RANK++LETR: Learn to Rank and Optimize Candidates for Line Segment Detection

Xin Tong **Baojie Tian** **Yufei Guo** **Zhe Ma**[*]
Intelligent Science & Technology Academy of CASIC
xin_tong@pku.edu.cn, mazhe_thu@163.com

Abstract

It is observed that the confidence score may fail to reflect the predicting quality accurately in previous proposal-based line segment detection methods, since the scores and the line locations are predicted simultaneously. We find that the line segment detection performance can be further improved by learning-based line candidate ranking and optimizing strategy. To this end, we build a novel end-to-end line detecting model named RANK++LETR upon deformable DETR architecture, where the encoder is used to select the line candidates while the decoder is applied to rank and optimize these candidates. We design line-aware deformable attention (LADA) module in which attention positions are distributed in a long narrow area and can align well with the elongated geometry of line segments. Moreover, we innovatively apply ranking-based supervision in line segment detection task with the design of contiguous labels according to the detection quality. Experimental results demonstrate that our method outperforms previous SOTA methods in prediction accuracy and gets faster inferring speed than other Transformer-based methods.

1 Introduction

Line segments and junctions are crucial information in structured scenes and are ubiquitous in human-made environments. An accurate line segment detection algorithm can significantly enhance various computer vision applications, such as 3D reconstruction [36; 2], camera calibration [22; 27], depth estimation [40], scene understanding [11], object detection [29], SLAM [18; 41], etc. Traditional geometric-based line segment detection algorithms usually extract low-level image features and group them into line segments. These methods often run at a fast speed, while may suffer from fragmented prediction. Learning-based methods achieve promising results by learning knowledge from image sets with supervision, which are able to detect longer and more meaningful line segments.

Proposal-based methods constitute a pivotal component within learning-based approaches and have been extensively studied recently. These models typically output target predictions such as endpoint coordinates or midpoint coordinates with endpoint offsets for line segments. Generally, these methods first simultaneously predict both the positional coordinates and confidence scores of candidate line segments, then select the top-ranked proposal-based on confidence scores as final predictions. Previous study [31] points out that some accurately detected line segments are assigned low confidence scores during prediction since confidence prediction and location regression of line segments are independent. Specifically, given the simultaneously predicted line candidates with confidence scores and positions, the detection performance can be significantly improved even if only proper scores are assigned. Based on the observation, we find that the line segment detection performance can be further improved by learning-based line candidate ranking and optimizing strategy.

Transformers depend on attention modules to gather relevant features. However, in the classic deformable attention module, the attention position of a query is usually around a reference point on

[*] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

the feature map, which is not easy to adapt to the long and narrow area for detecting line segments. Thus, we specially design a novel attention module named line-aware deformable attention (LADA), which can align well with the elongated geometry of line segments for better perceiving line features.

To effectively train our model for quality-aware ranking of line candidates, proper supervision is essential. Ranking-based losses aim to rank the positive predictions above negative ones and sort the high-score candidates over low-score ones, which naturally suit our line ranking task. We define the contiguous label according to the quality of the predictions, which are based on the distance of the nearest ground truth and the predictions. Then, ranking-based losses can be applied to promote higher scores for high-quality predictions.

In this work, we propose a novel end-to-end line detecting model named RANK++LETR upon deformable DETR architecture. For Transformer-based line detection methods, LETR uses DETR architecture where the backbone and the encoder are used to extract features and the decoder is used to generate line segments. RANK-LETR adopts an encoder-only network and directly predicts the lines from the encoder. Different from the above approaches, our method leverages the complete network architecture of deformable DETR in design philosophy. Specifically, we apply distinct supervision during the encoder and decoder stages: the encoder is guided to predict candidate line segments, while the decoder is responsible for ranking and refining these candidate line segments.

Our contributions can be summarized as follows: (1) We present a novel DETR-like line segment detection framework, where the encoder is used to generate candidate line segment proposals, while the decoder is used to optimize their confidence scores and locations. (2) We propose line-aware deformable attention where the perception field can be long and narrow to catch features along candidate line segments. (3) By defining the continuous label for line segment detection, we employ ranking-based supervision to optimize confidence scores in the decoder. (4) Experimental results demonstrate that our method outperforms previous SOTA approaches in prediction accuracy while running faster than previous Transformer-based models.

2 Related Work

2.1 Line Segment Detection.

Traditional line detection methods often rely on grouping image gradient [32; 1; 21] and pre-defined rules [9; 7; 34]. Recently, learning-based methods have achieved promising results. For junction based methods, DWP [13] predicts junction map and edge map in two branches before merging them. PPGNet [43] uses a point-pair graph to describe junctions and line segments. L-CNN [45] applies line proposal and LoI pooling to propose candidate lines and verify them. Methods with dense prediction first predict representation map and extract line segments with post-processing. AFM [35] proposes attraction field maps to represent the image space and uses a squeeze module to generate line segment maps. HAWP [38] builds a hybrid model considering 4D attraction field and further extends to holistic attraction field [37]. Lin *et al.* [17] apply deep Hough transform to the previous detection architectures. TP-LSD [14] introduces tri-points line segment representation for end-to-end detection. M-LSD [8] presents SoL augmentation and designs an extremely efficient architecture for fast detection. SOLD2 [26] and DeepLSD [25] apply unsupervised pipelines and are able to detect fine and sufficient line segments. Transformer-based method can directly output the locations of the line segments. LETR [33] models it as object detection and predicts line segments with DETR architecture. RANK-LETR [31] applies match predicting and re-ranking to improve the training efficiency and the recall of high quality predictions. In this work, we extend Transformer and proposal-based method with a pure learnable optimization module for better performance.

2.2 Visual Transformer for Detection

Visual Transformer for object detection task is originated in DETR [3], in which a Transformer-based encoder-decoder framework is adopted and end-to-end supervision is applied with bipartite matching. Zhu *et al.* [46] further proposes deformable DETR in which each query only focuses on a small set of keys with learnable locations. A denoising training method is presented in [16] and a query formulation using dynamic anchor boxes is introduced in [20] to speed up training convergence of DETRs, which are further extended in DINO [42] Hou *et al.* [12] designed a hierarchical query filtering strategy to reduce the computational redundancy of DETR. Visual Transformer is widely

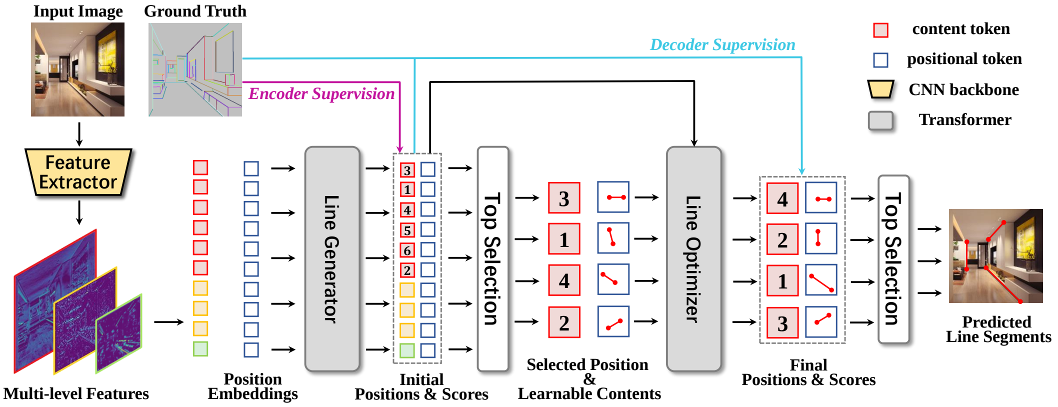


Figure 1: Overview of the proposed RANK++LETR. An image is first fed into a CNN-based feature extractor and multi-level feature maps are obtained from different layers. Then, initial line segments are generated by Transformer encoder based line segment generator with the multi-level features. Candidate line segments with high confidence scores are selected to initialize the reference points in the proposed line-aware deformable attention of the line optimizer. Finally, the confidence scores and positions of the candidate line segments are optimized, and we choose the candidate line segments according to their optimized confidence scores with non-maximum suppression as our final prediction. The entire method can be end-to-end training and inferring.

used in many other visual detection tasks. Xu *et al.* [33] apply DETR in line segment detection with a multi-scale encoder-decoder strategy. Tong *et al.* [30] use Transformer decoders in end-to-end vanishing point detection with Gaussian hemisphere division. Chenhang *et al.* [10] apply Transformer in 3D object detection with a set-to-set translation strategy. Tan *et al.* [28] utilize Transformers to represent context features and line segments for detecting and reconstructing 3D planes from a single image. Liu *et al.* [19] employed a Transformer-based architecture for end-to-end lane detection. Leveraging the encoder-decoder architecture of deformable DETR, we design to generate line segment proposals and optimize line candidates successively in a single framework.

2.3 Ranking-based Losses

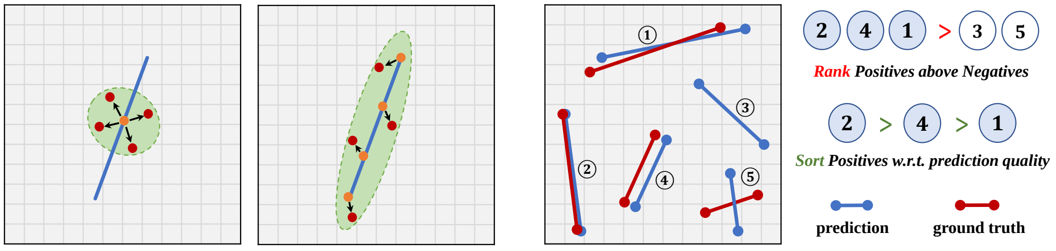
Ranking-based losses have received much attention in recent studies. Chen *et al.* [5] first propose Average Precision Loss to address the imbalance of foreground-background classification problem by framing object detection as a ranking task. Rank & Sort (RS) Loss that defines a ranking objective between positives and negatives as well as a sorting objective to prioritize positives with respect to their continuous IoUs is designed in [24]. Yavuz *et al.* [39] apply Bucketed Ranking-based (BR) Losses which group negative predictions into several buckets. Cetinkaya *et al.* [4] extend ranking-based Loss to edge detection with uncertainty modeling. Ranking-based loss is also used in 3D reasoning frameworks [15]. In this work, we naturally employ ranking-based losses to supervise the line optimizer for better confidence prediction.

3 Method

3.1 Line Segment Detection Modeling

Building upon deformable DETR encoder-decoder architecture, we model line segment detection as an end-to-end process including line proposal generation and optimization, with each line proposal parameterized by the position of its endpoints and the confidence score. Specifically, the encoder generates candidate line segment proposals with associated confidence scores and positions, while the decoder performs optimization through positional refinement and confidence re-ranking. The entire pipeline can be end-to-end training and inferring.

As shown in Fig.1, the proposed method takes images as input and finally predicts a given number of line segments sorted according to their confidence. It is initiated by processing input images through a CNN backbone to capture multi-level feature representations. Then these features subsequently


①
②
③
④
⑤
2 4 1 > 3 5
Rank Positives above Negatives
2 > 4 > 1
Sort Positives w.r.t. prediction quality
prediction
ground truth

(a) **(b)**

Figure 2: (a) The attention positions (red points) of a query are usually around a reference point (orange point) on the feature map in classic deformable attention module (left). In our proposed line-aware deformable attention (LADA, right), the attention positions are distributed in a long and narrow area and align well with the elongated geometry of candidate line segments (blue line). (b) Ranking-based supervisions rank the positives above the negatives and sort the positives with respect to their prediction qualities. We define the prediction quality relating to the distance between a predicted line segment and its nearest ground truth. The positives and negatives are also distinguished according to prediction quality, e.g., with a proper threshold.

undergo hierarchical encoding via a deformable Transformer encoder for candidate line proposal generation. The encoder implements a spatial-aware assignment mechanism where each spatial feature point is assigned to detect line segments whose centroid resides within its proximal receptive field. According to the confidence score predicted by the encoder, line segments with high scores are chosen as candidates for the decoder. In this work, we rank line segments with their prediction quality and refine their location simultaneously. The decoder takes learnable content and candidate line segment positions as initiating and gradually gathers information from the encoder feature maps. Finally, the decoder predicts the optimized confidence scores and their locations simultaneously in a pure learning-based manner.

3.2 Line-Aware Deformable Attention Module

In DETRs, each query adaptively aggregates information from feature vectors at specific attention positions in the corresponding feature map with the attention module. Typically, given the query vector $\boldsymbol{z}_q$ and feature map $\boldsymbol{x}$, the multi-head attention module can be represented as

$$\mathcal{A}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{h=1}^{H} \boldsymbol{W}_h [\sum_{k=1}^{K} A_{hqk} \cdot \boldsymbol{W}'_h \boldsymbol{x}(\boldsymbol{p})], \tag{1}$$

where h indexes the attention head and k indexes the sampled keys. $\boldsymbol{W}_h$ and $\boldsymbol{W}'_h$ are learnable projection weights. A_{hqk} is the *Scaled Dot-Product Attention* weight. $\boldsymbol{p}$ indicates the attention position of query vector $\boldsymbol{z}_q$ on feature map $\boldsymbol{x}$ at the h-th head and k-th sampled point. Here we adopt similar notations as deformable DETR [46] for a better understanding.

For the classical multi-head attention module, the attention positions are generally predefined to cover all feature point locations across the entire feature map. Thus, $\boldsymbol{p}$ can be defined as the k-th index of the predefined locations $\mathbf{G}$

$$\boldsymbol{p} = \mathbf{G}[k]. \tag{2}$$

Deformable attention introduces learnable deformable attention mechanism, where the attention positions for each query are determined by a reference point $\boldsymbol{p}_q$ and multiple learnable offsets, which can be determined as

$$\boldsymbol{p} = \boldsymbol{p}_q + \mathcal{F}_{hk}(\boldsymbol{z}_q), \tag{3}$$

where $\mathcal{F}_{hk}(\boldsymbol{z}_q)$ is a function of $\boldsymbol{z}_q$, e.g., a linear function. It allows the attention positions to be dynamically adapted based on the query.

Since our method uses attention module for line segment verification and refinement, the query vector should focus on regions near the corresponding candidate line segment. These regions are generally long and narrow as shown in Fig.2 (a). However, in the classic deformable attention module, the attention position of a query is usually around a reference point on the feature map, which is not easy to adapt to the long and narrow area for detecting line segments. To address this limitation, we propose a line-aware deformable attention (LADA) module, where the attention positions are aligned with the elongated geometry of candidate line segments. Specifically, $\boldsymbol{p}$ in LADA module can be represent as

$$\boldsymbol{p} = \frac{k-1}{K-1}\boldsymbol{s}_q + \frac{K-k}{K-1}\boldsymbol{e}_q + \mathcal{F}_{hk}(\boldsymbol{z}_q), \tag{4}$$

where $\boldsymbol{s}_q$ and $\boldsymbol{e}_q$ are two endpoints of the candidate line segment $\boldsymbol{l}_q$. With this design, the receptive field of the attention module is distributed along the candidate line segments. It enables the model to better perceive their alignment with semantically image features, e.g., edges and endpoints.

3.3 Ranking Line Candidates with Prediction Quality

For further optimizing the ranking through the prediction quality of the candidate line proposal, we employ ranking-based losses to supervise the confidence scores. The key to applying ranking-based supervision is to define contiguous labels that can reflect the line detection quality properly. We define a simple yet efficient contiguous label l_i based on the distances between the predicted lines and the nearest ground truths of them. Specifically, it can be defined as

$$l_i = max(0, 1 - \delta_l * min(\|\boldsymbol{e}_i - \boldsymbol{e}_i^*\|_2 + \|\boldsymbol{s}_i - \boldsymbol{s}_i^*\|_2, \|\boldsymbol{e}_i - \boldsymbol{s}_i^*\|_2 + \|\boldsymbol{s}_i - \boldsymbol{e}_i^*\|_2)), \tag{5}$$

where $\boldsymbol{e}_i, \boldsymbol{s}_i$ means the two endpoints of the i-th line segment and $\boldsymbol{e}_i^*, \boldsymbol{s}_i^*$ are two endpoints of the corresponding ground truth. δ_l is a factor that controls the threshold of distance. The ranking-based solution we used consists of two components that are visually exhibited in Fig.2 (b) for better understanding.

Ranking Positives over Negatives. Given the candidate line segments and their contiguous labels, ranking loss is used to rank the positives above the negatives. We consider the lines whose $l_i > 0$ as positives while others are considered as negatives. $\mathbf{P}$ indicates the set of positive line segments and $\mathbf{N}$ indicates the set of negatives. We define the ranking loss $\mathcal{L}_{rank}$ using a differentiable approximation of Average Precision following [4], which can be presented as

$$\mathcal{L}_{rank} = 1 - \frac{1}{|\mathbf{P}|}\sum_{i \in \mathbf{P}} \frac{\sum_{j \in \mathbf{N}} H(x_{ij})}{\sum_{j \in \mathbf{P} \cup \mathbf{N}} H(x_{ij})}, \tag{6}$$

where $H(x_{ij}) = max(1, min(0, (c_j - c_i)/2\delta_H + 0.5))$ is the step function with a δ_H-approximation around the step. c is the confidence score.

Sorting Positives with Prediction Quality. Each candidate has a different prediction quality according to the alignment degree with the corresponding ground truth, i.e., l_i. Thus, we supervise to sort the positive line segments with l_i, making the well-aligned predictions tend to get higher confidence scores. To this end, we use the sorting objective $\mathcal{L}_{sort}$ introduced in [24], which can be presented as

$$\mathcal{L}_{sort} = \frac{1}{|\mathbf{P}|}\sum_{i \in \mathbf{P}}\left(\frac{\sum_{j \in \mathbf{P}} H(x_{ij})(1 - l_i)}{\sum_{j \in \mathbf{P}} H(x_{ij})} - \frac{\sum_{j \in \mathbf{P}} H(x_{ij})[l_j \geq l_i](1 - l_j)}{\sum_{j \in \mathbf{P}} H(x_{ij})[l_j \geq l_i]}\right), \tag{7}$$

where the former term is the current sorting error and the latter term is the target sorting error, respectively. More details are referred to [24].

3.4 Training Strategy

During training, supervisions are added on both Transformer encoder and decoder, called encoder supervision and decoder Supervision, respectively. For the encoder supervision, binary cross-entropy loss $\mathcal{L}_{conf}^E$ and L2 loss $\mathcal{L}_{pos}^E$ are applied to supervise the confidence scores and positions of the predicted line segments. For the decoder Supervision, we use ranking-based supervision $\mathcal{L}_{rank}$ and $\mathcal{L}_{sort}$ to learn ranking the candidate line segments. Moreover, L2 loss $\mathcal{L}_{pos}^D$ is also used to optimize the positions of line segment candidates. The final loss we used can be represented as

$$\mathcal{L}_{total} = \lambda_c \mathcal{L}_{conf}^E + \lambda_{ep} \mathcal{L}_{pos}^E + \lambda_{dp} \mathcal{L}_{pos}^D + \lambda_r \mathcal{L}_{rank} + \lambda_s \mathcal{L}_{sort}. \tag{8}$$

Since the ranking supervision depends on the quality of the candidate line segments from the encoder, poor prediction results at the beginning will affect the training of the decoder. Therefore, we warm up the model for several epochs. Only the encoder is supervised during the beginning of training. Then both encoder and decoder are jointly trained after the encoder can predict meaningful line segments.

Method	Wireframe					YUD					FPS
	sAP^5	sAP^{10}	sF^{10}	sF^{15}	LAP	sAP^5	sAP^{10}	sF^{10}	sF^{15}	LAP	
LSD [32]	6.7	8.8	-	-	18.7	7.5	9.2	-	-	16.1	100.0
DWP [13]	3.7	5.1	-	-	6.6	2.8	2.6	-	-	3.1	2.2
AFM [35]	18.3	23.9	-	-	36.7	7.0	9.1	-	-	17.5	14.1
LGNN [23]	-	62.3	-	-	-	-	-	-	-	-	15.8
TP-LSD [14]	57.6	57.2	-	-	61.3	27.6	27.7	-	-	34.3	20.0
L-CNN [45]	58.9	62.8	61.3	62.4	59.8	25.9	28.2	36.9	37.8	32.0	16.6
M-LSD [8]	56.4	62.1	-	-	61.5	24.6	27.3	-	-	30.7	115.4*
M-LSD†[8]	63.3	67.1	-	-	64.2	27.5	28.5	-	-	32.4	32.9
HAWPv2 [38]	65.5	69.5	66.4	67.4	-	28.2	30.4	41.0	42.0	-	45
LETR [33]	59.2	65.6	66.1	67.4	65.1	24.0	27.6	39.6	41.1	32.5	5.4
RANK-LETR [31]	65.0	69.7	66.7	67.7	65.6	27.6	30.1	39.7	40.6	34.1	9.0
RANK++LETR (Ours)	**67.9**	**72.1**	**68.8**	**69.7**	**68.3**	**28.8**	**31.2**	**41.2**	**42.1**	**34.8**	12.4

Table 1: Quantitative comparisons on Wireframe [13] and YUD [6] datasets. We compare our proposed method with LSD [32], DWP [13], AFM [35], LGNN [23], TP-LSD[14], L-CNN [45], HAWPv2 [37], M-LSD [8], LETR [33] and RANK-LETR [31] methods. M-LSD† denotes the approach of combining M-LSD and HAWP. Average precision (sAP), F-score measurement (sF) and line matching average precision (LAP) are used as metrics for comprehensive comparisons. Our method outperforms previous SOTA methods in prediction accuracy and gets faster inferring speed than other Transformer-based methods.

4 Experimental Results

4.1 Experimental Setup

4.1.1 Datasets ans Metrics

We conduct our experiments in two publicly available datasets including the Wireframe dataset [13] and the YorkUrban dataset [6], which are widely used as line segment detection benchmarks. The Wireframe dataset contains 5,000 training and 462 testing images of man-made environments, while the YorkUrban dataset contains 102 testing images. The model is only trained on the Wireframe dataset and tested on both Wireframe and YorkUrban datasets as a typical protocol [14; 45]. For comprehensive comparison, we evaluate our models based on average precision (sAP), F-score measurement (sF) and line matching average precision (LAP). For fair comparison, we select no more than 500 prediction lines with high confidence scores of each image for quantitative analysis.

4.1.2 Implementation Details

Our training and evaluation are implemented in PyTorch. We use 4 NVIDIA V100 GPUs for training and 1 GPU for evaluation. We train our model for 240 epochs for warming up and 120 epochs for jointly optimizing. The learning rate is set as 5×10^{-4}. The image size and the batch size are set as 512×512 and 8, respectively. We use the AdamW optimizer and set weight decay as 10^{-4}.

The results of our method are predicted on the features of the resolution of 128×128. λ_c, λ_{ep}, λ_{dp}, λ_r, λ_s are set to $1, 10, 10, 1, 1$, respectively. Moreover, we use auxiliary loss on the early layer in the Transformer-based encoder with a factor of 0.8. K is set to 4 for sampling. Up to 500 line segments with high scores are detected with NMS for comparison in our method.

4.2 Comparison with the SOTA

We compare our method with previous state-of-the-art methods including LSD [32], DWP [13], AFM [35], LGNN [23], TP-LSD [14], L-CNN [45], HAWPv2 [37], M-LSD [8], LETR [33] and RANK-LETR [31]. All the methods are learning-based methods except the classical LSD. M-LSD†denotes the method of combining M-LSD and HAWP. LETR, RANK-LETR and our proposed RANK++LETR take Transformer as key architecture while other approaches mainly use convolutional neural networks. Some methods such as [26; 25] are not chosen in the comparison because they are designed to tend to generate finer line segments. Thus, it is unfair for these methods to compare on

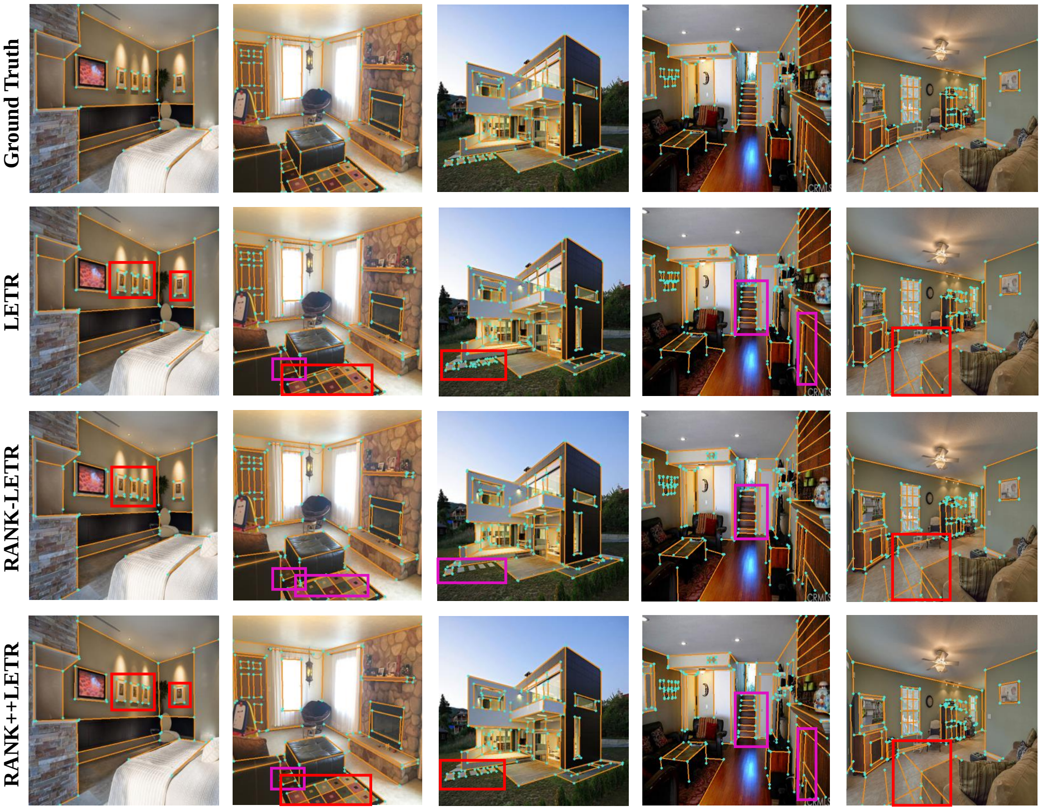


Figure 3: Visual examples of line segment detection results of Transformer-based methods including LETR [33], RANK-LETR [31] and our proposed RANK++LETR on the Wireframe dataset [13]. For a better visual experience, we highlight some significant differences of accurate detection with red bounding boxes and complete detection with purple bounding boxes. Our method tends to produce more accurate and complete line detection results.

standard wireframe and YorkUrban Dataset. The comparisons are conducted on the Wireframe dataset [13] and the YorkUrban dataset [6] and the results are listed in Table 1. The proposed RANK++LETR outperforms all previous SOTA methods in prediction accuracy. Especially, RANK++LETR gets about $\mathbf{2.9}$ percents improvement over RANK-LETR and $\mathbf{2.4}$ percents improvement over HAWPv2 on sAP^5 metric.

Moreover, RANK++LETR demonstrates superior efficiency compared to other Transformer-based approaches, primarily attributed to its minimalist architectural design that eliminates computationally intensive components such as global attention mechanisms and rotation augmentation operations. It is worth mentioning that a speed gap still persists when benchmarked against optimized CNN-based implementations, where we think that it is mainly due to the inherent computational complexity of Transformer architectures versus the convolutions. Our future work will focus on exploring lightweight Transformer or CNN-guided architecture for more efficient line detection approaches.

Visual examples of line segment detection results of Transformer-based methods including LETR, RANK-LETR and our proposed RANK++LETR on the Wireframe dataset are shown in Fig. 3. Our method tends to produce more accurate and complete line detection results. We highlight some significant differences in accurate detection with red bounding boxes and complete detection with purple bounding boxes.

To explore the generalization capability of the proposed method in non-structured scenarios, we directly applied the trained model to the NKL dataset [44] for semantic line detection. This dataset primarily contains natural landscape images. The detection results are shown in the Fig.4. The experiment demonstrates that even without training on the NKL dataset [44], our model still exhibits perception ability for semantic lines.

Line Optimizer	LADA	Ranking Loss	w/o Warm-up	R:S	Attention Points	Wireframe	
						sAP^{5}	sAP^{10}
-	-	-	-	-	-	63.2 (↓ 2.3)	68.1 (↓ 2.7)
✓	-	-	-	-	4	65.5 (-)	70.8 (-)
✓	✓	-	-	-	4	67.0 (↑ 1.5)	71.4 (↑ 0.6)
✓	-	✓	-	1:1	4	67.2 (↑ 1.7)	71.4 (↑ 0.6)
✓	✓	✓	-	1:1	4	**67.9** (↑ 2.4)	72.1 (↑ 1.3)
✓	✓	✓	✓	1:1	4	66.0	70.6
✓	✓	✓	-	1:0	4	67.3	71.6
✓	✓	✓	-	0:1	4	48.7	51.4
✓	✓	✓	-	1:1	2	67.4	71.5
✓	✓	✓	-	1:1	8	**67.9**	**72.2**

Table 2: Ablation and parameter study of our method on the Wireframe [13] dataset. We first construct a baseline method without decoder according to our modeling and then gradually add different components to explore their relevance and impact. Experimental results show that a second optimization can bring performance improvement, and both the LADA module and ranking loss contribute to better detecting results. Different numbers of attention points and weights of RS losses are also tested for a comprehensive study.

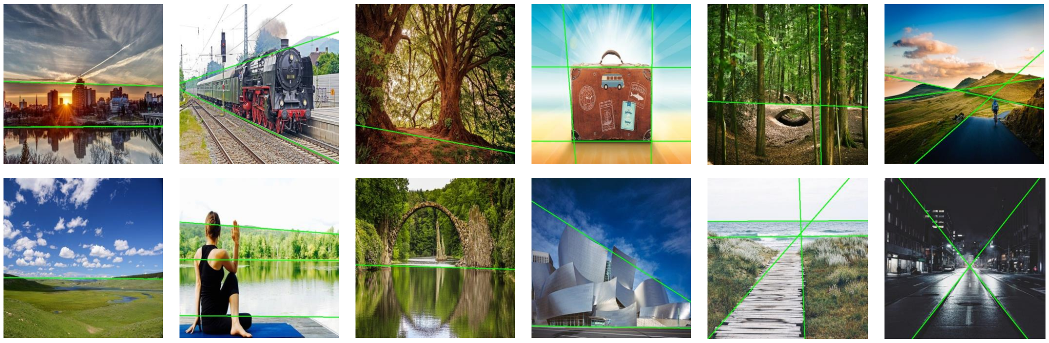


Figure 4: Visual examples for non-structured environments on NKL dataset [44]. Our method demonstrates strong generalization capability in semantic line detection scenarios.

4.3 Ablation and Parameter Study

To verify the effectiveness of components and find the influence of the hyperparameters in our proposed approach, we conduct an ablation and parameter study of RANK++LETR. The experience is conducted on the Wireframe dataset and *sAP* results are reported in Table 2.

As a baseline method, we apply ResNet50 as feature extractor and 6 layers Transformer encoder from classical deformable DETR for line segment detection with a matched prediction strategy, where the feature point closest to the centroid of a line segment is responsible for predicting it. Different from RANK-LETR, branch network and rotation enhancement network are no longer used. Based on the baseline, we use 6 layers deformable Transformer decoder and construct the complete proposal generating and candidate optimizing pipeline, where the selected proposals are used to initialize the inputs of the decoder. It reveals a significant performance gap between the proposed pipeline and baseline framework, which proves the effectiveness of our detection modeling. The line-aware deformable attention and ranking-based loss are then added individually. Both of them can bring obvious performance improvement. By combining the LADA module and ranking-based Loss together, we verify that our proposed method can get the best results benefiting from each novel component. Moreover, we find the warmup of encoder is also important to get a faster convergence for better results. The experimental results are listed in Table 2. The contribution of each component can be found during the performance differences.

The parameter study is conducted on LADA module and ranking-based losses. For LADA module, we explore the influence of the number of attention points, by changing the default 4 to 2 and 8, respectively. We find performance is not sensitive to the number of attention points when it exceeds

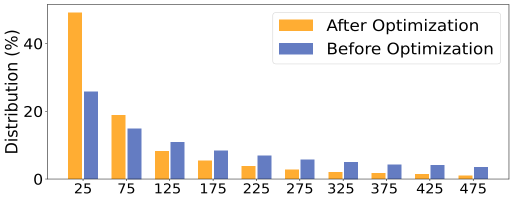


Figure 5: Statistics of the distribution on the rank of corresponding ground truth being recalled. The ranking-based supervision contributes to gaining high ranking distribution of the prediction, yielding a better line segment detection performance.

Table 3: Evaluation of the outputs before and after optimization, respectively. Adding line optimizer for learnable ranking the confidence scores and refining the positions brings an intuitive detecting performance improvement.

Metric	Before Optimize	After Optimize
sAP^5	64.1	**67.9**
sAP^{10}	68.8	**72.1**
sF^5	64.5	**66.3**
sF^{10}	67.4	**68.8**

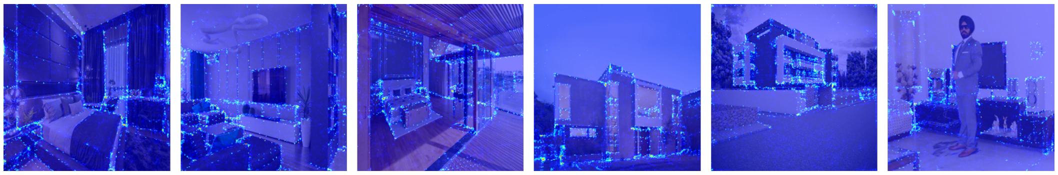


Figure 6: Attention maps of our proposed line-aware deformable attention module (LADA) in the decoder. We show the attention map of the last decoder layer for clear viewing. Brighter areas mean more attention. The attention points generally appear around the image lines, making the proposed LADA easier to capture line features.

4, which may be because it obtains enough useful information such as endpoints and edges. For ranking-based supervision, we test the ranking loss and sorting loss individually to find their roles in training. We observe that only adding ranking loss can bring limited improvement and only using sorting loss will even get worse performance. We think both loss terms should be used together in the task, which may be because they complement each other.

4.4 Analysis and Interpretation

In order to gain a deeper understanding of the proposed method, we conduct further analysis and feature visualization to exhibit intermediate process. As our method uses encoder to get initial proposals and optimize the score and position with decoder in successive processing, an intuitive way to explain the pipeline is to compare the predicting performance of the two outputs in one model directly. As demonstrated in Table.3, the line optimizer brings an intuitive detecting performance improvement.

We then compare the ranking quality to verify the effectiveness of the supervision and further explore the underlying reason for performance improvement. For the output results before and after line optimization, we select 500 line segments with high confidence and gather them into two groups, respectively. For each ground truth line segment, the closest line segment can be found from each group and the ranking of the line segment among its group can be recorded. It indicates that the corresponding ground truth will be recalled at the ranking in these predictions. It is obvious that a better method should gain higher ranking quality. In other words, with fewer predictions, more correct line segments can be detected. We statistically analyze the distribution of the rank of corresponding ground truth being recalled and the results are shown in Fig.5. The ranking-based supervision contributes to gaining high ranking distribution of the prediction, yielding a better line segment detection performance.

We also visualize the distribution of attention points in line-aware deformable attention (LADA). 6 images are randomly selected with the attention heat map masking on, which is shown in Fig.6. We show the attention map of the last decoder layer for clear viewing. Brighter areas mean more attention. The attention points generally appear around the image lines, demonstrating that the proposed LADA is easier to capture line features and more suitable for line detection tasks.

In addition to the module innovation, from a holistic perspective, the proposed method leverages the encoder-decoder design where the encoder generates line segment proposals while the decoder performs confidence re-ranking and position refinement of these segments. For the encoder, our method relies on its recall capability, whereas the decoder provides a secondary opportunity to enhance detection performance by ranking confidence and optimizing locations for the recalled line segments. Theoretically, since the encoder itself is trained as a line segment detector, cases where the proposed encoder itself demonstrates significantly inferior recall performance compared to other detection methods are unlikely to occur.

5 Conclusion

In this work, we develop proposal-based line segment detection methods with a novel pipeline where the high-quality line proposals are ranked and optimized with learnable features. To achieve this goal, we specially design a novel line-aware deformable attention (LADA) module in which attention positions are distributed in a long narrow area and can align well with the elongated geometry of line segments. For better supervising the ranking of selected proposals, ranking-based losses are employed and modified with proper contiguous labels generation to adapt line segment detection task. Building on the above techniques, we construct a novel line segment detection model with encoder-decoder architecture named RANK++LETR. Extensive experiments show that our method outperforms previous SOTA methods in prediction accuracy and gets faster inferring speed than other Transformer-based methods.

References

[1] Akinlar, C., Topal, C.: Edlines: A real-time line segment detector with a false detection control. Pattern Recognition Letters **32**(13), 1633–1642 (2011)

[2] Cai, Y., Wang, J., Yuille, A., Zhou, Z., Wang, A.: Structure-aware sparse-view x-ray 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11174–11183 (June 2024)

[3] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)

[4] Cetinkaya, B., Kalkan, S., Akbas, E.: Ranked: Addressing imbalance and uncertainty in edge detection using ranking-based losses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3239–3249 (June 2024)

[5] Chen, K., Li, J., Lin, W., See, J., Wang, J., Duan, L., Chen, Z., He, C., Zou, J.: Towards accurate one-stage object detection with ap-loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5119–5127 (2019)

[6] Denis, P., Elder, J.H., Estrada, F.J.: Efficient edge-based methods for estimating manhattan frames in urban imagery. In: European conference on computer vision. pp. 197–210. Springer (2008)

[7] Furukawa, Y., Shinagawa, Y.: Accurate and robust line segment extraction by analyzing distribution around peaks in hough space. Computer vision and image understanding **92**(1), 1–25 (2003)

[8] Gu, G., Ko, B., Go, S., Lee, S.H., Lee, J., Shin, M.: Towards light-weight and real-time line segment detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 726–734 (2022)

[9] Guil, N., Villalba, J., Zapata, E.L.: A fast hough transform for segment detection. IEEE transactions on image processing **4**(11), 1541–1548 (1995)

[10] He, C., Li, R., Li, S., Zhang, L.: Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8417–8427 (2022)

[11] Hofer, M., Maurer, M., Bischof, H.: Efficient 3d scene abstraction using line segments. Computer Vision and Image Understanding **157**, 167–178 (2017)

[12] Hou, X., Liu, M., Zhang, S., Wei, P., Chen, B.: Salience detr: Enhancing detection transformer with hierarchical salience filtering refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17574–17583 (June 2024)

[13] Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y.: Learning to parse wireframes in images of man-made environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 626–635 (2018)

[14] Huang, S., Qin, F., Xiong, P., Ding, N., He, Y., Liu, X.: Tp-lsd: Tri-points based line segment detector. In: European Conference on Computer Vision. pp. 770–785. Springer (2020)

[15] Kloepfer, D., Henriques, J.F., Campbell, D.: Loco: Learning 3d location-consistent image features with a memory-efficient ranking loss. Advances in Neural Information Processing Systems **37**, 124391–124419 (2025)

[16] Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13619–13627 (2022)

[17] Lin, Y., Pintea, S.L., van Gemert, J.C.: Deep hough-transform line priors. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. pp. 323–340. Springer (2020)

[18] Lin, Z., Zhang, Q., Tian, Z., Yu, P., Lan, J.: Dpl-slam: enhancing dynamic point-line slam through dense semantic methods. IEEE Sensors Journal (2024)

[19] Liu, R., Yuan, Z., Liu, T., Xiong, Z.: End-to-end lane shape prediction with transformers. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 3694–3702 (2021)

[20] Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329 (2022)

[21] Lu, X., Yao, J., Li, K., Li, L.: Cannylines: A parameter-free line segment detector. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 507–511. IEEE (2015)

[22] Magera, F., Hoyoux, T., Barnich, O., Van Droogenbroeck, M.: A universal protocol to benchmark camera calibration for sports. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3335–3346 (2024)

[23] Meng, Q., Zhang, J., Hu, Q., He, X., Yu, J.: Lgnn: A context-aware line segment detector. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 4364–4372 (2020)

[24] Oksuz, K., Cam, B.C., Akbas, E., Kalkan, S.: Rank & sort loss for object detection and instance segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3009–3018 (2021)

[25] Pautrat, R., Barath, D., Larsson, V., Oswald, M.R., Pollefeys, M.: Deeplsd: Line segment detection and refinement with deep image gradients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17327–17336 (2023)

[26] Pautrat, R., Lin, J.T., Larsson, V., Oswald, M.R., Pollefeys, M.: Sold2: Self-supervised occlusion-aware line description and detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11368–11378 (2021)

[27] Song, X., Kang, H., Moteki, A., Suzuki, G., Kobayashi, Y., Tan, Z.: Mscc: Multi-scale transformers for camera calibration. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3262–3271 (2024)

[28] Tan, B., Xue, N., Bai, S., Wu, T., Xia, G.S.: Planetr: Structure-guided transformers for 3d plane recovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4186–4195 (October 2021)

[29] Tang, X.s., Xie, X., Hao, K., Li, D., Zhao, M.: A line-segment-based non-maximum suppression method for accurate object detection. Knowledge-Based Systems **251**, 108885 (2022)

[30] Tong, X., Peng, S., Guo, Y., Huang, X.: End-to-end real-time vanishing point detection with transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5243–5251 (2024)

[31] Tong, X., Peng, S., Tian, B., Guo, Y., Huang, X., Ma, Z.: Improving transformer based line segment detection with matched predicting and re-ranking. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 7428–7436 (2025)

[32] Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. IEEE transactions on pattern analysis and machine intelligence **32**(4), 722–732 (2008)

[33] Xu, Y., Xu, W., Cheung, D., Tu, Z.: Line segment detection using transformers without edges. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4257–4266 (2021)

[34] Xu, Z., Shin, B.S., Klette, R.: Closed form line-segment extraction using the hough transform. Pattern Recognition **48**(12), 4012–4023 (2015)

[35] Xue, N., Bai, S., Wang, F., Xia, G.S., Wu, T., Zhang, L.: Learning attraction field representation for robust line segment detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1595–1603 (2019)

[36] Xue, N., Tan, B., Xiao, Y., Dong, L., Xia, G.S., Wu, T., Shen, Y.: Neat: Distilling 3d wireframes from neural attraction fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19968–19977 (June 2024)

[37] Xue, N., Wu, T., Bai, S., Wang, F.D., Xia, G.S., Zhang, L., Torr, P.H.: Holistically-attracted wireframe parsing: From supervised to self-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(12), 14727–14744 (2023)

[38] Xue, N., Wu, T., Bai, S., Wang, F., Xia, G.S., Zhang, L., Torr, P.H.: Holistically-attracted wireframe parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2788–2797 (2020)

[39] Yavuz, F., Cam, B.C., Dogan, A.H., Oksuz, K., Akbas, E., Kalkan, S.: Bucketed ranking-based losses for efficient training of object detectors. In: European Conference on Computer Vision. Springer (2024)

[40] Zavala, J.G.N., Martinez-Carranza, J.: Depth estimation from a single image using line segments only. In: Ibero-American Conference on Artificial Intelligence. pp. 331–341. Springer (2022)

[41] Zeng, D., Liu, X., Huang, K., Liu, J.: Epl-vins: Efficient point-line fusion visual-inertial slam with lk-rg line tracking method and 2-dof line optimization. IEEE Robotics and Automation Letters (2024)

[42] Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)

[43] Zhang, Z., Li, Z., Bi, N., Zheng, J., Wang, J., Huang, K., Luo, W., Xu, Y., Gao, S.: Ppgnet: Learning point-pair graph for line segment detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7105–7114 (2019)

[44] Zhao, K., Han, Q., Zhang, C.B., Xu, J., Cheng, M.M.: Deep hough transform for semantic line detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(9), 4793–4806 (2021)

[45] Zhou, Y., Qi, H., Ma, Y.: End-to-end wireframe parsing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 962–971 (2019)

[46] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in the experiments section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The information needed to reproduce the main experimental results of the paper is disclosed is the experimental section. The code are also provided in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code are provided in the supplemental material. The data is public available and have been widely used.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specify all the training and test details in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main results are reported with ablation study.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: Replace by [Yes]

   Justification: The computer resources are reported in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: the research conducted in the paper conform with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Related codes and papers are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: All the work is included in the paper and the supplemental material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing experiments.

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No potential risks.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No LLMs usage.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.