DenoiseRep: Denoising Model for Representation Learning

Zhengrui Xu[1] † zrxu23@bjtu.edu.cn
Guan'an Wang †‡ guan.wang0706@gmail.com
Xiaowen Huang [1,2,3] * xwhuang@bjtu.edu.cn
Jitao Sang [1,2,3] jtsang@bjtu.edu.cn

[1]**School of Computer Science and Technology, Beijing Jiaotong University**
[2]Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University
[3]Key Laboratory of Big Data & Artificial Intelligence
in Transportation(Beijing Jiaotong University), Ministry of Education

Abstract

The denoising model has been proven a powerful generative model but has little exploration of discriminative tasks. Representation learning is important in discriminative tasks, which is defined as *"learning representations (or features) of the data that make it easier to extract useful information when building classifiers or other predictors"* [4]. In this paper, we propose a novel Denoising Model for Representation Learning (*DenoiseRep*) to improve feature discrimination with joint feature extraction and denoising. *DenoiseRep* views each embedding layer in a backbone as a denoising layer, processing the cascaded embedding layers as if we are recursively denoise features step-by-step. This unifies the frameworks of feature extraction and denoising, where the former progressively embeds features from low-level to high-level, and the latter recursively denoises features step-by-step. After that, *DenoiseRep* fus es the parameters of feature extraction and denoising layers, and *theoretically demonstrates* its equivalence before and after the fusion, thus making feature denoising computation-free. *DenoiseRep* is a label-free algorithm that incrementally improves features but also complementary to the label if available. Experimental results on various discriminative vision tasks, including re-identification (Market-1501, DukeMTMC-reID, MSMT17, CUHK-03, vehicleID), image classification (ImageNet, UB200, Oxford-Pet, Flowers), object detection (COCO), image segmentation (ADE20K) show stability and impressive improvements. We also validate its effectiveness on the CNN (ResNet) and Transformer (ViT, Swin, Vmamda) architectures. Code is available at https://github.com/wangguanan/DenoiseRep.

1 Introduction

Denoising Diffusion Probabilistic Models (DDPM) [21] or Diffusion Model for short have been proven to be a powerful generative model [5]. Generative models can generate vivid samples (such as images, audio and video) by modeling the joint distribution of the data $P(X, Y)$, where X is the sample and Y is the condition. Diffusion models achieve this goal by adding Gaussian noise to the data and training a denoising model of inversion to predict the noise. Diffusion models can generate multi-formity and rich samples, such as Stable diffusion [50], DALL [47] series and Midjourney, these powerful image generation models, which are essentially diffusion models.

†Equal Contribution.
‡Project Lead.
*Corresponding Author.

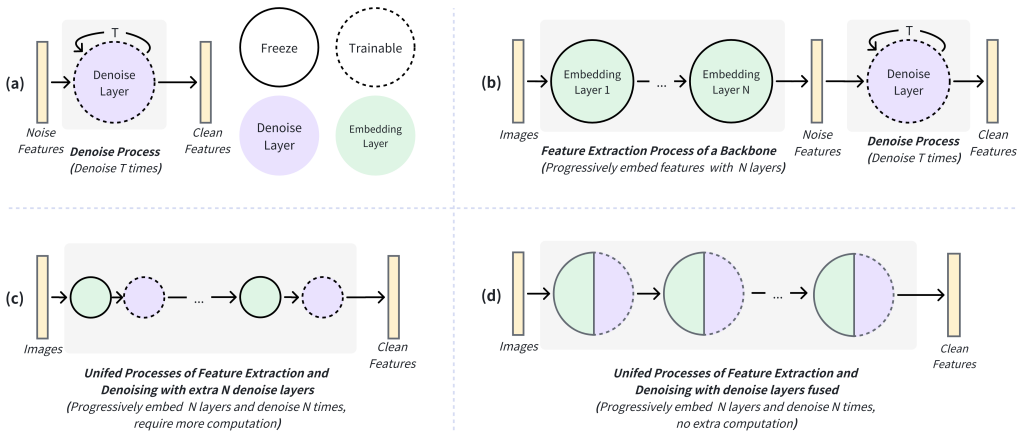

Figure 1: A brief description of our idea. (a) A typical denoising model for generative tasks recursively applies a denoising layer. (b) A naive idea that applies a denoising strategy to a discriminative model is applying a recursive denoise layer on the feature of a backbone and taking extra inference latency. (c,d) Our *DenoiseRep* first unifies the frameworks of feature extraction and denoising in a backbone pipeline, then merges parameters of the denoising layers into embedding layers, making the feature more discriminative without extra latency cost.

However, its application to discriminative models has not been extensively explored. Different from generative models, discriminative models predict data labels by modeling the marginal distribution of the data $P(Y|X)$. Y can be various labels, such as image tags for classification, object boxes for detection, and pixel tags for segmentation. Currently, there are several methods based on diffusion models implemented in specific fields. For example, DiffusionDet [7] is a new object detection framework that models object detection as a denoising diffusion process from noise boxes to object boxes. It describes object detection as a generative denoising process and performs well compared to previous mature object detectors. DiffSeg [52] for image segmentation, which is a method of unsupervised zero-shot sample segmentation using pre-trained models (stable diffusion). It introduces a simple and effective iterative merging process to measure the attention maps between KL divergence and merge them into an effective segmentation mask. The proposed method does not require any training or language dependency to extract the quality segmentation of any image.

The methods above are carefully designed for specific tasks and require a particular data structure. For example, DiffusionDet [7] uses noise boxes and DiffSeg [52] uses noise segmentation. In this paper, we explore a more general conception of how the denoising model can improve representation learning, *i.e. "learning representations (or features) of the data that make it easier to extract useful information when building classifiers or other predictors"* [4], and contribute to discriminative models. We take person Re-Identification (ReID) [66, 3] as a benchmark task. ReID aims to match images of a pedestrian under disjoint cameras, and is suffered by pose, lighting, occlusion and so on, thus requiring more identity-discriminative feature.

A straightforward approach is applying the denoising process to a backbone's final feature [26, 14], reducing noise in the final output and making the feature more discriminative, as Fig. 1(b) shows. However, this way can be computationally intensive. Because the denoising layer needs to be proceeded on the output of the previous one in a recursive and step-by-step manner. Considering that a backbone typically consists of cascaded embedding layers (e.g., convolution layer, multi-head attention layer), we propose a novel perspective: treating each embedding layer as a denoising layer. As shown in Fig. 1(c), it allows us to process the cascaded layers as if we are recursively proceeding through the denoising layer step-by-step. This method transforms the backbone into a series of denoising layers, each working on a different feature extraction level. While this idea is intuitive and simple, its practical implementation presents a significant challenge. The main issue arises from the requirement of the denoising layer for the input and output features to exist in the same feature space. However, in a typical backbone (*e.g.* ResNet [26], ViT [14])), the layers progressively map features

from a low level to a high level. It means that the feature space changes from layer to layer, which contradicts the requirement of the denoising layer.

To resolve all the difficulties above and efficiently apply the denoising process to improve discriminative tasks, our proposed Denoising Model for Representation Learning (*DenoiseRep*) is as below: Firstly, we utilize a well-trained backbone and keep it fixed throughout all subsequent procedures. This step is a free launch as we can easily use any publicly available backbone without requiring additional training time. This approach allows us to preserve the backbone's inherent characteristics of semantic feature extraction. Given the backbone and an image, we can get a list of features. Next, we train denoising layers on those features. The weights of denoising layers are randomly initialized and their weights are not shared. The training process is the same as that in DDPM [21], where the only difference is that the denoising layer in DDPM takes a dynamic $t \in [1, T]$, and our denoising layers take fixed $n \in [1, N]$, where n is the layer index, T is denoise times and N is backbone layer number as shown in Fig. 1(c). Finally, considering that the N denoising layers consume additional execution latency, we propose a novel feature extraction and feature denoising fusion algorithm. As shown in Fig. 1(d), the algorithm merges parameters of extra denoising layers into weights of the existing embedding layers, thus enabling joint feature extraction and denoising without any extra computation cost. We also *theoretically demonstrate* the total equivalence before and after parameter fusion. Please see Section 3.3 and Eq (7) for more details.

Our contributions can be summarized as follows:

(1) We propose a novel Denoising Model for Representation Learning (*DenoiseRep*), which innovatively integrates the denoising process, originating from generative tasks, into the discriminative tasks. It treats N cascaded embedding layers of a backbone as T times recursively proceeded denoising layers. This idea enables joint feature extraction and denoising is a backbone, thus making features more discriminative.

(2) The proposed *DenoiseRep* fuses the parameters of the denoising layers into the parameters of the corresponding embedding layers and *theoretically* demonstrates their equivalence. This contributes to a computation-efficient algorithm, which takes no extra latency.

(3) Extensive experiments on 4 ReID datasets verified that our proposed *DenoiseRep* can effectively improve feature performance in a label-free manner and performs better in the case of label-argumented supervised training or introduction of additional training data. We also extend *DenoiseRep* to large-scale (ImageNet), fine-grained (CUB200, Oxford-Pet, Flowers) image classifications, object detection (COCO) and image segmentation (ADE20K), showing its scalability.

2 Related Work

Generative models learn the distribution of inputs before estimating class probabilities. A generative model learns the data generation process by learning the probability distribution of the input data and generating new data samples. The generative models first estimate the conditional density of categories $P(x|y = k)$ and prior category probabilities $P(y = k)$ from the training data. The $P(x)$ is obtained by the full probability formula. So as to model the probability distribution of each type of data. Generative models can generate new samples by modelling data distribution. For example, Generative Adversarial Networks (GANs) [17, 43, 23] and Variational Autoencoders (VAEs) [24, 53, 69, 64] are both classic generative models that generate real samples by learning potential representations of data distributions, demonstrating excellent performance in data distribution modeling. Recent research has focused on using **diffusion models** for generative tasks. The diffusion model was first proposed by the article [51] in 2015, with the aim of eliminating Gaussian noise from continuous application to training images. The DDPM [21] proposed in 2020 have made the use of diffusion models for image generation mainstream. In addition to its powerful generation ability, the diffusion model also has good denoising ability through noise sampling, which can denoise noisy data and restore its original data distribution.

Discriminative models learn condition distribution, *i.e.* $P(y|x)$, where x is data and y is task-specific features. For example, classification tasks [1, 2, 13] map data to tags, retrieval tasks [36, 62] map data to a feature space where similar data should be near otherwise faraway, detection task [49, 27] map data to space position and size. **Person Re-Identification (ReID)** is a fine-grained retrieval task which identifies individuals among disjoint camera views. Considering its challenge to feature

discrimination, we take ReID as the major benchmark task and the others as auxiliary benchmarks. Existing ReID methods can be grouped into hand-crafted descriptors [35, 42, 65] incorporated with metric learning [25, 34, 71] and deep learning algorithms [58, 57, 56, 44, 16, 10]. State-of-the-art ReID models often leverage convolutional neural networks (CNNs) [28] to capture intricate spatial relationships and hierarchical features within person images. Attention mechanisms [54, 14], spatial-temporal modeling [31, 30], and domain adaptation techniques [9] have further enhanced the adaptability of ReID models to diverse and challenging real-world scenarios.

3 *DenoiseRep*: Denoising Model for Representation Learning

3.1 Review Representation Learning

Representation learning plays a pivotal role in discriminative tasks, which is defined as *"learning representations (or features) of the data that make it easier to extract useful information when building classifiers or other predictors"* [4]. A common architecture of discriminative tasks consists of a vision backbone to extract discriminative features (*e.g.*, ResNet [18], ViT [14]) and a task-specific head that operates on these features (*e.g.* MLP [26] for classification, RCNN [49] for object detection, FCN [40] for segmentation). It is evident that the vision backbone is central to representation learning. In this paper, we introduce a novel Denoising Model for Representation Learning (*DenoiseRep*), which integrates feature extraction and feature denoising within a single vision backbone. This approach aims to enhance the discriminative power of the features extracted.

3.2 Joint Feature Extraction and Feature Denoising

We refer to the diffusion modeling approach to denoise the noisy features through T-steps to obtain clean features. At the beginning, we use the features output from the backbone network as data samples for diffusion training, and get the noisy samples by continuously adding noise and learning through the network in order to simulate the data distribution of its features.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) \coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \tag{2}$$

where X_0 represents the feature vector output by the backbone, t represents the diffusion step size, β_t is a set of pre-set parameters, and X_t represents the noise sample obtained through diffusion process.

In the inference stage, as shown in Fig. 1(b), we perform T-step denoising on the output features, to obtain cleaner features and improve the expressiveness of the features.

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{4}$$

where X_t represents the feature vector output by the backbone in the inference stage. T is the denoising step size, representing the magnitude of the noise. We adjust t appropriately based on different datasets and backbones to obtain the optimal denoising amplitude. According to $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ denoise it step by step, and finally obtains X_0, which represents the clean feature after denoising.

3.3 Fuse Feature Extraction and Feature Denoising

As described in Section 3.2, the proposed method above could effectively improve the discriminability of features. Still, extra inference latency is introduced caused by recursive calling of the denoising layers. To solve the problem, we propose to fuse parameters of feature denoising layers into parameters of existing embedding layers of the backbone. The core idea is to expand the linear layer each transformer encoder block into two branches, one for its original embedding layer and the other for extra denoising layer. As shown in Fig. 2, during the training phase, we freeze the original embedding layers and only train the denoising layers. The training method is consistent with section 3.2, and the features are diffused and fed into the denoising layers. Please refer to Algorithm 1 for

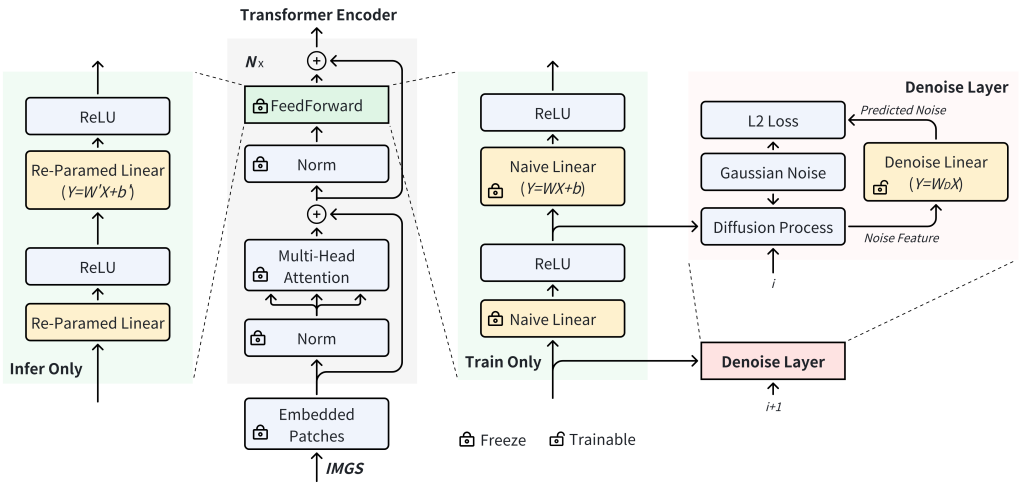

Figure 2: Pipeline of our proposed *DenoiseRep*. ViT consists of N cascaded transformer encoder layers. During the training phase (see the right side "Train Only" process), we freeze the backbone parameters and only train the extra denoising layers. In the inference stage (see the left side "Infer Only" process), we merge the parameters of denoising layers to corresponding encoder layers. So there is no extra inference latency cost. Please find definitions of W, b, W_D, W', i and b' in Algorithm 2.

more details. In the inference stage, we fuse the pre-trained parameters of embedding and denoising layers, merging the two branches into a single branch without additional inference time. Please note that, here we take the transformer architecture as an example, but *DenoiseRep* is suitable for CNN architecture. We demonstrate its scalibity on CNN in experiments. The derivation of parameter merging is as follows:

$$X_{t-1} = \frac{1}{\sqrt{a_t}}(X_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} D_\theta(X_t, t)) + \sigma_t z \tag{5}$$

where $a_t = 1 - \beta_t$, D_θ are the parameters of the prediction noise network.

$$\begin{aligned} Y &= WX + b \\ \frac{1}{\sqrt{a_t}} X_t - X_{t-1} &= \frac{1-a_t}{\sqrt{a_t}\sqrt{1-\bar{a}_t}} D_\theta X_t - \sigma_t z \\ \frac{1}{\sqrt{a_t}} Y_t - Y_{t-1} &= \frac{1-a_t}{\sqrt{a_t}\sqrt{1-\bar{a}_t}} D_\theta Y_t - \sigma_t z \end{aligned} \tag{6}$$

We make a simple transformation of Eq. (5) and multiply both sides simultaneously by W. The simplified equation can be obtained by bringing Y_t in terms of $WX_t + b$:

$$\begin{gathered} Y_{t-1} = [W - C_1(t)WW_D]X_t + WC_2(t)C_3 + b \\ C_1(t) = \frac{1-a_t}{\sqrt{a_t}\sqrt{1-\bar{a}_t}} \qquad C_2(t) = \frac{1-\bar{a_{t-1}}}{1-\bar{a}_t}\beta_t \qquad C_3 = Z \sim N(0, I) \end{gathered} \tag{7}$$

where W_D denotes the parameters of $D_\theta(X_t, t)$, X_t denotes the input of this linear layer, Y_t denotes the output of this linear layer, and Y_{t-1} denotes the result after denoising in one step of Y_t. Due to the cascading relationship of blocks, as detailed in Algorithm. 2, different t values are set according to the order between levels, and the one-step denoising of one layer is combined to achieve the denoising process of $Y_t \to Y_0$, ensuring the continuity of denoising and ultimately obtaining clean features. We split the original single branch into a dual branch structure. During the training phase, the backbone maintains its original parameters and needs to train the denoising module parameters. In the inference stage, as shown on the left side of Fig. 2, we use the method of reparameterization, to replace the original W parameter with W', where $W' = [W - C_1(t)WW_D]$ in Eq. (7), which has the same

Algorithm 1 Training

Input: The number of feature layers in the backbone N, features extracted from each layer $\{F_i\}_{i=1}^{N}$, the denoising module that needs to be trained $\{D_i(\cdot)\}_{i=1}^{N}$.

repeat
  for each $i \in [N, 1]$ **do**
    $t = i$: Specify the diffusion step t for the current layer based on the order of layers.
    $\epsilon \sim N(0, I)$: Randomly sample a Gaussian noise.
    $X_t = \sqrt{\bar{a}_t} F_i + \sqrt{1 - \bar{a}_t}\epsilon$: Forward diffusion process in Eq.(2).
    Take gradient descent step on $\nabla_\theta \left\| \epsilon - D_i(X_t, t) \right\|^2$
  end for
until converged

number of parameters as W, thus achieving the combination of FC operation and denoising without additional time cost. It is a **Computation-free** method.

In Eq. (7), we achieve one-step denoising $Y_t \to Y_{t-1}$. If we need to increase the denoising amplitude, we can extend it to two-step or multi-step denoising. The following is the derivation formula for two-step denoising:

$$\frac{1}{\sqrt{a_t}} Y_t - Y_{t-1} = C_1(t) D_\theta Y_t - \sigma_t z \tag{8}$$

$$\frac{1}{\sqrt{a_{t-1}}} Y_{t-1} - Y_{t-2} = C_1(t-1) D_\theta Y_{t-1} - \sigma_{t-1} z \tag{9}$$

We can obtain this by eliminating Y_{t-1} from Eq.(8) and Eq.(9) and replacing Y_t with $WX_t + b$:

$$\begin{gathered} Y_{t-2} = W'' X_t + C'' \\ W'' = \frac{1}{\sqrt{a_t - 1}} \{ \frac{W}{\sqrt{a_t}} - [C_1(t) + C_1(t-1)] W W_D + \sqrt{a_t} C_1(t-1) C_1(t) W W_D W_D \} \\ C'' = \frac{1}{\sqrt{a_t - 1}} [W C_2(t) + \sqrt{a_t} W C_2(t-1) - \sqrt{a_t} C1_(t-1) C_2(t) W W_D] Z + b \end{gathered} \tag{10}$$

Note that a single module completes two steps of denoising. To ensure the continuity of denoising, the t value should be sequentially reduced by 2.

Our proposed *DenoiseRep* is based on feature-level denoising and can be migrated to various downstream tasks. It denoises the features on each layer for better removal of noise at each stage, as the noise in the inference stage comes from multiple sources, which could be noise in the input image or noise generated while passing through the network. Denoising each layer avoids noise accumulation and gives better quality output. And according to the noise challenges brought by data in different scenarios, the denoising intensity can be adjusted by controlling t, β_t, and the number of denoising times, which has good generalization ability.

Algorithm 2 Inference

Input: The number of feature layers in the backbone N, features extracted from each layer $\{F_i\}_{i=1}^{N}$, trained denoising module parameters $\{W_{D_i}\}_{i=1}^{N}$ in Algorithm(1), after obtaining the initial feature F^N through patch_embed, it is necessary to remove N-step noise from it, the pre-trained parameters $\{W_i\}_{i=1}^{N}$ and $\{b_i\}_{i=1}^{N}$ for the backbone.

Output: Feature F^0 after denoising.

for each $i \in [N, 1]$ **do**
  $t = i$: Set the denoising amplitude based on the depth of the current layer.
  $W' = [W_i - C_1(t) W_i W_{D_i}]$, $b' = W_i C_2(t) C_3 + b_i$: Parameter fusion according to Eq.(7).
  $F^{t-1} = W' F^t + b'$: Fuse feature extraction and feature denoising.
end for
return F^0

3.4 Unsupervised Learning Manner

Our proposed *DenoiseRep* is label-free because its essence is a generative model that models data by learning its distribution. Thus the training loss contains only the $Loss_p$ of denoising layers:

$$Loss_p = \sum_{i=1}^{N} |\epsilon_i - D_{\theta_i}(X_{t_i}, t_i)| \tag{11}$$

where ϵ denotes the sampled noise, N denotes the number of denoising layers, X_t denotes the noise sample, t denotes the diffusion step, and $D_\theta(X_t, t)$ denotes the noise predicted by the denoising layer.

However, it is worth noting that our method is complementary to label if the label is available. $Loss_l$ is the task-specific supervised loss with label, λ is the trade-off parameter between two losses. The label-argumented learning is defined as:

$$Loss = (1 - \lambda)Loss_l + \lambda Loss_p \tag{12}$$

Results in experiments Section 4.1 shows the improvement from label.

4 Experiments

Table 1: Experimental results on various discriminative tasks.

Task	Model	Backbone	Dataset	Metric	*Baseline*	***+DenoiseRep***
Classification	SwinT [39]	SwinV2-T	ImageNet-1k	acc@1	81.82%	82.13%
Person-ReID	TransReID-SSL [41]	ViT-S	MSMT17	mAP	66.30%	67.33%
Detection	Mask-RCNN [19]	SwinV2-T	COCO	AP	42.80%	44.30%
Segmentation	FCN [40]	ResNet-50	ADE20K	BIoU	28.70%	29.90%

Our proposed *DenoiseRep* is a versatile method that can be incrementally applied to various discriminative tasks. Table 1 demonstrates that *DenoiseRep* yields stable and substantial improvements across image classification, object detection, image segmentation, and person re-identification. Given that person re-identification is a nuanced image retrieval task that poses a greater challenge to feature discriminability, we take it as our benchmark for model analysis. Details of the experimental settings are provided in Appendix A. Additional experimental results on various tasks are presented in Appendices B, C, D, and E.

4.1 Analysis of Label Informations

Table 2: *DenoiseRep* is a label-free method that can also be effectively complemented with labels when they are available. The table below analyzes the effectiveness of using labels. The baseline method, TransReID-SSL, is based on a ViT-small backbone. "Label-free" indicates training without labels, "label-augmented" refers to the use of labels, and "merged dataset" denotes the use of combined datasets without labels.

Method	DukeMTMC(%)	MSMT17(%)	Market1501(%)	CUHK-03(%)
TransReID-SSL	81.20	66.30	91.20	83.50
***+DenoiseRep* (label-free)**	81.72 (↑ 0.52)	66.87 (↑ 0.57)	91.82 (↑ 0.62)	83.72 (↑ 0.22)
***+DenoiseRep* (label-aug)**	82.12 (↑ 0.92)	67.33 (↑ 1.03)	92.05 (↑ 0.85)	84.11 (↑ 0.61)
***+DenoiseRep* (merged ds)**	81.78 (↑ 0.58)	66.99 (↑ 0.69)	91.80 (↑ 0.60)	83.86 (↑ 0.36)

As mentioned in Section 3.3, *DenoiseRep* is an unsupervised denoising module, and its training does not require the assistance of label information. We conducte the following experiments to identify three key issues.

(1) *Is this label-free and unsupervised training denoising plugin effective?* As shown in Table 2 (line2), compared with baseline method (line1), the baseline method performs better after adding our **label-free** plugin, which shows that our method does have denoising capability for features.

(2) *Could introducing label information for supervised training further improve performance?* Introducing label information is actually adding $Loss_l$ as mentioned in Section 3.4 as a supervised signal. As shown in Table 2 (line3), baseline method with label-argumented *DenoiseRep* achieve improvements of 0.32% - 0.70% on the mAP metric, indicating that our denoising plugin has label compatibility, in other words, the plug-in is effective for feature denoising regardless of label-argumented supervised or lable-free unsupervised training.

(3) *Since our plugin can perform unsupervised denoising of features, it is natural to think about whether adding more data for training the plugin could further improve its performance?* We merge four datasets for training and then test on each dataset using mAP to evaluate. Comparing the results of training on sigle dataset (line2) with on merged datasets (line4), we found that adopting other datasets for unsupervised learning can further improve the performance of *DenoiseRep*, which also proves that *DenoiseRep* has good generalization ability.

To demonstrate that our method can perform unsupervised learning and has good generalization, we merged four datasets and rearranged the sequence IDs to ensure the reliability of the experiment. The model is tested on the entire dataset. During the training process, we freeze the baseline parameters and only train the *DenoiseRep* module, without the need for labels, for unsupervised learning. Then test on a single dataset and compare the results of training on a single dataset. As shown in Table 2, it can be observed that adding unlabeled training data from different datasets can improve the model's performance on a single dataset, proving that this module has a certain degree of generalization.

4.2 Comparison with State-of-the-Art ReID Methods

We compare several state-of-the-art ReID methods on four datasets. One of the best performing comparison methods is TransReID-SSL, which is a series of ReID methods based on the ViT backbones. Other methods are based on structures such as CNNs. We add our method to TransReID-SSL series and observe their performance. As shown in Table 3, we have the following findings:

Table 3: Comparison with state-of-the-art ReID methods.

Method	Backbone	MSMT17		Market1501		DukeMTMC		CUHK03-L	
		mAP	R1	mAP	R1	mAP	R1	mAP	R1
MGN [59]	ResNet-50	–	–	86.90	95.70	78.40	88.70	67.40	68.00
OSNet [74]	OSNet	52.90	78.70	84.90	94.80	73.50	88.60	–	–
BAT-net [15]	GoogLeNet	56.80	79.50	87.40	95.10	77.30	87.70	76.10	78.60
ABD-Net [8]	ResNet-50	60.80	82.30	88.30	95.60	78.60	89.00	–	–
RGA-SC [68]	ResNet-50	57.50	80.30	88.40	96.10	–	–	77.40	81.10
ISP [76]	HRNet-W32	–	–	88.60	95.30	80.00	89.60	74.10	76.50
CDNet [29]	CDNet	54.70	78.90	86.00	95.10	76.80	88.60	–	–
Nformer [60]	ResNet-50	59.80	77.30	91.10	94.70	83.50	89.40	78.00	77.20
TransReID [20]	ViT-base-ics	67.70	85.30	89.00	95.10	82.20	90.70	84.10	86.40
TransReID	ViT-base	61.80	81.80	87.10	94.60	79.60	89.00	82.30	84.60
TransReID-SSL [41]	ViT-small	66.30	84.80	91.20	95.80	81.20	87.80	83.50	85.90
TransReID-SSL	ViT-base	75.00	89.50	93.10	96.52	84.10	92.60	87.80	89.20
CLIP-REID [32]	ViT-base	75.80	89.70	90.50	95.40	83.10	90.80	–	–
TransReID + ***DenoiseRep***	ViT-base-ics	68.10	85.72	89.56	95.50	82.35	90.87	84.15	86.39
TransReID + ***DenoiseRep***	ViT-base	62.23	82.02	87.25	94.63	80.12	89.33	82.44	84.61
TransReID-SSL + ***DenoiseRep***	ViT-small	67.33	85.50	92.05	96.68	82.12	88.72	84.11	86.47
TransReID-SSL + ***DenoiseRep***	ViT-base	75.35	89.62	**93.26**	**96.55**	**84.31**	**92.90**	**88.08**	**89.29**
CLIP-REID + ***DenoiseRep***	ViT-base	**76.30**	**90.60**	91.10	95.80	83.70	91.60	–	–

(1) Our method stands out on four datasets on ViT-base backbone with a large number of parameters, achieving almost the best performance on two evaluation metrics.

(2) The methods using our plugin outperforms the original methods with the same backbone on all datasets. In addition, the performance improvement of small-scale backbones with the addition of *DenoiseRep* is more significant than the large-scale backbones approach due to the fact that *DenoiseRep* is essentially a denoising module that removes the noise contained in the features during

the inference stage. For large-scale backbones, the extracted features already have good performance, so the denoising amplitude is limited. It has already fitted the dataset well. For small-scale backbones with poor performance, due to their limited fitting ability, there is a certain amount of noise in the extracted features during the inference stage. Denoising them can obtain better feedback.

(3) In fact, our method can be applied to any other backbone, just add it to each layer. In particular, the performance improvement of adding the denoising plugin to a poorly performing backbone might be more significant. This needs to be further verified in subsequent work. However, it is undeniable that we have verified the denoising ability of the *DenoiseRep* in the currently optimal ReID method.

In this section, a comparative analysis was conducted on four datasets to assess various existing ReID methods. These methods represent current mainstream ReID approaches, employing ResNet101, ViT-S, ViT-B, and ResNet50 as backbone architectures for feature extraction, respectively. Experimental results indicate that our proposed method outperforms other approaches in terms of both mAP and Rank-1 metrics.

4.3 Analysis of Parameter Fusion

The proposed *DenoiseRep* is computation-free. In section 3.3, we proved by theoretical derivation that inserting our denoising layer into each feature layer and fusion it does not introduce additional computation. In this section, we also conduct related validation experiments, the results of which are shown in Table 4.

Table 4: Parameter Fusion Performance Analysis. The *DenoiseRep*$^-$ denoises based on the features of the final layer, while the *DenoiseRep* denoises based on the features of each layer. The baseline method TransReID-SSL is based on ViT-small backbone.

Method	DukeMTMC	MSMT17	Market1501	CUHK-03	Inference Time
TransReID-SSL	81.20%	66.30%	91.20%	83.50%	**0.34s**
+DenoiseRep$^-$	81.56%	66.81%	91.07%	83.59%	0.39s (+15%)
+DenoiseRep	82.12%	67.33%	92.05%	84.11%	**0.34s** (+0%)

Compare to the baseline method TransReID-SSL, adding *DenoiseRep*$^-$ is able to improve the the performance, proving that feature based denoising is effective. However, it also brings extra inference latency (about 15%) because it is adding an extra parameter-independent denoising module at the end of the model.

Adopting *DenoiseRep* achieves a greater increase, it denoise the features on each layer, which can better remove noise at each stage because the noise in the inference stage comes from multiple aspects, which may be the noise in the input image or generated when passing through the network. Denoising each layer avoids noise accumulation and obtains a better quality output. Most importantly, since the operation of fusion can merge the parameters of the denoising module with the original parameters, the adoption of *DenoiseRep* does not take extra inference latency cost, which is a **computation-free** efficient approach.

4.4 Experiments on Classification Tasks

The *DenoiseRep* is based on denoising at the feature level and demonstrates strong generalization capabilities. To validate this generalization ability, we conduct experiments on other vision tasks to test the effectiveness of the *DenoiseRep*. We validate the generalization ability of *DenoiseRep* in image classification tasks on ImageNet-1k [12] datasets and three fine-grained image classification datasets (CUB200 [55], Oxford-Pet [46], and Flowers [45]). The accuracy index is chosen as the evaluation metric to assess model performance.

As shown in Table 5, we compare multiple classic backbones for representation learning on ImageNet-1k, and after adding the *DenoiseRep*, the accuracy of both top-1 and top-5 metrics improve without adding model parameters. Our method shows significant improvement in accuracy metrics compared to baseline on three fine-grained classification datasets. Prove that the *DenoiseRep* can improve the model's ability in image classification for different classification tasks. Additionally, our method proves to enhance the model's representation learning ability and extract more effective features

Table 5: The effectiveness of our method in image classification tasks was validated on three fine-grained classification datasets (CUB200, Oxford-Pet, Flowers) and ImageNet-1k.

Method	Datasets	Param	acc@1		acc@5	
			Baseline	***+DenoiseRep***	*Baseline*	***+DenoiseRep***
SwinV2-T [39]	ImageNet-1k	28M	81.82%	82.13%	95.88%	96.06%
Vmanba-T [38]	ImageNet-1k	30M	82.38%	82.51%	95.80%	95.89%
ResNet50 [18]	ImageNet-1k	26M	76.13%	76.28%	92.86%	92.95%
ViT-B [14]	CUB200	87M	91.78%	91.99%	–	–
ViT-B	Oxford-Pet	87M	94.37%	94.58%	–	–
ViT-B	Flowers	87M	99.12%	99.30%	–	–

through denoising without incurring additional time costs. More experimental analysis can be found in Table 7 in Section C of the appendix.

5 Conclusion

In this work, we demonstrate that the diffusion model paradigm is effective for feature level denoising in discriminative model, and propose a computation-free and label-free method: *DenoiseRep*. It utilizes the denoising ability of diffusion models to denoise the features in the feature extraction layer, and fuses the parameters of the denoising layer and the feature extraction layer, further improving retrieval accuracy without incurring additional computational costs. We validate the effectiveness of the *DenoiseRep* method on multiple common image discrimination task datasets.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (62202041), National Key Research and Development Program of China under Grant (2023YFC3310700) and Fundamental Research Funds for the Central Universities (2023JBMC057).

References

[1] Abien Fred Agarap. An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification. *arXiv preprint arXiv:1712.03541*, 2017.

[2] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[3] Apurva Bedagkar-Gala and Shishir K Shah. A survey of approaches and trends in person re-identification. *Image and vision computing*, 32(4):270–286, 2014.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

[5] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2024. doi: 10.1109/TKDE.2024.3361474.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[7] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023.

[8] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8351–8361, 2019.

[9] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[10] Yoonki Cho, Woo Jae Kim, Seunghoon Hong, and Sung-Eui Yoon. Part-based pseudo label refinement for unsupervised person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7308–7318, 2022.

[11] Ruihang Chu, Yifan Sun, Yadong Li, Zheng Liu, Chi Zhang, and Yichen Wei. Vehicle re-identification with viewpoint-aware metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8282–8291, 2019.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[13] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Pengfei Fang, Jieming Zhou, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Bilinear attention networks for person retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8030–8039, 2019.

[16] Dengpan Fu, Dongdong Chen, Jianmin Bao, Hao Yang, Lu Yuan, Lei Zhang, Houqiang Li, and Dong Chen. Unsupervised pre-training for person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14750–14759, 2021.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[20] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15013–15022, October 2021.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition*, pages 2288–2295, 2012.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[27] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Hanjun Li, Gaojie Wu, and Wei-Shi Zheng. Combined depth space based architecture search for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6729–6738, 2021.

[30] Jianxin Li, Shuai Zhang, Hui Xiong, and Haoyi Zhou. Autost: Towards the universal modeling of spatio-temporal sequences. *Advances in Neural Information Processing Systems*, 35:20498–20510, 2022.

[31] Jing Li, Yu Liu, and Lei Zou. Dyngcn: A dynamic graph convolutional network based on spatial-temporal modeling. In *Web Information Systems Engineering–WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part I 21*, pages 83–95. Springer, 2020.

[32] Siyuan Li, Li Sun, and Qingli Li. Clip-reid: exploiting vision-language model for image re-identification without concrete text labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1405–1413, 2023.

[33] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 152–159, 2014.

[34] Shengcai Liao and Stan Z Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *International Conference on Computer Vision*, pages 3685–3693, 2015.

[35] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Computer Vision and Pattern Recognition*, pages 2197–2206, 2015.

[36] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 27–35, 2015.

[37] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2167–2175, 2016.

[38] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

[39] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[41] Hao Luo, Pichao Wang, Yi Xu, Feng Ding, Yanxin Zhou, Fan Wang, Hao Li, and Rong Jin. Self-supervised pre-training for transformer-based person re-identification. *arXiv preprint arXiv:2111.12084*, 2021.

[42] Bingpeng Ma, Yu Su, and Frederic Jurie. Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 32(6-7):379–390, 2014.

[43] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[44] Hao Ni, Yuke Li, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Part-aware transformer for generalizable person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11280–11289, 2023.

[45] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[46] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[47] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[48] Joseph Redmon. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[49] Shaoqing Ren. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[52] Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion. *arXiv preprint arXiv:2308.12469*, 2023.

[53] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[56] Benzhi Wang, Yang Yang, Jinlin Wu, Guo-jun Qi, and Zhen Lei. Self-similarity driven scale-invariant learning for weakly supervised person search. *arXiv preprint arXiv:2302.12986*, 2023.

[57] Guan'an Wang, Yang Yang, Jian Cheng, Jinqiao Wang, and Zengguang Hou. Color-sensitive person re-identification. In *International Joint Conference on Artificial Intelligence*, pages 933–939, 2019.

[58] Guan'an Wang, Shuo Yang, Huanyu Liu, Zhicheng Wang, Yang Yang, Shuliang Wang, Gang Yu, Erjin Zhou, and Jian Sun. High-order information matters: Learning relation and topology for occluded person re-identification. 2020.

[59] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 274–282, 2018.

[60] Haochen Wang, Jiayi Shen, Yongtuo Liu, Yan Gao, and Efstratios Gavves. Nformer: Robust person re-identification with neighbor transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7307, 2022.

[61] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 79–88, 2018.

[62] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020.

[63] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[64] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *European Conference on Computer Vision*, pages 511–528. Springer, 2022.

[65] Yang Yang, Jimei Yang, Junjie Yan, Shengcai Liao, Dong Yi, and Stan Z Li. Salient color names for person re-identification. In *European conference on computer vision*, pages 536–551, 2014.

[66] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):2872–2893, 2021.

[67] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020.

[68] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Relation-aware global attention for person re-identification. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 3186–3195, 2020.

[69] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*, 2017.

[70] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *International Conference on Computer Vision*, pages 1116–1124, 2015.

[71] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):653–668, 2013.

[72] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. *arXiv preprint arXiv:1701.07717*, 2017.

[73] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[74] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3702–3712, 2019.

[75] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[76] Kuan Zhu, Haiyun Guo, Zhiwei Liu, Ming Tang, and Jinqiao Wang. Identity-guided human semantic parsing for person re-identification. In *European Conference on Computer Vision*, pages 346–363. Springer, 2020.

A Experimental settings

Datasets and Evaluation metrics. We conduct training and evaluation on four datasets: DukeMTMC-reID [72], Market-1501 [70], MSMT17 [61], and CUHK-03 [33]. These datasets encompass a wide range of scenarios for person re-identification. For accuracy, we use standard metrics including Rank-1 curves (The probability that the image with the highest confidence in the search results is the correct result.) and mean average precision (MAP). All the results are from a single query setting.

Implementation Details. We implement our method using Python on a server equipped with a 2.10GHz Intel Core Xeon (R) Gold 5218R processor and two NVIDIA RTX 3090 GPUs. The epochs we trained are set to 120, the learning rate is set to 0.0004, the batch size during training is 64, the inference stage is 256, and the diffusion step size t is set to 1000.

Training and evaluation. To better constrain the performance of the denoised features of the *DenoiseRep* for downstream tasks, we employ alternating fine-tune methods. The parameters of the *DenoiseRep* and baseline are trained alternately, and when training a part of the parameters, the rest of the parameters are frozen and fine-tuned for 10 epochs at a time, with a total number of epochs of 120. When evaluating, we average the results of the experiments under the same settings for 5 times, thus ensuring the reliability of the data.

B Experiment on Vehicle Identification

In the image retrieval task, we also conduct experiments to verify the effectiveness of our method on the vehicle recognition task. Vehicle recognition in practical scenarios often results in images containing a large amount of noise due to environmental factors such as lighting or occlusion, which increases the difficulty of detection. Our method is based on denoising to obtain features with better representation ability. Therefore, we want to experimentally verify whether the *DenoiseRep* plays a role in vehicle recognition tasks with higher noise levels. We select vehicleID [37] as the dataset, vehicle-ReID [11] as the baseline, and ResNet-50 as the feature extractor for the experiment.

Table 6: The performance of the *DenoiseRep* on vehicle recognition tasks.

Method	Backbone	Datasets	mAP	Rank-1
vehicle-ReID	ResNet-50	vehicleID	76.4%	69.1%
vehicle-ReID + ***DenoiseRep***	ResNet-50	vehicleID	77.3%	70.2%

From the results in Table 6, it can be seen that *DenoiseRep* demonstrates excellent performance in vehicle detection tasks. Compared to the baseline, adding the *DenoiseRep* significantly improves both mAP and Rank-1 metrics without incurring additional detection time costs. It verifies the denoising ability of the *DenoiseRep* in noisy environments.

C Experiment on Large Scale Image Classification Tasks

In this section, we aim to test the generalization ability of *DenoiseRep* in other tasks. We conduct experiments on two image classification datasets, ImageNet-1k and Cifar-10. These two datasets are both classic image classification datasets, rich in common images in daily life, and belong to large-scale image databases. ImageNet-1k is a subset of the ImageNet dataset, containing images from 1000 categories. Each category typically has hundreds to thousands of images, totaling over one million images. The Cfiar-10 contains 60000 32x32 pixel color images, divided into 10 categories. Each category contains 6000 images. To evaluate the effectiveness of our method, we use standard metrics, including Top-1 accuracy and Top-5 accuracy, which are commonly used to evaluate model performance in image classification tasks and are widely used on various image datasets, and we conduct detailed comparative experiments on multiple backbones and models with different parameter versions to verify the reliability of our method.

As shown in Table 7, we compare multiple classic backbones for representation learning on two datasets, and after adding the *DenoiseRep*, the accuracy metrics improves without adding model parameters. Our method demonstrates the capability to enhance the model's representation learning

Table 7: The effectiveness of our method in image classification tasks was validated on Cifar-10 and ImageNet-1k.

Method	Datasets	Param	acc@1		acc@5	
			Baseline	***+DenoiseRep***	*Baseline*	***+DenoiseRep***
SwinV2-T [39]	ImageNet-1k	28M	81.82%	82.13%	95.88%	96.06%
SwinV2-S [39]	ImageNet-1k	50M	83.73%	83.97%	96.62%	96.86%
SwinV2-B [39]	ImageNet-1k	88M	84.20%	84.31%	96.93%	97.06%
Vmanba-T [38]	ImageNet-1k	30M	82.38%	82.51%	95.80%	95.89%
Vmanba-S [38]	ImageNet-1k	50M	83.12%	83.27%	96.04%	96.22%
Vmanba-B [38]	ImageNet-1k	89M	83.83%	83.91%	96.55%	96.70%
ViT-S [14]	ImageNet-1k	22M	83.87%	84.02%	96.73%	96.86%
ViT-B [14]	ImageNet-1k	86M	84.53%	84.64%	97.15%	97.23%
ResNet50 [18]	ImageNet-1k	26M	76.13%	76.28%	92.86%	92.95%
ViT-S [14]	Cifar-10	22M	96.13%	96.20%	–	–
ViT-B [14]	Cifar-10	87M	98.02%	98.31%	–	–

ability and extract more effective features through denoising, all while maintaining the same time costs. Moreover, *DenoiseRep* generalizes effectively to image classification tasks.

D Experiment on Image Detection Task

In this section, we aim to test the generalization ability of *DenoiseRep* in image detection tasks. We conduct experiments on the COCO [22] dataset. The COCO (Common Objects in Context) dataset is a widely used dataset for large-scale image recognition, object detection, and image segmentation, particularly in computer vision tasks. It contains 80 types of objects, such as people, animals, daily necessities, etc., covering various common items in daily life. To verify that our method is model independent, we conduct experiments using different models including Mask-RCNN [19], Faster-RCNN [49], ATSS [67], YOLO [48], DETR [6] and CenterNet [75], as well as diverse backbones. To evaluate the effectiveness of our method, we used standard metrics including AP, AP_{50}, and AP_{75}, which are commonly used metrics in object detection tasks to evaluate model performance, particularly for evaluating the effectiveness of bounding box detection. It is also an important part of the COCO dataset evaluation criteria, which can measure the detection ability of the model in multiple categories and scales.

Table 8: The effectiveness of our method in image detection tasks was validated on COCO.

Methods	Backbones	AP		AP_{50}		AP_{75}	
		Baseline	***+DenoiseRep***	*Baseline*	***+DenoiseRep***	*Baseline*	***+DenoiseRep***
Mask-RCNN	SwinV2-T	42.8%	44.3%	65.1%	67.1%	47.0%	48.6%
	SwinV2-S	48.2%	49.0%	69.9%	70.9%	52.8%	53.8%
	ResNet-50	42.6%	43.2%	63.7%	65.0%	46.4%	46.8%
Faster-RCNN	ResNet-50	37.4%	38.3%	58.1%	58.8%	40.4%	41.0%
ATSS	ResNet-50	39.4%	39.9%	57.6%	58.2%	42.8%	43.2%
YOLO	DarkNet-53	27.9%	28.4%	49.2%	50.3%	28.3%	27.8%
DETR	ResNet-50	39.9%	40.8%	60.4%	59.9%	41.7%	42.9%
CenterNet	ResNet-50	40.2%	40.6%	58.3%	59.1%	43.9%	44.0%

As shown in Table 8, we compare multiple classic backbone networks across different methods. After adding *DenoiseRep*, the accuracy index shows improvement without the need for additional model parameters. This indicates that our method enhances the representation learning capability of the model and extracts more effective features through denoising, all while maintaining the same time costs. In addition, *DenoiseRep* well to generalized to image detection tasks.

E Experiment on Image Segmentation Task

In this section, our objective is to assess the generalization capability of *DenoiseRep* in image segmentation tasks. The image segmentation task aims to divide an image into multiple regions in order to identify and understand objects or areas within them. The main challenges it faces include: complex and varied backgrounds that can easily interfere with segmentation results, objects obstruct each other, making segmentation difficult, objects have diverse shapes and may undergo deformations, etc. We conduct experiments on the ADE20K [73] dataset using the current mainstream image segmentation models. ADE20K is a widely used scene segmentation dataset, mainly used for image segmentation tasks. It contains approximately 20000 images and over 150 different object and region categories. We choose mIoU and B-IOU as evaluation metrics to comprehensively evaluate the performance of image segmentation models. MIoU is the average of IoUs for all categories, which can effectively reflect the segmentation ability of the model on different categories. It provides a quantitative model for the accuracy of handling complex scenes by calculating the degree of overlap between the predicted area and the real area. A higher mIoU value means that the model can better identify and segment the target object. B-IoU focuses on evaluating the accuracy of segmentation boundaries and is particularly suitable for object edge segmentation tasks. It provides sensitivity to boundary details by measuring the degree of overlap between predicted boundaries and real boundaries.

Table 9: The effectiveness of our method in image segmentation tasks was validated on ADE20K.

Methods	Backbones	aAcc		B-IoU		mIoU	
		Baseline	***+DenoiseRep***	*Baseline*	***+DenoiseRep***	*Baseline*	***+DenoiseRep***
FCN [40]	ResNet-50	0.774	0.779	0.287	0.299	0.359	0.365
FCN	ResNet-101	0.793	0.796	0.306	0.316	0.396	0.404
SegFormer [63]	mit_b0	0.782	0.788	0.292	0.297	0.374	0.381
SegFormer	mit_b1	0.812	0.816	0.341	0.348	0.422	0.425

As shown in Table 9, we compare two classical backbone networks with different methods. After adding the *DenoiseRep*, both IoU metrics improve without adding model parameters. Practice proves that our method can improve the representation learning ability of the model and obtain more effective features through denoising without increasing additional time costs. In addition, *DenoiseRep* well to generalized to image segmentation tasks.

F Fairness Experiment

To ensure the fairness of the experiment, we compared the performance of the baseline method with our proposed method under the same conditions. Specifically, during the training process, we strictly controlled the experimental variables so that both the baseline method and our method ran under the same number of training epochs and hyperparameter settings.

Table 10: Comparison of performance between baseline method and our proposed method in the same additional training epochs. The baseline method TransReID-SSL is based on ViT-small backbone.

Epoch	**120**	**160**	**200**	**240**
Baseline	81.18%	81.19%	81.18%	81.16%
+*DenoiseRep*	81.18%	81.64%	82.00%	82.12%

The experimental results in Table 10 indicate that the baseline method did not show significant performance improvement under the same number of training epochs. This observation indicates that the performance improvement obtained is attributed to our proposed method, rather than an increase in training time or number of epochs, which validates the effectiveness of our method.

G Limitations

Our proposed method *DenoiseRep* improves the accuracy of current mainstream backbones while ensuring label-free and no additional computational costs, and it has been experimentally verified to

be generalizable in multiple image tasks. However, from the experimental results, it can be found that our method has limited improvement in model accuracy when generalized to general tasks, and in order to fuse the parameters of the denoising layer and the feature extraction layer, only one or two steps of denoise for each denoising layer, and the number of denoising layers is not more than that of the feature extraction layer, which limits the denoising intensity. We will continue to explore how to further improve the accuracy of the model without adding additional inference time costs or only adding a small number of additional parameters.

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: In the abstract and introduction, we have elaborated on the starting point and research direction of the paper, and summarized the contributions of the paper in detail.

   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: We have explained the limitations of our method and future improvement directions in the appendix.

   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: [Yes]

Justification: In the section "Denoising Representation for Person Re Identification", we validated our proposed theory through detailed formula derivation.

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: In the "Implementation Details" section, we introduced the training techniques and parameter settings, and the experimental code will be released on GitHub, so that readers can reproduce our method well.

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: [NA]

   Justification: We do not include the code in this submission, but in the future, we will organize the experimental code and documentation, and released on GitHub.

   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: We have provided a detailed explanation of the model parameters and training methods in the "Experimental settings" section of the appendix.

   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: [No]

   Justification: We took the average of the results of five independent repeated experiments as the main experiment, but did not provide a detailed analysis and discussion of their variance.

   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: [Yes]

   Justification: In the "Experimental settings" section of the appendix, we provide sufficient information on the computer resources.

   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 in this paper complies with NeuroIPS ethical standards in all aspects.

   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: We have added appropriate citation explanations to the existing papers, models, and datasets mentioned in this paper.

    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: The paper does not release new assets.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - 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: The paper does not involve crowdsourcing nor research with human subjects.

    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.