# Guiding Noisy Label Conditional Diffusion Models with Score-based Discriminator Correction

Supplementary Material

## A. Proof For Theorem

**Theorem 2** *Let $(\cdot, \mathbf{y}_r) \sim p_r$, $(\cdot, \mathbf{y}_f) \sim p_f$, and $D^t_\theta(\cdot,\cdot) = \sigma(g_\theta(\cdot,\cdot,t))$ be the logistic function of the discriminator. Assume $g$ satisfies the Lipschitz condition: there exists $L > 0$ such that for all $t \in [\epsilon, T]$, $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{y}$, we have $||g(\mathbf{x},\mathbf{y},t) - g(\mathbf{z},\mathbf{y},t)||_2^2 \leq L||\mathbf{x}-\mathbf{z}||_2^2$. Then, given an optimally trained $D_{\theta^*}$, we have*

$$\mathop{\mathbb{E}}_{\substack{\mathbf{x}_t,\mathbf{y},\tilde{\mathbf{y}},\\ \mathbf{y}_r,\mathbf{y}_f}}\left[\left\|\nabla_{\mathbf{x}_t}\log\frac{D_{\theta^*}(\mathbf{x}_t,\mathbf{y}_r)}{D_{\theta^*}(\mathbf{x}_t,\mathbf{y}_f)} - \nabla_{\mathbf{x}_t}\log\frac{p(\mathbf{x}_t|\mathbf{y})}{p(\mathbf{x}_t|\tilde{\mathbf{y}})}\right\|_2^2\right] \leq L + \mathop{\mathbb{E}}_{\mathbf{x}_t,\mathbf{y},\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t}\log\frac{p(\mathbf{x}_t|\mathbf{y})}{p(\mathbf{x}_t|\tilde{\mathbf{y}})}\right\|_2^2\right] \tag{10}$$

*Proof.* First, given the objective in Eq. (9), the optimal discriminator is easily derived as

$$D_{\theta^*}(\mathbf{x}_t,\mathbf{y}_r) = \frac{p(\mathbf{x}_t,\mathbf{y}_r)}{p(\mathbf{x}_t,\mathbf{y}_r)+p(\mathbf{x}_t,\mathbf{y}_f)}, \tag{14}$$

$$D_{\theta^*}(\mathbf{x}_t,\mathbf{y}_f) = \frac{p(\mathbf{x}_t,\mathbf{y}_f)}{p(\mathbf{x}_t,\mathbf{y}_r)+p(\mathbf{x}_t,\mathbf{y}_f)}, \tag{15}$$

Furthermore, we expose the relationship between $(\mathbf{y}_r, \mathbf{y}_f)$ and $\tilde{\mathbf{y}}$ as,

$$p(\mathbf{y}_r|\mathbf{x}_t) = p(\tilde{\mathbf{y}}, r|\mathbf{x}_t),\ p(\mathbf{y}_f|\mathbf{x}_t) = p(\tilde{\mathbf{y}}, f|\mathbf{x}_t), \tag{16}$$

$$p(\tilde{\mathbf{y}}|\mathbf{x}_t) = p(\tilde{\mathbf{y}},r|\mathbf{x}_t) + p(\tilde{\mathbf{y}}, f|\mathbf{x}_t). \tag{17}$$

Then, the gradient log ratio becomes

$$\begin{aligned}\nabla_{\mathbf{x}_t}\log\frac{D_{\theta^*}(\mathbf{x}_t,\mathbf{y}_r)}{D_{\theta^*}(\mathbf{x}_t,\mathbf{y}_f)} &= \nabla_{\mathbf{x}_t}\log\frac{p(\tilde{\mathbf{y}},r|\mathbf{x}_t)}{p(\tilde{\mathbf{y}},f|\mathbf{x}_t)}\\ &\overset{(i)}{=} \nabla_{\mathbf{x}_t}\log\frac{p_{\theta^*}(r|\mathbf{x}_t,\tilde{\mathbf{y}})}{1-p_{\theta^*}(r|\mathbf{x}_t,\tilde{\mathbf{y}})}\\ &\overset{(ii)}{=} \nabla_{\mathbf{x}_t}\log\frac{\sigma(g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}))}{1-\sigma(g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}))}\\ &= \nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}).\end{aligned}$$

where (i) is due to $p(\cdot,\tilde{\mathbf{y}}|\mathbf{x}_t) = p(\tilde{\mathbf{y}}|\mathbf{x}_t)p_{\theta^*}(\cdot|\tilde{\mathbf{y}},\mathbf{x}_t)$, (ii) is exactly the output of the discriminator, which is a sigmoid function. This allows us to break down the LHS in Eq. (10) to obtain

$$\begin{aligned}&\text{LHS}\\ &= \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_\theta(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right] - 2(F_1(\theta)-F_2(\theta)) + C_1\\ &= \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_\theta(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right] - \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_\theta(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right]\\ &\quad - 2F_1(\theta) + 2F_2(\theta) + \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_\theta(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right] + C_1,\end{aligned} \tag{18}$$

where

$$\begin{aligned}F_1(\theta^*) &= \mathop{\mathbb{E}}_{\mathbf{x}_t,\mathbf{y},\tilde{\mathbf{y}}}[\langle\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}), \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{y})\rangle],\\ F_2(\theta^*) &= \mathop{\mathbb{E}}_{\mathbf{x}_t,\mathbf{y},\tilde{\mathbf{y}}}[\langle\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}), \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\tilde{\mathbf{y}})\rangle],\\ C_1 &= \mathop{\mathbb{E}}_{\mathbf{x}_t,\mathbf{y},\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t}\log\frac{p(\mathbf{x}_t|\mathbf{y})}{p(\mathbf{x}_t|\tilde{\mathbf{y}})}\right\|_2^2\right] \quad \text{is a constant.}\end{aligned}$$

Equation (18) can be reformulated in the same manner as Song and Ermon [45], Vincent [49], which eventually becomes

$$\begin{aligned}\text{LHS} &= \mathop{\mathbb{E}}_{\mathbf{x}_0,\mathbf{x}_t,\mathbf{y},\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0,\mathbf{y})\right\|_2^2\right]\\ &\quad - \mathop{\mathbb{E}}_{\mathbf{x}_0,\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0,\tilde{\mathbf{y}})\right\|_2^2\right]\\ &\quad + \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right] + C_2\\ &\overset{(i)}{=} \mathop{\mathbb{E}}_{\mathbf{x}_0,\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0)\right\|_2^2\right]\\ &\quad - \mathop{\mathbb{E}}_{\mathbf{x}_0,\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}}) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0)\right\|_2^2\right]\\ &\quad + \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right] + C_2\\ &= \mathop{\mathbb{E}}_{\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t} g_{\theta^*}(\mathbf{x}_t,\tilde{\mathbf{y}})\right\|_2^2\right] + C_2,\end{aligned} \tag{19}$$

where (i) is the consequence of the unbiased estimator and

$$\begin{aligned}C_2 &= C_1 + \mathop{\mathbb{E}}_{\mathbf{x}_0,\mathbf{x}_t,\tilde{\mathbf{y}}}\left[\left\|\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0,\tilde{\mathbf{y}})\right\|_2^2\right]\\ &\quad - \mathop{\mathbb{E}}_{\mathbf{x}_0,\mathbf{x}_t,\mathbf{y}}\left[\left\|\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0,\mathbf{y})\right\|_2^2\right]\\ &= C_1\end{aligned} \tag{20}$$

Finally, from our assumption, $g_\theta(\cdot,\mathbf{y},t)$ has Lipschitz constant $L$, then we have

$$\text{LHS} \leq L + C_1 \tag{21}$$

## B. Synthetic Noisy Label Generation Process

**CIFAR-10** [27] is a $32 \times 32 \times 3$ color image dataset containing 10 classes, with 50,000 training and 10,000 test samples. As these datasets are assumed to be noise-free, we introduce three types of noisy labels for label injection. Below, we provide details on these noisy label-generation methods. Since symmetric noise involves simply assigning a random label, we omit further explanation.

**Asymmetric Noise (ASN).** [18, 51] For this type of noise, we followed the previous works by flipping label classes for CIFAR-10 as shown below.

| | | |
|---|---|---|
| Truck | $\Rightarrow$ | Automobile |
| Bird | $\Rightarrow$ | Airplane |
| Deer | $\Rightarrow$ | Horse |
| Cat | $\Leftrightarrow$ | Dog |

**Instance Dependent Noise (IDN)** We followed noise generation process as utilized at Cheng et al. [11], Xia et al. [52] as shown in Algorithm 3.

**Algorithm 3** Instance Dependent Noise Generation Process

**Require:** Clean samples $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$, Noise rate $\tau$

1: Sample instance flip rates $q \in \mathbb{R}^n$ from the truncated normal distribution $\mathcal{N}(\tau, 0.1^2, [0, 1])$;
2: Independently sample $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_c$ from the standard normal distribution $\mathcal{N}(0, 1^2)$;
3: **for** $i = 1, 2, ..., n$ **do**
4: $\quad \mathbf{p} = \mathbf{x}_i \times \mathbf{w}_{\mathbf{y}_i}$;
5: $\quad \mathbf{p}_{\mathbf{y}_i} = -\infty$;
6: $\quad \mathbf{p} = \mathbf{q}_i \times softmax(\mathbf{p})$;
7: $\quad \mathbf{p}_{\mathbf{y}_i} = 1 - \mathbf{q}_i$;
8: $\quad$ Randomly choose a label from the label space according to the possibilities $\mathbf{p}$ as noisy label $\tilde{\mathbf{y}}_i$;
9: **end for**

**Ensure:** Noisy samples $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)_{i=1}^n$

## C. Measure Confidence and Instability

Algorithm 4 directly estimates the confidence in the sampling process instead of the one-step denoise in the forward process. In our experiments, we use $N = 2,000$ samples to compute the mean $C(t)$ and $I(t)$.

## D. Guidelines for guidance interval searches ($S_{\text{clip\_min}}$, $S_{\text{clip\_max}}$).

In practice, we determine the interval as follows: (1) generate 1,000 samples from the pretrained models, while also storing intermediate denoised samples at each step $t$; (2) for each $t$, compute either the *confidence (C)* or *instability (I)* scores (Eqs. (6) and (7)); (3) finally, plot these values across $t$ and manually select ($S_{\text{clip\_min}}$, $S_{\text{clip\_max}}$) based on the observed trends. In Fig. 8, we provide an example using $C$ score on ImageNet, with the chosen $t \in (0.3, 45.0)$. The classifier $f(\cdot)$ choices range from small to large models and are publicly available via *huggingface*.

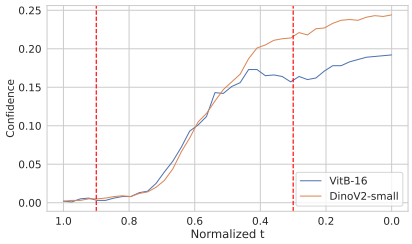



Figure 8. Confidence score on ImageNet 20% Symmetric noise. The classifiers are selected from open-source *huggingface*.

## E. Comparison with previous works

Chen et al. [9] suggests that a small perturbation may lead to a "better" diffusion model. However, we interpret this to image quality rather than image-condition alignment, as they use FID, IS, Precision, and Recall as standard metrics. This might be a capability of classifier-free guidance (CFG), where the "unconditional term" can mitigate some label noise. However, in Tab. 8, our findings indicate that CFG remains susceptible to label noise, as evidenced by Classification Accuracy Score (CAS) [40], though improved quality.

| **CIFAR-10** | | **Clean** | **Symm.** | **Asymm.** | **IDN** |
|---|---|---|---|---|---|
| **Metric** | | **0%** | **20%** | **20%** | **20%** |
| FID | (↓) | 5.68 | 5.11 | 5.29 | 4.99 |
| IS | (↑) | 3.83 | 4.24 | 3.95 | 4.29 |
| Precision | (↑) | 70.53 | 67.28 | 69.40 | 67.63 |
| Recall | (↑) | 50.57 | 55.50 | 52.60 | 55.63 |
| CAS | (↑) | **73.51** | 69.39 | 68.74 | 68.08 |

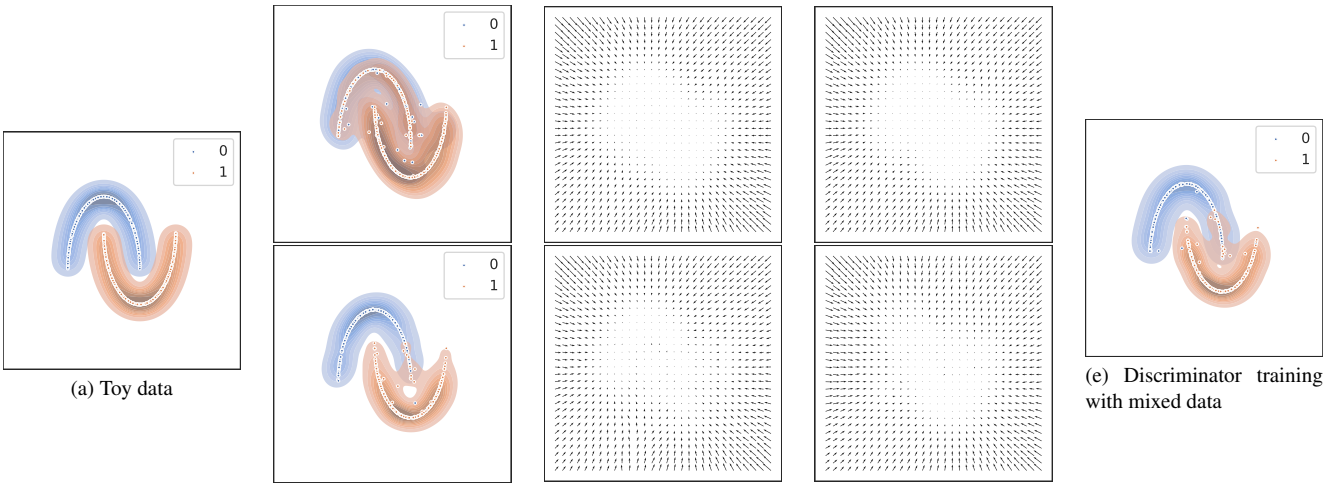
Table 8. CIFAR-10 trained with CFG, where label noises vary.

## F. Additional Results

In this section, we include more details about the sample selection and sampling of SBDC. All the experiments are run on 4 NVIDIA A100 GPUs.

### F.1. Toy Experiments

We show our analysis on the inter-twinning moon dataset in Fig. 9. The discriminator network is a 4-layered MLP with 256 neurons, 50% label of all samples are flipped and we only use half of the training data to train the discriminator network. Without proper guidance, the inaccurate score $s_\theta$ produced samples from a noisy distribution, as illustrated in Fig. 9b. In contrast, $s_\theta$ with the discriminator signal effectively steers $s_\theta$ towards $\nabla \log p_r^t$.




(a) Toy data (b) Generated samples (c) Class 0 gradient (d) Class 1 gradient (e) Discriminator training with mixed data

Figure 9. Comparison of the baseline model (first row) with our proposed method (second row). Discriminator Guidance is applied with $\gamma = 1.5$.

In practice, noise detection methods can still make mistakes in distinguishing samples. Therefore, we mixed corrupted samples into clean data at a ratio of 1:2 and vice versa for the discriminator training. Figure 9e shows that SBDC is mostly unaffected by noise in non-overlapping regions, while it may struggle to accurately classify samples in more complex regions. However, this can be mitigated by optimizing the parameters of the sampling process.

## F.2. Noise Detection Implementation Details

| | | CIFAR-10 | | | | | | CIFAR-100 | |
|---|---|---|---|---|---|---|---|---|---|
| | | 20 S | 20 A | 20 I | 40 I | 50 S | 80 S | 20 S | 40 S |
| P | (↑) | 87.1 | 87.8 | 87.7 | 94.7 | 96.2 | 79.8 | 94.1 | 97.5 |
| R | (↑) | 92.80 | **58.5** | 92.2 | 92.0 | 92.1 | 90.1 | 84.8 | 82.9 |
| F1 | (↑) | 89.84 | 70.2 | 89.9 | 93.3 | 94.1 | 84.6 | 89.2 | 89.6 |

Table 9. Performance of CORES on CIFAR-10 and CIFAR-100 datasets. S, A, and I denote Symmetric, Asymmetric, and IDN, respectively. Lower Recall undermines the robustness of the discriminator. This encourages prioritizing models capable of capturing more noisy data.

**Algorithm 4 Measurement of Confidence & Instability**

**Initialize** Pre-trained classifier $f$, $C[t] = list() \ \forall t \in \{1, \dots, T\}$, $I[t] = list() \ \forall t \in \{1, \dots, T-1\}$
**repeat**
  $\hat{\mathbf{x}}_T \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{y} \sim [K]$
  **for** $t = T, \dots, 1$ **do**
    Compute $\mathbf{y}^t_{\text{prev}} = f(\mathbf{x}_\theta(\hat{\mathbf{x}}_t, \mathbf{y}))$
    **if** $t \neq T$ **then**
      $I[t].append(\mathbb{1}[\mathbf{y}^t_{\text{prev}} \neq \mathbf{y}^t_{\text{prev}}])$
    **end if**
    $C[t].append(\mathbb{1}[\mathbf{y}^t_{\text{prev}} = \mathbf{y}])$
    Sample $\hat{\mathbf{x}}_{t-1} \sim q(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t, \mathbf{x}_\theta(\hat{\mathbf{x}}_t, \mathbf{y}))$
  **end for**
**until** $N$ iterations
**for** $t = T, \dots, 1$ **do**
  $C(t) = \texttt{numpy.mean}(C[t])$
**end for**
**for** $t = T-1, \dots, 1$ **do**
  $I(t) = \texttt{numpy.mean}(I[t])$
**end for**

**CORES.** To make the detection process of CORES more effective, we use the feature extracted from the pre-trained CLIP model, specifically the ViT-B/32 version, freeze it, and fine-tune the linear classifier layer. We also employ differentiable augmentations [61], containing horizontal flip, vertical flip, isotropic scaling, anisotropic scaling, fractional translation, and fractional rotation to provide the method with more diversity in detections. We train the model for 100 epochs, with a batch size of 64 on CIFAR-10 and CIFAR-100. We set the learning rate to 0.1 at the first 55 epochs and then decay it by a factor of 10 for the following training epochs. Other configurations are kept the same as specified in their paper. The noisy samples are taken from the last epoch and we report the corrupted samples detection performance of CORES at the last epoch in Table 9.

It shows the relatively high-quality detection of CORES in different settings. However, it remains suffering from asymmetric noise, which results in worse performance.

**Confident Learning.** We kept the setting of CL for Asymmetric noise the same as in Pleiss et al. [38], where the

| | | Tiny-Imagenet 200 | | |
|---|---|---|---|---|
| Metric | | EDM | TDSM | SBDC |
| FID | (↓) | 24.16 | 22.57 | **21.19** |
| IS | (↑) | 11.01 | 11.61 | **11.87** |
| Density | (↑) | 48.80 | 47.93 | **55.21** |
| Coverage | (↑) | 28.09 | 28.96 | **31.06** |

Table 10. Performance comparison on Tiny-Imagenet 200 dataset. We retrain the baseline and TDSM from scratch with their settings.

model selected is ResNet-50, training with the learning rate $0.1$ for epoch $[0, 150)$, $0.01$ for epoch $[150, 250)$, $0.001$ for epoch $[250, 350)$, momentum $0.9$, and weight decay $0.0001$.

### F.3. Real-world Experimental Details

Food101 [7] is a color image dataset containing 101 food categories, with 1,000 samples per category. Each class includes 250 images annotated by humans for testing, while the remaining 750 images have real-world label noise. Clothing-1M [54] is another real-world dataset with noisy labels, sourced from various online shopping websites, and consists of 1 million training images across 14 classes. We resize the dataset to resolution $64 \times 64$ for faster training. First, in the noise prediction stage, we use the default settings of CORES [11] and SIMIFEAT-r [63] for CLOTHING1M and FOOD101 respectively. Specifically, in each noise detection epoch, we use a batch size of 32 and sample 1000 mini-batches from the training data. The training process for CLOTHING1M has 120 epochs while FOOD101 has 100 epochs.

**Clean sample selection.** We label any sample flagged as noisy over the past 40 epochs as corrupted, except FOOD101, where high complexity leads to many clean samples being misclassified. We set an upper threshold of 5 to identify clean data. Both methods use the CLIP ViT-B/32 feature extractor.

**TinyImagenet Experiment.** We use CORES to filter the noisy data. Table 10 shows the proposed method consistently outperforms the baseline in the real-world dataset with a large number of classes.

### F.4. Experiments on ImageNet-128

We trained the diffusion models on ImageNet-128 with 20% Symmetric noise for 300K iterations. Table 11 presents the results with full trajectory guidance (SBDC-$f$) and with $\gamma$-gate (SBDC-$\gamma$), where $t \in [0.3, 45.0]$, demonstrating that SBDC remains effective even when scaling up.

### F.5. Sensitivity analysis on different noise detectors.

In Tab. 12, we present the effect of different noise detectors on the performance of SBDC. The results indicate that the performance does not vary significantly across detectors. In practice, the noise detector can be replaced with a strong pretrained classifier to enhance detection performance.

| **ImageNet** | **FID↓** | **IS↑** | **Pre.↑** | **Rec.↑** | **Den.↑** | **Cov.↑** |
|---|---|---|---|---|---|---|
| Original | 30.03 | 23.32 | 55.29 | **57.99** | 59.14 | 66.16 |
| SBDC-$f$ | **26.66** | **27.63** | **59.68** | 55.65 | **66.51** | **71.40** |
| SBDC-$\gamma$ | 26.94 | 27.19 | 59.40 | 55.91 | 66.17 | 71.33 |

Table 11. Comparison on ImageNet 20% Symm. noise.

| CIFAR-10 | | Orig. | CORES | AUM | Simi-v | Simi-r |
|---|---|---|---|---|---|---|
| ND Precision | (↑) | - | 94.7 | 94.18 | 88.75 | 85.30 |
| ND Recall | (↑) | - | 92.0 | 86.09 | 91.43 | 92.16 |
| FID | (↓) | 1.98 | 2.49 | 2.57 | 2.34 | 2.51 |
| Density | (↑) | 103.24 | 116.24 | 117.41 | 115.13 | 115.09 |
| Coverage | (↑) | 83.14 | 83.94 | 83.86 | 83.99 | 83.60 |
| CW-FID | (↓) | 29.72 | 13.81 | 14.28 | 13.96 | 14.50 |
| CW-Den. | (↑) | 75.39 | 106.77 | 108.07 | 104.96 | 105.32 |
| CW-Cov. | (↑) | 73.62 | 81.33 | 81.32 | 81.34 | 81.11 |

Table 12. CIFAR-10 40% IDN results under various ND methods.

### F.6. Extension to Others Pre-trained Diffusion Models

In Tab. 14, we show that the proposed method can also be extended to Variance-Preserving (VP) noise schedule [44] effectively and it indeed improves performance at different number of sampling steps.

### F.7. TDSM with Score-based Discriminator Correction

We conducted experiments by directly applying guidance on TDSM. We also report the result on Symmetric noise with 20% noise rate. The experimental results in Tabs. 13 and 15 show competitive results against EDM with SBDC.

### F.8. Comparison between SiMix and MixUp

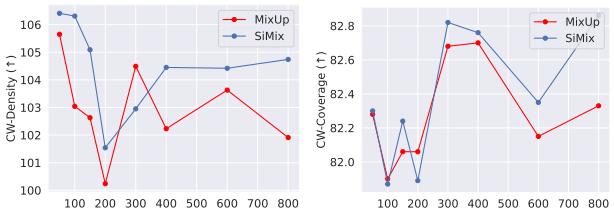



Figure 10. Comparison of quality and diversity for two data augmentation techniques throughout discriminator training.

Figure 10 shows that SiMix reduces artifact in the generation process, thereby improve the sample quality.

### F.9. Class correctioness evaluation.

Figure 11 shows the confusion matrix. SBDC indeed improves the class correctness.

| CIFAR-10 | | TDSM | SBDC (EDM) | SBDC (TDSM) | TDSM | SBDC (EDM) | SBDC (TDSM) | TDSM | SBDC (EDM) | SBDC (TDSM) |
|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | | Symmetric | | | Symmetric | | | Symmetric | | |
| | | 20% | | | 50% | | | 80% | | |
| FID | (↓) | **2.16** | 2.54 | 2.95 | 2.43 | **2.29** | 3.31 | 2.22 | **2.06** | 2.43 |
| IS | (↑) | **10.01** | **10.01** | 9.92 | 9.81 | **9.86** | 9.80 | 9.76 | **9.83** | 9.78 |
| Density | (↑) | 113.34 | 116.38 | **124.30** | 113.13 | 114.22 | **126.05** | 108.10 | 103.66 | **109.06** |
| Coverage | (↑) | **84.94** | 83.96 | 84.82 | 84.13 | 83.85 | **84.20** | **83.90** | 83.04 | 83.69 |
| CW-FID | (↓) | **11.00** | 11.81 | 11.79 | 18.20 | 16.89 | **14.47** | 59.93 | 48.69 | **37.11** |
| CW-Density | (↑) | 108.34 | 112.15 | **122.04** | 94.87 | 97.72 | **115.75** | 52.29 | 56.72 | **70.23** |
| CW-Coverage | (↑) | 83.66 | 82.60 | **83.94** | 79.18 | 79.46 | **81.54** | 56.44 | 59.09 | **67.98** |
| Metrics | | Asymmetric | | | Instance | | | Instance | | |
| | | 20% | | | 20% | | | 40% | | |
| FID | (↓) | 2.38 | **2.15** | 2.97 | 2.43 | **2.42** | 3.45 | **2.22** | 2.59 | 3.08 |
| IS | (↑) | 10.10 | 9.90 | **10.13** | 9.73 | **9.88** | 9.74 | 9.85 | **10.09** | 9.93 |
| Density | (↑) | 115.81 | 107.73 | **122.26** | 113.12 | 114.98 | **126.06** | 112.10 | 113.70 | **122.97** |
| Coverage | (↑) | **85.03** | 83.56 | 84.99 | **84.03** | 83.94 | 83.80 | **84.51** | 83.64 | 84.25 |
| CW-FID | (↓) | 10.94 | **10.61** | 11.95 | 12.05 | **11.63** | 12.78 | 18.10 | 14.68 | **13.59** |
| CW-Density | (↑) | 113.28 | 104.74 | **121.10** | 106.87 | 109.96 | **123.51** | 94.19 | 103.30 | **114.12** |
| CW-Coverage | (↑) | 84.32 | 82.86 | **84.38** | 82.57 | 82.52 | **82.86** | 80.08 | 80.34 | **82.21** |

Table 13. Performance comparison on CIFAR-10 dataset for different methods. We specify the model in which we apply SBDC in the parenthesis.

| CIFAR-10 | | 40% IDN Noise | | | | | |
|---|---|---|---|---|---|---|---|
| Metrics | | EDM | Ours | Clean | EDM | Ours | Clean |
| | | NFE=35 $(0.25, 0.75)$ | | | NFE=512 $(0.55, 0.85)$ | | |
| FID | (↓) | 9.56 | **8.85** | 9.70 | 2.22 | **2.21** | 2.16 |
| IS | (↑) | 10.17 | **10.27** | 10.30 | 9.85 | **9.87** | 9.98 |
| Density | (↑) | 81.67 | **95.28** | 83.54 | 103.66 | **106.63** | 107.88 |
| Coverage | (↑) | 70.98 | **71.61** | 71.80 | 83.02 | **83.73** | 84.26 |
| CW-FID | (↓) | 37.68 | **20.63** | 18.94 | 29.75 | **24.91** | 10.05 |
| CW-Den. | (↑) | 60.96 | **93.91** | 82.18 | 76.20 | **82.54** | 107.43 |
| CW-Cov. | (↑) | 61.43 | **70.37** | 71.32 | 74.04 | **76.56** | 84.16 |

Table 14. Performance comparison on CIFAR-10 dataset with VP noise schedule. The number in parenthesis corresponds to $(S_{\text{clip\_min}}, S_{\text{clip\_max}})$. The best results are in **bold**.

| CIFAR-100 | | TDSM | SBDC (EDM) | SBDC (TDSM) | TDSM | SBDC (EDM) | SBDC (TDSM) |
|---|---|---|---|---|---|---|---|
| Metrics | | Symmetric | | | Symmetric | | |
| | | 20% | | | 40% | | |
| FID | (↓) | 4.18 | **3.55** | 4.65 | 6.84 | **3.64** | 6.51 |
| IS | (↑) | 12.19 | 12.64 | **12.68** | 11.96 | 12.36 | **12.51** |
| Density | (↑) | 88.54 | 98.91 | **101.67** | 90.05 | 95.16 | **99.67** |
| Coverage | (↑) | 76.99 | **79.42** | 78.96 | 73.92 | **78.51** | 75.95 |
| CW-FID | (↓) | 76.77 | **68.94** | 69.45 | 91.13 | **77.13** | 78.05 |
| CW-Den. | (↑) | 72.21 | 90.48 | **94.04** | 61.27 | 76.76 | **82.24** |
| CW-Cov. | (↑) | 72.36 | **76.61** | 76.39 | 65.50 | **72.03** | 70.95 |

Table 15. Performance comparison on CIFAR-100 dataset for different methods. We specify the model in which we apply SBDC in the parenthesis.

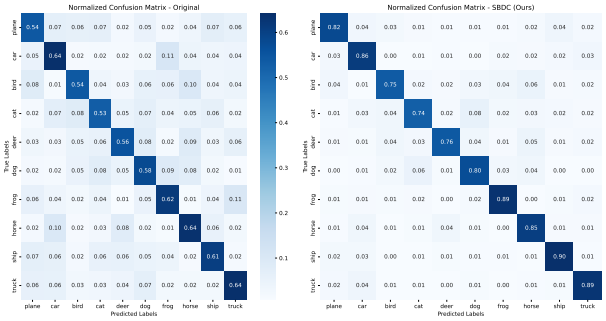
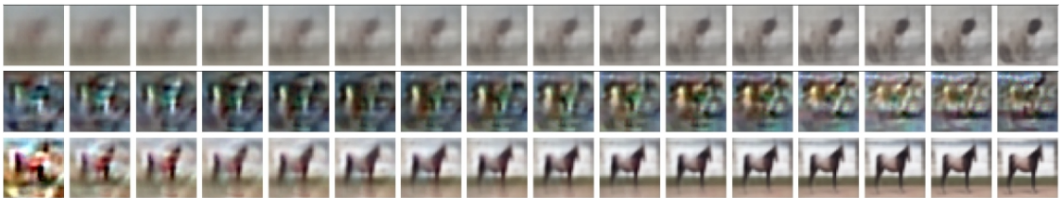



Figure 11. Confusion matrix on CIFAR10 - 40% IDN.

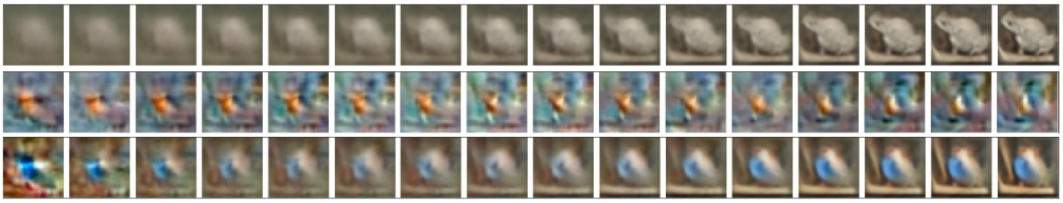
## F.10. Extended Examples

Figure 12 visualizes the generation process of two instances with class "horse" and "bird".

Figs. 13 to 16 show the uncurated generated images of the baseline, TDSM, and our models. Our models exhibit proficiency in producing signals to fix generated images.



(a) Instance of class ”horse”



(b) Instance of class ”bird”

Figure 12. Illustration of the sampling process with correct guidance for two initial noises. The first row is the origin, and the second and third rows are the discriminator gradient, and the refined denoise prediction, respectively.

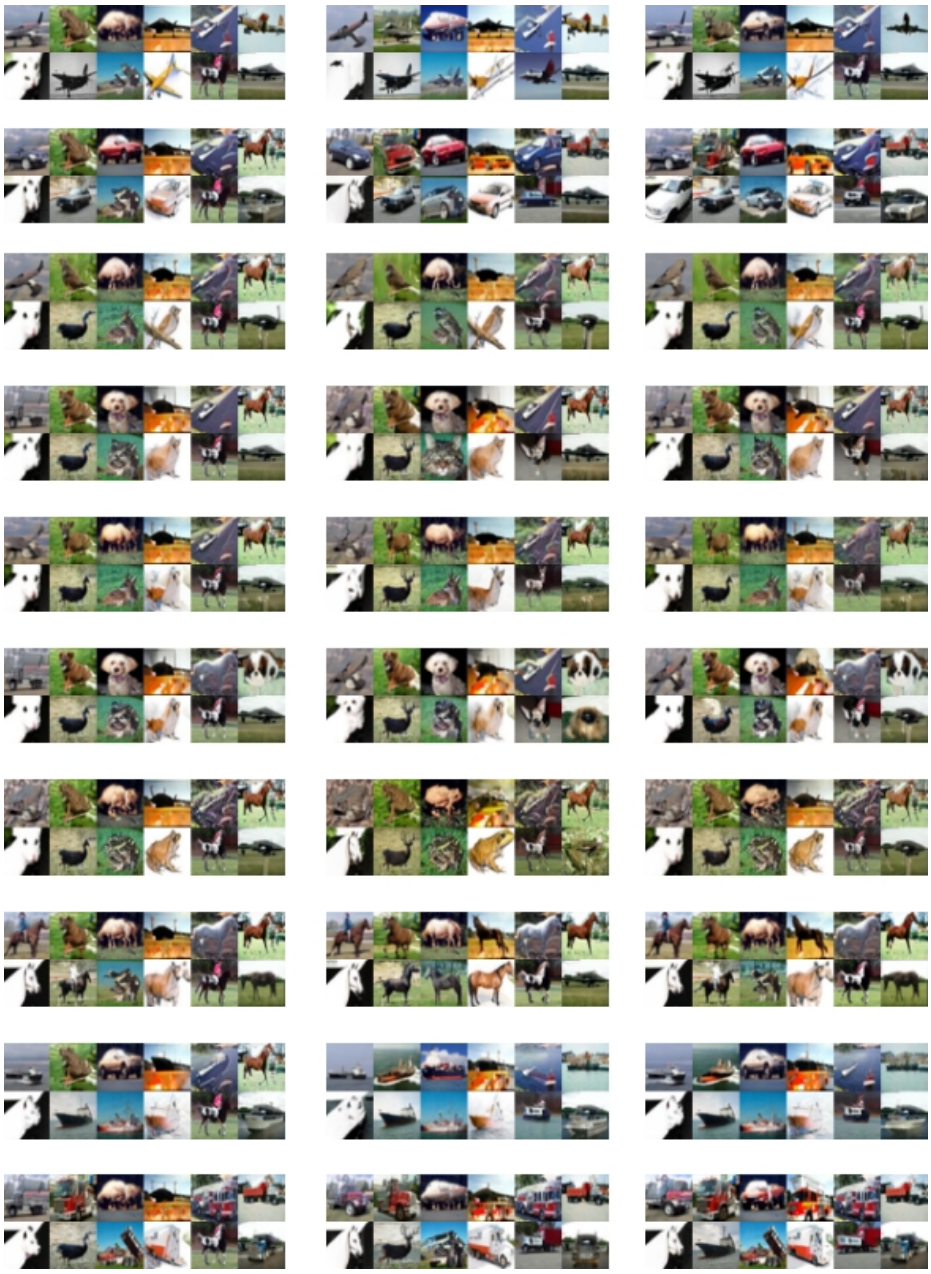


Figure 13. The uncurated generated images of baseline (left column), TDSM (middle column), and SBDC (right column) on the CIFAR-10 dataset with 50% symmetric noise. Each block has images of the same class. The class labels are *plane, car, bird, cat, deer, dog, frog, horse, ship, truck*, from top to bottom. All images are generated from the same noise.

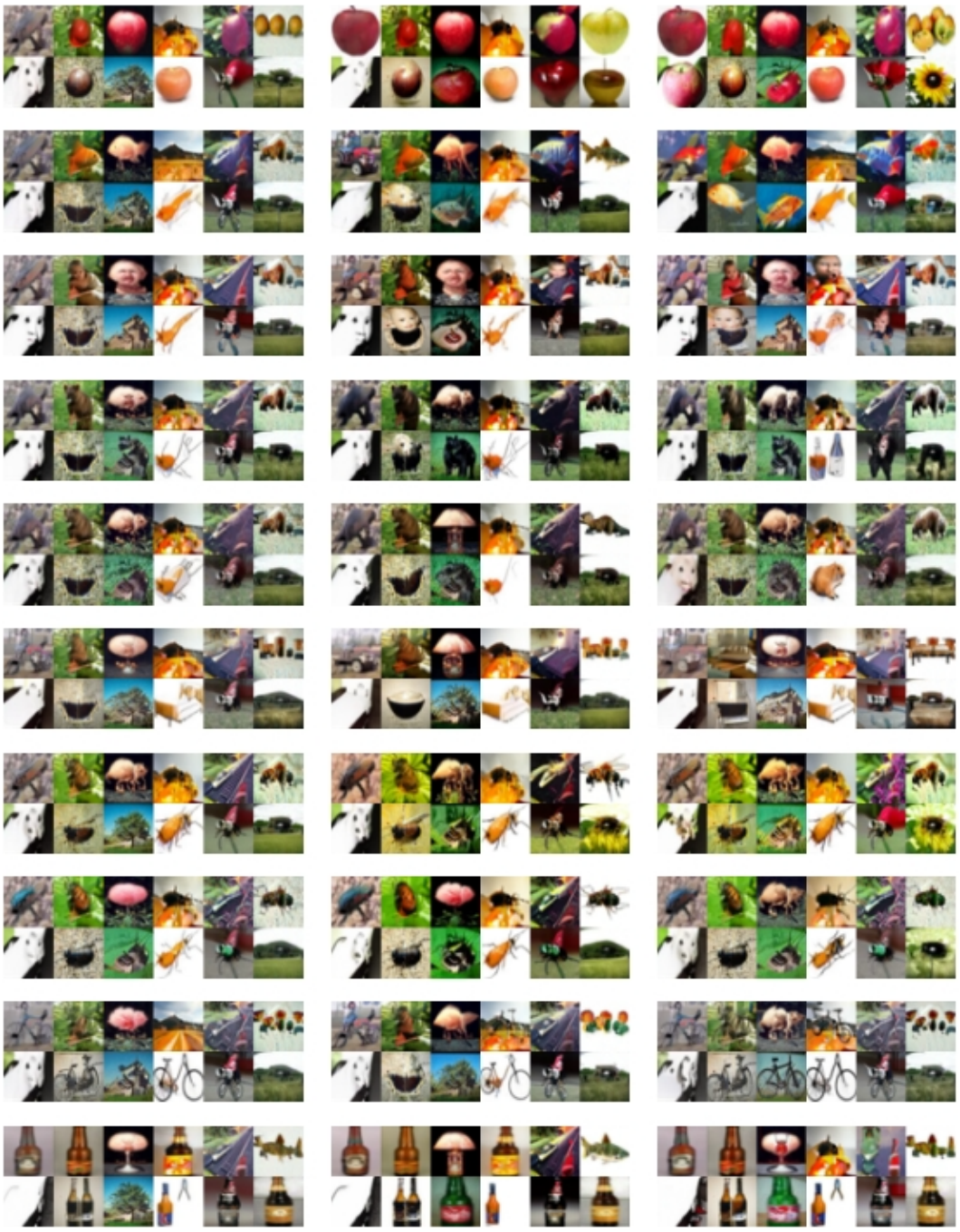


Figure 14. The uncurated generated images of baseline (left column), TDSM (middle column), and SBDC (right column) on the CIFAR-100 dataset with 40% symmetric noise. Each block has images of the same class. The class labels are *apple, aquarium fish, baby, bear, beaver, bed, bee, beetle, bicycle, bottle*, from top to bottom.

Figure 15. The uncurated generated images of baseline (left column), TDSM (middle column), and SBDC (right column) on the FOOD101 dataset. Each block has images of the same class. The class labels are *apple pie, baby back ribs, baklava, beef carpaccio, beef tartare, beet salad, beignets, bibimbap, bread pudding, breakfast burrito*, from top to bottom.

Figure 16. The uncurated generated images of baseline (left column), TDSM (middle column), and SBDC (right column) on the Clothing1M dataset. Each block has images of the same class. The class labels are *t-shirt, shirt, knitwear, chiffon, sweater, hoodie, and windbreaker*, from top to bottom.