BiHashFormer: Hash-Driven Dual-Branch Transformer for Efficient Object Detection in HRW Shots

Jingchen Huang*
Department of Automation
Tsinghua University
Beijing, China
huang-jc23@mails.tsinghua.edu.cn

Wenxi Li*
KLATASDS-MOE
School of Statistics
East China Normal University
Shanghai, China
wxli@sfs.ecnu.edu.cn

Chenyang Lyu*
Alibaba Token Hub
Alibaba Group
Hangzhou, China
lyuchenyang.dcu@gmail.com

Moran Liu
Beijing National Research Center for Information Science and Technology
Department of Automation
Tsinghua University
Beijing, China
Zhuoxi Lab, Hangzhou, China
liumoreran@gmail.com

Haozhe Lin†
Beijing National Research Center for Information Science and Technology
Tsinghua University
Beijing, China
linhz@tsinghua.edu.cn

Yuchen Guo†
Beijing National Research Center for Information Science and Technology
Tsinghua University
Beijing, China
yuchen.w.guo@gmail.com

Abstract

Recent advances in gigapixel-level imaging have brought High-Resolution Wide shots to the forefront of research. However, these images present significant challenges: extreme sparsity of foreground, gigapixel-level resolutions and interleaving of foreground and background. This causes traditional detectors and attention mechanisms to be hindered by background, resulting in inefficiency and inaccuracy. To tackle this problem, we propose BiHashFormer, a dual sparsified branch transformer built on hashing. RoI Selector will predict the foreground proportion for top-k selecting target-containing windows. The compression branch processes windows selected and refines attention into HashAttention by discarding half of the key–value pairs, reducing cost and improving efficiency. The compensation branch uses HashMiner to perform low-cost hash searches in the remaining regions, recovering features of missed objects. Experiments confirm that the selection relationship in HashAttention is one-way and reveal the preference of different queries for keys. In experiments on the gigapixel benchmark PANDA, BiHashFormer reduces 37.5% backbone FLOPs while improving AP_{50} to 81.0%. Moreover, this parallel dual-branch structure allows the model to avoid inference delays caused by inter-branch dependencies, while achieving much greater stability than single-branch models, specially with low rates of RoI selection.

1 Introduction

In computer vision, object detection has long been recognized as a central challenge, requiring algorithms to simultaneously determine the location and category of target instances within an

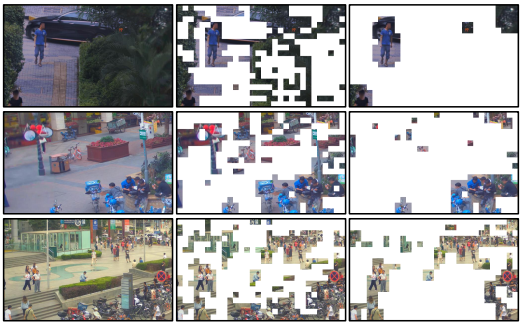
*Equal contribution.
†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

image. The standard output of such systems usually consists of a bounding box together with a class label. Landmark approaches, exemplified by Fast R-CNN [13], have achieved notable success on large-scale benchmarks such as MS COCO [17]. More recently, advances in nondestructive imaging have shifted attention toward High-Resolution Wide (HRW) detection [3, 11]. This setting involves analyzing imagery with both extremely fine spatial detail and expansive coverage, posing unique computational and modeling challenges. Representative scenarios include gigapixel drone photography [25], satellite-based observation [8, 9, 29], and panoramic camera data [4].

The first problem in HRW detection is that the objects we care about only take up a very small part of the whole picture. In normal detection datasets, the foreground can cover more than half of the image. But in HRW scenes, the foreground is usually less than 15% [16], because the images come from large real-world views that naturally include wide areas of background such as sky, buildings, or trees. This means detectors have to search through a lot of background to find a few targets, and even strong models like DINO [33], Deformable DETR [35], and RMT [12] often make mistakes when the background is too dominant. This problem is illustrated by Figure 1.

The second challenge in HRW detection is the heavy demand on computation. Many application scenarios, such as UAV inspection (VisDrone [10]) or medical analysis (VL-Health [20]), require detectors to respond in real time. Yet as image resolution grows—from millions of pixels in COCO [17] to billions in PANDA [27]—the processing cost rises sharply. Conventional detectors scale poorly with input size, and inference speed can drop by nearly two orders of magnitude, making real-time deployment difficult.



(a) Origin Images (b) SparseFormer (fixed ratio = 0.5) (c) BiHashFormer (Compr. & Compen.)

Figure 1: **Comparison of RoI between different backbones.** (b) SparseFormer selects a fixed ratio of regions for processing, regardless of input variability. (c) BiHashFormer combines lower selection ratio in the compression branch with dynamic compensation, making it more flexible and efficient. For the compensation branch, windows are drawn if any key in them is selected.

Thirdly, the strong interleaving of foreground and background. HRW detection often requires larger windows (in Swin [21]) to capture sufficient context, especially in crowded scenes. Yet enlarging the window in conventional attention introduces excessive background tokens, diluting the focus on true targets. For example, a person on a lawn may be overshadowed because the cumulative attention to numerous lawn tokens exceeds that of the human body, causing facial tokens to misdirect attention and lowering detection accuracy. Furthermore, this makes small objects or those surrounded by background more likely to be excluded during RoI selection, as shown in Figure 1.

Inspired by the widespread use of hashing in computer science, we propose BiHashFormer which combines dual-branch design with hashing, achieving both sparse attention and robust compensation for overlooked targets. Firstly, a variance-based scoring mechanism ranks windows, and a learned foreground predictor determines the image-dependent proportion of RoIs. For the selected RoIs, a sparse Attention mechanism called HashAttention is introduced. This approach compresses the key-value pairs for each query, leading to finer-grained computational reduction. At the same time, the quality of attention within RoIs will be improved. In the remaining regions, detection targets are localized within the Hamming feature space obtained through hash projection. Prototype encoding is employed for this localization and such compensation can be achieved with extremely low computational cost. On PANDA [27], BiHashFormer achieves a 37.5% reduction in FLOPs compared with SparseFormer [16], while significantly improving AP_{50}.

In summary, our key contributions are:

- We propose BiHashFormer, a dual-branch vision backbone based on transformer, which combines fine-grained sparse computation with global target compensation. It is able to effectively handle sparsely distributed objects intertwined with background in HRW images while reducing computational cost.

- We introduce HashAttention for the compression branch. By projecting features into a low-dimensional Hamming space, similarity computation is greatly reduced. The model compresses interactions between query and key-value pairs, achieving RoI-level sparse computation and refining attention results.
- We design HashMiner for the compensation branch. In the Hamming feature space, the model learns prototype codes for detection targets, enabling retrieval of objects from non-RoI regions with minimal overhead and providing compensation for small or difficult targets.
- We develop a label-free strategy with a special loss for prototype codes. Leveraging information from the compression branch, the model learns codes that capture broader and more accurate human-related semantics.

2 Related work

Object detection in high-resolution wide shots. High-resolution wide-shot (HRW) detection focuses on the challenge of identifying sparse targets within gigapixel-scale images. Unlike traditional benchmarks such as ImageNet [7], HRW scenes are characterized by extremely low target density (often below 15% coverage) and highly diverse backgrounds, as highlighted in PANDA [27]. Although mainstream detectors, including Faster R-CNN [13] and Deformable Convolutional Networks [5], perform well in dense environments, their computational burden grows rapidly with image resolution, leading to unacceptable latency. Shufflenet [32] improves efficiency by employing grouped convolutions to reduce complexity, yet it lacks mechanisms for selective computation across image regions. DynamicDet [19] introduces adaptive inference by routing inputs through different pathways based on estimated difficulty, but its binary partitioning strategy is unsuitable for HRW scenarios and the dual-branch design adds unnecessary overhead. Similarly, DGE [24] has been utilized to strengthen vision transformers, but its rigidity limits adaptability to the unique demands of HRW detection.

Vision transformers for detection. Transformers have achieved remarkable success in the field of natural language processing (NLP) and Vision Transformers (ViTs) are likewise revolutionizing detection by modeling long-range dependencies through attention. Swin [21] mitigates the rising computational cost of large images with window-based attention, yet falls short of the contextual demands of HRW images. Dual-branch models like DBCT-Net [28] fuse global and local features but fail to extract task-relevant information from massive non-RoI regions, leading to inefficiency in HRW detection. They share a common limitation: either treat all tokens equally or focus only on selected RoIs, overlooking the small and hard targets that are prevalent in HRW scenarios.

Sparse attention methods. To enhance the effectiveness of attention mechanisms in HRW detection, numerous sparsification approaches have been proposed. XAttention [30] leverages the sum along the anti-diagonal of the attention matrix to selectively compute more important blocks, which binds queries and key–value pairs together and limits expressiveness. Linformer [26] applies low-rank projections to compute attention in reduced dimensions, trading semantic richness for lower cost. DSA [22] attempts top-k sparsification based on attention weights from q–k dot products, but the computational overhead of the dot product itself is prohibitive in HRW scenarios.

3 Method

3.1 Overview of BiHashFormer

In the main network, input image is partitioned into non-overlapping patches of size ($\frac{H}{4} \times \frac{W}{4} \times 48$) where H and W denote the original size of input. Through linear embedding layer, the number of channel is projected into C in the first stage. In subsequent stages, the width and height of the feature maps are progressively halved while channel doubled by merging patches, reshaping the size into ($\frac{H}{8} \times \frac{W}{8} \times 2C$) for RoI Selector. The RoI Selector segments the feature map into $w \times w$ windows, further categorizing them into RoI and non-RoI regions based on variance, as shown in Figure 2. RoIs are fed into the compression branch for feature extraction, which accounts for the major computational cost of the model; non-RoI regions are processed in the compensation branch, where features are first filtered and then extracted to recover information from small and hard targets. The features extracted from the two branches are directly merged. The RoI Selector together with the

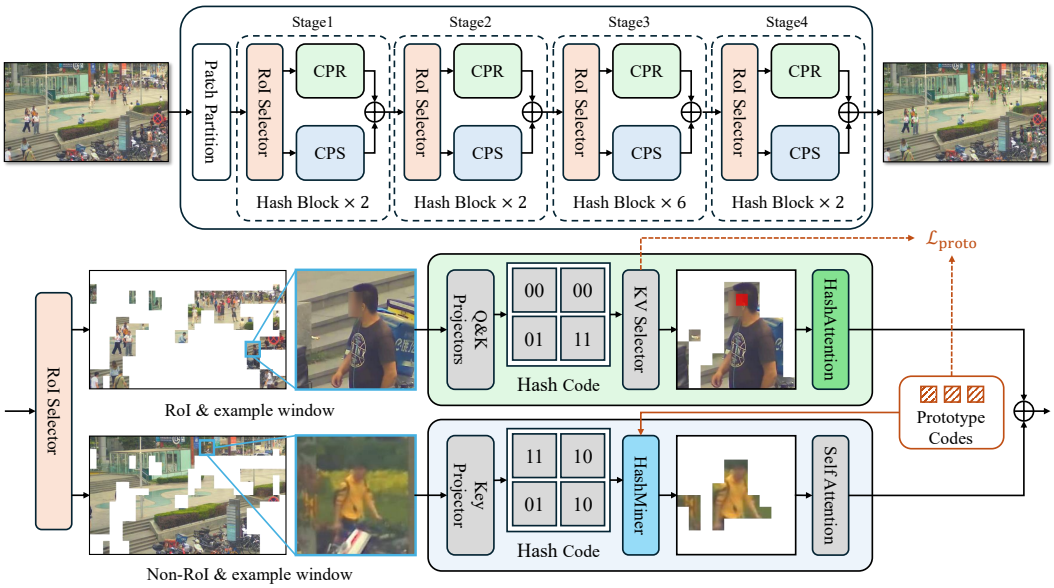


Figure 2: **Architecture overview of BiHashFormer.** CPR: **C**om**PR**ession branch. CPS: **C**om**P**en**S**ation branch. RoI Selector devides images into RoI for CPR and Non-RoI for CPS. Hash Code in the figure demonstrate an example of 2-bit for Eq. (9). An example of HashAttention is listed in the green box: for the red query, pixels in the image are selected by KV Selector. Prototype codes are trained in CPR through $\mathcal{L}_{proto}$, then used in HashMiner. HashMiner selects tokens for further feature extraction in CPS.

two branches forms a Hash Block, which serves as the basic unit of each stage. The model consists of four stages, with the number of Hash Blocks in each stage (i.e., the stage depth) set to $[2, 2, 6, 2]$.

In the compression branch, query and key will be separately projected into low dimensional Hamming space. HashAttention is then applied to RoIs window by window to extract features. Specifically, all queries participate in the attention computation, but the number of interacting key–value pairs is halved according to the Hamming distance. We illustrates the selected k-v pairs for an example query (colored red) in Figure 2. Prototype codes is learned within this branch through $\mathcal{L}_{proto}$ (see Section 3.5). In the compensation branch, only the keys are projected. HashMiner then identifies the target-associated tokens using prototype codes and applies conventional self-attention to extract features. An example window is shown in Figure 2.

3.2 RoI Selector

Variance-based token scoring. Identifying windows most likely to contain targets (i.e., RoIs) presents a critical challenge, which is the decisive factor for the computational efficiency of the compression branch. Observing that background regions (e.g., sky, plaza and roads) typically exhibit more uniform pixel distributions (i.e., lower variance) than object regions, we score each window using the variance of pixels. For the input feature map $\mathbf{F}$ in shape $(B \times H \times W \times C)$, we compute its average pooled result in shape $(B \times \frac{H}{w} \times \frac{W}{w} \times C)$ and interpolate it back first:

$$\mathbf{F}_{\text{pooled}} = \text{AvgPool}_{w \times w}(\mathbf{F}), \quad \hat{\mathbf{F}} = \text{Interpolate}(\mathbf{F}_{\text{pooled}}, w) \tag{1}$$

Here we obtain mean feature $\hat{\mathbf{F}}$, taking neighboring windows into account. Then we partition and flatten the residual for each window as below, where k means k^{th} window:

$$\hat{\mathbf{R}} = \mathbf{F} - \hat{\mathbf{F}}, \quad \mathbf{W}_k = \text{Flatten}(\hat{\mathbf{R}}_k) \tag{2}$$

A simple MLP f_{score} is employed to transform the token sequence into scores corresponding to each window, which are normalized subsequently:

$$s_k = f_{score}(\mathbf{W}_k), \quad \hat{s}_k = \frac{\exp(s_k)}{\sum_{j=1}^{N_w} \exp(s_j)} \tag{3}$$

Top-k selection. Assigning too many windows to the RoI leads to a surge in the computational cost of the compression branch, whereas too few windows cause most targets to be routed to the compensation branch, resulting in insufficient feature extraction. To this end, we use $\hat{\rho}$ generated by a MLP to select the N highest-scoring windows for the CPR:

$$N = \lceil \hat{\rho} \cdot N_w \rceil \tag{4}$$

Here N_w is the total number of winodws and N highest-scoring windows will be assigned as RoI. Taking $N_w = 110, \hat{\rho} = 0.15$ as an example, $N = \lceil 0.15 \cdot 110 \rceil = 17$, thus top-17 windows ranked by score will be sent to the compression branch and the rest for compensation.

Generating $\hat{\rho}$. The foreground proportion ρ measures spatial complexity. For boxes BBox_i in the input image coordinate system and image region Ω of size $H \times W$, it is the clipped union area divided by image area:

$$\rho = \frac{\text{Area}(\bigcup_i (\text{BBox}_i \cap \Omega))}{H \cdot W} \tag{5}$$

In order to obtain such a feature of the overall image, we used feature map for extraction through a MLP. Because the MLP's input dimensions are fixed and restricted to one-dimensional vectors, we apply a size-normalization operation to the feature maps:

$$\mathbf{F}_{\text{pooled}} = \text{AdpPool}(\mathbf{F}), \quad \mathbf{F}_{\text{flat}} = \text{Flatten}(\mathbf{F}_{\text{pooled}}) \tag{6}$$

The adaptive average pooling (AdpPool) downsamples the input feature map $\mathbf{F}$ (derived from the backbone) to a predetermined, fixed spatial resolution regardless of the original size of input images. Here the size for 4 stages is $[32^2,\ 16^2,\ 8^2,\ 4^2]$. The subsequent Flatten layer then serializes the two-dimensional output $\mathbf{F}_{\text{pooled}}$ into a one-dimensional feature vector $\mathbf{F}_{\text{flat}}$. The dimensionality of this vector is inherently determined by the output size chosen for the adaptive pooling and the number of feature channels C, and is strategically configured based on the specific stage within the BiHashFormer backbone where RoI Selector is integrated (larger dimensions for early layers capturing fine details, smaller for deeper semantic layers).

After obtaining the $\mathbf{F}_{\text{flat}}$, we employ a lightweight two-layer MLP, f_{pred}, to predict the foreground proportion $\hat{\rho}$ for RoI Selector in each stage as shown in Figure 2, marked as stage t:

$$\hat{\rho^t} = \sigma(f_{\text{pred}}(\mathbf{F}^t_{\text{flat}})) \tag{7}$$

where layer of Sigmoid ensures that $\hat{\rho}$ lies between 0 and 1. $\hat{\rho}$ will be used for top-k selection, shown in Equation (4). The MLP is trained by $\mathcal{L}_{select}$ below, where γ lays in the interval $(0, 1)$, motivating smaller $\hat{\rho}$ through L_1 penalty. We set γ to 0.1 through experiments.

$$\mathcal{L}_{\text{select}} = (\rho - \hat{\rho})^2 + \gamma \cdot \hat{\rho} \tag{8}$$

3.3 HashAttention

Q&K projectors. To evaluate token similarity efficiently in HRW detection, we project queries and keys into a low dimensional Hamming space. Compared with cosine similarity, its compact binary codes allow inexpensive integer operations:

$$z = \mathbf{P}^T_{Q/K} \cdot x, \quad b = \mathbf{1}[z > 0] \tag{9}$$

Here $x \in \mathrm{R}^d$ is a query or key, $\mathbf{P}_{Q/K} \in \mathrm{R}^{d \times k}$ and $k = 32$. Separate Q/K projectors produce elementwise binary codes $b \in \{0, 1\}^k$, with zero mapped to bit 0. Gradients use the Straight-Through Estimator (STE):

$$\frac{\partial \mathcal{L}}{\partial z} =: \frac{1}{2} \frac{\partial \mathcal{L}}{\partial b} \tag{10}$$

Notably, instead of projecting the raw tokens, we project the queries and keys. This is explained below.

Compress KV for each Q. A larger window size can better accommodate complex HRW scenarios. However, for the tokens within each window, this implies greater interference when computing attention. We expect each token to engage in attention interactions only with the tokens it truly requires. This implies that we must know what information each token needs, as well as what information it contains. This precisely corresponds to the query and key in the attention mechanism, thereby explaining why we choose them for projection. We employ the Hamming distance to measure the similarity between keys and queries:

$$dis_H(b_q, b_k) = ||b_q \oplus b_k||_1 \tag{11}$$

Here $\oplus$ denotes bitwise XOR, and $||\cdot||_1$ represents the count of different positions. A larger Hamming distance indicates that the two vectors are farther apart in the Hamming space, implying a greater discrepancy between the information required by the query and the information contained in the key. For a given query q, KV Selector selects the half of the keys with the smallest Hamming distance in the window, marked as $\mathbf{K}_{0.5}$. Each value is bound to the corresponding key from the same token, and compute the HashAttention (HA) as:

$$\text{HA}(q, \mathbf{K}_{0.5}, \mathbf{V}_{0.5}) = \text{softmax}\left(\frac{q\mathbf{K}_{0.5}^\top}{\sqrt{d_{\text{key}}}}\right)\mathbf{V}_{0.5} \tag{12}$$

HA is calculated in windows. Taking the case in Figure 2 as an example, we illustrate the projection results of four keys, simplified into a 2-bit form: 00, 00, 01, and 11. Considering a query projected as 11, it will select the key–value pairs corresponding to 01 and 11 (they get smaller dis_H as 1 and 0), and compute HashAttention in the original space.

3.4 HashMiner

K projector. To accommodate the characteristic of HRW scenarios where the foreground occupies only a small proportion, most windows are designated as non-RoI to reduce computational cost, as illustrated in Figure 2. However, small or difficult targets are often submerged within non-RoI regions, calling for an efficient method to extract.

As mentioned earlier, since the key contains the intrinsic information of the token, it is well-suited for selection. In the compression branch, the K projector has already been sufficiently trained. Therefore, we directly adopt the pre-trained projection matrix $\mathbf{P}_K$ to obtain the low dimensional Hamming features of the keys in the compensation branch (follows Equation (9)), thereby achieving parameter reduction. In this branch, the parameters of $\mathbf{P}_K$ are frozen.

Selection criteria. After obtaining the mapping of keys in the low dimensional Hamming space, we further need to identify which codes correspond to the semantics of "person". After obtaining these cluster centers c_i, HashMiner can rapidly filter out the tokens requiring feature compensation based on the given radius $r = 4$:

$$\mathcal{X}_{comp} = \left\{ x \;\middle|\; \min_{1 \le i \le N_c} dis_H(b_K(x), c_i) \le r \right\} \tag{13}$$

Here $b_K(x)$ is the binary code of token x's key and $N_c = 3$ is the number of prototype codes, chosen by ablation. The following subsection describes their training.

3.5 Prototype codes

Learning from the compression branch. Prototype codes are obtained by binarizing several trainable tokens into 0–1 representations, which need training data. Since the two branches within the same Hash Block share the key projector, the Hamming space corresponding to the keys remains consistent. As the compression branch processes RoIs that already contain rich target features, where prototype codes can be efficiently learned from. We employ the keys selected by the KV Selector to train the prototype codes, which reduces computation. Moreover, visualization experiments (Figure 3) demonstrate that the selected keys contain richer target information.

Table 1: **Computational cost and accuracy comparison of different models on PANDA.** GFLOPs are calculated exclusively for the backbone and neck when applicable. "F", "B" and "O" denote foreground, background and overall respectively. Baseline sources: [1, 3, 6, 11, 14–16, 18, 21, 31, 33, 34].

Method	Backbone	GFLOPs			AP			
		F	B	O	50	small	medium	large
CascadeRCNN	ResNet-101	15.54	295.24	310.78	-	0.227	0.579	0.765
ClusDet	ResNet-50	10.35	196.71	207.07	0.718	0.219	0.696	0.782
DMNet	ResNet-50	10.35	196.71	207.07	0.540	0.119	0.371	0.714
FasterRCNN	ResNet-50	10.35	196.71	207.07	0.705	0.203	0.712	0.760
GigaDet	CSP-DarkNet-53	**4.61**	87.59	92.20	0.684	0.210	0.599	0.762
PAN	ResNet-50	10.35	196.71	207.07	0.715	0.256	0.719	0.768
RetinaNet	ResNet-101	15.77	299.62	315.39	-	0.221	0.561	0.740
RT-DETR	ResNet-50	10.35	196.71	207.07	0.612	0.137	0.532	0.735
Dynamic-Head	Swin-T	<u>5.74</u>	109.1	114.8	0.592	0.165	0.537	0.694
Dynamic-Head	SparseFormer	6.29	58.35	64.64	0.771	0.364	0.740	**0.863**
Dynamic-Head	BiHashFormer (Ours)	7.11	**38.35**	**45.46**	<u>0.799</u>	<u>0.565</u>	<u>0.783</u>	0.841
DINO	ResNet-50	6.21	118.02	124.24	0.542	0.289	0.530	0.592
DINO	Swin-T	6.64	126.19	132.84	0.606	0.367	0.612	0.649
DINO	SparseFormer	6.90	68.81	75.71	0.780	0.508	0.781	0.823
DINO	BiHashFormer (Ours)	7.30	<u>40.01</u>	<u>47.31</u>	**0.810**	**0.591**	**0.792**	<u>0.847</u>

Training for prototype codes. Inspired by clustering methods in machine learning, we expect the clusters corresponding to prototype codes to be as compact as possible within classes and as dispersed as possible across classes. The intra-class loss is defined as follows:

$$d_j = \min_{1 \le i \le N_c} \frac{dis_H(b_{k_j}, c_i)}{k}, \quad \mathcal{L}_{intra} = \frac{1}{N_{\text{key}}} \sum_{j=1}^{N_{\text{key}}} d_j \tag{14}$$

b_{k_j} is the binary code of selected key j, c_i the ith prototype code, and N_{key} the number of selected keys. We use $k = 32$ and the Hamming distance in Equation (11).

According to the triangle inequality, ensuring that the distance between prototypes is greater than $2r$ guarantees that their covered spaces do not overlap. Based on this, we introduce an inter-class penalty term to encourage prototypes to encompass richer semantics:

$$\mathcal{L}_{cross} = \frac{1}{N_c(N_c-1)} \sum_{p \ne t} \max\{0, 2r + 1 - ||c_p - c_t||_1\} \tag{15}$$

Here $r = 4$ and $N_c = 3$ is the same as Equation (13). At initialization, each code has approximately k/N_c ones in a disjoint bit block and zeros elsewhere, separating the prototypes. For example, with $k = 3, N_c = 3$, the initialized codes are 100, 010 and 001. Prototype latents are optimized through their binary codes using the STE in Equation (10); inference uses hard binary codes:

$$\mathcal{L}_{proto} = \mathcal{L}_{intra} + \mathcal{L}_{cross} \tag{16}$$

4 Experiment

4.1 Datasets

Our primary evaluation is conducted on the PANDA [27] dataset, the first human-centric gigapixel-level benchmark featuring 13 training and 5 testing scenes with approximately 16 million annotated bounding boxes. The average image resolution of this dataset, 25,000 × 16,000 pixels, presents unprecedented computational demands, establishing a rigorous testbed for efficiency assessment.

4.2 Evaluation metrics

Model efficiency is quantified using FLOPs and standard COCO metrics, with particular emphasis on AP_{50}, AP_{small} ($< 96 \times 96$), AP_{medium} ($96 \times 96 \sim 288 \times 288$) and AP_{large} ($> 288 \times 288$). We further record the FLOPs on foreground (FLOPs-F) and background (FLOPs-B) to provide a more fine-grained evaluation of model. To ensure fair comparison, all detectors process uniformly cropped 1280×800 windows from test images. Given the dynamic token selection mechanism in BiHashFormer, its FLOPs are estimated by averaging measurements across the test set. All implementations utilized the MMDetection [2] framework under standardized 36-epoch training protocols.

4.3 Data analysis on PANDA

As presented in Table 1, traditional detectors consistently exceed 200 GFLOPs for backbone computation on PANDA [27], whereas modern Transformer-based approaches like Dynamic-head [6] and DINO [33] reduces computational costs to sub-100 GFLOPs. SparseFormer achieves further reduction to 75.71 GFLOPs through fixed-ratio window selection while delivering significant AP gains.

BiHashFormer is combined with Dynamic-head [6] and DINO [33] to evaluate its performance. The results indicate that BiHashFormer achieves superior detection performance with significantly lower computation. Compared with SparseFormer [16], our backbone achieves a 37.5% reduction in computational cost on DINO while improving AP_{50} by 3.85%. In comparison with Swin-T [21], the reduction in computation reaches 64.4%. Notably, employing BiHashFormer on DINO yields a 16.34% improvement in AP_{small} over SparseFormer, and this gain further increases to 55.22% on Dynamic-Head. It can be observed that the proportion of foreground computation in BiHashFormer has significantly increased, rising from 9.1% in SparseFormer to 15.43%. This indicates that our backbone achieves better efficiency.

In addition, we record the computational cost proportion of the two branches in BiHashFormer: 38.12 GFLOPs for the compression branch and 8.29 GFLOPs for the other. It can be observed that the compression branch constitutes the major computational overhead. Although the compensation branch receives a larger input region (85% of all), it accounts for only 17.9% of the branches computation, which highlights the efficiency of HashMiner's selection.

4.4 Study on HashAttention

Visualization of HashAttention. We selected several scenarios with different target densities and extracted representative queries from each, visualizing their interacted key–value pairs as shown in Figure 3. It can be observed that both foreground and background queries pay greater attention to the key–value pairs corresponding to humans. This strongly supports the effectiveness of employing the KV Selector to choose keys during prototype code training.

Further, the figure reveals an interesting pattern: queries from the head tend to interact with key–value pairs covering the entire body, whereas queries from the body and background focus more on the key–value pairs located at the boundary between foreground and background. This may be because the head, as a key marker of a person, serves as a suitable anchor for feature extraction, linking other parts of the body to form the entirety of the detection target. In contrast, the body and the surrounding environment focus more on differences, assisting the backbone in distinguishing between target and background features. Thereby backbone can separate them in the Hamming space better.

Different mechanisms of attention. For HashAttention, we need to verify whether it outperforms the classical Attention in HRW scenarios. In addition, the idea of selecting half of the keys for a query also requires examination. To this end, we retain Hamming distance as the criterion and explore different selection strategies. The strategies are marked as $A \rightarrow \rho\%\ B$: using all A, selecting $\rho\%\ B$ for each A. The classical attention can thus be denoted as $Q \rightarrow 100\%\ K$.

According to Table 2, HashAttention outperforms classical Attention in both computational cost and detection performance. Line $K \rightarrow 50\%\ Q$ demonstrates that the supply–demand relationship between Q and k is not symmetric, swapping the order results in a significant performance degradation although the computational cost remains nearly unchanged. Experiments conducted with different values of ρ indicate that 50% is an appropriate choice.

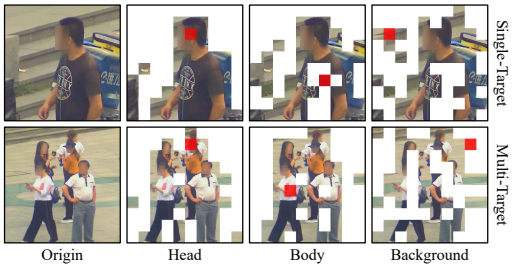


Figure 3: **Comparison of interacted KV for the chosen Q.** The pixel in red represent the query. Rows represent two target densities, while columns correspond to three types of queries.

Table 2: **AP and FLOPs with different selection strategies.** FLOPs: calculation of the compression branch. AP denotes the performance of the complete model. $A \rightarrow \rho\%\ B$: using all A, selecting $\rho\%\ B$ for each A. HA: HashAttention. Att: classical attention. M/L: medium/large.

Selection	GFLOPs	AP_{50}	AP_M	AP_L
$K \rightarrow 50\%\ Q$	38.14	0.785	0.775	0.815
$Q \rightarrow 20\%\ K$	**16.97**	<u>0.803</u>	0.788	0.836
$Q \rightarrow 50\%\ K$	<u>38.12</u>	**0.810**	<u>0.792</u>	**0.847**
$Q \rightarrow 80\%\ K$	60.28	0.799	**0.793**	0.828
$Q \rightarrow 100\%\ K$	80.11	0.797	0.783	<u>0.837</u>

4.5 Study on the compensation branch

Necessity of the compensation branch. Since the model requires at least one inference pathway and the compression branch serves as the main branch of BiHashFormer, we conduct an ablation study on the alternative compensation branch. First, we disabled this branch during inference. Although this reduced its computational cost, it significantly degraded detection performance, particularly for small objects, as evidenced by nearly a 51.4% drop in AP_{small} in Table X3 in the supplementary material.

Subsequently, we replaced this branch with a classical attention module for feature compensation. Specifically, the entire image was first down-sampled via average pooling with a reduction factor equal to the window size w. Attention computation was then performed on the down-sampled image, and the results from non-RoI regions were mapped back to the original resolution, forming a complete feature map together with the features obtained from the compression branch. This ensures that the compression branch remains unaffected while keeping the computational cost of the replacement comparable to the compensation branch. We define the efficiency as following to demonstrate the efficiency of different methods in compensating features for small objects, where AP^{NA}_{small} refers to the 0.287 in Table X3:

$$\text{Eff} = 100 \cdot (AP_{small} - AP^{NA}_{small})/\text{GFLOPs} \tag{17}$$

These two ablation experiments demonstrate the necessity of the compensation branch: it cannot be replaced by simple feature extraction methods, nor can it be directly removed.

Selection methods compared with HashMiner. We aim to determine whether the effect brought by HashMiner is truly efficient or merely redundant. To this end, we experimented with several alternative selection methods and focused on comparing their computational cost, latency and AP_{small}.

The results in Table X1 show that whether using random selection, training a simple MLP to decide selection, or leveraging gradient-based criteria, none of these approaches outperform HashMiner in compensating features for small objects, proved by AP_{small}. Moreover, both computational cost and latency remaining well-controlled on HashMiner.

4.6 Ablation study

Ablation study on adaptive sparsity rates. SparseFormer [16] also adopts the concept of sparsity rate, but it uses a fixed parameter preset to 0.7. We choose this parameter from $[0.05, 0.147, 0.3, 0.5, 0.7, 0.9]$ for comparison, where 0.147 is the average $\hat{\rho}$ on BiHashFormer. The experiments are conducted with both DINO [33] and Dynamic-Head [6].

As shown in Figure 4, thanks to the compensation branch, our model performs well even at small values of ρ, whereas the detection accuracy of SparseFormer drops significantly. With the same sparsity rate as BiHashFormer, its AP_{50} is consistently lower. This highlights the importance of dynamic sparsity in addressing target number variations in HRW scenarios.

Ablation study on the dimension k. The dimension k of the projected Hamming space influences the two branches. It determines how much semantic information is contained in each bit of the hash code. An excessively large k leads to overly fine semantic distinctions, undermining the purpose of a

Table 3: **AP and FLOPs with different k.** Smaller k leads to lower efficiency and worse results.

k	F	B	O	AP_{50}
16	**6.84**	48.17	55.01	0.801
32	7.30	40.01	47.31	**0.810**
64	7.00	**39.95**	**46.95**	0.805

Table 4: **AP and FLOPs with different prototype numbers.** GFLOPs-CPS: compensation branch only.

N_c	F	CPS	AP_{50}	AP_{small}
1	**7.12**	**5.86**	0.807	0.498
3	7.30	40.01	**0.810**	**0.591**
5	7.29	17.41	0.806	0.582

low-rank feature space. An excessively small k results in semantic entanglement, making it difficult to identify clear boundaries. This is validated in Table 3.

Ablation study on γ. We select five values of γ for ablation. Increasing γ reduces computational cost at the expense of AP50, highlighting the trade-off between efficiency and accuracy. This is shown in Table X4 in the supplementary material.

Ablation study on the window size w. To handle the more complex scenarios and larger scale variations in HRW shots, we increase the window size, thereby allowing attention interactions among tokens to cover a broader range. In Swin, increasing w leads to a sharp rise in the computation of attention. We alleviate this issue through RoI Selector and HashAttention. Among the four tested values [7, 14, 21, 28], the best-performing setting is 21, as reported in Table X2.

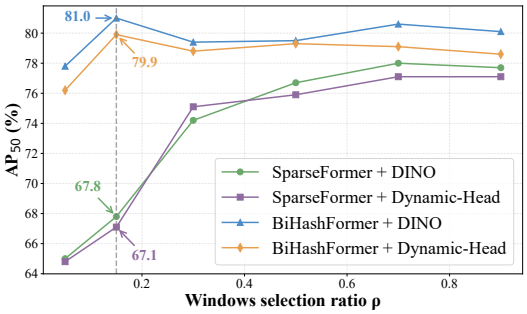


Figure 4: **AP_{50} vs. windows selection ratio ρ on SparseFormer and BiHashFormer.** The values annotated correspond to the AP_{50} at $\rho = 0.147$.

Ablation study on the number of prototype codes. Similar to the K-Means clustering method, the number of prototype codes has a substantial impact on the final clustering performance. According to the results in Table 4, we set this parameter to 3 for the optimal effect of the compensation branch. The prototypes retrieve foreground tokens. The detector head predicts their categories.

4.7 Supplementary experiments

Section 2 of the supplementary material reports inference latency, DOTA/VisDrone generalization, and STE training stability. On COCO, BiHashFormer with Mask R-CNN and a 3× schedule reaches 48.3 AP using three prototypes, versus 46.0 for Swin-T, with 35.6% fewer backbone/neck FLOPs. Across 80 categories, three prototypes give the best accuracy among the tested counts, supporting class-agnostic foreground retrieval. With the same DINO head, BiHashFormer reaches 0.810 AP_{50} at 47.31 GFLOPs and 2.8 s, outperforming the complete Linformer and XAttention configurations. With matched DINOv3 [23] ViT-S+/16 initialization and 36-epoch training, it gains 4.0 AP_{50} and 11.6 AP_{small} points over the baseline with 64.4% fewer FLOPs. On eight RTX 4090 GPUs with one image per GPU, training takes 19.7 h versus SparseFormer's 23.2 h and reaches 0.75 AP_{50} five epochs earlier. Full protocols and results appear in supplementary Tables X9–X13.

5 Conclusion

By deeply integrating hashing concepts with object detection, we propose BiHashFormer, a dual-branch vision transformer. It employs RoI Selector to sparsely select RoI, leverages Hamming distance to refine the attention mechanism into a sparsified HashAttention and utilizes HashMiner to dynamically compensate for object features in non-RoI regions. Thereby it reduces FLOPs by 64.4% versus Swin-T [21] and 37.5% versus SparseFormer [16] while achieving 81.0% AP_{50} on PANDA [27]. This dual-branch paradigm establishes a new standard for efficient large-scale vision systems, particularly for real-time applications with constrained resources.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China (NSFC; Nos. 82441013 and 62572268); the National Science and Technology Major Project (No. 2022ZD0119402); the Ministry of Science and Technology of China (contract No. 2024YFB2809103); the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM904); the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C01142); and the Tsinghua University – Migu Xinkong Culture Technology (Xiamen) Co., Ltd. Joint Research Center for Intelligent Light Field and Interaction Technology, Tsinghua University - Joint R&D Project (R2511DKW) – Phase 2. We also acknowledge the KLATASDS-MOE.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open MMlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[3] Kai Chen, Zerun Wang, Xueyang Wang, Dahan Gong, Longlong Yu, Yuchen Guo, and Guiguang Ding. Towards real-time object detection in gigapixel-level video. *Neurocomputing*, 477:14–24, 2022.

[4] Hao Chen, Yuqi Hou, Chenyuan Qu, Irene Testini, Xiaohan Hong, and Jianbo Jiao. 360+ x: A panoptic multi-modal scene understanding dataset. In *CVPR*, pages 19373–19382, 2024.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[6] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, pages 7373–7382, 2021.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[8] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning RoI transformer for detecting oriented objects in aerial images. *arXiv preprint arXiv:1812.00155*, 2018.

[9] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, et al. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7778–7796, 2021.

[10] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Lin, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *ICCVW*, 2019.

[11] Jiahao Fan, Huabin Liu, Wenjie Yang, John See, Aixin Zhang, and Weiyao Lin. Speed up object detection on gigapixel-level images with patch arrangement. In *CVPR*, pages 4653–4661, 2022.

[12] Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. RMT: Retentive networks meet vision transformers. In *CVPR*, pages 5641–5651, 2024.

[13] Ross Girshick. Fast r-CNN. In *ICCV*, pages 1440–1448, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Changlin Li, Taojiannan Yang, Sijie Zhu, Chen Chen, and Shanyue Guan. Density map guided object detection in aerial images. In *CVPRW*, pages 190–191, 2020.

[16] Wenxi Li, Yuchen Guo, Jilai Zheng, Haozhe Lin, Chao Ma, Lu Fang, and Xiaokang Yang. SparseFormer: Detecting objects in HRW shots via sparse vision transformer. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4851–4860, 2024.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[19] Zhihao Lin, Yongtao Wang, Jinhe Zhang, and Xiaojie Chu. DynamicDet: A unified dynamic architecture for object detection. In *CVPR*, pages 6282–6291, 2023.

[20] Tianwei Lin, Wenqiao Zhang, Sijing Li, Yuqian Yuan, Binhe Yu, Haoyuan Li, Wanggui He, Hao Jiang, Mengze Li, Xiaohui Song, et al. HealthGPT: A medical large vision-language model for unifying comprehension and generation via heterogeneous knowledge adaptation. *arXiv preprint arXiv:2502.09838*, 2025.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[22] Liu Liu, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yufei Ding, and Yuan Xie. Dynamic sparse attention for scalable transformer acceleration. *IEEE Transactions on Computers*, 71(12):3165–3178, 2022.

[23] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[24] Lin Song, Songyang Zhang, Songtao Liu, Zeming Li, Xuming He, Hongbin Sun, Jian Sun, and Nanning Zheng. Dynamic grained encoder for vision transformers. *Advances in Neural Information Processing Systems*, 34:5770–5783, 2021.

[25] Fredrik Svanström, Cristofer Englund, and Fernando Alonso-Fernandez. Real-time drone detection and tracking with visible, thermal and acoustic sensors. In *ICPR*, pages 7265–7272. IEEE, 2021.

[26] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[27] Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, et al. PANDA: A gigapixel-level human-centric video dataset. In *CVPR*, pages 3268–3278, 2020.

[28] Quanli Wang, Xin Jin, Qian Jiang, Liwen Wu, Yunchun Zhang, and Wei Zhou. DBCT-Net: A dual branch hybrid CNN-transformer network for remote sensing image fusion. *Expert Systems with Applications*, 233:120829, 2023.

[29] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. DOTA: A large-scale dataset for object detection in aerial images. In *CVPR*, pages 3974–3983, 2018.

[30] Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring. *arXiv preprint arXiv:2503.16428*, 2025.

[31] Fan Yang, Heng Fan, Peng Chu, Erik Blasch, and Haibin Ling. Clustered object detection in aerial images. In *ICCV*, pages 8311–8320, 2019.

[32] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.

[33] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[34] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024.

[35] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 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 can accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 reason behind a pattern in section 4.4 need further investigation.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: The proof is correct.

Guidelines:

- The answer [N/A] 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: Hyper-parameters are given.

   Guidelines:

   - The answer [N/A] 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: [No]

Justification: Codes in PyTorch are given in the supplementary material. But the complete code has not been released yet.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: They are in the section 4.

   Guidelines:

   - The answer [N/A] 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: Some experiments take the average value of many runs.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: AGX Orin and RTX 4090 are used.

   Guidelines:

   - The answer [N/A] 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: We reviewed the guidelines and found no conflicts.

   Guidelines:

   - The answer [N/A] 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: [Yes]

    Justification: They are both discussed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] 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: All reused assets are cited with their sources.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: New assets are not released yet.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: N/A

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.