SUPPLEMENTARY MATERIAL

1. ADDITIONAL RELATED WORK

1.1. Revisit of Villian Dataset Distillation

Dataset Distillation (DD) [1] seeks to compress a large training dataset into a small synthetic set that retains similar training utility. Central to most methods is the use of a *surrogate model*, typically a lightweight ConvNet (e.g., depth=3 for CIFAR-10, depth=5 for ImageNet subsets), which provides differentiable feedback during optimization.

Formally, DD can be expressed as finding a synthetic dataset $\mathcal{D}_{\text{syn}}$ such that models trained on it closely mimic those trained on the original dataset $\mathcal{D}$:

$$\min_{\mathcal{D}_{\text{syn}}} \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \left|\ell\big(\Phi_{\theta(\mathcal{D})}(x), y\big) - \ell\big(\Phi_{\theta(\mathcal{D}_{\text{syn}})}(x), y\big)\right| \quad (1)$$

where $\Phi_{\theta(\mathcal{D})}$ and $\Phi_{\theta(\mathcal{D}_{\text{syn}})}$ denote predictors trained on the real and synthetic datasets.

Since solving Eq. (1) directly is infeasible, existing works adopt a *bi-level optimization* framework: the inner loop updates the surrogate on synthetic data, while the outer loop updates synthetic samples so that the surrogate's behavior aligns with training on real data. This paradigm underpins popular strategies such as gradient matching, trajectory matching, and feature alignment, but also introduces structural bias tied to the surrogate architecture, hindering cross-architecture generalization.

1.2. High-Frequency Patterns in Surrogate-Driven Dataset Distillation

Building on the bi-level paradigm, the distilled dataset is optimized to mimic the learning dynamics of a surrogate model. However, prior studies have shown that convolutional networks—commonly adopted as surrogates—derive much of their discriminative power from high-frequency components in the input [2]. Consequently, when synthetic data are updated according to surrogate feedback, they inevitably inherit these high-frequency patterns. While such patterns may align well with the surrogate's inductive bias, they introduce structural bias that hinder generalization to architectures with different spectral preferences (e.g., ViTs, Efficient Net).

This phenomenon motivates a frequency-based diagnosis of distilled datasets. In particular, in the main text, we analyze Fourier spectral energy to quantitatively compare real and distilled data. Our experiments confirm that surrogate-driven distillation consistently amplifies high-frequency energy relative to natural images, aligning with the above intuition of surrogate-induced spectral bias.

1.3. Generative based Method

Generative models have been widely used to improve scalability in dataset distillation, especially for large-scale and high-resolution datasets. GAN-based methods such as IT-GAN [3] optimize latent vectors with condensation and diversity losses to efficiently generate informative samples, while diffusion-based approaches like D4M [4] employ latent diffusion and prototype learning to produce high-quality, high-resolution data. Other strategies, including the decoupling techniques of the SRe2L series [5] and soft label methods [6], further enhance scalability by simplifying optimization and stabilizing training.

Despite their effectiveness, these methods depend on the existing distillation pipeline or the guidance of surrogate models, which limits flexibility. This motivates approaches that operate directly on the synthesized datasets, enabling refinement without altering the underlying distillation process.

Algorithm 1 PReDD: Post-Distillation Refinement via Truncated Reverse Diffusion

Require: Distilled dataset $\mathcal{D}_{\text{syn}} = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^{|\mathcal{D}_{\text{syn}}|}$, pre-trained VAE (E, D), diffusion model ϵ_θ, initial sampling step t_{init}, truncated reverse step K, guidance weight w. $\alpha_{n(t)}$ is the noise scale at scheduler index $n(t)$, $\bar{\alpha}_{n(t)} = \prod_{s=1}^{n(t)} \alpha_s$, $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})$, and $\sigma_{n(t)}$ is the variance from the noise schedule.
Ensure: Refined dataset $\mathcal{D}_{\text{ref}}$

for each $(\tilde{\mathbf{x}}_i, \tilde{y}_i) \in \mathcal{D}_{\text{syn}}$ **do**
  Encode image: $\tilde{\boldsymbol{\zeta}}_0 \leftarrow E(\tilde{\mathbf{x}}_i)$
  Sample intermediate noised latent:
    $\tilde{\boldsymbol{\zeta}}_{t_{\text{init}}} = \sqrt{\bar{\alpha}_{n(t_{\text{init}})}}\,\tilde{\boldsymbol{\zeta}}_0 + \sqrt{1-\bar{\alpha}_{n(t_{\text{init}})}}\,\boldsymbol{\xi}$
  Set $\tilde{\boldsymbol{\zeta}}_K = \tilde{\boldsymbol{\zeta}}_{t_{\text{init}}}, \quad t_{\text{init}} < K < T$
  for $t = K, K-1, \ldots, 1$ **do**
    Compute classifier-free guided denoiser:
      $\epsilon_{\text{CFG}} = \epsilon_\theta(\tilde{\boldsymbol{\zeta}}_t, n(t), \varnothing)$
        $+ w \cdot \big(\epsilon_\theta(\tilde{\boldsymbol{\zeta}}_t, n(t), \tilde{y}_i) - \epsilon_\theta(\tilde{\boldsymbol{\zeta}}_t, n(t), \varnothing)\big)$
    Update latent (truncated reverse step):
      $\tilde{\boldsymbol{\zeta}}_{t-1} = \frac{1}{\sqrt{\alpha_{n(t)}}}\Big(\tilde{\boldsymbol{\zeta}}_t - (1-\alpha_{n(t)})\epsilon_{\text{CFG}}\Big) + \sigma_{n(t)}\boldsymbol{\xi}$
  end for
  Refined latent: $\tilde{\boldsymbol{\zeta}}_0^{\text{ref}} \leftarrow \tilde{\boldsymbol{\zeta}}_0$
  Decode: $\hat{\mathbf{x}}_i \leftarrow D(\tilde{\boldsymbol{\zeta}}_0^{\text{ref}})$
  Add $(\hat{\mathbf{x}}_i, \tilde{y}_i)$ to $\mathcal{D}_{\text{ref}}$
end for
return $\mathcal{D}_{\text{ref}}$

2. IMPLEMENTATION DETAILS

2.1. Evaluations

Our experiments focus on high-resolution datasets. Specifically, we use subsets of ImageNet, including ImageNet-Fruits, ImageNet-Squawk, and others. All images generated by the diffusion model are at a resolution of 256×256, but for evaluation, we resize them to 128×128 to ensure a consistent evaluation standard and facilitate comparison with conventional distillation methods. Unlike approaches based purely on generative models, we focus on improving the cross-architecture performance of synthetic datasets, and therefore our baselines are refined from surrogate-based distillation

Table 1: ImageNet Subsets performance on the surrogate model (ConvNet-D5) of DM and MTT

Method	IPC = 1						IPC = 10					
	Fruits	Meow	Nette	Squawk	Woof	Yellow	Fruits	Meow	Nette	Squawk	Woof	Yellow
DM [7]	19.5±1.0	18.3±1.7	28.9±1.5	27.3±0.8	20.2±0.6	32.9±1.6	26.2±0.5	23.2±0.8	40.0±1.1	32.7±1.2	24.0±0.6	41.5±0.6
Ours	**11.8**±0.6	**15.0**±1.2	**18.9**±1.5	**18.0**±0.8	**17.4**±1.8	**23.5**±1.2	**19.4**±0.9	**22.5**±1.1	**37.1**±2.2	**30.1**±1.0	**20.9**±1.3	**31.2**±1.1
MTT [8]	24.2±0.7	27.8±1.1	44.6±0.5	12.0±1.6	27.3±1.0	40.7±2.9	35.3±1.4	37.1±1.3	58.0±1.5	49.9±1.8	32.9±0.8	53.8±0.8
Ours	**21.8**±2.3	**20.6**±0.8	**33.9**±1.9	**10.6**±1.0	**23.5**±1.1	**31.5**±1.2	**30.6**±1.4	**30.7**±1.2	**53.4**±2.4	**42.3**±1.1	**28.4**±2.0	**46.1**±1.6

methods. All experiments can be conducted on a single GPU, such as an NVIDIA RTX 4090 or A800.

2.2. Diffusion Model Setup

Our post-distillation refinement framework, PReDD, uses a pre-trained DiT backbone, a Transformer-based model designed for feature extraction and image generation within diffusion models. To refine distilled datasets, we adopt the DDPM (Denoising Diffusion Probabilistic Model) strategy. In practice, we implement this using SpacedDiffusion, which accelerates the reverse diffusion process compared to standard Gaussian diffusion by selectively skipping intermediate diffusion steps while maintaining high-quality image reconstruction. Here, the timesteps mentioned in the main text refer to the continuous sampling steps, while the corresponding noise levels follow the spaced schedule.

A key aspect of our approach is operating in the latent space: distilled images are first encoded into a lower-dimensional latent representation using a pre-trained VAE (`sd-vae-ft-mse` [9]). This latent encoding reduces high-frequency patterns inherited from surrogate-driven distillation. The refinement is then performed via truncated reverse diffusion with classifier-free guidance (CFG), which iteratively restores semantic details. This design simultaneously suppresses excessive high-frequency noise and enhances class-specific semantics in the refined images.

2.3. Performance Analysis on Surrogate Model

As shown in Table 1, our method results in a noticeable performance drop when evaluated on the original surrogate model. This degradation arises because our refinement suppresses the high-frequency patterns that are typically exploited by convolutional surrogates during distillation. Importantly, this provides indirect evidence for our claim in the main text: the high-frequency patterns preserved in conventional distilled datasets are closely tied to the inductive bias of specific surrogate architectures. By mitigating these patterns, our approach improves the cross-architecture generalization of distilled datasets, even though it sacrifices performance on the surrogate itself.

2.4. Noise Injection and Reverse Steps

Before performing truncated reverse diffusion, PReDD applies controlled noise injection to the latent representations of the distilled dataset. Specifically, Gaussian noise corresponding to an initial timestep t_{init} is added to each latent image. This step aligns the latent encoding with the diffusion prior, ensuring that the subsequent reverse diffusion proceeds stably and produces semantically coherent refinements.

As discussed in our analysis of the reverse diffusion steps K, datasets distilled under different configurations may necessitate distinct t_{init} and K values for optimal refinement. To facilitate reproducibility, Table 2 provides the selected t_{init} and K for each distillation method across different IPC.

Table 2: Selected initial timesteps (t_{init}) and reverse diffusion steps (K) for different distillation methods and IPC settings.

Distillation Method	**IPC=1** (t_{init}/K)	**IPC=10** (t_{init}/K)
DM [7]	10 / 15	20 / 20
NCFM [10]	25 / 30	30 / 35
MTT [8]	10 / 10	10 / 10
EDF [11]	10 / 10	10 / 15

3. REFERENCES

[1] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *arXiv preprint arXiv:1811.10959*, 2018.

[2] H. Wang, X. Wu, Z. Huang, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8684–8694.

[3] B. Zhao and H. Bilen, "Synthesizing informative training samples with gan," *arXiv preprint arXiv:2204.07513*, 2022.

[4] D. Su, J. Hou, W. Gao, Y. Tian, and B. Tang, "Dˆ 4: Dataset distillation via disentangled diffusion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5809–5818.

[5] Z. Yin, E. Xing, and Z. Shen, "Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective," *Advances in Neural Information Processing Systems*, vol. 36, pp. 73 582–73 603, 2023.

[6] T. Qin, Z. Deng, and D. Alvarez-Melis, "A label is worth a thousand images in dataset distillation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 131 946–131 971, 2024.

[7] B. Zhao and H. Bilen, "Dataset condensation with distribution matching," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 6514–6523.

[8] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu, "Dataset distillation by matching training trajectories," in *CVPR*, 2022, pp. 4750–4759.

[9] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *ICCV*, 2023, pp. 4195–4205.

[10] S. Wang, Y. Yang, Z. Liu, C. Sun, X. Hu, C. He, and L. Zhang, "Dataset distillation with neural characteristic function: A minmax perspective," in *CVPR*, 2025, pp. 25 570–25 580.

[11] K. Wang, Z. Li, Z.-Q. Cheng, S. Khaki, A. Sajedi, R. Vedantam, K. N. Plataniotis, A. Hauptmann, and Y. You, "Emphasizing discriminative features for dataset distillation in complex scenarios," in *CVPR*, 2025, pp. 30 451–30 461.

Fig. 1: ImageNet-Fruits. Top: DM. Bottom: Refined (ours).

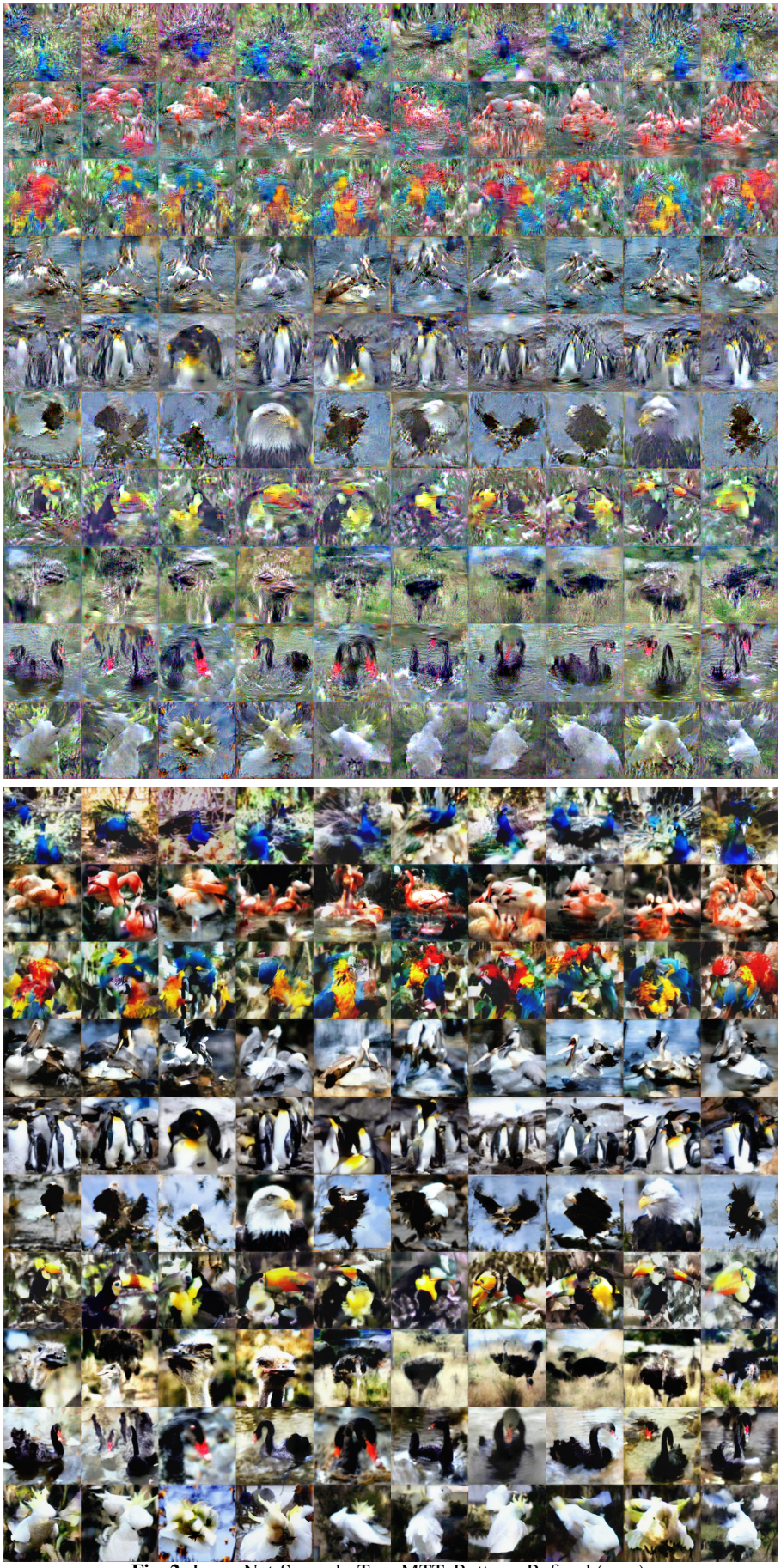

Fig. 2: ImageNet-Squawk. Top: MTT. Bottom: Refined (ours).

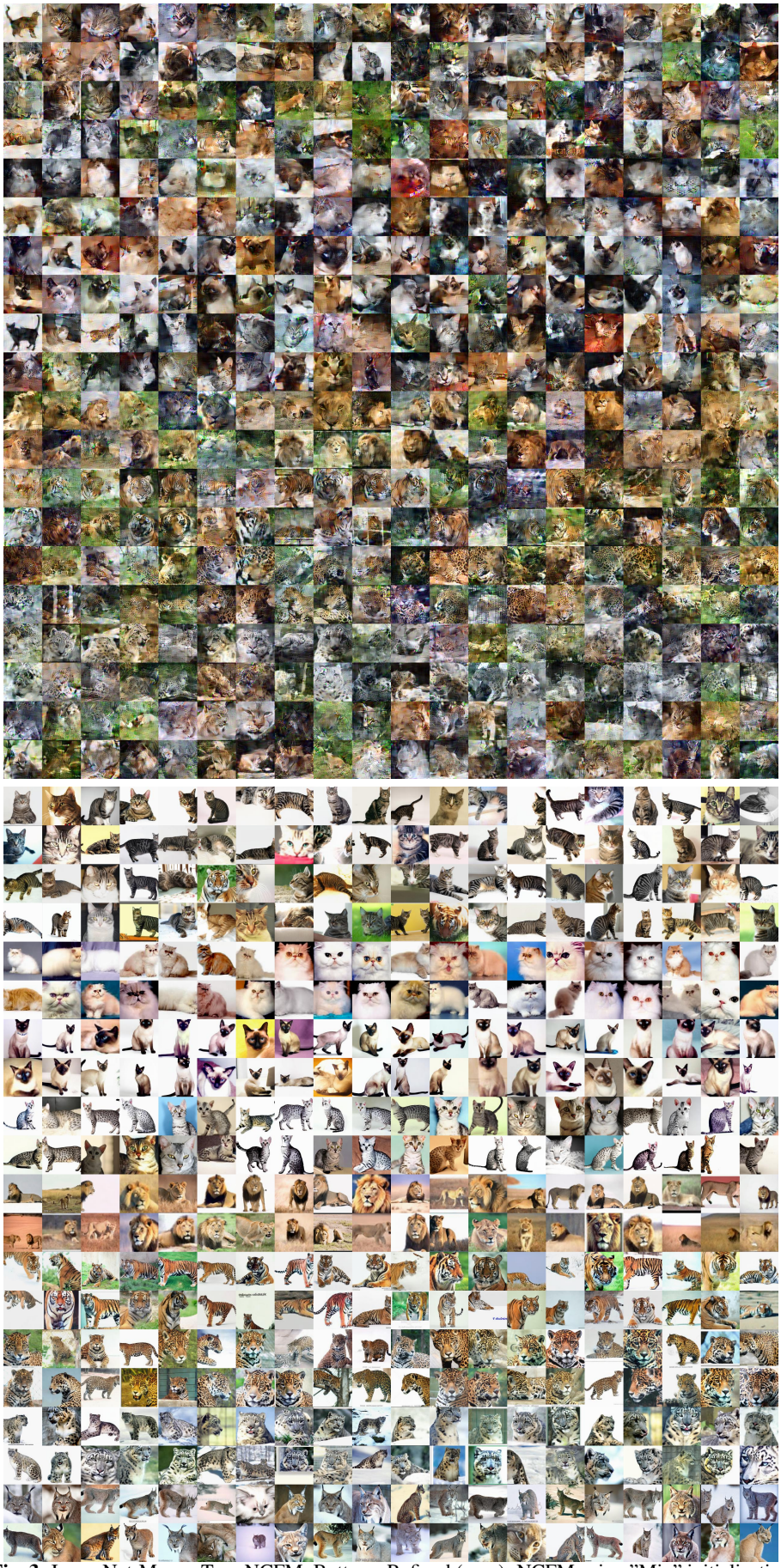

Fig. 3: ImageNet-Meow. Top: NCFM. Bottom: Refined (ours). NCFM using ”Mix” initialization