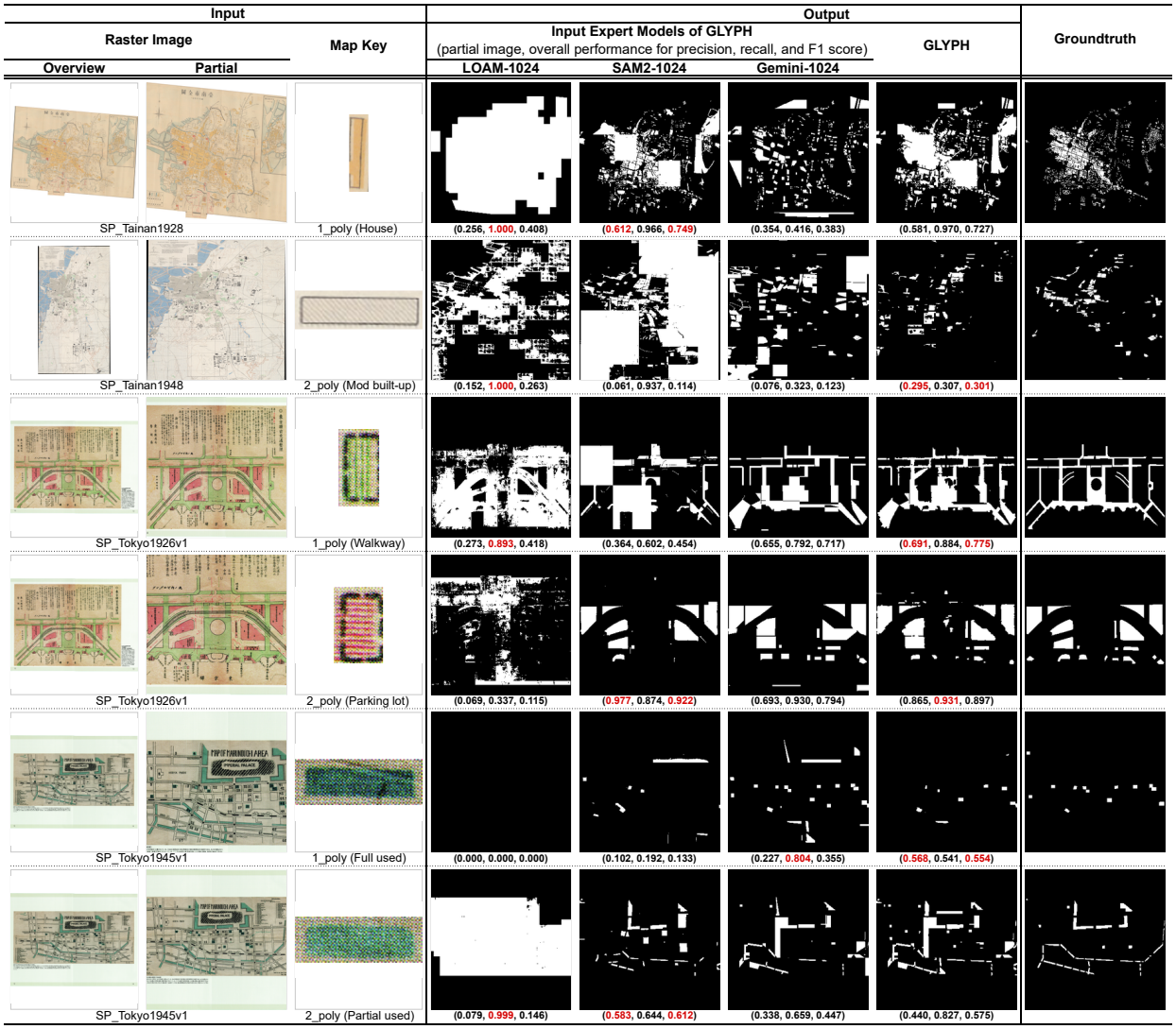


Figure 5.12: Case study for our GLYPH and its input expert models on the SP dataset.

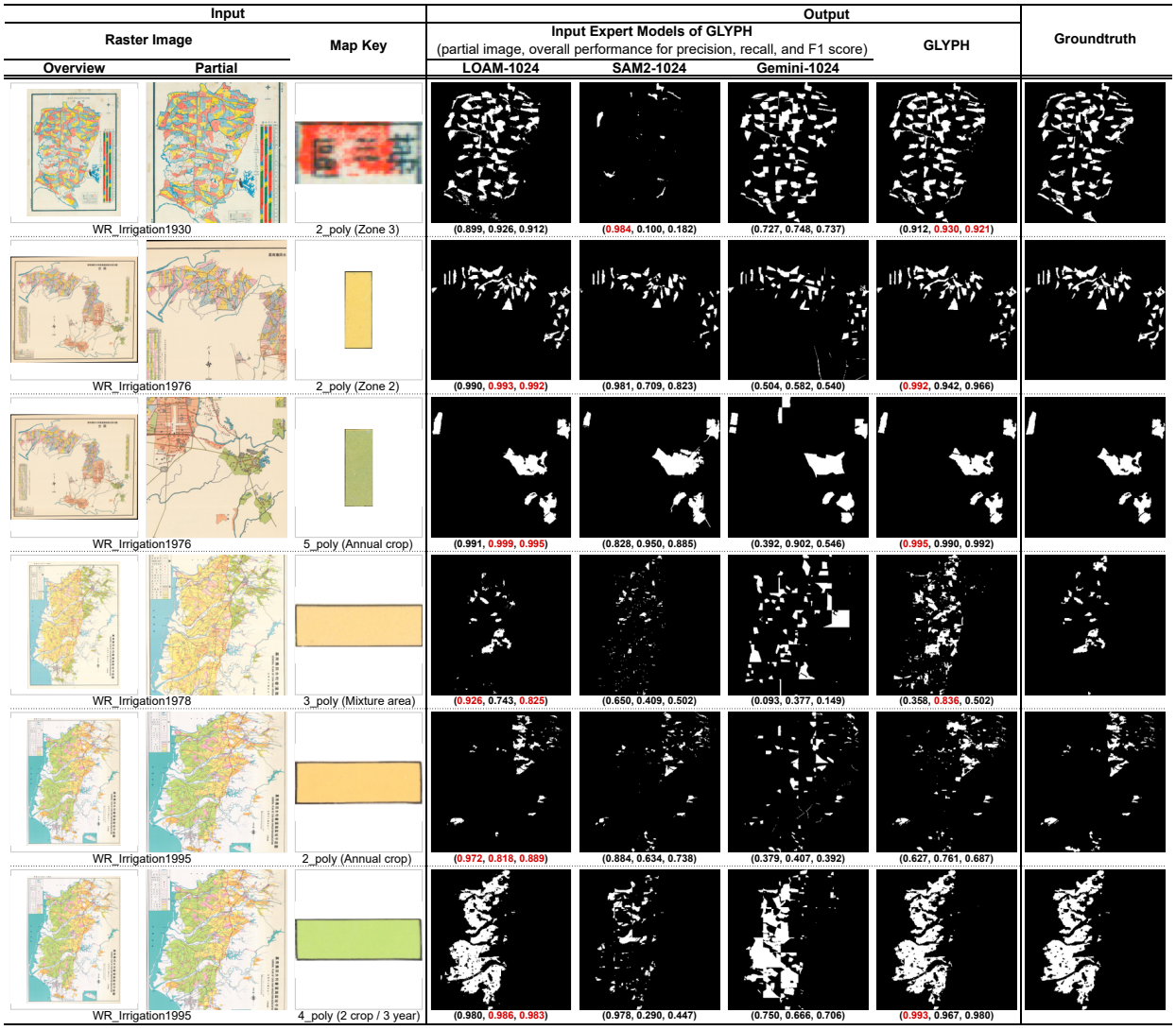


Figure 5.13: Case study for our GLYPH and its input expert models on the WR dataset.

dataset used to train the dedicated polygon digitization LOAM (Chapter 2).

We present the overall performance in MMPQ, F1@8, and NBDR in Table 5.12. Although GLYPH outperforms the dedicated learning-based model LOAM in MMPQ, LOAM achieves the best F1@8 among all methods, including GLYPH. This can be attributed to the increased emphasis on instance-level processing in GLYPH compared to LOAM, a pixel-level inference model introduced in Chapter 2.

Table 5.12: Summary of evaluation performance on the USGS datasets (GE). For each evaluation metric, the best performance within a method family is in bold, and the best performance overall is in red. Values are presented in mean±std unless otherwise noted. "N.A." indicates NBDR is not applicable when a method returns empty results across all cases.

Dataset / Metric	GE		
Method - Tile Size	MMPQ ↑	F1@8 ↑	NBDR ↓
GLYPH-1024 (Ours)	**0.67**	0.67±0.32	15.28±115.11
LOAM-1024	0.61	**0.71±0.37**	**9.65±53.65**
SAM2-4096	0.24	0.31±0.35	38.50±134.32
SAM2-2048	0.32	0.38±0.36	27.48±124.20
SAM2-1024	0.36	0.39±0.36	24.82±115.70
SAM2-0512	**0.39**	0.42±0.36	20.81±108.15
SAM2-0256	0.38	**0.44±0.36**	**18.31±103.33**
SAM3-4096	0.00	0.00±0.01	56.44±88.46
SAM3-2048	0.00	0.01±0.05	24.04±44.60
SAM3-1024	**0.01**	0.01±0.03	19.47±41.89
SAM3-0512	0.01	**0.01±0.03**	**17.76±41.53**
Gemini-3-flash-4096	0.19	0.14±0.20	14.43±44.06
Gemini-3-flash-2048	0.27	0.26±0.23	13.39±66.10
Gemini-3-flash-1024	0.33	0.41±0.23	**5.02±29.93**
Gemini-3-flash-0512	0.32	0.40±0.23	5.52±21.30
Gemini-3-flash-0256	**0.35**	**0.43±0.26**	6.20±18.58
Gemini-3.1-pro-4096	0.03	0.03±0.10	46.23±141.72
Gemini-3.1-pro-2048	0.12	0.11±0.16	15.40±62.39
Gemini-3.1-pro-1024	0.25	0.36±0.23	**4.60±36.94**
Gemini-3.1-pro-0512	0.26	0.38±0.22	9.95±96.01
Gemini-3.1-pro-0256	**0.31**	**0.42±0.24**	6.58±33.54
GPT-4o-4096	0.00	0.00±0.02	83.40±159.62
GPT-4o-2048	0.00	0.01±0.04	47.39±131.50
GPT-4o-1024	0.03	0.04±0.08	21.07±71.43
GPT-4o-0512	0.05	0.07±0.11	18.28±68.52
GPT-4o-0256	**0.15**	**0.18±0.18**	**7.25±30.84**
GPT-5.2-pro	0.00	0.00±0.00	N.A.
Claude-sonnet-4.5	0.00	0.00±0.00	N.A.
Claude-opus-4.6	0.00	0.00±0.00	N.A.

With the increased number of average polygon map keys per map and the greater reliance

on patterns or text to distinguish among polygon features, the accuracy of other expert models decreased significantly for this in-domain dataset with respect to LOAM compared to the five out-of-domain datasets (FT, SA, SO, SP, and WR). Nonetheless, both GLYPH and LOAM statistically significantly outperform all remaining comparative methods, including SAM2, SAM3, the Gemini series, and the GPT series.

We conduct statistical analysis and present Fisher's LSD grouping results in Table 5.13. The ANOVA p-values reported in the final rows of the table are consistently smaller than $5e^{-324}$ and underflows to zero in double precision across metrics. This indicates that there are statistically significant differences between solutions across all methods and metrics.

Despite having a better average NBDR compared to LOAM and GLYPH, all the other expert models and comparative methods fail to achieve either "A" or "B" grouping in Fisher's LSD test. The dedicated learning-based model, LOAM, consistently achieves the best statistical grouping results on all evaluation metrics for this in-domain GE dataset. The cross-domain generalization approach, GLYPH, achieves slightly worse results for this in-domain dataset and consistently achieves the second-best statistical grouping results across all evaluation metrics. Although Gemini 3.1 Pro with a tile size of 1024×1024 achieves the best quantitative NBDR as reported in Table 5.12, it falls to the second tier in Fisher's LSD test as presented in 5.13. This shows that even though Gemini 3.1 Pro under this optimal tile size setting (ranked "B") may achieve results with decent average and a lower standard deviation in NBDR metric, LOAM (ranked "A") and GLYPH (ranked "B") are able to have their solutions achieving superior performance in NBDR for most or more cases with some outliers compared to those of Gemini 3.1 Pro.

Regarding comparative methods, SAM2 shows less fluctuation in accuracy when adjusting its tile sizes than the Gemini series does. It is difficult to determine or estimate the optimal tile size in advance for all these methods. Still, under the optimal settings for each method and metric, their quantitative performance is similar, and there is no statistically significant difference among the solutions of SAM2, Gemini 3 Flash, and Gemini 3.1 Pro. However,

Table 5.13: Fisher's LSD grouping results on the USGS datasets (GE) and across evaluation metrics. Each cell shows the Fisher-LSD grouping letter from one-way ANOVA and Fisher's LSD test at α=0.05 over map-level quantitative results; A denotes the statistically best group. For each metric, the best group within a method family is in bold, and the best group overall is in red. The last two rows report the one-way ANOVA p-value and the LSD threshold. We use $log(NBDR)$ to stabilize variance across methods in the NBDR metric, and p-values are shown in a compact scientific form e^n_m to denote $m \times 10^n$. P@8, R@8, and F1@8 refer to precision, recall, and F1 score, respectively, with a tolerance radius of 8 pixels. "N.A." indicates NBDR is not applicable when a method returns empty results across all cases. The "$\rightarrow 0$" indicates that the p-value is smaller than $5e^{-324}$ and underflows to zero in double precision.

Dataset / Metric	GE				
Method - Tile Size	MMPQ	P@8	R@8	F1@8	NBDR
GLYPH-1024 (Ours)	**B**	**B**	**B**	**B**	**B**
LOAM-1024	**A**	**A**	**A**	**A**	**A**
SAM2-4096	E	E	I	G	I
SAM2-2048	D	D	G	E	G
SAM2-1024	D	D	G	D	G
SAM2-0512	**C**	**C**	F	**C**	**F**
SAM2-0256	**C**	**C**	**E**	**C**	**F**
SAM3-4096	**K**	**M**	P	**N**	M
SAM3-2048	**K**	**M**	P	**N**	L
SAM3-1024	**K**	**M**	**O**	**N**	**K**
SAM3-0512	**K**	**M**	**O**	**N**	**K**
Gemini-3-flash-4096	I	I	L	J	I
Gemini-3-flash-2048	H	G	J	H	E
Gemini-3-flash-1024	F	**D**	E	D	**C**
Gemini-3-flash-0512	F	E	D	D	E
Gemini-3-flash-0256	**E**	E	**C**	**C**	F
Gemini-3.1-pro-4096	K	K	O	M	K
Gemini-3.1-pro-2048	J	I	M	K	H
Gemini-3.1-pro-1024	G	F	H	F	**B**
Gemini-3.1-pro-0512	G	E	G	E	D
Gemini-3.1-pro-0256	**E**	**D**	**D**	**C**	E
GPT-4o-4096	K	M	P	N	N
GPT-4o-2048	K	L	P	N	L
GPT-4o-1024	K	J	O	M	J
GPT-4o-0512	K	H	N	L	I
GPT-4o-0256	**I**	**G**	**K**	**I**	**F**
GPT-5.2-pro	K	N	P	N	N.A.
Claude-sonnet-4.5	K	N	P	N	N.A.
Claude-opus-4.6	K	N	P	N	N.A.
ANOVA p-value	$\rightarrow 0$	$\rightarrow 0$	$\rightarrow 0$	$\rightarrow 0$	$\rightarrow 0$
LSD (α=0.05)	0.02	0.03	0.02	0.02	0.20

SAM2 seems to perform slightly better in MMPQ and P@8, Gemini 3 Flash has better R@8 and F1@8, while Gemini 3.1 Pro has better NBDR. This can be attributed to their characteristics. SAM2 excels at instance segmentation and thus achieves decent instance-level accuracy. With our proposed post-entity-linking process, it is able to obtain better precision compared to VLMs, which may not strictly consider the mutual exclusiveness among polygon map keys. Gemini 3 Flash is currently the state-of-the-art model for visual-language tasks in the Gemini series and achieves higher recall and F1 scores at the pixel level than the remaining comparative methods. On the other hand, while Gemini 3.1 Pro was not particularly designed for handling visual-language tasks, as the most advanced reasoning Gemini model, it is still able to derive simplified polygon geometries from complex USGS geological maps, with better NBDR than other VLMs. While Gemini 3.1 Pro seems to achieve slightly better accuracy in terms of P@8 compared to Gemini 3 Flash, Gemini 3 Flash still surpasses Gemini 3.1 Pro for most metrics under the same tile size.

As reported in Table 5.3 for the out-of-domain datasets, Gemini 3.1 Pro tends to achieve slightly better performance compared to Gemini 2.5 Pro. Based on Table 5.5, we expect Gemini 2.5 Pro to have a similar monetary cost compared to Gemini 3.1 Pro. Therefore, we skip evaluating Gemini 2.5 Pro on this GE dataset, given its potentially high cost and limited accuracy compared to other VLMs in the Gemini series, especially Gemini 3 Flash.

Regarding the remaining comparative methods for Table 5.12 and 5.13, decreasing the tile size generally leads to a better quantitative performance of metrics and grouping tier in Fisher's LSD test. GPT 4o, even at its optimal tile size setting, still fails to surpass SAM2. With a tile size of 256×256, GPT 4o can achieve a similar or slightly better grouping tier to the worst cases of Gemini 3 Flash and Gemini 3.1 Pro. Similar to the results in Table 5.12, while SAM3 can retrieve some polygon feature geometries for maps with extremely few polygon map keys (e.g., 2 or fewer), it consistently falls into the worst or second-worst tier. As with the five out-of-domain datasets, GPT 5.2 Pro, Claude Sonnet 4.5, and Claude Opus 4.6 still return nothing for this GE dataset.

Regarding cost and efficiency, as listed in Table 2.1, the number of polygon map keys per map is significantly higher for this in-domain GE dataset than for the five datasets. This leads to an increase in the token size per API request. We provide the runtime and monetary API cost per map on this GE dataset in Table 5.14. Based on the SAM2-0256 entry (SAM2 with a tile size of 256×256), given the growing runtime relative to the five smaller datasets in Table 5.5, we skip evaluating SAM2 with a tile size of 128×128. Similarly, for the SAM3-0256 entry (SAM3 with a tile size of 256×256), we only process this setup with parts of the maps in the GE dataset due to its long runtime with extremely low accuracy, as shown in Table 5.12. For the GPT-4o-0128 entry (GPT 4o with a tile size of 128×128), since its monetary cost per map is significantly higher than the other methods, we process this setup only on parts of the maps in the GE dataset and report the runtime and API cost here, while skipping its accuracy and corresponding statistical grouping results.

To conclude, while the cross-domain GLYPH achieves slightly worse performance than the dedicated learning-based approach LOAM trained on the in-domain GE dataset, it still outperforms other comparative methods with statistically significant improvements.

5.5 Related Work

Digitizing Historical Maps. Maps are often the only source of information about the Earth surveyed using geodetic techniques [16]. Extracting and digitizing geographic information from historical maps helps experts, whether researchers from other domains or not [3], understand and conduct research on several geography-related topics more easily [40, 55]. However, extracting different types of features, including texts [35, 53], points [66], lines [20, 32, 85], or polygons [66, 81, 86, 84], and georeferencing them [43] from historical maps requires diverse technologies to address distinct technical challenges. Accordingly, DIGMAPPER [22] is a modern historical map digitization system that provides a holistic framework with modular components for georeferencing and feature extraction from the map. Our work follows its

Table 5.14: Average runtime and API cost per map on the USGS dataset (GE). For API-based methods, their runtime is dominated by external service latency or limitations and may not be algorithmically meaningful. N.A. indicates that no API request is required. The best performance within a method family is in bold.

Method - Tile Size	Avg. Time / Map (min.)	Avg. API Cost / Map (USD)
GLYPH-1024	67.36	1.67
LOAM-1024	**38.66**	N.A.
SAM2-4096	6.90	N.A.
SAM2-2048	**5.60**	N.A.
SAM2-1024	8.69	N.A.
SAM2-0512	24.61	N.A.
SAM2-0256	106.98	N.A.
SAM3-4096	**2.65**	N.A.
SAM3-2048	4.69	N.A.
SAM3-1024	11.55	N.A.
SAM3-0512	37.38	N.A.
SAM3-0256*	79.55	N.A.
Gemini-3-flash-4096	**2.88**	**0.15**
Gemini-3-flash-2048	4.50	0.68
Gemini-3-flash-1024	9.19	1.67
Gemini-3-flash-0512	12.51	6.67
Gemini-3-flash-0256	25.11	32.77
Gemini-3.1-pro-4096	**0.56**	**0.74**
Gemini-3.1-pro-2048	0.92	3.07
Gemini-3.1-pro-1024	22.85	12.48
Gemini-3.1-pro-0512	15.85	33.75
Gemini-3.1-pro-0256	44.51	139.95
GPT-4o-4096	**1.36**	**0.16**
GPT-4o-2048	2.44	0.59
GPT-4o-1024	2.74	2.18
GPT-4o-0512	5.04	8.39
GPT-4o-0256	15.62	41.53
GPT-4o-0128*	291.06	544.79

modularized setting to constrain polygon feature extraction.

Polygon Extraction from Raster Maps. For polygon extraction from raster maps, most research focuses on extracting few types of polygon features from maps. For instance, some formulate it as a foreground detection problem and apply a series of image-processing techniques [4]. While the U-Net architecture [64] has demonstrated its efficacy in binary segmentation tasks for medical images [69] and single-feature segmentation from historical maps. Some previous research integrates the U-Net with a transformer to extract water bodies [81], buildings [28], roads [32], hydrological features [82], or archaeological features [24] from historical maps. Similarly, some previous research leverages pre-trained segmentation models such as SAM [36] or SAM2 [63] to extract based on few targeted polygon types, achieving promising accuracy under limited training data [86, 84]. Some frame it as a polygon boundary detection or instance segmentation problem and apply a transformer architecture that leverages common characteristics or constraints [81, 87, 90]. Despite their promising accuracy, the above-mentioned approaches require additional processing to handle arbitrary polygon map keys that are not present in the training dataset.

Some legend-oriented approaches [49, 59] treat map keys as dynamic prompts or references, allowing for the extraction of arbitrary polygon items at inference time. However, these models are frequently trained on specific map series with consistent cartographic conventions and exhibit limited generalization when applied to map styles that differ significantly [50, 51].

Domain Adaptation and Large Visual-language Models.

Some previous research has leveraged common semantic cues across historical maps to propagate knowledge and to support better segmentation or feature extraction with minimal human intervention [83, 90]. While some focus on generating synthetic training data to reduce the reliance on manual annotations and increase the coverage of map styles [5, 42, 58].

Beyond map-specific literature, combinations of test-time adaptation (TTA) and mixture-of-experts (MoE) mechanisms offer strategies for identifying and adapting domain shift with pseudo-label guidance [38, 76]. On the other hand, with rapid development in recent years,

pre-trained large vision-language models have shown promising results on reasoning geospatial information, whether from historical maps [61, 88] or not [30, 44].

Altogether, these approaches inspire parts of our design, in which GLYPH applies a combined TTA with MoE to leverage expert models' polygon-extraction solutions and adapt to various map styles.

5.6 Summary

This chapter addresses the research problem of cross-domain polygon extraction from historical maps under significant domain shift and a lack of target-domain annotations. We leverage the map legends to guide a test-time adaptive mixture-of-experts framework, GLYPH, which reconciles the semantic and geometric strengths of dedicated models, segmentation foundation models, and large vision-language models. Our comprehensive evaluation across five diverse historical map datasets demonstrates the efficacy of our approach, outperforming state-of-the-art methods on both structural correctness and boundary alignment. The results show that integrating specialized domain models with general-purpose vision models through adaptive fusion can overcome the limitations of individual experts in unseen domains. In addition, by facilitating large-scale vectorization of historical archives, this enables the investigation of long-term environmental and urban systems, supporting downstream analyses related to historical geographic information science.

Chapter 6

Conclusion and Future Direction

6.1 Conclusion

This dissertation addresses the challenge of converting the polygonal features in massive archives of historical thematic raster maps into structured, analysis-ready vector data. While these maps contain invaluable information for critical mineral assessment and long-term environmental studies, their interpretation is often hindered by numerous unseen and varied polygon map keys, significant color inconsistencies, the presence of uncolored field drafts, and significantly diverse cartographic styles. We define polygon metadata and present a comprehensive metadata-driven machine-learning framework that leverages polygon metadata, such as map keys, textual labels, and polygon boundaries in raster format, to automate the digitization, recoloring, and colorization of polygonal features in historical maps.

The proposed frameworks address distinct stages and needs for map interpretation. For uncolored draft maps, SHADING learns to transform achromatic sketches into colorized maps with respect to geological conventions. To address color inconsistencies in existing map archives, REPOLISH learns to correct and align map content with legend definitions. Then, LOAM encodes polygon metadata into intermediate bitmaps for precise feature extraction of in-domain map collections. Finally, to enhance the efficacy across diverse historical map collections, GLYPH establishes a generalized mixture-of-experts framework that adapts to

unseen map styles at inference time without requiring target-domain annotations. Together, these frameworks provide a robust pathway for preserving and utilizing historical geographic information at a large scale.

6.2 Contribution of the Research

This dissertation presents a unified framework for automated polygon digitization and interpretation from historical thematic maps under both in-domain and cross-domain settings. First, we develop a metadata-driven approach that generates multiple intermediate representations, capturing color, text, and boundaries, to learn to extract polygon features from raster images for in-domain datasets. Second, we propose a map-recoloring method to automatically identify and correct inconsistencies in color assignments between polygonal features in the map content area and the map legend for in-domain datasets, thereby improving the reliability of downstream tasks. Third, we introduce a semantic-restoration approach to automatically colorize achromatic draft maps using a conditional generative framework that integrates sketch and semantic reasoning over the referenced visual appearances for in-domain datasets. Finally, we present a legend-guided, test-time adaptive mixture-of-experts framework that learns to generalize polygon digitization across diverse historical map collections without requiring target-domain annotations. Together, these contributions advance the automation, robustness, and generalizability of large-scale polygon digitization from historical maps.

6.3 Future Direction

Building upon the frameworks presented in this dissertation, several directions remain for future research. We present the future directions as follows.

For polygon extraction, one direction is to refine the incorporation of geometric constraints for the polygonal features, whether in historical maps or not. Precisely, one may explore more accurate boundary extraction and dynamically incorporate various polygonal geometric

priors. This would ensure that the outputs are applicable to distinct downstream tasks and reduce the need for post-editing by the domain experts. One interesting direction is to further involve domain experts, annotators, or downstream users in the machine-learning process [57, 60]. This includes quantitative, qualitative, and narrative feedback on the preliminary and final outputs, which would help improve understanding of the process and needs of polygon digitization from historical maps, as well as the divergence across various domains. By learning from feedback through active learning or other adaptive enhancements, the machine might be able to better understand and digitize polygon features from maps under different circumstances [41, 46, 61]. Another interesting direction is to further refine the text-label association and prevent pattern matching from producing false positives [53, 54]. By better addressing misaligned text labels or patterns caused by limited polygonal space or printing artifacts, this can improve the completeness of polygonal features and significantly reduce manual post-editing effort. The other direction is to extend to multi-feature extraction [22, 66]. As some previous research has formulated line extraction as a polygon extraction problem [84], we may extend the metadata-driven recognition models to other geographic representations, specifically line and point features. This will require modifications to the metadata preprocessing stage to capture distinct structural and semantic information relevant to roads, contours, etc. However, our approach may serve as a good baseline component or comparator for feature extraction of unseen feature styles, regardless of whether the features are polygons, lines, or points.

For map recoloring, one direction is to enhance the reasoning of complex symbols. While current region representations are primarily color-based, cartographic symbols such as intricate hatching patterns and overlapping translucent textures may be crucial for distinguishing distinct polygon map keys in some historical archives [66]. Therefore, expanding the recoloring framework to handle more complex cartographic symbolization or out-of-domain datasets might also help improve its generalizability. Another potential direction might be to further exploit contextual information, such as the shapes of polygonal features and their neighboring

features [67]. On the other hand, since the recoloring model is domain-dependent, another direction is to conditionally generate synthetic data to support its training [42, 84].

For map colorization, the main limitation lies in its domain-dependent nature and the reliance on textual patterns in polygon map keys. To overcome the domain-dependent limitation, one direction is to generate synthetic training data to simulate a broader range of historical map styles and degradation artifacts [5, 42], which would enhance model robustness. To overcome the reliance on textual patterns, one may further exploit the geographic priors of the polygonal features [10]. Despite the need for additional training data and limited accuracy in semantic color coding, one can still achieve decent results in colorization. Another interesting direction is to exploit combinatorial optimization [48, 91] or distance-based learning techniques [73, 74] for the semantic color coding and polygonal entity linking in colorization. Based on most color-coding schemas across various historical map collections, a conditioned hierarchical optimization approach might be suitable with some reformulations [47, 96].

For generalization across map styles, despite already achieving better performance than all incorporated expert models at this stage in most cases, one direction is to further investigate the ability and limitations of different expert models at test time. With sufficient understanding of the incorporated models, this may help achieve better, or at least decent, results, even if all expert models fail drastically under particular cartographic styles. In addition, we aim to extend the framework to handle more complex cartographic symbolization, which remains challenging for current color-based region representations. Besides, we plan to explore the geometry priors and constraints of polygon features to further reduce the manual post-editing required to transform raster archives into structured, linked geospatial data.

Bibliography

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods". In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2274–2282.

[2] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[3] Claudio Affolter, Sidi Wu, Yizi Chen, and Lorenz Hurni. "Generative AI in map-making: A technical exploration and its implications for cartographers". In: *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*. 2025, pp. 884–893.

[4] Mauricio Giraldo Arteaga. "Historical map polygon and feature extractor". In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction*. 2013, pp. 66–71.

[5] Lukas Arzoumanidis, Julius Knechtel, Jan-Henrik Haunert, and Youness Dehbi. "Automatic Uncertainty-Aware Synthetic Data Bootstrapping for Historical Map Segmentation". In: *arXiv preprint arXiv:2511.15875* (2025).

[6] John Canny. "A computational approach to edge detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (2009), pp. 679–698.

[7] Nicolas Carion et al. "Sam 3: Segment anything with concepts". In: *arXiv preprint arXiv:2511.16719* (2025).

[8] Hernan Carrillo, Michaël Clément, Aurélie Bugeau, and Edgar Simo-Serra. "Diffusart: Enhancing line art colorization with conditional diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3486–3490.

[9] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. "Palette-based photo recoloring." In: *ACM Trans. Graph.* 34.4 (2015), pp. 139–1.

[10] Lin Che, Yizi Chen, Tanhua Jin, Martin Raubal, Konrad Schindler, and Peter Kiefer. "Unsupervised Urban Land Use Mapping with Street View Contrastive Clustering and a Geographical Prior". In: *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*. 2025, pp. 28–38.

[11] Gang Chen, Guipeng Zhang, Zhenguo Yang, and Wenyin Liu. "Multi-scale patch-GAN with edge detection for image inpainting". In: *Applied intelligence* 53.4 (2023), pp. 3917–3932.

[12] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. "Transunet: Transformers make strong encoders for medical image segmentation". In: *arXiv preprint arXiv:2102.04306* (2021).

[13] Weiye Chen, Zhihao Wang, Zhili Li, Yiqun Xie, Xiaowei Jia, and Anlin Li. "Deep semantic segmentation for building detection using knowledge-informed features from LiDAR point clouds". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 2022, pp. 1–4.

[14] Yao-Yi Chiang and Craig A Knoblock. "A general approach for extracting road vector data from raster maps". In: *International Journal on Document Analysis and Recognition (IJDAR)* 16 (2013), pp. 55–81.

[15] Yao-Yi Chiang, Craig A Knoblock, Cyrus Shahabi, and Ching-Chien Chen. "Automatic and accurate extraction of road intersections from raster maps". In: *GeoInformatica* 13 (2009), pp. 121–157.

[16] Yao-Yi Chiang, Stefan Leyk, and Craig A Knoblock. "A survey of digital map processing techniques". In: *ACM Computing Surveys (CSUR)* 47.1 (2014), pp. 1–44.

[17] Yuanzheng Ci, Xinzhu Ma, Zhihui Wang, Haojie Li, and Zhongxuan Luo. "User-guided deep anime line art colorization with conditional adversarial networks". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 1536–1544.

[18] Dorin Comaniciu and Peter Meer. "Mean shift: A robust approach toward feature space analysis". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002), pp. 603–619.

[19] Commission Internationale de l'Eclairage (CIE). *CIE 1976 L*a*b* Colour Space*. International Commission on Illumination, 1976.

[20] Weiwei Duan, Yao-Yi Chiang, and Craig A Knoblock. "LDTR: Linear Object Detection Transformer for Accurate Graph Generation by Learning the N-Hop Connectivity Information". In: *International Conference on Document Analysis and Recognition*. Springer. 2025, pp. 40–59.

[21] Weiwei Duan, Yao-Yi Chiang, Craig A Knoblock, Vinil Jain, Dan Feldman, Johannes H Uhl, and Stefan Leyk. "Automatic alignment of geographic features in contemporary vector data and historical maps". In: *Proceedings of the 1st workshop on artificial intelligence and deep learning for geographic knowledge discovery*. 2017, pp. 45–54.

[22] Weiwei Duan et al. “DIGMAPPER: A Modular System for Automated Geologic Map Digitization”. In: *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*. 2025, pp. 717–728.

[23] P Ganesan, V Rajini, and R Immanuvel Rajkumar. “Segmentation and edge detection of color images using CIELAB color space and edge detectors”. In: *INTERACT-2010* (2010), pp. 393–397.

[24] Arnau Garcia-Molsosa, Hector A Orengo, Dan Lawrence, Graham Philip, Kristen Hopper, and Cameron A Petrie. “Potential of deep learning segmentation for the extraction of archaeological features from historical map series”. In: *Archaeological Prospection* 28.2 (2021), pp. 187–199.

[25] M.A. Goldman, J.M. Rosera, G.W. Lederer, G.E. Graham, A. Mishra, and A. Yepremyan. *Training and validation data from the AI for Critical Mineral Assessment Competition*. 2023. DOI: 10.5066/P9FXSPT1.

[26] Jiaze He, Jian Xiao, Yuanjie Cao, Jing He, Siyu Li, Jin Huang, Ruhan He, and Jianlin Zhu. “Region-assisted line drawing colorization through diffusion model”. In: *The Visual Computer* 41.8 (2025), pp. 5769–5780.

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[28] Magnus Heitzler and Lorenz Hurni. “Cartographic reconstruction of building footprints from historical maps: A study on the Swiss Siegfried map”. In: *Transactions in GIS* 24.2 (2020), pp. 442–461.

[29] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[30] Krzysztof Janowicz, Zilong Liu, Gengchen Mai, Zhangyu Wang, Ivan Majic, Alexandra Fortacz, Grant McKenzie, and Song Gao. “Whose Truth? Pluralistic Geo-Alignment for (Agentic) AI”. In: *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*. 2025, pp. 799–803.

[31] Wei Ji, Jingjing Li, Qi Bi, Wenbo Li, and Li Cheng. “Segment anything is not always perfect: An investigation of sam on different real-world applications”. In: *arXiv preprint arXiv:2304.05750* (2023).

[32] Chenjing Jiao, Magnus Heitzler, and Lorenz Hurni. “A fast and effective deep learning approach for road extraction from historical maps by automatically generating

training data with symbol reconstruction". In: *International Journal of Applied Earth Observation and Geoinformation* 113 (2022), p. 102980.

[33] Alchan Kim. *FastSLIC: Optimized SLIC Superpixel*. Preprint, available on GitHub. Accessed: 2026-01-26. 2019. URL: https://github.com/Algy/fast-slic.

[34] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. "Tag2pix: Line art colorization using text tag with secat and changing loss". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9056–9065.

[35] Jina Kim, Zekun Li, Yijun Lin, Min Namgung, Leeje Jang, and Yao-Yi Chiang. "The mapKurator system: a complete pipeline for extracting and linking text from historical maps". In: *arXiv preprint arXiv:2306.17059* (2023).

[36] Alexander Kirillov et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023).

[37] Dae Geon Lee, Young Ha Shin, and Dong-Cheon Lee. "Land cover classification using SegNet with slope, aspect, and multidirectional shaded relief images derived from digital surface model". In: *Journal of Sensors* 2020 (2020), pp. 1–21.

[38] Tianwu Lei, Silin Chen, Bohan Wang, Zhengkai Jiang, and Ningmu Zou. "Adapted-moe: Mixture of experts with test-time adaption for anomaly detection". In: *International Conference on Intelligent Computing*. Springer. 2025, pp. 427–441.

[39] Zekun Li. "Generating historical maps from online maps". In: *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*. 2019, pp. 610–611.

[40] Zekun Li, Yao-Yi Chiang, Sasan Tavakkol, Basel Shbita, Johannes H Uhl, Stefan Leyk, and Craig A Knoblock. "An automatic approach for generating rich, linked geo-metadata from historical map images". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3290–3298.

[41] Zekun Li, Malcolm Grossman, Mihir Kulkarni, Muhao Chen, Yao-Yi Chiang, et al. "Mapqa: Open-domain geospatial question answering on map data". In: *arXiv preprint arXiv:2503.07871* (2025).

[42] Zekun Li, Runyu Guan, Qianmu Yu, Yao-Yi Chiang, and Craig A Knoblock. "Synthetic map generation to provide unlimited training data for historical map text detection". In: *Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*. 2021, pp. 17–26.

[43] Zekun Li, Fandel Lin, Yijun Lin, Yao-Yi Chiang, and Craig A Knoblock. "An Ensemble Approach to Text-Based Georeferencing of Historical Maps". Under review. 2026.

[44] Zekun Li, Wenxuan Zhou, Yao-Yi Chiang, and Muhao Chen. "Geolm: Empowering language models for geospatially grounded language understanding". In: *arXiv preprint arXiv:2310.14478* (2023).

[45] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. "Topological map extraction from overhead images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1715–1724.

[46] Fandel Lin, Ding-Ying Guo, and Jer-Yann Lin. "A Machine-Learning Approach to Recognizing Teaching Beliefs in Narrative Stories of Outstanding Professors". In: *International Conference on Artificial Intelligence in Education*. Springer. 2023, pp. 739–745.

[47] Fandel Lin and Hsun-Ping Hsieh. "Conntrans: a two-stage concentric annealing approach for multi-criteria distributed competitive stationary resource searching". In: *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 2021, pp. 163–174.

[48] Fandel Lin and Craig A Knoblock. "Indirect cooperation in distributed stationary-resource searching with predefined destinations". In: *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*. 2023, pp. 1–12.

[49] Fandel Lin, Craig A Knoblock, Basel Shbita, Binh Vu, Zekun Li, and Yao-Yi Chiang. "Exploiting Polygon Metadata to Understand Raster Maps-Accurate Polygonal Feature Extraction". In: *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*. 2023, pp. 1–12.

[50] Fandel Lin, Craig A Knoblock, Binh Vu, and Yao-Yi Chiang. "Exploiting Polygon Metadata to Recolor Historical Maps". In: *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*. 2025, pp. 1122–1125.

[51] Fandel Lin, Craig A Knoblock, Binh Vu, Basel Shbita, and Yao-Yi Chiang. "Exploiting Polygon Metadata to Colorize Draft Maps". In: *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*. 2025, pp. 1126–1129.

[52] Fandel Lin, Zekun Li, Yao-Yi Chiang, and Craig A Knoblock. "Cross-domain Polygon Extraction from Historical Maps via Legend-guided Semantic Fusion". Under review. 2026.

[53] Yijun Lin and Yao-Yi Chiang. "Hyper-local deformable transformers for text spotting on historical maps". In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024, pp. 5387–5397.

[54] Yijun Lin et al. “ICDAR 2025 competition on historical map text detection, recognition, and linking”. In: *International Conference on Document Analysis and Recognition*. Springer. 2025, pp. 568–585.

[55] Tiange Liu, Pengfei Xu, and Shihui Zhang. “A review of recent advances in scanned topographic map processing”. In: *Neurocomputing* 328 (2019), pp. 75–87.

[56] Xueting Liu, Wenliang Wu, Chengze Li, Yifan Li, and Huisi Wu. “Reference-guided structure-aware deep sketch colorization for cartoons”. In: *Computational Visual Media* 8 (2022), pp. 135–148.

[57] Ziyi Liu, Claudio Affolter, Sidi Wu, Yizi Chen, and Lorenz Hurni. “An Efficient System for Automatic Map Storytelling: A Case Study on Historical Maps”. In: *AGILE: GIScience Series* 6 (2025), p. 5.

[58] Marta López-Rauhut, Hongyu Zhou, Mathieu Aubry, and Loic Landrieu. “Segmenting France Across Four Centuries”. In: *International Conference on Document Analysis and Recognition*. Springer. 2025, pp. 3–22.

[59] Shirui Luo, Aaron Saxton, Albert Bode, Priyam Mazumdar, and Volodymyr Kindratenko. “Critical Minerals Map Feature Extraction Using Deep Learning”. In: *IEEE Geoscience and Remote Sensing Letters* 20 (2023), pp. 1–5. DOI: 10.1109/LGRS.2023.3310915.

[60] Dino Pedreschi et al. “Human-AI coevolution”. In: *Artificial Intelligence* 339 (2025), p. 104244.

[61] Jiyoon Pyo et al. “FRIEDA: Benchmarking Multi-Step Cartographic Reasoning in Vision-Language Models”. In: *arXiv preprint arXiv:2512.08016* (2025).

[62] Qianru Qiu, Xueting Wang, and Mayu Otani. “Multimodal color recommendation in vector graphic documents”. In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 4003–4011.

[63] Nikhila Ravi et al. “Sam 2: Segment anything in images and videos”. In: *arXiv preprint arXiv:2408.00714* (2024).

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[65] Mahmoud Saeedimoghaddam and Tomasz F Stepinski. “Automatic extraction of road intersection points from USGS historical map series using deep convolutional neural

networks". In: *International Journal of Geographical Information Science* 34.5 (2020), pp. 947–968.

[66] Aaron Saxton et al. "Accurate Feature Extraction from Historical Geologic Maps Using Open-Set Segmentation and Detection". In: *Geosciences* 14.11 (2024), p. 305.

[67] Basel Shbita, Binh Vu, Fandel Lin, and Craig A Knoblock. "Embedding spatial and semantic contexts for geo-entity typing in smart city applications". In: *Companion Proceedings of the ACM on Web Conference 2025*. 2025, pp. 1724–1732.

[68] Min Shi, Jia-Qi Zhang, Shu-Yu Chen, Lin Gao, Yu-Kun Lai, and Fang-Lue Zhang. "Deep line art video colorization with a few references". In: *arXiv preprint arXiv:2003.10685* (2020).

[69] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. "U-net and its variants for medical image segmentation: A review of theory and applications". In: *IEEE access* 9 (2021), pp. 82031–82057.

[70] Hunsoo Song and Jinha Jung. "Challenges in building extraction from airborne LiDAR data: ground-truth, building boundaries, and evaluation metrics". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 2022, pp. 1–4.

[71] Geologic Data Subcommittee. *Fgdc digital cartographic standard for geologic map symbolization*. Tech. rep. Citeseer, 2006.

[72] Gemini Team et al. "Gemini: a family of highly capable multimodal models". In: *arXiv preprint arXiv:2312.11805* (2023).

[73] Binh Vu, Craig A Knoblock, and Fandel Lin. "A domain-independent approach for semantic table interpretation". In: *International Semantic Web Conference*. Springer. 2025, pp. 235–252.

[74] Binh Vu, Craig A Knoblock, Basel Shbita, and Fandel Lin. "Exploiting distant supervision to learn semantic descriptions of tables with overlapping data". In: *International Semantic Web Conference*. Springer. 2024, pp. 116–134.

[75] Ronald R Wahl. "The use of topology on geologic maps". In: *Digital Mapping Techniques '04—Workshop Proceedings*. Vol. 2004. 2004, p. 159.

[76] Hancong Wang, Yue Yu, Hairong Zheng, and Tong Zhang. "Test-Time Adaptation of Medical Vision-Language Models with Mixture of Modality Experts". In: *Proceedings of the 33rd ACM International Conference on Multimedia*. 2025, pp. 4649–4658.

[77] Linhan Wang, Shuo Lei, Jianfeng He, Shengkun Wang, Min Zhang, and Chang-Tien Lu. "Self-Correlation and Cross-Correlation Learning for Few-Shot Remote Sensing Image Semantic Segmentation". In: *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*. 2023, pp. 1–10.

[78] Ning Wang, Muyao Niu, Zhihui Wang, Kun Hu, Bin Liu, Zhiyong Wang, and Haojie Li. "Region assisted sketch colorization". In: *IEEE Transactions on Image Processing* 32 (2023), pp. 6142–6154.

[79] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. "Freesolo: Learning to segment objects without annotations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14176–14186.

[80] Yi Wang, Menghan Xia, Lu Qi, Jing Shao, and Yu Qiao. "PalGAN: Image colorization with palette generative adversarial networks". In: *European Conference on Computer Vision*. Springer. 2022, pp. 271–288.

[81] Sidi Wu, Yizi Chen, Konrad Schindler, and Lorenz Hurni. "Cross-attention Spatio-temporal Context Transformer for Semantic Segmentation of Historical Maps". In: *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*. 2023, pp. 1–9.

[82] Sidi Wu, Magnus Heitzler, and Lorenz Hurni. "Leveraging uncertainty estimation and spatial pyramid pooling for extracting hydrological features from scanned historical topographic maps". In: *GIScience & Remote Sensing* 59.1 (2022), pp. 200–214.

[83] Sidi Wu, Konrad Schindler, Magnus Heitzler, and Lorenz Hurni. "Domain adaptation in segmenting historical maps: A weakly supervised approach through spatial co-occurrence". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 197 (2023), pp. 199–211.

[84] Xue Xia, Randall Balestriero, Tao Zhang, Yixin Zhou, Andrew Ding, Dev Saini, and Lorenz Hurni. "MapSAM2: Adapting SAM2 for Automatic Segmentation of Historical Map Images and Time Series". In: *arXiv preprint arXiv:2510.27547* (2025).

[85] Xue Xia, Chenjing Jiao, and Lorenz Hurni. "Contrastive Pretraining for Railway Detection: Unveiling Historical Maps with Transformers". In: *Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*. 2023, pp. 30–33.

[86] Xue Xia, Daiwei Zhang, Wenxuan Song, Wei Huang, and Lorenz Hurni. "MapSAM: adapting segment anything model for automated feature detection in historical maps". In: *GIScience & Remote Sensing* 62.1 (2025), p. 2494883.

[87] Xue Xia, Tao Zhang, Magnus Heitzler, and Lorenz Hurni. "Vectorizing historical maps with topological consistency: A hybrid approach using transformers and contour-based instance segmentation". In: *International Journal of Applied Earth Observation and Geoinformation* 129 (2024), p. 103837.

[88] Shuo Xing, Zezhou Sun, Shuangyu Xie, Kaiyuan Chen, Yanjia Huang, Yuping Wang, Jiachen Li, Dezhen Song, and Zhengzhong Tu. "Can Large Vision Language Models Read Maps Like a Human?" In: *arXiv preprint arXiv:2503.14607* (2025).

[89] Xin Xu. "An unsupervised building footprints delineation approach for large-scale LiDAR point clouds". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 2022, pp. 1–4.

[90] Yunshuang Yuan, Frank Thiemann, Thorsten Dahms, and Monika Sester. "Semantic segmentation of time-series of historical maps by learning from only one map". In: *International Journal of Cartography* (2025), pp. 1–15.

[91] Han Zhang, Oren Salzman, TK Satish Kumar, Ariel Felner, Carlos Hernández Ulloa, and Sven Koenig. "A* pex: Efficient approximate multi-objective search on graphs". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 32. 2022, pp. 394–403.

[92] Qing Zhang, Chunxia Xiao, Hanqiu Sun, and Feng Tang. "Palette-based image recoloring using color decomposition optimization". In: *IEEE Transactions on Image Processing* 26.4 (2017), pp. 1952–1964.

[93] Qing-Long Zhang and Yu-Bin Yang. "Sa-net: Shuffle attention for deep convolutional neural networks". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 2235–2239.

[94] Nanxuan Zhao, Quanlong Zheng, Jing Liao, Ying Cao, Hanspeter Pfister, and Rynson WH Lau. "Selective region-based photo color adjustment for graphic designs". In: *ACM Transactions on Graphics (TOG)* 40.2 (2021), pp. 1–16.

[95] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. "Polyworld: Polygonal building extraction with graph neural networks in satellite images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1848–1857.

[96] Paweł Zyblewski and Szymon Wojciechowski. "How to RETIRE Tabular Data in Favor of Discrete Digital Signal Representation". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2025, pp. 119–135.

Appendix A

Segmenting Content Area and Map Keys

A.1 Automated Map Segmentation

The main digitization modules (LOAM and GLYPH) and the supplementary modules (REPOLISH and SHADING) both take the raster map with identified polygon map keys and the map content area as input. Humans can label the bounding boxes of the polygon map keys and the map content area with a few clicks, which is accurate enough and relatively easier than annotating polygonal features in the map content. Still, we provide an automated approach to segment the map content area and polygon map keys from the raster map, turning the entire polygon digitization pipeline into a fully automated process with zero human input, aside from providing the raster map.

Some of the following content for this automated map segmentation module is presented in MapLocator [43] to support map georeferencing.

A.2 Approach to Map Segmentation

The map segmentation module aims to isolate the map content area from frames, margins, legends, and background regions that commonly appear in historical maps. First, we employ the Segment Anything Model (SAM) [63] to extract high-level, visually coherent areas. To refine these regions, we use color gradients and Canny edge detection [6] to capture fine-

grained cartographic structures, such as map borders, graticules, and coastline-like outlines. Morphological consolidation and connected-component analysis are then used to consolidate a single dominant map region. To enforce local spatial regularity, we apply fast SLIC [33] to derive superpixels that preserve color and texture consistency within each segment. Followed by a second SAM pass to compensate for incomplete or broken boundaries caused by scanning or storage artifacts.

This module outputs a bounding polygon along with its associated bounding box. While most map content areas are rectangular, integrating SAM enables the recovery of irregular boundaries, providing geometric priors for downstream tasks.

A.3 Dataset

For the polygon map keys, we use the USGS geologic map benchmark as introduced in Chapter 2. We treat the labeled bounding box of the polygon map keys as the ground truth.

Since the USGS geologic map benchmark lacks an explicit ground truth label for the map content area, we use a series of sampled topographic maps (Historical Topographic Map Collection, or **HTMC**). Proposed in MapLocator [43], the dataset comprises 505 maps and is selected using a stratification strategy to ensure balanced map scale, publication era, and geographic coverage.

For this map segmentation task, no map is used for training, and all datasets are treated as out-of-domain.

A.4 Evaluation

For map-key segmentation, our approach achieves an instance-based F1 score of 0.71 with an area thresholding ratio of 0.50. Despite its lower accuracy compared to manual labeling, the results can serve as preliminary labels and reduce the time spent identifying all polygon map keys from the raster map.

Table A.1: Map segmentation performance on the HTMC dataset.

Method	IoU ↑	Precision ↑	Recall ↑	F1 Score ↑	Complet. ↑
Ours	0.98±0.04	0.99±0.03	1.00±0.02	0.99±0.02	100.0%
Gemini 3 Flash	0.95±0.12	0.99±0.03	0.96±0.12	0.97±0.08	95.4%
GPT 4o	0.43±0.15	0.86±0.11	0.47±0.17	0.58±0.16	5.5%

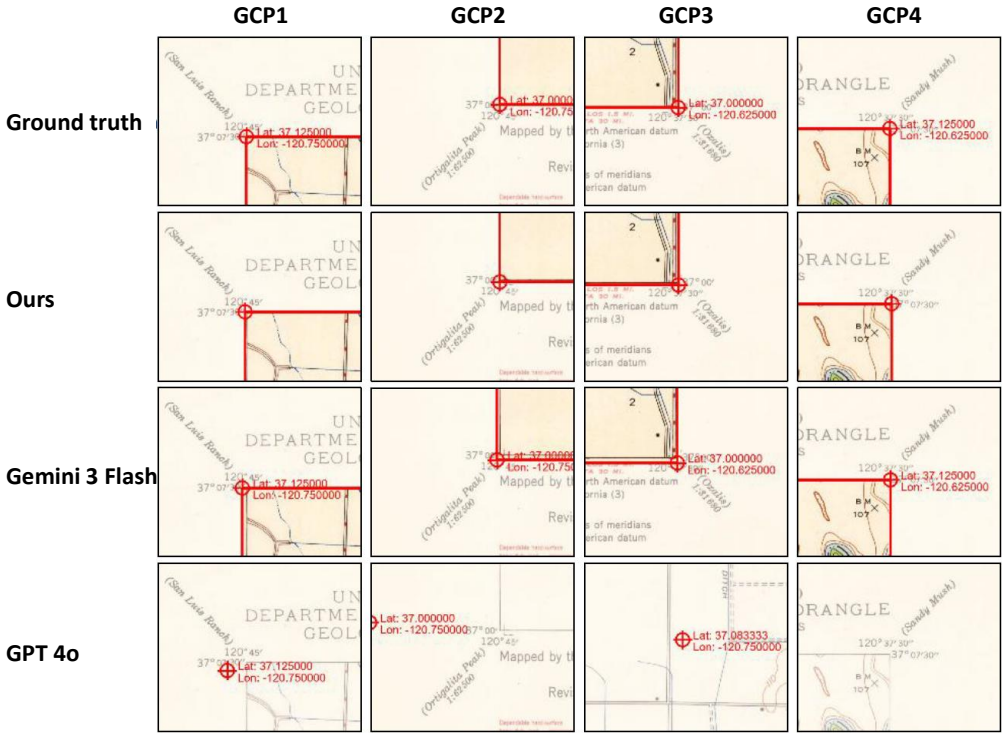


Figure A.1: Comparison of map segmentation results. We display the geocoordinates alongside the identified corners (i.e., source GCPs). GPT fails to predict the map content area.

For map-content segmentation, we report the accuracy in terms of IoU, precision, recall, and F1 score in A.1. Our approach outperforms Gemini 3 Flash and GPT 4o across all metrics, achieving IoU, precision, recall, and F1 scores above 0.98. In addition, we present a case study in A.1. Although Gemini 3 Flash achieves precision comparable to our approach with accurate text-based geocoordinate detection, it has lower, fluctuating recall and IoU due to its limited ability to locate pixel coordinates. While GPT 4o can extract text related to ground control points (GCPs), it suffers from a low completion rate due to an insufficient number of identified GCPs and fails to accurately locate pixel coordinates for this segmentation task.

Appendix B

Details for Generalizing Digitization

B.1 Test-time Representation and Optimization

To ensure the completeness of Chapter 5, we provide a unified formulation of how the components introduced in the methodology section (Section 5.2.3 to Section 5.2.4) interact at test time to produce the final prediction for GLYPH.

Region and Legend Representations. Let $\mathcal{R} = \{r_i\}$ denote the set of minimal polygonal instances obtained from boundary-guided region partitioning. For each minimal polygonal instance r_i, a feature representation is extracted as

$$z_i = \phi(r_i),$$

where $\phi(\cdot)$ corresponds to the feature extraction procedure, embedding color statistics and gradient-based descriptors.

For each polygon map key ℓ_k, we similarly obtain an anchor representation

$$z_k^{(L)} = \phi(\ell_k),$$

which serves as a reference for that class.

Expert Predictions and Vote Aggregation. Let $v_i^{(A)}, v_i^{(B)}, v_i^{(C)} \in \mathbb{R}^{K+1}$ denote the

region-level vote counts obtained from the three expert model solutions, where K is the number of polygon map keys. The index $k \in \{0, 1, \ldots, K\}$ corresponds to class indices, with $k = 0$ representing the background class and $k \geq 1$ corresponding to polygon map keys. These vote counts are normalized to obtain

$$p_i^{(A)} = \frac{v_i^{(A)}}{\sum_k v_{ik}^{(A)}}, \quad p_i^{(B)} = \frac{v_i^{(B)}}{\sum_k v_{ik}^{(B)}}, \quad p_i^{(C)} = \frac{v_i^{(C)}}{\sum_k v_{ik}^{(C)}}.$$

Each expert also produces a majority label for region r_i, denoted as

$$y_i^{(A)},\ y_i^{(B)},\ y_i^{(C)} \in \{0, 1, \ldots, K\}.$$

Pseudo-label Generation from Expert Agreement. To obtain reliable self-supervision signals at test time, we derive pseudo labels based on inter-expert agreement. A region is considered reliable if at least two experts agree:

$$y_i^{(A)} = y_i^{(B)} \ \text{ or } \ y_i^{(A)} = y_i^{(C)} \ \text{ or } \ y_i^{(B)} = y_i^{(C)}.$$

For such regions, the pseudo label $\tilde{y}_i$ is assigned by a majority vote among the agreeing experts. Regions without sufficient agreement are excluded from pseudo-label supervision.

Representation Learning via Contrastive Loss. GLYPH performs lightweight per-image representation learning to align region features with legend anchors. Let z denote the embedded features produced by a learnable embedding function. We apply a supervised contrastive loss

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\sum_{j \in \mathcal{P}(i)} \exp(\text{sim}(z_i, z_j)/T)}{\sum_{j \neq i} \exp(\text{sim}(z_i, z_j)/T)},$$

where $\mathcal{P}(i)$ denotes the set of samples (map-key pixels and pseudo-labeled regions) sharing the same polygon map key as sample i, and $\text{sim}(\cdot, \cdot)$ is cosine similarity. T is a temperature parameter that controls the concentration of the contrastive distribution, and it is set to 0.2.

Two types of samples are used: (1) pixels extracted from polygon map keys with known

labels, and (2) minimal polygonal instances with pseudo labels obtained from expert agreement.

We then define

$$\mathcal{L}_{\text{legend}} = \frac{1}{|\mathcal{S}_{\text{legend}}|} \sum_{i \in \mathcal{S}_{\text{legend}}} \mathcal{L}_{\text{contrast}}(i),$$

and

$$\mathcal{L}_{\text{region}} = \frac{1}{|\mathcal{S}_{\text{region}}|} \sum_{i \in \mathcal{S}_{\text{region}}} \mathcal{L}_{\text{contrast}}(i),$$

where $\mathcal{L}_{\text{contrast}}(i)$ denotes the contrastive loss computed for sample i.

The overall representation learning objective is

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{legend}} + \lambda_{\text{region}} \mathcal{L}_{\text{region}},$$

where $\lambda_{\text{region}} = 0.5$.

Adaptive Weighting (Gating). To adaptively balance the contributions of different expert solutions, GLYPH employs a gating module that produces weights per image

$$w = (w_A, w_B, w_C), \quad \sum_e w_e = 1,$$

where $e \in \{A, B, C\}$ indexes the expert solutions.

The gating input is a feature vector

$$x = [x_{\text{style}}, x_{\text{agreement}}],$$

where x_{style} consists of global image statistics, including the mean and standard deviation in various color spaces, and average gradient magnitude. While the $x_{\text{agreement}}$ consists of aggregated agreement statistics derived from expert predictions, including mean and median of majority confidence, and consistency across experts.

The gating function is implemented as a lightweight neural network

$$w = \text{softmax}(\text{MLP}(x)),$$

where the MLP is a shallow network with a single hidden layer that maps the input feature vector to three output scores corresponding to the expert solutions. This design allows the gating module to adaptively adjust expert contributions based on both global style variations (x_{style}) and the reliability of expert predictions ($x_{\text{agreement}}$).

The gating module is jointly optimized with the representation learning component during per-image test-time adaptation.

Gating Regularization. To control the sharpness of expert weighting, we introduce an entropy-based regularization term

$$\mathcal{L}_{\text{gate}} = \lambda_{\text{gate}} \left| H(w) - \tau_g \right|,$$

where

$$H(w) = -\sum_e w_e \log w_e.$$

The target entropy τ_g is determined by inter-expert disagreement:

$$\tau_g = \tau_{\text{base}} + \tau_{\text{scale}} \cdot \text{disagree\_rate},$$

where disagree_rate is defined based on the majority labels $y_i^{(A)}, y_i^{(B)}, y_i^{(C)}$ assigned by the expert solutions as

$$d_i = \frac{1}{3}\Big(\mathbf{1}[y_i^{(A)} \neq y_i^{(B)}] + \mathbf{1}[y_i^{(A)} \neq y_i^{(C)}] + \mathbf{1}[y_i^{(B)} \neq y_i^{(C)}]\Big),$$

$$\text{disagree\_rate} = \frac{1}{|\mathcal{R}|}\sum_i d_i.$$

In our implementation, we set $\tau_{\text{base}} = 0.4$, $\tau_{\text{scale}} = 0.8$, and $\lambda_{\text{gate}} = 0.1$. This regularization

is applied during per-image optimization to adaptively control the confidence of expert weighting based on agreement.

Adaptive Fusion. The fused prediction for each region is computed as

$$p_i = w_A p_i^{(A)} + w_B p_i^{(B)} + w_C p_i^{(C)}.$$

where $p_i \in \mathbb{R}^{K+1}$ denotes the fused class score vector, and p_{ik} denotes its k-th entry (k-th class) corresponding to polygon map key ℓ_k (with an additional index $k = 0$ for the background class).

Similarity-based Refinement. To incorporate legend guidance, we compute cosine similarity between region embeddings and legend anchors:

$$S_{ik} = \mathrm{sim}(z_i, z_k^{(L)}).$$

The foreground score is computed as

$$\mathrm{score}_{ik} = p_{ik} + \lambda_{\mathrm{sim}} S_{ik},$$

where p_{ik} denotes the fused score of assigning minimal polygonal instance r_i to the polygon map key ℓ_k, and S_{ik} denotes the cosine similarity between the region embedding z_i and the legend anchor $z_k^{(L)}$. The λ_{sim} is set to 1.0 in our implementation.

Each minimal polygonal instance r_i is then assigned the label corresponding to the maximum foreground score.

To handle background assignment, we compute a background score as $p_{i0} + \lambda_{\mathrm{bg}}$, where λ_{bg} is a background buffer and is set to 0.25 in our implementation. A region is assigned to the background class if

$$p_{i0} + \lambda_{\mathrm{bg}} > \max_{k \geq 1} \mathrm{score}_{ik},$$

or if

$$\max_{k\geq 1} S_{ik} < \tau_{\text{bg}} \quad \text{and} \quad p_{i0} > (1 - \tau_{\text{bg}}).$$

where τ_{bg} is the rejection threshold and is set to 0.80 in our implementation.

The above formulation describes how GLYPH integrates per-image representation learning, agreement-based pseudo labeling, adaptive expert weighting, and similarity-guided refinement to produce final polygon digitization results without requiring supervision in the target map domain.

B.2 Results on Pairwise Fusion Improvement

Following Section 5.4.4.8 and to assess the robustness of GLYPH (Chapter 5) in improving suboptimal inputs, we conduct a pairwise fusion analysis across comparative methods. Since GLYPH operates on three inputs, we construct each evaluation setting by fixing one input as LOAM (Chapter 2) and pairing it with two additional methods. This ensures a consistent anchor while enabling controlled comparison across diverse model combinations.

For each pair of comparative methods, we evaluate GLYPH's performance relative to the best-performing individual input among the three candidates. We present the results in a series of tables, with each cell in the table reporting the performance improvement of GLYPH over the "best-of-three-inputs" baseline. This quantifies the benefit of the fusion achieved by GLYPH beyond selecting the strongest standalone models.

Each block in the table represents a pair of comparative methods. The row and column ordering reflects progressively smaller tile sizes from left to right and from top to bottom, allowing us to examine how spatial granularity affects fusion behavior. The diagonal cells correspond to cases where the same method and tile size are paired with itself, effectively isolating LOAM's contribution to GLYPH. These cells serve as a reference to understand whether gains arise from complementary information across methods or simply from the fusion mechanism itself.

FT Dataset

We present the pairwise expert fusion improvement under GLYPH for the FT dataset in Figure B.1 for MMPQ, Figure B.2 for F1@8, Figure B.3 for P@8, and Figure B.4 for R@8.

In these figures, rows and columns enumerate candidate expert models (comparative methods, in the order of tile sizes for each method), and each cell represents a pair. The value in each cell denotes the improvement of GLYPH when combining LOAM with the pair of comparative methods. The improvement is defined as the performance of GLYPH minus the best performance among LOAM and the pair of two integrated comparative methods. Larger positive values indicate stronger complementarity among the methods. This series of pairwise results enables the interpretation of GLYPH's ability to synergize LOAM's solutions with different pairs of models' solutions and conditions.

With its emphasis on polygonal geometries, GLYPH achieves improved MMPQ compared to the best standalone methods, including LOAM, for most pairs. The improvement in MMPQ seems more obvious when integrated with at least one VLM at a smaller tile size. The tile size of the integrated SAM2 or SAM3 solutions does not seem to affect the improvement.

By setting a smaller tile size (rightward for each method; downward-rightward for each block of cells), most pairs of combinations achieve better R@8 improvement across their three integrated solutions. In contrast, there is no clear trend in P@8 improvement across pairs of fused methods. This may be due to the generally high precision of solutions derived by LOAM. Consequently, most pairs of combinations yield a slightly improved F1@8 across all integrated solutions. The only exception is the SAM3, which results in an exceptionally good improvement in F1@8 when integrated with Gemini 3 Flash compared to the integration with other methods. This may be attributed to the significant difference between the solutions of Gemini 3 Flash, as well as LOAM, against SAM3, which makes GLYPH to rely only on LOAM and Gemini 3 Flash to produce the self-calibrated results.

The results for the FT dataset show that GLYPH can synergize LOAM's solutions across different model pairs, even when one of the two integrated methods produces poor solutions.

Figure B.1: Pairwise expert fusion improvement under GLYPH for the FT dataset in MMPQ. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

Figure B.2: Pairwise expert fusion improvement under GLYPH for the FT dataset in F1@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

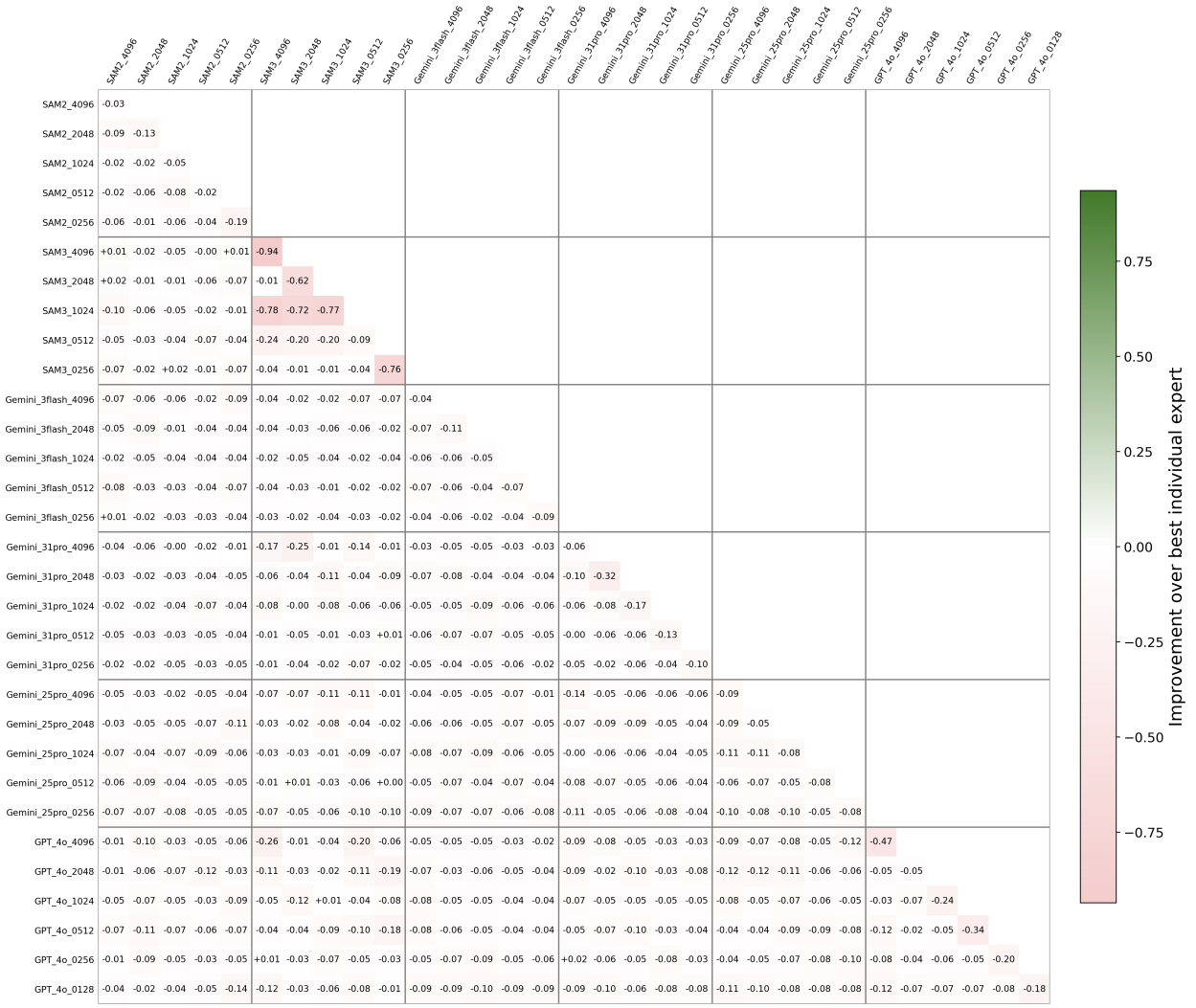


Figure B.3: Pairwise expert fusion improvement under GLYPH for the FT dataset in P@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

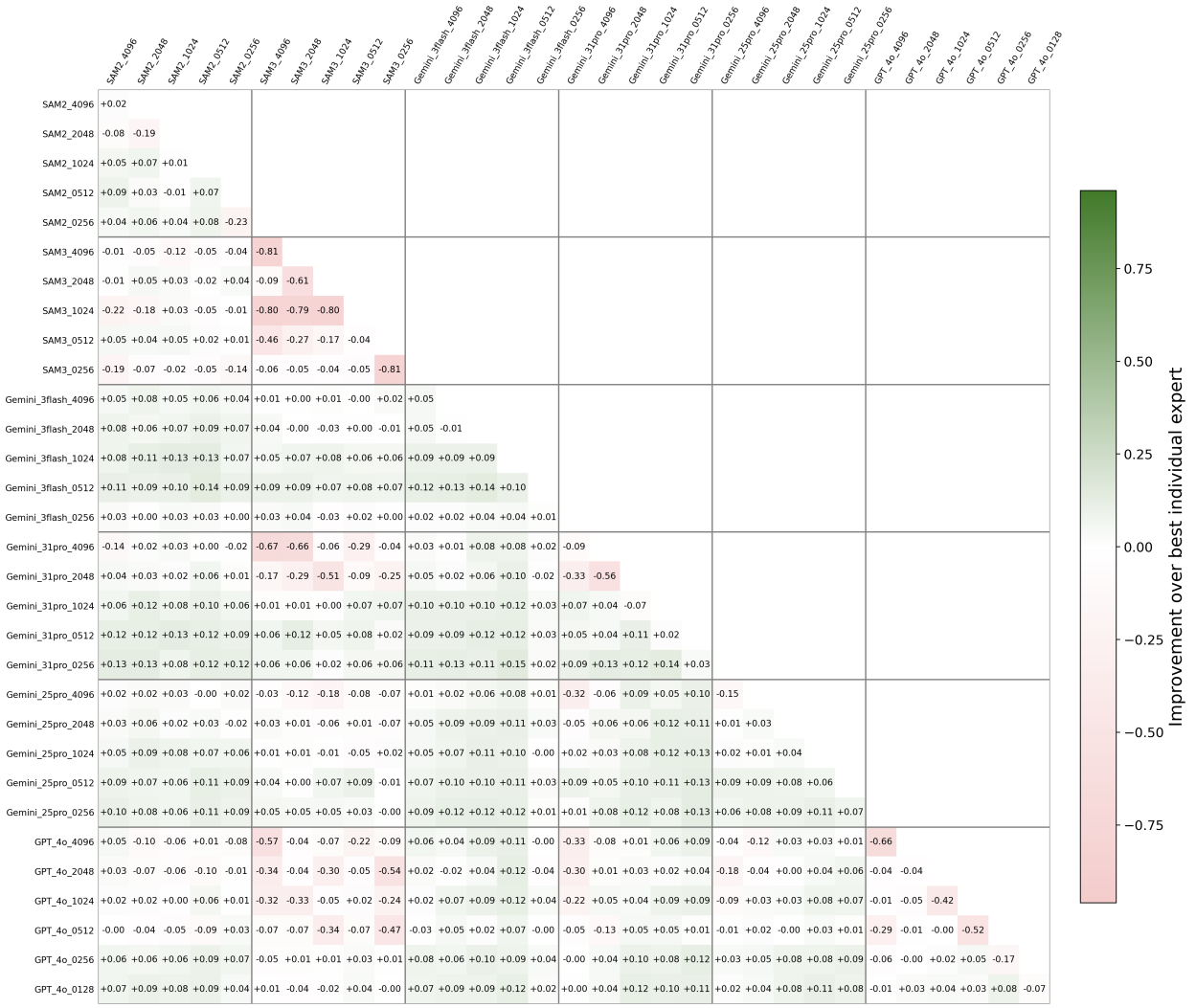


Figure B.4: Pairwise expert fusion improvement under GLYPH for the FT dataset in R@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

SA Dataset

We present the pairwise expert fusion improvement under GLYPH for the SA dataset in Figure B.5 for MMPQ, Figure B.6 for F1@8, Figure B.7 for P@8, and Figure B.8 for R@8.

While setting a smaller tile size for VLMs tends to yield a more significant improvement in R@8, the trends in P@8 and F1@8 improvements are less clear. As presented in Table 5.3, LOAM's solutions achieve performance worse than many of the comparative methods. Therefore, we can observe clearer improvements in both MMPQ and F1@8 across all pairs of combinations when treated as the input solutions, along with LOAM's, to GLYPH.

The pairwise improvement results in the SA dataset demonstrate GLYPH's ability to leverage and reconcile three methods' solutions, each with mediocre standalone performance, into a self-calibrated solution with significantly better instance-based and pixel-based accuracies.

SO Dataset

We present the pairwise expert fusion improvement under GLYPH for the SO dataset in Figure B.9 for MMPQ, Figure B.10 for F1@8, Figure B.11 for P@8, and Figure B.12 for R@8.

Given that LOAM's solutions achieve high accuracy on this out-of-domain dataset, the improvements GLYPH can bring across all metrics are limited. Still, we observe slight improvements in MMPQ across different sets of three integrated methods. The pair of adopting two SAM3-based solutions consistently yields negative gains for the GLYPH, which may be due to the limited consistency and informative patterns within SAM3's solutions on our task. This negative effect is mitigated when combining SAM3 solutions with another method, in which case GLYPH can rely on methods with more identifiable success or failure patterns, including LOAM and any of the remaining comparative methods. For pixel-level accuracy (F1@8, P@8, and R@8), the differences for the remaining pairs are not significant relative to each set's best model, which is LOAM in most cases.

The pairwise improvement results in the SO dataset demonstrate GLYPH's ability to maintain decent accuracy when one of the three integrated methods consistently produces high-quality results, identified via the legend anchor and consensus mechanism.

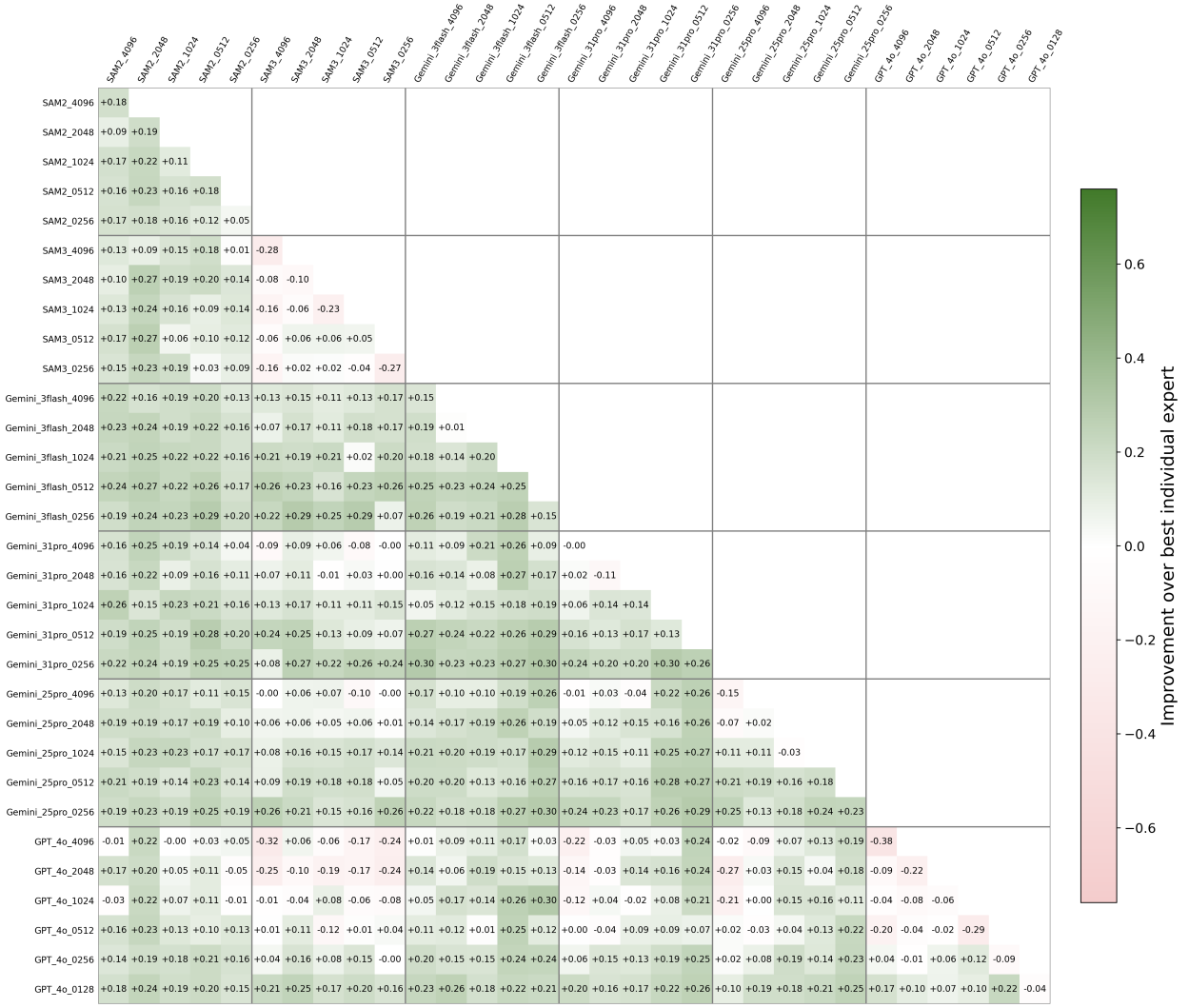


Figure B.5: Pairwise expert fusion improvement under GLYPH for the SA dataset in MMPQ. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

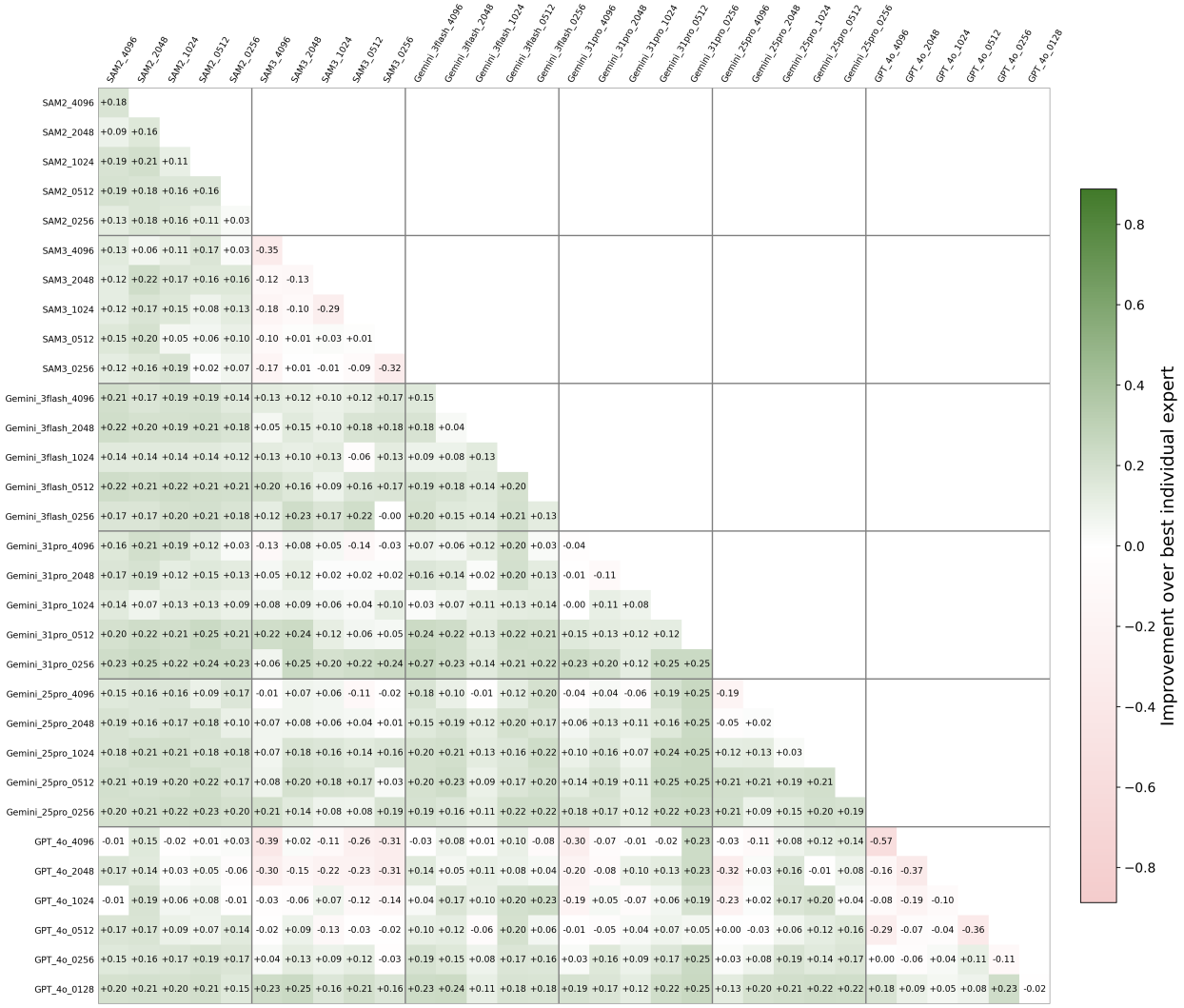


Figure B.6: Pairwise expert fusion improvement under GLYPH for the SA dataset in F1@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

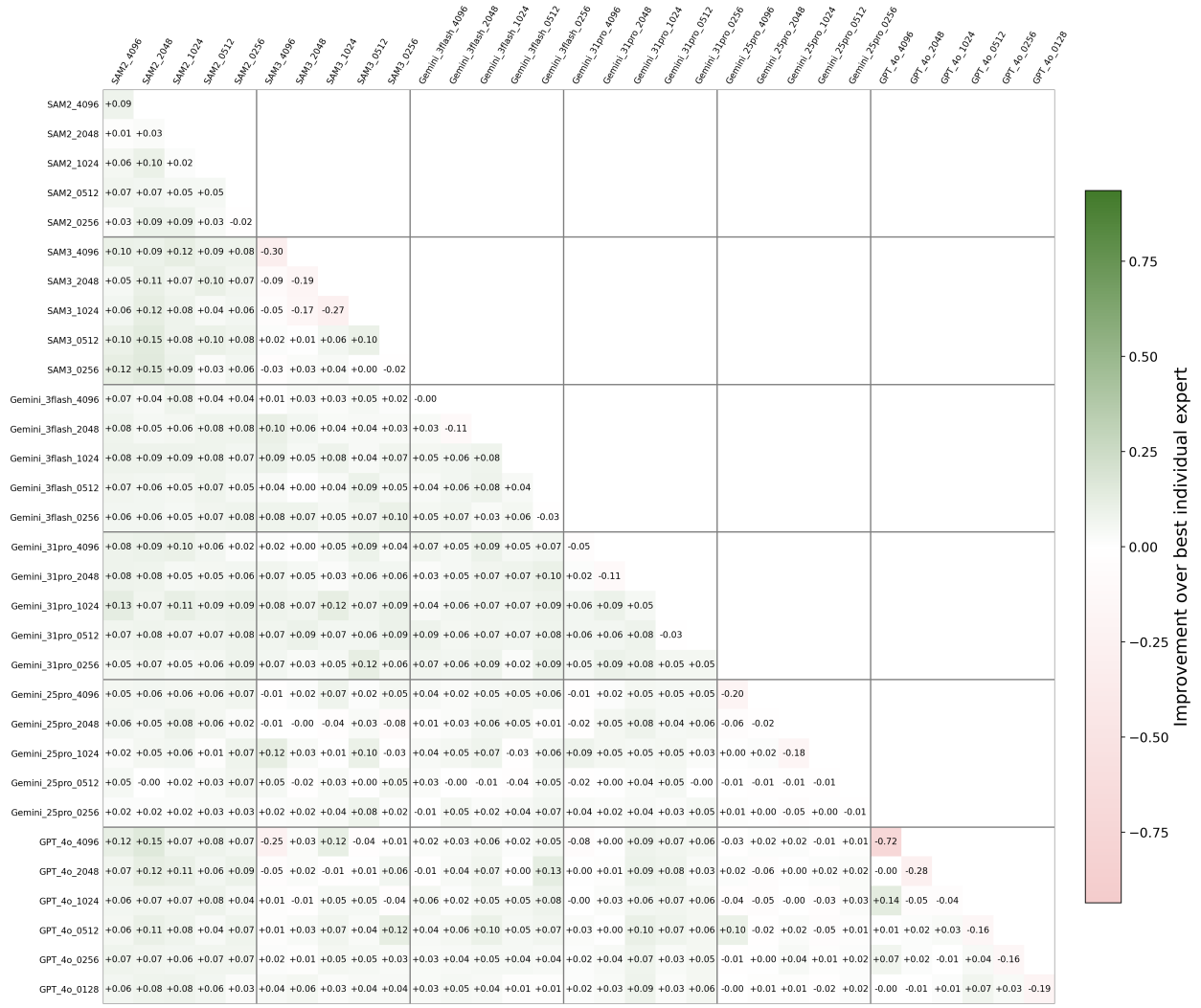


Figure B.7: Pairwise expert fusion improvement under GLYPH for the SA dataset in P@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

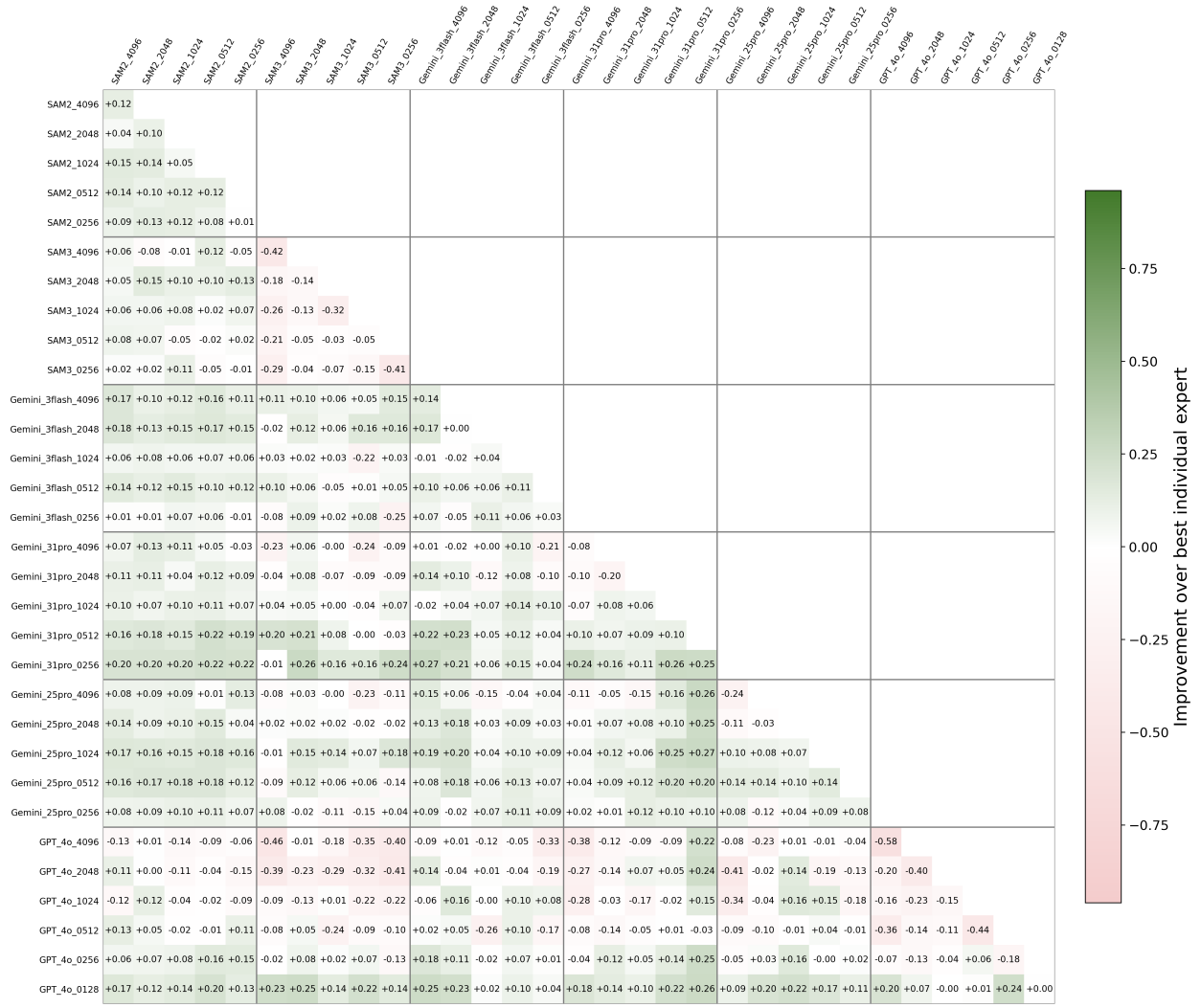


Figure B.8: Pairwise expert fusion improvement under GLYPH for the SA dataset in R@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

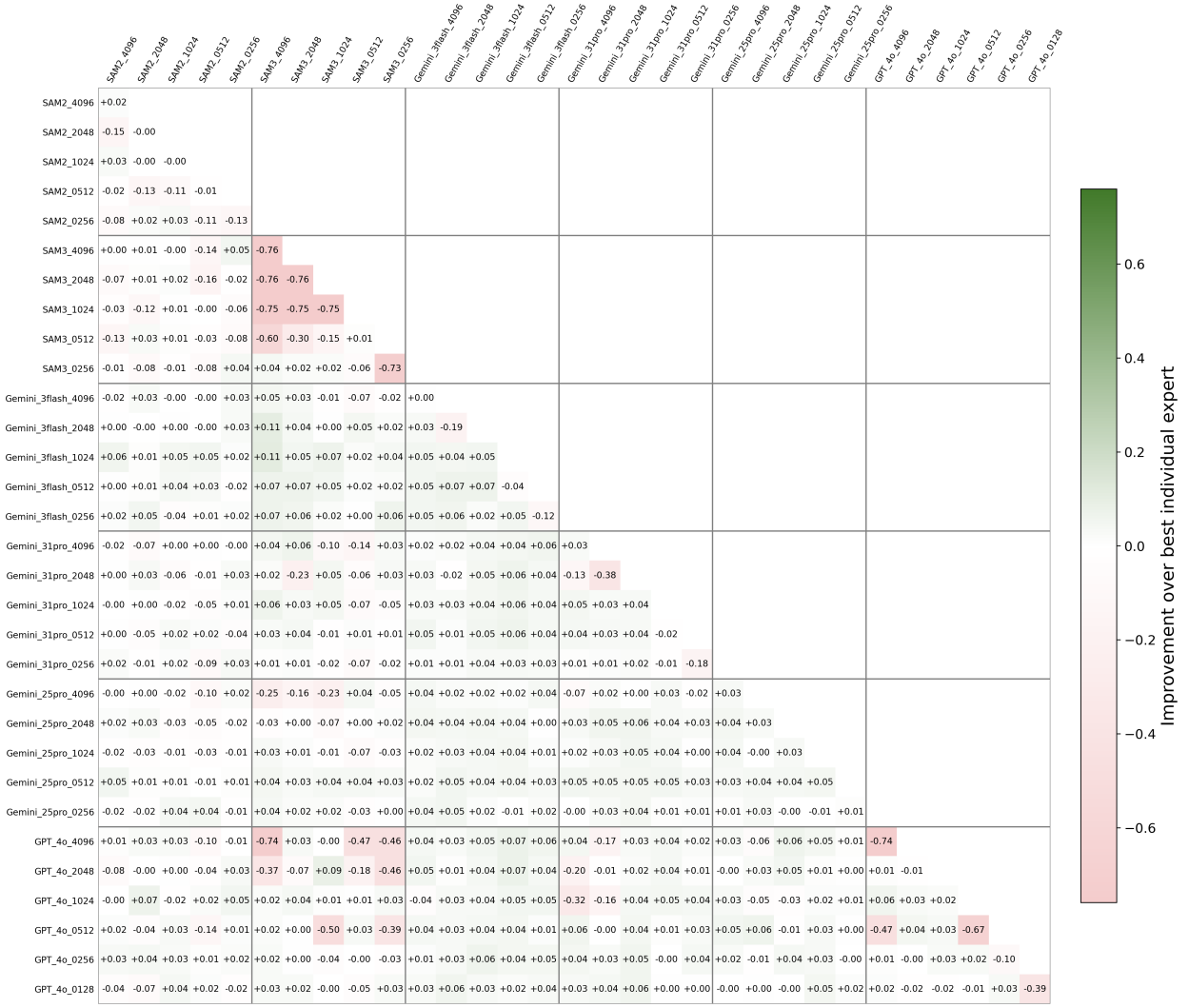


Figure B.9: Pairwise expert fusion improvement under GLYPH for the SO dataset in MMPQ. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

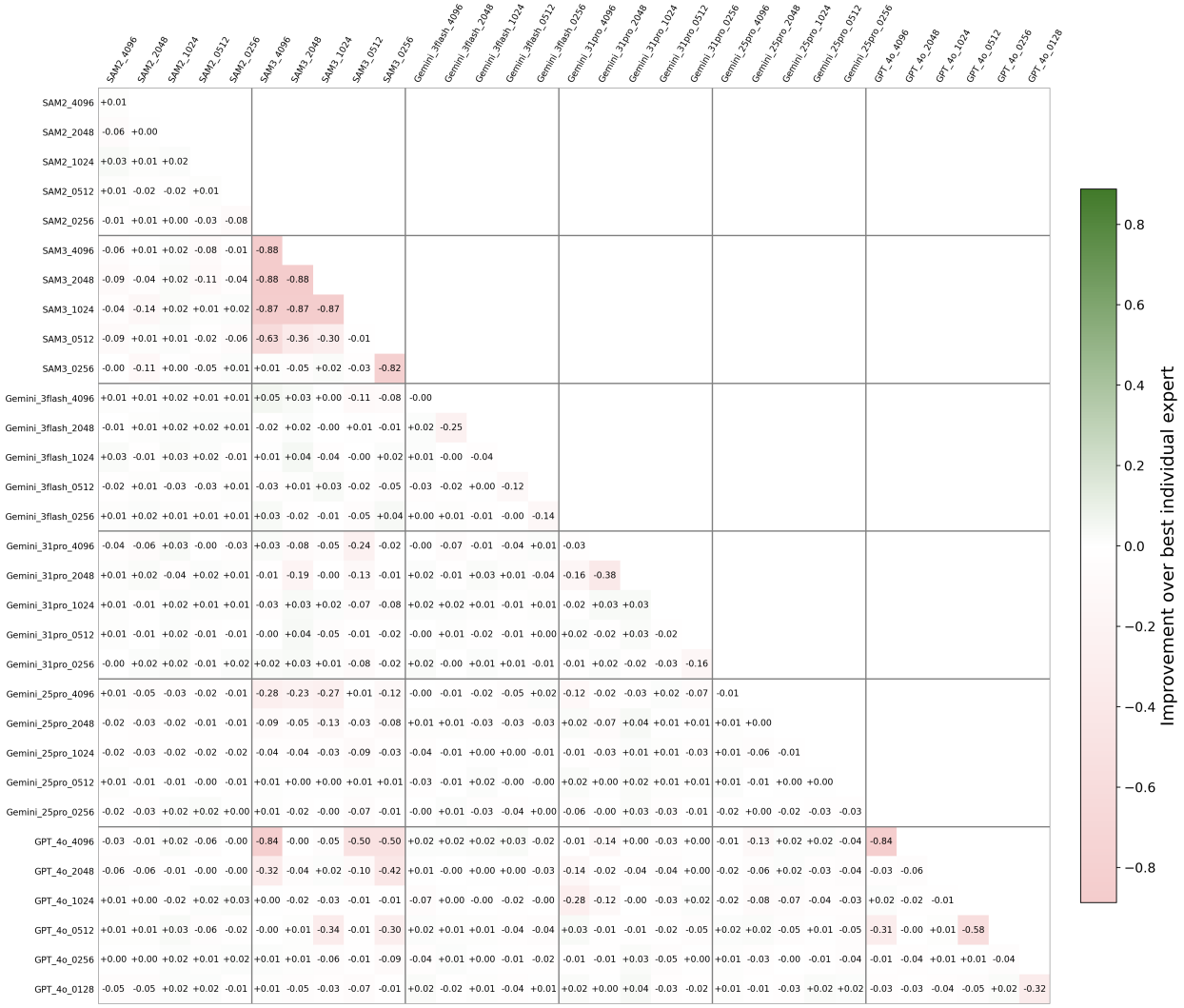


Figure B.10: Pairwise expert fusion improvement under GLYPH for the SO dataset in F1@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

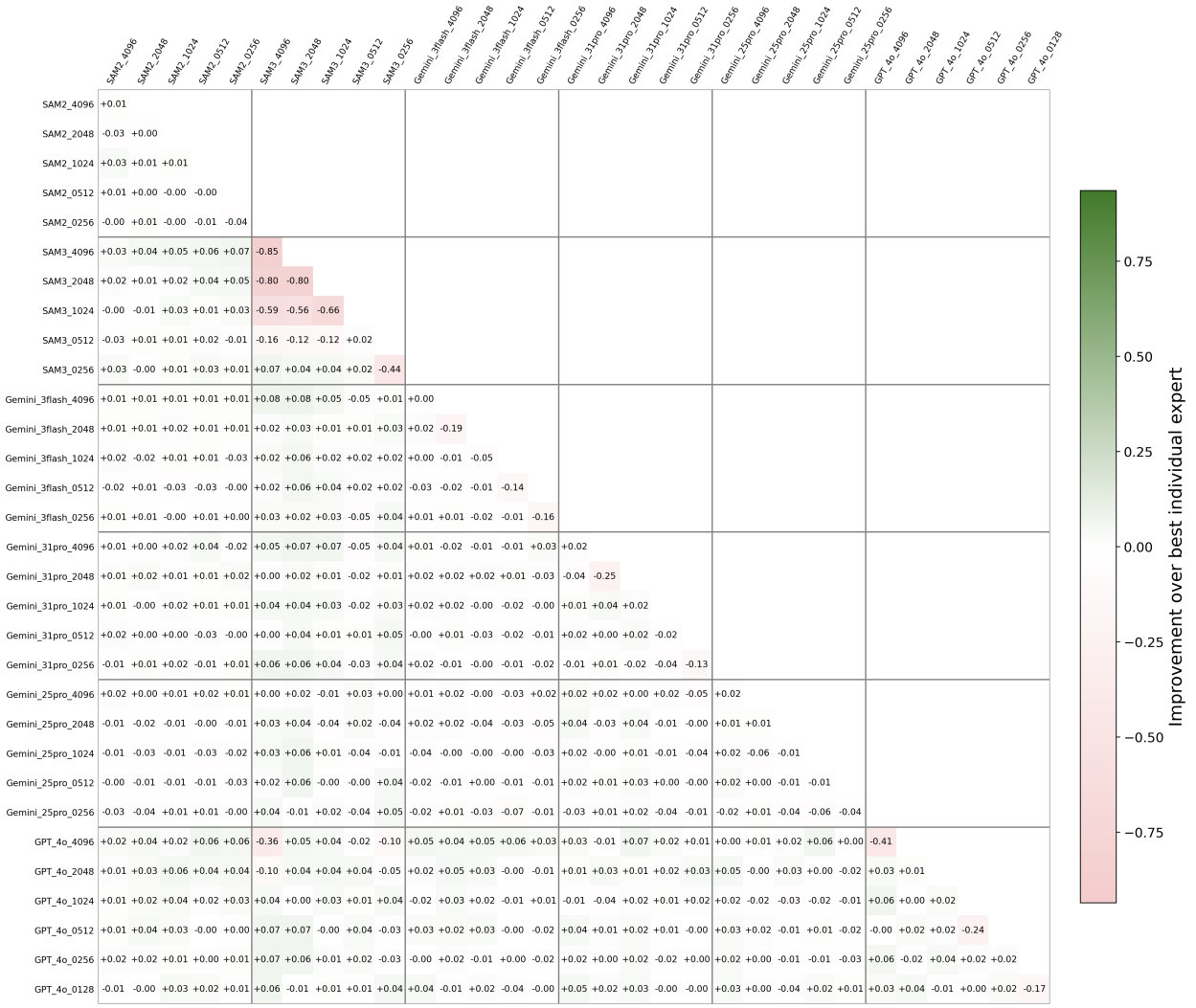


Figure B.11: Pairwise expert fusion improvement under GLYPH for the SO dataset in P@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

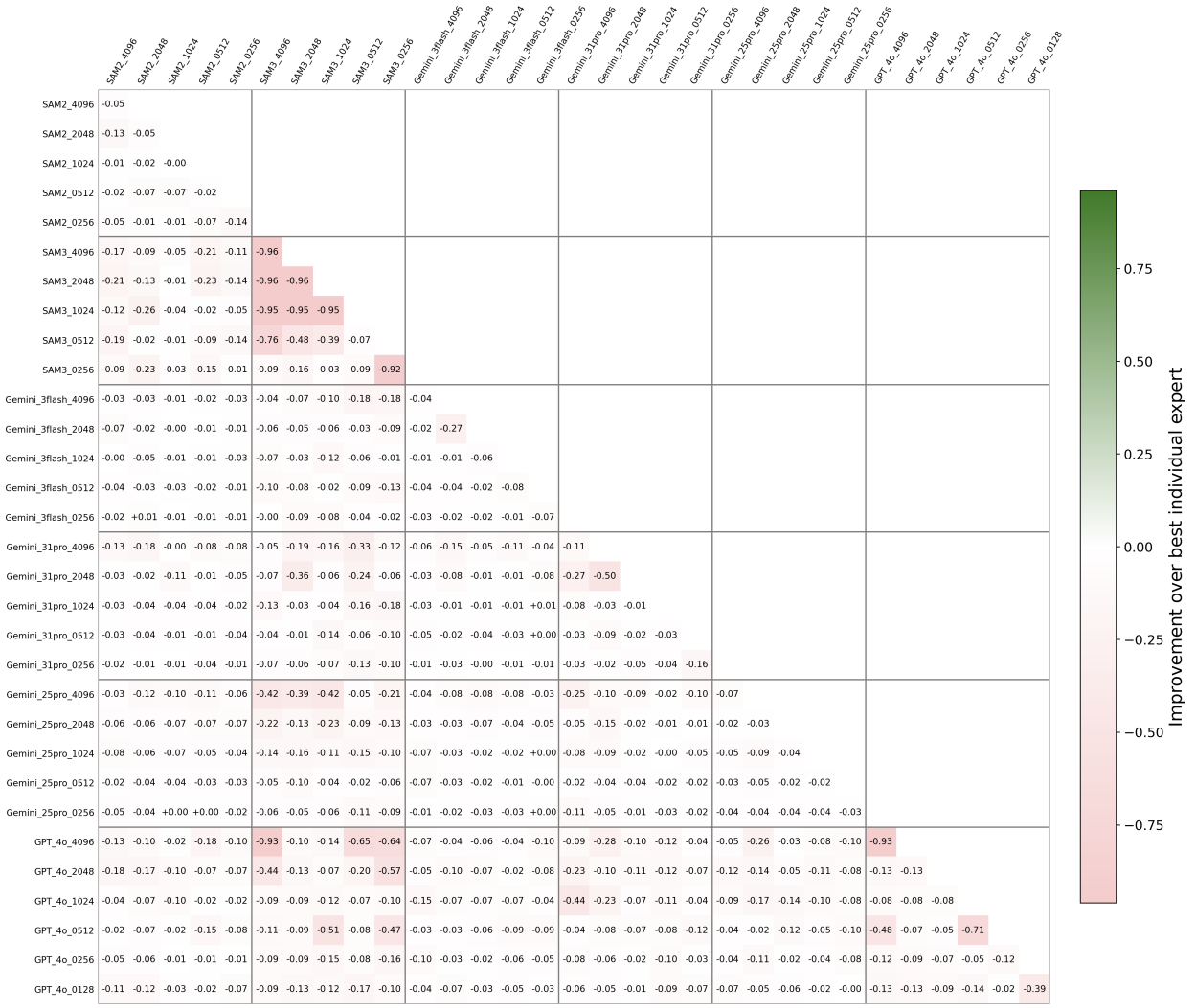


Figure B.12: Pairwise expert fusion improvement under GLYPH for the SO dataset in R@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

SP Dataset

We present the pairwise expert fusion improvement under GLYPH for the SP dataset in Figure B.13 for MMPQ, Figure B.14 for F1@8, Figure B.15 for P@8, and Figure B.16 for R@8.

The SP dataset is an out-of-domain dataset with significantly different cartographic styles compared to the in-domain GE dataset; hence, LOAM's accuracy is limited by extremely high recall and low precision, and we observe some improvement for each set of the three integrated methods, especially in MMPQ, P@8, and F1@8.

The degradation in R@8 is universal, due to LOAM's high recall. However, we observe a clear improvement in P@8 for most pairs, except for SAM2, which also shows less degradation in R@8. This may be attributed to the higher P@8 and stability due to entity linking in SAM2's standalone solutions. Although the integration of SAM3's solutions bring improvement in P@8, this does not contribute to their F1@8. For VLMs, setting a smaller tile size tends to bring a larger improvement in pixel-based accuracy, as depicted in the downward-rightward direction for each block of F1@8 in Figure B.14, demonstrating their ability to address CMYK-printed images. However, this trend is not that obvious for instance-based accuracy.

The pairwise improvement results in the SP dataset demonstrate GLYPH's ability to adaptively leverage methods' solutions with identifiable patterns in their accuracies across cartographic styles. This enables GLYPH to achieve decent results and noticeable improvement over the three integrated solutions, despite their mediocre standalone performance.

WR Dataset

We present the pairwise expert fusion improvement under GLYPH for the WR dataset in Figure B.17 for MMPQ, Figure B.18 for F1@8, Figure B.19 for P@8, and Figure B.20 for R@8.

The pairwise improvement results show an interesting trend: all combinations achieve significant improvements in MMPQ, but with a slight negative gain in pixel-based metrics. This is due to the exceptionally high F1@8 for LOAM's solutions, and the generally low

instance-based accuracy across all methods, including LOAM. As most integrated comparative methods struggle with this offset-printing dataset, GLYPH has difficulty improving the pixel-based accuracy of the final outputs. Meanwhile, the uneven color, nested polygon geometries, incomplete anchoring, and overlapping features limit GLYPH's ability to better identify the strengths and weaknesses across integrated solutions at test time.

The pairwise improvement results in the WR dataset reveal GLYPH's limitations, as all integrated methods produce inaccurate, overly complicated polygon geometries.

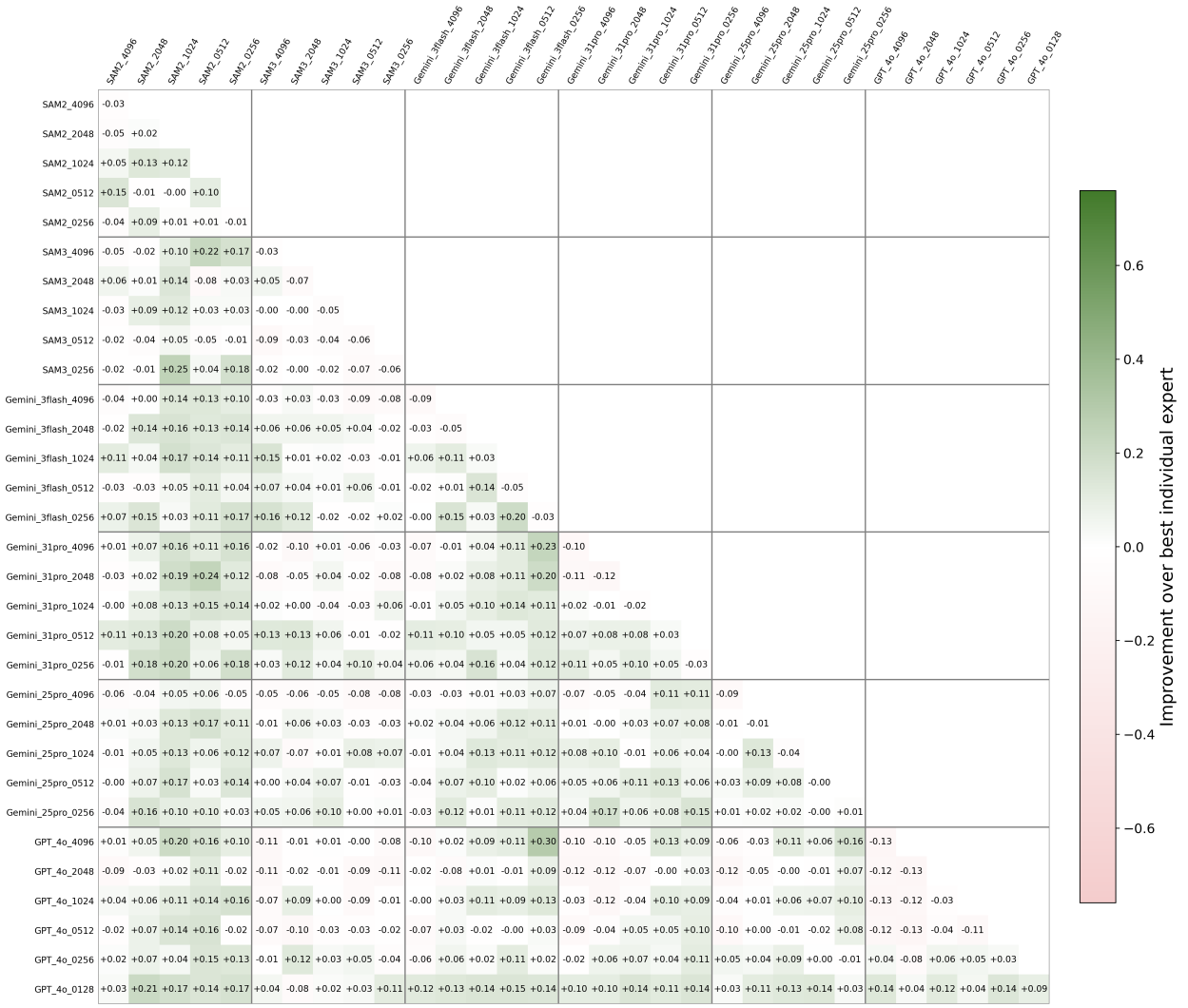


Figure B.13: Pairwise expert fusion improvement under GLYPH for the SP dataset in MMPQ. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

Figure B.14: Pairwise expert fusion improvement under GLYPH for the SP dataset in F1@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

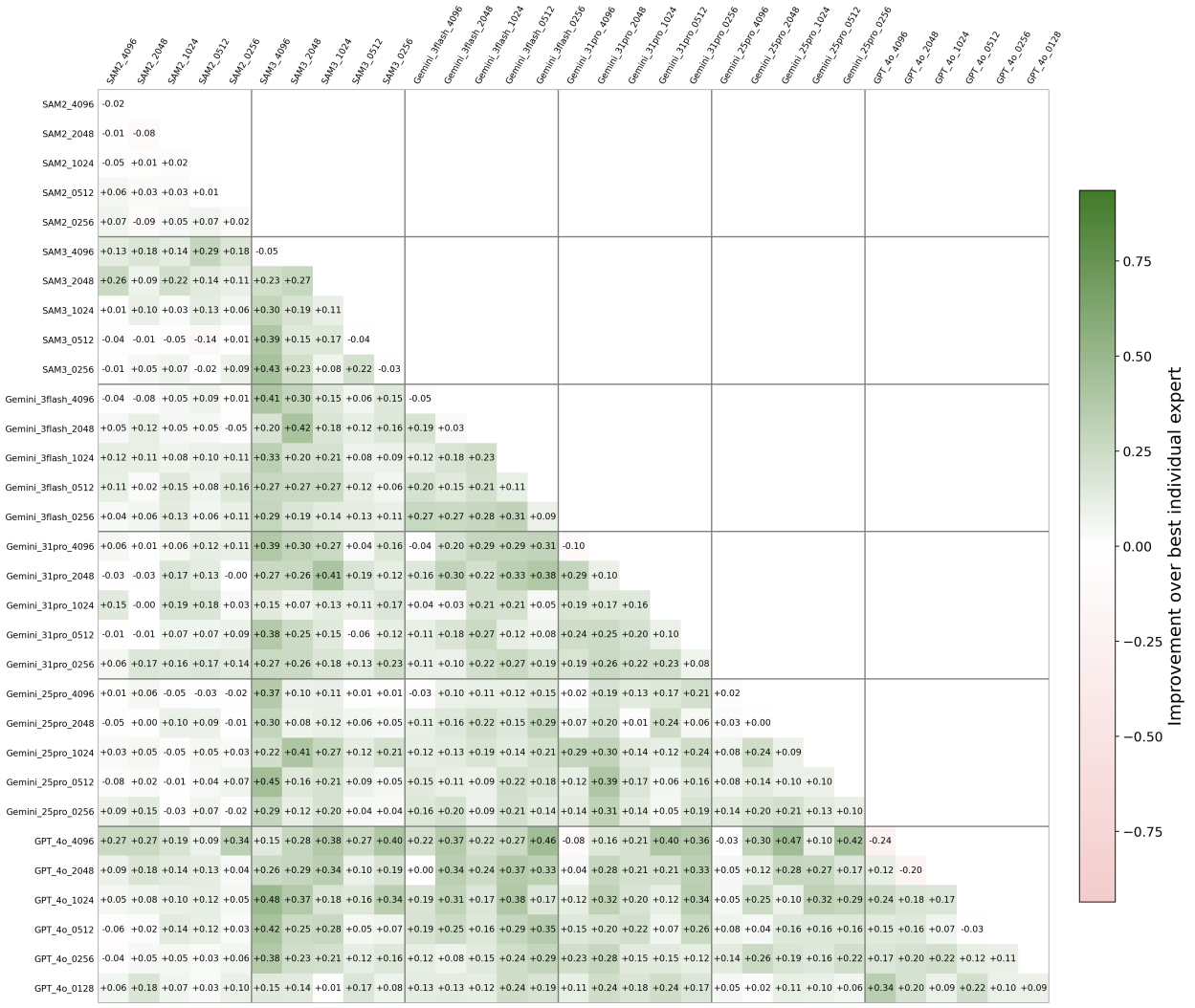


Figure B.15: Pairwise expert fusion improvement under GLYPH for the SP dataset in P@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

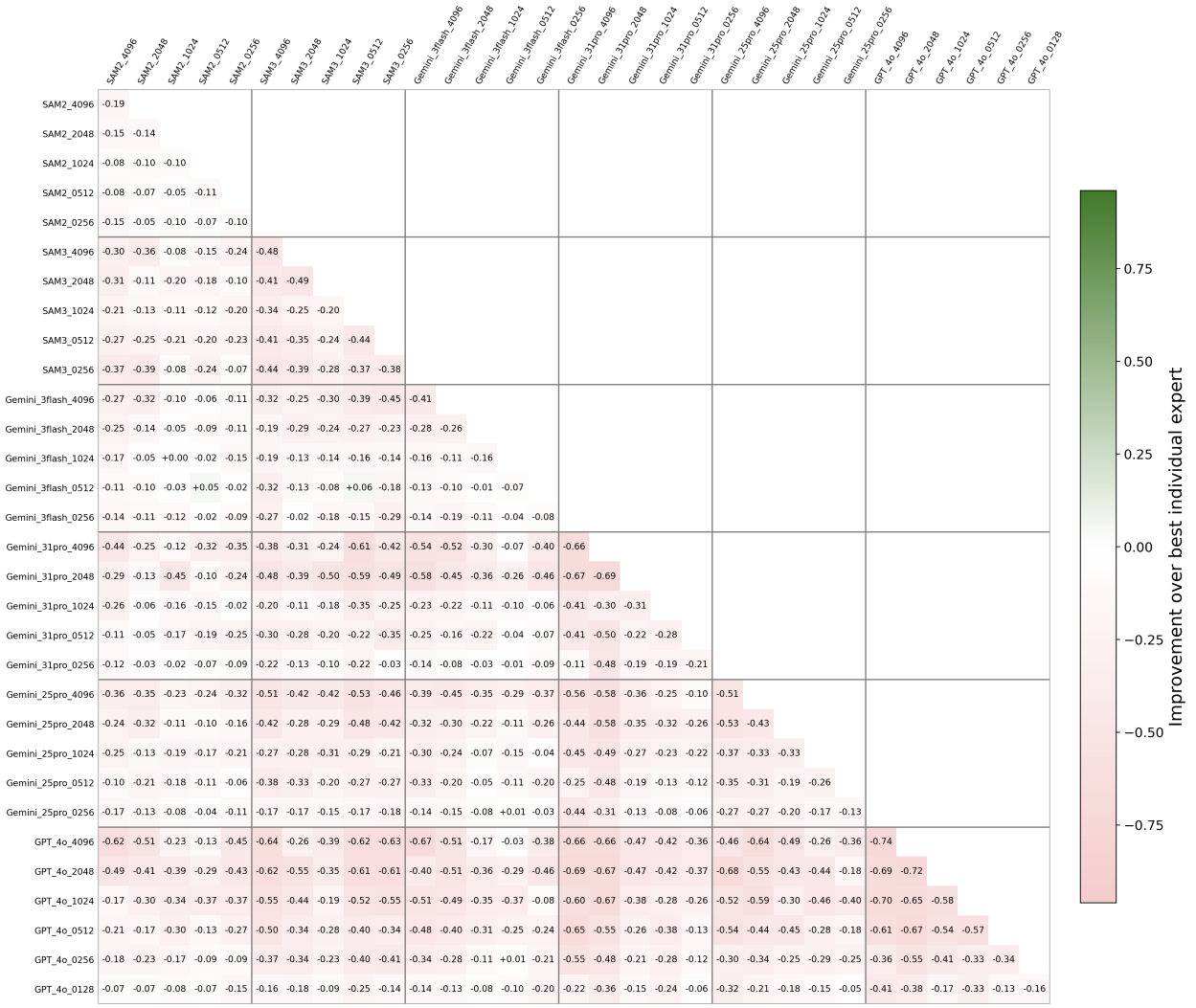


Figure B.16: Pairwise expert fusion improvement under GLYPH for the SP dataset in R@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

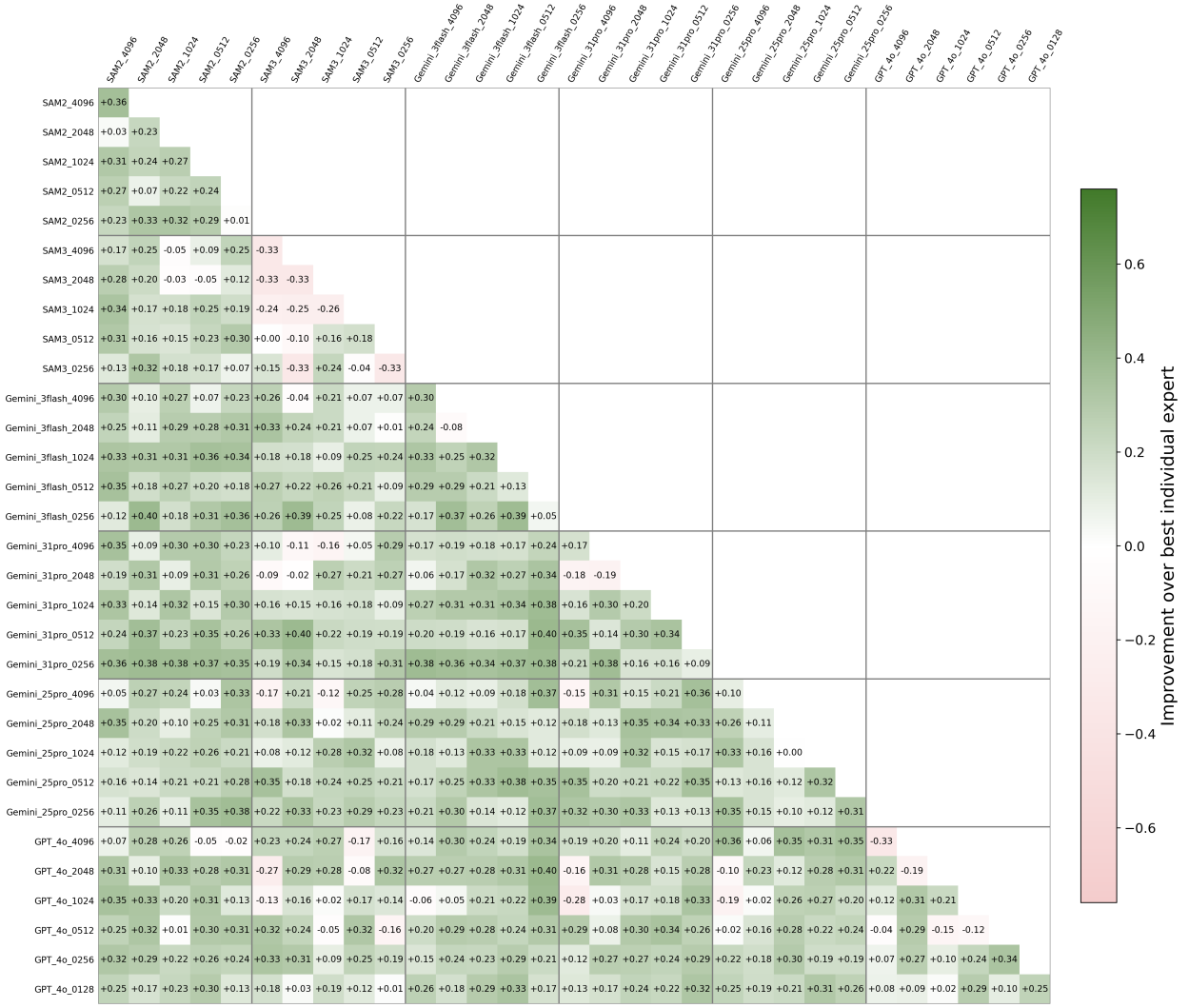


Figure B.17: Pairwise expert fusion improvement under GLYPH for the WR dataset in MMPQ. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

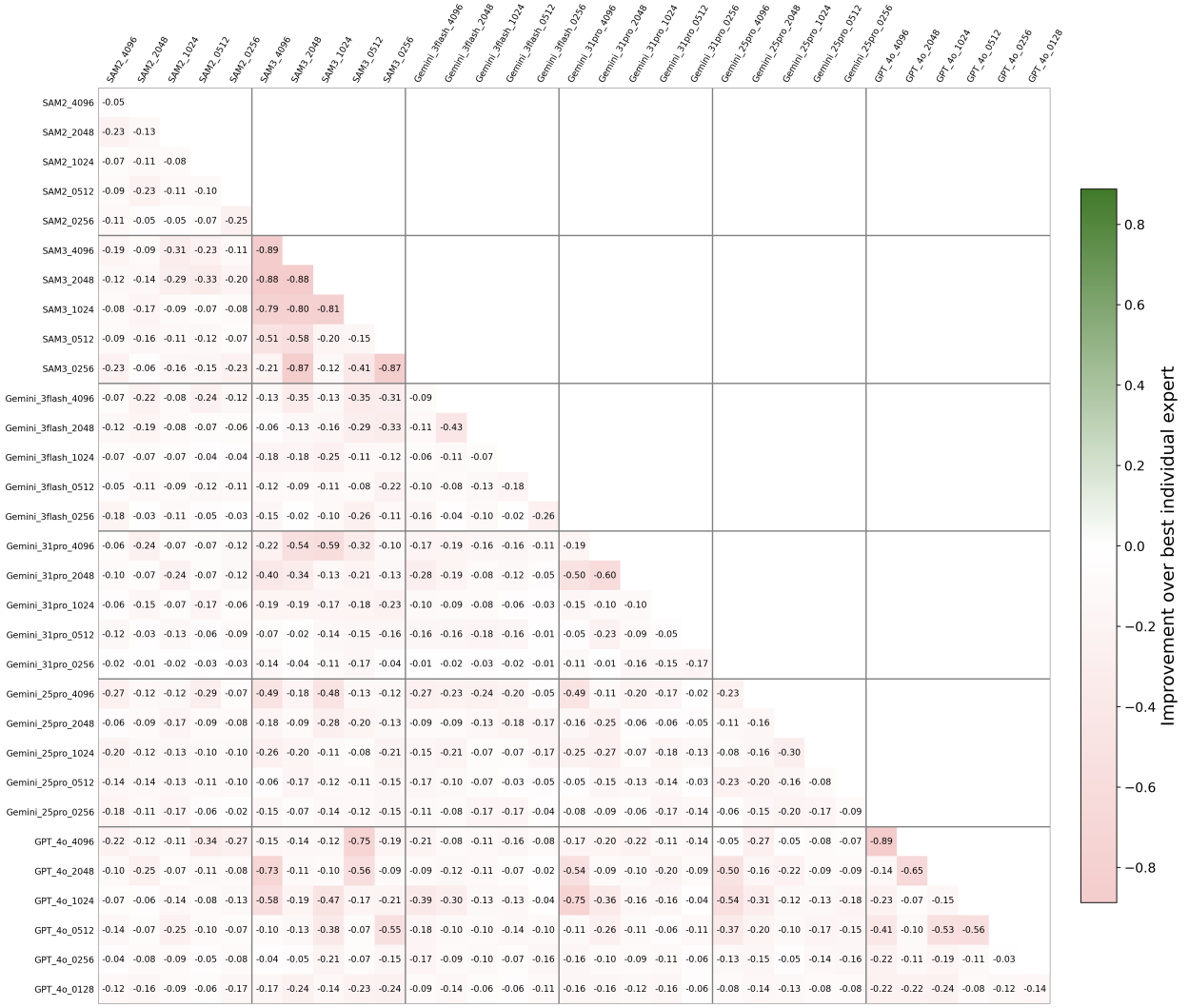


Figure B.18: Pairwise expert fusion improvement under GLYPH for the WR dataset in F1@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

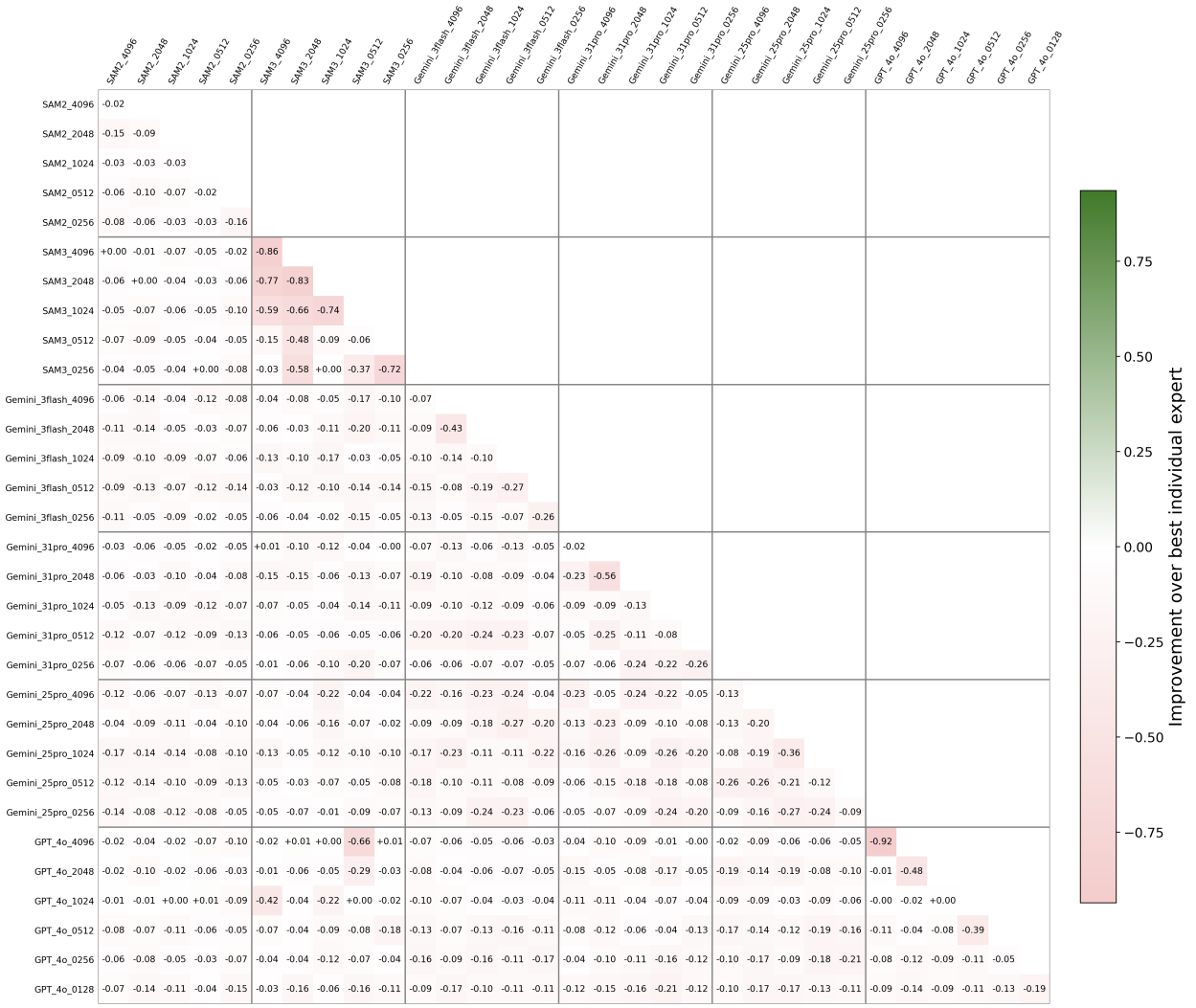


Figure B.19: Pairwise expert fusion improvement under GLYPH for the WR dataset in P@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.

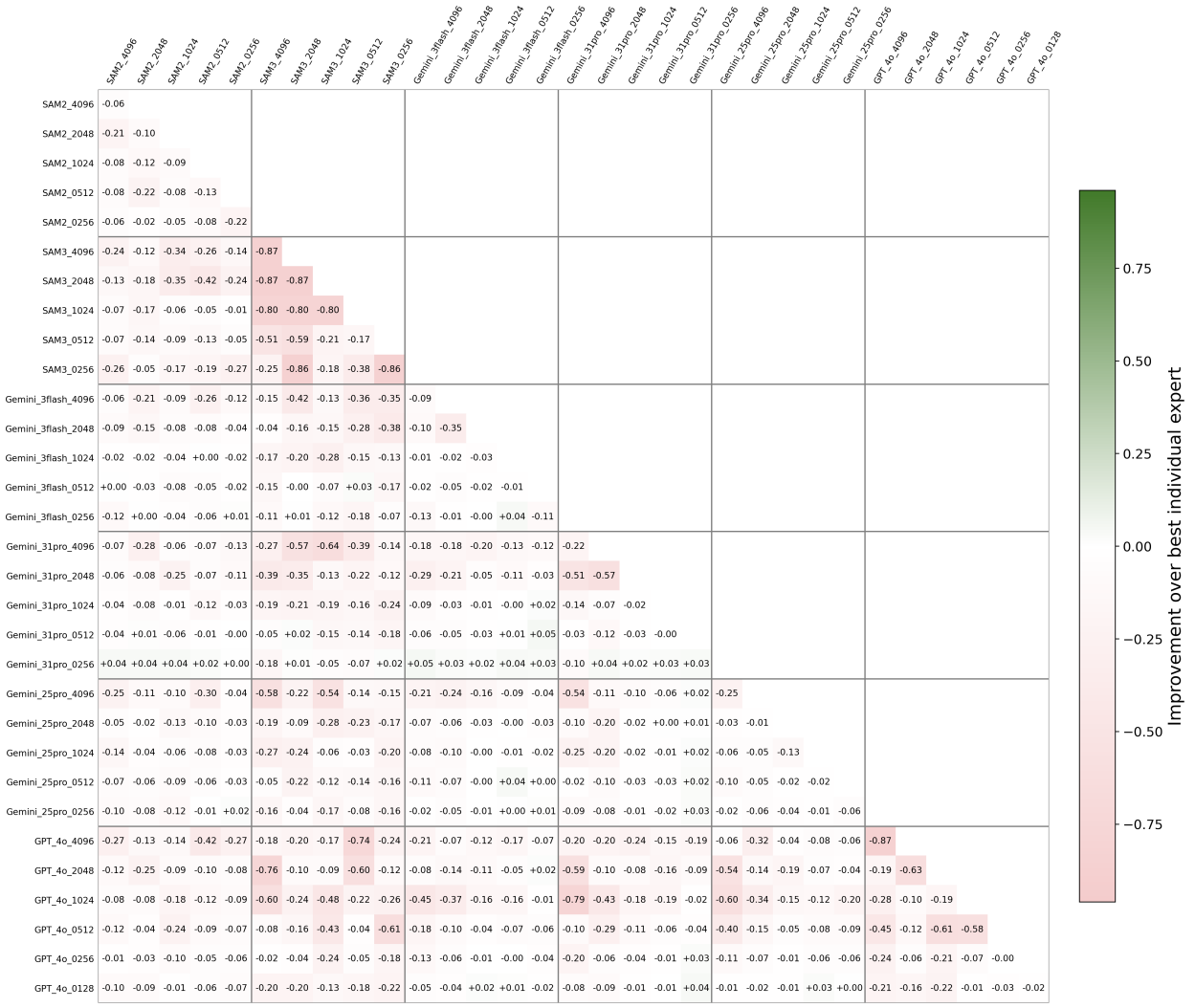


Figure B.20: Pairwise expert fusion improvement under GLYPH for the WR dataset in R@8. Each cell reports the performance gain for using the pair of two experts with LOAM as the third one. Positive values (green) indicate that GLYPH outperforms the best individual expert, while negative values (red) indicate degradation.