

ET-SAM: Efficient Point Prompt Prediction in SAM for Unified Scene Text Detection and Layout Analysis

Xike Zhang^{1†}, Maoyuan Ye^{1†}, Juhua Liu^{1(✉)}, and Bo Du¹

School of Computer Science, National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China
{zhangxike, yemaoyuan, liujuhua, dubo}@whu.edu.cn

Abstract. Previous works based on Segment Anything Model (SAM) have achieved promising performance in unified scene text detection and layout analysis. However, the typical reliance on pixel-level text segmentation for sampling thousands of foreground points as prompts leads to *unsatisfied inference latency* and *limited data utilization*. To address above issues, we propose **ET-SAM**, an **E**fficient framework with two decoders for unified scene **T**ext detection and layout analysis based on **SAM**. Technically, we customize a lightweight point decoder that produces word heatmaps for achieving a few foreground points, thereby eliminating excessive point prompts and accelerating inference. Without the dependence on pixel-level segmentation, we further design a joint training strategy to leverage existing data with heterogeneous text-level annotations. Specifically, the datasets with multi-level, word-level only, and line-level only annotations are combined in parallel as a unified training set. For these datasets, we introduce three corresponding sets of learnable task prompts in both the point decoder and hierarchical mask decoder to mitigate discrepancies across datasets. Extensive experiments demonstrate that, compared to the previous SAM-based architecture, ET-SAM achieves about **3**× inference acceleration while obtaining competitive performance on HierText, and improves an average of 11.0% F-score on Total-Text, CTW1500, and ICDAR2015. The code and models are available at ET-SAM.

Keywords: Segment Anything Model · Unified Scene Text Detection · Layout Analysis · Point Prompt Prediction · Joint Training

1 Introduction

Visual text contained in images conveys rich and explicit semantic information which is pivotal for scene comprehension. Meanwhile, visual text encompasses a hierarchical structure ranging from word and text-line to full paragraph, with each granular level serving a distinct role for varying applications [23, 26, 27, 38,

†: Equal contribution. ✉: Corresponding author.

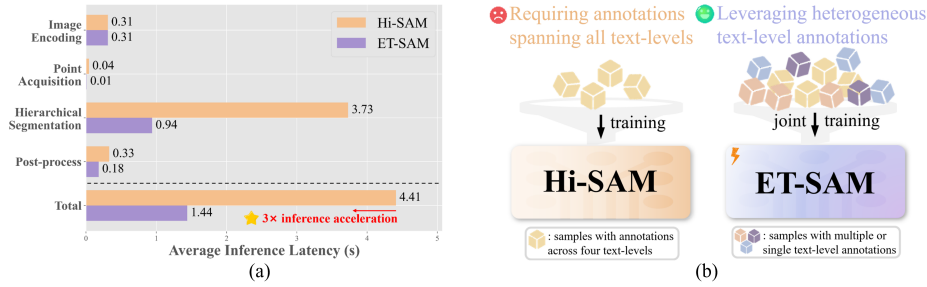


Fig. 1: (a) ET-SAM achieves 3× inference acceleration on HierText [27] via predicting sparse points as visual prompts, significantly reducing the latency at hierarchical segmentation and post-processing stages. (b) To facilitate data scalability, we design a joint training strategy to leverage samples with mixing text-level annotations.

41]. However, previous scene text detection methods [8, 20, 23, 26, 33, 39, 40, 44] typically focus on a specific text level with an individual suit of parameters, lacking a unified framework for simultaneously tackling all text levels. Consequently, the unified scene text detection and layout analysis benchmark [27] has been introduced to holistically evaluate the performance across word, text-line, and paragraph levels. This benchmark is accompanied by a baseline method, the Unified Detector (UD), which leverages the Detection Transformer (DETR) [3] paradigm. Each object query in UD is optimized to segment individual words within a text line, while a dedicated layout branch constructs an affinity matrix across queries to facilitate their clustering into paragraphs.

Subsequently, built upon Segment Anything Model (SAM) [17], Hi-SAM [38] innovatively integrates four-level text segmentation and enhances the accuracy via a dual-decoder framework: one segments pixel-level text for generating foreground point prompts, while the other segments text structures from words to paragraphs under the guide of point prompts. However, Hi-SAM exhibits limitations in terms of *inference latency* and *data utilization*. Concretely, Hi-SAM randomly samples thousands of foreground points from the pixel-level segmentation result as visual prompts, introducing excessive inference cost at the hierarchical mask decoding and post-processing stages. Moreover, Hi-SAM necessitates comprehensive annotations spanning all text granularities for every training instance, a requirement that is constrained by the scarcity of such densely labeled data in current datasets. Thus, the scarcity of suitable training data impedes scalability, thereby constraining applications in real-world scenarios.

To address the issues, we present ET-SAM, a more efficient and scalable framework (as shown in Fig. 1) for unified scene text detection and layout analysis. Technically, to reduce the point prompts and speed up the inference, we introduce a customized lightweight point decoder that generates word-level heatmaps, enabling the efficient localization of sparse foreground points characterized by peak activation values. Moreover, we devise a joint training strategy that enables the combination of datasets with heterogeneous text-level annotations to facilitate data scalability. To disambiguate diverse word densities and segmentation

targets while alleviating annotation inconsistencies, we further integrate distinct sets of learnable task prompts within the point decoder and mask decoder.

In summary, our main contributions are three-fold:

- We present ET-SAM, enabling efficient point prompt prediction and speeding up the inference of SAM-based architecture with a tailored point decoder.
- We devise a joint training strategy to harness existing datasets characterized by heterogeneous text-level annotations, thereby fostering the scaling of data.
- Extensive experiments demonstrate that ET-SAM not only delivers roughly **3×** inference acceleration over Hi-SAM with competitive performance on HierText but also secures a substantial **11.0%** gain in average F-score on Total-Text, CTW1500, and ICDAR2015.

2 Related Work

2.1 Scene Text Detection

Scene text detection has witnessed significant advancements in recent years, with most methodologies broadly classified into segmentation-based and regression-based paradigms. Segmentation-based approaches [1, 22, 23, 32, 33, 37] formulate text detection as a dense pixel-wise segmentation task. For example, PANet [33] introduces a pixel aggregation mechanism to associate pixels with distinct text kernels. DBNet [22] incorporates differentiable binarization to jointly predict threshold maps alongside binary segmentation masks. In comparison, regression-based methods [10, 11, 20, 21, 24, 26, 39, 41, 44, 46, 47] localize text regions via bounding box regression or contour curve fitting. For instance, ABCNet [24] and DeepSolo [41] employ Bezier curves to model the top and bottom boundaries of text instances, whereas FCENet [47] reconstructs contours using the inverse Fourier transform. Despite these strides, such methods typically operate at a singular text granularity with dedicated parameter sets, lacking a unified framework capable of addressing diverse text levels across words, text-lines, and paragraphs.

2.2 Unified Scene Text Detection and Layout Analysis

Long *et al.* [27] pioneered unified scene text detection and layout analysis by introducing the HierText dataset and the Unified Detector (UD). Built upon the DETR [3] paradigm, UD optimizes object queries for word and text-line segmentation while employing a layout branch to cluster them into paragraphs. Subsequently, Hi-SAM [38] advanced this field by integrating the Segment Anything Model [17] for hierarchical text segmentation, enabling fine-grained segmentation from pixel-level strokes up to paragraphs. Nevertheless, Hi-SAM exhibits limitations regarding inference latency and data scalability. Specifically, its reliance on random point sampling incurs overheads, while requiring comprehensive multi-granularity annotations hinders scaling up training data. In

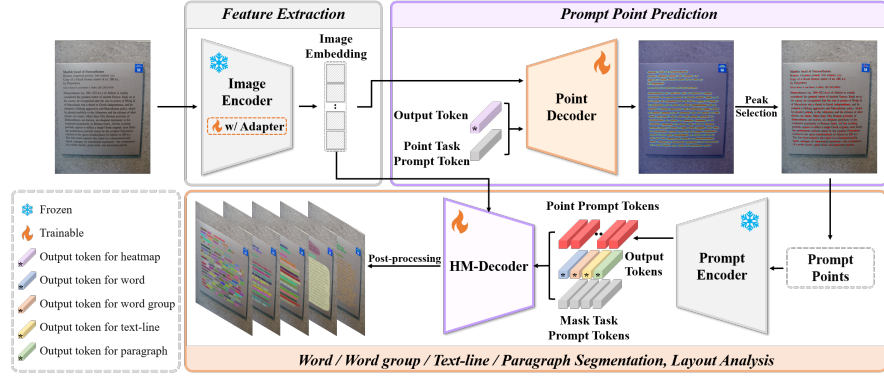


Fig. 2: Overview of ET-SAM framework. Extracted from the word heatmap produced by our point decoder, sparse points are used as visual prompts, thereby accelerating subsequent inference. Under the guide of point prompts and specific mask task prompts, HM-Decoder segments word, word group, text-line, and paragraph masks, which are finally used to achieve layout analysis via a union-find algorithm.

comparison, we provide a more efficient and scalable framework. This work introduces a customized lightweight point decoder to predict sparse foreground points for acceleration, alongside a joint training strategy which leverages heterogeneous annotation sources. Note that the structure modifications are fully compatible to SAM components, which can also be applied to other potential SAM-based methods.

2.3 Segment Anything Model

The Segment Anything Model (SAM) [17] has established itself as a cornerstone of image segmentation, exhibiting exceptional generalization and adaptability across a spectrum of domains, ranging from image matting [18], 3D segmentation [4], remote sensing [9, 31], medical image segmentation [42, 43], and video tracking [5, 36]. Building upon this foundation, SAM 2 [30] incorporates a streaming memory mechanism to enable unified segmentation and tracking for both static images and dynamic videos. Subsequently, SAM 3 [2] further consolidates object detection, segmentation, and tracking into a single framework driven by concept prompts. Following the recent advance [38] in unified scene text detection and layout analysis, this work presents ET-SAM, a specialized and efficient architecture which is also based on SAM.

3 Methodology

3.1 Overview of ET-SAM

The overall framework is depicted in Fig. 2. ET-SAM comprises four primary components: (1) an image encoder adapted from SAM with adapters, aligned

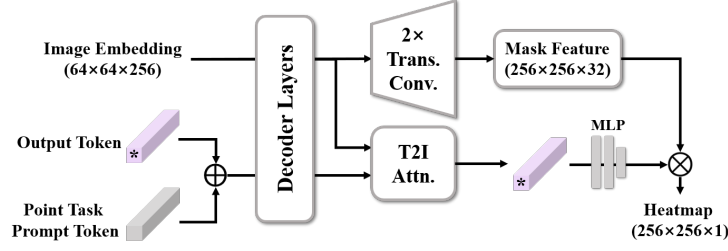


Fig. 3: Structure of Point Decoder. ‘Trans. Conv.’ and ‘T2I Attn.’ denote transposed convolution and token-to-image attention, respectively.

with the configuration of Hi-SAM [38]; (2) a streamlined, trainable point decoder derived from SAM’s mask decoder, following PseCo [13] paradigm; (3) a prompt encoder inherited directly from SAM; and (4) a hierarchical mask decoder (HM-Decoder), which is a specialized modification of SAM’s mask decoder. Specifically, the point decoder generates a word-centric heatmap from image embeddings, where word centers exhibit peak activations. The coordinates of local maxima are then extracted as sparse points which are subsequently encoded by the prompt encoder and fed into the HM-Decoder to produce masks across hierarchical text granularities. Leveraging these hierarchical masks, layout analysis is naturally derived as an auxiliary output, consistent with Hi-SAM paradigm. To orchestrate diverse segmentation tasks and alleviate inter-dataset discrepancies, both the point decoder and the HM-Decoder are augmented with learnable task prompts. In Sec. 3.2, we first detail the task prompts within the point decoder (Sec. 3.3) and HM-Decoder (Sec. 3.4), followed by descriptions of the joint training strategy (Sec. 3.5) and inference workflow (Sec. 3.6).

3.2 Task Prompt

Heterogeneity across datasets poses a challenge to model optimization. For instance, HierText [27] exhibits high word density and employs quadrilateral annotations for text lines, whereas CTW1500 [25] utilizes curved representations for text lines. Such disparities may degrade cross-dataset performance. To mitigate these discrepancies, in both the point decoder and the HM-Decoder, we incorporate learnable task prompts that are initialized with simple embeddings. These prompts serve as conditional signals: in the point decoder, they adapt to varying word densities, while in the HM-Decoder, they specify distinct segmentation objectives. Specifically, we define three task configurations ($N_{task} = 3$):

- *Task 0* targets multi-level datasets (*e.g.*, HierText), featured with dense word distributions and requiring simultaneous prediction of hierarchical masks;
- *Task 1* necessitates only word-level mask prediction;
- *Task 2* focuses on text-line level mask prediction.

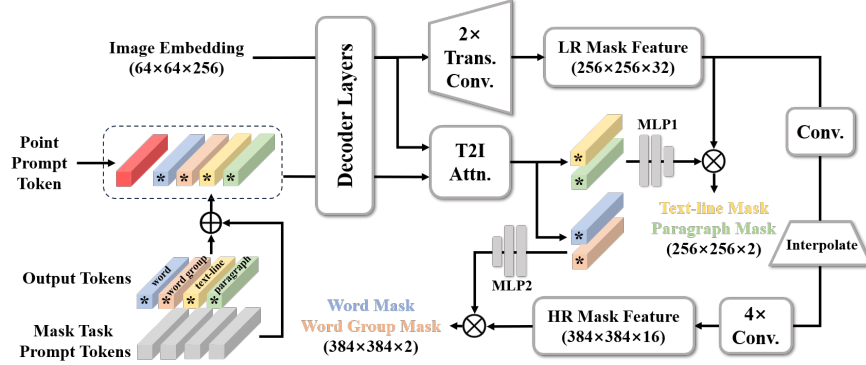


Fig. 5: Structure of HM-Decoder. Four output tokens charge in segmentation at word, word group, text-line, and paragraph level, under the guide of point prompts and learnable mask task prompt tokens. ‘Interpolate’ indicates interpolating the spatial resolution to 384×384 . The IoU prediction process is omitted here.

line, and a paragraph (along with predicting their respective IoU scores). The detailed structure is illustrated in Fig. 5.

Concretely, the output tokens $t_{h_out} \in \mathbb{R}^{1 \times 4 \times 256}$ and the mask task prompt tokens $t_{h_task} \in \mathbb{R}^{N_{task} \times 4 \times 256}$ are learnable embeddings, where N_{task} denotes the number of tasks while the dimension of 4 corresponds to the number of output masks. Given K point prompt tokens $t_{point} \in \mathbb{R}^{K \times N_p \times 256}$ encoded by the prompt encoder and a selected task i , we first compute the element-wise sum of t_{h_out} and the i -th task prompt token $t_{h_task}[i]$. The resulting tensor is subsequently broadcast to align with the number of point prompt tokens t_{point} , yielding an intermediate representation $t'_{h_out} \in \mathbb{R}^{K \times 4 \times 256}$. Then, t'_{h_out} is concatenated with t_{point} , forming $[t'_{h_out}; t_{point}] \in \mathbb{R}^{K \times (4+N_p) \times 256}$ which is fed into the HM-Decoder.

After the transposed convolution and token-to-image attention operations, the updated output tokens $\hat{t}_{h_out} \in \mathbb{R}^{K \times 4 \times 256}$ and low-resolution mask features $F_{lr} \in \mathbb{R}^{K \times 256 \times 256 \times 32}$ are produced. To generate the mask logits $M_{l,p} \in \mathbb{R}^{K \times 256 \times 256 \times 2}$ for text-line and paragraph hierarchies, the corresponding slice of output tokens $\hat{t}_{h_out}$ is projected via a MLP subsequently fused with the low-resolution mask features F_{lr} through a spatially point-wise product. Conversely, for the word and word group hierarchies, the corresponding sliced output tokens are processed by another MLP. These transformed output tokens are then combined with high-resolution features $F_{hr} \in \mathbb{R}^{K \times 384 \times 384 \times 16}$, which are derived from F_{lr} via interpolation and convolutional refinement, to produce the final word and word group mask logits $M_{w,wg} \in \mathbb{R}^{K \times 384 \times 384 \times 2}$. Consistent with the design of Hi-SAM [38], we generate word and word group masks at an elevated spatial resolution for better segmentation quality.

3.5 Joint Training

Acknowledging the scarcity of multi-level annotations beyond HierText [27], we stratify existing datasets into word-level, text-line-level, and multi-level categories. Our objective is to jointly leverage these datasets to achieve uniform representation learning across text hierarchies. However, naive joint training poses challenges: if individual batches are dominated by samples from a single dataset or text granularity, the optimization trajectory may fluctuate significantly between batches. To mitigate these issues, we first preprocess datasets within each category into a unified format, then randomly shuffle and aggregate them to construct three consolidated datasets: a multi-level set, a word-level set, and a text-line-level set. These sets are subsequently aligned in parallel to form a unified training pool, with its total length dictated by the largest set. During joint training, each batch is constructed to include one sample from each of the three categories. Within every epoch, the shorter datasets are iteratively resampled until they match the length of the unified dataset.

The trainable components of our framework comprise: (1) the adapters integrated within the image encoder, (2) the point decoder, and (3) the HM-Decoder. Specifically, the optimization of the point decoder is governed by the Mean Squared Error (MSE) loss, which quantifies the discrepancy between the predicted heatmaps $\hat{H}$ and the ground-truth targets H . This loss function is formulated as: $\mathcal{L}_{point} = \|H - \hat{H}\|_2^2$.

For optimizing HM-Decoder, a certain number of prompt points and their corresponding masks are first sampled. For HierText dataset, we randomly sample 10 text-lines from each image, obtain their corresponding word groups and paragraphs, and then randomly select one word from each word group. Next, we extract extreme points from the target heatmap and take their intersections with the word masks. Similarly, for word- and text-line-level datasets, we randomly sample 10 words or text-lines. We then extract extreme points from the heatmaps and take their intersections with the corresponding masks. Finally, for each mask, up to 2 points are randomly selected as prompt points, yielding at most 20 prompt points paired with ground-truth masks across four text granularities. These prompt points are fed into the prompt encoder to obtain point prompt tokens, guiding the segmentation of hierarchical masks. The losses for word, word group, text-line, and paragraph masks are all composed of Binary Cross-Entropy (BCE) loss, Dice loss [29], and IoU MSE loss, with an equal weighting ratio of 1:1:1. The overall loss function is defined as follows:

$$\mathcal{L} = 50 \times \mathcal{L}_{point} + \mathcal{L}_{word} + \mathcal{L}_{word_group} + \mathcal{L}_{line} + 0.5 \times \mathcal{L}_{para}. \quad (2)$$

Additionally, text-line datasets do not contain word-level ground truth, which conflicts with the word-centric nature of our heatmap generation process described in Sec. 3.3. To resolve this discrepancy, we first train the point decoder using multi-level and word-level datasets, then use the trained model to generate word heatmaps as pseudo labels. These heatmaps are further refined using available text-line annotations to eliminate errors, such as suppressing false positives.

3.6 Inference Process

For a given image and selected task, the point decoder initially generates a word-level heatmap. A 3×3 max-pooling operation is then applied to extract extrema from the heatmap, and points whose values exceed a predefined threshold are selected as prompt points. These points are fed into the prompt encoder in batches of 100 to obtain point prompt tokens, which guide the HM-Decoder to segment masks and predict IoU scores across four hierarchical levels.

For *task 0*, we first filter out text-lines with low IoU and apply Matrix NMS [34] to suppress overlapping text-lines. The corresponding word groups and paragraphs associated with the removed text-lines are also discarded. The IoU and NMS threshold are both set to 0.5 by default. Then, we compute the IoU matrix of the paragraph masks and use a union-find algorithm to realize layout analysis following previous works [27, 38]. Finally, the corresponding word groups and text lines are aggregated accordingly, completing the layout analysis [27].

For *tasks 1 and 2*, we first discard low-quality masks whose IoU scores are below 0.5, and then apply Matrix NMS with a threshold of 0.5 to eliminate overlapping masks, obtaining the final word or text-line masks.

4 Experiments

4.1 Datasets

HierText [27] comprises 8,281 training, 1,724 validation, and 1,634 test images. It is distinguished by its hierarchical annotations spanning word, text-line, and paragraph levels. Statistically, each image contains hundreds of words, exhibiting a distribution that is both sufficiently dense and relatively uniform.

Total-Text [6] contains 1,255 training images and 300 test images, featuring a large number of curved text instances and providing word-level annotations.

TextSeg [35] encompasses 2,646 training, 340 validation, and 1,038 test images. TextSeg offers both word-level and pixel-level annotations, covering a diverse array of scene and artistic text instances.

ICDAR2013 [15] contains 229 training and 233 test images, focusing primarily on horizontal scene text with word-level bounding box annotations. Compared to ICDAR2013, **ICDAR2015** [14] presents a more challenging scenario with 1,000 training and 500 test images. ICDAR2015 incorporates multi-oriented text instances, which are annotated with quadrilateral boxes.

CTW1500 [25] consists of 1,000 training and 500 test images, featuring the text-line level benchmark with annotations in arbitrary shapes.

4.2 Implementation Details

Joint Training. The majority parameters of ET-SAM are initialized from SAM before training. Specifically, the image encoder adopts SAM’s ViT-L [17]. The prompt encoder is identical to that of SAM, while the point decoder and HM-Decoder are partially initialized from SAM’s mask decoder. We use AdamW [28]

Table 1: Performance comparison on the test set of HierText.

Method	Word					Text-line					Layout Analysis				
	PQ	F	P	R	T	PQ	F	P	R	T	PQ	F	P	R	T
UD [27]	48.21	61.51	67.54	56.47	78.38	62.23	79.91	79.64	<u>80.19</u>	77.87	53.60	68.58	76.04	62.45	78.17
Hi-SAM-L [38]	<u>63.10</u>	<u>81.83</u>	87.22	<u>77.06</u>	<u>77.11</u>	<u>66.17</u>	84.85	90.66	79.74	<u>77.99</u>	57.61	74.49	80.43	<u>69.37</u>	77.34
Ours (Joint)	62.52	80.84	85.01	77.06	77.33	65.40	84.08	88.90	79.75	77.79	<u>57.74</u>	<u>74.51</u>	80.75	69.16	77.49
Ours (FT)	63.60	81.94	<u>85.81</u>	78.41	<u>77.61</u>	66.30	<u>84.82</u>	<u>89.17</u>	80.88	78.16	58.82	75.39	81.85	69.88	<u>78.02</u>

($\beta_1 = 0.9, \beta_2 = 0.999, weight_decay = 0.05$) as the optimizer with an initial learning rate of $1e^{-4}$. We train the model on 8 NVIDIA Tesla V100 (32GB) GPUs, with a batch size of 3 per GPU. We follow the data augmentations used in Hi-SAM [38]. Joint training runs for 120 epochs on a unified data pool including HierText, Total-Text, TextSeg, ICDAR2013, ICDAR2015, and CTW1500, with the learning rate divided by 10 at the 100 epochs.

Fine-tuning. We fine-tune the model after joint training on 4 NVIDIA Tesla V100 (32GB) GPUs with a total batch size of 8, using the same optimizer and data augmentation strategy, and an initial learning rate of $1e^{-4}$. On HierText, ET-SAM is trained for 50 epochs, with the learning rate decayed by a factor of 10 at the 35th epoch. On Total-Text, CTW1500, and ICDAR2015, it is trained for 80 epochs, with the learning rate divided by 10 after 60 epochs.

4.3 Experimental Results

The main experimental results are evaluated on the HierText [27], Total-Text [6], CTW1500 [25], and ICDAR2015 [14] test sets, considering both the joint training results and the fine-tuning results. For HierText, we follow previous works to use Panoptic Quality (PQ) [16], F-score (F), Precision (P), Recall (R), and Tightness (T) as metrics, while F-score, Precision, and Recall are used for other test sets.

Unified Scene Text Detection and Layout Analysis. Table 1 compares the comprehensive performance on the test set of HierText. Under the joint training protocol, ET-SAM exhibits a marginal deficit compared to Hi-SAM [38] at the word and text-line levels, yet surpasses it in layout analysis. After fine-tuning, ET-SAM achieves consistent gains across all hierarchies, with improvements of 1.08% PQ and 1.10% F-score at the word level, 0.90% PQ and 0.74% F-score at the text-line level, and 1.08% PQ and 0.88% F-score on layout analysis, resulting in overall superior performance compared to Hi-SAM.

Scene Text Detection on Specific Text Granularities. As shown in Tab. 2, we evaluate the text detection performance of different text granularities on Total-Text, CTW1500, and ICDAR2015 against state-of-the-art (SOTA) methods. In contrast to Hi-SAM, which cannot leverage mixed-granularity annotations during training, our proposed ET-SAM demonstrates substantial improvements across all three benchmarks following the joint training stage, yielding an average F-score gain of 10.7%. Upon dataset-specific fine-tuning, ET-SAM achieves F-scores of 87.0% on Total-Text, 87.9% on CTW1500, and 83.2% on ICDAR2015, culminating in a remarkable 11.0% average improvement over Hi-SAM. However, a performance disparity persists between our unified framework

Table 2: Performance comparison on Total-Text, CTW1500, and ICDAR15.

Method	Total-Text			CTW1500			ICDAR15		
	F	P	R	F	P	R	F	P	R
PANet [33]	85.0	89.3	81.0	83.7	86.4	81.2	82.9	84.0	81.9
DBNet++ [23]	86.0	88.9	83.2	85.3	87.9	82.8	87.3	90.9	83.9
PCR [7]	85.2	88.5	82.0	84.7	87.2	82.3	-	-	-
ABCNet v2 [26]	87.0	90.2	84.1	84.7	85.6	83.8	88.1	90.4	86.0
FCENet [47]	85.8	89.3	82.5	85.5	87.6	83.4	86.2	90.1	82.6
TESTR [45]	86.9	93.4	81.4	87.1	92.0	82.6	90.0	90.3	89.7
TextBPN++ [44]	88.5	91.8	85.3	85.5	87.3	83.8	-	-	-
DPTText-DETR [39]	89.0	91.8	86.4	88.8	91.7	86.2	-	-	-
DeepSolo [41]	90.0	92.9	87.4	88.9	93.2	85.0	90.1	92.4	87.9
SRFormer [1]	89.7	91.5	87.9	89.6	89.4	89.8	-	-	-
ESTextSpotter [12]	90.0	92.0	88.1	90.0	91.5	88.6	91.0	92.5	89.6
DocSAM [19]	77.0	72.1	82.6	84.2	80.5	88.1	-	-	-
UD [27]	87.9	85.0	91.1	86.0	84.6	87.4	-	-	-
Hi-SAM-L (zero-shot) [38]	70.2	65.5	75.6	77.4	80.1	74.9	77.4	70.3	86.1
Ours (Joint)	86.4	86.0	86.7	87.1	88.1	86.2	83.5	85.0	82.1
Ours (FT)	87.0	86.4	87.5	87.9	87.3	88.4	83.2	80.6	85.9



Fig. 6: Visualizations of scene text detection and layout analysis.

and specialized SOTA detectors such as DPTText-DETR and SRFormer. We attribute this gap primarily to discrepancies in training data. In the appendix, we



Fig. 7: Visualizations of failure cases.

Table 3: Effectiveness of the point decoder. ‘†’ indicates replacing the S-Decoder in Hi-SAM with our designed point decoder for producing foreground points.

Method	Word		Text-line		Layout		Average Points	Average Time
	PQ	F	PQ	F	PQ	F		
Hi-SAM-L	63.10	81.83	66.17	84.85	57.61	74.49	1500	4.4s
Hi-SAM-L	61.04	78.82	62.29	79.30	54.59	70.43	280	1.2s
Hi-SAM-L [†]	63.50	81.74	66.96	85.30	58.35	74.63	268	1.2s
Ours (Joint)	62.52	80.84	65.40	84.08	57.74	74.51	287	1.4s
Ours (FT)	63.60	81.94	66.30	84.82	58.82	75.39	286	1.4s

demonstrate that ET-SAM can finally surpass DPText-DETR by an average of 1.2% F-score across benchmarks with the same training data. In the future, we could further scale up the training data volume for ET-SAM to obtain superior performance.

Visualizations. We provide visualizations across different datasets in Fig. 6 and Fig. 7, including several failure cases. Overall, most word- and text-line-level masks are accurate and of high quality. However, some word instances are partially incomplete when the background is visually inconsistent, and imperfections remain in text-line and paragraph-level segmentation results.

4.4 Ablation Studies

Point Decoder. In Tab. 3, we validate the effectiveness of the point decoder. Hi-SAM randomly samples 1500 foreground points from pixel-level text segmentation outputs of its S-Decoder. While reducing the sampling quota improves efficiency, it results in significant performance degradation. In contrast, substituting the S-Decoder with the point decoder yields the variant Hi-SAM-L[†], which achieves superior performance alongside a $3.6\times$ inference acceleration (reducing latency from 4.4s to 1.2s). In terms of performance, Hi-SAM-L[†] consistently outperforms Hi-SAM-L, especially achieving higher PQ and F-score at the text-line and layout hierarchies, indicating more accurate and informative visual prompts. By further customizing the hierarchical mask decoder and applying joint training, our full method maintains competitive performance and

Table 4: Ablations on task prompts used in point decoder and HM-Decoder.

Point	Mask	HierText						Total-Text			CTW1500		
		Word		Text-line		Layout		Word			Text-line		
		PQ	F	PQ	F	PQ	F	F	P	R	F	P	R
×	×	62.35	80.87	65.08	83.69	56.77	73.38	81.99	78.54	85.77	86.79	85.76	87.83
×	✓	62.36	80.79	65.45	83.96	57.23	73.76	84.13	81.85	86.54	86.30	85.12	87.50
✓	×	62.64	81.08	65.66	84.27	57.56	74.10	86.00	86.14	85.86	86.75	87.04	86.46
✓	✓	62.52	80.84	65.40	84.08	57.74	74.51	86.37	86.02	86.72	87.14	88.07	86.24

Table 5: Ablations on the data combination during joint training.

#Row	Data	HierText						Total-Text			CTW1500		
		Word		Text-line		Layout		Word			Text-line		
		PQ	F	PQ	F	PQ	F	F	P	R	F	P	R
1	HierText	63.50	81.74	66.96	85.30	58.35	74.63	69.02	62.62	76.87	73.48	73.01	73.96
2	HierText+Total-Text+CTW1500	61.96	80.59	64.96	83.85	57.18	74.16	85.93	85.51	86.36	87.45	87.17	87.72
3	Row#2+IC13+IC15+TextSeg	62.52	80.84	65.40	84.08	57.74	74.51	86.37	86.02	86.72	87.14	88.07	86.24

high efficiency. Overall, predicting sparse points by the point decoder as visual prompts serves as a robust and efficient alternative to Hi-SAM’s random sampling, facilitating significant computational savings without compromising the performance.

Task Prompts. We investigate the impact of task prompts across diverse benchmarks in Tab. 4. On HierText, a dataset characterized by high word density and hierarchical text structures, the integration of either task prompt yields performance gains. The simultaneous application of prompts to both the point decoder and HM-Decoder results in the best layout analysis performance. On Total-Text, task prompts also bring clear performance gains. The task prompt in the point decoder plays a more essential role, and combining it with the mask task prompt further improves the F-score. On CTW1500, using either task prompt alone slightly decreases the F-score, but employing both task prompts together still achieves the best performance. Overall, these results underscore the effectiveness of task prompts in facilitating adaptation across diverse data distributions and varying text granularities.

Influence of Joint Training Data. Table 5 presents a comparative analysis of model performance on HierText, Total-Text, and CTW1500 under varying joint training data configurations. Initially, when trained exclusively on HierText, the model demonstrates limited cross-dataset generalization on Total-Text and CTW1500. With the incorporation of Total-Text and CTW1500 in joint training, the performance on these two single-level datasets is significantly improved, with F-score improvements of 16.91% and 13.97%, respectively. This outcome underscores the flexibility and data scalability inherent in our joint training strategy. Meanwhile, performance degradation is observed on HierText, probably attributable to data distribution shifts inherent to the relatively small-scale heterogeneous data mixing. Subsequently, upon further integrating ICDAR2013,

Table 6: Ablations on point threshold for filtering points on word heatmap.

Point Threshold	HierText						Total-Text			CTW1500		
	Word		Text-line		Layout		Word			Text-line		
	PQ	F	PQ	F	PQ	F	F	P	R	F	P	R
0.5	62.35	80.75	65.18	84.06	57.87	74.88	84.24	79.41	89.70	86.69	85.33	88.10
0.6	62.52	80.84	65.40	84.08	57.74	74.51	85.45	82.53	88.57	87.05	86.68	87.43
0.7	62.17	80.11	64.17	81.96	56.36	72.43	86.37	86.02	86.72	87.14	88.07	86.24
0.8	56.98	72.73	54.72	69.02	47.99	61.41	83.73	88.60	79.36	86.41	89.55	83.48

Table 7: Inference latency under different point thresholds.

Point Threshold	HierText		Total-Text		CTW1500	
	Average	Average	Average	Average	Average	Average
	Points	Time	Points	Time	Points	Time
0.5	296	1.05s	85	0.50s	109	0.50s
0.6	287	1.02s	82	0.48s	107	0.50s
0.7	263	0.92s	80	0.47s	105	0.49s
0.8	182	0.72s	73	0.46s	101	0.48s

ICDAR2015, and TextSeg, noticeable improvements are observed on Total-Text and HierText. Collectively, we demonstrate that our method establishes a viable pathway for data scalability. We anticipate that expanding data volume and scenario coverage will further enhance the overall performance and robustness.

Influence of Prompt Point Threshold. In Tab. 6, we present a sensitivity analysis regarding the threshold for sampling prompt points derived from the word-level heatmap. The empirical results indicate an inherent trade-off, *i.e.*, increasing the threshold reduces the density of sampled prompt points and consequently enhances precision at the expense of recall. Specifically, on HierText, a threshold of 0.6 achieves the optimal balance across various text granularities. Conversely, Total-Text and CTW1500 benefit from a stricter threshold of 0.7 to maximize the F-score. Consequently, our experimental protocol standardizes the point threshold at 0.6 for HierText and 0.7 for the remaining benchmarks. We additionally study the effect of the point threshold on overall inference latency on an A800 GPU in Tab. 7. As the point threshold increases, the average number of sampled prompt points decreases, leading to a reduction in inference latency. However, the resulting improvement in efficiency is marginal in practice, therefore we set the threshold considering performance.

Loss Weight Configuration. The weighting of word, text-line, and paragraph mask losses follows Hi-SAM and the heatmap loss weight is set to 50 to balance its magnitude with other loss terms. As shown in Tab. 8, the setting of the heatmap loss weight essentially reflects a trade-off between prompt point accuracy and mask quality. When the weight is relatively small, insufficient prompt point accuracy leads to degraded performance in dense text scenarios. Conversely, when the weight is relatively large, it slightly compromises mask quality,

Table 8: Ablations on heatmap loss weight.

Heatmap Loss Weight	HierText						Total-Text			CTW1500		
	Word		Text-line		Layout		Word			Text-line		
	PQ	F	PQ	F	PQ	F	F	P	R	F	P	R
25	61.61	79.85	64.90	83.63	56.68	73.23	86.57	86.07	87.08	87.37	87.38	87.35
50	62.52	80.84	65.40	84.08	57.74	74.51	86.37	86.02	86.72	87.14	88.07	86.24
100	62.06	80.60	65.34	84.23	56.75	73.52	86.91	87.15	86.68	86.19	86.59	85.79

resulting in a performance drop on both HierText and CTW1500. Therefore, we empirically set the heatmap loss weight to 50, which provides a reasonable balance between prompt point accuracy and mask quality.

5 Conclusions and Limitations

In this work, we present ET-SAM for unified scene text detection and layout analysis, addressing the critical bottlenecks of inference latency and data utilization that may hinder the practical application of SAM-based frameworks. Instead of relying on dense, random foreground point sampling, ET-SAM leverages a tailored lightweight decoder to derive sparse, high-confidence points from word-level heatmaps, successfully eliminating excessive computational overhead. Furthermore, we effectively harmonize heterogeneous annotations in diverse datasets via designing a joint training strategy and introducing task prompt tokens. Extensive experiments demonstrate that, compared to Hi-SAM, ET-SAM achieves approximately $3\times$ inference acceleration with competitive performance on HierText and realizes a substantial 11.0% improvement in average F-score across Total-Text, CTW1500, and ICDAR2015.

However, ET-SAM is primarily designed for text detection and cannot be directly applied to high-level text understanding tasks, as its learned visual features are not aligned with vision-language models, limiting its end-to-end multi-modal application. Future work could explore incorporating semantic modeling and cross-modal alignment to address this limitation.

Acknowledgements

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project under Grant 2025ZD0123602, in part by the National Natural Science Foundation of China under Grants 62225113, U23B2048 and 625B2132, in part by the Innovative Research Group Project of Hubei Province under Grant 2024AFA017, in part by the Science and Technology Major Project of Hubei Province under Grants 2024BAB046 and 2025BCB026, and in part by the New Cornerstone Science Foundation through the XPLOER PRIZE. This work was also supported by WHU-Kingsoft Joint Lab. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

1. Bu, Q., Park, S., Khang, M., Cheng, Y.: Srformer: text detection transformer with incorporated segmentation and regression. In: AAAI. vol. 38, pp. 855–863 (2024)
2. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. In: ICLR (2026)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229 (2020)
4. Cen, J., Zhou, Z., Fang, J., Yang, C., Shen, W., Xie, L., Zhang, X., Tian, Q.: Segment anything in 3d with nerfs. In: NeurIPS. vol. 36, pp. 25971–25990 (2023)
5. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv:2305.06558 (2023)
6. Ch’ng, C.K., Chan, C.S.: Total-text: A comprehensive dataset for scene text detection and recognition. In: ICDAR. vol. 1, pp. 935–942. IEEE (2017)
7. Dai, P., Zhang, S., Zhang, H., Cao, X.: Progressive contour regression for arbitrary-shape scene text detection. In: CVPR. pp. 7393–7402 (2021)
8. Du, B., Ye, J., Zhang, J., Liu, J., Tao, D.: I3cl: Intra-and inter-instance collaborative learning for arbitrary-shaped scene text detection. IJCV **130**(8), 1961–1977 (2022)
9. Gao, J., Zhang, D., Wang, F., Ning, L., Zhao, Z., Li, X.: Combining sam with limited data for change detection in remote sensing. IEEE TGRS (2025)
10. He, H., Ye, M., Zhang, J., Liu, J., Du, B., Tao, D.: Gomatching: A simple baseline for video text spotting via long and short term matching. NeurIPS **37**, 25663–25686 (2024)
11. He, H., Zhang, J., Ye, M., Liu, J., Du, B., Tao, D.: Gomatching++: Parameter- and data-efficient arbitrary-shaped video text spotting and benchmarking. IEEE TCSVT (2026)
12. Huang, M., Zhang, J., Peng, D., Lu, H., Huang, C., Liu, Y., Bai, X., Jin, L.: Estextspotter: Towards better scene text spotting with explicit synergy in transformer. In: ICCV. pp. 19495–19505 (2023)
13. Huang, Z., Dai, M., Zhang, Y., Zhang, J., Shan, H.: Point segment and count: A generalized framework for object counting. In: CVPR. pp. 17067–17076 (2024)
14. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: ICDAR. pp. 1156–1160. IEEE (2015)
15. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: ICDAR. pp. 1484–1493. IEEE (2013)
16. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR. pp. 9404–9413 (2019)
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV. pp. 4015–4026 (2023)
18. Li, J., Jain, J., Shi, H.: Matting anything. In: CVPR. pp. 1775–1785 (2024)
19. Li, X.H., Yin, F., Liu, C.L.: Docsam: Unified document image segmentation via query decomposition and heterogeneous mixed learning. In: CVPR. pp. 15021–15032 (2025)
20. Liao, M., Shi, B., Bai, X.: Textboxes++: A single-shot oriented scene text detector. IEEE TIP **27**(8), 3676–3690 (2018)

21. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: A fast text detector with a single deep neural network. In: AAAI. vol. 31 (2017)
22. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: AAAI. vol. 34, pp. 11474–11481 (2020)
23. Liao, M., Zou, Z., Wan, Z., Yao, C., Bai, X.: Real-time scene text detection with differentiable binarization and adaptive scale fusion. IEEE TPAMI **45**(1), 919–931 (2022)
24. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In: CVPR. pp. 9809–9818 (2020)
25. Liu, Y., Jin, L., Zhang, S., Luo, C., Zhang, S.: Curved scene text detection via transverse and longitudinal sequence connection. PR **90**, 337–345 (2019)
26. Liu, Y., Shen, C., Jin, L., He, T., Chen, P., Liu, C., Chen, H.: Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. IEEE TPAMI **44**(11), 8048–8064 (2021)
27. Long, S., Qin, S., Panteleev, D., Bissacco, A., Fujii, Y., Raptis, M.: Towards end-to-end unified scene text detection and layout analysis. In: CVPR. pp. 1049–1059 (2022)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
29. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV. pp. 565–571. Ieee (2016)
30. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. In: ICLR (2025)
31. Wang, D., Zhang, J., Du, B., Xu, M., Liu, L., Tao, D., Zhang, L.: Samrs: Scaling-up remote sensing segmentation dataset with segment anything model. In: NeurIPS. vol. 36, pp. 8815–8827 (2023)
32. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: CVPR. pp. 9336–9345 (2019)
33. Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu, G., Shen, C.: Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In: ICCV. pp. 8440–8449 (2019)
34. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic and fast instance segmentation. NeurIPS **33**, 17721–17732 (2020)
35. Xu, X., Zhang, Z., Wang, Z., Price, B., Wang, Z., Shi, H.: Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In: CVPR. pp. 12045–12055 (2021)
36. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv:2304.11968 (2023)
37. Ye, J., Chen, Z., Liu, J., Du, B.: Textfusenet: Scene text detection with richer fused features. In: IJCAI. vol. 20, pp. 516–522 (2020)
38. Ye, M., Zhang, J., Liu, J., Liu, C., Yin, B., Liu, C., Du, B., Tao, D.: Hi-sam: Marrying segment anything model for hierarchical text segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(3), 1431–1447 (2025)
39. Ye, M., Zhang, J., Zhao, S., Liu, J., Du, B., Tao, D.: Dptext-detr: Towards better scene text detection with dynamic points in transformer. In: AAAI. vol. 37, pp. 3241–3249 (2023)
40. Ye, M., Zhang, J., Zhao, S., Liu, J., Liu, T., Du, B., Tao, D.: Deepsolo++: Let transformer decoder with explicit points solo for multilingual text spotting. arXiv preprint arXiv:2305.19957 (2023)

41. Ye, M., Zhang, J., Zhao, S., Liu, J., Liu, T., Du, B., Tao, D.: Deepsolo: Let transformer decoder with explicit points solo for text spotting. In: CVPR. pp. 19348–19357 (2023)
42. Yue, W., Zhang, J., Hu, K., Wu, Q., Ge, Z., Xia, Y., Luo, J., Wang, Z.: Part to whole: Collaborative prompting for surgical instrument segmentation. arXiv:2312.14481 (2023)
43. Yue, W., Zhang, J., Hu, K., Xia, Y., Luo, J., Wang, Z.: Surgicalsam: Efficient class promptable surgical instrument segmentation. In: AAAI (2023)
44. Zhang, S.X., Yang, C., Zhu, X., Yin, X.C.: Arbitrary shape text detection via boundary transformer. IEEE TMM **26**, 1747–1760 (2023)
45. Zhang, X., Su, Y., Tripathi, S., Tu, Z.: Text spotting transformers. In: CVPR. pp. 9519–9528 (2022)
46. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. In: CVPR. pp. 5551–5560 (2017)
47. Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., Zhang, W.: Fourier contour embedding for arbitrary-shaped text detection. In: CVPR. pp. 3123–3131 (2021)