Understanding Diagrams with Explicit Intermediate Visual Representation

Wangcheng Xu WANGCHENG.XU@NORTHWESTERN.EDU
Kenneth D. Forbus FORBUS@NORTHWESTERN.EDU
Qualitative Reasoning Group, Northwestern University, 2233 Tech Drive, Evanston, IL 60208 USA

Abstract

Diagram understanding remains a challenge for current Vision-Language Models (VLMs), which often fail to accurately capture the fine-grained spatial and relational information essential for deep comprehension. Furthermore, their opaque internal states hinder effective human-machine collaboration. Inspired by human cognition, we propose an alternative approach that prioritizes the creation of explicit, human-readable representations. Producing intermediate visual representations that are compatible with the cognitively-inspired CogSketch, our system extends the effort of Hybrid Primal Sketch, which combines computer vision techniques to produce structured, symbolic descriptions of diagrams for CogSketch to further encode. This method generates explicit representations of visual elements and their qualitative spatial relationships, which can then support higher-level visual reasoning. Our approach is highly interpretable, lightweight, and training-free. We demonstrate its advantage on diagram understanding by extracting the underlying structural information in two genres of charts and diagrams.

1. Introduction

Diagram understanding is a crucial area of AI research, as diagrams are important for visual communication of complex and structural knowledge in science, engineering, and daily tasks. A long-standing challenge is for machines to extract and interpret conceptual and structured information from diagrams. One prominent approach, following the success of pre-trained language models, is pre-trained Vision-Language Models (VLMs) that jointly learn from both images and text at large scale. VLMs provide impressive out-of-box capabilities for a range of vision tasks, including diagram understanding. However, VLMs as standalone systems have significant limitations when collaborating with humans for diagram understanding due to their end-to-end nature. Users have no access to, or control over, the internal organization of visual features and entities that lead to the generated textual answers. When VLMs make mistakes, it's difficult to identify the source of the error. Moreover, VLMs have been shown to struggle with spatial reasoning that is trivial for humans (Wang et al. 2024b) and could be corrected with simple visual heuristics. However, without access to their intermediate representations, such heuristics cannot be integrated.

We explore an alternative approach, Heuristic-based Visual Ensemble (HVE), that complements spatial and geometric heuristics with advances in visual processing techniques for object detection, text spotting, visual segmentation, edge detection, and line segment detection to extract and process explicit visual representations, such as bars and ticks in bar charts and arrows

in a food web diagram. Heuristics like proximity, shape, and alignments between visual elements play an important role in diagram understanding. Using such heuristics, we argue, enables extracting the underlying structured information in an intuitive and interpretable manner, in contrast to black-box VLMs. Our approach draws inspiration from computational systems for high-level visual analysis like CogSketch (Forbus et al., 2011), which facilitates visual problem-solving (Forbus & Lovett, 2021) and sketch recognition (Chen et al., 2023) in a human-like manner. The Hybrid Primal Sketch (Forbus et al., 2024) used off-the-shelf vision components to produce inputs for CogSketch to further analyze. Our HVE extends the Hybrid Primal Sketch by leveraging the latest visual processing models to produce and analyze the necessary intermediate representations.

Our interest is centered around a paradigm for human-like visual processing that can be extended to a broad range of diagram genres without the need for training with substantial computing resources and effort to prepare task-specific data. We focus on off-the-shelf models and heuristics-based analysis. Our approach has the advantage of recognizing and interpreting fine-grained visual elements, thus complementing VLM's capabilities without fine-tuning. Specifically, in HVE, the bottom-up analysis built on visual elements that can be incorporated with top-down conceptual clues and constraints facilitated by a VLM. Our contributions in this paper are:

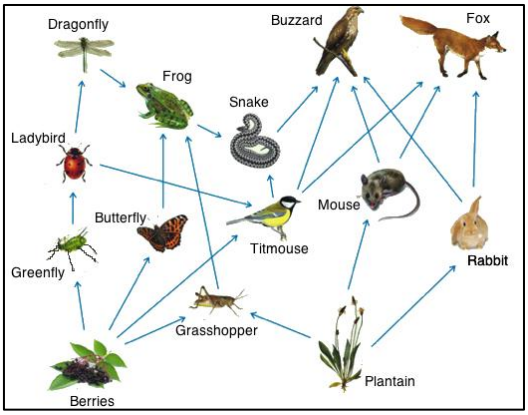
- We explore a human-like approach for diagram understanding by using an ensemble of computer vision techniques that produce explicit intermediate representations (e.g. entities, labels, arrows) that can be utilized by high-level vision systems like CogSketch, demonstrating relation and data extraction from images of diagrams.
- We show that our training-free approach can achieve competitive performance on real-world diagram datasets compared to standalone VLMs, sometimes with greater efficiency. Moreover, we show that our ensemble of other visual processing techniques with VLMs provides surprising performance improvements for a range of VLMs from small to large and from open-source to top-tier closed-source models.

We begin with background and related work, including CogSketch, VLMs and other relevant computer vision techniques. We discuss the design of HVE, demonstrating its operation on bar charts and food webs, with comprehensive evaluation on FoodWebs (Krishnamurthy et al., 2016) and VisText (Tang et al., 2023) datasets. We close with conclusions and future work.

2. Background & Related Work

2.1 CogSketch

CogSketch (Forbus et al., 2011) is an open-domain sketch understanding system that organizes ink into glyphs with conceptual labels. Glyphs serve as visual objects, and there are three types: entities, relations, and annotations. These provide a natural intermediate representation for understanding diagrams like the example food web and bar chart in Figure 1. Unlike most sketch understanding systems that emphasize automated recognition, CogSketch focuses on human-like reasoning about conceptual and relational information on top of visual elements (i.e., ink, glyphs) and conceptual labels. It can analyze the ink that constitutes a glyph to understand its shape, e.g., decomposing ink into edges and junctions and recombining them into surface-like structures (Forbus et al., 2017) and can organize ink into multi-level part-based qualitative representations



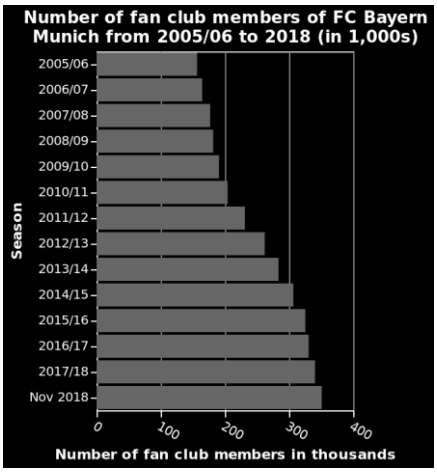


Figure 1: Example of a diagram from FoodWebs (left) and a bar chart from VisText (right)

for sketch recognition (Chen et al., 2023). CogSketch can be used to model human visual and geometrical reasoning (Forbus & Lovett, 2021). Therefore, CogSketch also serves as a platform to model high-level human vision. This paper is inspired by the prior work on CogSketch that automatically extracts glyphs by using neural object detection models for visual relation detection (Chen & Forbus, 2021). One of our goals is to combine a broader ensemble of visual processing techniques to support CogSketch handling more diverse and complex visual elements in diagrams like those in the FoodWebs and VisText datasets.

2.2 Vision Language Models

Vision-language models are based on training with vast sets of image-text pairs, and have gained popularity because of their effectiveness in a broad range of visual tasks. While they can be directly applied to downstream tasks such as image classification, visual question answering, and visual information extraction, their performance may not be optimal without fine-tuning. VLMs operate in an end-to-end manner, where the response of a query is produced without explicit intermediate visual representations, thus hardly inspectable. We use a range of VLMs of different sizes and include open-source and closed-source ones to compare but also combine with our approach. We use LLaVA-NeXT (Liu et al., 2024) and another variant, LLaVA-CoT (Xu et al., 2024) that achieved state-of-the-art results by training with multistage reasoning data. We also use CogVLM 2, a high-performance VLM that utilizes frozen pre-trained language models (Wang et al., 2023), the recent, widely recognized leading open-source VLM, Qwen2-VL (Wang et al., 2024a), Qwen2.5-VL (Bai et al., 2025), and closed-source models like GPT-4o, GPT-4o-mini (OpenAI, 2024), and Gemini (Google, 2024).

2.3 Scene Text Detection and Recognition

Text detection and recognition are essential in many applications to convert text in images to machine-readable strings. Progress in scene text detection provides off-the-shelf models with reasonable performance for locating and extracting text across diverse image types, from street signs to license plates. Such models can be conveniently plugged into a diagram understanding system for extracting text information from the diagrams. We use the UNITS model (Kil et al., 2023), which is a unified scene text spotter that can detect text in arbitrary shapes by unifying various detection formats and using starting-point prompting to handle more text instances than it was trained on. For ablation, we also use a detect-then-recognize pipeline, first detecting the area of text occurrence and then feeding features of the detected area to the text recognizer to extract the text. We use CRAFT (Baek et al., 2019), a model trained at the character level for robust and flexible text detection, for text box detection, along with the LPV model (Zhang et al., 2023), a lightweight, high-performance scene text recognizer.

2.4 Segment Anything Model

Segment Anything Model (SAM) (Kirillov et al., 2023) is the pioneering framework for zero-shot or visually prompted image segmentation with superior performance through pretraining on a massive segmentation dataset. Large-scale pretraining enables SAM to capture a generalized notion of visual entity and thus segment a broad variety of objects. It allows either segmentation of all visual entities in the entire image or interactive segmentation with a focus specified by extra visual prompts with points or bounding boxes. We also use the latest version, SAM 2 (Ravi et al., 2024), for an ablation study.

2.5 Open-Vocabulary Object Detection

While traditional object detection models are limited to a closed set of annotated categories, open-vocabulary object detection that identifies novel objects has received increasing attention, benefitting from vision-language pretraining. OWLv2 (Minderer et al., 2023) takes a step further to scaling up with self-training, resulting in state-of-the-art performance for open-vocabulary detection. We find such a model still incapable of detecting specific categories of objects presented in styles that are rare in natural images, such as depictions of zooplankton in diagrams. However, it's surprisingly useful for detecting more general and common objects even without fine-tuning. More specifically, we use OWLv2 to detect arrows in diagrams. The predicted bounding boxes from OWLv2 are used as visual prompts to SAM for arrow segments. Such a detect-and-segment pipeline is inspired by GroundingDINO (Liu et al., 2023).

2.6 Edge and Line Segment Detection

Edge detection is a classic computer vision task that identifies and locates sharp discontinuities of brightness and color intensity in an image. Such low-level visual features are useful since they often correspond to object boundaries or specific textures. The classic Canny edge detector (Canny 1986) is simple and powerful but is outperformed by neural-network-based methods on noisy, natural images. TEED (Soria Poma et al., 2023) reaches state-of-the-art edge detection performance via a lightweight CNN with only 58K parameters. We use TEED for edge detection as a starting point for building polylines that trace lines and curves in the diagrams. The edge

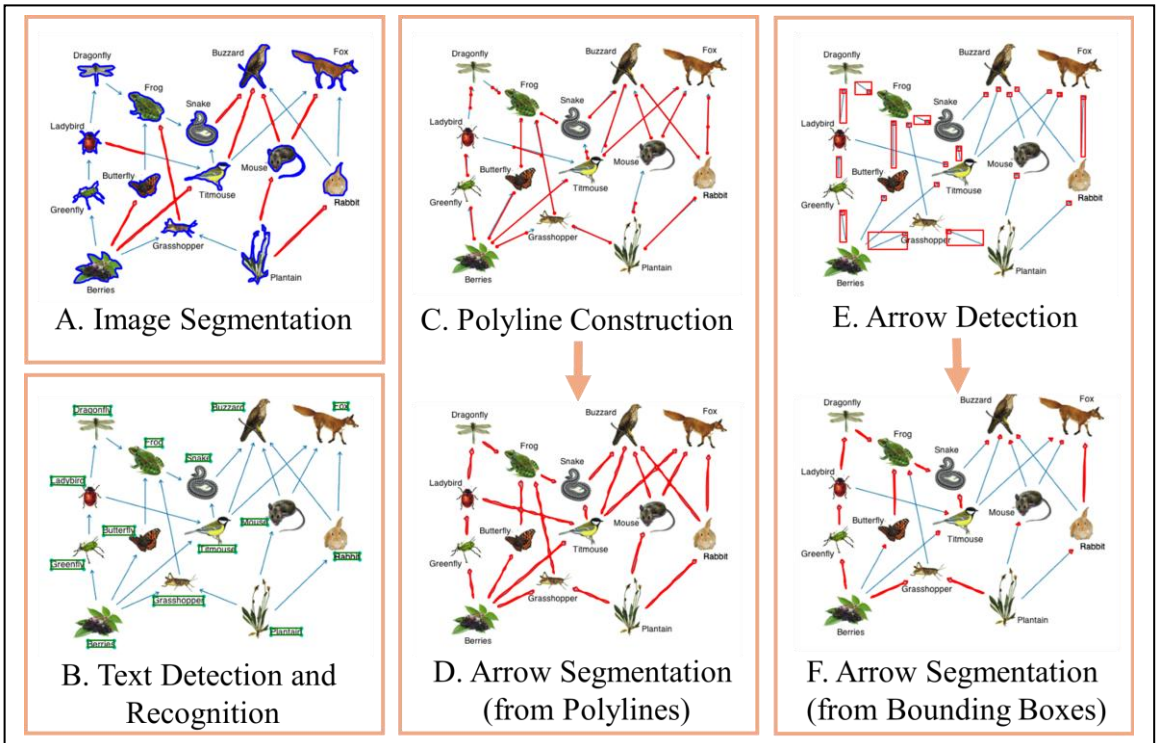


Figure 2: Visual elements produced by HVE from food web diagrams are overlayed on top of the original images. Those related to arrows are in red, including polylines, detected arrow bounding boxes, and arrow segments. The visual object segments are in blue. The detected text boxes are in green.

mask of the entire image is processed through a Probabilistic Hough Transform (Galamhos et al., 1999) algorithm for a set of individual line segments, which are combined into polylines. We also use the LSD algorithm for finding line segments (Gioi et al., 2012).

3. Heuristic-based Visual Ensemble

Our novel approach, HVE, uses CogSketch, a sketch understanding system designed to model human-like visual reasoning and grounded in cognitive science theories suggesting that analogy is central to intelligence. The process of visual understanding within this framework can be roughly conceptualized as three levels. At the first level (Level 1), basic visual elements are extracted from an image. At the second level (Level 2), these elements are analyzed through spatial heuristics (e.g., proximity, containment) to determine their relationships. Finally, at the third level (Level 3), this structured representation supports higher-level reasoning, such as problem-solving.

CogSketch has a strong track record of success in modeling high-level (Level 3) cognitive tasks. For example, it solves Raven's Progressive Matrices problems by using analogical reasoning to identify relational patterns across matrix rows, inferring the missing image in a way that matches adult human performance, as well as simpler geometric analogies and an oddity task (Forbus & Lovett, 2021). In modeling spatial ability, it has simulated the cognitive processes involved in mental rotation tasks by transforming 2D shapes to match a target orientation, and in

paper-folding puzzles by mentally folding a 2D pattern into a 3D cube to determine how the edges align (Lovett & Forbus, 2013). This high-level reasoning capability is built upon a flexible foundation for perceptual processing (Levels 1 and 2). For instance, in a hybrid model for visual relation detection (Chen & Forbus, 2021), the system used deep learning models to extract object bounding boxes and masks as its basic visual elements (Level 1). CogSketch then processed this information to compute qualitative spatial representations encoding pose, category, and spatial relationships (e.g., topology via RCC8, relative position, and size) between object pairs, which were used to classify the visual relation between them (Level 2). Similarly, for sketch recognition, CogSketch employed part-based hierarchical analogical learning. It first generated a decomposition tree from a sketch's digital ink, segmenting the object into a hierarchy of constituent parts like edges and closed edge-cycles (Level 1). It then constructed multi-level qualitative representations of these parts, describing their geometric properties and their spatial arrangements at different levels of detail, from coarse-grained contours to finer interior features (Level 2). This hierarchical representation then supported a data-efficient analogical learning process for classification.

This work is an effort to extend CogSektch for understanding diagrams, such as food webs. Diagrams pose an additional challenge, as they combine various types of visual elements, including text, illustrations, lines, and symbols, within a single image. Our long-term goal is a general-purpose approach, capable of interpreting a wide variety of diagrams. A key step towards this goal is creating a general recipe for Level 1 processing—the extraction of basic visual elements—that works across different diagrammatic genres and styles, which provides the raw material for higher-level processing and reasoning. The use of general-purose open-ended vision models in HVE (e.g., SAM, OWLv2, and UNITS) for detecting and segmenting basic visual elements naturally supports this goal, as these models are agnostic to domains, flexible for visual variations and and can incorporate top-down guidance from higher-level constraints. However, they often are not sufficient to cover all visual elements necessary for a complete diagram understanding, so we also settled by combining other lower-level computer vision processing, which are relatively more brittle and inflexible because of their heuristic nature. This means that our current HVE still partilly relies on experts for additional engineering to extract basic visual elements for each genres of diagrams, but it’s a step toward a more automatic process.

3.1 Diagram Understanding

We analyze two genres of diagrams here as a first step towards a comprehensive model. One genre includes diagrams consisting of discrete entities connected via visual elements such as arrows, essentially forms of discrete graphs, such as food webs, life cycle diagrams, and concept maps. The other genre includes all kinds of data-representing charts, such as line charts, pie charts, and bar charts. We use food web diagrams and bar charts in our experiments. There are other types of diagram out of the scope of this work, such as structural diagrams where the entities representing parts have more complex depictions (Lockwood et al., 2008).

We focus on the task of extracting consumption relations (Level 2) from food web diagrams. The goal is to extract all consumption relations, e.g., (consumedBy <Prey> <Predator>) or <prey>-consumedby-<predator>, depending on the formalism. Given the ubiquity of

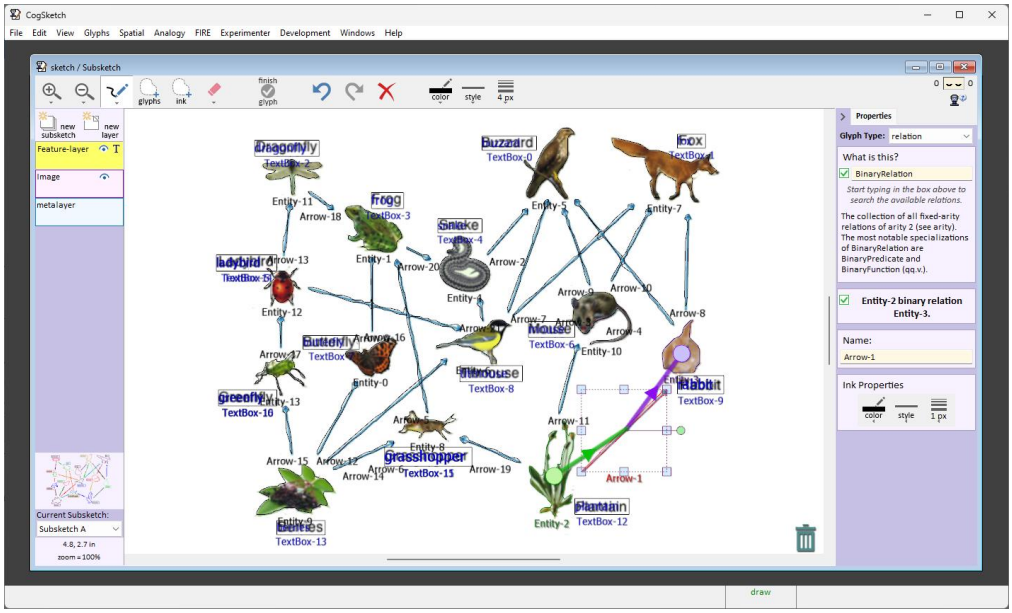


Figure 3: Final glyph representations in CogSketch UI. The arrow glyph clicked at the bottom right demonstrates the knowledge that Arrow-1 connects from Entity-2 (plantain) to Entity-3 (rabbit).

consumedBy relations in food web diagrams, we often simplify here to (<prey>, <predator>). The underlying structured information can be grounded on visual segments of three types of entities: objects, relations, and text boxes. We follow the convention in CogSketch and describe these three types of grounding visual segments as three glyph types.

In food webs like Figure 1 for example, there are often visual depictions of a type of creature (e.g. snake) which are labeled (e.g. "Snake"), and arrows denoting consumption relationships (e.g. snakes eat frogs and are eaten in turn by buzzards). The three types of glyphs correspond to objects (entities), arrows (relations), and text labels (annotations) in food web diagrams. In some food web diagrams, the illustrations are left out and only the labels are used. Understanding the consumption relationships requires first segmenting the image into entities, detecting text labels and arrows (Level 1), and finally determining what the arrows are connecting and hence the relationship (Level 2). The key intermediate visual representations are shown in Figure 2. The final output of three types of glyphs (objects illustrations, arrows, and text labels) in Figure 3 in CogSketch UI come from different methods:

- Objects illustrations: SAM (Kirillov et al., 2023) produces high-quality segmentation of objects. It often also captures text labels and arrows as objects, which are either masked by previous processing or filtered out later in the process.
- Arrows (general): OWLv2 (Minderer et al., 2023) detects arrows with "arrow" as the text prompt. We found it surprisingly good at detecting arrows of different styles, but it often overlooks thin arrows, only a few pixels in width, which are quite common in these diagrams. OWLv2 outputs the bounding boxes of detected arrow instances, which are provided as visual prompts to SAM for arrow segments.
- Arrows (fine-grained): Detection of thin arrows involves fine-grained visual features, using a separate process starting with low-level edges. TEED (Soria Poma et al., 2023) produces a

map for the edges of all visual contents in the diagrams, which is converted to a binary mask. We apply Probabilistic Hough Transform (Galamhos et al., 1999) on the edge mask for likely occurrences of line segments. This is more sensitive to thin arrow bodies than OWLv2. The line segments are combined into polylines that could be potential arrow shafts, which can guide SAM for segmenting the arrows.

- Text labels: UNITS (Kil et al, 2023) detects all occurrences of text in the diagram, providing both the bounding boxes and the recognized text strings.

With these intermediate visual representations, two types of relations are analyzed: the attachment of text labels to object glyphs and the reference of an arrow to the source and target object glyphs it connects. They are grounded on the glyphs and stored as knowledge in CogSketch (Figure 3). These visual relations ground and support the prediction of conceptual relations.

For bar charts, we target extracting the underlying data (Level 3) on top of analyzing the relations between elements like bars, guide ticks and labels (Level 2). We use similar visual representations in bar chart understanding that correspond to entity and annotation glyphs. Visual elements like bars, guide lines and tick marks are represented as entity glyphs, detected by area growth and line segment detection algorithms. Labels along each axis are detected with UNITS and represented as annotation glyphs, which are involved in attachment relations with bars.

For both diagrams, our general strategy is to focuses on one type of visual element at a time, beginning with components that more reliably generate an element type (e.g., text detection) before those that produce more ambiguous and noisy outputs. Once one set of visual elements is processed, the corresponding segments in the image are masked out to reduce distraction in the later processing of other visual elements. Then, we ubiquitously use various spatial and geometric heuristics for processing visual representations produced by component of the ensemble and analyzing their relations at different levels.

The visual heuristics are of course imperfect. Coming to a final interpretation for diagrams with more complex and ambiguous layouts may require richer semantics, including domain-specific information. As an attempt in this direction, we use a VLM for top-down filtering, but more importantly, the intermediate representations HVE constructs support such richer semantics, thereby providing both reasonable results on most diagrams and a foundation for improvements.

3.2 Processing and Analysis Procedure for Food Web Diagrams

1. Text Labels (Figure 2B):
    a. Extraction: Text is detected and recognized using the UNITS model.
    b. Post-processing heuristics: Geometrically proximal and aligned text boxes are merged into single entities. This heuristic ensures that multi-word labels are treated as a single unit.
2. General Arrows (Figure 2E & 2F):
    a. Extraction: The OWLv2 open-vocabulary model detects bounding boxes for arrows, which are then used as prompts for SAM to produce high-quality arrow segments.
    b. Post-processing heuristics: Processed in step 5 along with arrows from other sources.
3. Entity Illustrations (Figure 2A):

   a. Extraction: The Segment Anything Model (SAM) generates an initial set of all visual segments in the diagram.
   b. Post-processing heuristics: First, segments that are excessively large or small are filtered out. Next, overlapping segments are deduplicated based on an Intersection-over-Union (IoU) threshold, and segments already identified as text or arrows are removed. Finally, remaining segments are classified as either entities or potential arrow-like structures by analyzing the uniformity of their width.
4. Thin Arrows (Figure 2C & 2D):
   a. Extraction: For thin arrows missed by OWLv2, a multi-step process is used. First, TEED produces a precise edge map, on which a Probabilistic Hough Transform detects straight line segments.
   b. Post-processing heuristics: A tracing heuristic constructs polylines (potential arrow shafts) from these segments by prioritizing and connecting long, collinear lines. The points on these polylines then prompt SAM to segment the thin arrow.
5. Integration and Relationship Inference:
   a. Processing heuristics: All arrow candidates from the general and thin-arrow steps are combined. They are validated by analyzing width uniformity, and their directionality (head vs. tail) is determined using Principal Component Analysis (PCA) on the segment's points.
   b. Final Analysis: Heuristics establish the final relationships. An arrow-to-object association is determined by calculating proximity from the arrow's head and tail to nearby entities. An entity-to-text association links a non-textual visual entity to its closest text box, effectively connecting illustrations to their labels.

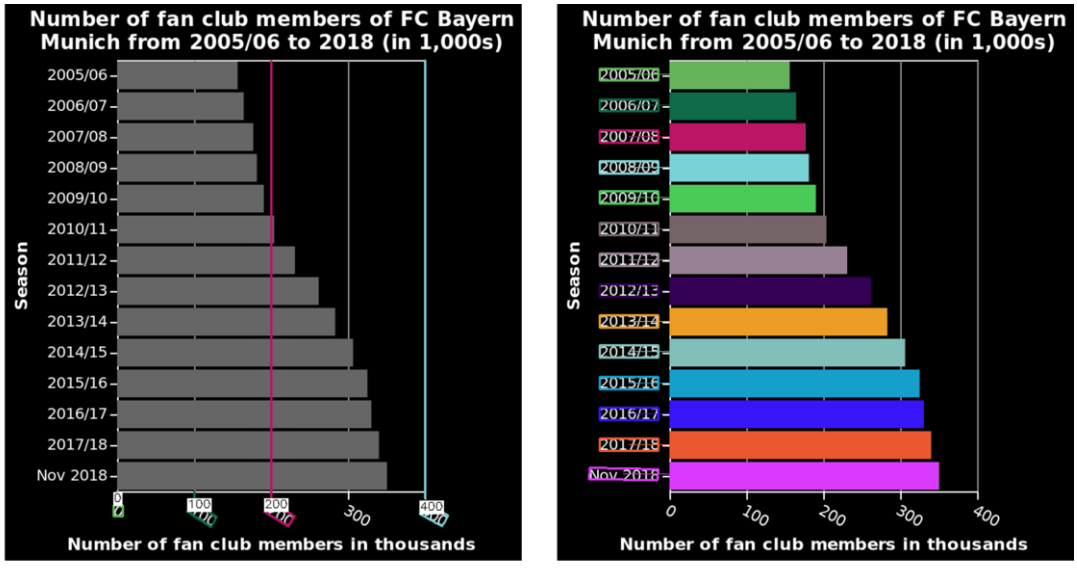
3.3 Processing and Analysis Procedure for Bar Charts

1. Text Labels & Tick Identification:
   a. Extraction: Text is detected and recognized using the UNITS model.
   b. Post-processing heuristics: A chaining heuristic groups individual text boxes into complete labels based on proximity and collinearity. The results are provided to a VLM (Qwen2.5-7B-Instruct) for filtering and classifying tick labels along both axes.
2. Bars:
   a. Extraction: The image is first partitioned into regions of uniform color to generate initial candidates for chart elements.
   b. Post-processing heuristics: These regions are filtered for bar candidates using shape analysis heuristics that favor high aspect ratios and solidity. The final set of bars is identified by finding the largest group of candidates that share a consistent thickness and alignment.
3. Axes, Ticks, and Guide Lines:
   a. Extraction: Non-bar-like uniform color regions are first analyzed to identify complex, multi-part line structures such as L-shaped axes, using a thinness

constraint (area-to-perimeter ratio) to filter for line-like shapes. To decompose these complex structures, the relevant regions are reduced to a one-pixel-wide topological skeleton, from which straight horizontal and vertical line segments are then extracted.

b. Post-processing heuristics: Raw line segments are refined by merging overlapping fragments and connecting collinear segments with small gaps. Numeric axis ticks are then identified by associating line segments perpendicular to the bar orientation with the nearest numeric tick labels (Figure 4A). This process assumes a constraint of uniform spatial offset between the line segments and the labels they point to. A search is performed to find the offset that maximizes the number of aligned lines while minimizing their positional error.

4. Integration and Value Extraction:
   a. Processing heuristics: Bar-to-tick associations (Figure 4B) for categorical labels are handled as a one-to-one assignment problem using the Hungarian algorithm. The cost matrix for this assignment is based on the minimum distance between bars and tick labels. The results are then validated with an additional constraint: the relative spacing between the assigned bars must map to the relative spacing of their corresponding ticks along the axis.
   b. Final Analysis: A scale function is generated by running a linear regression on the positions of numeric axis ticks and their corresponding labels, creating a pixel-to-data-value transformation. The final numerical value of each bar is calculated by applying the derived scale function to the pixel coordinate of the bar's value-determining edge.



A. Numeric Axis Ticks to Tick Labels B. Bars to Tick Labels

Figure 4: Visualization of visual elements assignments in HVE for bar charts. Axis Ticks (A) and Bars (B) are color-coded to illustrate their assignments with the text labels in the matched colors.

4. Experiment

4.1 Datasets

The VisText dataset (Tang et al., 2023) contains over 8,000 chart images with various orientations, color schemes, and uses of chart elements. More than 4000 of them are bar charts. Uniquely, all VisText examples include numeric value labels on the bars, which makes accurate data extraction more challenging as it requires comprehensive reasoning about the chart's components. Furthermore, its annotations include unstructured descriptions of the underlying data—a feature often missing from other chart-understanding datasets that typically provide answers only to a predefined set of questions. Structured ground truths are parsed with the assistance of a VLM (Qwen2.5-7B-Instruct).

The AI2 Foodweb dataset (Krishnamurthy et al., 2016) has 490 examples of food web diagrams. While the dataset was created for question-answering about the diagram, each diagram example includes annotations of the food chains in the image. The annotation are strings that contain multiple chains of entity labels connected with "->" symbol (e.g., "sand lance -> kittiwake -> fox"). We decomposed these chain annotations into ordered tuples representing consumption relation pairs, where each pair contains one string for the predator and one string for the prey (e.g., ("sand lance", "kittiwake") and ("kittiwake", "fox")).

4.2 Evaluation

For the VisText dataset, we evaluate the quality of extracted bar data by assessing the predicted labels and their corresponding values against ground truth. To account for minor variations, label matching is performed using a flexible string comparison based on Levenshtein edit distance; a match is accepted if the distance is less than a predefined matching ratio (0.3) of the shorter string's length. For matched labels, the accuracy of the extracted numerical values is calculated as the minimum of the ratio of predicted to ground truth value and vice versa. The evaluation focuses on three metrics: Overall Accuracy, the average accuracy across all ground truth labels, where unmatched labels counted as zero; Numeric Precision, measured as the average accuracy considering only successfully matched label-value pairs; and Label Recall, represented by the ratio of successfully matched labels to the total number of ground truth labels.

For the Foodweb dataset, we measure the quality of extracted consumption relations through average precision, recall, and the corresponding F1 score. A consumption relation match requires both predator and prey strings to match the gold annotation, using the same flexible string matching method to tolerate minor variations with a matching ratio of 0.3.

We conducted more comprehensive investigation on Foodweb Dataset. We first evaluate HVE without a VLM in the ensemble along with VLMs as standalone methods. For VLMs, we use straightforward prompts for extracting consumption relations in the food web diagrams. To study the synergy of using VLMs in HVE in an additional experiment, we used prompts of similar style while also including the reference relations extracted from bottom-up analysis in HVE as part of the textual context, which turned out to be beneficial with each of the VLMs we tested. Moreover, two more modifications were explored experimentally to narrow down the impact of including a VLM in HVE compared to using the same VLM as a standalone model.

One question is whether recognized text alone could be the main contributor to VLMs' improvement with HVE, since using OCR with VLMs has been shown to increases the performance in other visual tasks (Chen et al., 2022). Therefore, we include an additional test

Table 1: Results on bar data extraction.

Method	Overall Accuracy	Numeric Precision	Label Recall
HVE	54.1%	78.8%	65.6%
Qwen2.5-VL-7B	81.8%	86.1%	94.6%
Qwen2.5-VL-32B	77.0%	82.7%	92.4%
Qwen2.5-VL-72B	82.0%	86.0%	95.0%
Gemini2.5-flash	83.4%	87.4%	95.2%

case for all VLMs that only includes the recognized text labels of UNITS in the prompt. Another interesting question is how much do VLMs still rely on visual information when a lot of relational information is extracted by HVE and provided through textual input. VLMs have been shown to achieve impressive accuracy on image-based QA datasets even when the image is not provided (Chen et al., 2024). Therefore, in another test case, we provide the same VLMs with only textual prompt that includes both recognized text labels by UNITS and extracted relations by HVE.

4.3 Result

The results for bar data extraction are shown in Table 1, and results for consumption relations extraction are shown in Table 2. The key observations, mostly on the latter, are the following:

- Although falling short on bar charts, HVE achieved competitive performance on food web diagrams compared to standalone VLMs with 10 times fewer parameters in its ensemble.
- Combining HVE improves all VLMs we tested. Also, the positive effect of combining HVE is beyond simple augment with text recognition represented by combining UNITS outputs in VLMs. All VLMs had better scores with HVE than with just UNITS.
- The synergy of HVE and VLMs still depends on the visual information instead of solely based the knowledge VLMs have learned in the language space. From Table 2, we can see that removing the image leads to worse results in the most cases.

Moreover, the common pattern of the HVE integration is that it only slightly reduces the precision but substantially increases the recall. In the case where a standalone VLM has a much

Table 2: Results for consumption-relation extraction. We report the standalone HVE and directly prompted VLM baselines (*Base*), plus performance changes for three VLM variants: *+HVE*, *+UNITS*, and *-Image*. *+HVE* adds HVE-extracted relations to the VLM prompt, capturing HVE-VLM synergy. To analyze the synergy, *+UNITS* adds only the text recognized by UNITS. *-Image* includes both recognized text and extracted relations in the prompt but removes the image from the input.

Method	Size	Precision	Recall	F1
		Base/-Image/+UNITS/+HVE	Base/-Image/+UNITS/+HVE	Base/-Image/+UNITS/+HVE
HVE	<1B	57.8%	54.4%	56.0%
LLaVA-CoT	11B	62.5% / -11.7% / -2.1% / -8.7%	47.0% / +9.0% / +1.1% / +11.5%	53.7% / -0.4% / -0.2% / +2.4%
CogVLM 2	19B	58.6% / -12.3% / -6.0% / -2.2%	34.9% / +6.4% / -2.7% / +17.2%	43.7% / -0.1% / -3.7% / +10.4%
LLaVA-Next	34B	36.5% / +20.6%/ +9.7% /+10.3%	19.2% / +33.2%/+15.0%/ +41.1%	25.2% /+29.4%/+14.1%/ +27.5%
Qwen2-VL-72B	72B	75.4% / -14.4% / -1.7% / -0.7%	51.7% / +5.4% / +2.4% / +17.7%	61.4% / -2.4% / +1.0% / +10.5%
Gemini-1.5-flash	-	77.9% / -10.8% / +1.5% / -2.9%	65.9% / -7.8% / +3.7% / +10.0%	71.4% / -9.1% / +2.8% / +4.0%
Gemini-1.5-pro	-	74.8% / -16.5% / +0.8% / -1.1%	68.6% / -1.5% / +2.1% / +7.4%	71.6% / -9.2% / +1.5% / +3.3%
Gemini-2.0-flash	-	80.9% / -20.8% / +0.6% / -2.7%	72.8% / -10.1% / -0.8% / +3.6%	76.7% / -15.3%/ -0.2% / +0.6%
GPT-4o-mini	-	67.7% / -1.4% / -3.9% / -0.8%	41.5% / +11.0%/ +2.4%/ +24.8%	51.4% / +7.2% / +0.6% / +15.2%
GPT-4o	-	74.2% / -5.3% / +1.9% / +7.8%	60.5% / -6.1% / +2.9% / +10.1%	66.7% / -5.9% / +2.5% / +9.2%

Table 3: Changes of extracted true positive when combine HVE

Base VLM	Lost (avg)	Added (avg)	New (avg)	New (max)
LLaVA-CoT	1.66	3.29	0.48	19
CogVLM 2	1.57	3.39	0.32	7
LLaVA-Next	0.51	5.33	0.53	6
Qwen2-VL-72B	1.36	3.38	0.81	10
Gemini-1.5-flash	0.98	2.23	0.67	10
Gemini-1.5-pro	1.26	1.93	0.56	12
Gemini-2.0-flash	1.01	1.46	0.47	8
GPT-4o-mini	1.06	4.08	0.73	8
GPT-4o	1.38	2.47	0.8	12

higher precision than HVE, VLMs are able to apply the advantage of higher-precision inference on the additional reference relations from HVE. We further investigate the changes in recall by looking at true positives for HVE (TP_{HVE}), standalone VLMs (TP_{VLM}) and VLM-HVE combinations ($TP_{VLM+HVE}$). $TP_{HVE} \cup TP_{VLM}$ approximates the set of expected true relations if VLM can add all true relations from HVE. We divide the change into three types: relations in TP_{VLM} but lost in $TP_{VLM+HVE}$, relations not in TP_{VLM} but added from TP_{HVE} to $TP_{VLM+HVE}$, and new relations in $TP_{VLM+HVE}$ that are outside $TP_{HVE} \cup TP_{VLM}$. We aggregate the counts of these three types of relation changes and compute the average across all diagrams in Table 3. For new relations, we also include the maximum across diagrams.

As expected, the added true relations that come from TP_{VLM} are considerably more than the lost ones in general. The new true relations that are outside $TP_{HVE} \cup TP_{VLM}$ also contribute to the recall improvements, even though less than the number of lost ones in average. However, it's interesting to note that VLM-HVE integrations are able to capture many more new relations in the best case, ranging from 7-19 new true relations for different VLMs used with HVE. This implies that the reference relations from HVE could also serve as few-shot examples for in-context-learning effects (Brown et al., 2020) on these diagrams.

5. Ablation Study

The visual processing pipeline can be divided into four components that handle distinct visual elements: object illustrations, general arrows, fine-grained arrows, and text labels. The text detection model is crucial for entity label quality, which directly impacts the accuracy of extracted consumption relations. The segmentation model is equally important since all three other components rely on it to produce segments for visual elements.

HVE's modular design allows component substitution with functionally equivalent alternatives. We investigate how different segmentation and text detection models affect HVE's performance, though we limit our scope rather than exhaustively testing all available models for each computer vision subtask. We also assess the impact of removing either the general or fine-grained arrow processing components. In each test case, we modify only one component, keeping the rest the same as the main HVE configuration used in Section 5. We run modified HVE through all food web diagrams with the same metrics for consumption relation extraction.

5.1 Ablation of Text Detection and Recognition

Table 4: Ablation of text detection models

Text Models	Threshold	Text Labels		Consumption Relation	
		Precision	Recall	Precision	Recall
UNIT	0.0	81.5%	76.8%	54.2%	51.2%
	0.1	83.4%	78.6%	56.0%	52.8%
	0.2	84.3%	79.4%	57.0%	53.7%
	0.3	85.0%	80.1%	57.8%	54.4%
CRAFT +LPV	0.0	63.0%	60.0%	35.3%	33.5%
	0.1	75.5%	71.9%	46.6%	43.9%
	0.2	84.8%	80.8%	56.3%	53.1%
	0.3	86.9%	82.7%	58.9%	55.7%

We evaluated a two-stage approach combining CRAFT for text detection and LPV for text recognition within detected boxes. We also measured precision and recall between predicted and gold entity labels, using multiple edit distance ratio thresholds, which turned out to strongly correlate with performance of consumption relation extraction. Table 4 shows the results.

5.2 Ablation of Segmentation Model

We tested HVE with different size of SAM models and also one size of newer SAM 2. The results on consumption relation extraction are shown in Table 5. While SAM2.1-L unexpectedly underperformed compared to other SAM variants in precision and recall, our results confirm HVE's robustness across different segmentation models.

5.3 Ablation of Segmentation Model

The flexibility of HVE enables better coverage through separate components to handle general and fine-grained arrows. We conducted ablation experiments by removing components individually to evaluate their contributions to consumption relations extraction with results shown in Table 5. HVE with only general arrow detection (OWLv2) slightly decreased recall. Using only fine-grained arrow detection yields lower precision and recall, which is expected since it focuses on thin arrows through spatial and geometric analysis. This confirms the value of the dual-component approach.

6. Conclusions and Future Work

We explore HVE, a human-like approach that first recognizes visual elements and then uses intuitive heuristics about space and geometry to analyze conceptual and relational information, which is also grounded on segments of the visual elements. We show that HVE can do bar chart

Table 5: Ablation of segmentation models (left) and arrow detection components (right)

Segmentation Models	Precision	Recall	F1	Arrow Detection	Precision	Recall	F1
SAM-B	56.2%	52.8%	54.5% (-1.5%)	Both	57.8%	54.4%	56.0%
SAM-L	57.8%	54.4%	56.0%	Only General	58.7%	48.9%	53.4% (-2.6%)
SAM-H	58.3%	53.5%	55.8% (-0.2%)	Only Fine-grained	48.9%	32.6%	39.1% (-16.9%)
SAM2.1-L	54.8%	50.7%	52.7% (-3.3%)				

data extraction and is especially effeicient on food web consumption relation extraction, with combined lightweight components of less than 1B parameters, and achieves competitive performance as VLMs more than 10 times larger. An additional ablation study was also conducted to learn about the effect of each ensemble component.

Besides being useful as a standalone method, our experiments also find the unexpected synergy of HVE and VLM through a simple, low-effort integration by including the extraction outputs of HVE in the textual prompt of VLMs. To confirm the benefit of the VLM-HVE integration, we introduce additional comparisons and show that the improvements on VLMs brought by HVE integration are beyond the simple augmentation with text recognition. We further investigate the mechanism this synergy through the lens of true positive relations predicted by VLM, HVE, and VLM-HVE integrations.

The range of depictions used in diagrams is broad, and we have not captured all the ways information is depicted in food webs, let alone other kinds of diagrams not covered in this work. Current HVE has significant limitations at each levels. It still depends on relatively specialized lower-level processing for extracting and processing some basic visual elements (Level 1, e.g., extracting thin arrows, bars and guide ticks) in the ensemble due to the weakness of open-ended vision models on recognizing them. At higher levels, HVE seems to seems to lack the rich semantics and domain-specific information as top-down constraints for complete analysis and interpretation of relations between visual elements in complex layouts.

One natural next step is to explore other general approaches to close the gap between the capability of open-ended vision models and the broad range of basic visual elements in various genres of diagrams. Another area for exploration is a more systematic design for top-down constraints potential with VLMs that can more naturally integrated within HVE. Beyond the two types of tasks in this work, we also plan to explore ways of analogical reasoning, which is at the core of CogSketch, with the intermediate respesentations made avaible with HVE.

Acknowledgements

This research was supported by the Office of Naval Research.

References

Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character region awareness for text detection. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9357-9366).

Brown, T. B., […] (2020). Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Bai, S., […] (2025). *Qwen2.5-VL Technical Report*. ArXiv, abs/2502.13923.

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*, 679-698.

Chen, K., & Forbus, K. (2021). Visual relation detection using hybrid analogical learning. In *Proceedings of AAAI 2021*.

Chen, K., Forbus, K., Srinivasan, B., Chhaya, N., & Usher, M. (2023). Sketch recognition via part-based hierarchical analogical learning. In *Proceedings of IJCAI 2023, Macao, China*.

Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., & Zhao, F. (2024). *Are we on the right way for evaluating large vision-language models?* ArXiv, abs/2403.20330.

Chen, X., […] (2022). *PaLI: A jointly-scaled multilingual language-image model*. ArXiv, abs/2209.06794.

Forbus, K., & Lovett, A. (2021). Same/different in visual reasoning. *Current Opinion in Behavioral Sciences, 37*, 63-68.

Forbus, K. D., Chang, M., McLure, M., & Usher, M. (2017). The cognitive science of sketch worksheets. *Topics in Cognitive Science*.

Forbus, K., Chen, K., Xu, W., & Usher, M. (2024). *Hybrid primal sketch: Combining analogy, qualitative representations and computer vision for scene understanding*. Advances in Cognitive Systems Conference, 2024. https://arxiv.org/abs/2407.04859

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2011). CogSketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, 1-19.

Galamhos, C., Kittler, J., & Matas, J. (1999). Progressive probabilistic Hough transform for line detection. In *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1* (pp. 554-560).

Gemini Team, Google. (2024). *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*. ArXiv, abs/2403.05530.

Gioi, R. G., Jakubowicz, J., Morel, J., & Randall, G. (2012). LSD: a line segment detector. *Image Process. Line, 2*, 35-55.

Kil, T. H., Kim, S., Seo, S., Kim, Y., & Kim, D. (2023). Towards unified scene text spotting based on sequence generation. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 15223-15232).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W., Dollár, P., & Girshick, R. B. (2023). Segment anything. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision* (pp. 3992-4003).

Krishnamurthy, J., Tafjord, O., & Kembhavi, A. (2016). Semantic parsing to probabilistic programs for situated question answering. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 160-170). Association for Computational Linguistics.

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., & Lee, Y. J. (2024). *LLaVA-NeXT: Improved reasoning, OCR, and world knowledge*.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2023). *Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection*. ArXiv, abs/2303.05499.

Lockwood, K., Lovett, A., Forbus, K., Dehghani, M., & Usher, J. (2008). A theory of depiction for sketches of physical systems. In *Proceedings of QR 2008*.

Lovett, A.M., & Forbus, K.D. (2013). Modeling Spatial Ability in Mental Rotation and Paper-Folding. *Cognitive Science, 35*.

Minderer, M., Gritsenko, A. A., & Houlsby, N. (2023). Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems, 36*.

OpenAI. (2024). *GPT-4o System Card*. ArXiv, abs/2410.21276.

Ravi, N., […] (2024). *SAM 2: Segment anything in images and videos*. ArXiv, abs/2408.00714.

Soria Poma, X., Li, Y., Rouhani, M., & Sappa, A. D. (2023). Tiny and efficient model for the edge detection generalization. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision Workshops* (pp. 1356-1365).

Tang, B. J., & Boggust, A. (2023). VisText: A benchmark for semantically rich chart captioning. *Annual Meeting of the Association for Computational Linguistics*.

Wang, J., Ming, Y., Shi, Z., Vineet, V., Wang, X., & Joshi, N. (2024b). *Is a picture worth a thousand words? Delving into spatial reasoning for vision language models*. ArXiv, abs/2406.14852.

Wang, P., […] (2024a). *Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution*. ArXiv, abs/2409.12191.

Wang, W., […] (2023). *CogVLM: Visual expert for pretrained language models*. ArXiv, abs/2311.03079.

Xu, G., Jin, P., Li, H., Song, Y., Sun, L., & Yuan, L. (2024). *LLaVA-CoT: Let vision language models reason step-by-step*. ArXiv, abs/2411.10440.

Zhang, B., Xie, H., Wang, Y., Xu, J., & Zhang, Y. (2023). Linguistic more: Taking a further step toward efficient and accurate scene text recognition. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (pp. 1704-1712).