Temporal Prototype Alignment for Frozen-Feature Dataset Distillation

Rui Ding[1] **Jing Hu**[1] **Mei Chen**[1] **Xi Wu**[1]
Hualin Zhou[3] **Fan Wu**[2] **Kehua Guo**[2*] **Tao Gu**[4]

[1]Chengdu University of Information Technology [2]Central South University
[3]Macquarie University [4]Shanghai Jiao Tong University

Abstract

Dataset distillation condenses large-scale training sets into compact synthetic data. Recent frozen-feature distillation methods, such as Linear Gradient Matching, make ImageNet-scale synthesis feasible by matching gradients in pretrained representation spaces, but their reliance on exhaustive spatial view ensembles incurs memory and I/O costs that grow linearly with the number of augmentations. We propose Temporal Prototype Alignment (TPA), a temporal gradient estimation framework for frozen-feature dataset distillation. TPA replaces spatial Monte Carlo gradient averaging with a temporal prototype estimator that accumulates stable semantic signals across optimization steps. An adaptive momentum gate modulates the estimator according to local signal reliability, reducing gradient variance while keeping memory complexity independent of the number of spatial views. To improve cross-architecture transfer, we introduce a consensus alignment objective that distills shared structure from multiple pretrained teachers while mitigating model-specific artifacts. Across ImageNet-scale and fine-grained benchmarks, TPA substantially reduces memory overhead, enables distillation on consumer-grade GPUs, and improves transfer to heterogeneous CNN and Vision Transformer backbones. These results suggest that online temporal estimation provides an efficient alternative to spatial ensembling for frozen-feature dataset distillation.
Project page: https://ruiding1.github.io/TPA-project/

1 Introduction

Dataset distillation synthesizes a compact labeled dataset such that a model trained on the distilled data achieves performance comparable to training on the full dataset [1]. Traditional research has primarily focused on the training-from-scratch paradigm, in which synthetic data is optimized to supervise randomly initialized networks [2–6]. Meanwhile, modern vision pipelines increasingly favor linear probing over fixed, pretrained representations. This paradigm shift motivates dataset distillation methods that operate directly in frozen feature space and scale to large backbones.

A recent breakthrough, Linear Gradient Matching (LGM) [7] enabled ImageNet-scale distillation by transitioning from training-from-scratch to gradient alignment within frozen feature spaces. Despite its impressive performance at one image per class (IPC=1), LGM relies on

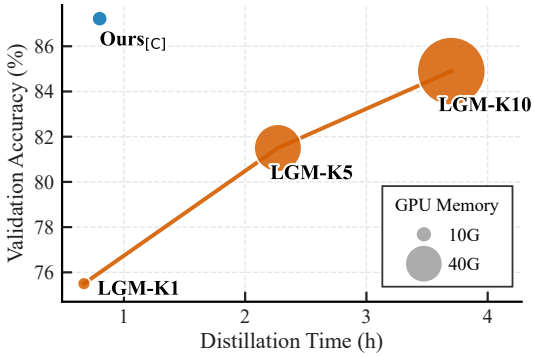


Figure 1: **Efficiency and Performance Comparison (IPC=1).** Our method achieves superior accuracy compared to LGM with substantially reduced training costs. Specifically, Ours[C] matches LGM with a 10× memory reduction. This efficiency enables Ours[C+D+M] ([C]: CLIP, [D]: DINOv2, [M]: MoCo-v3) to leverage cross-model consensus, achieving new SOTA results.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).
*Corresponding author.

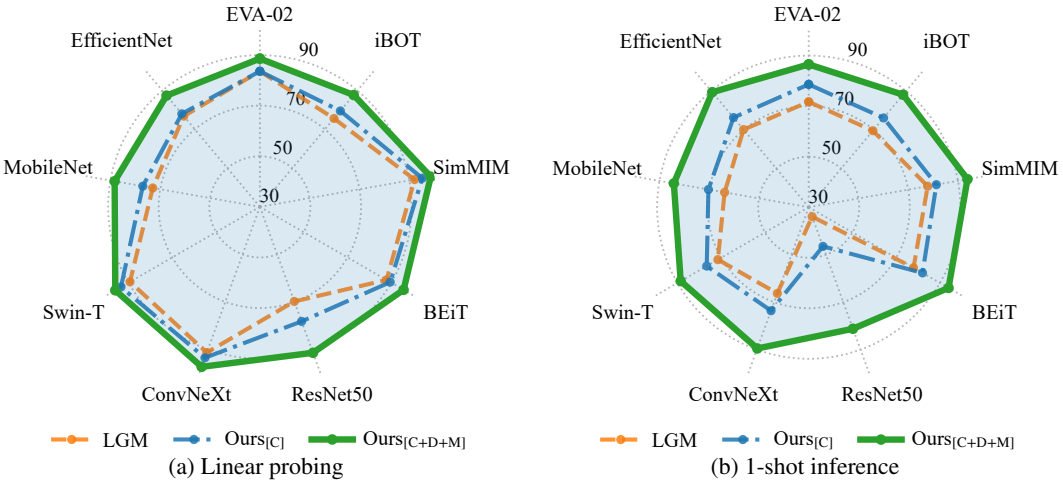


Figure 2: **Cross-architecture generalization and black-box transferability.** Ours$_{[C+D+M]}$ consistently improves transfer across 9 unseen heterogeneous target models under both linear probing and training-free 1-shot inference.

differentiable augmentation ensembles, typically using K=10 views per step. These multi-view augmentations stabilize gradient estimates and enforce cross-view consistency, thereby preventing view-specific shortcuts. However, this strategy incurs a linear surge in memory and I/O overhead, reaching 81 GB on ImageNet-100. Scaling to ImageNet-1K further requires enterprise-grade resources, such as 4×H200 GPUs for over 12 hours. More critically, this memory bottleneck restricts existing methods to a single-teacher setting, thereby limiting their ability to generalize across architectures. Because any individual model inevitably encodes architecture-specific inductive biases, synthetic data distilled from a single teacher remain constrained by these biases, irrespective of augmentation intensity. This limitation fundamentally challenges the scalability and transferability of current frozen-feature distillation paradigms.

In this paper, we challenge the necessity of computationally expensive spatial ensembling strategies. Our key observation is that, although gradients computed from single augmented views are inherently stochastic and view-dependent, their temporal trajectories preserve stable semantic signals under diverse augmentations. By shifting from spatial ensembling to temporal integration, we reconstruct robust distillation prototypes from historical gradient information, thereby eliminating the substantial computational and memory overhead associated with multi-view processing. Moreover, the resulting efficiency enables distillation across heterogeneous models, aligning with the Platonic Representation Hypothesis [8], which posits that diverse large-scale models converge toward shared representations.

Based on this insight, we introduce Adaptive Temporal Momentum, a gating mechanism that modulates gradient accumulation based on the local signal-to-noise ratio. This mechanism dynamically suppresses high-variance gradient outliers while rapidly adapting to consistent semantic signals, thus distilling stable prototypes from stochastic updates. To ensure that distilled data encode model-invariant knowledge, we propose Contrastive Consensus Alignment. Rather than matching gradients within a single architecture, we cast dataset distillation as a consensus-seeking process across heterogeneous backbones using a variational mutual information objective. This formulation ensures the synthesized images encapsulate universal patterns rather than model-specific biases.

The results are striking. As shown in Figure 1, our method distilled from CLIP achieves superior accuracy on ImageNet-100 using only 7 GB of GPU memory on an RTX 4090, compared to 81 GB required by LGM. This achievement represents a 10× memory reduction and 4× acceleration. Figures 2a and 2b confirm the consistent superiority of the proposed consensus framework over existing baselines in both linear probing and 1-shot inference across nine heterogeneous architectures, spanning both CNNs and ViTs. The main contributions are as follows:

- We propose a paradigm shift from spatial ensembling to temporal integration via a gating mechanism that reconstructs stable distillation prototypes from single-view gradients, substantially reducing memory and computational overhead.
- We propose contrastive consensus alignment to distill architecture-agnostic consensus knowledge from heterogeneous foundation models, significantly enhancing the cross-model transferability of the distilled images.

- Our method achieves new SOTA results on ImageNet-scale benchmarks under both linear probing and 1-shot inference, achieving a $4\times$ speedup and a $10\times$ memory reduction on a single consumer RTX 4090.

2 Related Work

Dataset Distillation. Dataset Distillation emerges as a promising solution by condensing large datasets into compact synthetic sets that retain the training efficacy [9]. Early work formulated distillation as a bi-level optimization problem, where the synthetic set is directly optimized by backpropagating through multiple inner-loop training steps [10–16]. Subsequent methods replaced long unrolled training with more tractable objectives, such as *gradient matching* [3, 17–20] for aligning update directions, and *trajectory matching* [21–23] for matching multi-step optimization trajectories. But, these approaches often struggle to scale due to the memory and instability inherent to bi-level optimization. Recent efforts have pushed condensation toward ImageNet scale, such as exact gradient computation for ImageNet-1K [24–27] and multi-stage large-scale condensation pipelines [28, 29]. Crucially, most existing literature assumes *training from scratch* on randomly initialized networks [30–33]. In contrast, LGM pioneers ImageNet-scale distillation within the frozen feature space. However, LGM relies on augmentation ensembles to stabilize gradients and enforce multi-view consistency, causing linear growth in GPU memory and I/O.

Our work shows that such expensive concurrent ensembles are unnecessary. Instead, we exploit *temporal integration* over stochastic single-view gradients to reconstruct stable distillation prototypes, while further promoting architecture-agnostic transfer via cross-backbone consensus

Vision Foundation Models. Self-supervised learning has become the dominant paradigm for learning transferable visual representations from unlabeled data. Major families include contrastive learning [34–37], self-distillation approaches such as DINO [38], and masked reconstruction methods exemplified by MAE [39–41]. In parallel, self-supervised learning has been successfully scaled to Vision Transformers, e.g., MoCo-v3 [42], while large-scale pre-training methods such as DINOv2 yield general-purpose visual features without supervision [43]. Additionally, vision-language models like CLIP further enrich representations through natural-language supervision, demonstrating strong transfer and zero-shot capabilities [44–46]. Under linear evaluation protocols, these diverse pre-training paradigms often produce similarly useful representations. This aligns with the *Platonic Representation Hypothesis* [8], which posits that as models and data scale, learned representations tend to converge toward a shared statistical structure across architectures and even modalities [47].

This observation motivates our design: data distilled from a single frozen backbone may inherit model-specific biases, whereas enforcing alignment across heterogeneous foundation models can distill model-invariant semantics. Thus, we formulate distillation as a cross-backbone consensus process, achieving generalization beyond any single backbone.

3 Method

3.1 Problem Formulation and Preliminaries

Problem Setup. Given a full real dataset $\mathcal{D} = \{(x^i_{\text{real}}, y^i_{\text{real}})\}_{i=1}^{|\mathcal{D}|}$, dataset distillation aims to synthesize a much smaller labeled dataset $\mathcal{S} = \{(x^i_{\text{syn}}, y^i_{\text{syn}})\}_{i=1}^{|\mathcal{S}|}$, where $|\mathcal{S}| \ll |\mathcal{D}|$, while preserving the essential knowledge of $\mathcal{D}$. Under the dominance of pre-trained vision foundation models, we follow the standard linear probing protocol: the feature extractor ϕ remains frozen to preserve the pre-trained representations, and the training process focuses on learning a linear classifier $W \in \mathbb{R}^{C\times F}$, where C denotes the number of classes and F is the feature dimension.

Revisiting Linear Gradient Matching. To obtain the discriminative information of $\mathcal{D}$ into $\mathcal{S}$, LGM aligns the gradients of the linear classifier derived from both datasets. At each distillation iteration, a linear classifier is randomly sampled $W \sim \mathcal{N}(0, 1)$, and the objective is formulated to minimize the cosine distance between the gradients:

$$\mathcal{L}_{\text{meta}} = 1 - \cos\left(\text{vec}(\nabla_W \ell_{\text{real}}), \text{vec}(\nabla_W \ell_{\text{syn}})\right), \tag{1}$$

where ℓ_{real} and ℓ_{syn} denote the cross-entropy losses evaluated on the real and synthetic datasets, respectively. To obtain a reliable estimate of the gradient direction under stochastic data augmentations,

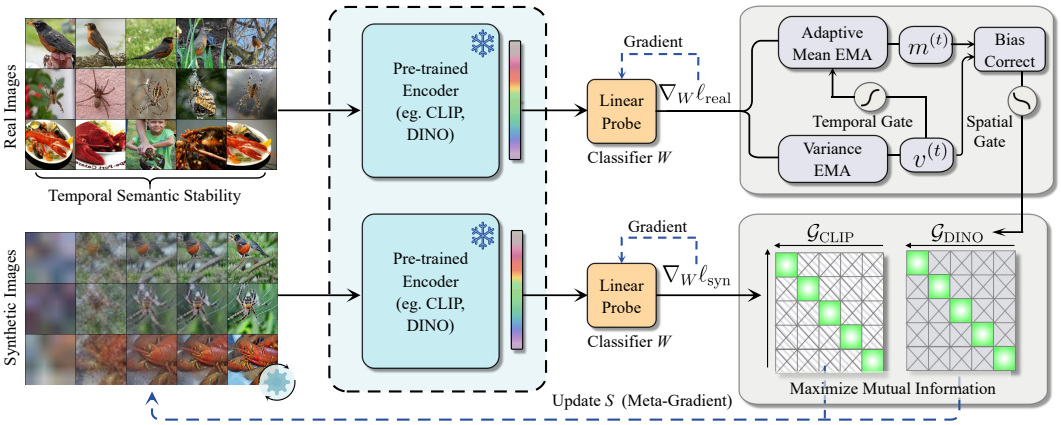


Figure 3: **Overview of Temporal Prototype Alignment.** (i): To reconstruct stable prototypes $\mathcal{G}$, an Adaptive Temporal Momentum module decouples semantic signals and imposes implicit multi-view regularization, using a temporal gate and spatial gate. (ii): $\mathcal{G}$ guides the synthesis of images. To ensure architecture-agnostic transferability, we perform Contrastive Consensus Alignment, maximizing the mutual information between $\nabla_{W_m} \ell_{\text{syn}}$ and $\mathcal{G}_m$ across heterogeneous backbones (e.g., CLIP, DINO) via a variational InfoNCE objective.

LGM employs a spatial Monte Carlo approximation by aggregating gradients across an ensemble of K concurrent views. Specifically, the synthetic loss is computed as:

$$\ell_{\text{syn}} = \frac{1}{K} \sum_{k=1}^{K} \text{CE}\left(W \phi(A_k(x_{\text{syn}})), y_{\text{syn}}\right), \tag{2}$$

with an analogous formulation for ℓ_{real}, where $A_k(\cdot)$ denotes differentiable data augmentations (e.g., RandomResizedCrop). In the IPC=1 regime, the multi-view objective is used to reduce gradient stochasticity and enforce consistency, thereby preventing view-specific shortcuts.

Limitations. Despite its effectiveness, LGM suffers from severe computational overhead. Processing K views (typically K=10) per iteration inflates the batch size to $2 \times C \times K$, leading to a linear growth in memory and I/O overhead. For instance, distilling ImageNet-1K requires **12 hours of training on 4×H200 GPUs**. Crucially, naively reducing K is not a viable solution, as it both introduces gradient stochasticity and weakens the multi-view consistency constraint.

These limitations motivate a paradigm shift from costly spatial averaging to efficient temporal integration. Figure 3 shows that our approach addresses these challenges through two synergistic modules: (i) Adaptive Temporal Momentum that stabilizes stochastic single-view gradients into robust distillation prototypes, and (ii) Contrastive Consensus Alignment that distills architecture-agnostic patterns by maximizing mutual information across heterogeneous models.

3.2 Adaptive Temporal Momentum

To circumvent the bottleneck of LGM, we reduce the augmentation ensemble size from K=10 to K=1. Thus, $\nabla_W \ell_{\text{real}}$ becomes highly stochastic and view-dependent. Departing from the traditional use of gradient history for acceleration, we reinterpret this temporal sequence as a mechanism to enforce multi-view consistency over time.

Specifically, we aim to reconstruct a stable distillation prototype $\mathcal{G}$ that captures the invariant semantic signal shared across random augmentations. This formulation entails two core challenges: (i) *Temporal gating:* the update trajectory must be gated by signal quality, as blind accumulation risks embedding outliers into the prototype. (ii) *Feature selectivity*: the filter must explicitly distinguish stable semantic signals from transient artifacts, suppressing feature dimensions dominated by high-frequency noise.

To this end, we propose Adaptive Temporal Momentum, a gating mechanism that explicitly decouples semantic signal from stochastic noise. Let m and v in $\mathbb{R}^{C \times F}$ denote the first- and second-order moment estimates of $\nabla_W \ell_{\text{real}}$. We modulate the accumulation of these moments via a soft temporal gate $\mu \in \mathbb{R}^{C \times F}$. At step t, the moments $m^{(t)}$ and $v^{(t)}$ update as

$$m^{(t)} = \mu^{(t)} \odot m^{(t-1)} + \left(1 - \mu^{(t)}\right) \odot \nabla_W \ell_{\text{real}}^{(t)}, \tag{3}$$

and

$$v^{(t)} = \beta v^{(t-1)} + (1-\beta)\left(\nabla_W \ell_{\text{real}}^{(t)}\right)^2, \tag{4}$$

where $\odot$ denotes the element-wise product and β is a fixed variance decay rate. Bounded by $\mu_{\min}$ and $\mu_{\max}$, the temporal gate $\mu^{(t)}$ is modulated by the local Signal-to-Noise Ratio rather than the static decay used in standard EMA:

$$\mu^{(t)} = \mu_{\min} + (\mu_{\max} - \mu_{\min}) \tanh\left(\frac{\sqrt{v^{(t)}}}{|m^{(t-1)}| + \epsilon}\right), \tag{5}$$

where ϵ is a small constant, and $\sqrt{v^{(t)}}/|m^{(t-1)}|$ approximates the inverse SNR. This operates as a Kalman-like Gain, which attenuates updates when transient noise dominates, while accelerating adaptation to consistent semantic trends.

To eliminate the initialization bias inherent to exponential moving averages, we compute the bias-corrected moment estimates:

$$\hat{m}^{(t)} = \frac{m^{(t)}}{1 - \prod_{i=1}^{t} \mu^{(i)}}, \quad \hat{v}^{(t)} = \frac{v^{(t)}}{1 - \beta^t}. \tag{6}$$

Subsequently, denoting the total optimization steps as T, we derive the distillation prototype trajectory $\mathcal{G} = \{\mathcal{G}^{(t)}\}_{t=1}^{T}$ via adaptive inverse-variance reweighting:

$$\mathcal{G}^{(t)} = \left(\frac{1}{\sqrt{\hat{v}^{(t)}} + \epsilon}\right) \odot \hat{m}^{(t)}, \tag{7}$$

where the element-wise factor acts as an adaptive spatial gate that suppresses noise-dominated dimensions while amplifying features with stable semantic consistency. Unlike conventional momentum that controls *how fast* optimization proceeds, this second-order moment determines *what* to learn by selectively filtering semantically reliable dimensions. Figure 4 reports the cosine similarity between consecutive gradient steps: real gradients with $K = 1$ fluctuate sharply under single-view noise, whereas our temporal integration reconstructs a stable distillation prototype that guides synthetic data toward consistent features.

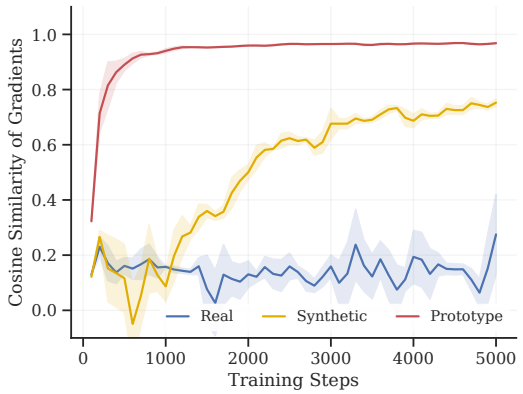


Figure 4: **Gradient Angular Stability Analysis.** Real gradients exhibit high stochasticity with similarity scores near 0.1. In contrast, our prototype maintains high temporal stability, providing a robust signal for synthesis.

3.3 Contrastive Consensus Alignment

Robust distilled data should capture invariant knowledge transferable across heterogeneous models. Hence, we propose distilling consensus knowledge from an ensemble of M diverse pre-trained backbones $\{\phi_m\}_{m=1}^{M}$, where each backbone features a dimension F_m and a corresponding model-specific linear head $W_m \in \mathbb{R}^{C \times F_m}$. Instead of relying on geometric matching, we frame the distillation process from an information-theoretic perspective.

Formally, we formulate this as maximizing the mutual information $\mathcal{I}(\cdot;\cdot)$ between the synthetic gradients of ϕ_m and its distillation prototypes $\mathcal{G}_m$ across heterogeneous feature spaces. We compute the objective with respect to W_m:

$$x_{\text{syn}}^{*} = \arg\max_{x_{\text{syn}}} \sum_{m=1}^{M} \mathcal{I}\left([\nabla_{W_m} \ell_{\text{syn}}]_y; \mathcal{G}_{m,y}\right), \tag{8}$$

where x_{syn}^{*} serves as a shared information bottleneck. Since x_{syn} is the only learnable parameter, it is driven by joint mutual information maximization to converge upon the semantic intersection of diverse architectural frameworks. Since directly optimizing mutual information is computationally intractable, we follow established works [48–50] to maximize a variational lower bound. Specifically, the mutual information is bounded by the InfoNCE loss as follows:

$$\sum_{m=1}^{M} \mathcal{I}\left([\nabla_{W_m} \ell_{\text{syn}}]_y; \mathcal{G}_{m,y}\right) \geq M \log(C) - \sum_{m=1}^{M} \mathcal{L}_{\text{NCE}}^{m}. \tag{9}$$

Table 1: **Linear Probing Accuracy (%) on ImageNet-Scale Datasets with 1 Image Per Class.** Notably, our single-model approach (i.e., Ours$_{[C/D/M]}$) achieves comparable performance to LGM while significantly reducing compute costs. MGD^3 and CaO2 are DiT-based generative dataset distillation baselines.

Train Set (1 Img/Cls)	ImageNet-100				ImageNet-1K			
	CLIP	DINOv2	MoCo-v3	Average	CLIP	DINOv2	MoCo-v3	Average
Full Dataset	92.5±0.0	95.2±0.1	89.4±0.3	92.4±0.2	78.7±0.0	83.0±0.0	76.5±0.0	79.4±0.0
Neighbors	67.8±0.3	86.0±0.2	77.1±0.1	76.9±0.2	38.8±0.1	67.7±0.1	56.4±0.0	54.3±0.1
Centroids	77.1±0.1	86.9±0.3	77.7±0.1	80.6±0.2	53.9±0.0	69.5±0.1	57.4±0.0	60.3±0.1
Random	56.6±1.6	74.8±2.8	61.4±2.6	64.3±2.4	31.7±0.5	50.3±0.5	38.8±0.6	40.3±0.5
MGD^3 [55]	68.5±0.2	82.1±0.1	72.1±0.2	74.2±0.2	45.4±0.1	62.4±0.1	50.9±0.1	52.9±0.1
CaO2 [56]	68.7±0.2	83.0±0.1	75.0±0.2	75.6±0.2	47.7±0.1	65.1±0.1	53.6±0.1	55.4±0.1
LGM [7]	84.9±0.1	91.5±0.1	83.4±0.1	86.6±0.1	63.0±0.0	75.0±0.1	63.2±0.0	67.1±0.0
Ours$_{[C/D/M]}$	86.7±0.2	92.2±0.1	85.2±0.1	88.0±0.1	63.8±0.1	75.6±0.1	63.6±0.1	67.7±0.1
Ours$_{[C+D+M]}$	**87.4**±0.1	**92.4**±0.1	**86.5**±0.1	**88.8**±0.1	**64.8**±0.1	**76.5**±0.1	**64.3**±0.2	**68.5**±0.1

Table 2: **Cross-Architecture Linear Probing (%) and Memory Overhead on ImageNet-100.** Our method achieves superior generalization across 9 unseen architectures with 10x memory reduction compared to LGM.

Distill Model	GPU Mem.	Self-Supervised Models				Supervised Models					Average
		EVA-02	iBOT	SimMIM	BEiT	ResNet50	ConvNext	Swin-T	MobileV3	EffNet	
LGM trained on the 4 × RTX 4090											
C	81G	83.6±0.3	75.6±0.0	92.1±0.1	87.8±0.1	69.8±0.4	91.3±0.3	89.3±0.4	73.0±0.0	76.8±0.2	82.1±0.2
D	81G	85.2±0.1	83.4±0.3	94.5±0.0	94.9±0.1	88.7±0.3	94.2±0.3	94.8±0.1	80.2±0.3	85.1±0.1	89.0±0.2
Ours trained on the RTX 4090											
C	7G	83.6±0.4	79.5±0.3	95.2±0.2	89.5±0.1	78.2±0.5	93.5±0.2	93.4±0.0	77.0±0.0	78.0±0.4	85.3±0.3
D	7G	86.6±0.0	84.4±0.4	96.4±0.3	95.0±0.5	86.9±0.2	96.4±0.3	94.6±0.4	79.6±0.3	83.9±0.4	89.3±0.3
C+D	8G	88.3±0.4	87.5±0.0	98.0±0.1	95.0±0.1	**91.4**±0.3	**97.4**±0.1	95.9±0.1	86.8±0.1	86.5±0.3	92.0±0.2
C+D+M	8G	**88.6**±0.2	**87.7**±0.1	**98.6**±0.1	**95.6**±0.2	**91.4**±0.0	97.3±0.1	**96.0**±0.1	**88.3**±0.4	**87.4**±0.1	**92.3**±0.1

Accordingly, our objective is to minimize the summation of model-specific InfoNCE losses. We define the class-matched prototype $\mathcal{G}_{m,y}$ as the positive for row $[\nabla_{W_m}\ell_{\text{syn}}]_y$, while the prototypes from other classes $\{\mathcal{G}_{m,j}\}_{j\neq y}$ serve as negative samples. The contrastive formulation encourages the synthetic gradients to capture class-discriminative information by aligning with their corresponding prototypes. The model-specific InfoNCE loss is defined as:

$$\mathcal{L}^m_{\text{NCE}} = -\mathbb{E}_y \left[\log \frac{\exp\left(\text{sim}([\nabla_{W_m}\ell_{\text{syn}}]_y, \mathcal{G}_{m,y})/\tau\right)}{\sum_{j=1}^{C} \exp\left(\text{sim}([\nabla_{W_m}\ell_{\text{syn}}]_y, \mathcal{G}_{m,j})/\tau\right)} \right]. \tag{10}$$

where y indexes a class row, sim is cosine similarity, and τ is the temperature. We achieve consensus distillation via joint InfoNCE optimization. For example, utilizing CLIP captures semantic context, while DINO preserves structural details, ensuring x_{syn} captures shared patterns across feature spaces.

4 Experiments

Datasets. We evaluate the proposed method on four benchmark datasets spanning both large-scale classification and specialized fine-grained recognition. Specifically, we use ImageNet-1K [51] and its subset ImageNet-100 for large-scale validation, and Food-101 [52] and Flowers-102 [53] to assess fine-grained performance.

Implementation Details. All experiments are conducted on NVIDIA RTX 4090 GPUs at a resolution of 224×224. We employ the Adam optimizer [54] to update the synthetic images for 5,000 iterations. We fix $\mu_{\min}$=0.5 and $\mu_{\max}$=0.99 across all datasets. We utilize the frozen ViT-B architectures of CLIP, DINOv2, and MoCo-v3. Unless otherwise specified, reported means and standard deviations summarize five runs. More details are provided in the Appendix.

Evaluation and Baselines. We evaluate distilled images via linear probing and 1-shot black-box transfer: the former trains a classifier on frozen source features, while the latter uses distilled images as class prototypes for training-free nearest-neighbor inference across nine unseen architectures.

We compare our method with selection-based baselines, LGM [7], and two DiT-based generative dataset distillation baselines, MGD^3 [55] and CaO2 [56]. The selection baselines include random sampling, class centroids, and nearest neighbors, while LGM represents the current state-of-the-art for

Table 3: **Linear Probing Accuracy (%) on Fine-Grained Datasets with 1 Image Per Class.** Our cross-model consistency method yields significant performance gains over LGM, notably achieving an improvement of 7.2% on Food-101. For fine-grained datasets, ImageNet-1K class-conditional DiT samples from MGD^3 and CaO2 are mapped to target classes by CLIP text similarity before linear probing.

Train Set (1 Img/Cls)	Flowers-102				Food-101			
	CLIP	DINOv2	MoCo-v3	Average	CLIP	DINOv2	MoCo-v3	Average
Full Dataset	93.4±0.1	99.7±0.0	82.2±0.1	91.8±0.1	91.9±0.0	92.7±0.1	78.3±0.0	87.6±0.0
Neighbors	69.4±0.8	99.4±0.2	48.6±1.8	72.5±0.9	58.8±1.3	73.8±0.4	25.2±1.3	52.6±1.0
Centroids	77.5±0.7	99.4±0.1	48.2±2.6	75.0±1.1	74.1±0.6	75.6±0.1	22.7±2.0	57.5±0.9
Random	67.9±1.1	98.8±0.4	36.8±2.2	67.8±1.2	48.5±1.0	52.8±1.9	13.5±1.7	38.3±1.5
MGD^3 [55]	6.4±0.2	7.2±0.1	6.6±0.3	6.7±0.2	5.0±0.2	3.8±0.1	3.5±0.3	4.1±0.2
CaO2 [56]	6.9±0.3	7.1±0.2	6.8±0.4	6.9±0.2	5.6±0.3	4.6±0.2	3.5±0.4	4.6±0.3
LGM [7]	79.5±1.2	99.6±0.0	56.4±3.1	78.5±1.5	78.7±1.6	**83.7±0.3**	31.6±3.2	64.7±1.7
Ours$_{[C/D/M]}$	79.8±0.8	**99.7±0.1**	57.1±1.3	78.9±0.6	80.7±0.8	83.3±0.2	49.1±1.2	71.0±0.7
Ours$_{[C+D+M]}$	**84.3±0.2**	**99.7±0.1**	**61.8±1.0**	**81.9±0.5**	**82.6±0.6**	83.6±0.2	**49.4±0.7**	**71.9±0.5**

Table 4: **Black-Box 1-Shot Transfer Performance (%) and Training Time on ImageNet-100.** Distilled images are used as prototypes for training-free inference. 'Time' denotes total distillation hours.

Distill Model	GPU Time	Self-Supervised Models				Supervised Models					Average
		EVA-02	iBOT	SimMIM	BEiT	ResNet50	ConvNext	Swin-T	MobileV3	EffNet	
LGM trained on the 4 × RTX 4090											
C	3.6h	71.7±0.2	69.5±0.3	77.8±0.1	77.9±0.2	34.0±0.4	66.3±0.2	71.5±0.1	63.8±0.3	70.1±0.2	66.9±0.2
D	3.6h	80.4±0.1	84.1±0.2	88.5±0.3	92.8±0.2	61.5±0.3	85.0±0.1	83.4±0.2	73.2±0.3	77.7±0.2	80.7±0.2
Ours trained on the RTX 4090											
C	0.8h	78.6±0.1	76.1±0.2	81.3±0.3	82.0±0.1	46.5±0.4	73.6±0.2	76.6±0.1	70.3±0.3	76.3±0.2	73.5±0.2
D	0.8h	82.1±0.2	83.0±0.1	90.1±0.2	**94.5±0.1**	63.1±0.3	85.3±0.1	83.8±0.2	74.0±0.0	80.2±0.2	81.8±0.2
C+D	1.7h	85.7±0.1	85.7±0.2	91.3±0.2	93.6±0.3	73.2±0.2	86.7±0.1	84.3±0.3	80.5±0.2	85.8±0.2	85.2±0.2
C+D+M	2.6h	**86.5±0.1**	**88.1±0.2**	**93.8±0.1**	93.9±0.2	**81.2±0.3**	**89.5±0.1**	**88.5±0.2**	**84.2±0.2**	**89.4±0.1**	**88.4±0.2**

distillation in frozen feature spaces. MGD^3 and CaO2 synthesize class-conditional training images using DiT-based generators instead of selecting real samples. We also report full-dataset performance as an empirical upper bound. Details of all baseline settings are deferred to the Appendix.

4.1 Performance on Large-Scale Benchmarks

We evaluate the performance of our method on ImageNet-100 and ImageNet-1K under the IPC=1. As reported in Table 1, the proposed consensus-driven variant, Ours$_{[C+D+M]}$, establishes new state-of-the-art results across both scales. On ImageNet-100, our full method achieves an average accuracy of 88.8%, exceeding LGM by 2.2 percentage points. This performance leadership extends to the full ImageNet-1K dataset, where Ours$_{[C+D+M]}$ reaches 68.5%, surpassing LGM by 1.4 points. Qualitatively, the distillation visualizations in Figure 5 demonstrate that our synthesized images for both ImageNet scales are semantically coherent, capturing salient textural and structural features.

As shown in Figure 1, our method provides a stronger accuracy-efficiency trade-off than LGM. Specifically, LGM requires 3.6 hours of distillation time and a peak GPU memory of 81 GB to reach 84.9% accuracy with CLIP on ImageNet-100. In contrast, the single-backbone variant reported in Table 1, Ours$_{[C/D/M]}$, reaches 86.7% CLIP accuracy while requiring only 0.8 hours and 7 GB memory. Scaling to three backbones further raises Ours$_{[C+D+M]}$ to 87.4% CLIP accuracy in 2.6 hours with only 8 GB memory, making high-performance distillation feasible on consumer-grade hardware.

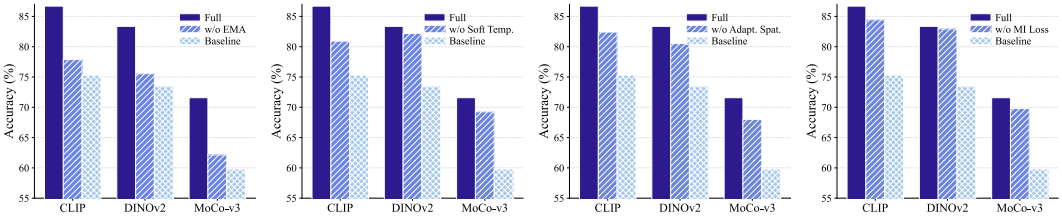
4.2 Cross-Architecture Transferability

A key objective of our method is to distill architecture-agnostic images that ensure cross-model generalization. We validate this capability through a comprehensive evaluation protocol comprising linear probing for cross-model adaptation and 1-shot inference for black-box transfer. As shown in Table 2, although distilled data is inherently limited by source model quality, single-model baselines suffer additional degradation from feature misalignment. In contrast, the proposed consensus method, Ours$_{[C+D+M]}$, achieves consistently higher accuracy across heterogeneous architectures while retaining practical efficiency. Our method completes distillation in just 2.6 hours on a single RTX 4090.



(a) ImageNet-1K (b) ImageNet-100 (c) Food-101 (d) Flowers-102

Figure 5: **Qualitative Results of Distillation Prototypes.** These images capture highly condensed, class-discriminative semantics.



(a) EMA Update (b) Soft Temporal Gate (c) Adaptive Spatial Gate (d) MI Objective

Figure 6: **Ablation Studies on ImageNet-100**. We evaluate the linear probing accuracy on the teacher (CLIP) and cross-model backbones (DINOv2, MoCo-v3). "Full" is our complete framework, while "Baseline" denotes the performance without any proposed components.

The superiority of our approach is most evident in the 1-shot results presented in Table 4. Images distilled from a single backbone often suffer severe degradation when transferred to dissimilar architectures, a failure mode best exemplified by the collapse of LGM-C to 34.0% accuracy on ResNet-50. In sharp contrast, Ours$_{[C+D+M]}$ achieves 81.2% on the same target, effectively bridging the representational gap between transformers and convolutional networks. Averaged across nine unseen models, our framework attains 88.4%, outperforming the LGM-C and LGM-D baselines. As further visualized in the radar charts of Figures 2a and 2b, these results suggest that our method captures visual features that transfer across the tested models.

4.3 Fine-Grained Visual Classification

Unlike standard classification task, fine-grained datasets such as Food-101 and Flowers-102 demand capturing subtle discriminative features to distinguish closely related categories. As indicated in Table 3, Ours$_{[C+D+M]}$ achieves an average accuracy of 71.9% on Food-101, exceeding LGM by 7.2 percentage points. This advantage is most pronounced in recovering failure modes, where we boost the MoCo-v3 baseline from 31.6% to 49.4%. On Flowers-102, Ours$_{[C+D+M]}$ reaches a new state-of-the-art average accuracy of 81.9%, while the DINOv2 source-matched variant matches the empirical ceiling at 99.7%. Qualitatively, the distilled images in Figure 5 demonstrate that prototypes encapsulate the distinctive structural and textural patterns.

4.4 Ablation Study

We conduct comprehensive ablation studies on ImageNet-100 under IPC=1 to quantify the individual contribution of each core component. The distilled data are evaluated by linear probing on the source CLIP model, and their cross-architecture transferability is further assessed on unseen DINOv2 and MoCo-v3 backbones. As shown in Figure 6, removing temporal integration leads to a sharp performance drop across all models, confirming that single-view updates are dominated by transient noise. This establishes historical accumulation as a core mechanism for smoothing optimization trajectories and recovering invariant semantic signals.

Temporal-Spatial Integration. Figure 6 further validates the importance of both temporal and spatial adaptivity. Replacing the SNR-modulated soft temporal gate with a static decay consistently degrades performance, indicating that dynamically weighting updates is essential for decoupling persistent

Table 7: **Teacher Heterogeneity and Consensus Capacity on ImageNet-100.** We report results for adding heterogeneous teacher biases. * denotes 1-shot transfer accuracy. Gen. denotes the generative prior from MAE. ConvN, MobV3, and EffN denote ConvNeXt, MobileNetV3, and EfficientNet.

Config.	Bias	Mem.	CLIP	DINOv2	MoCo-v3	RN50*	ConvN*	MobV3*	EffN*
C+D+M	ViT	8.0GB	**87.4**	**92.4**	**86.5**	81.2	89.5	84.2	**89.4**
+MAE	Gen.	8.2GB	**87.4**	92.0	86.2	81.6	89.7	84.3	89.0
+ConvN	CNN	12.9GB	86.6	92.2	86.3	**82.3**	**90.7**	**84.4**	88.5

signal from timestep-dependent noise. Similarly, removing the adaptive spatial gate, especially inverse-variance reweighting, compromises transferability, with a notable drop on MoCo-v3. These results show that temporal accumulation, adaptive temporal gating, and spatial feature selectivity are complementary mechanisms for producing robust and transferable distilled prototypes.

Mutual Information Objective. Figure 6d shows that removing the InfoNCE-based objective compromises data quality. This supports class-wise contrastive matching between synthetic gradients and prototypes as an effective objective for cross-architecture transfer in this setting.

Model Combinations. Table 5 shows that the full triplet ensemble achieves the most robust generalization. Removing any single teacher leads to a performance drop on the corresponding architecture, indicating that fusing complementary teacher perspectives is essential for forming a consensus prototype that avoids overfitting to any single representation space. Although multi-teacher distillation increases FLOPs, TPA keeps the sequential C+D+M peak memory at only 8.0GB. As shown in Table 6, two RTX 4090 GPUs reduce C+D+M distillation from 156m to 109m and complete C+D in 52m with a 14.2GB joint footprint, whereas LGM's 81 GB footprint makes comparable commodity-GPU parallelization difficult.

Table 5: **Ablation of Teacher Components on ImageNet-100.** ✓ denotes that the teacher is used for distillation.

Distilled Models			Validation Accuracy (%)			
C	D	M	CLIP	DINOv2	MoCo-v3	Avg.
✓	✓		86.8	92.3	83.6	87.5
✓		✓	86.6	89.9	86.3	87.6
	✓	✓	82.1	92.3	86.2	86.8
✓	✓	✓	**87.4**	**92.4**	**86.5**	**88.8**

Table 6: **Empirical Compute Complexity on ImageNet-100.** Sequential runs teachers one by one. Distributed splits teacher/batch workloads across two RTX 4090 GPUs.

Setup	Strategy	GPU	Mem.	Time
C+D+M	Sequential	1×4090	**8.0GB**	156m
C+D	Distributed	2×4090	14.2GB	**52m**
C+D+M	Distributed	2×4090	14.2GB	**109m**

Consensus Capacity. Table 7 studies whether enlarging the teacher set beyond the ViT triplet further improves consensus distillation. Adding MAE introduces a generative prior with little memory overhead, but yields only marginal changes and slightly reduces DINOv2 and MoCo-v3 probing accuracy. Adding ConvNeXt brings a stronger convolutional texture bias and improves transfer to CNN targets, e.g., ResNet-50 increases from 81.2% to 82.3% and ConvNeXt from 89.5% to 90.7%. However, this gain comes with a higher memory cost and a small drop on ViT-oriented linear probing. These results show that consensus distillation is not simply improved by adding more teachers. Heterogeneous teachers may introduce useful architecture-specific priors, but overly broad consensus can also force conflicting signals into the same synthetic image. The triplet therefore offers the best performance-cost trade-off in our setting.

4.5 Additional Analysis

Increasing synthetic capacity. At IPC=5, TPA (C+D+M) reaches 89.1% on ImageNet-100 and 74.8% on Food-101, exceeding LGM by 2.0 and 3.6 percentage points (Table 8). The corresponding encoder-level results appear in Appendix Table 16. Compared with single-teacher TPA at the same IPC, consensus alignment improves accuracy by 1.7 points on both datasets, extending the multi-teacher gains in Section 4.4 to a larger synthetic set. Table 9 examines the effect of synthetic-set size within single-teacher TPA by fixing the training protocol and increasing IPC from 1 to 5. Accuracy improves across all three encoders by 0.78–1.28 points, averaging 0.97 points. The consistent gains across CLIP, DINOv2, and MoCo-v3 show that additional synthetic capacity benefits different pretrained representations within the same optimization budget.

Table 8: Higher-IPC performance. IPC=5 linear probing accuracy (%) averaged over CLIP, DINOv2, and MoCo-v3, reported as five-run means and standard deviations. LGM uses $K = 2$ and TPA uses $K = 1$. The single-teacher batch size is 128.

Method	ImageNet-100	Food-101
LGM	87.1 ± 0.3	71.2 ± 0.3
TPA (single)	87.4 ± 0.2	73.1 ± 0.3
TPA (C+D+M)	$\mathbf{89.1} \pm 0.1$	$\mathbf{74.8} \pm 0.2$
Gain over LGM (pp)	+2.0	+3.6

Table 9: Effect of synthetic-set size. ImageNet-100 linear probing accuracy (%) using the same encoder for distillation and evaluation. Both IPC settings use the full ATM estimator, batch size 256, $K = 1$, and 5,000 steps, with one evaluation per synthetic set.

Encoder	IPC=1	IPC=5	Δ (pp)
CLIP	86.52	87.80	+1.28
DINOv2	92.00	92.86	+0.86
MoCo-v3	85.84	86.62	+0.78
Mean	88.12	89.09	+0.97

Temporal prototype quality. Table 10 measures agreement between the temporal prototypes and an independent reference formed from 512 real-gradient observations with inverse-RMS weighting. Cosine similarity increases from 0.9074 in the early stage to 0.9757 in the middle stage and 0.9824 in the late stage. Scale-invariant MSE decreases from 0.1733 to 0.0480 and then to 0.0348, giving an early-to-late reduction of approximately 80%. The largest gains occur during the initial accumulation period, followed by smaller improvements as more observations are incorporated. Whereas Figure 4 measures agreement between consecutive updates, these metrics quantify fidelity to a reference that uses the same form of feature weighting as the prototype estimator. Together, the two analyses show that ATM combines temporal consistency with increasing reference fidelity, supporting its use as a supervision target for synthetic-image optimization.

Table 10: Evolution of prototype quality. One ImageNet-100/CLIP trajectory with batch size 256. Metrics average the listed snapshots and all 100 classes against the independent inverse-RMS reference.

Stage (optimization steps)	Cosine $\uparrow$	$1 - \text{cosine} \downarrow$	SI-MSE $\downarrow$
Early (100/500)	0.9074	0.0926	0.1733
Middle (1,000/2,000)	0.9757	0.0243	0.0480
Late (3,000/4,000/5,000)	0.9824	0.0176	0.0348

Limitations and future directions. TPA relies on class labels and frozen teachers to define consistent gradient targets throughout distillation. Extending the method to unlabeled data requires learning class correspondences alongside the synthetic images, while evolving teachers introduce gradient drift that the temporal estimator must track. These extensions motivate studying how adaptive gating and inverse-RMS weighting interact with changing supervision. The teacher trade-offs in Table 7 also motivate automatic selection of complementary pretrained models. Comparing class-level relations on training data could provide a common basis for selecting teachers with different feature dimensions. Such selection would aim to balance transfer across target architectures with distillation cost, building on the complementary supervision provided by the current teacher triplet.

5 Conclusion

In this paper, we introduced Temporal Prototype Alignment, a scalable dataset distillation framework that replaces computationally expensive spatial ensembling with temporal accumulation. Adaptive Temporal Momentum (ATM) constructs stable teacher-local prototypes, and class-wise contrastive alignment transfers their complementary supervision to a shared synthetic dataset. In the ImageNet-100 comparison with LGM, single-teacher TPA reduces peak GPU memory from 81 GB to 7 GB, a reduction of approximately 91%, while improving accuracy. TPA (C+D+M) further improves transfer to unseen target models, including convolutional networks such as ResNet-50 and ConvNeXt and vision transformers (Tables 2 and 4). These results support temporal prototype alignment as an efficient approach to transferable dataset distillation with frozen encoders.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 42375148, 62372472, U24A20333, 62472443, and 62376287, and by the Natural Science Foundation of Hunan Province under Grant Nos. 2024JJ4068, 2024JJ3032, and 2025JJ60421. Additional support was provided by the Noncommunicable Chronic Diseases National Science and Technology Major Project under Grant Nos. 2023ZD0508200 and 2023ZD0508201, and by the International Science and Technology Innovation Joint Base of Machine Vision and Medical Image Processing in Hunan Province under Grant No. 2021CB1013. This work was also supported by the Open Research Fund of Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation under Grant No. B202405, and by the Scientific Research Program of FuRong Laboratory under Grant No. 2025PT5009.

Competing interests. The authors declare no competing interests.

References

[1] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. 2018.

[2] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021.

[3] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *ICML*, 2021.

[4] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *CVPR*, 2022.

[5] Mingyang Chen, Jiawei Du, Bo Huang, Yi Wang, Xiaobo Zhang, and Wei Wang. Influence-guided diffusion for dataset distillation. In *ICLR*, 2025.

[6] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *CVPR*, 2024.

[7] George Cazenavette, Antonio Torralba, and Vincent Sitzmann. Dataset distillation for pre-trained self-supervised vision models. In *NeurIPS*, 2025.

[8] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *ICML*, 2024.

[9] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. In *NeurIPS*, 2022.

[10] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *ICLR*, 2021.

[11] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In *NeurIPS*, 2021.

[12] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *NeurIPS*, 2022.

[13] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *ICML*, 2020.

[14] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. In *NeurIPS*, 2022.

[15] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. In *NeurIPS*, 2022.

[16] Noel Loo, Ramin Hasani, Mathias Lechner, and Daniela Rus. Dataset distillation with convexified implicit gradients. 2023.

[17] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *WACV*, 2023.

[18] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Gao Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *CVPR*, 2022.

[19] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *ICML*, 2022.

[20] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *CVPR*, 2023.

[21] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *ICLR*, 2024.

[22] Dai Liu, Jindong Gu, Hu Cao, Carsten Trinitis, and Martin Schulz. Dataset distillation by automatic training trajectories. In *ECCV*, 2024.

[23] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *CVPR*, 2023.

[24] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *ICML*, 2023.

[25] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *ICCV*, 2023.

[26] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *NeurIPS*, 2023.

[27] Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. In *CVPR*, 2025.

[28] Zeyuan Yin and Zhiqiang Shen. Dataset distillation via curriculum data synthesis in large data era. 2024.

[29] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *CVPR*, 2024.

[30] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *ICML*, 2022.

[31] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. Dream: Efficient dataset distillation by representative matching. In *ICCV*, 2023.

[32] Kai Wang, Jianyang Gu, Hansong Zhang, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. Dim: Distilling dataset into generative model. In *ECCV*, 2024.

[33] Zhiheng Ma, Anjia Cao, Funing Yang, Yihong Gong, and Xing Wei. Curriculum dataset distillation. *IEEE Transactions on Image Processing*, 2025.

[34] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[36] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[37] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[38] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[39] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[40] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023.

[41] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 2024.

[42] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[45] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[46] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[47] Shitong Shao, Zikai Zhou, Huanran Chen, and Zhiqiang Shen. Elucidating the design space of dataset condensation. In *NeurIPS*, 2024.

[48] Junyu Zhang, Daochang Liu, Shichao Zhang, and Chang Xu. Contrastive sampling chains in diffusion models. In *NeurIPS*, 2023.

[49] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[50] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019.

[51] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[52] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[53] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[54] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[55] Jeffrey A. Chan Santiago, Praveen Tirupattur, Gaurav Kumar Nayak, Gaowen Liu, and Mubarak Shah. MGD^3: Mode-guided dataset distillation using diffusion models. In *ICML*, 2025.

[56] Haoxuan Wang, Zhenghao Zhao, Junyi Wu, Yuzhang Shang, Gaowen Liu, and Yan Yan. CaO2: Rectifying inconsistencies in diffusion-based dataset distillation. In *ICCV*, pages 4722–4731, 2025.

A Appendix Overview

The appendix is organized to provide implementation details, theoretical justification, supplementary diagnostics, and qualitative evidence supporting the main paper. Appendix C describes the full experimental protocol, including the distillation setup, teacher and target model suites, data preprocessing, linear probing evaluation, and training-free 1-shot transfer evaluation. Appendix D.1 gives the derivation of the variational lower bound used by the mutual-information objective, clarifying how the InfoNCE loss encourages cross-model consensus between synthetic gradients and temporal prototypes. Appendix D.2 then reports supplementary diagnostics, including spatial-view sensitivity, batch-size fairness, teacher scaling under extreme compression, extended black-box transfer, I/O profiling, and the relation to train-from-scratch dataset distillation. Appendix D.8 studies the sensitivity of the temporal gate thresholds, while Appendix D.9 evaluates the method under a larger IPC=5 distillation budget. Finally, Appendix E presents additional synthesized-image visualizations for both single-teacher and multi-teacher settings, including ImageNet-scale and fine-grained datasets.

B Impact Statement

This work contributes to the field of Effective Dataset Distillation by introducing a memory-efficient framework that operates on frozen features. A primary societal impact of our research is the reduction of computational resources required for training and data synthesis, which aligns with the goals of Green AI and sustainable computing. By making high-quality synthetic data accessible with limited hardware resources, we hope to facilitate broader participation in data-efficient learning research. We do not foresee immediate negative societal consequences, though general considerations regarding the use of synthetic data apply.

C Experimental Details

C.1 Implementation Details

We implement Temporal Prototype Alignment in PyTorch and run the reported distillation experiments on NVIDIA RTX 4090 GPUs. Unless otherwise specified, each synthetic set is optimized for $T = 5{,}000$ iterations with Adam [54] using learning rate $\eta = 2 \times 10^{-3}$. The source teachers are frozen throughout distillation. The default consensus configuration uses CLIP ViT-B/16, DINOv2 ViT-B/14, and MoCo-v3 ViT-B/16.

Synthetic images are represented with a multi-scale pyramid parameterization. A new resolution band is activated every 200 iterations, and each band is initialized with $\mathcal{N}(0, 1)$. The rendered synthetic image is obtained by summing the bilinearly upsampled pyramid bands and applying a smooth clamp:

$$I_{\text{syn}} = \frac{1}{2} + \frac{1}{2} \tanh \left(\sum_{l=0}^{L} \texttt{Upsample}(P_l) \right). \tag{11}$$

Real images are resized to 256×256 and randomly cropped to 224×224 during distillation. The differentiable augmentation pipeline applies random resized crop, horizontal flip with probability 0.5, and Gaussian perturbations $\mathcal{N}(\mu = 0.5, \sigma = 0.2)$. TPA uses one differentiable view per iteration ($K = 1$), while LGM baselines use their reported spatial-view settings.

C.2 Model Suite and Evaluation

We evaluate each distilled set with both source backbones and unseen target models. The target suite spans masked-image-modeling transformers (iBOT, BEiT v2, SimMIM, EVA-02), hierarchical transformers (Swin), and convolutional architectures (ResNet-50, ConvNeXt, EfficientNet, MobileNetV3). Real-image baselines follow the LGM protocol [7]. For distillation, the default batch size is 256 for single-teacher runs and 128 for multi-teacher runs. The original single-teacher IPC=5 results in Table 16 also use 128, whereas Table 9 uses 256 for both IPC values.

Linear probing. For each evaluation backbone, we freeze the pretrained feature extractor and use the distilled images as the only training set for a newly initialized linear classifier. The classifier is

trained from scratch on top of fixed features, without updating the backbone or using any real training images. Test images are resized to 256 and center-cropped to 224×224 before feature extraction.
1-shot inference. For the non-parametric evaluation, each distilled image is encoded by the target backbone and treated as the prototype of its corresponding class. Each test image is assigned to the class whose prototype has the highest cosine similarity, without training a target-specific classifier or performing any target-backbone adaptation.

D Additional Results

D.1 Derivation of the Variational Lower Bound

Condition on a fixed head and prototype bank, with classes sampled uniformly. Assume distinct class prototypes, so $H(\boldsymbol{c}) = \log C$. The bound below characterizes the class information captured by synthetic gradient rows relative to their corresponding teacher-local prototypes. Let $\boldsymbol{v} = [\nabla_{W_m} \ell_{\text{syn}}]_y$ denote the synthetic gradient and $\boldsymbol{c} = \mathcal{G}_{m,y}$ denote the corresponding class-matched prototype. The set $\{\mathcal{G}_{m,j}\}_{j \neq y}$ represents the $C-1$ negative prototypes. The variational probability $q_m(\boldsymbol{c} \mid \boldsymbol{v})$ is defined as:

$$q_m(\boldsymbol{c} \mid \boldsymbol{v}) = \frac{\exp(\text{sim}(\boldsymbol{v}, \boldsymbol{c})/\tau)}{\sum_{j=1}^{C} \exp(\text{sim}(\boldsymbol{v}, \mathcal{G}_{m,j})/\tau)} \tag{12}$$

The derivation proceeds as follows:

$$\begin{aligned}
\mathcal{L}_{\text{NCE}}^{m} &= -\mathbb{E}_{(\boldsymbol{v},\boldsymbol{c})\sim p(\boldsymbol{v},\boldsymbol{c})}\Big[\log q_m(\boldsymbol{c} \mid \boldsymbol{v})\Big] \\
&= \mathbb{E}_{\boldsymbol{v}}\left[\sum_{\boldsymbol{c}\in\{\mathcal{G}_{m,1},\ldots,\mathcal{G}_{m,C}\}} p(\boldsymbol{c} \mid \boldsymbol{v})\Big(-\log q_m(\boldsymbol{c} \mid \boldsymbol{v})\Big)\right] \\
&= \mathbb{E}_{\boldsymbol{v}}\left[\sum_{\boldsymbol{c}} p(\boldsymbol{c} \mid \boldsymbol{v})\Big(-\log q_m(\boldsymbol{c} \mid \boldsymbol{v})\Big)\right] \\
&= \mathbb{E}_{\boldsymbol{v}}\left[\sum_{\boldsymbol{c}} p(\boldsymbol{c} \mid \boldsymbol{v})\Big(-\log p(\boldsymbol{c} \mid \boldsymbol{v})\Big) + \sum_{\boldsymbol{c}} p(\boldsymbol{c} \mid \boldsymbol{v}) \log \frac{p(\boldsymbol{c} \mid \boldsymbol{v})}{q_m(\boldsymbol{c} \mid \boldsymbol{v})}\right] \\
&= \mathbb{E}_{\boldsymbol{v}}\left[\sum_{\boldsymbol{c}} p(\boldsymbol{c} \mid \boldsymbol{v})\Big(-\log p(\boldsymbol{c} \mid \boldsymbol{v})\Big)\right] + \mathbb{E}_{\boldsymbol{v}}\left[\sum_{\boldsymbol{c}} p(\boldsymbol{c} \mid \boldsymbol{v}) \log \frac{p(\boldsymbol{c} \mid \boldsymbol{v})}{q_m(\boldsymbol{c} \mid \boldsymbol{v})}\right] \\
&= H(\boldsymbol{c} \mid \boldsymbol{v}) + \mathbb{E}_{\boldsymbol{v}}\Big[\text{KL}\big(p(\cdot \mid \boldsymbol{v}) \,\|\, q_m(\cdot \mid \boldsymbol{v})\big)\Big] \\
&\geq H(\boldsymbol{c} \mid \boldsymbol{v}) \\
&= H(\boldsymbol{c}) - \mathcal{I}(\boldsymbol{v}; \boldsymbol{c}) \\
&= \log(C) - \mathcal{I}(\boldsymbol{v}; \boldsymbol{c})\,.
\end{aligned} \tag{13}$$

By rearranging the inequality, we validate the variational lower bound presented in the main paper:

$$\sum_{m=1}^{M} \mathcal{I}\left([\nabla_{W_m} \ell_{\text{syn}}]_y; \mathcal{G}_{m,y}\right) \geq M \log(C) - \sum_{m=1}^{M} \mathcal{L}_{\text{NCE}}^{m}. \tag{14}$$

Hence, jointly minimizing $\mathcal{L}_{\text{NCE}}^{m}$ maximizes the variational lower bound, enforcing x_{syn} as a shared information bottleneck that distills semantic intersections across heterogeneous architectures.

D.2 Sensitivity to Spatial Ensemble Size

We further analyze how the number of differentiable spatial views affects LGM under the single-teacher ImageNet-100 setting. This study is intended to clarify the role of spatial ensembling in gradient-based distillation and to contrast it with the temporal aggregation mechanism used by TPA. As shown in Table 11, LGM benefits substantially from increasing K: larger spatial ensembles

improve both source-model accuracy and cross-model transfer, especially on MoCo-v3. However, when K is reduced to 1, its transfer performance drops sharply, indicating that single-view gradient estimates are highly sensitive to augmentation-induced variance.

Table 11: **Sensitivity to Spatial Ensemble Size on ImageNet-100.** We report linear probing accuracy (%) after CLIP-teacher distillation, evaluated on CLIP, DINOv2, and MoCo-v3. Increasing K averages more spatial views in LGM, while TPA uses $K = 1$ and accumulates information temporally.

Method	K	CLIP	DINOv2	MoCo-v3	Average
LGM	1	77.9	71.7	44.2	64.6
LGM	2	82.3	78.7	50.4	70.5
LGM	3	83.6	78.0	53.4	71.7
LGM	10	84.9	84.5	60.1	76.5
TPA	1	**86.7**	**83.2**	**70.5**	**80.1**

These results highlight a key distinction between the two strategies. LGM improves stability by averaging gradients over simultaneous spatial views, which increases the per-iteration computational cost as K grows. In contrast, TPA keeps $K = 1$ and reconstructs stable supervision through temporal prototype accumulation. The stronger $K = 1$ performance of TPA therefore suggests that temporal integration can provide an effective alternative to large spatial ensembles, improving scalability while preserving cross-backbone transferability.

D.3 Batch-Size Analysis for Single-Teacher Distillation

We additionally study how the effective batch size affects single-teacher distillation on ImageNet-100. This analysis complements the K-sensitivity study by separating two sources of supervision density: the number of spatial views per sample and the number of real samples processed per iteration. LGM uses a large batch together with multiple spatial views, which provides a strong per-iteration estimate but incurs substantial memory cost. TPA instead uses temporal prototype accumulation, allowing us to increase the real-image batch size while keeping the memory footprint small.

Table 12: **Batch-Size and Memory Analysis on ImageNet-100.** We report source-matched linear probing accuracy (%). Each column uses its own teacher for distillation and evaluation. TPA maintains low GPU memory usage as batch size increases, without requiring large spatial ensembles.

Method	Batch	GPU Mem.	CLIP	DINOv2	MoCo-v3	Avg.
LGM	1000	81.2GB	84.9	91.5	83.4	86.6
TPA	256	7.0GB	**86.7**	92.2	85.2	88.0
TPA	512	7.1GB	86.2	**92.4**	**85.7**	**88.1**

The results show that increasing the TPA batch size from 256 to 512 yields similar average accuracy, suggesting that the method is not highly sensitive to this range once the temporal prototype estimates are stable. More importantly, both TPA configurations remain substantially more memory-efficient than the LGM single-teacher setting while achieving higher average accuracy. This supports the view that TPA improves the efficiency of the supervision estimate through temporal aggregation rather than relying on a large per-iteration spatial sampling budget.

D.4 Teacher Scaling under Extreme Compression

We additionally study whether using a stronger teacher backbone directly translates into better distilled prototypes under the IPC=1 constraint. This analysis complements the main results by separating two factors: the representational strength of an individual teacher and the capacity of a single synthetic prototype to absorb information from multiple teachers. As shown in Table 13, replacing DINOv2 with DINOv3 improves the single-teacher setting, suggesting that stronger teacher representations can provide more informative supervision.

Table 13: **Teacher Scaling under IPC=1 on ImageNet-100.** We report linear probing accuracy (%) across evaluation backbones. Stronger teachers improve single-teacher distillation, while gains become smaller in the multi-teacher setting due to the limited information capacity of one synthetic image per class.

Configuration	Setup Detail	CLIP	DINOv2	MoCo-v3	DINOv3	Avg.
Single Teacher	DINOv2	78.5	92.2	81.3	89.9	85.5
Single Teacher	DINOv3	81.6	91.1	83.1	92.2	87.0
Triplet Teacher	C+DINOv2+M	87.4	**92.4**	**86.5**	90.3	89.2
Triplet Teacher	C+DINOv3+M	**87.9**	91.1	**86.5**	**92.3**	**89.5**

The multi-teacher results show a different trend. Although DINOv3 improves its own evaluation column and slightly increases the average accuracy, the gain over the CLIP+DINOv2+MoCo-v3 triplet remains modest. This suggests that, under IPC=1, performance is not determined solely by teacher strength. It is also constrained by the finite capacity of a single prototype and by how complementary the teacher signals are. We therefore use the CLIP+DINOv2+MoCo-v3 consensus as our primary setting, as it provides a strong cost–performance trade-off while preserving broad cross-backbone transferability.

D.5 Extended Black-Box 1-Shot Transfer

We provide an extended black-box transfer evaluation to examine whether the distilled prototypes remain useful beyond the teacher backbones used during distillation. In this setting, each synthetic image is treated as a class prototype, and test samples are classified by nearest-neighbor cosine similarity under unseen target encoders. The target suite covers masked-image-modeling transformers, hierarchical transformers, and convolutional networks, allowing us to assess whether the distilled set captures broadly transferable semantics rather than features tied to a particular teacher family.

Table 14: **Extended Black-Box 1-Shot Transfer on ImageNet-100.** We report 1-shot nearest-prototype accuracy (%) on unseen target backbones. TPA achieves stronger average transfer while requiring substantially less distillation time.

Method	Setup	Time	EVA	SimMIM	BEiT	RN50	ConvN	Swin	MobV3	Eff.	Avg.
LGM	C+D	6.7h	82.2	89.0	93.0	66.3	85.2	83.6	78.1	82.5	82.5
LGM	C+D+M	9.7h	83.0	90.5	93.5	74.2	88.2	85.8	81.6	85.3	85.3
TPA	C+D+M	2.6h	**86.5**	**93.8**	**93.9**	**81.2**	**89.5**	**88.5**	**84.2**	**89.4**	**88.4**

The results show that adding teachers improves LGM, but the resulting transfer performance still trails TPA despite the higher distillation cost. The gains of TPA are particularly pronounced on structurally different CNN targets such as ResNet-50 and EfficientNet, suggesting that temporal consensus does not merely optimize for the source teacher representations. Instead, it helps recover class-level prototypes that are more invariant across architectural families, which is important for black-box reuse of distilled datasets.

D.6 I/O Throughput and DataLoader Profiling

Beyond accuracy and GPU memory, we also profile the input pipeline to understand how reducing spatial redundancy affects end-to-end distillation efficiency. All measurements are collected on ImageNet-100 over 5,000 iterations using a PCIe NVMe SSD and 16 DataLoader workers. As shown in Table 15, replacing large spatial ensembles with $K = 1$ temporal integration substantially reduces DataLoader latency. Compared with the multi-teacher LGM setting, TPA reduces pure I/O load from 0.062s to 0.008s per iteration, corresponding to a $7.8\times$ reduction, and decreases total preparation time from 0.220s to 0.043s per iteration, corresponding to a $5.1\times$ reduction. Meanwhile, the effective throughput increases from 11.9 GB/s to 21.6 GB/s, yielding a $1.8\times$ improvement.

Table 15: **I/O Throughput and Latency Profiling on ImageNet-100.** We report representative per-iteration DataLoader profiling statistics. LGM relies on spatial view ensembling, whereas TPA uses K=1 and accumulates supervision temporally.

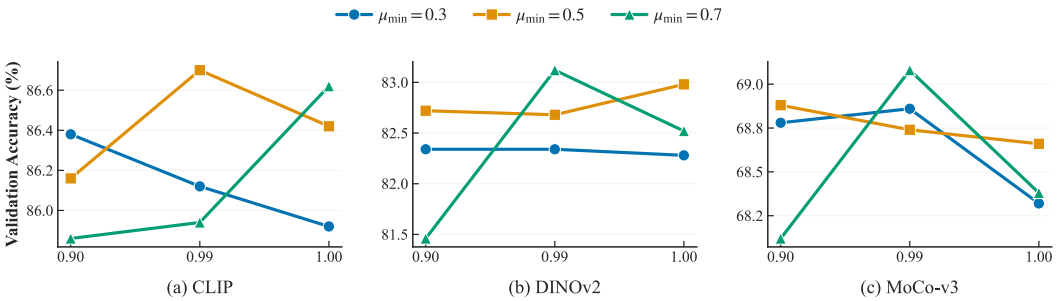
Method	Setup	I/O Load sec	Aug. Wait sec	Total Prep sec/iter	Throughput GB/s
LGM	CLIP	0.062	0.052	0.115	11.9
LGM	C+D+M	0.062	0.158	0.220	11.9
TPA	C+D+M	**0.008**	**0.034**	**0.043**	**21.6**

These profiling results show that the efficiency gain of TPA is not limited to lower GPU memory usage. By avoiding repeated spatial views within each iteration, TPA reduces input-pipeline waiting time and nearly doubles effective throughput. This supports the design choice of shifting redundancy reduction from simultaneous spatial averaging to temporal prototype accumulation, which improves scalability at both the compute and data-loading levels.

D.7 Relation to Train-from-Scratch Dataset Distillation

Our study focuses on distilling data for frozen pretrained feature extractors, where the synthetic set is used to train a linear classifier or to form nearest-prototype classifiers on top of fixed representations. This setting differs from traditional train-from-scratch dataset distillation, which optimizes synthetic images so that a randomly initialized neural network can be trained end-to-end from them. The two paradigms therefore impose different optimization targets and evaluation protocols: frozen-feature distillation measures how well synthetic images preserve class-discriminative semantics in pretrained representation spaces, whereas train-from-scratch distillation must additionally support feature learning from extremely limited data.

This distinction becomes especially important at ImageNet scale under IPC=1. Existing train-from-scratch methods can suffer severe optimization collapse in this regime. For example, RDED-style large-scale distilled data reports around 6.6% top-1 accuracy under the train-from-scratch evaluation protocol [29]. We therefore keep the main comparisons within the frozen-feature distillation setting, where LGM and our method share the same evaluation objective.



(a) CLIP (b) DINOv2 (c) MoCo-v3

Figure 7: **Hyperparameter Sensitivity Analysis of Temporal Gating.** We evaluate the linear probing accuracy on ImageNet-100 by varying the maximal gate threshold $\mu_{\max}$ and the minimum gate threshold $\mu_{\min}$ (lines). The results are reported on the source teacher (a) CLIP, and unseen target architectures (b) DINOv2 and (c) MoCo-v3. The default setting ($\mu_{\max} = 0.99$ and $\mu_{\min} = 0.5$) achieves the best trade-off between source-model performance and cross-model generalization.

D.8 Parameter Sensitivity

To verify the robustness of the temporal gating mechanism, we analyze the impact of the gate thresholds $\mu_{\max}$ and $\mu_{\min}$, as illustrated in Figure 7. The results show that the method is stable across a broad range of gate settings, with CLIP accuracy varying only moderately as the thresholds change. The best CLIP performance is obtained at $\mu_{\max} = 0.99$ and $\mu_{\min} = 0.5$, reaching 86.7%. This suggests that a high upper retention bound is useful for suppressing noisy single-view updates, while a moderate lower bound still allows the temporal prototype to adapt to reliable semantic signals. We therefore use $\mu_{\max} = 0.99$ and $\mu_{\min} = 0.5$ as the default configuration in our experiments.

Table 16: **Performance Analysis at IPC=5.** We report linear probing accuracy (%) with 5 images per class. Our method consistently outperforms the state-of-the-art LGM across all datasets. The significant improvement on the fine-grained Food-101 dataset demonstrates our method's ability to capture complex intra-class distributions.

Train Set (5 Img/Cls)	ImageNet-100				Food-101			
	CLIP	DINOv2	MoCo-v3	Average	CLIP	DINOv2	MoCo-v3	Average
Full Dataset	92.5±0.0	95.2±0.1	89.4±0.3	92.4±0.1	91.9±0.0	92.7±0.0	78.3±0.0	87.6±0.0
Neighbors	81.5±0.3	86.4±0.2	79.9±0.1	82.6±0.2	77.5±0.1	81.4±0.1	38.9±0.0	65.9±0.1
Centroids	82.5±0.2	87.6±0.2	79.0±0.1	83.0±0.4	78.0±0.4	80.5±0.3	32.0±0.3	63.5±0.0
Random	78.3±0.9	83.4±0.8	70.4±0.4	77.4±0.7	67.8±0.7	73.4±0.7	29.4±0.4	56.9±0.8
LGM	85.8±0.1	91.7±0.2	83.8±0.4	87.1±0.3	83.3±0.2	83.8±0.4	46.4±0.1	71.2±0.3
Ours$_{[C/D/M]}$	86.3±0.2	91.9±0.1	84.1±0.2	87.4±0.2	84.7±0.1	83.9±0.1	50.7±0.7	73.1±0.3
Ours$_{[C+D+M]}$	**87.9**±0.2	**92.6**±0.0	**86.7**±0.1	**89.1**±0.1	**85.3**±0.1	**84.7**±0.1	**54.4**±0.4	**74.8**±0.2

D.9 Quantitative and Qualitative Analysis at IPC=5

Table 16 provides the encoder-level IPC=5 results underlying Table 8, including full-data and sampling baselines. LGM uses $K = 2$ because of memory constraints on four RTX 4090 GPUs. TPA (C+D+M) improves over single-teacher TPA on every listed evaluation encoder for both datasets, indicating that the aggregate gains are shared across the three representations. Increasing IPC from 1 to 5 raises TPA (C+D+M) accuracy from 71.9% to 74.8% on Food-101 and from 88.8% to 89.1% on ImageNet-100. Thus, the observed capacity gain is larger on the fine-grained dataset, where multiple images per class can represent variations in appearance. On Food-101, the 2.9-point gain from increasing IPC and the 3.6-point advantage over LGM at IPC=5 describe complementary effects of synthetic capacity and distillation method. Table 9 further examines capacity under a fixed single-teacher protocol.

The synthesized images illustrate how the larger set represents variation within each class. Figure 13 shows structural and textural differences among Food-101 images, including variations in ingredients and presentation. On ImageNet-100, Figure 14 shows differences in object pose, background, and viewpoint among the five images of each class. These visualizations complement the encoder-level evaluation by showing the appearance variation represented in the synthetic sets.

E Additional Visualizations

We provide additional visualizations of the synthesized images. In Figure 8 and Figure 9, we present samples from Food-101 and ImageNet-100 distilled using the single CLIP teacher. Furthermore, to demonstrate the effectiveness of our method under the multi-teacher setting, we showcase results distilled using the ensemble of CLIP, DINOv2, and MoCo-v3. We illustrate these outcomes for Food-101 in Figure 10, for ImageNet-100 in Figure 11, and for Flowers-102 in Figure 12.

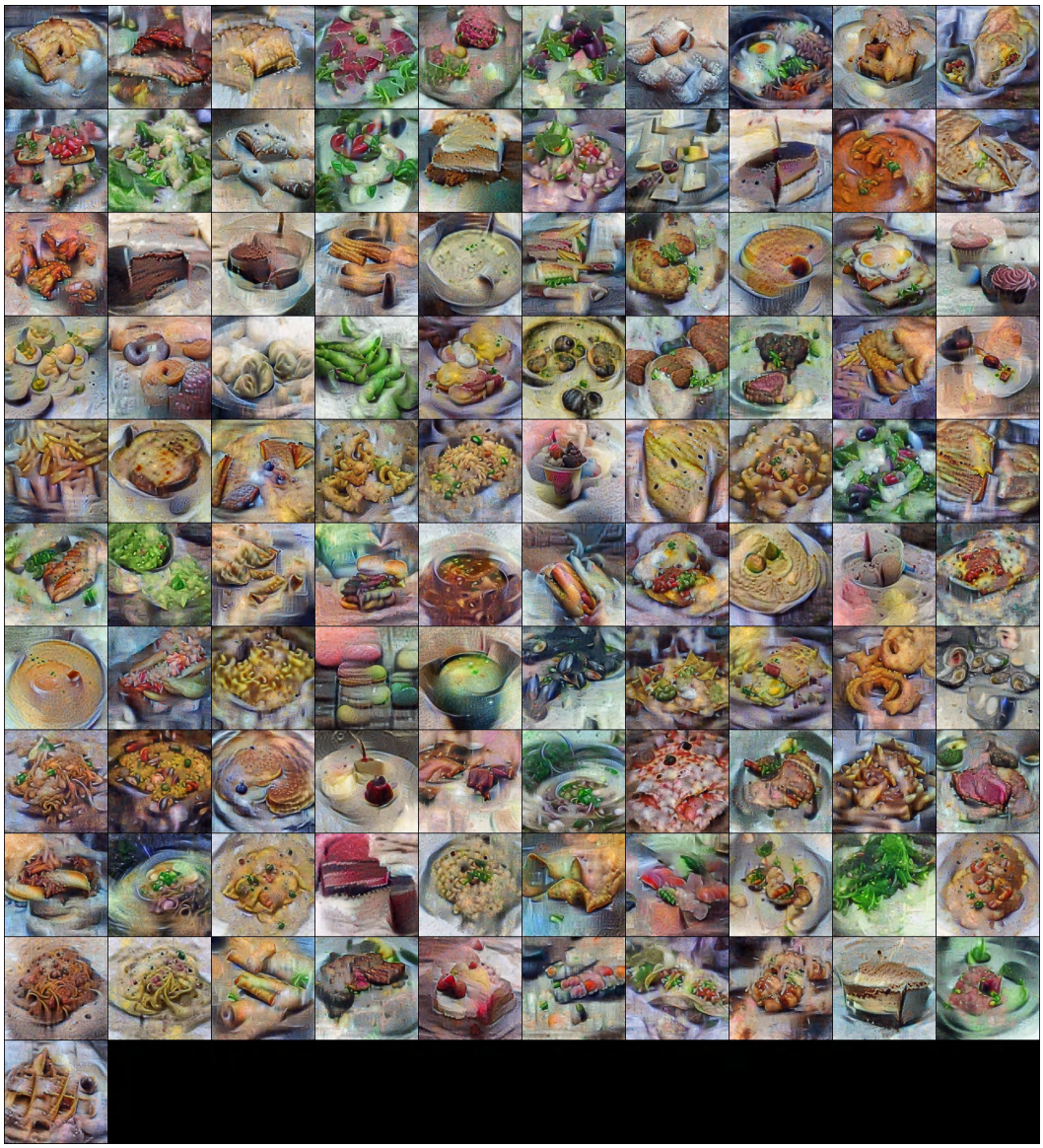


Figure 8: Food-101 distilled using CLIP.



Figure 9: ImageNet-100 distilled using CLIP.

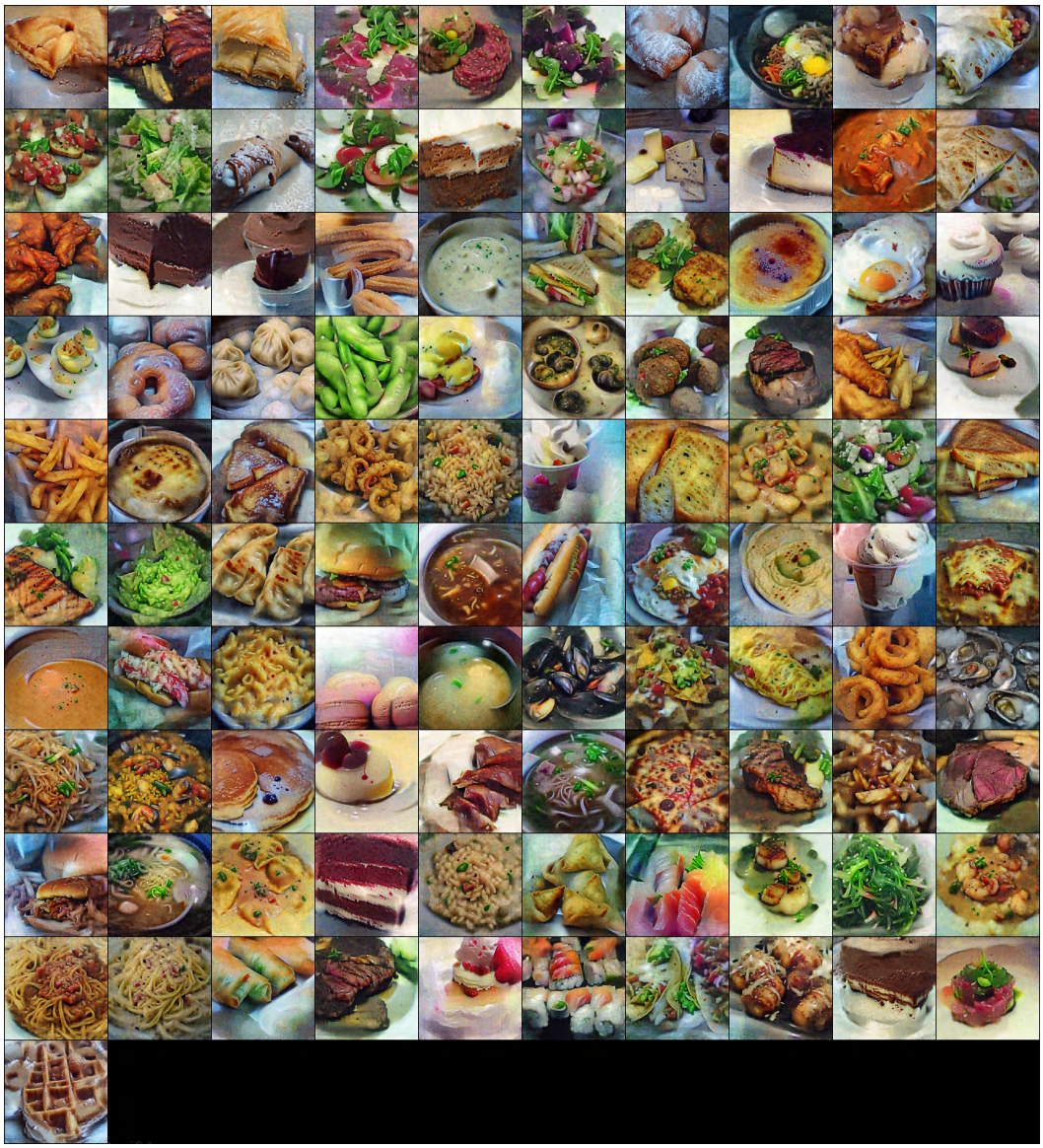


Figure 10: Food-101 distilled using CLIP, DINOv2, and MoCo-v3.

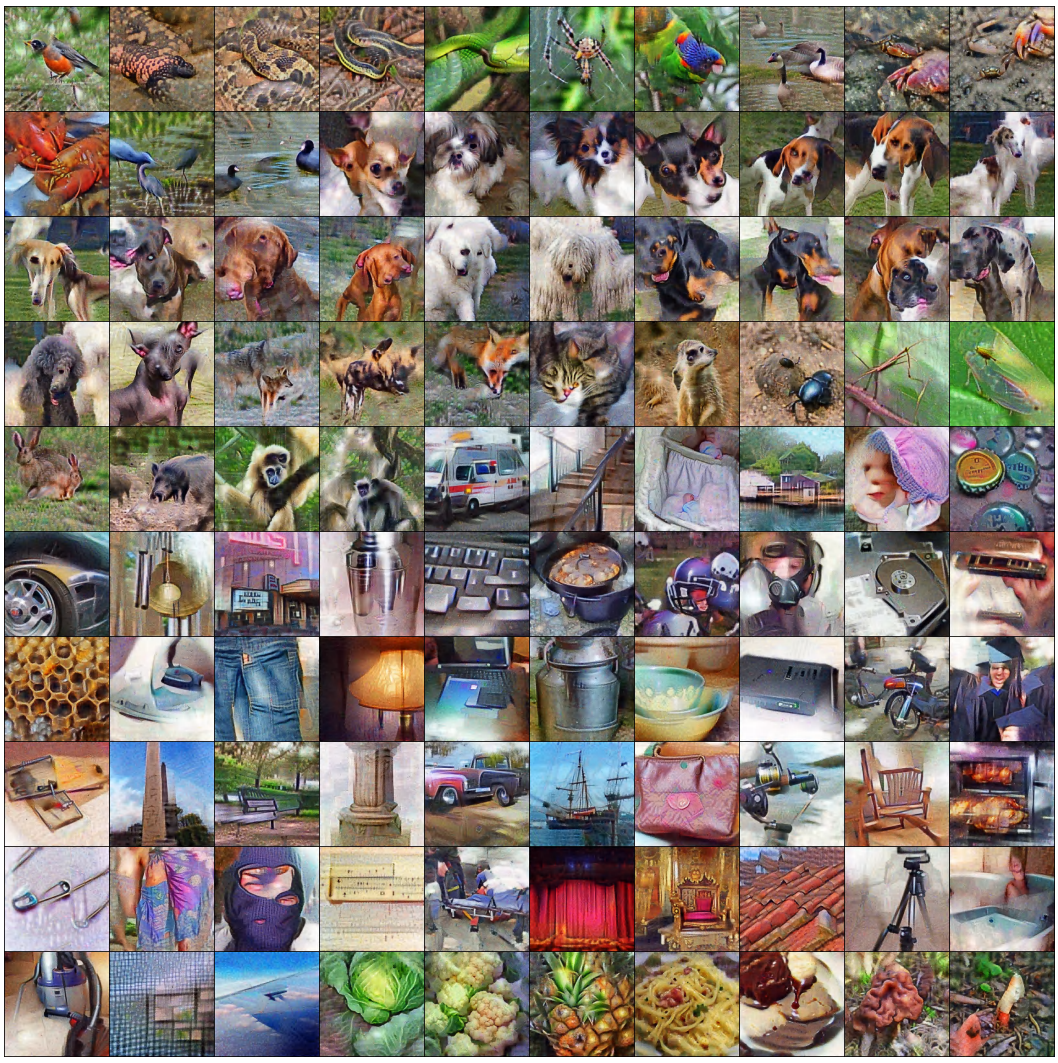


Figure 11: ImageNet-100 distilled using CLIP, DINOv2, and MoCo-v3.

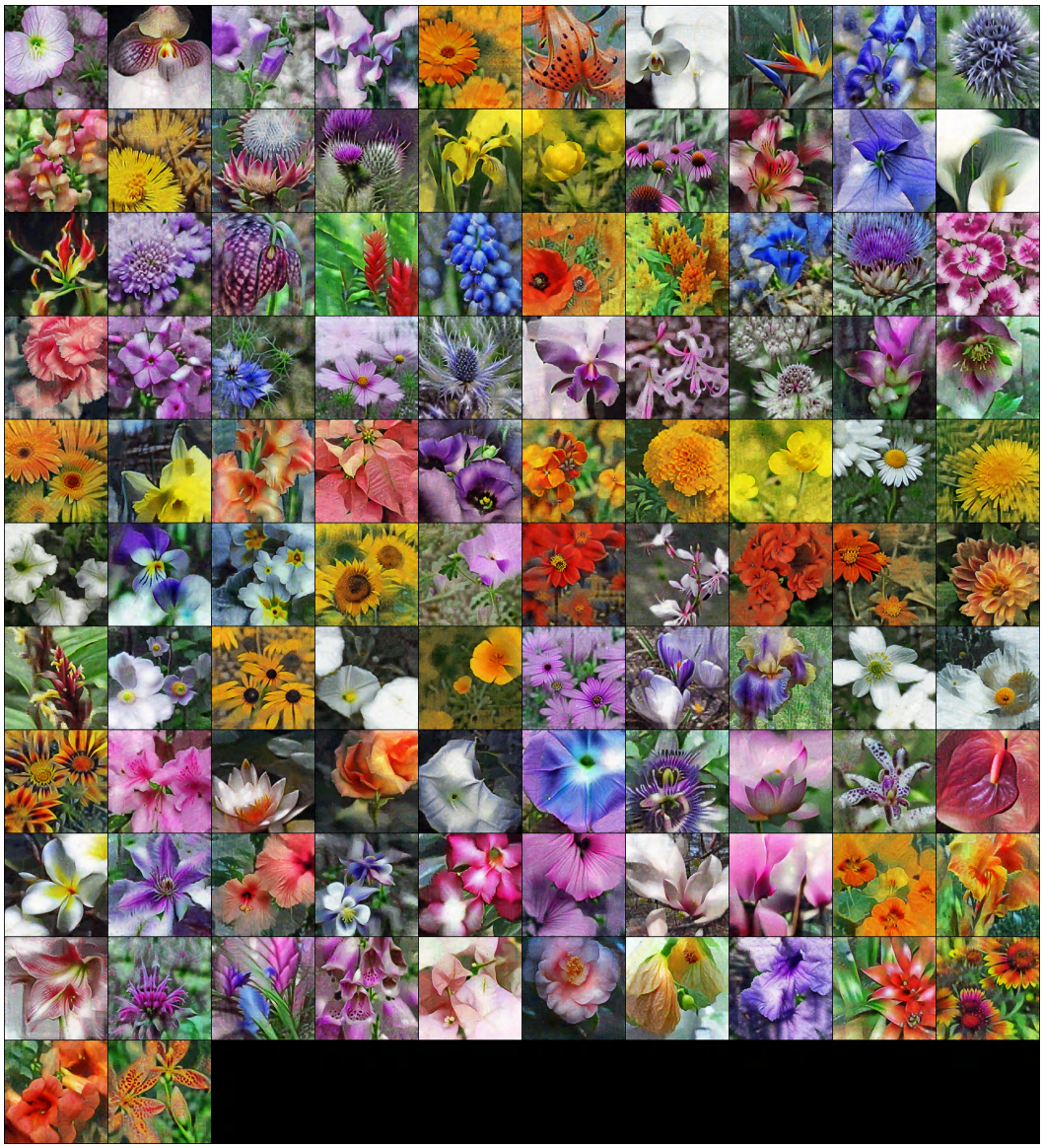


Figure 12: Flowers-102 distilled using CLIP, DINOv2, and MoCo-v3.

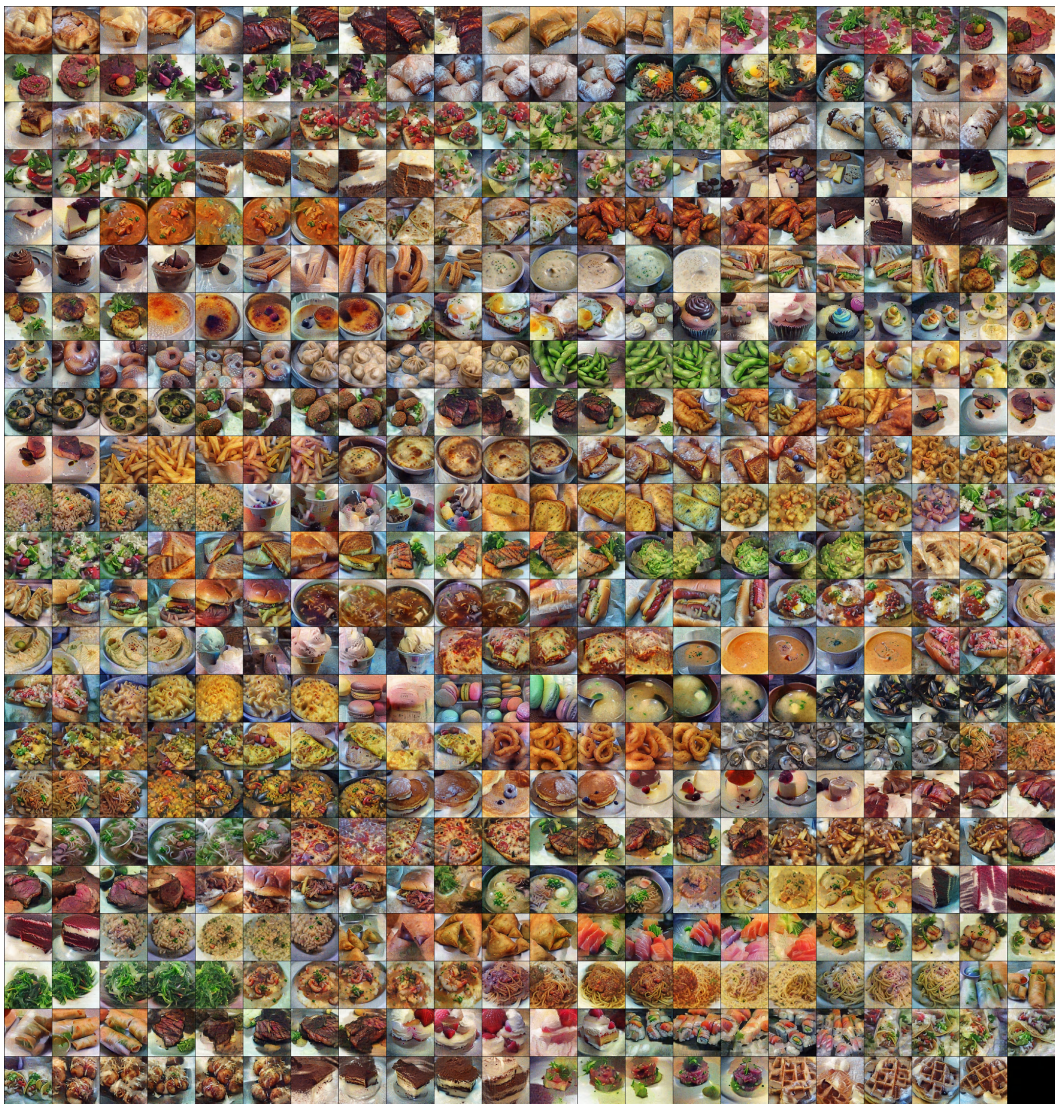


Figure 13: Food-101 distilled using CLIP, DINOv2, and MoCo-v3 (IPC=5).



Figure 14: ImageNet-100 distilled using CLIP, DINOv2, and MoCo-v3 (IPC=5).

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 abstract and Introduction state the scope and main contributions of TPA, including temporal prototype estimation, adaptive momentum gating, cross-model consensus alignment, and efficiency/transfer claims. These claims are supported by the experimental results in the main tables and figures.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper discusses the frozen-feature and IPC-focused setting, the computational constraints motivating the method, and the evaluated dataset/model scope in the Method, Experiments, and Appendix diagnostics.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: The theoretical component is limited to the variational mutual-information objective. The assumptions and objectives are stated in the Method section. The Appendix provides the full lower-bound derivation used by the method.

Guidelines:

- The answer [N/A] 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 Method and Appendix describe the objective, teacher backbones, data preprocessing, augmentation, optimizer, training iterations, and evaluation protocols needed to reproduce the main results.

Guidelines:

- The answer [N/A] 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: The project page is linked in the abstract. Code and data access is restricted to the authors. The Method and Appendix describe the experimental protocol, with implementation and resource information on the project page.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The Experiments section and Appendix specify datasets, resolution, frozen teachers, Adam optimization for 5,000 iterations, augmentations, linear probing, and 1-shot transfer protocols.

   Guidelines:

   - The answer [N/A] 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: The main result tables report mean $\pm$ standard deviation over five runs. Table 9 gives one evaluation per synthetic set. Table 10 reports checkpoint-averaged measurements along one prototype trajectory, describing its evolution during optimization.

Guidelines:

- The answer [N/A] 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: The paper reports RTX 4090 hardware, GPU memory, and wall-clock time for TPA and LGM comparisons. Additional profiling and distributed-compute details are provided in the appendix tables.

   Guidelines:

   - The answer [N/A] 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: The research uses public benchmark datasets and pretrained vision models, does not involve human subjects, and conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: The Appendix Impact Statement discusses reduced computational resource requirements and broader accessibility for data-efficient learning research. It also notes general considerations associated with the use of synthetic data.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained generator, scraped dataset, or foundation model. The method evaluates compact synthetic images for benchmark dataset distillation.

Guidelines:

- The answer [N/A] 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: [No]

Justification: The paper cites the datasets, baselines, and pretrained models used in the experiments. However, the current manuscript does not explicitly enumerate the license or terms of use for each existing asset.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: This paper does not publicly release a new dataset, model, or code asset. It reports a method and experimental results.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The paper does not involve crowdsourcing experiments or human-subject research, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development and experiments do not use LLMs as an important, original, or non-standard component. Any writing, editing, or formatting assistance is outside the scope requiring declaration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.