

---

# Supplementary Material for Unlocking Dataset Distillation with Diffusion Models

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Definitions of DC, DM, and MTT

2 **Dataset Condensation (DC)** ensures alignment by deriving the gradients via a classification error  
3 [6]. It calculates the loss on real ( $\ell^{\mathcal{T}}$ ) and the respective synthetic data ( $\ell^{\mathcal{S}}$ ). Next, it minimizes the  
4 distance between the gradients of both network instances. More concretely,

$$\mathcal{L}_{DC} = 1 - \frac{\nabla_{\theta} \ell^{\mathcal{S}}(\theta) \cdot \nabla_{\theta} \ell^{\mathcal{T}}(\theta)}{\|\nabla_{\theta} \ell^{\mathcal{S}}(\theta)\| \|\nabla_{\theta} \ell^{\mathcal{T}}(\theta)\|}. \quad (1)$$

5 **Distribution Matching (DM)** obtains gradients by minimizing the logits on the real and synthetic  
6 datasets. It enforces the feature extractor (ConvNet) to produce similar features for real and synthetic  
7 images [5]. The distribution matching loss is

$$\mathcal{L}_{DM} = \sum_c \left\| \frac{1}{|\mathcal{T}_c|} \sum_{\mathbf{x} \in \mathcal{T}_c} \psi(\mathbf{x}) - \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{s} \in \mathcal{S}_c} \psi(\mathbf{s}) \right\|^2, \quad (2)$$

8 where  $\mathcal{T}_c, \mathcal{S}_c$  are the real and synthetic images for a class  $c$ .

9 **Matching Training Trajectories (MTT)** concentrates on the trajectory of network parameters [1]. In  
10 more detail, MTT exploits several trained instances of a model, called experts, and stores the training  
11 trajectory of parameters  $\{\theta_t^*\}_0^T$  at predetermined intervals, called expert trajectories. For dataset  
12 distillation, MTT samples a random set of parameters  $\theta_t^*$  from the trajectory at a given timestamp.  
13 Next, it trains a new network,  $\hat{\theta}_{t+N}$ , initialized with the parameters on the respective synthetic images  
14 (for  $N$  iterations). Finally, the distance between the trajectory on the real dataset,  $\theta_{t+M}^*$  with  $M$  steps,  
15 and the trajectory on the synthetic one,  $\hat{\theta}_{t+N}$ , is minimized. As a result, MTT tries to mimic the  
16 original dataset’s training path (trajectory of parameters) with the synthetic images:

$$\mathcal{L}_{MTT} = \frac{\|\hat{\theta}_{t+N} - \theta_{t+M}^*\|^2}{\|\theta_t^* - \theta_{t+M}^*\|^2}. \quad (3)$$

## 17 B Justification for Modified Reverse Process in Distillation

18 Our core modification to the reverse diffusion step is crucial for improving the gradient flow back to  
19 the initial latent  $\mathcal{Z}$ , allowing end-to-end optimization. A natural question arises regarding the validity  
20 of this modified process, as it intentionally deviates from standard diffusion sampling procedures  
21 designed for high-fidelity image generation that perfectly match the learned data distribution.

22 **Different Objectives: Distillation vs. Faithful Sampling.** The key insight is that the objective  
23 of dataset distillation differs fundamentally from standard image generation. Distillation aims to  
24 synthesize a small set of maximally informative samples that enable efficient training of downstream  
25 models, prioritizing the encoding of essential class-discriminative features over photorealism [4, 1, 6].



Figure 1: Influence of our modified reverse process in a classical image generation setting (unconditional FFHQ). It shows that the residual connections alter the generation process significantly, leading to abstract artifacts and the loss of coherence expected in a facial dataset: **(top)** with modification and **(bottom)** without modification.

Pixel-perfect adherence to the original data distribution is not necessarily required or even optimal; abstract or stylized images often yield excellent distillation performance if they capture core class characteristics effectively [2].

**Impact of Modification.** Our modification introduces a direct dependency on the initial noisy state  $\mathbf{z}_T$  throughout the reverse process. While preserving the Markov property (as shown in the main paper) and the representational power of the pre-trained denoiser  $f_\theta$ , this change means the resulting process  $p_\theta(\mathbf{z}_0|\mathbf{z}_T, \mathbf{c})$  no longer guarantees sampling exactly from the original distribution  $\mu$  learned by the LDM. If applied without the corrective feedback loop of distillation optimization (e.g., for unconditional generation), this modification can lead to more abstract outputs that deviate from the expected style, as illustrated with an FFHQ-trained model in Figure 1. This deviation is expected, as the process is no longer constrained solely by the standard denoising objective. We also observe a slight reduction in sample diversity (measured by average LPIPS between generated samples: 0.386 with modification vs. 0.420 without, on ImageNet samples), likely due to the persistent influence of the fixed  $\mathbf{z}_T$ .

**Suitability for Distillation.** Crucially, within the dataset distillation framework, the latent codes  $\mathcal{Z}$  (and conditioning  $\mathbf{c}$ ) are continuously optimized to minimize the distillation loss  $\mathcal{L}$ . This optimization process actively counteracts potential adverse effects of the modified sampling path by guiding the generation towards producing images (even if abstract) that are highly effective for the downstream task defined by  $\mathcal{L}$ . The essential properties needed are: (1) sufficient generative capacity to create diverse, class-relevant features, which the LDM provides; and (2) strong gradient flow for optimization, which our modification enables (as also shown in Figure 2). The empirical success of LD3M - significantly outperforming GLaD and AE-only baselines, and performing robustly even with randomly initialized LDMs - demonstrates that this trade-off (sacrificing perfect distribution matching for tractable optimization) is highly beneficial for the specific goal of dataset distillation. The resulting "abstract" representations effectively encode class information for robust generalization. Therefore, while distinct from standard sampling, our modified reverse process is a well-justified and necessary component for unlocking diffusion models for effective, end-to-end dataset distillation.

## C Hyper-Parameters for Distillation Algorithms

**LDM.** For all our LDM experiments, we set the unconditional guidance scale to the default value of 3. For  $128 \times 128$  images, we used max. time steps of 10, and for  $256 \times 256$  images, we used 20.

**DC.** We utilize a learning rate of  $10^{-3}$  throughout our DC experiments to update the latent code representation and the conditioning information.



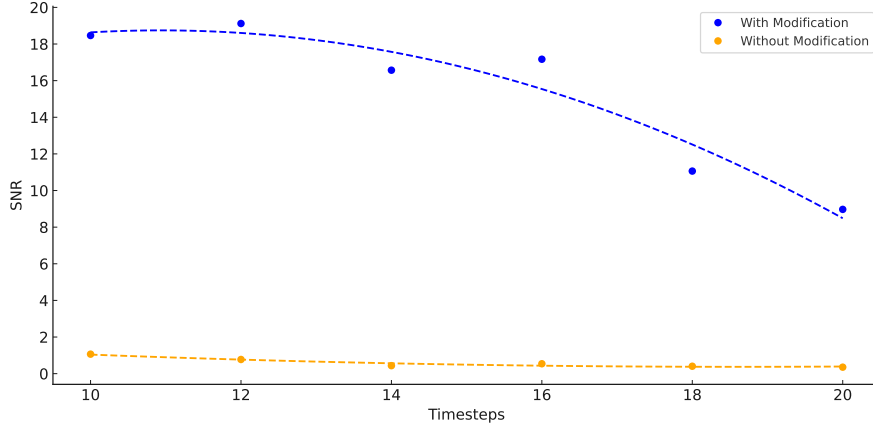


Figure 2: Gradient flow analysis comparing the Signal-to-Noise Ratio (SNR) of gradient norms for LD3M with and without our modification. Diffusion demonstrates a more stable gradient flow, indicating enhanced optimization dynamics. Dashed lines show a polyfit plot to highlight the trends.

Table 1: Common hyperparameters for training the distillation algorithms used in this work.

Parameter	Value
DSA Augmentations	Color / Crop / Cutout / Flip / Scale / Rotate
Iteration (Distillation)	5,000 (128 × 128) / 10,000 (256 × 256)
Momentum	0.5
Batch Real	256
Batch Train	256
Batch Test	128

58 **DM.** In every DM experiment, we adopt a learning rate of  $10^{-2}$ , applying it to updates of the latent  
59 code representation alongside the conditioning information.

60 **MTT.** For MTT experiments, a uniform learning rate of 10 is applied to update the latent code  
61 representation and the conditioning information. We buffered 5,000 trajectories for expert training,  
62 each with 15 training epochs. We used ConvNet-5 and InstanceNorm. During dataset distillation, we  
63 used three expert epochs, max. start epoch of 5 and 20 synthetic steps.

## 64 D Large Scale Datasets

65 Although LD3M is compatible with various distillation algorithms -including DC, DM, and MTT  
66 -our current experiments focus on baseline variants that do not leverage inter-class relationships  
67 during optimization. This is an essential avenue for further improvement: incorporating inter-  
68 class information (e.g., through contrastive losses or hierarchical label structures) may enhance the  
69 discriminative quality of the synthetic data. Future work will explore how LD3M’s expressive latent  
70 trajectories can be used to facilitate such structured, cross-class-aware distillation.

## 71 E Limitations

72 While LD3M improves dataset distillation compared to GLaD, it is essential to acknowledge certain  
73 limitations. A primary concern arises from the linear addition in the diffusion process, which may  
74 not sufficiently combat the vanishing gradient problem for larger time steps, as observed in our  
75 experiments [3]. Further alternative strategies for integrating the initial state  $\mathbf{z}_T$  in the diffusion  
76 process should be evaluated to address this issue, e.g., non-linear progress towards 0 as  $t$  approaches 0.  
77 These alternative approaches could offer more nuanced and dynamic ways to manage the influence of  
78  $\mathbf{z}_T$  across different stages of the diffusion, potentially mitigating the problem of vanishing gradients  
79 and enhancing the overall efficacy of the distillation process.

Table 2: Class listings for our ImageNet subsets.

Dataset	0	1	2	3	4	5	6	7	8	9
ImageNet-A	Leonberg	Proboscis Monkey	Rapeseed	Three-Toed Sloth	Cliff Dwelling	Yellow Lady's Slipper	Hamster	Gondola	Orca	Limpkin
ImageNet-B	Spoonbill	Website	Lorikeet	Hyena	Earthstar	Trolleybus	Echidna	Pomeranian	Odometer	Ruddy Turnstone
ImageNet-C	Freight Car	Hummingbird	Fireboat	Disk Brake	Bee Eater	Rock Beauty	Lion	European Gallinule	Cabbage Butterfly	Goldfinch
ImageNet-D	Ostrich	Samoyed	Snowbird	Brabancon Griffon	Chickadee	Sorrel	Admiral	Great Gray Owl	Hornbill	Ringlet
ImageNet-E	Spindle	Toucan	Black Swan	King Penguin	Potter's Wheel	Photocopier	Screw	Tarantula	Scilloscope	Lycanid
ImageNette	Tench	English Springer	Cassette Player	Chainsaw	Church	French Horn	Garbage Truck	Gas Pump	Golf Ball	Parachute
ImageWoof	Australian Terrier	Border Terrier	Samoyed	Beagle	Shih-Tzu	English Foxhound	Rhodesian Ridgeback	Dingo	Golden Retriever	English Sheepdog
ImageNet-Birds	Peacock	Flamingo	Macaw	Pelican	King Penguin	Bald Eagle	Toucan	Ostrich	Black Swan	Cockatoo
ImageNet-Fruits	Pineapple	Banana	Strawberry	Orange	Lemon	Pomegranate	Fig	Bell Pepper	Cucumber	Granny Smith Apple
ImageNet-Cats	Tabby Cat	Bengal Cat	Persian Cat	Siamese Cat	Egyptian Cat	Lion	Tiger	Jaguar	Snow Leopard	Lynx

## 80 F Hardware and Software

81 All experiments were run on a workstation equipped with an NVIDIA RTX A6000 GPU (48 GB  
82 VRAM). Our implementation uses PyTorch 1.10.1 with torchvision 0.11.2, and we build upon the  
83 GLaD library for dataset distillation with a generative prior.

## 84 References

- 85 [1] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu. Dataset distillation by matching  
86 training trajectories. In *CVPR*, 2022.
- 87 [2] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu. Generalizing dataset distillation  
88 via deep generative prior. In *CVPR*, pages 3739–3748, 2023.
- 89 [3] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem  
90 solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1998.
- 91 [4] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros. Dataset distillation. *arXiv preprint*, 2018.
- 92 [5] B. Zhao and H. Bilen. Dataset condensation with distribution matching. In *WACV*, 2023.
- 93 [6] B. Zhao, K. R. Mopuri, and H. Bilen. Dataset condensation with gradient matching. *ICLR*, 2020.

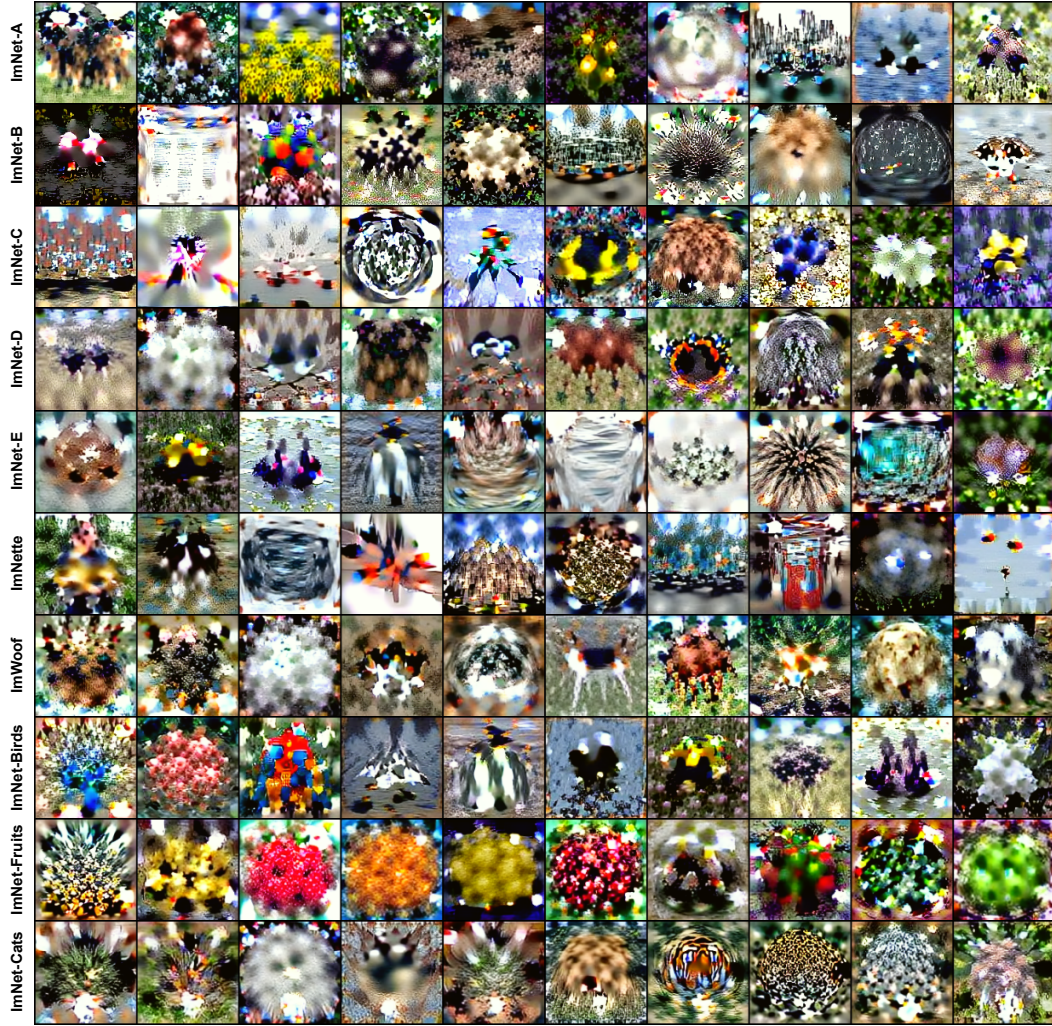


Figure 3: Images distilled by MTT in LD3M for IPC=1.



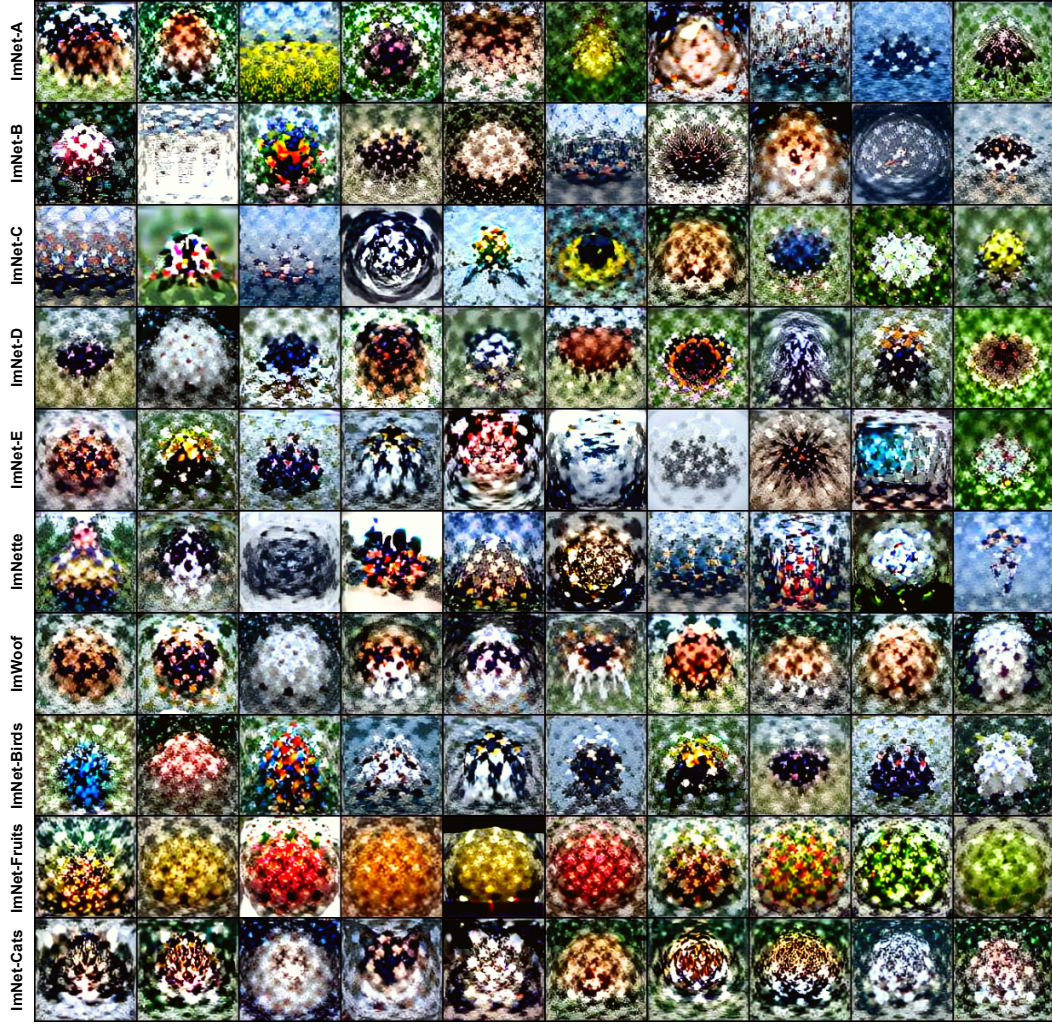


Figure 4: Images distilled by DC in LD3M for IPC=1.



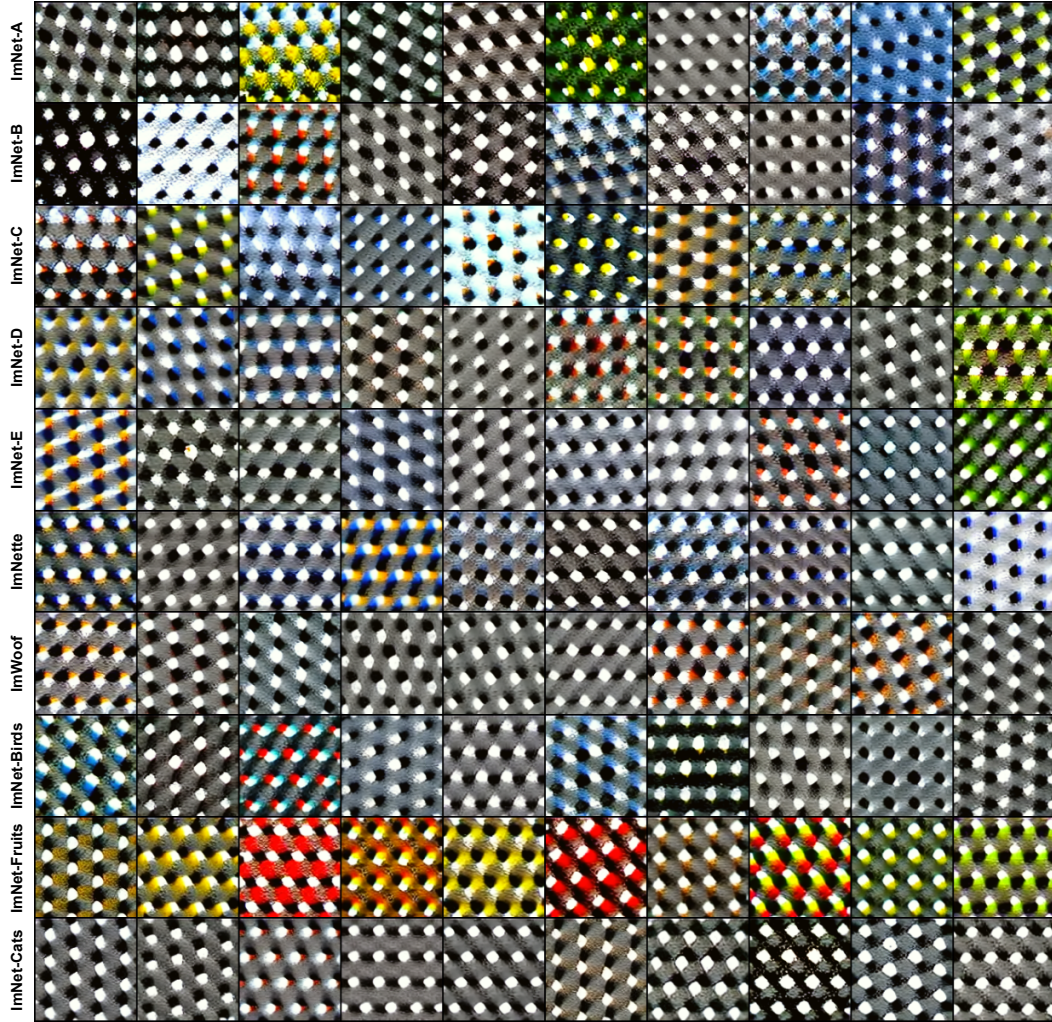


Figure 5: Images distilled by DM in LD3M for IPC=1.





Figure 6: Images distilled by DC in LD3M for IPC=10 and ImageNet-A.





Figure 7: Images distilled by DC in LD3M for IPC=10 and ImageNet-B.





Figure 8: Images distilled by DC in LD3M for IPC=10 and ImageNet-C.





Figure 9: Images distilled by DC in LD3M for IPC=10 and ImageNet-D.



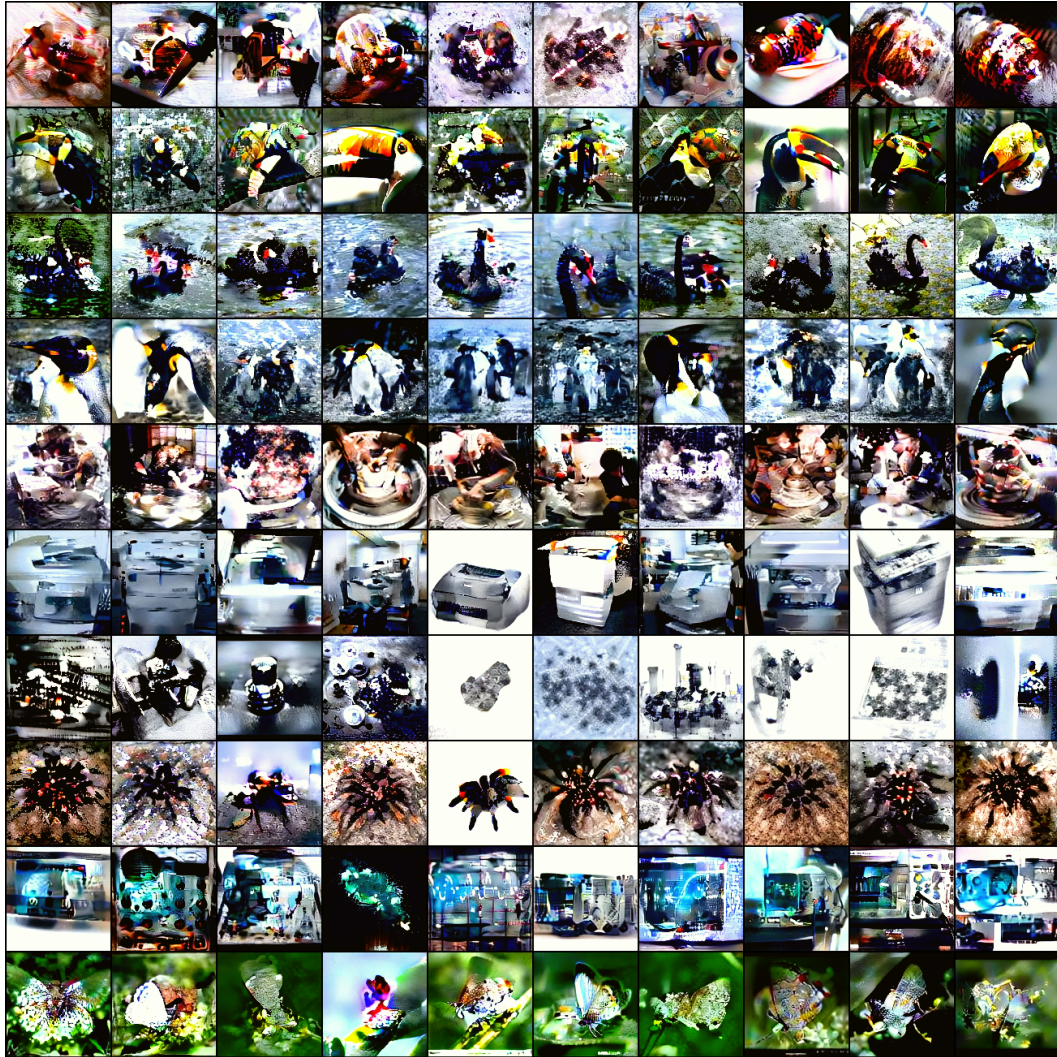


Figure 10: Images distilled by DC in LD3M for IPC=10 and ImageNet-E.



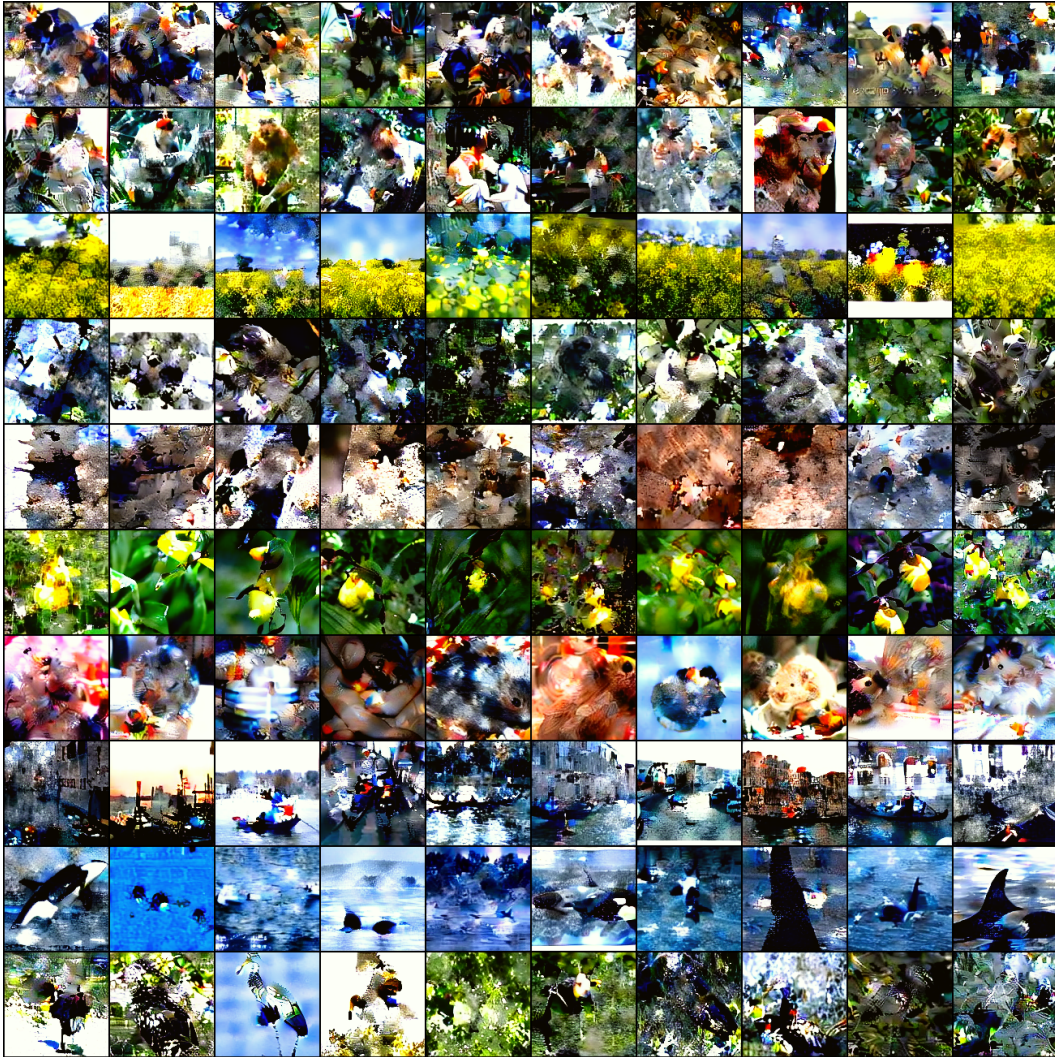


Figure 11: Images distilled by DM in LD3M for IPC=10 and ImageNet-A.



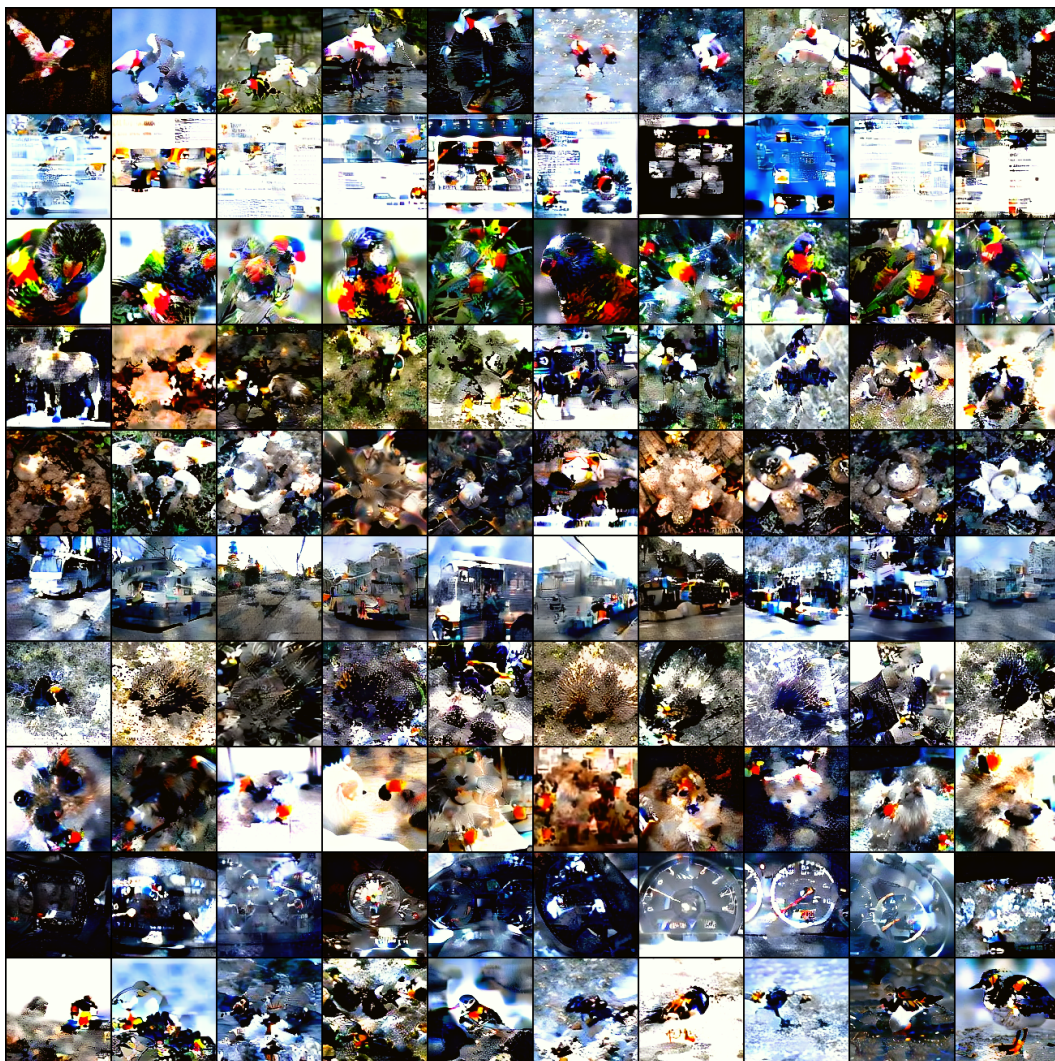


Figure 12: Images distilled by DM in LD3M for IPC=10 and ImageNet-B.



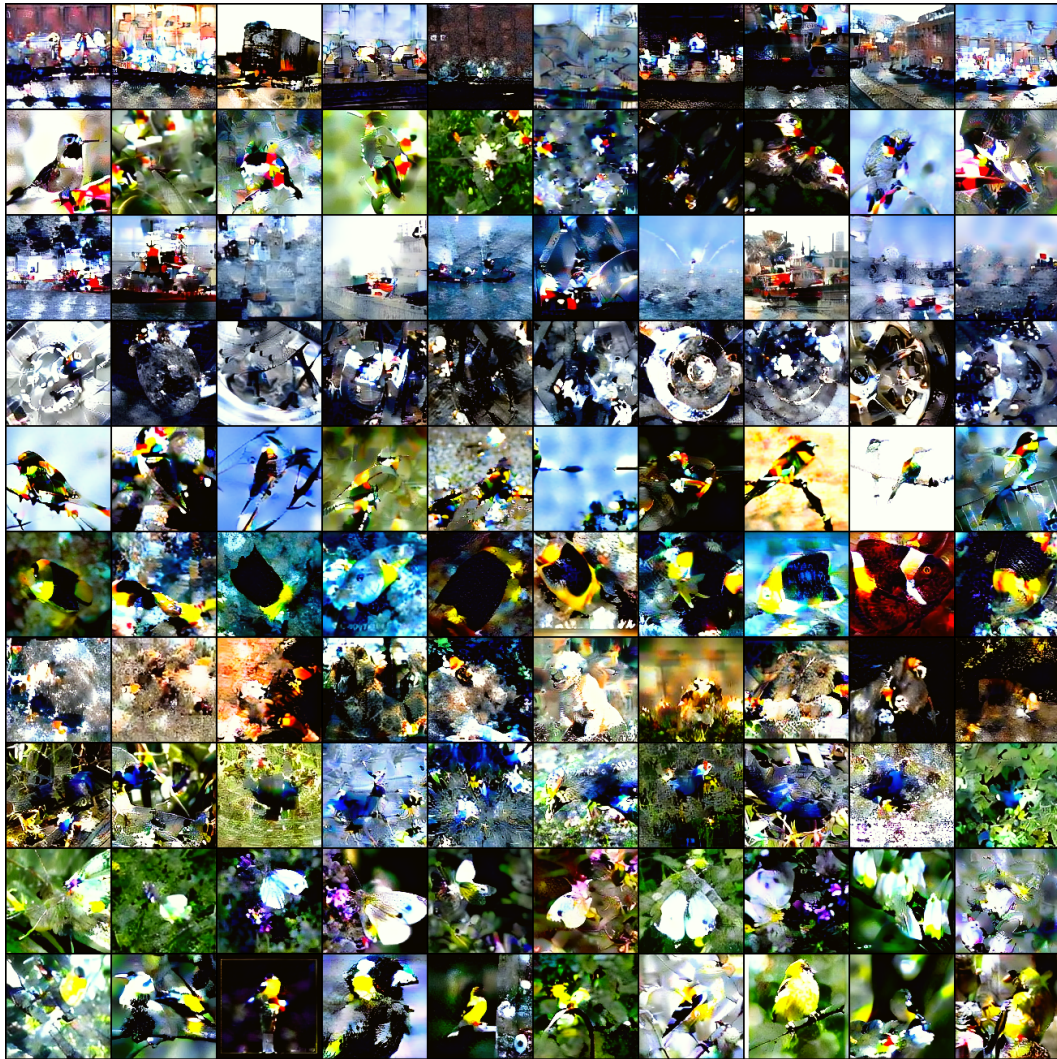


Figure 13: Images distilled by DM in LD3M for IPC=10 and ImageNet-C.



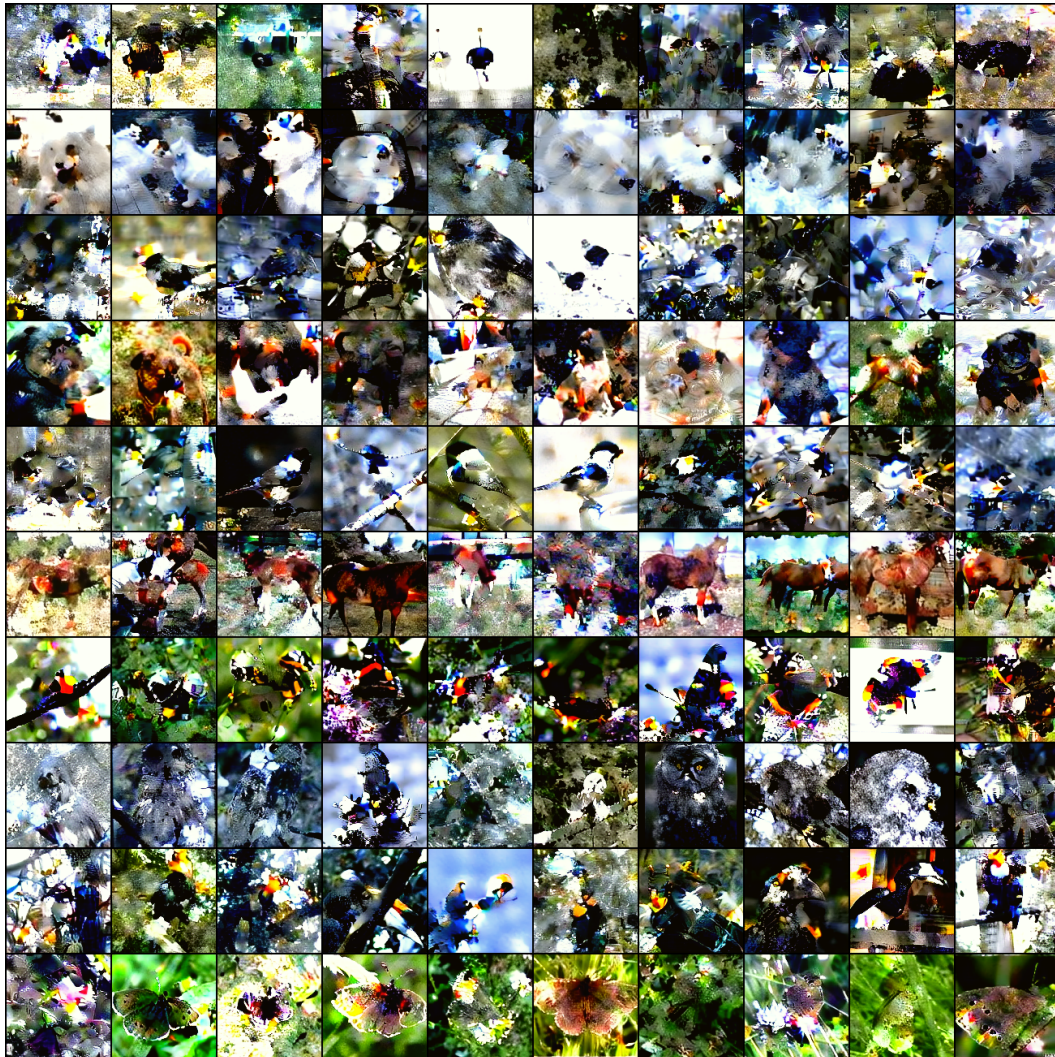


Figure 14: Images distilled by DM in LD3M for IPC=10 and ImageNet-D.