Estimating Transition Matrix with Diffusion Models for Instance-Dependent Label Noise

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning with noisy labels is a common problem in weakly supervised learning, where the transition matrix approach is a prevalent method for dealing with label noise. It estimates the transition probabilities from a clean label distribution to a noisy label distribution and has garnered continuous attention. However, existing transition matrix methods predominantly focus on class-dependent noise, making it challenging to incorporate feature information for learning instance-dependent label noise. This paper proposes the idea of using diffusion models for estimating transition matrix in the context of instance-dependent label noise. Specifically, we first estimate grouped transition matrices through clustering. Then, we introduce a process of adding noise and denoising with the transition matrix, incorporating features extracted by unsupervised pre-trained models. The proposed method enables the estimation of instance-dependent transition matrix and extends the application of transition matrix method to a broader range of noisy label data. Experimental results demonstrate the significant effectiveness of our approach on both synthetic and real-world datasets with instance-dependent noise. The code will be open sourced upon acceptance of the paper.

1 Introduction

For classification problems with given labels, deep neural networks have demonstrated significant improvements compared to traditional methods in recent years [25]. The efficacy of deep neural networks heavily relies on the accuracy of the labels. Directly incorporating polluted erroneous labels into network learning can result in the network fitting the noise, potentially severely impacting the predictive performance of the network [8]. However, in reality, obtaining accurate annotated data can be prohibitively expensive, and a substantial amount of data comes from the Internet or is annotated by non-expert annotators, inevitably containing noisy labels. Therefore, researching and promoting methods to mitigate the damage to models and make them more robust in the face of label noise data is a highly worthwhile problem to investigate, known as the problem of learning with noisy labels [23, 10, 34, 1].

Different approaches have been proposed to address the problem of label noise. One category [31, 22] involves the design of specialized loss functions or network structures to enhance the model's robustness against noisy labels. Another major category focuses on sample selection [2, 10, 14], where samples are partitioned into a set of clean samples and a set of contaminated noisy samples based on the magnitude of the loss or the similarity of extracted features. The labels of the noisy samples are then modified or their weights are reduced, followed by learning using semi-supervised methods. Sample selection methods are currently mainstream and have achieved promising results. However, the selection process relies heavily on intuition and lacks theoretical support. Additionally, the sample selection procedure is often complex and computationally intensive. In contrast, another

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

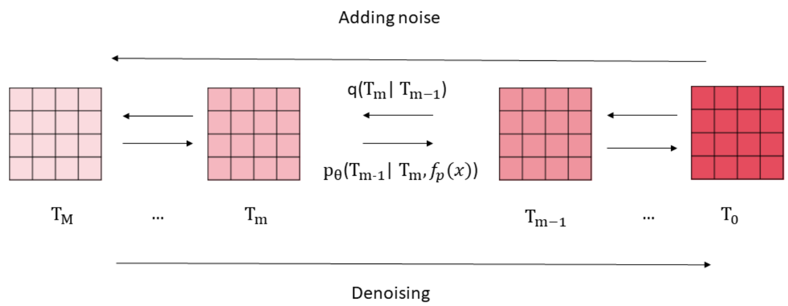

Figure 1: Diffusion Model for Transition Matrix.

significant category of methods is the transition matrix method [34, 17, 12, 42], which estimates the transition probabilities from the clean label distribution to the noisy label distribution. This class of methods reveals the generation process of noisy labels and exhibits statistical consistency, often accompanied by theoretical analyses as methodological support. As a result, they have garnered continuous attention and occupy an important position in various algorithms for learning with noisy labels.

In transition matrix methods, accurate estimation of the transition matrix is crucial. If an accurate estimation of the transition matrix can be obtained, along with the observed data for estimating the posterior distribution of the noisy labels, it is possible to infer the distribution of clean labels for neural network learning. Previous transition matrix methods [34, 17, 39] have mainly focused on class-dependent label noise, where a single transition matrix is estimated for all samples, which is typically straightforward. However, for instance-dependent label noise and complex real-world data, the label transition probabilities for each sample are not entirely identical. The transition matrix often depends on the specific features of individual samples, requiring the estimation of a separate transition matrix for each sample. However, in most cases, a single observed label corresponds to each sample in the dataset, making it an identifiability problem to estimate a separate transition matrix for each sample [20]. Although some methods [33, 41, 15] have utilized separate small networks to generate the transition matrix or divided the data into groups to transform it into a grouped class-dependent scenario, there still exist significant estimation errors and a lack of incorporating features effectively into the estimation of the transition matrix.

To better incorporate the feature information of images into the estimation of the transition matrix, this work employs conditional diffusion models. The diffusion model originates from generative models and has been widely applied in various computer vision tasks in recent years [36, 7], showing remarkable results. The proposed method revolves around the core idea of replacing image samples in the original diffusion process with a transition matrix. The matrix undergoes a process of adding noise and denoising, where the denoising step incorporates the sample features extracted by a pre-trained model as conditions. This generates a feature-dependent transition matrix. The constructed diffusion module is illustrated in Figure 1. Additionally, considering the assumption that instance-dependent label noise is usually correlated with features [6], clustering methods are utilized at the feature level to group samples. Preliminary estimations of the transition matrices are obtained for each group, which are then incorporated into the diffusion module for learning. The overall framework of the method is depicted in Figure 2.

The subsequent sections are organized as follows. Section 2 presents an in-depth review of the relevant works. In Section 3, we introduce our proposed model framework. Section 4 outlines the experimental analysis conducted on diverse synthetic and real-world noisy datasets, along with comparisons against other existing methods. Finally, we provide concluding in Section 5. The primary contributions of this paper can be summarized as follows:

- We propose a method that utilizes diffusion models to add noise and denoise on the transition matrix, incorporating image features extracted through pre-trained encoder.
- By combining the transition matrix-based diffusion model with feature-based clustering, we establish a framework capable of addressing instance-dependent label noise problems.

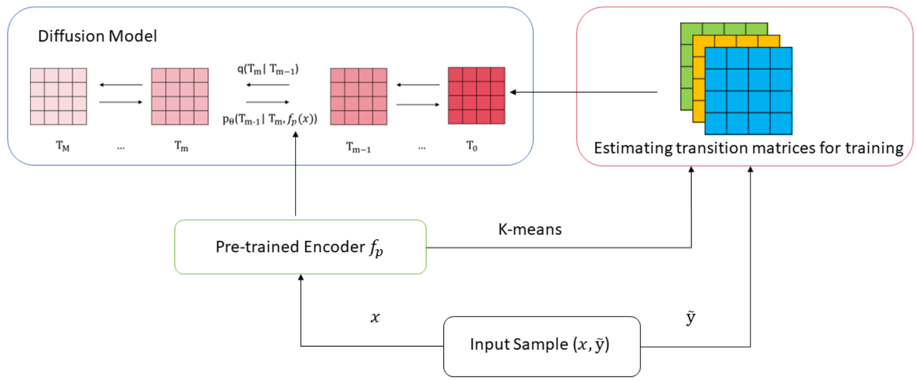

Figure 2: The overall framework of DTM.

- Our method demonstrates significant improvements over other transition matrix methods on both synthetic and real-world noisy datasets, and it achieves comparable performance to state-of-the-art methods.

2 Related Works

2.1 Transition Matrix Methods

Most previous methods for estimating transition matrix in the presence of label noise have primarily focused on class-dependent noise scenarios, simplifying the estimation process. Methods such as [24, 34] assume the existence of anchor points to identify the transition matrix. [17] and [39] introduce different regularization techniques to relax the anchor point assumption. Additionally, [26, 38] apply techniques such as meta-learning to estimate the transition matrix, but these approaches may require more clean data and computational resources. While these methods are effective for handling class-dependent label noise, they are not suitable for instance-dependent noise or real-world noisy data.

However, estimating an individual transition matrix for each sample without additional assumptions or multiple noisy labels is infeasible [20]. To approximate the estimation of the instance-dependent transition matrix, [9] utilize an adaptation layer that estimates the transition matrix based on the output of each sample. [37] employs a separate network to estimate the transition matrix based on Bayesian labels. Some methods, such as [33, 30, 41], employ clustering to learn part-dependent or group-dependent matrices, which can be viewed as a compromise between instance-dependent and class-dependent methods. Other approaches, including [6, 12], utilize the similarity in the feature space to aid in learning the transition matrix. Although these instance-dependent transition matrix methods achieve identifiability through specialized treatments, they have not effectively utilized feature information in the learning process, resulting in errors in estimating feature-dependent transition matrices.

2.2 Diffusion Models

Diffusion models, as generative models, have played a significant role in computer vision [36, 7]. Prominent examples include DDPM [11], DDIM [27], score matching methods [28], and methods based on stochastic differential equations [29]. Diffusion models and their variants have been applied to various computer vision tasks such as image generation, image-to-image translation, text-to-image generation, among others. However, their application to the problem of label noise is relatively novel. To the best of our knowledge, only one existing work [3] has utilized diffusion models for addressing this problem. However, this work treats labels as the output of the diffusion model, which limits their expressive power due to the low dimension of the labels. Moreover, it overly relies on directly incorporating image features as conditions in the label generation process, which depends heavily on pre-trained models and may not be as reasonable as incorporating them into the transition matrix that reveals the process of noise generation. Experimental results also support this perspective.

3 Method

In this section, we present the definitions of symbols and introduce our method of using **D**iffusion models to construct the **T**ransition **M**atrix (DTM).

3.1 Preliminaries

Let $\mathcal{X} \subset \mathbb{R}^d$ be the input image space, $\mathcal{Y} = \{1, 2, \cdots, C\}$ be the label space, where C is the number of classes. Random variables $(X, Y), (X, \tilde{Y}) \in \mathcal{X} \times \mathcal{Y}$ denote the underlying data distributions with true and noisy labels respectively. In general, we can not observe the latent true data samples $\mathbb{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, but can only obtain the corrupted data $\tilde{\mathbb{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{N}$, where $\tilde{y} \in \mathcal{Y}$ is the noisy label corrupted from the true label y, while denote corresponding one-hot label as $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$.

Transition matrix methods use a matrix $\boldsymbol{T}(\boldsymbol{x}) \in [0, 1]^{C \times C}$ to represent the probability from clean label to noisy label, where the ij-th entry of the transition matrix is the probability that the instance $\boldsymbol{x}$ with the clean label i corrupted to a noisy label j. The matrix satisfies the requirement that the sum of each row $\sum_{j=1}^{C} \boldsymbol{T}_{ij}(\boldsymbol{x})$ is 1, and usually has the requirement for $\boldsymbol{T}_{ii}(\boldsymbol{x}) > \boldsymbol{T}_{ij}(\boldsymbol{x}), \forall j \neq i$. Let $P(\boldsymbol{Y}|X = \boldsymbol{x}) = [P(Y = 1|X = \boldsymbol{x}), \cdots, P(Y = C|X = \boldsymbol{x})]^\top$ be the clean class-posterior probability and $P(\tilde{\boldsymbol{Y}}|X = \boldsymbol{x}) = [P(\tilde{Y} = 1|X = \boldsymbol{x}), \cdots, P(\tilde{Y} = C|X = \boldsymbol{x})]^\top$ be the noisy class-posterior probability, the formula can be write as:

$$P(\tilde{\boldsymbol{Y}}|X = \boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x})^\top P(\boldsymbol{Y}|X = \boldsymbol{x}). \tag{1}$$

By estimating the transition matrix and the noisy class-posterior probability, the clean class-posterior probability can be inferred by

$$P(\boldsymbol{Y}|X = \boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x})^{-\top} P(\tilde{\boldsymbol{Y}}|X = \boldsymbol{x}), \tag{2}$$

where the symbol $-\top$ denotes the transpose of the inverse matrix.

The majority of existing methods [24, 10, 17] focus on studying the class-dependent and instance-independent transition matrix, i.e., $\boldsymbol{T}(\boldsymbol{x}) \equiv \boldsymbol{T}$ for $\forall \boldsymbol{x}$. However, these methods are not applicable to instance-dependent noise scenarios where the transition matrix $\boldsymbol{T}(\boldsymbol{x})$ varies with respect to the input X. The main focus of our work is to utilize the feature information from input images to construct a instance-dependent transition matrix $\boldsymbol{T}(\boldsymbol{x})$.

3.2 Diffusion Model for Transition Matrix

We adopt the classic DDPM model [11] from diffusion models as a reference to perform noise addition and denoising on the transition matrix. The diagram is illustrated in Figure 1.

For the forward diffusion process beginning with transition matrix $\boldsymbol{T}_0 \sim q(\boldsymbol{T})$, the process of gradually adding noise is obtained according to the following Markov process:

$$q\left(\boldsymbol{T}_m \mid \boldsymbol{T}_{m-1}\right) = \mathcal{N}\left(\boldsymbol{T}_m; \sqrt{1-\beta_m}\boldsymbol{T}_{m-1}, \beta_m \mathbf{I}\right), \tag{3}$$

for $m = 1, 2, \cdots, M$, where we use M to replace T, which is usually used in other diffusion models, in above equation for distinguishing from the symbol of transition matrix $\boldsymbol{T}$.

We aim to make the distribution of $q(\boldsymbol{T}_M)$ approach a standard normal distribution $\mathcal{N}(0, \mathbf{I})$ and through $\boldsymbol{T}_M$ to conduct the reverse denoising process by fitting a neural network $\boldsymbol{\mu}_\theta$ to fit the disttibution:

$$p_\theta\left(\boldsymbol{T}_{m-1} \mid \boldsymbol{T}_m\right) = \mathcal{N}\left(\boldsymbol{T}_{m-1}; \boldsymbol{\mu}_\theta\left(\boldsymbol{T}_m, \boldsymbol{x}, f_p, m\right), \tilde{\beta}_m \mathbf{I}\right), \tag{4}$$

where define $\tilde{\beta}_m = \frac{1-\bar{\alpha}_{m-1}}{1-\bar{\alpha}_m}\beta_m, \alpha_m = 1 - \beta_m, \bar{\alpha}_m = \prod_{i=1}^{m} \alpha_i$. The f_p in equation (4) denotes the pre-trained encoder for feature extraction.

The diffusion model can be learned by optimizing the evidence lower bound:

$$\mathcal{L}_{\mathrm{ELBO}}=\mathbb{E}_q\left[\mathcal{L}_M+\sum_{m>1}^{M}\mathcal{L}_{m-1}+\mathcal{L}_0\right], \tag{5}$$

where

$$\begin{aligned}
&\mathcal{L}_0=-\log p_\theta\left(\boldsymbol{T}_0\mid \boldsymbol{T}_1\right),\\
&\mathcal{L}_{m-1}=D_{\mathrm{KL}}\left(q\left(\boldsymbol{T}_{m-1}\mid \boldsymbol{T}_m,\boldsymbol{T}_0\right)\|p_\theta\left(\boldsymbol{T}_{m-1}\mid \boldsymbol{T}_m\right)\right),\\
&\mathcal{L}_M=D_{\mathrm{KL}}\left(q\left(\boldsymbol{T}_M\mid \boldsymbol{T}_0\right)\|p_\theta\left(\boldsymbol{T}_M\right)\right).
\end{aligned} \tag{6}$$

Similar to the derivation and simplification process of DDPM, when a pre-trained encoder f_p is provided along with the training data incorporating the initial transition matrix $\boldsymbol{T}$, the learning algorithm for the diffusion model is presented in Algorithm 1.

Algorithm 1 Diffusion Model for Transition Matrix

Input: Training data $\{\boldsymbol{x}_i,\boldsymbol{T}_i\}_{i=1}^N$, pre-trained encoder f_p.
while not converged **do**
  Sample $(\boldsymbol{x}_0,\boldsymbol{T}_0)$ from data
  Sample $m\sim\{1,\cdots,M\}$
  Sample noise $\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})$
  Take gradient descent step on the loss:

$$\nabla_\theta\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_m}\boldsymbol{T}_0+\sqrt{1-\bar{\alpha}_m}\boldsymbol{\epsilon},\boldsymbol{x}_0,f_p,m\right)\right\|^2$$

end while

Next, for each image $\boldsymbol{x}$, we can sample the corresponding transition matrix $\boldsymbol{T}(\boldsymbol{x})$ as shown in Algorithm 2.

Algorithm 2 Sample for Transition Matrix

Sample $\boldsymbol{T}_M\sim\mathcal{N}(0,\mathbf{I})$
for $m=M,\cdots,1$ **do**
  $\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ if $t>1$, else $\boldsymbol{z}=\mathbf{0}$
  $\boldsymbol{T}_{m-1}=\frac{1}{\sqrt{\alpha_m}}\left(\boldsymbol{T}_m-\frac{1-\alpha_m}{\sqrt{1-\bar{\alpha}_m}}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{T}_m,\boldsymbol{x},f_p,m\right)\right)+\sigma_m\boldsymbol{z}$
end for
Output: $\boldsymbol{T}_0$

3.3 Feature-Dependent Framework

From Algorithm 1, it can be observed that there are two components of the diffusion process that need to be provided in advance: the pre-trained encoder f_p and the initial input $\boldsymbol{T}(\boldsymbol{x})$.

The pre-trained encoder f_p can be obtained through self-supervised learning or directly using the large model like CLIP. In our experiments, we employ the commonly used SimCLR [4] method in contrastive learning as the feature extraction model.

On the other hand, the part involving the transition matrix $\boldsymbol{T}(\boldsymbol{x})$ used for learning the diffusion model is also related to the pre-trained encoder f_p. Based on the assumption that the noise transition probability depends on image features, we adopt a group-dependent transition matrix as the initial input. We perform clustering algorithms at the feature extraction level $f_p(\boldsymbol{x})$, using the K-means method in our experiments, to group the image data. Then, based on the method VolMinNet [17], we train class-dependent transition matrices for each group and obtain the initial transition matrix $\boldsymbol{T}(\boldsymbol{x})$ for each image $\boldsymbol{x}$, which is then used as input in Algorithm 1. It is worth to note that the initial $\boldsymbol{T}(\boldsymbol{x})$ used as input for the diffusion process does not require different for each $\boldsymbol{x}$. However, the denoising process of the diffusion model will further incorporate the feature information into the learning of the transition matrix.

After obtaining the instance-dependent estimated transition matrix $\boldsymbol{T}(\boldsymbol{x})$, the neural network can be learned to fit the clean label distribution by the loss function:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \ell \left(\boldsymbol{T}(\boldsymbol{x}_i)^\top f_{\boldsymbol{\phi}}(\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i \right), \tag{7}$$

where $f_{\boldsymbol{\phi}}(\cdot) : \mathcal{X} \to \Delta^{C-1}$ ($\Delta^{C-1} \subset [0,1]^C$ is the C-dimensional simplex) is a differentiable function represented by a neural network with parameters $\boldsymbol{\phi}$ and ℓ is a loss function usually using cross-entropy (CE) loss.

The schematic diagram of the proposed framework is shown in Figure 2, and the pseudocode is presented in Algorithm 3.

Algorithm 3 A framework of DTM

Input: Training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, pre-trained encoder f_p, diffusion model $\boldsymbol{\epsilon}_\theta$, classification neural network $f_{\boldsymbol{\phi}}$.

1: Utilize input data to train f_p or directly utilizing f_p to extract features.
2: Perform K-means on feature space and estimate the transition matrix for each group to get data $\{\boldsymbol{x}_i, \boldsymbol{T}_i\}_{i=1}^{N}$.
3: Train the diffusion model $\boldsymbol{\epsilon}_\theta$ with Algorithm 1.
4: Sample instance-dependent train matrix $\boldsymbol{T}(\boldsymbol{x})$ for any input image $\boldsymbol{x}_i$ with Algorithm 2.
5: Update the parameters of the classification network by incorporating the transition matrix $\boldsymbol{T}(\boldsymbol{x}_i)$ into equation (7).

Output: Network parameters $\boldsymbol{\phi}$.

3.4 Matrix Transformation

Considering that the transition matrix typically require the sum of each row $\sum_{j=1}^{C} \boldsymbol{T}_{ij}(\boldsymbol{x})$ is 1, and for $\boldsymbol{T}_{ii}(\boldsymbol{x}) > \boldsymbol{T}_{ij}(\boldsymbol{x}), \forall j \neq i$, we employ a transformation during the update learning process in our practical experiments.

We utilize a $C \times C$ weight matrix $\boldsymbol{W} = (w_{ij})$ to assist in the process. Denote matrix $\boldsymbol{A}$ as $\boldsymbol{A}_{ii} = 1 + \sigma(w_{ii})$ for all $i \in \{1, 2, \ldots, C\}$ and $\boldsymbol{A}_{ij} = \sigma(w_{ij})$ for all $i \neq j$ where σ is the sigmoid function. Then we do the normalization $\boldsymbol{T}_{ij} = \frac{\boldsymbol{A}_{ij}}{\sum_{k=1}^{C} \boldsymbol{A}_{kj}}$ to get the transition matrix $\boldsymbol{T}$.

Through this transformation, we ensure that the learned transition matrix has row sums equal to 1 and that the diagonal elements are the largest in each row. In practical experiments, we apply the diffusion modeling discussed in subsection 3.2 to the matrix $\boldsymbol{W}$, and then transform it into the transition matrix $\boldsymbol{T}$ for application. To simplify the notation, we uniformly use the term of transition matrix $\boldsymbol{W}$ to represent it, unless it leads to singularity.

4 Experiments

In this section, we present experimental findings to showcase the effectiveness of our proposed method compared to other methods. We evaluate our approach on both synthetic instance-dependent noisy datasets and real-world noisy datasets.

4.1 Datasets

We conduct experiments on following image classification datasets: CIFAR-10 and CIFAR-100 [13], CIFAR-10N and CIFAR-100N [32], Clothing1M [35], Webvision and ILSVRC12 [16]. Among them, CIFAR-10 and CIFAR-100 both have 32 × 32 × 3 color images including 50,000 training images and 10,000 test images. CIFAR-10 has 10 classes while CIFAR-100 has 100 classes. We generate instance-dependent noisy data on CIFAR-10 and CIFAR-100 with noise rates ranging from 10% to 50%, following the same generation method as in [33]. CIFAR-10N has three annotated labels, namely Random1, Random 2 and Random 3. The "Aggregate" is the aggregation of three noisy labels by majority voting, and the "Worst" is the dataset with the worst case. For CIFAR-100N, each image contains a coarse label and a fine label given by a human annotator. Clothing1M is a real-world dataset consisting of 1 million training images, consisting of 14 categories. WebVision contains 2.4 million images crawled from the websites using the 1,000 concepts in ImageNet ILSVRC12, but only the first 50 classes of the Google image subset are used in our experiments. For the validation set selection in our BTR method, we randomly sampled 10 samples from each observed class for each dataset to form the validation set, while the remaining samples were used for the training set.

4.2 Experimental Setup

For the pre-trained model, we employ the commonly used SimCLR model [4] from contrastive learning, which directly performs self-supervised learning on input images without utilizing additional datasets. For the diffusion model, we follow the setup similar to DDPM [11] to set $\beta_1 = 10^{-4}, \beta_M = 0.02$ and utilize a similar U-Net network architecture but we reduce the M from 1000 to 10 to accelerate the learning process. As for the classification network, it may vary depending on the specific dataset. More specifically, for CIFAR-10/10N, we use ResNet-18 as the backbone network with batch size 128 and learning rate 0.05. For CIFAR-100/100N, we use ResNet-34 network with batch size 128, learning rate 0.02. For clothing1M, we use a ResNet-50 pre-trained with 10 epochs, batch size 64, learning rate 0.002 for network and divided by 10 after the 5th epoch. We use InceptionResNetV2 network on Webvision, with 100 epochs, batch size 32, learning rate 0.02 for network and divided by 10 after the 30th and 60th epoch. For clustering, we utilize the K-means method, where the number of clusters is set to 10 times the number of classes in the datasets. For the initialization of transition matrix, the update method and setting are consistent with [17]. While the updates for other parameters are performed using the stochastic gradient descent optimization method.

Table 1: Test accuracy with instance-dependent noise on CIFAR-10/100.

	CIFAR-10				
	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	88.86±0.23	86.93±0.17	82.42±0.44	76.68±0.23	58.93±1.54
VolMinNet	89.97±0.57	87.01±0.64	83.80±0.67	79.52±0.83	61.90±1.06
PeerLoss	90.89±0.07	89.21±0.63	85.70±0.56	78.51±1.23	59.08±1.05
BLTM	90.45±0.72	88.14±0.66	84.55±0.48	79.71±0.95	63.33±2.75
PartT	90.32±0.15	89.33±0.70	85.33±1.86	80.59±0.41	64.58±2.86
MEIDTM	92.91±0.07	92.26±0.25	90.73±0.34	85.94±0.92	73.77±0.82
SOP	93.58±0.31	93.07±0.45	92.42±0.43	89.83±0.77	82.52±0.97
CC	95.24±0.20	93.68±0.12	93.31±0.46	**94.97±0.09**	91.19±0.34
LRA	95.87±0.42	94.70±0.28	93.79±0.40	92.72±0.29	90.95±0.43
DTM	**96.45±0.17**	**95.90±0.21**	**95.14±0.20**	94.82±0.31	**92.04±0.42**
	CIFAR-100				
	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	66.55±0.23	63.94±0.51	61.97±1.16	58.70±0.56	56.63±0.69
VolMinNet	67.78±0.62	66.13±0.47	61.08±0.90	57.35±0.83	52.60±1.31
PeerLoss	65.64±1.07	63.83±0.48	61.64±0.67	58.30±0.80	55.41±0.28
BLTM	68.42±0.42	66.62±0.85	64.72±0.64	59.38±0.65	55.68±1.43
PartT	67.33±0.33	65.33±0.59	64.56±1.55	59.73±0.76	56.80±1.32
MEIDTM	69.88±0.45	69.16±0.16	66.76±0.30	63.46±0.48	59.18±0.16
SOP	74.09±0.52	73.13±0.46	72.14±0.46	68.98±0.58	64.24±0.86
CC	80.52±0.22	79.61±0.19	77.34±0.31	76.58±0.25	72.68±0.36
LRA	81.20±0.16	80.53±0.29	78.22±0.19	76.55±0.31	72.97±0.51
DTM	**82.96±0.25**	**82.04±0.32**	**80.87±0.45**	**78.56±0.60**	**74.85±0.56**

4.3 Comparison Methods

In our experiments, we included the following common transition matrix and baseline methods as comparison: (1) VolMinNet [17], (2) PeerLoss [21] (3) BLTM [37], (4) PartT [33], (5) MEIDTM [6], as well as state-of-the-art methods for learning with noisy labels: (6) Co-teaching [10], (7) ELR+ [18], (8) DivideMix [14], (9) SOP and SOP+ [19], (10) PGDF [5], (11) CC [40], (12) LRA [3] with SimCLR as encoder similarly.

Table 2: Test accuracy on CIFAR-10N and CIFAR-100N.

	CIFAR-10N					CIFAR-100N
	Aggregate	Random 1	Random 2	Random 3	Worst	Noisy
Co-teaching	91.20±0.13	90.33±0.13	90.30±0.17	90.15±0.18	83.83±0.13	60.37±0.27
ELR+	94.83±0.10	94.43±0.41	94.20±0.24	94.34±0.22	91.09±1.60	66.72±0.07
DivideMix	95.01±0.71	95.16±0.19	94.89±0.23	95.03±0.20	92.56±0.42	71.13±0.48
SOP+	95.61±0.13	95.28±0.13	95.31±0.10	95.39±0.11	93.24±0.21	67.81±0.23
PGDF	95.35±0.12	94.95±0.21	94.78±0.34	94.92±0.28	94.22±0.29	67.76±0.35
CC	95.63±0.21	95.11±0.31	94.93±0.37	95.09±0.21	94.24±0.40	71.21±0.22
LRA	94.57±0.23	94.19±0.17	94.38±0.42	94.02±0.32	93.20±0.59	70.96±0.53
DTM	**96.13±0.17**	**95.98±0.22**	**96.01±0.28**	**95.78±0.34**	**94.93±0.21**	**72.51±0.30**

4.4 Experimental Results on Synthetic Datasets

We primarily validated our proposed method DTM against previous instance-based transition matrix methods on synthetic CIFAR-10/100 noise datasets. These methods mainly focus on estimating the transition matrix and some methods applicable to instance-dependent label noise. We performed 5 independent runs for each experimental configuration, and the average values and standard deviations of each experiment are presented in Table 1.

The results demonstrate that our proposed DTR method outperforms other methods of the same category across various noise rates. It is evident that traditional transition matrix methods for class-dependent noise as VolMinNet exhibit subpar performance when handling instance-dependent noise. While even advanced transition matrix methods for instance-dependent label noise such as BLTM, ParT and MEIDTM, still show significant gaps compared to our method.

Furthermore, as the noise rates increase, the test accuracy of existing transition matrix methods significantly decline. This is particularly pronounced in the case of CIFAR-100 with 50% instance-dependent noise (IDN) data, where all transition matrix methods achieve test accuracy below 60%. In contrast, our proposed DTR method achieves a remarkable test accuracy of 74.85%, showcasing its exceptional performance. That demonstrates relatively robust performance of DTM with only a slight decrease as the noise rate increases.

This experiment clearly demonstrates that there is a significant performance gap between previous transition matrix methods and other advanced techniques, such as CC and LRA, when dealing with instance-dependent noise problems. However, the experimental results indicate that our proposed method DTM, which incorporates the diffusion model into the estimation of the transition matrix, outperforms these advanced techniques, except for the case of 40% noise in CIFAR-100, where our method slightly underperforms CC. It is evident that by leveraging the diffusion modeling to estimate the transition matrix, we effectively incorporate the image's feature information, leading to a substantial improvement in the effectiveness of the transition matrix.

4.5 Experimental Results on Real-World Datasets

In addition to synthetic datasets, we also applied our method to real-world datasets and compared it with other state-of-the-art techniques for handling label noise problems. The results are presented in Table 2 and Table 3.

Table 3: Test accuracy on Clothing1M, Webvision and ILSVRC12.

	Clothing1M	Webvision	ILSVRC12
Co-teaching	69.2	63.6	61.5
ELR+	74.81	77.78	70.29
DivideMix	74.76	77.32	75.20
SOP+	74.98	77.60	75.29
PGDF	75.19	81.47	75.45
CC	75.40	79.36	76.08
LRA	75.32	80.05	76.64
DTM	**75.57**	**81.95**	**77.55**

The results demonstrate that regardless of the type of noise labels, whether it is aggregated, random, or the worst-case scenario in CIFAR-10N, as well as in CIFAR-100N with more label categories, our method consistently achieves the best results in handling real-world noise. When dealing with large datasets like Clothing1M and complex image datasets like Webvision, DTM also performs comparably to other state-of-the-art methods.

Through extensive experiments on five real-world datasets and the rusults on synthetic datasets above, our method outperforms the LRA method, which also utilizes the diffusion model for label noise problems. The LRA method models label diffusion with fewer dimensional information and lacks the rationale of our method, which considers noise generation from a transfer probability distribution perspective. The experiments demonstrate that our method achieves better learning performance by effectively integrating the transition matrix with the diffusion model.

4.6 Ablation Study

Besides the aforementioned experiments, we conducted ablation studies on proposed DTM method to assess the importance of each component. Table 4 presents the comparative results under 20% and 40% instance-dependent noise rates, where "w/o" denotes "without". We conducted ablation experiments on three components of our method, they are diffusion module, pre-trained encoder module, and clustering module respectively. "w/o diffusion" indicates directly using the features extracted by the pre-trained model for the classification task with the transition matrix. "w/o pre-train" means not extracting features through self-supervised learning and directly utilizing the classification network with the diffusion model. "w/o clustering" indicates that the initial transition matrix used for the diffusion model is the same for all samples.

Table 4: Ablation study of DTR. The data in the table represents the test accuracy.

	CIFAR-10		CIFAR-100	
	IDN-0.2	IDN-0.4	IDN-0.2	IDN-0.4
w/o pre-train	90.52	83.61	66.17	61.79
w/o clustering	92.25	88.35	71.93	66.47
w/o diffusion	93.74	91.66	79.82	73.51
DTR	**95.90**	**94.82**	**82.04**	**78.56**

From the results in Table 4, it can be observed that regardless of which component of diffusion module, pre-trained encoder module and clustering module is missing, the performance is consistently weaker compared to the original DTM. This indicates that each module plays a crucial role in our method. Our approach effectively combines the transition matrix, diffusion model, and pre-trained feature extraction, leading to significant improvements.

5 Conclusion

In this paper, we propose a method that models the transition matrix using diffusion models, incorporating the feature information extracted by a pre-trained encoder into the estimation of the transition matrix. This approach enables the model to handle instance-dependent label noise with a wider range of applicability. Experimental results on both synthetic and real-world noisy datasets demonstrate the effectiveness of our proposed method.

References

[1] Görkem Algan and Ilkay Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771, 2021.

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[3] Jian Chen, Ruiyi Zhang, Tong Yu, Rohan Sharma, Zhiqiang Xu, Tong Sun, and Changyou Chen. Label-retrieval-augmented diffusion models for learning from noisy labels. *arXiv preprint arXiv:2305.19518*, 2023.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[5] Wenkai Chen, Chuang Zhu, and Mengting Li. Sample prior guided robust model learning to suppress noisy labels. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–19. Springer, 2023.

[6] De Cheng, Tongliang Liu, Yixiong Ning, Nannan Wang, Bo Han, Gang Niu, Xinbo Gao, and Masashi Sugiyama. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16630–16639, 2022.

[7] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[8] Amit Daniely and Elad Granot. Generalization bounds for neural networks via approximate description length. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*, 2016.

[10] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, 31, 2018.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[12] Zhimeng Jiang, Kaixiong Zhou, Zirui Liu, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. An information fusion approach to learning with instance-dependent label noise. In *International Conference on Learning Representations*, 2021.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[14] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[15] Shikun Li, Xiaobo Xia, Hansong Zhang, Yibing Zhan, Shiming Ge, and Tongliang Liu. Estimating noise transition matrix with label correlations for noisy multi-label learning. *Advances in Neural Information Processing Systems*, 35:24184–24198, 2022.

[16] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[17] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*, pages 6403–6413. PMLR, 2021.

[18] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33:20331–20342, 2020.

[19] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, pages 14153–14172. PMLR, 2022.

[20] Yang Liu, Hao Cheng, and Kun Zhang. Identifiability of label noise transition matrix. In *International Conference on Machine Learning*, pages 21475–21496. PMLR, 2023.

[21] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR, 2020.

[22] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.

[23] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in Neural Information Processing Systems*, 26, 2013.

[24] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[25] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[26] Jun Shu, Qian Zhao, Zongben Xu, and Deyu Meng. Meta transition adaptation for robust deep learning with noisy labels. *arXiv preprint arXiv:2006.05697*, 2020.

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[28] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[30] Jialu Wang, Yang Liu, and Caleb Levy. Fair classification with group-dependent label noise. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 526–536, 2021.

[31] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[32] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

[33] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020.

[34] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32, 2019.

[35] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[36] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[37] Shuo Yang, Erkun Yang, Bo Han, Yang Liu, Min Xu, Gang Niu, and Tongliang Liu. Estimating instance-dependent bayes-label transition matrix using a deep neural network. In *International Conference on Machine Learning*, pages 25302–25312. PMLR, 2022.

[38] LIN Yong, Renjie Pi, Weizhong Zhang, Xiaobo Xia, Jiahui Gao, Xiao Zhou, Tongliang Liu, and Bo Han. A holistic view of label noise transition matrix in deep learning and beyond. In *The Eleventh International Conference on Learning Representations*, 2022.

[39] Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *International Conference on Machine Learning*, pages 12501–12512. PMLR, 2021.

[40] Ganlong Zhao, Guanbin Li, Yipeng Qin, Feng Liu, and Yizhou Yu. Centrality and consistency: two-stage clean samples identification for learning with instance-dependent noisy labels. In *European Conference on Computer Vision*, pages 21–37. Springer, 2022.

[41] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In *International Conference on Machine Learning*, pages 12912–12923. PMLR, 2021.

[42] Zhaowei Zhu, Jialu Wang, and Yang Liu. Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. In *International Conference on Machine Learning*, pages 27633–27653. PMLR, 2022.

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 content and contributions of the work are reflected in the abstract and introduction.

   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: In the experimental section, we analyze the applicability and limitations of our method.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [NA]

Justification: The focus of the work is on application and does not include a theoretical proof component.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: The paper provides a detailed description of the model construction and the specifics of the experimental data.

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: [No]

Justification: Upon acceptance of our paper, we will provide open-source code. The data we used is from commonly available open-source datasets.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: The experimental section of the paper provides details of the model and data.

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [Yes]

Justification: We conducted multiple repeated experiments to validate our approach and performed ablation experiments.

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: We list the relevant details in the experimental section.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We submitted the paper following the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We discuss the positive implications of our work and ensure it does not have any negative societal impact.

    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: There are no concerns in this regard regarding this work.

    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: The data and code used in our work are all publicly available and open-source.

    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 currently does not include any 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.