FRAGILE BY DISTILLATION: NATIVE-OBJECTIVE VICINAL ALIGNMENT FOR RELIABLE SYNTHETIC SETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset distillation (DD) plays an important role in creating compact and efficient training sets for reducing downstream computational costs and enabling data sharing. However, previous works have overlooked a critical concern: the condensed datasets are vulnerable to malicious modifications during delivery. Our investigation shows that this weakness is prevalent across representative distillation methods and does not consistently diminish at larger distillation budgets. Models trained on these compromised sets suffer significant performance degradation. To bridge this gap, we introduce **N**ative **O**bjective **V**icinal **A**lignment (NOVA), a framework which can be incorporated into diverse matching-based DD approaches by the optimization of their native distillation objectives. The key innovation of NOVA lies in jointly matching the synthetic set and its perturbed neighbors to the same real-data target so that the matching holds across a neighborhood of the delivered set rather than at a single point. This matching reduces the shift in the native training signal induced by perturbation. Theoretical analysis and extensive experiments show that NOVA substantially improves the downstream performance of condensed datasets under malicious modifications while preserving clean utility and requiring no changes to the synthetic-set format or the downstream training pipeline. Across seven DD methods, five datasets, and three budgets, NOVA improves accuracy under different perturbations unseen during distillation in 96% of the evaluated settings, with limited impact on clean accuracy.

1 INTRODUCTION

Deep learning has achieved remarkable progress in visual recognition (He et al., 2016; Dosovitskiy et al., 2021; Deng & Yang, 2025). However, these gains rely increasingly on large-scale training data (Sun et al., 2017), which makes repeated model training, data storage, and data sharing ever more costly, especially for practitioners with limited computational resources. Dataset distillation (DD) (Wang et al., 2018) offers a promising remedy. By matching gradients (Zhao et al., 2021), distributions (Zhao & Bilen, 2023), features (Wang et al., 2022), or training trajectories (Cazenavette et al., 2022) between synthetic and real data, DD condenses a large dataset into a small yet highly informative synthetic set. Because of its compactness, the resulting set can be stored, transferred, and reused across many downstream training runs. This reuse, however, leaves a window between distillation and downstream training in which the synthetic set can be changed (Liu et al., 2023; Yang et al., 2026). Most existing DD research focuses on ensuring that models trained on condensed datasets achieve accuracy comparable to those trained on the original data (Yin et al., 2023; Guo et al., 2024). However, they have overlooked the integrity of the condensed dataset itself, which is susceptible to perturbations modification after distillation and before downstream training.

This oversight is consequential, because distilled sets turn out to be far more fragile than real data. As illustrated in Fig. 1(a) and (b), an attacker who perturbs the synthetic set S after distillation can degrade the models subsequently trained on it. Under the same subset size and perturbation budget, distilled sets suffer larger accuracy drops than real subsets (Fig. 1(c)), and this gap does not consistently shrink as IPC increases (Fig. 1(d)). This vulnerability follows from the extreme compression that DD performs. To approximate a large dataset with only a few synthetic images, DD concentrates the influence of many real examples into a small training artifact. Because this artifact is used repeatedly throughout optimization, an altered signal affects successive parameter updates

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

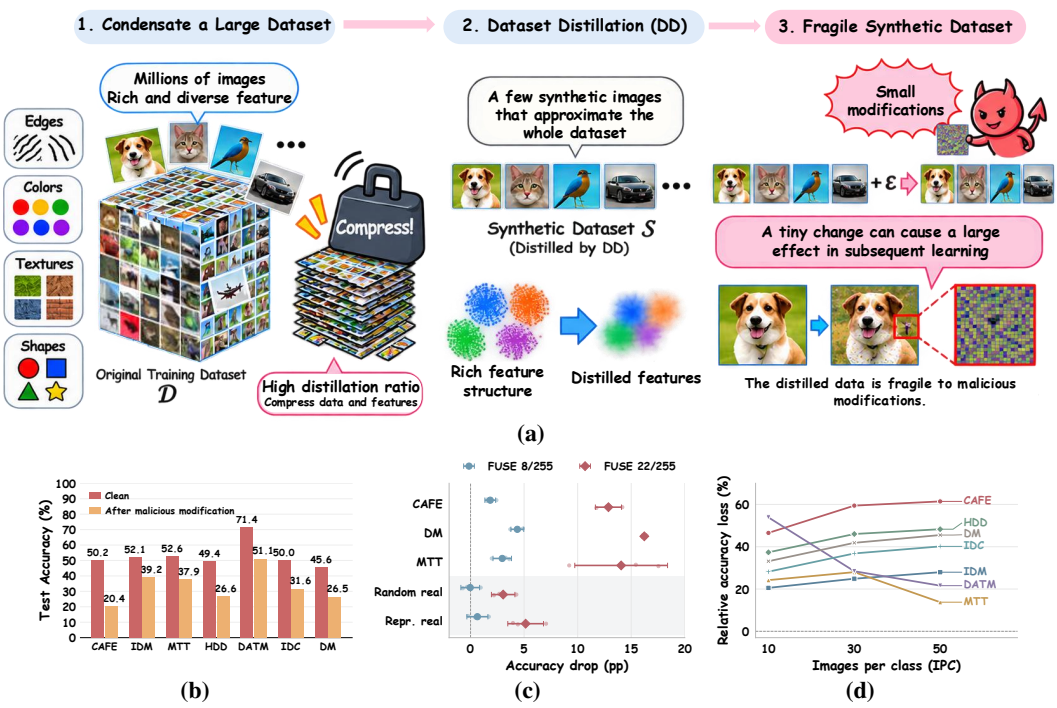


Figure 1: **Distilled training sets are vulnerable to post-distillation perturbations.** (a) Dataset distillation and adversarial perturbations happen in the delivered synthetic set. (b) Clean and mean modified-set accuracy across seven DD methods on CIFAR-10, HDD denotes DM-HDD. (c) Under identical IPC, perturbation budgets (FUSE), and downstream training, CAFE, DM, and MTT degrade more than equally sized random and representative real subsets. (d) Increasing IPC does not consistently reduce relative accuracy loss.

and accumulates along the training trajectory. Such amplification would be harmless if the synthetic set were stable under small perturbations, yet nothing in the distillation process encourages this stability. Standard DD objectives are pointwise, they evaluate the utility of the synthetic set only at the distilled solution, without constraining how the training signal it induces varies in the surrounding neighborhood. As a result, a synthetic set can match the real-data target closely while remaining highly sensitive to small adversarial perturbations. However, how to construct a condensed dataset whose general training utility remains stable under malicious post-distillation modification is still unexplored.

To bridge this gap, we introduce Native-Objective Vicinal Alignment (NOVA), a framework that augments matching-based DD objectives while producing a standard synthetic set. A key question is how to define the vicinity of a synthetic set. Different DD methods match different training signals, so the vicinity that matters is determined by each method's own objective. NOVA searches for a bounded pixel perturbation that increases the backend's native matching loss and aligns both the clean and perturbed synthetic sets with the same real-data-derived target. This regularizes the training signal in the neighborhood of the delivered set, while the clean branch preserves its nominal utility. Because NOVA only wraps the native objective, it leaves the target, the output format, and the downstream training procedure unchanged. Theoretically, we further show how a persistent change in the training signal accumulates along downstream optimization, which explains why stabilizing this signal is essential for utility that persists after delivery.

Extensive experiments show that NOVA mitigates the accuracy degradation caused by perturbations to the delivered set across feature, distribution, gradient, and trajectory matching methods, multiple datasets, and different distillation budgets. The evaluation attacks are not used during distillation, allowing us to test whether the protection transfers to unseen perturbations. Our main contributions are threefold. **(i)** We identify a delivery-time vulnerability in dataset distillation: perturbations to the delivered synthetic set substantially reduce downstream accuracy across diverse DD methods, and distilled sets are more fragile than equally sized real subsets under matched conditions. **(ii)** We show

that pointwise DD objectives leave the local sensitivity of the native training signal uncontrolled and provide empirical evidence that this missing constraint contributes to delivery-time fragility. Building on this insight, we propose NOVA, a vicinal alignment framework that aligns both the clean set and its perturbed neighbors with the same real-data target, while preserving the standard synthetic-set format and requiring no change to downstream training. **(iii)** We analyze how persistent discrepancies in the training signal propagate through downstream optimization, and validate NOVA across DD methods, datasets, and distillation budgets.

2 PRELIMINARIES

Matching-based dataset distillation. Let D be a real training set and $S = (X_S, Y_S)$ a synthetic set of far fewer samples, synthesized so that a model trained on S approaches the test performance of one trained on D. The bi-level formulation of Wang et al. (2018) is computationally prohibitive, and subsequent work replaces its inner optimization with surrogate matching objectives that share a common form. At each iteration a target signal τ_B is computed from real data, a corresponding signal Φ_B is extracted from the synthetic set, and the two are aligned under a distance $\mathcal{D}$,

$$S_B^\star = \arg\min_S \mathbb{E}_\xi \left[\mathcal{L}_B(S; D, \xi)\right], \qquad \mathcal{L}_B(S; D, \xi) = \mathcal{D}(\Phi_B(S; \xi),\ \tau_B(D; \xi)), \tag{1}$$

where ξ collects the stochastic state of the iteration, such as a network initialization, minibatches, augmentation parameters, or a segment of an expert trajectory. The families differ in what Φ_B extracts, whether parameter updates (Zhao et al., 2021; Zhao & Bilen, 2021), classwise feature statistics (Zhao & Bilen, 2023; Zhao et al., 2023), layerwise activations (Wang et al., 2022), or the parameter path of a model trained on real data (Cazenavette et al., 2022). These quantities live in different spaces and are compared under different distances, yet every instance of Eq. 1 certifies the utility of S at a single point, namely the one the optimization returns.

Perturbations on the delivered set. That point is not necessarily the set a downstream user trains on. A distilled set is produced once and then handed over, and between the two moments its pixels are a file like any other. We therefore consider an adversary who alters them after distillation has finished,

$$S \oplus \delta = (X_S + \delta,\ Y_S), \qquad \Delta_\epsilon = \{\delta : \|\delta\|_\infty \le \epsilon\}, \tag{2}$$

with ϵ specified in pixel space. The source data are clean and the distillation run is trusted. Labels, soft labels, distillation code, downstream learner, and test inputs all remain unchanged, and the user trains on $S \oplus \delta$ without knowing that it was altered. What the adversary is after is not the prediction on any particular input but the utility of the whole training process that S induces, so we measure the damage by the clean-test accuracy of a model trained from scratch on the set the user receives.

3 A CLOSE LOOK AT MATCHING-BASED DISTILLATION

3.1 THE POINTWISE OBJECTIVE LEAVES SENSITIVITY UNCONTROLLED

Consider a delivered set $S \oplus \delta$ with $\delta \in \Delta_\epsilon$ (Eq. 2). For brevity, we write $\mathcal{L}_B(S; D)$ for the native loss in Eq. 1 with the stochastic state ξ fixed. Expanding it around S to first order gives

$$\Delta\mathcal{L}_B(\delta) := \mathcal{L}_B(S \oplus \delta; D) - \mathcal{L}_B(S; D) = \epsilon\ \|\nabla_{X_S}\mathcal{L}_B(S; D)\|_1\ \rho_B(\delta; S) + O(\epsilon^2), \tag{3}$$

where $\|\nabla_{X_S}\mathcal{L}_B\|_1$ is the sensitivity of the native signal to pixel displacement, and $\rho_B(\delta; S) = \langle \nabla_{X_S}\mathcal{L}_B, \delta\rangle / (\epsilon\|\nabla_{X_S}\mathcal{L}_B\|_1) \in [-1, 1]$ is the alignment of δ with the loss gradient. The worst case $\rho_B = 1$ is attained by $\delta = \epsilon\,\mathrm{sign}(\nabla_{X_S}\mathcal{L}_B)$ (Appendix B.1). The damage thus factorizes into a budget fixed by the threat model, an alignment determined by the attacker, and a sensitivity that is a property of the returned set. This sensitivity appears nowhere in Eq. 1. Two solutions can satisfy $\mathcal{L}_B(S_1; D) \approx \mathcal{L}_B(S_2; D)$ while $\|\nabla_{X_{S_1}}\mathcal{L}_B\|_1 \ll \|\nabla_{X_{S_2}}\mathcal{L}_B\|_1$. The pointwise objective cannot distinguish them, and when their clean utilities are comparable, clean accuracy alone offers no reason to prefer the more stable one. In practice, distillation tends to land on the sensitive side: by compressing the information of many real examples into a few synthetic images, it yields sets whose native signal is markedly more sensitive than that of equally sized real subsets.

3.2 ALIGNMENT CONVERTS SENSITIVITY INTO A DISPLACED TRAINING SIGNAL

An isotropic perturbation realizes little of this sensitivity, for a random sign pattern, ρ_B has zero mean and a standard deviation that shrinks as the sensitivity spreads over more pixels (Appendix B.2). Perturbations built from class or model information instead align systematically with the sensitive directions and produce a much larger native-signal shift (Fig. 2, left). Since matching-based DD uses $\mathcal{L}_B$ as a surrogate for downstream learning behavior, this shift carries over to the downstream gradient $g(\theta; S)$, yielding a per-step discrepancy $b_t(\delta) = \|g(\theta_t; S \oplus \delta) - g(\theta_t; S)\|$. Across methods, perturbations that induce a larger native-signal shift also induce a larger downstream discrepancy as shown in Fig. 8.

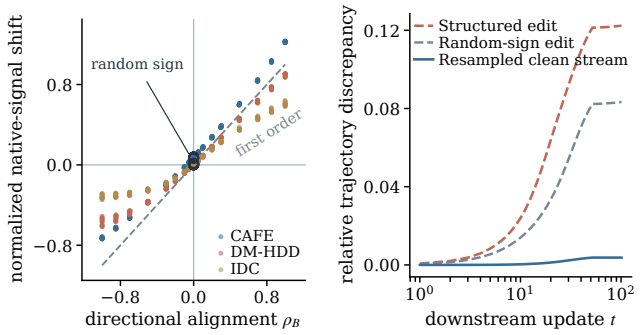


Figure 2: **Alignment and persistence.** Left: normalized native-signal shift under structured and random-sign perturbations at $\epsilon = 8/255$. Right: trajectory discrepancy of ConvNet from CAFE, comparing fixed perturbations to the delivered set with a changed stochastic training stream on clean data.

3.3 Stored Perturbations Accumulate during Training

A displaced signal would be a minor matter if the set were read once. It is not: the delivered set is the sole source of supervision, so the same δ enters every update. Let $d_t = \|\theta'_t - \theta_t\|$ be the distance between the trajectories trained on $S \oplus \delta$ and S from the same initialization. If g is L_θ-smooth in θ, then

$$d_T \leq \eta \sum_{t=0}^{T-1} (1 + \eta L_\theta)^{T-1-t} \, b_t(\delta), \tag{4}$$

so every step contributes its own discrepancy, and earlier contributions are carried forward and amplified (Appendix B.3). Empirically, fixed perturbations to the delivered pixels separate the trajectories far more than changing the stochastic training stream on clean data (Fig. 2, right).

3.4 What a Remedy Has to Control

Distillation fixes the value of $\mathcal{L}_B$ at the returned set but leaves its sensitivity unconstrained; an attacker converts this sensitivity into a displaced training signal through alignment; and downstream training reintroduces the resulting discrepancy at every step. A remedy therefore has to extend control from the returned point to the whole of Δ_ϵ. Because Eq. 3 concerns the signal a particular method extracts, this neighborhood must be controlled through $\mathcal{L}_B$ rather than through pixel similarity alone. The next section gives a tractable criterion that does so.

4 Native-Objective Vicinal Alignment

Standard DD constrains the native matching loss only at the returned set S, leaving its sensitivity uncontrolled. Yet the native objective of each method already specifies the training signal a useful set must preserve, and nothing prevents it from being enforced around S as well. NOVA builds on this observation and extends the native objective from S to its neighborhood, unlocking the robustness that the base method is capable of but never asked for.

4.1 Vicinal Alignment Objective

NOVA requires the native loss in Eq. 1 to remain small not only at S but throughout its neighborhood:

$$\min_S \; \mathbb{E}_\xi \Big[\mathcal{L}_B(S; D, \xi) + \lambda \max_{\delta \in \Delta_\epsilon} \mathcal{L}_B(S \oplus \delta; D, \xi) \Big]. \tag{5}$$

The first term is the original objective, and λ controls the weight of the neighborhood, with $\lambda = 0$ recovering the base method. By Eq. 3, the objective locally equals $(1 + \lambda)\mathcal{L}_B(S; D, \xi) + \lambda\epsilon\|\nabla_{X_S}\mathcal{L}_B(S; D, \xi)\|_1 + O(\epsilon^2)$, so NOVA penalizes exactly the sensitivity term that standard DD leaves free. **The key design of NOVA is that the clean and perturbed sets are matched to the *same* real-data target $\tau_B(D; \xi)$, rather than to each other**. Keeping the perturbed signal close to the clean one is not enough, since the two signals could agree while both drifting away from τ_B. Anchoring both to τ_B instead requires them to preserve the training signal that the base method intends to distill. In addition, the neighborhood is searched through $\mathcal{L}_B$ itself, so each DD method is protected along the directions that disrupt its own training signal, while its original objective remains intact.

4.2 Instantiating the Native Objective

NOVA only needs the native loss as a differentiable function of the synthetic pixels, which every method in Sec. 2 already provides. For the neighborhood term to measure sensitivity to the synthetic pixels alone, the clean and perturbed branches must differ only in those pixels. Within each distillation iteration t, we therefore (1) compute $\tau_B(D; \xi_t)$ once and keep it fixed during the neighborhood search, and (2) sample the stochastic state ξ_t once and reuse it for all evaluations. Since Φ_B and $\mathcal{D}$ differ across methods, the same procedure searches gradient, feature, distribution, or trajectory space depending on the base method, with no method-specific design.

4.3 Optimizing the Vicinal Objective

We optimize Eq. 5 by alternating between an inner search for a high-loss neighbor and an outer update of the synthetic set; the full procedure is given in Appendix C.

Inner maximization. With τ_B and ξ_t fixed, we find the perturbation to which the native loss reacts most strongly. Starting from $\delta^0 = 0$, we apply projected gradient ascent:

$$\delta^{k+1} = \mathrm{Proj}_{\Delta_\epsilon}\Big[\delta^k + \alpha\ \mathrm{sign}\left(\nabla_\delta \mathcal{L}_B(\mathrm{sg}(S) \oplus \delta^k; D, \xi_t)\right)\Big], \qquad k = 0, \ldots, K-1, \tag{6}$$

where $\mathrm{sg}(\cdot)$ stops gradients into the synthetic set. The search is driven only by $\mathcal{L}_B$ and involves none of the attacks used at evaluation time.

Outer minimization. Given δ^K, we update the synthetic pixels on the clean and perturbed sets together:

$$X_S \leftarrow X_S - \beta\,\nabla_{X_S}\Big[\mathcal{L}_B(S; D, \xi_t) + \lambda\,\mathcal{L}_B\big(S \oplus \mathrm{sg}(\delta^K); D, \xi_t\big)\Big]. \tag{7}$$

The first term preserves the original objective, and the second lowers the native loss at the neighbor found by the inner search. Since δ^K is detached, it serves as a fixed neighbor rather than a variable optimized jointly with S.

4.4 From Native-Signal Stability to Downstream Stability

NOVA controls the native loss within Δ_ϵ, whereas the quantity of interest is the behavior of a model trained on the delivered set. We connect the two through the accumulation analysis in Sec. 3. Because the clean and perturbed branches share the target τ_B, a triangle-type inequality for $\mathcal{D}$ bounds the gap between their native signals by their respective losses (Appendix B.5). Thus, if NOVA keeps $\mathcal{L}_B \le \gamma_B$ throughout Δ_ϵ, the native signal can shift by at most $O(\sqrt{\gamma_B})$. Since matching-based DD uses $\mathcal{L}_B$ as a surrogate for downstream training behavior, we assume that the downstream gradient discrepancy is locally controlled by the native-signal discrepancy with a method-dependent constant c_B, which gives $b_t(\delta) \le c_B\sqrt{\gamma_B}$. Substituting this into Eq. 4 yields

$$\|\theta'_T - \theta_T\| \le \eta\, c_B\sqrt{\gamma_B}\sum_{t=0}^{T-1}(1 + \eta L_\theta)^t. \tag{8}$$

The full procedure of NOVA is summarized in Algorithm 1 in Appendix C.

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

Datasets. We evaluate on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011), EuroSAT (Helber et al., 2019), and Tiny-ImageNet (Le & Yang, 2015), covering natural, digit, remote-sensing, and larger-scale images. We use IPC $\in \{10, 30, 50\}$ on the first four datasets and IPC$= 10$ on Tiny-ImageNet.

Base methods and attacks. We apply NOVA to seven DD methods spanning four matching families: feature matching (CAFE (Wang et al., 2022)), distribution matching (DM (Zhao & Bilen, 2023), IDM (Zhao et al., 2023), DM-HDD (Li et al., 2025)), gradient matching (IDC (Kim et al., 2022)), and trajectory matching (MTT (Cazenavette et al., 2022), DATM (Guo et al., 2024)). Each distilled set is attacked after distillation by five delivery-time attacks, EMN (Huang et al., 2021), GUE (Liu et al., 2024), LSP and CWN (Yu et al., 2022), and FUSE (Cai et al., 2026), all with budget $\epsilon = 8/255$. None of these attacks is used during distillation, so all results measure robustness to unseen perturbations. We report clean accuracy and the mean accuracy over attacked sets.

Baselines. The primary baseline for each method is its Vanilla version. We further compare NOVA with the robust DD methods RDC (Kim & Song, 2025), GUARD (Xue et al., 2025), and ROME (Zhou et al., 2025), which were designed for other threats and are adapted to our setting. Appendix D.1 gives the complete attack, comparison, and result-matrix protocols.

Implementation details. All main experiments use $\lambda = 1$, a NOVA training radius of $\epsilon = 8/255$, and $K = 3$ inner ascent steps with step size $2/255$. MTT is used $K = 1$ because each inner evaluation requires a 20-step trajectory unroll. For the settings of the perturbation attacks, we follow the original papers and use the default hyperparameters provided in their codebases, as detailed in Appendix D.1. We retain each DD's native synthesis and ConvNet evaluation protocol, with identical settings for Vanilla and NOVA. These hyperparameters are fixed in advance rather than tuned on the evaluation attacks, and a larger λ or ϵ yields even higher attacked accuracy. All experiments are run on NVIDIA RTX 3090 GPU without exhaustive tuning.

5.2 MAIN RESULTS

We train and evaluate a ConvNet on the distilled set S produced by each DD method, using identical hyperparameters and training protocol. Van. and NOVA denote the original set and its NOVA-augmented counterpart.

NOVA consistently mitigates the delivery-time vulnerability. Table 1 shows that these attacks lower the accuracy of Vanilla sets by 21.3 points on average, and all seven methods suffer large drops on every dataset. The vulnerability is therefore shared across matching families rather than specific to one method. NOVA improves attacked accuracy in 172 of the 175 attacked cells, by 7.4 points on average, and reduces the average gap between clean and attacked accuracy from 21.3 to 13.7 points. The gains are largest where Vanilla is most fragile: on SVHN, NOVA raises CAFE under CWN from 15.19 to 63.87 and DATM under LSP from 28.96 to 75.52, close to their clean accuracy. In the three remaining cells (IDM under FUSE on SVHN, and DATM and IDC under LSP on CIFAR-100), NOVA falls behind Vanilla by no more than 0.7 points. On clean sets, NOVA stays within 3 points of Vanilla in 30 of the 35 method–dataset pairs and improves accuracy in 13 of them, with an average change of only -0.2 points. MTT improves on all five datasets, by as much as 11.4 points on SVHN, while the largest drop is for DM on EuroSAT. The neighborhood term thus adds robustness at little cost to clean accuracy. NOVA also returns an ordinary synthetic set with the same format and number of images as Vanilla as shown in Appendix G. Table 5 reports per-IPC detailed results.

Comparison with robust DD methods. Since no existing defense targets perturbations to the delivered set, we compare NOVA with robust DD methods built for related threats, namely RDC (Kim & Song, 2025), GUARD (Xue et al., 2025), and ROME (Zhou et al., 2025), all of which aim to make downstream models robust to test-time adversarial examples. For each method, Base denotes the same method with only its protection component disabled, run with the same codebase, DD method, IPC, and evaluation protocol; for NOVA, Base is Vanilla CAFE. Comparing each defense with its own Base separates the effect of the protection mechanism from that of the underlying DD

Table 1: Average test accuracy across IPC10/30/50 on four datasets and IPC10 results on Tiny-ImageNet under diverse perturbations.

Method	Attack	CIFAR-10			CIFAR-100			SVHN			EuroSAT			Tiny-ImageNet		
		Van.	NOVA	Δ	Van.	NOVA	Δ	Van.	NOVA	Δ	Van.	NOVA	Δ	Van.	NOVA	Δ
CAFE	Clean	**48.76**	47.93	-0.83	**30.69**	28.68	-2.01	73.36	**74.45**	+1.09	**65.32**	61.79	-3.53	**11.85**	10.98	-0.87
	EMN	22.50	**43.76**	+21.26	12.35	**16.85**	+4.50	22.08	**50.60**	+28.51	33.19	**54.42**	+21.23	9.55	**10.76**	+1.21
	GUE	16.80	**34.14**	+17.34	13.14	**23.69**	+10.56	15.46	**37.70**	+22.25	32.03	**52.47**	+20.44	8.78	**10.24**	+1.46
	LSP	16.05	**35.41**	+19.36	9.04	**22.17**	+13.12	12.07	**34.85**	+22.79	19.42	**37.61**	+18.20	11.02	**11.88**	+0.86
	CWN	26.07	**45.16**	+19.08	16.16	**26.87**	+10.71	15.19	**63.87**	+48.68	26.35	**53.37**	+27.02	11.64	**11.84**	+0.20
	FUSE	24.79	**45.73**	+20.94	24.35	**28.54**	+4.19	73.32	**74.55**	+1.23	46.78	**60.46**	+13.67	7.68	**10.82**	+3.14
IDM	Clean	**51.47**	51.35	-0.12	32.55	**32.80**	+0.25	**79.64**	79.32	-0.32	**68.77**	66.64	-2.13	**16.03**	14.53	-1.50
	EMN	29.39	**30.52**	+1.13	8.10	**8.26**	+0.16	44.14	**51.23**	+7.09	35.54	**37.46**	+1.91	2.73	**4.80**	+2.07
	GUE	39.07	**40.51**	+1.44	20.04	**21.02**	+0.98	72.07	**74.26**	+2.19	38.22	**41.66**	+3.44	5.76	**12.87**	+7.11
	LSP	44.70	**45.88**	+1.18	27.46	**28.31**	+0.85	66.87	**70.64**	+3.78	41.17	**46.33**	+5.16	10.71	**13.47**	+2.76
	CWN	35.59	**37.53**	+1.94	13.04	**13.82**	+0.78	34.97	**46.78**	+11.80	30.68	**34.55**	+3.87	6.66	**12.56**	+5.90
	FUSE	45.02	**46.83**	+1.81	30.67	**31.14**	+0.46	**79.17**	78.98	-0.19	56.06	**59.39**	+3.34	4.28	**9.76**	+5.48
MTT	Clean	49.64	**58.95**	+9.32	33.20	**33.90**	+0.71	49.33	**60.76**	+11.43	69.01	**72.04**	+3.03	19.93	**20.12**	+0.19
	EMN	44.15	**55.06**	+10.91	30.91	**32.23**	+1.32	33.26	**51.16**	+17.90	43.35	**53.17**	+9.81	14.72	**18.64**	+3.92
	GUE	24.43	**25.06**	+0.63	22.60	**23.68**	+1.08	27.13	**31.09**	+3.96	41.08	**50.04**	+8.96	11.08	**16.72**	+5.64
	LSP	33.17	**43.00**	+9.84	26.13	**26.81**	+0.68	20.09	**23.59**	+3.50	23.31	**30.65**	+7.34	10.35	**13.64**	+3.29
	CWN	44.15	**54.02**	+9.87	30.14	**30.90**	+0.76	37.43	**45.98**	+8.55	34.27	**43.18**	+8.92	12.70	**13.46**	+0.76
	FUSE	46.94	**54.16**	+7.22	32.82	**33.41**	+0.59	50.37	**60.57**	+10.20	59.97	**65.74**	+5.76	8.96	**10.64**	+1.68
DM-HDD	Clean	48.32	**49.12**	+0.80	31.06	**31.37**	+0.31	**77.16**	76.85	-0.31	**67.44**	64.30	-3.14	**12.29**	11.81	-0.48
	EMN	32.60	**34.84**	+2.24	17.34	**21.36**	+4.02	42.20	**59.56**	+17.36	37.04	**41.87**	+4.83	7.04	**11.91**	+4.87
	GUE	19.64	**20.27**	+0.63	8.94	**9.89**	+0.95	18.74	**20.36**	+1.62	31.29	**36.46**	+5.17	6.55	**10.64**	+4.09
	LSP	20.83	**22.74**	+1.90	6.01	**7.81**	+1.80	12.53	**15.08**	+2.55	20.12	**22.38**	+2.26	3.24	**7.41**	+4.17
	CWN	29.93	**33.41**	+3.48	11.09	**13.04**	+1.95	18.16	**35.47**	+17.32	25.13	**30.49**	+5.36	4.57	**9.91**	+5.34
	FUSE	31.64	**34.39**	+2.75	26.14	**27.06**	+0.91	76.21	**76.70**	+0.50	44.42	**52.77**	+8.35	7.82	**8.53**	+0.71
DATM	Clean	**69.24**	65.58	-3.66	**48.57**	46.97	-1.60	78.68	**81.56**	+2.88	**80.70**	80.13	-0.57	**14.43**	10.45	-3.98
	EMN	60.18	**64.75**	+4.57	46.21	**46.62**	+0.41	65.44	**80.81**	+15.36	30.99	**39.28**	+8.29	7.28	**8.66**	+1.38
	GUE	24.29	**50.54**	+26.25	28.96	**36.42**	+7.46	35.57	**55.85**	+20.27	19.98	**26.66**	+6.68	6.83	**7.22**	+0.39
	LSP	37.17	**61.32**	+24.15	**39.09**	38.78	-0.31	28.96	**75.52**	+46.56	11.52	**25.84**	+14.32	3.51	**8.94**	+5.43
	CWN	53.44	**64.70**	+11.26	34.76	**38.53**	+3.78	44.66	**80.86**	+36.20	11.29	**36.24**	+24.95	2.76	**7.88**	+5.12
	FUSE	53.85	**63.95**	+10.10	45.38	**46.48**	+1.10	78.15	**82.41**	+4.26	33.93	**44.41**	+10.48	4.27	**6.15**	+1.88
IDC	Clean	**47.97**	45.17	-2.80	**27.80**	25.20	-2.60	**75.55**	74.47	-1.08	**68.22**	66.61	-1.61	**8.70**	7.58	-1.12
	EMN	25.02	**29.80**	+4.77	5.11	**7.96**	+2.85	27.11	**46.77**	+19.65	31.05	**33.39**	+2.33	1.89	**2.95**	+1.06
	GUE	28.84	**37.70**	+8.86	14.68	**17.33**	+2.65	63.00	**71.85**	+8.85	29.50	**36.89**	+7.40	3.25	**9.18**	+5.93
	LSP	38.62	**41.59**	+2.97	**21.61**	20.96	-0.65	59.41	**71.56**	+12.16	32.26	**43.65**	+11.39	6.78	**7.53**	+0.75
	CWN	31.84	**39.37**	+7.53	9.95	**16.83**	+6.88	24.48	**66.37**	+41.88	27.90	**38.09**	+10.19	3.88	**6.38**	+2.50
	FUSE	30.47	**38.31**	+7.84	25.38	**27.75**	+2.37	75.69	**76.57**	+0.88	38.83	**45.56**	+6.73	4.35	**8.44**	+4.09
DM	Clean	45.38	**46.70**	+1.32	**29.97**	29.70	-0.28	74.81	**74.95**	+0.14	**62.60**	57.14	-5.45	10.40	**13.38**	+2.98
	EMN	32.84	**33.72**	+0.88	16.16	**19.47**	+3.31	40.22	**59.32**	+19.10	35.94	**39.49**	+3.55	2.77	**7.99**	+5.22
	GUE	20.77	**21.21**	+0.45	8.84	**10.79**	+1.95	16.17	**21.73**	+5.56	29.21	**34.11**	+4.90	4.83	**9.61**	+4.78
	LSP	20.33	**21.85**	+1.53	6.19	**8.71**	+2.52	12.35	**16.12**	+3.77	20.26	**22.99**	+2.73	2.30	**8.54**	+6.24
	CWN	29.63	**32.24**	+2.61	10.96	**13.66**	+2.70	16.81	**38.80**	+21.99	25.37	**31.37**	+5.99	2.77	**10.73**	+7.96
	FUSE	31.46	**33.68**	+2.22	25.69	**26.65**	+0.96	73.83	**74.79**	+0.96	44.38	**51.63**	+7.25	3.72	**7.28**	+3.56

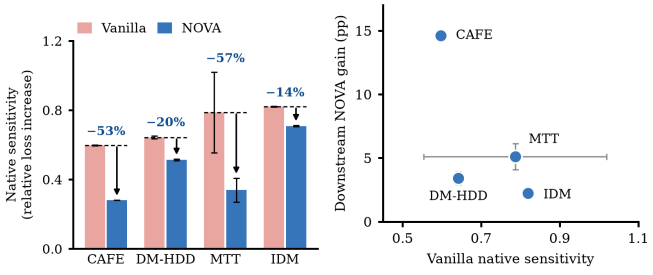
Table 2: Matched comparison with related baselines at IPC10. Results are reported as the average over EMN, GUE, LSP, CWN, and FUSE.

Defense	Variant	CIFAR-10				CIFAR-100			
		Clean	Attack Avg.	Worst	Gain	Clean	Attack Avg.	Worst	Gain
NOVA	Base	42.66	25.87	17.80	–	24.24	18.51	12.69	–
	NOVA	42.73	34.31	26.52	**+8.45**	23.08	22.88	21.49	**+4.37**
RDC	Base	11.32	9.48	7.14	–	30.12	24.39	17.28	–
	RDC	10.35	9.16	7.16	-0.32	30.74	24.86	17.54	+0.47
GUARD	Base	24.87	21.58	17.71	–	24.96	18.69	14.15	–
	GUARD	24.69	21.88	18.11	+0.30	24.57	19.86	14.77	+0.65
ROME	Base	38.85	38.89	35.65	–	17.67	17.88	16.72	–
	ROME	44.35	41.84	39.04	+2.95	18.91	18.89	18.67	+1.01

algorithm. As Table 2 shows, NOVA improves the attack average over its Base by 8.45 points on CIFAR-10 and 4.37 points on CIFAR-100, while the other defenses gain at most 2.95 and 1.01 points. The margin is larger in the worst case, where NOVA improves accuracy by 8.72 and 8.80 points and the other defenses by at most 3.39 and 1.95 points. NOVA also shrinks the clean–attacked gap from 16.79 to 8.42 points on CIFAR-10 and almost closes it on CIFAR-100 (from 5.73 to 0.20 points). ROME, by contrast, gains mainly because its clean accuracy improves, not because its gap narrows. Robustness built for other threats therefore carries over only marginally to delivery-time perturbations, and protecting the delivered set requires constraining its own training signal.

5.3 Mechanism Analysis

NOVA reduces native sensitivity, and larger reductions yield larger gains. We measure native sensitivity as the relative increase in $\mathcal{L}_B$ under the worst-case perturbation in Δ_ϵ. Fig. 3(a) shows that NOVA lowers this sensitivity for all four methods, by 57% for MTT, 53% for CAFE, 20% for DM-HDD, and 14% for IDM, which flattens the native objective around the delivered set. The size of this reduction tracks downstream robustness. MTT and CAFE, which have the largest reductions, gain 5.1 and 14.6 points, whereas DM-HDD and IDM, which have the smallest, gain only 3.4 and 2.2 points. The gain does not, however, follow the sensitivity of the Vanilla set (Fig. 3(b)): CAFE has the lowest initial sensitivity yet benefits the most. What matters is how much NOVA stabilizes the native signal, not how sensitive the set was to begin with.



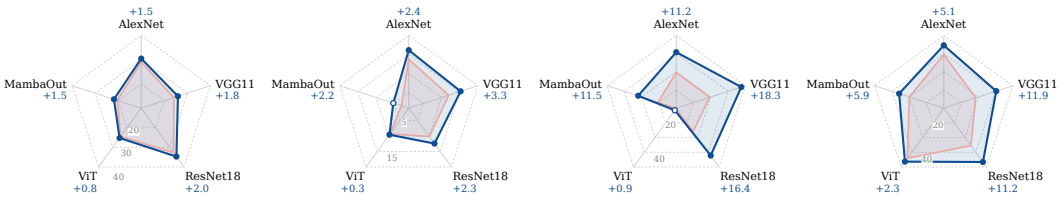
(a) (b)

Figure 3: NOVA reduces native sensitivity on every backend, and backends with a larger reduction gain more robustness. (a) Native sensitivity of Vanilla and NOVA. (b) Reduction in native sensitivity next to the downstream gain of NOVA.

5.4 Generalization to Downstream Models

Generalization across architectures. Although the delivered set is distilled with a ConvNet, downstream users may train any architecture on it. We therefore train AlexNet, VGG11, ResNet18, ViT, and MambaOut on sets distilled by CAFE, DM-HDD, IDM, and MTT at IPC$=10$, and evaluate them under EMN, LSP, and CWN. Fig. 4 shows that NOVA improves attacked accuracy on all five architectures and all four datasets. The gains are largest on SVHN and EuroSAT, which are also the datasets where Vanilla sets are most fragile in Table 1. On SVHN, NOVA improves VGG11 by 18.3 points and ResNet18 by 16.4 points, and on EuroSAT it improves both by more than 11 points. ViT benefits the least, with gains between 0.3 and 2.3 points. Because the native signal is computed with a ConvNet, the stabilized set transfers most directly to architectures with a similar convolutional inductive bias, including the gated-convolution MambaOut, and less directly to attention-based models.

Generalization across depths. Network depth is also chosen by the user rather than fixed by the delivered set. In Table 3, we train ConvNets of depth 1 to 4 on CIFAR-10 sets distilled at depth 3. The attacks degrade Vanilla at every depth, and NOVA improves the attack average for all three methods at every depth while keeping clean accuracy within 0.7 points of Vanilla. The only case where NOVA does not help is MTT at depth 1 under EMN, where both Vanilla and NOVA remain near chance because such a shallow network cannot learn from the set. The gains also hold beyond the distillation depth: at depth 4, NOVA still improves the attack average by 6.7, 2.6, and 3.3 points for CAFE, IDM, and MTT, respectively. The protection carried by the delivered set therefore transfers across the architectures and depths that downstream users are likely to choose.



(a) CIFAR10 (b) CIFAR100 (c) SVHN (d) EuroSAT

Figure 4: Cross-architectural comparison.

Table 3: Transfer across downstream ConvNet depths on CIFAR-10 at IPC10. Entries are *Vanilla* | *NOVA* test accuracy (%). Avg. is the mean over EMN, LSP, and CWN

Backend	Victim	Clean		EMN		LSP		CWN		Avg.	
CAFE	ConvNetD1	40.72	**40.78**	33.34	**38.69**	20.28	**27.22**	19.26	**24.36**	24.29	**30.09**
	ConvNetD2	**42.91**	42.56	32.45	**37.36**	19.96	**28.96**	31.10	**39.02**	27.84	**35.11**
	ConvNetD3	42.66	**42.73**	26.67	**34.80**	17.80	**26.52**	28.79	**39.98**	24.42	**33.77**
	ConvNetD4	40.58	**40.66**	32.46	**37.28**	23.25	**34.26**	35.29	**39.45**	30.33	**37.00**
IDM	ConvNetD1	42.24	**42.98**	33.39	**35.64**	40.38	**41.37**	33.86	**36.52**	35.88	**37.84**
	ConvNetD2	45.74	**46.33**	32.65	**35.46**	43.23	**44.43**	36.28	**38.86**	37.39	**39.58**
	ConvNetD3	46.82	**47.08**	31.40	**33.99**	43.20	**44.35**	37.98	**41.81**	37.53	**40.05**
	ConvNetD4	47.78	**48.09**	33.20	**36.45**	44.27	**45.53**	40.91	**44.23**	39.46	**42.07**
MTT	ConvNetD1	11.45	**12.34**	**13.31**	12.88	13.81	**14.45**	10.75	**11.62**	12.62	**12.98**
	ConvNetD2	43.87	**46.15**	40.25	**44.53**	26.80	**34.95**	36.43	**42.03**	34.49	**40.50**
	ConvNetD3	**53.81**	53.73	48.84	**51.99**	33.39	**43.33**	48.57	**52.09**	43.60	**49.14**
	ConvNetD4	**51.40**	50.70	49.28	**51.03**	38.21	**45.09**	48.40	**49.72**	45.30	**48.61**

Table 4: Controlled ablation of NOVA using CAFE on CIFAR-10 at IPC10.

Method	Clean	EMN	GUE	LSP	CWN	Attack Avg.	Worst
Vanilla CAFE	42.41 ± 0.34	24.84 ± 0.89	18.16 ± 1.00	14.89 ± 0.34	27.45 ± 0.41	21.34 ± 0.18	14.89 ± 0.34
+ Random Neighborhood	42.51 ± 0.34	27.42 ± 1.07	24.34 ± 1.19	20.13 ± 1.44	35.61 ± 1.23	26.87 ± 1.18	20.13 ± 1.44
+ Functional Stability	42.32 ± 0.17	27.54 ± 0.81	20.19 ± 0.34	15.27 ± 0.47	30.15 ± 0.72	23.29 ± 0.32	15.27 ± 0.47
+ Native Consistency	42.71 ± 0.18	34.25 ± 1.54	31.21 ± 0.06	28.79 ± 0.75	39.70 ± 0.59	33.49 ± 0.51	28.79 ± 0.75
+ NOVA ($K = 1, \alpha = 8/255$)	42.21 ± 0.37	36.36 ± 0.75	32.20 ± 0.48	28.72 ± 0.98	40.09 ± 0.29	34.34 ± 0.49	28.72 ± 0.98
+ NOVA ($K = 3, \alpha = 2/255$)	$\mathbf{42.42 \pm 0.48}$	$\mathbf{37.29 \pm 1.16}$	$\mathbf{34.49 \pm 0.85}$	$\mathbf{33.20 \pm 0.83}$	$\mathbf{40.93 \pm 0.26}$	$\mathbf{36.48 \pm 0.36}$	$\mathbf{33.20 \pm 0.83}$
+ NOVA ($K = 4, \alpha = 2/255$)	42.29 ± 0.13	36.79 ± 0.85	33.11 ± 1.35	30.07 ± 0.93	40.14 ± 0.42	35.03 ± 0.48	30.07 ± 0.93

5.5 Ablation Study

The neighbor must be searched on the native signal. Table 4 ablates the vicinal term of Eq. 5 with CAFE on CIFAR-10 at IPC$=10$, changing one component at a time. Drawing δ as random signs (Random Neighborhood) raises the attack average from 21.34 to only 26.87, since random directions rarely align with the sensitive directions of the native loss. Replacing the native signal with a one-step training map (Functional Stability) does worse, at 23.29. Native Consistency keeps the native search but only requires the perturbed loss to equal the clean one. It reaches 33.49 and lifts the worst case from 14.89 to 28.79, so most of the gain comes from stabilizing the signal that the base method distills. This constraint, however, still allows both losses to drift away from τ_B together. Anchoring to τ_B, NOVA surpasses it in attack average with a single ascent step (34.34) and performs best at $K = 3$, reaching 36.48 on average and 33.20 in the worst case.

We also provide a summary of the efficiency and hyperparameter sensitivity studies in Appendix F.2 and Appendix F.1.

6 Conclusion

This work broadens the goal of dataset distillation from utility at the distilled solution to utility that persists after delivery. We identified a delivery-time vulnerability: a bounded perturbation to the delivered pixels, degrades downstream accuracy across all matching families, and distilled sets lose more accuracy than equally sized real subsets. We traced this vulnerability to the pointwise nature of standard DD objectives, which constrain the native training signal at the returned set but leave its sensitivity in the surrounding neighborhood uncontrolled. NOVA extends the native objective to this neighborhood by searching for high-loss neighbors with the method's own loss and aligning both the clean set and these neighbors with the same real-data target. Across seven DD methods and five datasets, NOVA improves accuracy under unseen perturbations in 172 of 175 settings, while keeping clean accuracy within 3 points of the base method in 30 of 35 method–dataset pairs and leaving the synthetic-set format and downstream training unchanged.

Limitations and future work. NOVA approximates the worst-case neighbor with a few ascent steps and therefore does not certify robustness over the whole perturbation budget. Deriving certified guarantees for distilled sets is a natural next step. In addition, our evaluation considers attacks constructed without knowledge of NOVA. Studying adaptive adversaries that target the vicinal objective itself would further clarify the robustness of delivered sets.

AI USE STATEMENT

In this work, we used generative AI tools for code optimization, language polishing of the experimental and results sections, and the creation of Fig. 1(a). We have not used generative AI tools for developing the core research idea, implementing the proposed method, or writing the substantive content of the manuscript, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for improving the readability and presentation of author-written code and text. We have reviewed all AI-assisted work. AI-assisted code was manually reviewed and tested by the authors for correctness, AI-polished text was checked to ensure that the technical meaning and scientific claims were unchanged, and the AI-assisted figure was manually verified to accurately reflect the intended content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide a demo implementation in the supplementary materials, including the implementation details of our method and an executable example supporting validation and testing across multiple datasets. The main paper and appendix describe the experimental setup, evaluation protocol, datasets, attack settings, and other configurations used in our experiments. The baseline methods and dataset-distillation frameworks considered in this work are publicly available through their corresponding official repositories. The provided demo and supplementary implementation details are designed to facilitate adapting our method to these public implementations and reproducing the reported experiments.

REFERENCES

Jiale Cai, Gezheng Xu, Zhihao Li, Ruiyi Fang, Ruizhi Pu, Di Wu, Qicheng Lao, Charles Ling, and Boyu Wang. Full-spectrum unlearnable examples via spectral equalization. In *International Conference on Machine Learning*, 2026.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10718–10727, 2022.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to ImageNet-1k with constant memory. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 6565–6590, 2023. URL https://proceedings.mlr.press/v202/cui23e.html.

Haojin Deng and Yimin Yang. Context-enriched contrastive loss: Enhancing presentation of inherent sample connections in contrastive learning framework. *IEEE Transactions on Multimedia*, 27: 429–441, 2025. doi: 10.1109/TMM.2024.3521796.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rTBL8OhdhH.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2021.

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 *Proceedings of the 39th International Conference on Machine Learning*, pp. 11102–11118, 2022. URL https://proceedings.mlr.press/v162/kim22c.html.

Nicole Hee-Yeon Kim and Hwanjun Song. Robust dataset condensation using supervised contrastive learning. In *IEEE/CVF International Conference on Computer Vision*, pp. 2857–2866, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Technical report, Stanford University CS231N, 2015.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Hyperbolic dataset distillation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Shuang Liu, Yihan Wang, and Xiao-Shan Gao. Game-theoretic unlearnable example generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21349–21358, 2024.

Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Backdoor attacks against dataset distillation. In *Proceedings of the Network and Distributed System Security Symposium*. Internet Society, 2023. doi: 10.14722/ndss.2023.24287.

Noel Loo, Alaa Maalouf, Ramin Hasani, Mathias Lechner, Alexander Amini, and Daniela Rus. Large scale dataset distillation with domain shift. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32759–32780, 2024. URL https://proceedings.mlr.press/v235/loo24a.html.

Brian Moser, Federico Raue, Sebastian Palacio, Stanislav Frolov, and Andreas Dengel. Unlocking dataset distillation with diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/69413f87e5a34897cd010ca698097d0a-Abstract-Conference.html.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Cheng Shen, Yew-Soon Ong, and Joey Tianyi Zhou. CondenseLM: LLMs-driven text dataset condensation via reward matching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1237–1252, 2025. doi: 10.18653/v1/2025.emnlp-main.65. URL https://aclanthology.org/2025.emnlp-main.65/.

Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. D^4M: Dataset distillation via disentangled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5809–5818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Su_D4_Dataset_Distillation_via_Disentangled_Diffusion_Model_CVPR_2024_paper.html.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 843–852, 2017.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. CAFE: Learning to condense dataset by aligning features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12196–12205, 2022.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Eric Xue, Yijiang Li, Haoyang Liu, Peiran Wang, Yifan Shen, and Haohan Wang. Towards adversarially robust dataset distillation by curvature regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9041–9049, 2025. doi: 10.1609/aaai.v39i9.32978.

Ziyuan Yang, Ming Yan, Yi Zhang, and Joey Tianyi Zhou. Poisoned distillation: Injecting backdoors into distilled datasets without raw data access. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(2):1444–1452, 2026. doi: 10.1609/aaai.v40i2.37119.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e91fb65c6324a984ea9ef39a5b84af04-Abstract-Conference.html.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2367–2376, 2022.

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pp. 12674–12685, 2021.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6514–6523, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021.

Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7856–7865, 2023.

Tianhang Zheng and Baochun Li. RDM-DC: Poisoning resilient dataset condensation with robust distribution matching. In *Conference on Uncertainty in Artificial Intelligence*, pp. 2541–2550, 2023.

Zheng Zhou, Wenquan Feng, Qiaosheng Zhang, Shuchang Lyu, Qi Zhao, and Guangliang Cheng. ROME is forged in adversity: Robust distilled datasets via information bottleneck. In *International Conference on Machine Learning*, pp. 78687–78713, 2025.

APPENDIX: TABLE OF CONTENTS

A Related Work **14**

B Theoretical Details **14**

B.1 Derivation of the First-order Expansion 14

B.2 Alignment of Random and Structured Perturbations 15

B.3 Derivation of the Persistent-discrepancy Recursion 15

B.4 Persistence of the Discrepancy across Steps 16

B.5 From Native-objective Control to Downstream Stability 17

C NOVA Optimization Procedure **18**

D Experimental Details **18**

D.1 Evaluation Protocols 18

D.2 Fig. 1 Protocols 19

E Experimental Results **19**

E.1 Main Results 19

E.2 Attack Average across Four Datasets 20

E.3 Direction and the Small-subset Control 20

F Ablation Studies **22**

F.1 Hyperparameter Sensitivity 22

F.2 Computation Cost of NOVA 22

G Result Analysis **23**

G.1 Native-signal Drift and Downstream Degradation 23

G.2 Matched Synthetic Artifacts across Backends 23

G.3 Matched Synthetic Artifacts across Datasets 24

A Related Work

Dataset distillation and condensation. Dataset distillation (DD), also known as dataset condensation (DC), replaces a large training set with a small synthetic set that preserves its training utility (Wang et al., 2018). Existing methods can be grouped by the signal they match: training gradients (Zhao et al., 2021; Zhao & Bilen, 2021), multi-layer features or classwise distributions (Wang et al., 2022; Zhao & Bilen, 2023; Kim et al., 2022), and optimization trajectories (Cazenavette et al., 2022; Guo et al., 2024). A complementary line improves parameterization and scale. TESLA and SRe^{2}L extend condensation to ImageNet-scale data (Cui et al., 2023; Yin et al., 2023), while D3S reframes large-scale DD as domain shift (Loo et al., 2024). More recent methods exploit large pretrained generators: D^{4}M and LD3M synthesize data through diffusion models (Su et al., 2024; Moser et al., 2025), and CondenseLM uses an LLM with reward matching for text dataset condensation (Shen et al., 2025). These works primarily improve clean utility, efficiency, scale, or transfer. We instead ask whether the delivered synthetic set retains its utility after modification.

Robust distilled models and delivered-set integrity. GUARD and ROME construct synthetic data whose downstream models resist adversarially perturbed test inputs (Xue et al., 2025; Zhou et al., 2025). Although they were designed for a different intervention point, their curvature or perturbed-feature regularization can also stabilize the stored training artifact. We therefore treat them as strong cross-threat baselines and measure this transfer explicitly, rather than assuming that the difference in threat model makes them irrelevant. Their methods encode robustness through a particular surrogate: GUARD controls curvature, while ROME aligns feature distributions of clean and perturbed synthetic data. NOVA instead uses each backend's native objective both to find vulnerable neighbors and to align them with the same real-data target. This yields one construction-time adapter for feature, distribution, gradient, and trajectory matching objectives while retaining an ordinary synthetic-set interface. Other work addresses source corruption (Zheng & Li, 2023; Kim & Song, 2025) or implants targeted backdoors during distillation (Liu et al., 2023). Poisoned Distillation is closest to our attack surface because it modifies a benign set during distribution without raw-data access, but its objective is a targeted backdoor that preserves clean utility (Yang et al., 2026). We study the complementary problem: preserving general clean-test utility when the delivered synthetic pixels are maliciously modified before downstream training.

B Theoretical Details

B.1 Derivation of the First-order Expansion

We derive Eq. 3 of the main text. Let $\Phi_B(\cdot;\xi)$ be differentiable in the synthetic pixels at X_S, and let $\mathcal{D}(\cdot,\tau_B)$ be differentiable at $\Phi_B(S;\xi)$. Write $\mathcal{L}_B(S;D,\xi) = \mathcal{D}(\Phi_B(S;\xi),\tau_B(D;\xi))$ as in Eq. 1.

Expansion of the extracted signal. For $\delta \in \Delta_\epsilon$,

$$\Phi_B(S \oplus \delta;\xi) = \Phi_B(S;\xi) + J_{\Phi_B}(S)\,\delta + o(\|\delta\|), \qquad J_{\Phi_B}(S) = \frac{\partial \Phi_B(S;\xi)}{\partial X_S}, \tag{9}$$

where $J_{\Phi_B}(S)$ maps a displacement of the synthetic pixels to a displacement of the training signal. The codomain of J_{Φ_B} is the space in which the method compares signals, and differs across families. For gradient matching it is the parameter space of the network used to compute updates, for distribution and feature matching it is a space of activation statistics, and for trajectory matching it is the parameter space at the endpoint of an unrolled optimization.

Composition with the distance. Composing Eq. 9 with $\mathcal{D}$ and expanding to first order,

$$\mathcal{L}_B(S \oplus \delta; D,\xi) = \mathcal{L}_B(S;D,\xi) + \langle \nabla_{X_S}\mathcal{L}_B(S;D,\xi),\ \delta\rangle + o(\|\delta\|), \tag{10}$$

with

$$\nabla_{X_S}\mathcal{L}_B(S;D,\xi) = J_{\Phi_B}(S)^\top \nabla_\Phi \mathcal{D}(\Phi_B(S;\xi),\ \tau_B(D;\xi))\,. \tag{11}$$

Eq. 11 shows that the gradient combines two factors. The Jacobian determines how the signal moves under a pixel displacement, and the gradient of the distance determines how much a movement of the signal costs at the current position. Neither factor is constrained by Eq. 1, which fixes only the value $\mathcal{D}(\Phi_B(S),\tau_B)$.

Consequently, the clean objective does not rule out two solutions satisfying

$$\mathcal{L}_B(S_1; D) \approx \mathcal{L}_B(S_2; D), \qquad \left\|\nabla_{X_{S_1}} \mathcal{L}_B\right\|_1 \ll \left\|\nabla_{X_{S_2}} \mathcal{L}_B\right\|_1 . \tag{12}$$

This is the distinction summarized in Sec. 3.1: comparable pointwise matching losses need not imply comparable local sensitivity.

Maximization over the budget. Discarding the higher-order term and maximizing the linear form over $\Delta_\epsilon = \{\delta : \|\delta\|_\infty \le \epsilon\}$,

$$\max_{\|\delta\|_\infty \le \epsilon} \langle \nabla_{X_S} \mathcal{L}_B,\ \delta \rangle = \epsilon \left\|\nabla_{X_S} \mathcal{L}_B\right\|_1 , \tag{13}$$

which is the equality case of Hölder's inequality between the dual norms ℓ_∞ and ℓ_1, attained at $\delta^\star = \epsilon\ \mathrm{sign}(\nabla_{X_S} \mathcal{L}_B)$. Substituting into Eq. 10 gives the worst case $\rho_B = 1$ of Eq. 3.

Range of validity. The expansion is accurate while the higher-order term is small relative to $\epsilon \|\nabla_{X_S} \mathcal{L}_B\|_1$, which holds for budgets small compared to the curvature scale of $\mathcal{L}_B$ at S. The perturbation budgets considered in this work fall in this regime for the methods we examine. Beyond it, the first-order term remains a lower bound on the attainable increase whenever $\mathcal{L}_B$ is convex along the ascent direction, and the qualitative conclusion is unchanged, since the maximization in Eq. 5 is solved numerically rather than through the expansion. The expansion serves to identify the quantity that governs the increase, and is not used as a surrogate during optimization.

B.2 Alignment of Random and Structured Perturbations

Eq. 13 is attained only by a displacement aligned with $\mathrm{sign}(\nabla_{X_S} \mathcal{L}_B)$. An arbitrary $\delta \in \Delta_\epsilon$ realizes

$$\langle \nabla_{X_S} \mathcal{L}_B,\ \delta \rangle = \epsilon \left\|\nabla_{X_S} \mathcal{L}_B\right\|_1 \cdot \rho_B(\delta; S), \qquad \rho_B(\delta; S) = \frac{\langle \nabla_{X_S} \mathcal{L}_B,\ \delta \rangle}{\epsilon \left\|\nabla_{X_S} \mathcal{L}_B\right\|_1} \in [-1, 1], \tag{14}$$

so the realized increase is the maximum scaled by the alignment coefficient ρ_B.

Random displacements. Let δ have independent coordinates uniform on $\{-\epsilon, +\epsilon\}$, the vertex distribution of the budget ball, and let d denote the number of writable pixel coordinates. Then $\mathbb{E}[\rho_B] = 0$ and

$$\mathrm{Std}(\rho_B) = \frac{\|\nabla_{X_S} \mathcal{L}_B\|_2}{\|\nabla_{X_S} \mathcal{L}_B\|_1} = \frac{1}{\sqrt{d_{\text{eff}}}}, \qquad d_{\text{eff}} = \frac{\|\nabla_{X_S} \mathcal{L}_B\|_1^2}{\|\nabla_{X_S} \mathcal{L}_B\|_2^2} \in [1, d]. \tag{15}$$

When the sensitivity is spread evenly across coordinates, d_{eff} approaches d and a random displacement realizes an $O(d^{-1/2})$ fraction of the attainable increase. When it is concentrated, d_{eff} can be much smaller and random alignment correspondingly stronger. Continuous uniform or Gaussian noise of matched budget gives the same dependence with a different constant.

Structured displacements. Perturbations produced by optimizing an objective on the data or on a surrogate model are not independent across coordinates. They follow a pattern determined by class membership or by the features a network responds to, and the same pattern recurs across the samples of a class. When that pattern shares support and sign with $\nabla_{X_S} \mathcal{L}_B$ on a constant fraction of coordinates, ρ_B is bounded away from zero independently of d, and the realized increase remains a fixed fraction of the maximum as the synthetic set grows. The gap between structured and budget-matched random perturbations reported in the main text is the difference between these two regimes.

Why alignment occurs without access to $\mathcal{L}_B$. The perturbations used for evaluation are constructed without reference to the distillation objective. Alignment arises because both $\nabla_{X_S} \mathcal{L}_B$ and the perturbation are shaped by the same discriminative structure of the data. The gradient of a matching objective is largest on the pixel regions that carry class information, since those are the regions through which Φ_B responds, and perturbations built to disrupt learning concentrate on the same regions. Alignment is therefore a consequence of both quantities depending on the class structure, and does not require the adversary to know which method produced the set.

B.3 Derivation of the Persistent-discrepancy Recursion

Before comparing the two trajectories, expanding the downstream gradient in the synthetic pixels gives

$$g(\theta_t; S \oplus \delta) - g(\theta_t; S) = J_{X_S} g(\theta_t; S)\, \delta + O(\|\delta\|^2), \qquad b_t(\delta) = \|g(\theta_t; S \oplus \delta) - g(\theta_t; S)\| . \tag{16}$$

Thus the per-step discrepancy is locally governed by $J_{X_S} g$, which in general differs from $\nabla_{X_S} \mathcal{L}_B$. We now derive Eq. 4. Let $g(\theta; S)$ denote the gradient of the downstream training loss evaluated on S at parameters θ, and consider two runs with a common initialization $\theta'_0 = \theta_0$,

$$\theta_{t+1} = \theta_t - \eta\, g(\theta_t; S), \qquad \theta'_{t+1} = \theta'_t - \eta\, g(\theta'_t; S \oplus \delta). \tag{17}$$

Assumption. We assume $g(\cdot\,; S)$ is L_θ-Lipschitz in θ, which holds for the smooth losses and bounded architectures considered here and is the standard condition under which gradient descent trajectories are compared.

One-step recursion. Subtracting the two updates and inserting $\pm\,\eta\, g(\theta_t; S \oplus \delta)$,

$$\begin{aligned}\theta'_{t+1} - \theta_{t+1} = (\theta'_t - \theta_t) &- \eta\,[g(\theta'_t; S \oplus \delta) - g(\theta_t; S \oplus \delta)] \\ &- \eta\,[g(\theta_t; S \oplus \delta) - g(\theta_t; S)]\,.\end{aligned} \tag{18}$$

The first bracket compares the same training set at two different parameter values and is bounded by $L_\theta \|\theta'_t - \theta_t\|$ under the Lipschitz assumption. The second bracket compares two training sets at the same parameter value and is the discrepancy introduced by the modification. Applying the triangle inequality gives the one-step recursion

$$\left\|\theta'_{t+1} - \theta_{t+1}\right\| \le (1 + \eta L_\theta)\, \|\theta'_t - \theta_t\| + \eta\, b_t(\delta). \tag{19}$$

This is Eq. 4 with $d_t = \|\theta'_t - \theta_t\|$.

Unrolling. Let $a_t = \|\theta'_t - \theta_t\|$ and $\beta = 1 + \eta L_\theta$. From $a_0 = 0$ and $a_{t+1} \le \beta a_t + \eta b_t(\delta)$,

$$a_T \le \eta \sum_{t=0}^{T-1} \beta^{\,T-1-t}\, b_t(\delta) \;\le\; \eta \left(\max_{t<T} b_t(\delta)\right) \sum_{t=0}^{T-1} \beta^{\,t}, \tag{20}$$

where the geometric sum evaluates to $(\beta^T - 1)/(\eta L_\theta)$, so the amplification factor grows with the number of steps and with the product ηL_θ.

Reading of the bound. The bound separates two quantities. The factor $\max_t b_t(\delta)$ depends on the delivered set and on the perturbation, and is the quantity a construction-time method can influence. The geometric factor depends only on the downstream schedule, and is outside the control of whoever produces the set. A method that reduces $b_t(\delta)$ therefore attenuates the deviation at every horizon, whereas nothing about the synthetic set can reduce the amplification itself. The bound describes the divergence of the optimization path and does not by itself certify final test accuracy, since a bounded parameter deviation may correspond to different accuracy gaps depending on the loss surface.

B.4 Persistence of the Discrepancy across Steps

Eq. 20 bounds the deviation using the largest per-step discrepancy, and is loose when the $b_t(\delta)$ are small on average. Two properties of the delivery setting keep them from being small on average.

The perturbation is fixed, not resampled. Stochastic gradient noise is resampled at every step and has zero mean conditional on the parameters, so its contributions to consecutive updates are uncorrelated and the deviation it induces grows as $O(\sqrt{T})$. A perturbation written into the stored artifact is the same at every step. The discrepancy $g(\theta_t; S \oplus \delta) - g(\theta_t; S)$ is a deterministic function of θ_t and points in a direction that varies slowly along the trajectory, so consecutive contributions are correlated and the deviation accumulates linearly before the geometric factor is applied. Averaging over steps does not remove it.

The modified samples are not diluted. In a real training set of N examples, a perturbation confined to m of them contributes to a minibatch gradient in proportion to m/N, and the remaining examples supply a clean signal that dominates the update. A distilled set contains $M \ll N$ samples and the modification covers all of them, so every minibatch is drawn entirely from modified data. The clean component that would otherwise dilute the discrepancy is absent, and the same modified samples recur in every epoch.

Relation to the matched-subset comparison. These two properties are what the comparison against real subsets in the main text isolates. Matching the number of samples, the perturbation

budget, and the downstream protocol removes the dilution effect and the difference in training set size, leaving the sensitivity of the extracted signal as the remaining source of the gap. The larger degradation observed on distilled sets under these matched conditions is consistent with Eq. 11, in which the Jacobian of a mapping optimized to compress an entire corpus into few samples has no reason to be small.

B.5 From Native-objective Control to Downstream Stability

This appendix derives Eq. 8. Appendix B.3 already establishes the general trajectory recursion and its unrolled form in Eq. 20. It remains to bound the per-step discrepancy $b_t(\delta)$ by the native quantity NOVA controls. The common-target argument requires the native discrepancy to obey a triangle-type relation. We write this condition as

$$\mathcal{D}(\Phi_B(S \oplus \delta),\, \Phi_B(S)) \le \kappa_B[\mathcal{D}(\Phi_B(S \oplus \delta),\, \tau_B) + \mathcal{D}(\Phi_B(S),\, \tau_B)]\,. \tag{21}$$

Here $\kappa_B = 1$ when $\mathcal{D}$ is a metric, while $\kappa_B = 2$ covers squared Euclidean distance by applying the triangle inequality before squaring.

B.5.1 Bounding the Per-step Discrepancy

The two terms of Eq. 5 are both computed against τ_B. Suppose NOVA reaches a solution for which each is at most $\varepsilon_B/(2\kappa_B)$ over the whole neighborhood,

$$\mathcal{D}(\Phi_B(S), \tau_B) \le \frac{\varepsilon_B}{2\kappa_B}, \qquad \mathcal{D}(\Phi_B(S \oplus \delta), \tau_B) \le \frac{\varepsilon_B}{2\kappa_B} \quad \text{for all } \delta \in \Delta_\epsilon. \tag{22}$$

Requiring the second inequality for every δ rather than for the maximizer alone is not an additional assumption, since the maximizer attains the largest value. Applying Eq. 21 bounds the drift of the native signal,

$$\mathcal{D}(\Phi_B(S \oplus \delta),\, \Phi_B(S)) \le \varepsilon_B. \tag{23}$$

Relating this drift to $b_t(\delta)$ requires the native signal to reflect the downstream update it induces. We write this as the existence of a constant $c_B > 0$ with

$$\|g(\theta; S \oplus \delta) - g(\theta; S)\| \le c_B \sqrt{\mathcal{D}(\Phi_B(S \oplus \delta),\, \Phi_B(S))} \tag{24}$$

for all θ and all $\delta \in \Delta_\epsilon$. Eq. 24 holds with $c_B = 1$ when Φ_B is the downstream gradient itself and $\mathcal{D}$ is the squared Euclidean distance, which is the gradient-matching case. For the other families it states that two synthetic sets carrying the same feature statistics, layerwise activations, or trajectory endpoint also induce comparable downstream updates. This is the property under which a matching objective substitutes for training on real data, so a method for which it failed would not serve its original purpose either.

Combining Eq. 23 with Eq. 24 bounds the per-step discrepancy uniformly in t and in δ,

$$b_t(\delta) \le c_B \sqrt{\varepsilon_B}, \tag{25}$$

and substituting into Eq. 20 gives Eq. 8,

$$\|\theta'_T - \theta_T\| \le \eta\, c_B \sqrt{\varepsilon_B} \sum_{t=0}^{T-1} (1 + \eta L_\theta)^t. \tag{26}$$

B.5.2 Reading of the Bound

The bound holds uniformly over Δ_ϵ because Eq. 22 is required throughout the neighborhood, which is what the inner maximization of Eq. 5 enforces. A formulation that controlled the native loss at the clean point alone would give Eq. 25 only at $\delta = 0$ and would say nothing about the set a downstream user receives.

Two aspects of the statement are worth separating. The factor ε_B is a property of the delivered set and is the quantity Eq. 5 drives down, so it is available to whoever produces the set. The amplification factor is fixed by the downstream learning rate, smoothness, and number of steps, none of which the

producer observes. A construction-time method therefore attenuates the deviation at every horizon without being able to change how the horizon scales.

In practice the inner maximization is solved approximately by Eq. 6, so ε_B is the discrepancy attained at the returned iterate rather than the exact maximum over Δ_ϵ, and the bound is stated with respect to the neighborhood the ascent explores. The bound also tracks the divergence of the optimization path rather than the final accuracy gap. A bounded parameter deviation corresponds to different accuracy gaps depending on the loss surface, and Eq. 26 constrains the mechanism through which the damage develops rather than certifying its outcome.

C NOVA Optimization Procedure

Algorithm 1 summarizes one NOVA-augmented distillation run. The backend constructs its real-data target once per outer iteration, and the same target and stochastic state are reused by the clean branch, the neighborhood search, and the perturbed branch. Only the synthetic pixels differ between these evaluations. Any non-image variables and backend state follow the original backend update.

Algorithm 1 Native-Objective Vicinal Alignment (NOVA).

Require: Backend B, real data D, initial synthetic set S_0, outer steps T, neighborhood radius ϵ, inner steps K, ascent step α, outer step η, weight λ

for $t = 0, \dots, T-1$ **do**
  Sample the real-data batch and backend stochastic state ξ_t
  Construct and detach the backend target T_B from the real batch
  Define the native closure $\ell_B(s; T_B, \xi_t)$
  $\delta^0 \leftarrow 0$
  for $k = 0, \dots, K-1$ **do**
    $g_\delta \leftarrow \nabla_\delta \ell_B\big(\mathrm{sg}(S_t) \oplus \delta^k; T_B, \xi_t\big)$
    $\delta^{k+1} \leftarrow \mathrm{Proj}_{\Delta_\epsilon}\big(\delta^k + \alpha\ \mathrm{sign}(g_\delta)\big)$
  end for
  $L_{\mathrm{clean}} \leftarrow \ell_B(S_t; T_B, \xi_t)$
  $L_{\mathrm{vic}} \leftarrow \ell_B\big(S_t \oplus \mathrm{sg}(\delta^K); T_B, \xi_t\big)$
  $X_{S_{t+1}} \leftarrow X_{S_t} - \eta \nabla_{X_{S_t}} \left(L_{\mathrm{clean}} + \lambda L_{\mathrm{vic}}\right)$
  Update other synthetic variables and backend state using the native recipe
end for
return S_T

D Experimental Details

D.1 Evaluation Protocols

Attacks. All attacks are held out during synthesis, keep the labels unchanged, and modify only the delivered image tensor. We consider optimization-based, generator-based, and constructed class-correlated perturbations. Unless stated otherwise, attack budgets are specified as ℓ_∞ bounds in $[0, 1]$ pixel units, following Eq. 2. NOVA uses a fixed training radius of $8/255$ in all main experiments, independent of the attack budget.

Perturbation space. DD backends store synthetic images in mean/std-normalized space. We compute each perturbation δ in pixel space and convert it channel-wise before adding it to the stored tensor, $\tilde{X} = X_S + \delta/\sigma$, where σ is the per-channel standard deviation of the dataset. Thus, the pixel-space ℓ_∞ norm of the applied perturbation matches the stated budget rather than being scaled by $1/\sigma$. We do not clip or quantize the resulting tensor because synthetic images may already fall outside $[0, 1]$ in pixel space, and clipping would introduce an additional modification. We also log the applied pixel-space ℓ_∞ norm for every perturbed set.

Optimization-based. EMN (Huang et al., 2021) is optimized directly on each delivered synthetic set with $\epsilon = 8/255$. We use 20 outer rounds, each consisting of 20 victim-training steps followed by eight signed perturbation updates with step size $\epsilon/4$, with projection onto the ℓ_∞ ball after each

update. The victim model is a freshly initialized ConvNet with depth 3, width 128, instance normalization, and average pooling, matching the architecture used for downstream evaluation. EMN is therefore white-box with respect to the evaluator architecture, but it does not share the evaluator's weights or initialization. For IDC, EMN optimizes the stored tensor while backpropagating through the IDC decoding operation.

Generator-based. GUE (Liu et al., 2024) applies its generator independently to each image, with the output scaled to $\ell_\infty = 64/255$. FUSE (Cai et al., 2026) instead averages generator outputs within each class. Its nominal scale is $8/255$, but the generator ends with a $\tanh$ layer followed by a fixed $2.7\times$ scaling factor, resulting in a measured ℓ_∞ norm of $21.6/255$, reported as $22/255$ in the figures. Both attacks use a single generator forward pass and do not re-optimize the generator on the synthetic set. For CIFAR-10, we use the released generators. For CIFAR-100, SVHN, EuroSAT, and Tiny-ImageNet, we train dataset-specific generators using the released implementations and the real training split of each dataset, without using any synthetic data. Generator inputs are clamped to $[0, 1]$ to match their training distribution. This clamping is applied only to the generator input; the resulting perturbation is added to the unclipped synthetic tensor as described above.

Constructed. LSP (Yu et al., 2022) constructs tiled, label-aligned, linearly separable patch features. The original method constrains the patch perturbation in ℓ_2. For consistency with the other attacks, we rescale the patch values so that the maximum absolute pixel change is exactly $32/255$, yielding an ℓ_∞-bounded variant. CWN assigns a fixed random-sign pattern of magnitude $32/255$ to each class and is therefore ℓ_∞-bounded by construction.

The reported attack average assigns equal weight to all attacks evaluated on a dataset. Independent per-image random-sign noise at $8/255$ is included as an unstructured control but is excluded from this average.

Comparisons. Each Vanilla/NOVA pair uses the same DD backend, synthesis seed, and downstream protocol. Random Neighborhood controls for additional neighborhood samples; Functional Stability replaces the native target with a one-step training map; Native Consistency applies a generic clean and perturbed consistency penalty; and One-step Native Ascent controls for using a single adversarial direction. RDC, GUARD, and ROME address different threats and are therefore treated as cross-threat references. In particular, ROME retains its required decoder, so we interpret only its change from its matched parent rather than comparing its absolute accuracy with NOVA.

Result matrix. The matrix contains paired clean and attacked test accuracy for every available backend, dataset, IPC and attack cell. Differences are computed only within a matched Vanilla/NOVA pair; unavailable combinations are left missing rather than treated as zeros.

D.2 Fig. 1 Protocols

Fig. 1(b) reports CIFAR-10 IPC30 results from three synthesis seeds and averages the five structured modifications used in the main evaluation. Panel (c) compares 100-image CAFE, DM, and MTT sets with random and teacher-feature K-means real subsets. All sets use the same ConvNet initialization and 101-epoch training recipe. FUSE is evaluated at realized budgets of $8/255$ and $21.6/255$ (displayed as $22/255$); points denote three set-construction seeds and intervals denote sample standard deviations. Panel (d) reports relative accuracy loss on CIFAR-10 for every complete method–IPC pair in the results table, averaging EMN, GUE, LSP, CWN, and FUSE at their stated budgets. Lines connect measured IPC settings and do not imply monotonic behavior between them.

E Experimental Results

E.1 Main Results

Table 5 supplements Table 1 with the individual IPC10, IPC30, and IPC50 results used to form the main-table averages.

Table 5: IPC-resolved test accuracy (%) after delivery-stage attacks. Each cell reports *Vanilla* | *NOVA*; bold identifies the better member of each pair. The source data remain clean and attacks modify only the delivered synthetic images. All reported pairs average three synthesis seeds.

Method	Attack	CIFAR-10			CIFAR-100			SVHN			EuroSAT			Tiny-ImageNet
		10	30	50	10	30	50	10	30	50	10	30	50	10
CAFE	Clean	42.41 | 42.41	**50.18** | 49.89	**53.69** | 51.50	**24.24** | 23.08	**32.35** | 30.43	**35.48** | 32.54	63.44 | **66.68**	77.05 | **77.62**	**79.59** | 79.05	**55.79** | 54.70	**68.01** | 64.02	**72.17** | 66.66	**11.85** | 10.98
	EMN	24.84 | **37.11**	20.61 | **45.21**	22.05 | **48.95**	20.12 | **23.42**	8.99 | **15.16**	7.93 | **11.97**	20.99 | **27.98**	21.30 | **61.72**	23.96 | **62.09**	34.64 | **45.82**	32.56 | **58.48**	32.37 | **58.95**	9.55 | **10.76**
	GUE	18.16 | **34.49**	16.42 | **34.93**	15.82 | **32.99**	13.70 | **21.49**	13.08 | **24.46**	12.63 | **25.13**	16.62 | **31.65**	13.39 | **40.09**	16.36 | **41.37**	32.13 | **46.26**	31.82 | **55.04**	32.15 | **56.11**	8.78 | **10.24**
	LSP	14.89 | **33.20**	16.93 | **36.83**	16.32 | **36.19**	12.69 | **23.09**	8.12 | **24.79**	6.32 | **18.62**	11.01 | **17.33**	12.35 | **41.81**	12.84 | **45.42**	21.48 | **31.80**	18.55 | **40.25**	18.22 | **40.79**	11.02 | **11.88**
	CWN	27.45 | **40.93**	24.88 | **46.53**	25.89 | **48.01**	22.27 | **23.21**	14.71 | **29.31**	11.51 | **28.09**	16.83 | **52.18**	13.91 | **68.72**	14.84 | **70.72**	31.06 | **49.99**	25.74 | **55.00**	22.25 | **55.13**	11.64 | **11.84**
	FUSE	28.02 | **40.68**	22.92 | **47.35**	23.44 | **49.16**	**23.75** | 23.17	**30.55** | 30.05	18.75 | **32.40**	64.01 | **66.96**	77.19 | **77.70**	78.77 | **79.00**	50.33 | **53.67**	53.82 | **62.84**	36.20 | **64.86**	7.68 | **10.82**
IDM	Clean	46.27 | **47.07**	**52.13** | 51.88	**56.00** | 55.10	27.43 | **28.40**	**33.84** | 33.73	**36.37** | 36.27	76.34 | **76.45**	**79.85** | 79.77	**82.74** | 81.75	**62.85** | 59.90	**70.21** | 68.25	**73.25** | 71.76	**16.03** | 14.53
	EMN	30.22 | **32.40**	29.25 | **30.27**	28.70 | **28.89**	7.80 | **8.11**	8.06 | **8.26**	**8.44** | 8.41	45.64 | **55.15**	44.78 | **49.95**	41.99 | **48.58**	35.82 | **38.72**	35.73 | **37.84**	35.08 | **35.81**	2.73 | **4.80**
	GUE	34.36 | **36.28**	39.59 | **40.96**	43.25 | **44.28**	16.43 | **17.60**	20.85 | **21.63**	22.85 | **23.84**	62.56 | **68.03**	75.30 | **75.88**	78.36 | **78.88**	32.73 | **37.47**	38.97 | **42.00**	42.96 | **45.52**	5.76 | **12.87**
	LSP	42.24 | **43.80**	45.92 | **46.53**	45.95 | **47.31**	23.67 | **25.25**	28.15 | **28.56**	30.57 | **31.13**	64.92 | **70.08**	70.90 | **73.14**	64.78 | **68.71**	37.21 | **41.73**	41.21 | **48.74**	45.08 | **48.52**	10.71 | **13.47**
	CWN	35.91 | **39.11**	35.49 | **37.25**	35.38 | **36.24**	15.94 | **17.33**	12.34 | **12.79**	10.85 | **11.35**	37.79 | **54.03**	35.03 | **45.45**	32.10 | **40.85**	30.83 | **36.25**	30.44 | **34.39**	30.77 | **33.02**	6.66 | **12.56**
	FUSE	41.08 | **43.30**	45.55 | **47.50**	48.44 | **49.69**	26.04 | **27.01**	31.81 | **32.12**	34.17 | **34.28**	75.51 | **75.96**	**79.76** | 79.38	**82.25** | 81.60	51.88 | **52.81**	57.46 | **62.26**	58.83 | **63.11**	4.28 | **9.76**
MTT	Clean	49.95 | **51.13**	52.61 | **61.63**	46.35 | **64.10**	**31.77** | 30.84	33.55 | **35.00**	34.27 | **35.87**	36.21 | **42.43**	48.75 | **70.49**	63.03 | **69.35**	57.63 | **65.49**	**71.03** | 70.41	78.36 | **80.22**	19.93 | **20.12**
	EMN	45.71 | **49.33**	44.96 | **55.17**	41.79 | **60.69**	**29.62** | 29.31	31.64 | **32.80**	31.48 | **34.58**	28.07 | **37.98**	32.62 | **59.01**	39.08 | **56.48**	41.55 | **49.38**	46.64 | **55.40**	41.87 | **54.72**	14.72 | **18.64**
	GUE	23.33 | **28.92**	**23.49** | 20.12	**26.48** | 26.14	20.03 | **20.07**	23.59 | **24.39**	24.18 | **26.58**	20.25 | **24.15**	26.11 | **31.18**	35.04 | **37.94**	30.23 | **36.31**	50.38 | **55.03**	42.64 | **58.78**	11.08 | **16.72**
	LSP	31.68 | **39.71**	32.03 | **37.16**	35.79 | **52.14**	25.46 | **25.67**	26.68 | **27.70**	26.25 | **27.07**	17.14 | **21.39**	21.77 | **25.35**	21.36 | **24.04**	21.69 | **22.58**	26.50 | **32.49**	21.75 | **36.89**	10.35 | **13.64**
	CWN	45.30 | **48.78**	44.34 | **53.67**	42.80 | **59.61**	**29.37** | 28.65	30.30 | **31.65**	30.75 | **32.39**	29.50 | **38.50**	38.72 | **50.97**	44.06 | **48.47**	31.16 | **34.39**	40.44 | **48.32**	31.20 | **46.84**	12.70 | **13.46**
	FUSE	43.26 | **48.06**	44.60 | **53.86**	52.96 | **60.56**	**31.46** | 30.66	33.49 | **34.32**	33.51 | **35.24**	36.29 | **42.04**	50.38 | **70.70**	64.45 | **68.98**	50.05 | **59.54**	65.51 | **67.40**	64.36 | **70.27**	8.96 | **10.64**
DM-HDD	Clean	42.99 | **43.88**	49.36 | **50.16**	52.62 | **53.33**	24.93 | **25.01**	32.39 | **32.79**	35.87 | **36.31**	**72.97** | 72.05	**78.53** | 78.33	79.97 | **80.16**	**59.13** | 56.07	**69.85** | 66.52	**73.34** | 70.32	**12.29** | 11.81
	EMN	34.15 | **37.40**	31.78 | **33.81**	31.88 | **33.31**	18.84 | **24.72**	17.20 | **21.16**	15.98 | **18.21**	41.51 | **62.20**	42.79 | **60.59**	42.31 | **55.90**	37.44 | **43.47**	36.72 | **41.77**	36.95 | **40.36**	7.04 | **11.91**
	GUE	19.34 | **21.04**	19.38 | **19.61**	**20.20** | 20.16	7.40 | **8.77**	9.54 | **10.47**	9.88 | **10.42**	16.94 | **18.99**	19.23 | **20.07**	20.05 | **22.03**	26.39 | **30.99**	32.09 | **38.70**	35.40 | **39.70**	6.55 | **10.64**
	LSP	20.03 | **22.87**	21.50 | **23.21**	20.97 | **22.13**	5.83 | **8.43**	6.05 | **7.88**	6.15 | **7.11**	11.66 | **14.90**	13.60 | **16.30**	12.33 | **14.05**	20.09 | **21.82**	19.01 | **22.07**	21.26 | **23.24**	3.24 | **7.41**
	CWN	30.62 | **35.57**	28.92 | **32.54**	30.25 | **32.13**	13.16 | **18.10**	10.68 | **11.51**	9.44 | **9.52**	16.28 | **39.45**	18.53 | **34.36**	19.66 | **32.61**	26.25 | **32.12**	23.66 | **30.02**	25.47 | **29.33**	4.57 | **9.91**
	FUSE	30.52 | **34.88**	31.64 | **34.53**	32.75 | **33.75**	22.81 | **23.39**	26.85 | **28.09**	28.77 | **29.69**	71.10 | **71.77**	77.74 | **78.45**	79.78 | **79.89**	43.36 | **48.34**	42.16 | **53.90**	47.73 | **56.07**	7.82 | **8.53**
DATM	Clean	**63.92** | 60.83	**71.38** | 67.35	**72.41** | 68.55	**44.84** | 43.23	**49.84** | 46.94	**51.04** | 50.74	68.68 | **78.24**	**83.87** | 83.05	**83.49** | 83.38	**75.26** | 74.54	**82.46** | 82.34	**84.39** | 83.51	**14.43** | 10.45
	EMN	49.93 | **59.28**	63.48 | **66.93**	67.12 | **68.03**	43.28 | **43.93**	**47.42** | 47.24	47.94 | **48.69**	45.47 | **76.97**	75.56 | **82.76**	75.30 | **82.69**	32.93 | **43.67**	30.75 | **39.77**	29.29 | **34.39**	7.28 | **8.66**
	GUE	16.20 | **39.98**	25.33 | **52.94**	31.33 | **58.69**	24.02 | **37.78**	30.55 | **39.64**	**32.32** | 31.84	16.98 | **43.51**	39.90 | **60.64**	49.84 | **63.39**	15.75 | **28.10**	20.41 | **27.01**	23.78 | **24.87**	6.83 | **7.22**
	LSP	14.10 | **53.16**	44.19 | **64.88**	53.23 | **65.93**	36.78 | **40.47**	40.10 | **42.04**	**40.39** | 33.83	11.05 | **66.55**	34.86 | **80.14**	40.97 | **79.88**	11.99 | **28.83**	10.29 | **24.59**	12.27 | **24.09**	3.51 | **8.94**
	CWN	33.85 | **59.07**	61.70 | **66.91**	64.77 | **68.13**	40.69 | **40.71**	36.21 | **41.80**	27.37 | **33.09**	22.44 | **76.94**	54.04 | **82.70**	57.51 | **82.94**	11.80 | **40.12**	11.02 | **36.16**	11.05 | **32.45**	2.76 | **7.88**
	FUSE	33.35 | **57.60**	60.79 | **66.31**	67.40 | **67.93**	42.99 | **43.25**	46.27 | **47.24**	46.87 | **48.94**	67.45 | **77.47**	83.32 | **84.81**	83.68 | **84.95**	25.91 | **47.12**	37.11 | **44.59**	38.76 | **41.51**	4.27 | **6.15**
IDC	Clean	**42.64** | 39.30	**50.04** | 47.27	**51.22** | 48.93	**24.58** | 20.64	**29.65** | 26.37	**29.17** | 28.59	**72.73** | 71.60	**76.47** | 75.46	**77.46** | 76.36	**59.92** | 58.18	**71.21** | 69.84	**73.53** | 71.81	**8.70** | 7.58
	EMN	25.85 | **29.86**	24.63 | **29.80**	24.59 | **29.73**	5.94 | **8.75**	4.90 | **8.07**	4.50 | **7.06**	24.73 | **40.32**	27.98 | **50.26**	28.63 | **49.72**	31.83 | **32.72**	30.56 | **34.31**	30.77 | **33.13**	1.89 | **2.95**
	GUE	28.38 | **35.33**	29.85 | **39.15**	28.29 | **38.63**	13.80 | **16.31**	15.72 | **18.66**	14.53 | **17.02**	56.00 | **67.02**	66.69 | **74.02**	66.31 | **74.52**	30.52 | **35.58**	28.75 | **37.96**	29.22 | **37.14**	3.25 | **9.18**
	LSP	36.96 | **37.61**	40.07 | **43.31**	38.82 | **43.84**	**19.86** | 19.85	**23.08** | 21.96	**21.90** | 21.08	57.67 | **66.95**	64.84 | **74.35**	55.71 | **73.39**	30.83 | **39.04**	33.55 | **45.02**	32.40 | **46.89**	6.78 | **7.53**
	CWN	33.18 | **38.00**	31.59 | **41.39**	30.74 | **38.71**	13.64 | **18.49**	9.63 | **18.29**	6.59 | **13.72**	27.51 | **63.74**	22.91 | **68.81**	23.03 | **66.55**	29.70 | **37.91**	27.64 | **39.53**	26.35 | **36.82**	3.88 | **6.38**
	FUSE	28.70 | **35.13**	31.91 | **40.95**	30.81 | **38.85**	22.33 | **23.17**	27.17 | **29.91**	26.64 | **30.18**	72.79 | **73.74**	76.64 | **77.53**	77.64 | **78.44**	34.50 | **42.55**	42.90 | **46.25**	39.10 | **47.89**	4.35 | **8.44**
DM	Clean	41.90 | **42.41**	45.57 | **47.39**	48.66 | **50.30**	**23.17** | 22.97	**31.75** | 31.44	**35.00** | 34.68	**68.95** | 68.66	76.81 | **77.04**	78.68 | **79.16**	**54.57** | 48.86	**63.35** | 59.21	**69.87** | 63.36	10.40 | **13.38**
	EMN	36.25 | **36.94**	31.56 | **32.49**	30.72 | **31.73**	16.43 | **20.31**	16.62 | **20.40**	15.43 | **17.71**	38.97 | **61.98**	40.70 | **61.58**	40.98 | **54.40**	35.59 | **41.79**	35.16 | **38.03**	37.07 | **38.65**	2.77 | **7.99**
	GUE	20.93 | **22.10**	20.31 | **20.55**	**21.06** | 20.99	7.37 | **10.55**	9.44 | **11.13**	9.72 | **10.69**	13.31 | **21.69**	17.60 | **22.36**	17.59 | **21.14**	25.23 | **29.73**	28.59 | **33.74**	33.81 | **38.86**	4.83 | **9.61**
	LSP	20.51 | **22.52**	20.46 | **22.28**	20.01 | **20.76**	6.15 | **9.87**	6.13 | **8.52**	6.28 | **7.73**	11.27 | **16.70**	13.60 | **17.23**	12.18 | **14.43**	19.94 | **23.54**	19.67 | **21.54**	21.16 | **23.89**	2.30 | **8.54**
	CWN	31.15 | **35.17**	28.29 | **31.01**	29.44 | **30.53**	12.78 | **18.94**	10.60 | **12.00**	9.50 | **10.04**	15.17 | **46.69**	17.53 | **35.61**	17.72 | **34.09**	25.68 | **33.66**	24.94 | **30.17**	25.50 | **30.27**	2.77 | **10.73**
	FUSE	31.21 | **34.30**	31.87 | **33.46**	31.31 | **33.29**	21.62 | **22.19**	27.11 | **28.46**	28.33 | **29.30**	68.03 | **69.23**	76.10 | **76.70**	77.36 | **78.44**	38.81 | **45.80**	46.21 | **53.57**	48.11 | **55.51**	3.72 | **7.28**

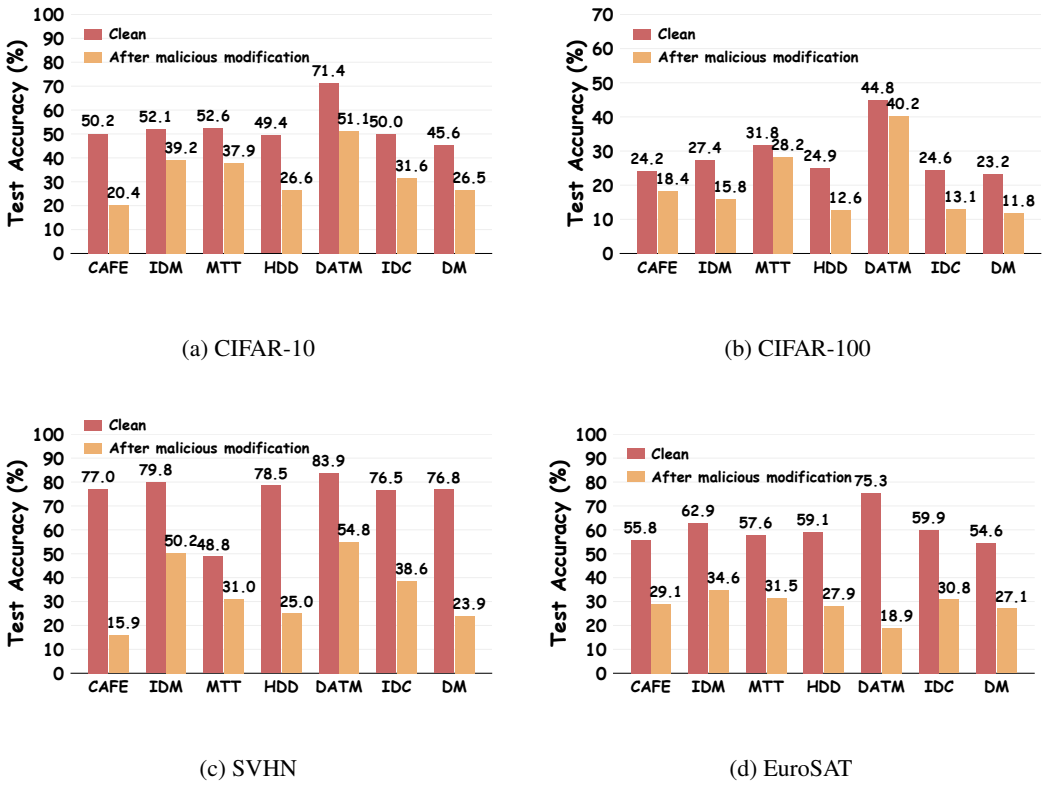
E.2 Attack Average across Four Datasets

Fig. 5 summarizes the clean-to-modified accuracy gap across datasets at a representative measured IPC. Every orange bar is an equal-weight mean over the complete structured-attack suite available for that dataset; missing cells are left missing rather than set to zero. The figure shows that the delivery-stage failure is not confined to CIFAR-10, while the variation across methods and datasets cautions against a universal magnitude claim.

Fig. 6 gives the corresponding absolute-accuracy view across CIFAR-10 IPC10, IPC30, and IPC50. Increasing IPC improves clean utility for most methods but does not guarantee that the modified-set curve approaches the clean curve. Lines only connect measured settings and do not imply monotonic behavior between them.

E.3 Direction and the Small-subset Control

Fig. 7 asks whether fragility follows only from training on few samples. It compares 100-image CAFE, DM, and MTT sets with class-balanced random and teacher-feature representative real subsets. All conditions share the same ConvNet initialization and 101-epoch consumer recipe. At the realized FUSE budget of $22/255$, the tested distilled sets lose more accuracy than the equally sized real subsets, whereas random-sign noise has a much smaller effect. This controlled result supports sensitivity beyond sample count in the tested setting, but does not identify distillation as the sole cause because clean utility and tensor amplitude still differ.



(a) CIFAR-10

(b) CIFAR-100

(c) SVHN

(d) EuroSAT

Figure 5: Vanilla DD accuracy before and after malicious modification across datasets.

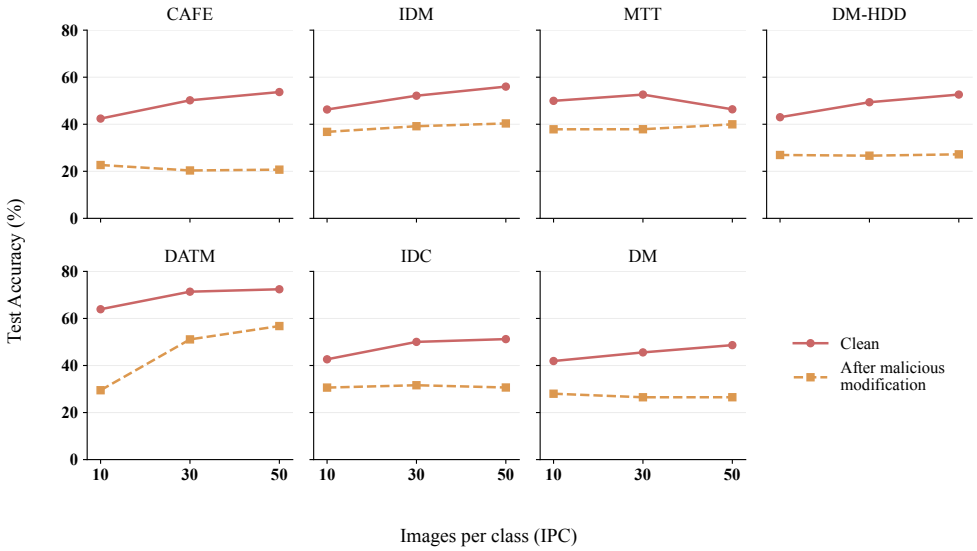


Figure 6: **Absolute accuracy across IPC for Vanilla DD on CIFAR-10.** The modified-set curve is the equal-weight mean of EMN, GUE, LSP, CWN, and FUSE at their stated budgets. A point is shown only when all five attack results are available. Some IPC10 DATM, IDC, and DM clean points use one synthesis seed; the full source and seed counts accompany the figure data.

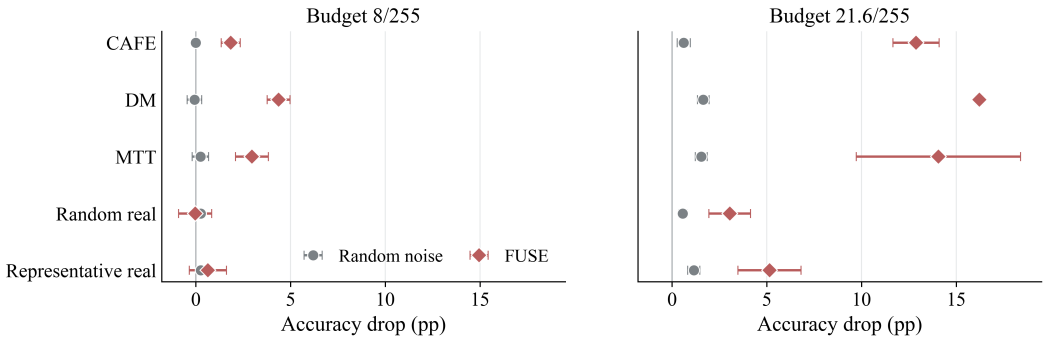


Figure 7: **Distilled sets versus equally small real subsets.** CIFAR-10 IPC10 (100 images per set) under matched training and actual perturbation budgets. Points and intervals show means and sample standard deviations across three construction seeds; faint points are individual seeds. The same 10,000 clean test images are used throughout.

F Ablation Studies

F.1 Hyperparameter Sensitivity

Table 6 varies the robust loss weight and inner radius without changing the evaluation protocol. A larger weight improves all three attacked accuracies without reducing clean accuracy at the reported precision. A larger radius gives the strongest attacked average, with a small clean-accuracy cost. We retain the pre-specified $\lambda = 1$, $\epsilon = 8/255$ setting in the main comparison rather than selecting a setting on the held-out attacks.

Table 6: NOVA sensitivity on CAFE, CIFAR-10, IPC10. Values are absolute accuracy (%) averaged over three synthesis seeds.

λ	ϵ	Clean	EMN	LSP	CWN	Attack Avg.
1	$8/255$	42.73	34.80	26.52	39.98	33.77
2	$8/255$	**42.74**	37.88	31.51	40.75	36.71
1	$16/255$	41.93	**39.95**	**35.30**	**41.06**	**38.77**

F.2 Computation Cost of NOVA

NOVA modifies only the distillation stage. The delivered synthetic set retains the format produced by its backend, and downstream users train on it with the original pipeline, so NOVA introduces no additional downstream cost. Its extra distillation cost comes from the inner search and the vicinal term. With the default $K = 3$ solver, each outer update requires four additional evaluations of the backend's native loss. The resulting overhead therefore depends on the fraction of a distillation iteration spent evaluating that loss.

Table 7 reports the synthesis wall-clock time of NOVA relative to its matched Vanilla backend. For each Vanilla–NOVA pair, we use the same code, dataset, IPC, seed, and number of distillation iterations on an NVIDIA RTX 3090 GPU. Runtime is measured from the synthesis logs over the same iteration range, and we report the average per-pair ratio for each backend.

Table 7: **Synthesis cost of NOVA relative to Vanilla.** Time is the mean synthesis wall-clock time in minutes. Mean is the average of NOVA's time divided by that of the matched Vanilla run. Downstream training cost is unchanged for every method.

Method	Matching	Time (min)		Mean
		Vanilla	NOVA	
CAFE	Feature	25.1	29.3	1.14×
DM	Distribution	19.0	29.5	1.54×
DM-HDD	Distribution	1.1	2.4	2.26×
DATM	Trajectory	335.3	961.0	2.87×

G Result Analysis

G.1 Native-signal Drift and Downstream Degradation

Fig. 8 directly tests the empirical link between the native quantity controlled during construction and the damage observed after downstream training. Each point is a matched backend–dataset–IPC cell and reports the change from Vanilla to NOVA. Because native objectives have different numerical scales, the horizontal axis uses the within-cell relative change rather than comparing their absolute values across backends. The two changes have the same sign in all 10 matched CAFE and IDC cells: NOVA reduces both native-signal drift and downstream accuracy loss in every case. Each backend contributes the same five dataset–IPC settings. Their rank association is $\rho_s = 0.79$ ($p = 0.006$). The result provides paired evidence that native-signal stabilization accompanies lower downstream degradation in the evaluated settings; it does not assert that absolute native losses are comparable across matching families.

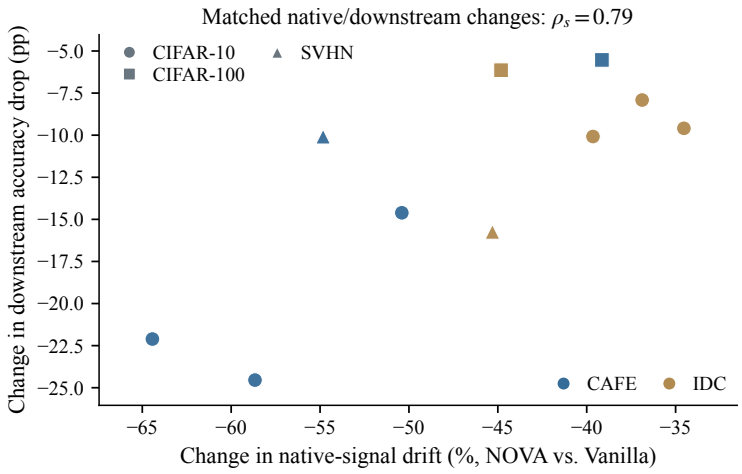


Figure 8: **Native-signal drift and downstream degradation.** Changes from Vanilla to NOVA over the five dataset–IPC settings shared by CAFE and IDC, yielding five points per backend with matched native and downstream measurements. Native drift is the worst-case relative drift found at $\epsilon = 8/255$; downstream degradation is the clean-to-attacked accuracy drop averaged over applicable held-out attacks. All 10 cells lie in the lower-left quadrant.

G.2 Matched Synthetic Artifacts across Backends

Fig. 9 shows actual matched IPC10 artifacts before and after adding NOVA to all seven main backends: CAFE, DM, IDM, DM-HDD, IDC, MTT, and DATM. The methods use different synthetic parameterizations, so visual appearance is not comparable across rows. Within each row, however, the artifact format and number of stored images are unchanged. This confirms that NOVA returns an ordinary synthetic set and does not require an auxiliary decoder at use time.

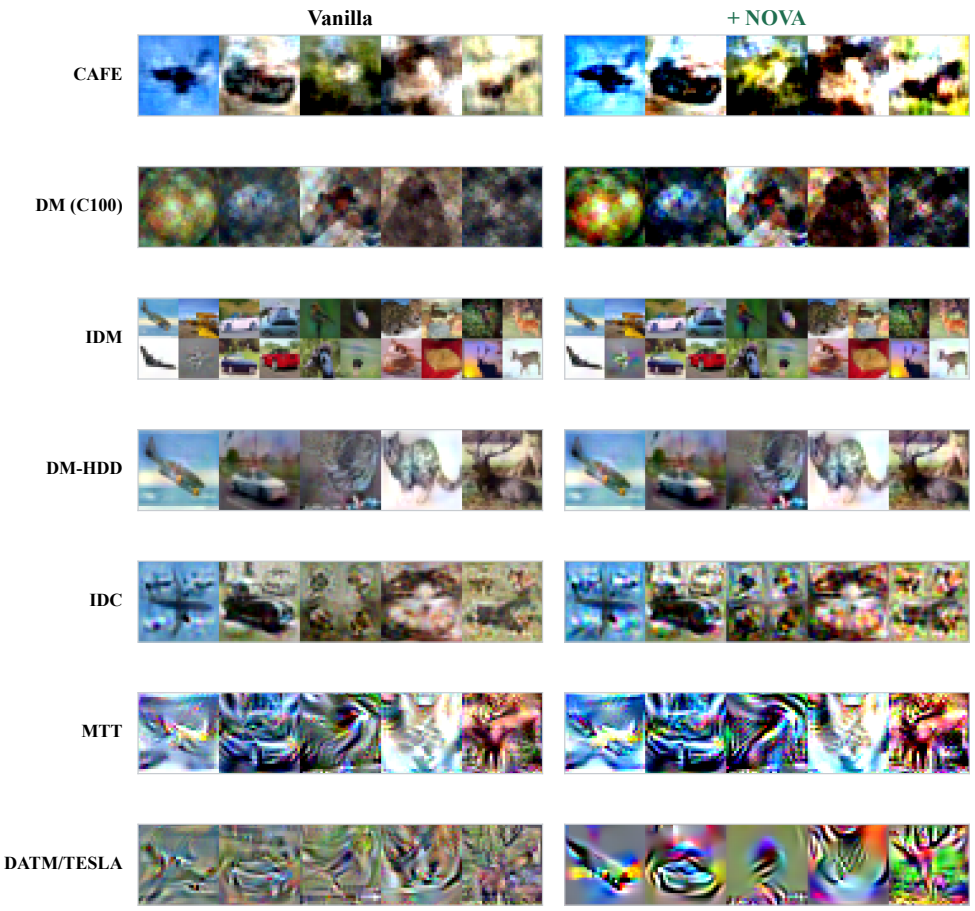


Figure 9: **Actual synthetic artifacts before and after NOVA.** Five stored examples from distinct classes are shown for CAFE, DM, IDM, DM-HDD, IDC, MTT, and DATM. The DM row uses CIFAR-100 IPC10; all other rows use CIFAR-10 IPC10. No delivery-stage perturbation, per-image contrast normalization, or display amplification is applied.

G.3 Matched Synthetic Artifacts across Datasets

Fig. 10 complements the seven-backend view with matched CAFE artifacts on CIFAR-10, CIFAR-100, and SVHN. All panels use IPC10 and synthesis seed 0. The CIFAR-100 panels show the same eight selected classes in both arms; the other panels show the full 10×10 stored grids.

Fig. 11 adds a second cross-dataset view using DM. Together, the two dataset galleries cover CIFAR-10, CIFAR-100, SVHN, and EuroSAT with two distinct matching families where matched checkpoints are available.

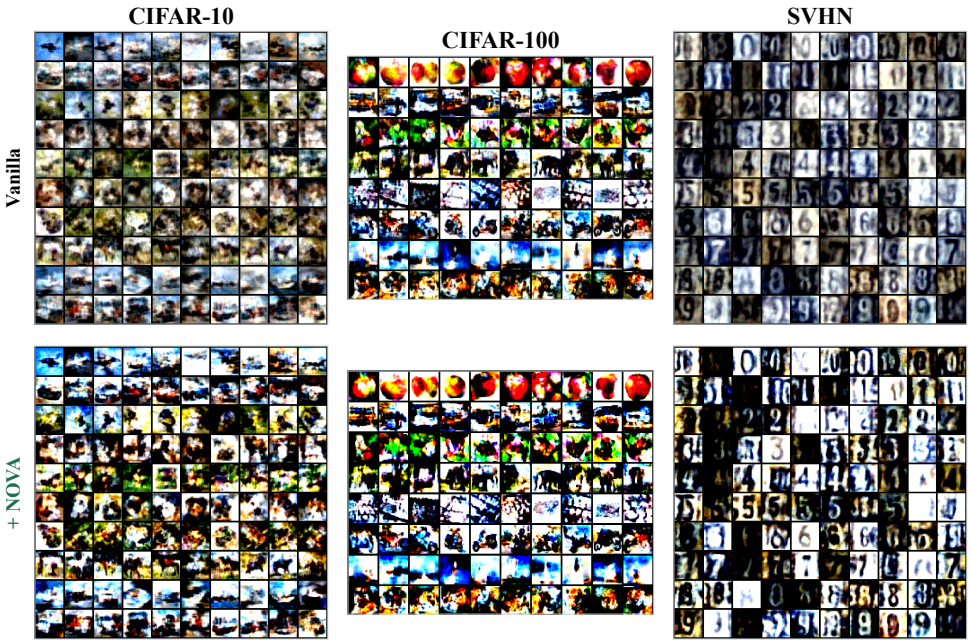


Figure 10: **Actual CAFE artifacts across datasets.** Matched Vanilla and NOVA sets on CIFAR-10, CIFAR-100, and SVHN at IPC10 and synthesis seed 0. Pixels are de-normalized and clipped only for display; no contrast enhancement or delivery-stage perturbation is applied.

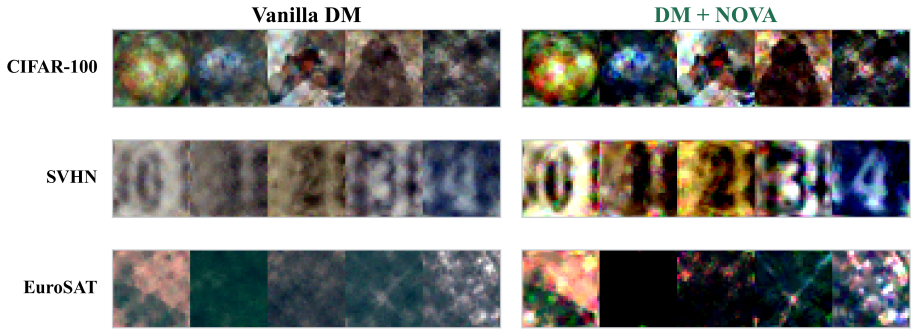


Figure 11: **Actual DM artifacts across datasets.** Five stored examples from distinct classes are shown for matched Vanilla and NOVA sets on CIFAR-100, SVHN, and EuroSAT at IPC10 and synthesis seed 0. Pixels are de-normalized with the corresponding dataset statistics and clipped only for display.