# Efficient Dataset Distillation via Minimax Diffusion

Jianyang Gu[1], Saeed Vahidian[2], Vyacheslav Kungurtsev[3], Haonan Wang[4], Wei Jiang[1], Yang You[4], Yiran Chen[2]

[1] Zhejiang University, [2] Duke University, [3] Czech Technical University, [4] National University of Singapore

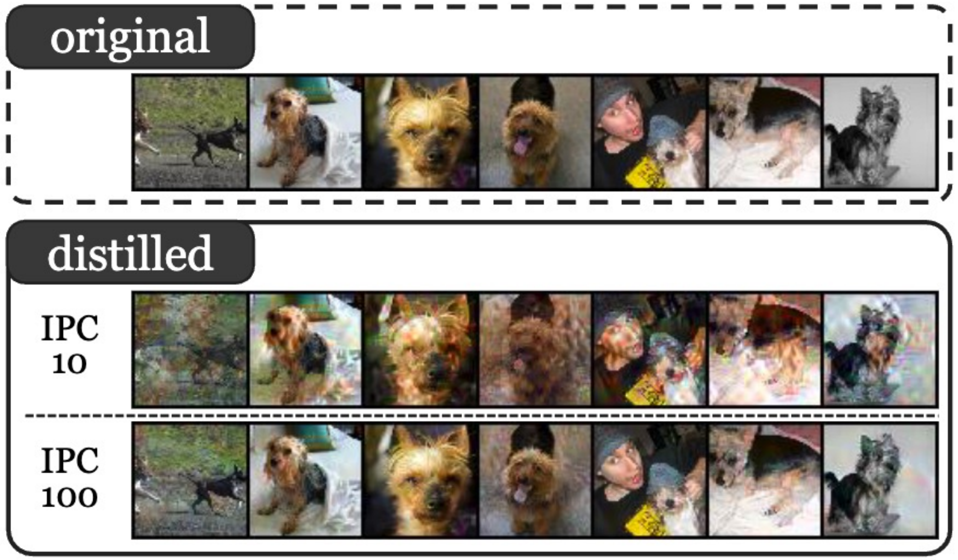
# Minimax Diffusion

## Background

Previous DD methods engage in sample-wise optimization at the pixel level or embedidng level. Such scheme suffers from problems of two perspectives:

1) The parameter space is positively correlated with the size of the target surrogate dataset, leading to more time and computational resource demands for distilling larger datasets.

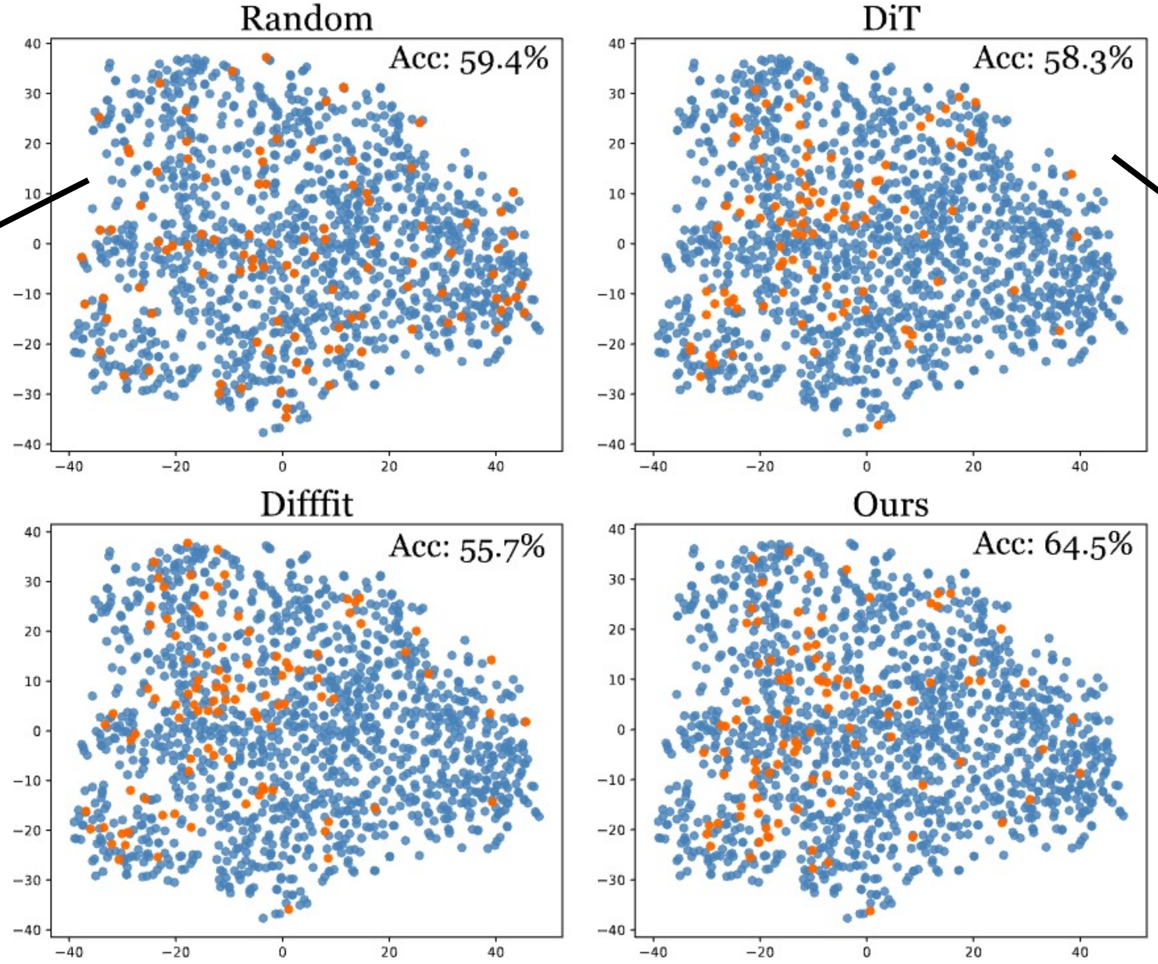
2) The larger parameter space also increases the optimization complexity. Distilling larger-IPC datasets generates smaller pixel modifications.




We intend to incorporate diffusion models to design a more efficient dataset distillation scheme.

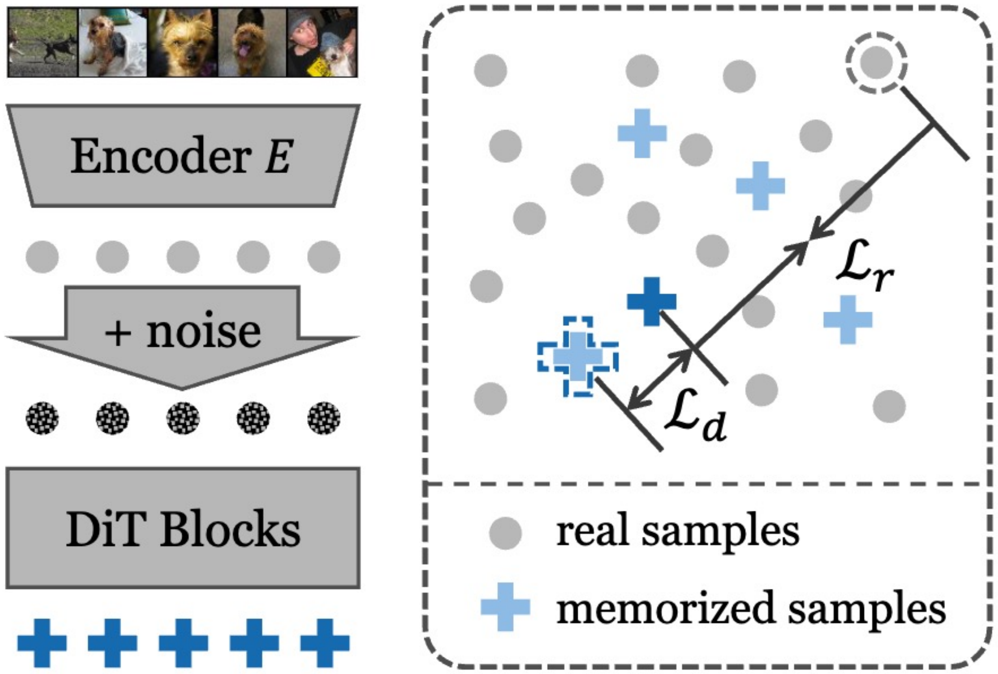
## Empirical Study

Good in diveristy, but not emphasizing the high-density area




The representativeness is good, but lacking diversity

## Method




$\mathcal{M}$: storing original embeddings

$\mathcal{D}$: storing predicted embeddings

---

**Algorithm 1:** Minimax Diffusion Fine-tuning

---

**Input:** initialized model parameter $\theta$, original dataset $\mathcal{T} = \{(\mathbf{x}, y)\}$, encoder $E$, class encoder $E_c$, time step $t$, variance schedule $\bar{\alpha}_t$, real embedding memory $\mathcal{M}$, predicted embedding memory $\mathcal{D}$

**Output:** optimized model parameter $\theta^*$

**for** *each step* **do**

Obtain the original embedding: $\mathbf{z}_0 = E(\mathbf{x})$

Obtain the class embedding: $\mathbf{c} = E_c(y)$

Sample random noise: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Add noise to the embedding: $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$

Predict the noise $\epsilon_\theta(\mathbf{z}_t, \mathbf{c})$ and recovered embedding $\hat{\mathbf{z}}_\theta(\mathbf{z}_t, \mathbf{c}) = \mathbf{z}_t - \epsilon_\theta(\mathbf{z}_t, \mathbf{c})$

Update the model parameter with Eq. (5)

Enqueue the real embedding: $\mathcal{M}_r \leftarrow \mathbf{z}_0$

Enqueue the predicted embedding: $\mathcal{M}_d \leftarrow \hat{\mathbf{z}}_\theta(\mathbf{z}_t, \mathbf{c})$

**end**

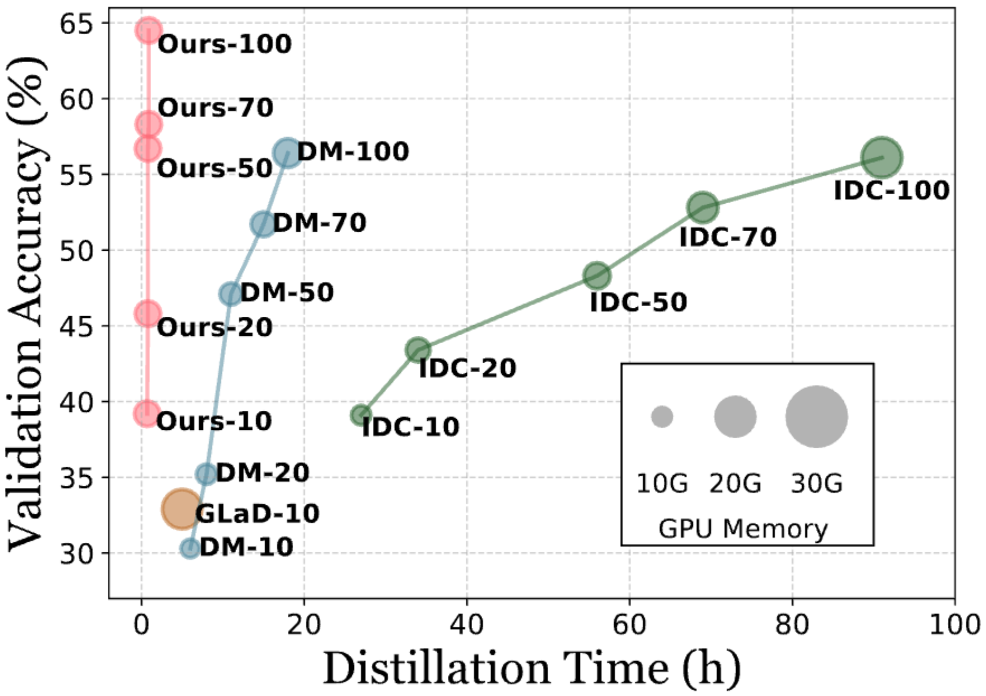
---

$$\mathcal{L}_r = \arg\max_{\theta} \min_{m \in [N_M]} \sigma\left(\hat{\mathbf{z}}_\theta(\mathbf{z}_t, \mathbf{c}), \mathbf{z}_m\right).$$

$$\mathcal{L}_d = \arg\min_{\theta} \max_{d \in [N_D]} \sigma\left(\hat{\mathbf{z}}_\theta(\mathbf{z}_t, \mathbf{c}), \mathbf{z}_d\right).$$

# Minimax Diffusion

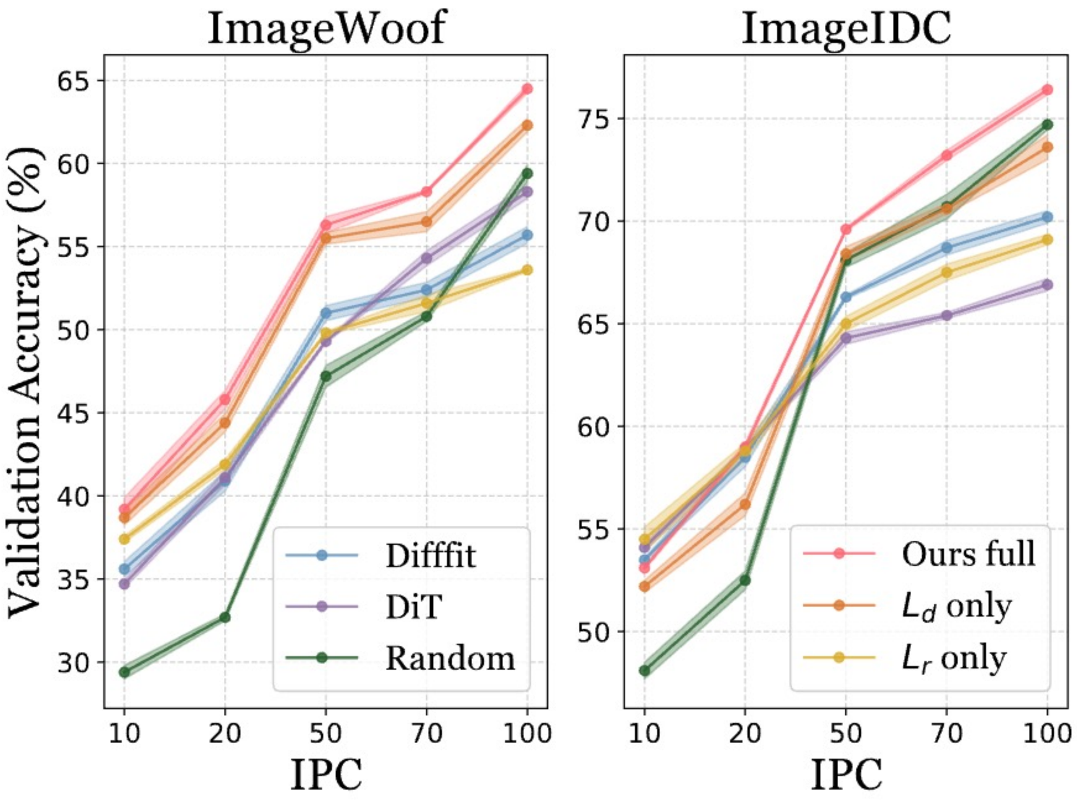
## Experiments




| IPC (Ratio) | Test Model | Random | DiT [33] | DM [53] | IDC-1 [21] | GLaD [4] | Ours |
|---|---|---|---|---|---|---|---|
| 10 (0.8%) | ConvNet-6 | $24.3_{\pm1.1}$ | $34.2_{\pm1.1}$ | $26.9_{\pm1.2}$ | $33.3_{\pm1.1}$ | $33.8_{\pm0.9}$ | $\mathbf{37.0}_{\pm1.0}$ |
| | ResNetAP-10 | $29.4_{\pm0.8}$ | $34.7_{\pm0.5}$ | $30.3_{\pm1.2}$ | $39.1_{\pm0.5}$ | $32.9_{\pm0.9}$ | $\mathbf{39.2}_{\pm1.3}$ |
| | ResNet-18 | $27.7_{\pm0.9}$ | $34.7_{\pm0.4}$ | $33.4_{\pm0.7}$ | $37.3_{\pm0.2}$ | $31.7_{\pm0.8}$ | $\mathbf{37.6}_{\pm0.9}$ |
| 20 (1.6%) | ConvNet-6 | $29.1_{\pm0.7}$ | $36.1_{\pm0.8}$ | $29.9_{\pm1.0}$ | $35.5_{\pm0.8}$ | - | $\mathbf{37.6}_{\pm0.2}$ |
| | ResNetAP-10 | $32.7_{\pm0.4}$ | $41.1_{\pm0.8}$ | $35.2_{\pm0.6}$ | $43.4_{\pm0.3}$ | - | $\mathbf{45.8}_{\pm0.5}$ |
| | ResNet-18 | $29.7_{\pm0.5}$ | $40.5_{\pm0.5}$ | $29.8_{\pm1.7}$ | $38.6_{\pm0.2}$ | - | $\mathbf{42.5}_{\pm0.6}$ |
| 50 (3.8%) | ConvNet-6 | $41.3_{\pm0.6}$ | $46.5_{\pm0.8}$ | $44.4_{\pm1.0}$ | $43.9_{\pm1.2}$ | - | $\mathbf{53.9}_{\pm0.6}$ |
| | ResNetAP-10 | $47.2_{\pm1.3}$ | $49.3_{\pm0.2}$ | $47.1_{\pm1.1}$ | $48.3_{\pm1.0}$ | - | $\mathbf{56.3}_{\pm1.0}$ |
| | ResNet-18 | $47.9_{\pm1.8}$ | $50.1_{\pm0.5}$ | $46.2_{\pm0.6}$ | $48.3_{\pm0.8}$ | - | $\mathbf{57.1}_{\pm0.6}$ |
| 70 (5.4%) | ConvNet-6 | $46.3_{\pm0.6}$ | $50.1_{\pm1.2}$ | $47.5_{\pm0.8}$ | $48.9_{\pm0.7}$ | - | $\mathbf{55.7}_{\pm0.9}$ |
| | ResNetAP-10 | $50.8_{\pm0.6}$ | $54.3_{\pm0.9}$ | $51.7_{\pm0.8}$ | $52.8_{\pm1.8}$ | - | $\mathbf{58.3}_{\pm0.2}$ |
| | ResNet-18 | $52.1_{\pm1.0}$ | $51.5_{\pm1.0}$ | $51.9_{\pm0.8}$ | $51.1_{\pm1.7}$ | - | $\mathbf{58.8}_{\pm0.7}$ |
| 100 (7.7%) | ConvNet-6 | $52.2_{\pm0.4}$ | $53.4_{\pm0.3}$ | $55.0_{\pm1.3}$ | $53.2_{\pm0.9}$ | - | $\mathbf{61.1}_{\pm0.7}$ |
| | ResNetAP-10 | $59.4_{\pm1.0}$ | $58.3_{\pm0.8}$ | $56.4_{\pm0.8}$ | $56.1_{\pm0.9}$ | - | $\mathbf{64.5}_{\pm0.2}$ |
| | ResNet-18 | $61.5_{\pm1.3}$ | $58.9_{\pm1.3}$ | $60.2_{\pm1.0}$ | $58.3_{\pm1.2}$ | - | $\mathbf{65.7}_{\pm0.4}$ |

Minimax diffusion surpasses other methods with much less requirement on training time and computational resources.

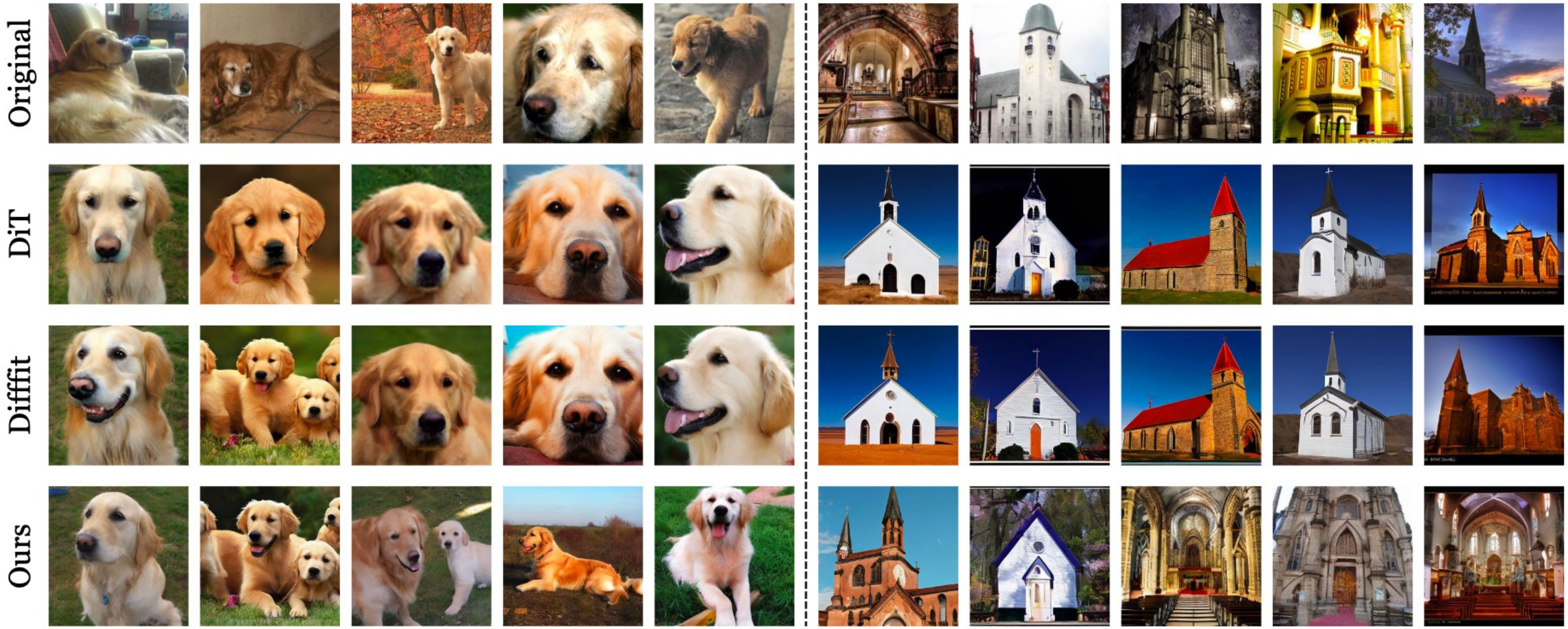
# Experiments




| $\mathcal{L}_r$ | $\mathcal{L}_r$ w\m | $\mathcal{L}_d$ | $\mathcal{L}_d$ w\m | ImageWoof 10-IPC | ImageWoof 50-IPC | ImageIDC 10-IPC | ImageIDC 50-IPC |
|---|---|---|---|---|---|---|---|
| - | - | - | - | $35.6_{\pm 0.9}$ | $51.0_{\pm 0.9}$ | $53.5_{\pm 0.2}$ | $66.3_{\pm 0.2}$ |
| ✓ | - | - | - | $34.4_{\pm 1.1}$ | $47.1_{\pm 0.5}$ | $49.6_{\pm 0.7}$ | $60.2_{\pm 1.2}$ |
| - | ✓ | - | - | $37.4_{\pm 0.4}$ | $49.5_{\pm 1.0}$ | $54.5_{\pm 1.2}$ | $65.0_{\pm 0.8}$ |
| - | - | ✓ | - | $35.7_{\pm 0.8}$ | $48.3_{\pm 0.6}$ | $51.5_{\pm 0.6}$ | $64.8_{\pm 0.8}$ |
| - | - | - | ✓ | $38.7_{\pm 0.9}$ | $54.9_{\pm 0.7}$ | $52.2_{\pm 0.6}$ | $68.4_{\pm 0.7}$ |
| ✓ | - | ✓ | - | $38.3_{\pm 0.5}$ | $54.9_{\pm 0.4}$ | $53.3_{\pm 0.5}$ | $66.8_{\pm 0.5}$ |
| - | ✓ | - | ✓ | $39.2_{\pm 1.3}$ | $56.3_{\pm 1.0}$ | $53.1_{\pm 0.2}$ | $69.6_{\pm 0.2}$ |

The representativeness constraint improves performance on small IPCs. The diveristy constraint brings larger performance improvement. But grouping them together achieves the best performance.

## Experiments




Minimax diffusion leads to better representativeness and diversity for the generated images.

# Thanks