GTS: GAP-TARGETED SYNTHESIS FOR BUDGET-EFFICIENT SYNTHETIC DATA SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have made synthetic images a practical additional source of training data for visual recognition. In this scaling context, dataset distillation compresses a real training set into a small synthetic replacement, an objective misaligned with settings where the complete real dataset is already available. We instead ask how much accuracy a minimal addition of synthetic images can gain when added on top of that complete real set. We introduce *Gap-Targeted Synthesis* (GTS), a data-centric framework that diagnoses where synthetic data are needed before generating them. GTS constructs two complementary structural references solely from real features. A mode gap is an observed class mode whose coverage is insufficient; a variation gap is a class-specific interval missing along variation directions shared across classes. GTS converts these deficits into class- and target-level generation quotas and guides a pre-trained diffusion model toward the assigned feature targets during sampling. Gap diagnosis and synthesis require no downstream student classifier, so the same synthetic set can be reused across training architectures. Neither stage draws from an oversized candidate pool. On ImageWoof2, the reported fixed-budget comparisons show higher mean top-1 accuracy than unguided sampling across three student architectures, while paired-noise measurements show that guidance reaches the assigned mode and variation targets more often.

1 INTRODUCTION

Image generation models have scaled with data and compute, and diffusion models, together with more recent flow matching formulations (Lipman et al., 2023), now generate class-conditioned images at high fidelity and large scale (Peebles & Xie, 2023). This scale has made synthetic imagery a practical source of training data for visual recognition. Scaling synthetic data into a training pipeline does not by itself guarantee proportional gains, however: the literature on recursive training with generated data reports diminishing and non-monotonic returns (He et al., 2023; Azizi et al., 2023), and in the extreme, model quality degrades once generated outputs re-enter training across generations (Shumailov et al., 2024; Alemohammad et al., 2024).

Within this scaling context, dataset distillation targets a specific problem: it compresses a large real training set into a small number of synthetic images so a classifier trained on that compact substitute approaches the accuracy obtained on the full dataset (Gu et al., 2024; Chan-Santiago et al., 2025; Wang et al., 2025; 2026). This compression objective is well matched to settings that lack access to the complete real dataset, but once the complete real dataset is already available, replacing it with a compact substitute is a misaligned goal. The relevant question under full real-data access is instead how to maximize the marginal accuracy gain of a fixed, limited synthetic budget added on top of the complete real training set.

Direct class-conditional sampling from a pre-trained diffusion generator draws from its learned distribution without reference to what the real training set already contains (Peebles & Xie, 2023), so it can repeatedly regenerate variations that real images already represent well. Prior work has studied the scaling behavior of synthetic training data directly and reported diminishing and non-monotonic accuracy gains as the amount of real or synthetic data changes, but this line of work characterizes how much synthetic data helps without treating that budget efficiency itself as an objective to optimize (He et al., 2023; Azizi et al., 2023). Diffusion-based dataset distillation methods pursue a

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.

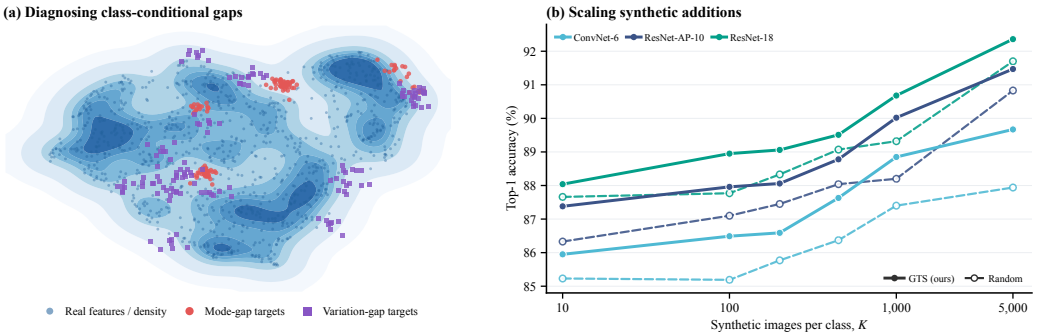

Figure 1: Diagnosing gaps and scaling synthetic additions on ImageWoof2. Left: a representative class in feature space. Gap diagnosis itself is performed in the high-dimensional DINOv2 feature space; for visualization only, the real features and the diagnosed targets are projected to two dimensions with UMAP (McInnes et al., 2018). Blue contours and points show the real-data density. Red mode gap markers indicate relatively under-represented observed modes; purple variation gap markers indicate under-covered intervals along shared variation axes. Target centers are expanded into local marker clouds for visibility; the markers are not generated samples. Right: top-1 accuracy for ConvNet-6, ResNet-AP-10, and ResNet-18 using the complete real set plus a budget-matched Random (dashed open circles) or GTS (solid filled circles) addition, including $K = 5000$. Lines report means over three student seeds; standard deviations are omitted for clarity.

compact synthetic set with strong coverage or discriminative utility (Gu et al., 2024; Chan-Santiago et al., 2025; Wang et al., 2025; 2026); their coverage target is defined for a compact set meant to replace the real data, and it does not address gaps that remain inside a dataset that is already complete. Diffusion-based augmentation methods instead generate semantic variations through seed optimization or language-guided editing (Zhang et al., 2023; Dunlap et al., 2023), yet their target signals do not compare a class's observed feature mass against a reference computed from the real data, so they cannot identify which regions of a class remain under-covered. Feedback-guided synthesis steers sampling with uncertainty from a downstream classifier (Askari Hemmat et al., 2024); its targeting signal is tied to that classifier and does not provide a reference computed from the real data alone. Two problems therefore remain open in the full-real setting: budget efficiency, how much accuracy a small addition of synthetic images can provide beyond the complete real set, has not itself been treated as an objective, and answering it first requires identifying what the real data is missing, a question that low density alone cannot resolve because a sparsely populated region can equally reflect missing data or a variation the class does not support.

We address these two problems by building the reference directly from the structure of the real training set, before any image is generated. Removing each class's mean feature and pooling the residuals across classes yields directions of within-class variation shared across the dataset. Each diagnostic uses the structural reference suited to its evidence. For observed modes, we compare the within-class mode mass with an equal-mass reference over that class's observed modes. Along shared directions, we compare the class's variation mass with a leave-one-class-out reference constructed from the other classes. Deficits relative to these references identify where real coverage falls short. A first diagnostic, the mode gap, treats an observed mode with few real examples as a supported but under-represented variation. A second diagnostic, the variation gap, treats an interval along a shared variation direction with little or no real mass in a class as a variation present elsewhere in the dataset but missing from that class. Fig. 1 illustrates both diagnostics in one representative class and previews how the resulting additions scale against random sampling. The measured deficit converts directly into a generation quota, and a pre-trained diffusion generator is guided toward the resulting point and interval targets during sampling, so the synthetic budget is spent on the diagnosed gaps in the real data.

We evaluate GTS in the full-real additive setting across three student architectures and five main synthetic budgets on ImageWoof2, against budget-matched controls and four prior diffusion-based dataset-distillation methods, and extend the control comparison to two additional ImageNet subsets, a distribution-shift evaluation, and an ablation of its two gap diagnostics. We find that GTS attains the highest mean top-1 accuracy in the reported fixed-budget comparisons.

Our contributions are summarized as follows:

- We formulate the centralized full-real additive setting as fixed-budget placement over class-conditional feature-space targets: marginal utility depends on where a limited synthetic budget is spent, not only on its size.
- We propose GTS, which builds a structural reference directly from the real data, defines two forms of class-conditional coverage gap (the mode gap and the variation gap), converts each into a generation quota, and guides a pre-trained diffusion model directly toward the resulting targets during sampling.
- We evaluate GTS in the full-real additive setting across three student architectures and five synthetic budgets on ImageWoof2, comparing against budget-matched controls and prior diffusion-based dataset-distillation methods, and extend the control comparison to two further ImageNet subsets, finding that it attains the highest mean top-1 accuracy in the reported fixed-budget comparisons. Code will be released with the formal version.

2 Related work

Diffusion-based augmentation. Early work establishes that class-conditional diffusion models can augment ImageNet at scale by fine-tuning a text-to-image model on ImageNet and adding its samples to the real training set (Azizi et al., 2023). Subsequent work focuses on controlling the content and diversity of generated samples. GIF optimizes the latent feature of each seed image for class-maintained informativeness and sample diversity (Zhang et al., 2023), while DA-Fusion uses text-conditioned image-to-image diffusion to produce semantic edits of real examples and learns a pseudo-prompt for each new concept from few labeled images (Trabucco et al., 2024). ALIA first uses vision-language models to describe domains present in a dataset, then applies language-guided image editing and filters edits that are too weak or damage class-relevant content (Dunlap et al., 2023). Diff-Mix addresses the fidelity–diversity tradeoff through inter-class image translation, changing the foreground toward a target class while retaining the reference image's background and assigning an interpolated label (Wang et al., 2024). A separate line targets imbalanced recognition. Fill-Up learns class-specific text tokens through textual inversion and generates additional images to balance class frequencies (Shin et al., 2023); feedback-guided synthesis instead uses one-shot feedback from a classifier trained on the real data to steer a latent diffusion model toward samples that are useful and diverse for that classifier (Askari Hemmat et al., 2024). FedEAS studies budget-aware synthetic augmentation for label-skewed federated learning, allocating samples across client–class pairs from label counts (Lee et al., 2026). These methods improve synthetic augmentation through different forms of generation control, but none explicitly diagnoses coverage gaps in a fully observed real training set, and none allocates a fixed synthetic budget according to such gaps. FedEAS makes allocation explicit, and its target is label imbalance across federated clients, which leaves feature-space coverage within a full centralized real training set unaddressed. Diff-Mix is especially close in its use of cross-class information, but its cross-class operation is image translation for foreground–context recombination. In GTS, other classes instead provide a reference for shared within-class variation after removing class-specific mean features; generation targets the deficient variation of the requested class and retains its fixed class label.

Diffusion-based dataset distillation. Diffusion priors have also been used to construct compact synthetic datasets that replace a much larger real training set. Minimax Diffusion fine-tunes a diffusion generator with minimax criteria that promote representativeness and diversity (Gu et al., 2024), while D^4M uses latent diffusion and category prototypes to produce an architecture-independent distilled set (Su et al., 2024). MGD^3 avoids generator fine-tuning by discovering data modes, guiding sampling toward them, and stopping guidance to reduce artifacts (Chan-Santiago et al., 2025). CaO_2 addresses objective and condition inconsistencies through probability-informed candidate selection followed by latent refinement for conditional alignment (Wang et al., 2025), and Influence-Guided Diffusion steers sampling with trajectory influence to promote training utility and diversity without retraining the diffusion model (Chen et al., 2025). Most closely related, IMS^3 identifies distributional aggregation in diffusion-based distillation, where likelihood training concentrates samples in high-density regions and under-represents boundary regions. It uses inversion-matching fine-tuning to broaden coverage and selective subgroup sampling to retain subsets that are representative within classes and distinctive across classes (Wang et al., 2026). These methods aim to improve the cov-

erage or discriminative value of a compact synthetic set intended to replace the original training data. Accordingly, they are evaluated by training on the distilled set in place of the original real training set. IMS^3 also addresses low-density coverage within the dataset-distillation setting, where the goal is to construct a compact replacement for the real training set, leaving coverage deficits for additive synthesis unaddressed. In contrast, GTS retains the complete real training set and diagnoses coverage gaps from frozen real features before generation. By allocating a fixed synthetic budget according to these gaps and guiding sampling toward the corresponding targets, GTS focuses synthetic additions on regions that are under-covered by the real training data.

3 PRELIMINARIES

3.1 ADDITIVE SETTING AND BUDGET

Let $\mathcal{D}_{\text{real}} = \{(x_i, y_i)\}_{i=1}^{N}$ be a labeled training set with C classes. We retain all N real images. The nominal budget K denotes the number of synthetic images added per class, and $K_{\text{tot}} = CK$ is the total number added to the dataset. Untargeted methods use K images for every class, while GTS redistributes the same total budget across classes. The student is trained on $\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{GTS}}$, where $|\mathcal{D}_{\text{GTS}}| = K_{\text{tot}}$, and evaluated on held-out real images. This additive objective differs from dataset distillation, which trains the student on a compact synthetic replacement for the real set.

3.2 DDIM SAMPLING

Let x_t denote the noisy VAE latent (Rombach et al., 2022) at reverse step t, initialized by $x_T \sim \mathcal{N}(0, I)$. A pre-trained class-conditional diffusion model predicts the noise $\epsilon_\theta(x_t, t, c)$ for class c. With deterministic sampling ($\eta = 0$) used throughout, a reverse step follows the DDIM sampling update (Song et al., 2021)

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, \widehat{x}_0(x_t) + \sqrt{1 - \bar{\alpha}_{t-1}}\, \epsilon_\theta(x_t, t, c). \tag{1}$$

The clean latent in this update is the Tweedie estimate

$$\widehat{x}_0(x_t) = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\, \epsilon_\theta(x_t, t, c)}{\sqrt{\bar{\alpha}_t}}. \tag{2}$$

Here, $\bar{\alpha}_t$ is the cumulative signal coefficient of the fixed diffusion schedule, and $\sqrt{1 - \bar{\alpha}_t}$ is its noise standard deviation. Because $\widehat{x}_0$ is differentiable with respect to x_t, an objective defined on its decoded image can modify selected reverse steps without retraining the diffusion model.

4 METHOD

In the full-real additive setting, GTS treats synthesis as a budget-placement problem. Given $\mathcal{D}_{\text{real}}$ and a fixed budget K_{tot}, it diagnoses coverage gaps in frozen real features, allocates class and target quotas from the measured deficits, and realizes the targets with guided DDIM sampling. Generation is completed before student training, yielding a frozen $\mathcal{D}_{\text{GTS}}$ that can be reused across architectures. Fig. 2 summarizes this pipeline. We next detail gap diagnosis, budget allocation, target realization, and synthetic-set construction.

4.1 STRUCTURAL GAP ANALYSIS FROM REAL DATA

Effective placement of a limited synthetic budget requires identifying regions of the real distribution that are supported but insufficiently covered by the retained data. Low density alone cannot distinguish an under-represented class variation from a region unsupported by that class, so we require structural evidence that a target is both supported and under-covered. We represent every real image with a frozen DINOv2 encoder $z_i = \phi(x_i)$ (Oquab et al., 2024). Each reference specifies the mass a region would have under the corresponding structural assumption. For each diagnostic, we estimate both the observed mass p and a reference mass q from the real features. Their positive difference defines the coverage deficit

$$g(b) = [q(b) - p(b)]_+. \tag{3}$$

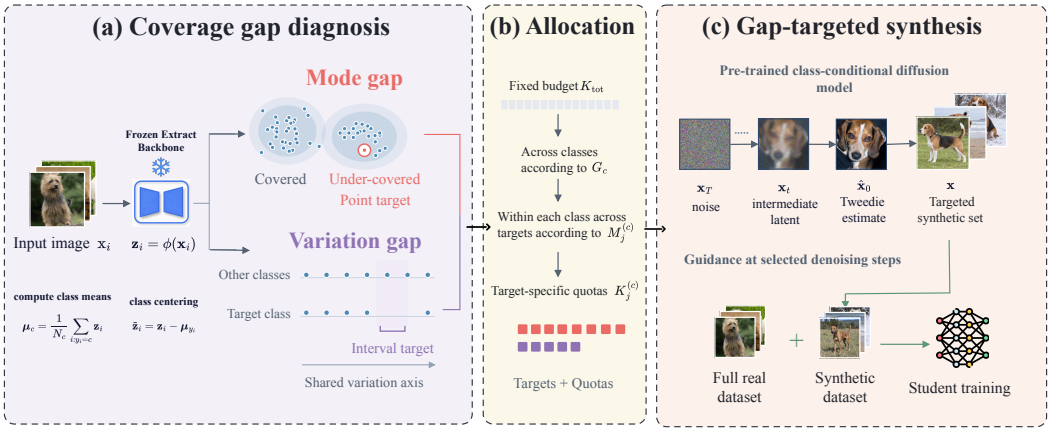

Figure 2: Overview of Gap-Targeted Synthesis (GTS). Real-data analysis identifies mode under-coverage and missing shared variation, then converts the detected gaps into mode gap point targets, variation gap interval targets, and explicit quotas. Gap-guided DDIM sampling generates a compact targeted set that is combined with the full real training set for student training.

The deficit assigned to a retained mode or interval is its *gap mass*. It determines the generation quota for that target. The two diagnostics provide complementary evidence: the mode gap measures under-coverage among observed modes, whereas the variation gap identifies a shared within-class variation that is under-covered in the target class.

Mode gap: mode under-coverage. The mode gap targets variations supported by real examples but represented unevenly within a class. For class c, we cluster its normalized features into R_c observed modes. Every retained mode contains real members, so it supplies its own evidence that the target is valid for class c. Let $\pi_{c,r}$ be the fraction of class-c samples in mode r. We compare this observed fraction with an equal-mass reference of $1/R_c$ per observed mode:

$$M_{c,r}^{\text{mode}} = \left[\frac{1}{R_c} - \pi_{c,r} \right]_+ . \tag{4}$$

A positive mass identifies a mode that is relatively under-represented among the observed modes. Its feature center anchors a mode gap point target.

Variation gap: missing shared variation. The variation gap uses shared variation across other classes to identify under-covered regions beyond the target class's observed mode centers. We subtract the corresponding class mean from every raw feature, $\tilde{z}_i = z_i - \mu_{y_i}$, pool these class-centered residuals, and apply PCA to obtain shared within-class variation axes. Centering reduces the contribution of class identity before PCA, so these axes describe variations that can recur across classes, such as pose or viewpoint. Along each axis, the target-class histogram $p_{c,k}$ is compared with a leave-one-class-out histogram $q_{-c,k}$ formed from the other real classes. A supported bin is retained when the target class contains less than a fixed fraction of its reference mass. Its deficit is

$$M_{c,k,b}^{\text{var}} = \frac{q_{-c,k}(b) - p_{c,k}(b)}{|\mathcal{V}|}, \quad q_{-c,k}(b) \geq \tau, \quad p_{c,k}(b) < \rho q_{-c,k}(b), \tag{5}$$

where $\mathcal{V}$ is the set of shared axes. The retained bin defines a variation gap interval target in the coordinate system of class c. Its support therefore comes from a within-class variation observed elsewhere in the same real dataset. See App. B for the exact feature standardization and bin construction.

4.2 Coverage-aware quota allocation

A diagnosis supports budget efficiency only when it determines placement of the fixed image budget. Each retained target j belongs to a diagnostic $s(j) \in \{\text{mode}, \text{var}\}$ and carries a raw gap mass $M_j^{(c)}$. Because the two diagnostics estimate mass in different coordinate systems, we apply a fixed

diagnostic calibration $\lambda_{s(j)}$ before allocation. Let $\overline{M}_j^{(c)} = \lambda_{s(j)} M_j^{(c)}$ and $G_c = \sum_j \overline{M}_j^{(c)}$. We allocate the total budget across classes in proportion to their weighted gap masses G_c, subject to per-class lower and upper bounds, then distribute each class budget K_c among its targets in proportion to their weighted masses. The resulting integer quotas $K_j^{(c)}$ preserve the total budget exactly, so only its placement across classes and targets changes. App. C gives the calibration values, allocation formula, and rounding procedure.

4.3 Gap-guided diffusion

The allocation in Sec. 4.2 determines how many synthetic samples are assigned to each detected gap, but standard diffusion sampling does not ensure that they reach the corresponding target region. We therefore introduce gap-guided diffusion, which converts each allocated mode point or variation interval into a differentiable guidance objective during sampling. At selected reverse steps, the DiT predicts class-conditional noise $\epsilon_t = \epsilon_\theta(x_t, t, c)$ in VAE latent space. We use that prediction to form the clean-latent estimate $\widehat{x}_0(x_t)$, decode it through the VAE, and extract a DINOv2 feature z_t to measure discrepancy from the assigned target. The guided steps are the only reverse steps at which this path is evaluated. The mode gap uses a seed-specific point near the assigned real mode, whereas the variation gap uses an interval on a shared variation axis. For the assigned class c and shared axis v_k, its variation gap coordinate is $u_{t,k} = v_k^\top (z_t - \mu_c)/s_k$, using the frozen scale s_k. See App. B for its definition. With unit mode gap target a_j and variation gap interval I_j, the two losses are

$$\mathcal{L}_{\text{mode}} = (1 - \text{norm}(z_t)^\top a_j)^2, \qquad \mathcal{L}_{\text{var}} = d(u_{t,k}, I_j)^2 \,, \tag{6}$$

where $d(u, [\ell, r]) = \max(\ell - u, u - r, 0)$. The first loss points the sample toward an under-covered observed mode. The second moves only when the decoded feature lies outside the missing-variation interval. These objectives use the diagnosed feature-space targets directly and require no downstream classifier.

To translate the feature-space target into a sampling update, we use $s = s(j)$ to select the corresponding loss in Eq. (6). Backpropagating this loss through DINOv2, the VAE decoder, and the clean-latent estimate gives a gradient with respect to the current latent x_t. We modify the DDIM noise prediction with a norm-capped guidance correction:

$$\epsilon_t' = \epsilon_t + \text{Cap}_\gamma\big(\beta_s \sqrt{1 - \bar{\alpha}_t}\, \nabla_{x_t} \mathcal{L}_s; \epsilon_t\big) \,. \tag{7}$$

Here, β_s controls the guidance strength; the primary configuration uses $\beta_{\text{mode}} = 112.5$ and $\beta_{\text{var}} = 60$. The factor $\sqrt{1 - \bar{\alpha}_t}$ follows from the standard score-to-noise conversion; see App. D for the derivation. The class condition preserves the requested label, while the additional gradient controls the within-class target. The norm cap stabilizes the correction. For the variation gap, the main loss is the interval distance in Eq. (6). During the final guided steps, a constrained repair can reduce motion orthogonal to the target axis after the sample has entered both the detected interval and its inner control interval. A repair is accepted only if both memberships are preserved and the orthogonal displacement decreases. These operations use the full VAE-decoder and frozen-encoder path. See App. D for their exact construction, schedule, and constants.

4.4 Synthetic-set construction

To keep budget placement independent of any downstream learner, generation is completed and frozen before student optimization. For every target, we generate exactly $K_j^{(c)}$ images from independent initial noises and add all allocated images to the complete real training set. The resulting synthetic set is student-independent and can be reused across architectures. App. A gives the complete procedure in Algorithm 1.

5 Experiments

We evaluate the same sequence established in Sec. 4: diagnose supported coverage gaps, allocate a fixed budget, realize the assigned targets, and measure their downstream utility. The main comparisons test whether gap targeting makes a fixed synthetic budget more useful than unguided generation; paired-noise experiments and component ablations isolate target realization and the two diagnostic signals.

Table 1: ImageWoof2 top-1 accuracy (%) in the full-real additive setting. The best results are marked as bold.

			Controlled baselines			Prior DD methods				Ours
K	Student	Full	Random	Rebalance	Select	MiniMax	CaO_2	MGD^3	IMS^3	GTS
10	ConvNet-6	$84.20_{\pm 0.37}$	$85.23_{\pm 0.11}$	$85.09_{\pm 0.34}$	$84.73_{\pm 0.38}$	$85.13_{\pm 0.12}$	$85.13_{\pm 0.17}$	$84.83_{\pm 0.21}$	$85.20_{\pm 0.37}$	$\mathbf{85.95}_{\pm 0.41}$
	ResNet-AP-10	$85.80_{\pm 0.07}$	$86.33_{\pm 0.26}$	$86.89_{\pm 0.25}$	$87.25_{\pm 0.11}$	$87.07_{\pm 0.05}$	$86.93_{\pm 0.12}$	$86.70_{\pm 0.08}$	$86.93_{\pm 0.26}$	$\mathbf{87.38}_{\pm 0.07}$
	ResNet-18	$87.00_{\pm 0.32}$	$87.66_{\pm 0.09}$	$87.66_{\pm 0.21}$	$87.35_{\pm 0.34}$	$87.73_{\pm 0.05}$	$87.63_{\pm 0.21}$	$87.73_{\pm 0.05}$	$87.67_{\pm 0.12}$	$\mathbf{88.04}_{\pm 0.13}$
100	ConvNet-6	$84.20_{\pm 0.37}$	$85.19_{\pm 0.08}$	$85.44_{\pm 0.24}$	$85.98_{\pm 0.13}$	$85.60_{\pm 0.58}$	$85.37_{\pm 0.19}$	$85.87_{\pm 0.29}$	$85.37_{\pm 0.26}$	$\mathbf{86.49}_{\pm 0.14}$
	ResNet-AP-10	$85.80_{\pm 0.07}$	$87.10_{\pm 0.43}$	$87.02_{\pm 0.07}$	$87.30_{\pm 0.11}$	$87.23_{\pm 0.21}$	$87.37_{\pm 0.31}$	$87.37_{\pm 0.26}$	$87.13_{\pm 0.29}$	$\mathbf{87.96}_{\pm 0.12}$
	ResNet-18	$87.00_{\pm 0.32}$	$87.77_{\pm 0.29}$	$87.83_{\pm 0.09}$	$88.44_{\pm 0.46}$	$87.90_{\pm 0.22}$	$88.03_{\pm 0.26}$	$88.10_{\pm 0.29}$	$87.87_{\pm 0.24}$	$\mathbf{88.95}_{\pm 0.37}$
200	ConvNet-6	$84.20_{\pm 0.37}$	$85.77_{\pm 0.52}$	$85.99_{\pm 0.29}$	$86.18_{\pm 0.03}$	$85.82_{\pm 0.18}$	$85.75_{\pm 0.46}$	$86.33_{\pm 0.22}$	$86.03_{\pm 0.34}$	$\mathbf{86.59}_{\pm 0.08}$
	ResNet-AP-10	$85.80_{\pm 0.07}$	$87.45_{\pm 0.63}$	$87.68_{\pm 0.25}$	$87.50_{\pm 0.37}$	$87.55_{\pm 0.31}$	$87.91_{\pm 0.42}$	$87.91_{\pm 0.08}$	$87.99_{\pm 0.26}$	$\mathbf{88.06}_{\pm 0.18}$
	ResNet-18	$87.00_{\pm 0.32}$	$88.33_{\pm 0.35}$	$88.72_{\pm 0.13}$	$87.96_{\pm 0.07}$	$88.39_{\pm 0.24}$	$88.27_{\pm 0.07}$	$88.50_{\pm 0.32}$	$88.34_{\pm 0.13}$	$\mathbf{89.06}_{\pm 0.20}$
450	ConvNet-6	$84.20_{\pm 0.37}$	$86.37_{\pm 0.21}$	$86.92_{\pm 0.52}$	$86.49_{\pm 0.34}$	$86.49_{\pm 0.26}$	$86.51_{\pm 0.82}$	$87.33_{\pm 0.41}$	$86.77_{\pm 0.33}$	$\mathbf{87.63}_{\pm 0.37}$
	ResNet-AP-10	$85.80_{\pm 0.07}$	$88.04_{\pm 0.63}$	$88.67_{\pm 0.81}$	$88.14_{\pm 0.22}$	$87.91_{\pm 0.03}$	$87.76_{\pm 0.42}$	$88.44_{\pm 0.35}$	$88.22_{\pm 0.11}$	$\mathbf{88.78}_{\pm 0.23}$
	ResNet-18	$87.00_{\pm 0.32}$	$89.07_{\pm 0.21}$	$88.93_{\pm 0.46}$	$89.08_{\pm 0.83}$	$89.08_{\pm 0.11}$	$88.93_{\pm 0.31}$	$89.16_{\pm 0.27}$	$88.75_{\pm 0.09}$	$\mathbf{89.51}_{\pm 0.27}$
1000	ConvNet-6	$84.20_{\pm 0.37}$	$87.40_{\pm 0.05}$	$87.45_{\pm 0.09}$	$87.35_{\pm 0.21}$	$87.50_{\pm 0.14}$	$87.20_{\pm 0.26}$	$88.00_{\pm 0.17}$	$87.40_{\pm 0.58}$	$\mathbf{88.85}_{\pm 0.24}$
	ResNet-AP-10	$85.80_{\pm 0.07}$	$88.20_{\pm 0.23}$	$89.23_{\pm 0.47}$	$88.62_{\pm 0.09}$	$89.00_{\pm 0.33}$	$88.90_{\pm 0.52}$	$89.70_{\pm 0.31}$	$88.50_{\pm 0.08}$	$\mathbf{90.02}_{\pm 0.22}$
	ResNet-18	$87.00_{\pm 0.32}$	$89.32_{\pm 0.15}$	$89.49_{\pm 0.47}$	$89.13_{\pm 0.33}$	$89.10_{\pm 0.12}$	$88.70_{\pm 0.08}$	$89.80_{\pm 0.22}$	$89.28_{\pm 0.16}$	$\mathbf{90.68}_{\pm 0.05}$

5.1 Experimental settings

Datasets. ImageWoof2 (Howard, 2019) is the primary benchmark; its ten related dog classes combine substantial within-class variation with high between-class similarity. ImageNette (Howard, 2019) and ImageIDC, the ten-class ImageNet subset introduced by Kim et al. (2022), test the same additive protocol under different class structures. We also use the corresponding classes from ImageNet-Sketch (Wang et al., 2019) and the two ImageNet-V2 variants (Recht et al., 2019), together with ImageWoof2-C, which applies the ImageNet-C common-corruption protocol to ImageWoof2 (Hendrycks & Dietterich, 2019), for distribution-shift evaluation. Construction and evaluation details are given in App. E.

Baselines. Random sampling is the direct, unguided class-conditional DiT control (Peebles & Xie, 2023), with the gap gradient disabled. Rebalance retains GTS's class quotas without within-class guidance, Select chooses samples nearest to the targets from a larger unguided pool, and Feedback uses classifier uncertainty (Askari Hemmat et al., 2024). We also compare with MiniMax Diffusion (Gu et al., 2024), CaO_2 (Wang et al., 2025), MGD^3 (Chan-Santiago et al., 2025), and IMS^3 (Wang et al., 2026). All additive comparisons retain the full real set and match the total synthetic budget.

Matched sampling budget. Gap guidance rewrites the noise prediction at selected reverse steps rather than adding steps, so the guided and unguided arms use the same number of function evaluations: the per-image sampler audit records 50 deterministic DDIM steps and 50 DiT evaluations per image in both arms. The reported gains therefore cannot be attributed to a larger sampling budget. App. K reports the wall-clock cost of the guidance path.

Metrics. The primary metric is top-1 accuracy, reported as mean $\pm$ standard deviation over three student seeds. Mode and variation target-hit rates measure whether generated images reach their assigned targets.

5.2 Classification comparisons

ImageWoof2. The central question is whether gap targeting increases the marginal utility of a fixed synthetic budget beyond the complete real set. Tab. 1 answers this question on ImageWoof2 by matching the number of added images across methods and student architectures. GTS attains the highest mean top-1 accuracy at every budget and for all three student architectures, and the ordering over the controls holds at all five reported budgets. At $K = 100$ with ResNet-18 it exceeds unguided placement by 1.18 points, class-level reallocation by 1.12, and post-generation selection by 0.51. These fixed-budget comparisons isolate placement quality from the number of added images when assessing downstream accuracy.

ImageNette and ImageIDC. Tab. 2 tests the same fixed-budget placement decision on two ImageNet subsets with ResNet-18 at $K = 100$ and $K = 450$. GTS attains the highest mean top-1 accuracy in all four dataset-budget cells, exceeding the full-real baseline by 0.56 to 2.00 points and the strongest controlled baseline in each cell by 0.15 to 0.60. Which control ranks second varies with the dataset and the budget, so the fixed-budget ordering found on ImageWoof2 carries to class structures that differ from the primary benchmark.

Table 2: ResNet-18 top-1 accuracy (%) on ImageNette and ImageIDC under matched additive budgets of K synthetic images per class, i.e., $10 \times K$ total synthetic images (mean $\pm$ standard deviation over three student seeds).

Dataset	K	Full real	Random	Rebalance	Select	Feedback	GTS
ImageNette	100	$92.94_{\pm 0.14}$	$93.35_{\pm 0.21}$	$93.17_{\pm 0.08}$	$93.00_{\pm 0.38}$	$93.32_{\pm 0.24}$	$\mathbf{93.50}_{\pm 0.13}$
	450	$92.94_{\pm 0.14}$	$93.61_{\pm 0.21}$	$93.91_{\pm 0.33}$	$94.19_{\pm 0.25}$	$94.14_{\pm 0.41}$	$\mathbf{94.37}_{\pm 0.12}$
ImageIDC	100	$93.00_{\pm 0.20}$	$93.80_{\pm 0.20}$	$93.40_{\pm 0.20}$	$93.40_{\pm 0.35}$	$94.00_{\pm 0.35}$	$\mathbf{94.60}_{\pm 0.40}$
	450	$93.00_{\pm 0.20}$	$93.40_{\pm 0.20}$	$94.40_{\pm 0.00}$	$94.00_{\pm 0.35}$	$94.40_{\pm 0.20}$	$\mathbf{95.00}_{\pm 0.35}$

5.3 Distribution-level analysis

We test whether gap placement transfers beyond clean evaluation in Tab. 3, using ResNet-AP-10 trained on clean ImageWoof2 at $K = 100$. GTS is highest on all four shifted evaluation sets, exceeding the full-real baseline by 0.31 to 5.06 points and unguided placement by 0.74 to 2.96, so gap placement decided on clean data also transfers to shifted evaluation. Fig. 3 then uses paired initial noises to measure attainment of the assigned targets. Guidance reaches the assigned mode point in 95% of samples against 31% unguided, and the assigned variation interval in 84% and 80% at $K = 100$ and $K = 1000$ against 1.4% and 2.0%, so variation targets are almost never reached by chance.

Table 3: Distribution-shift top-1 accuracy (%) for ResNet-AP-10 at $K = 100$. Entries are mean $\pm$ standard deviation. Threshold0.7-10 contains 10 images per class (100 total), giving a 1-percentage-point accuracy resolution. See App. E for the evaluation protocol.

Eval. set	Full real	Random	GTS
ImageNet-Sketch-10	$42.67_{\pm 0.98}$	$44.77_{\pm 3.41}$	$\mathbf{47.73}_{\pm 3.35}$
ImageWoof2-C Avg.	$70.55_{\pm 0.41}$	$70.12_{\pm 0.09}$	$\mathbf{70.86}_{\pm 0.23}$
ImageNet-V2-Threshold0.7-10	$77.33_{\pm 0.58}$	$77.67_{\pm 0.58}$	$\mathbf{79.00}_{\pm 1.00}$
ImageNet-V2-TopImages-10	$84.00_{\pm 1.73}$	$87.00_{\pm 2.52}$	$\mathbf{88.00}_{\pm 1.53}$

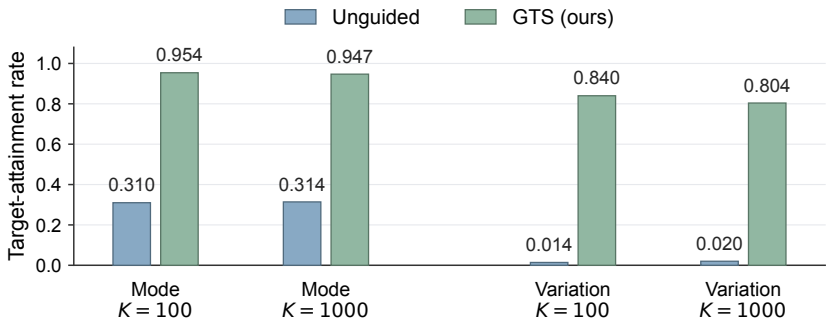

Figure 3: Paired-noise target-attainment rates on ImageWoof2.

5.4 Ablation and sensitivity analysis

The mode gap and variation gap are complementary structural gap diagnostics: one measures imbalance among supported modes, whereas the other transfers shared variation evidence across classes.

Tab. 4 evaluates them separately and together while holding the total synthetic budget fixed. When one diagnostic is removed, its quota is reallocated to the remaining targets. Either diagnostic alone improves on using neither, and combining the two gives the highest accuracy under the same total synthetic budget, for every student architecture and both budgets.

Table 4: Ablation of the two structural gap components on ImageWoof2. Mode and Variation columns indicate whether the mode gap and variation gap diagnostics, respectively, are included. Entries are top-1 accuracy (%). The shaded row in bold is the full GTS configuration.

Gap components		ConvNet-6		ResNet-AP-10		ResNet-18	
Mode	Variation	$K = 100$	$K = 450$	$K = 100$	$K = 450$	$K = 100$	$K = 450$
×	×	$85.19_{\pm 0.08}$	$86.37_{\pm 0.21}$	$87.10_{\pm 0.43}$	$88.04_{\pm 0.63}$	$87.77_{\pm 0.29}$	$89.07_{\pm 0.21}$
✓	×	$85.62_{\pm 0.14}$	$87.17_{\pm 0.19}$	$87.22_{\pm 0.31}$	$88.57_{\pm 0.15}$	$88.55_{\pm 0.23}$	$89.17_{\pm 0.03}$
×	✓	$85.98_{\pm 0.25}$	$87.55_{\pm 0.33}$	$87.38_{\pm 0.05}$	$88.65_{\pm 0.19}$	$88.69_{\pm 0.47}$	$89.19_{\pm 0.30}$
✓	✓	$\mathbf{86.49}_{\pm 0.14}$	$\mathbf{87.63}_{\pm 0.37}$	$\mathbf{87.96}_{\pm 0.12}$	$\mathbf{88.78}_{\pm 0.23}$	$\mathbf{88.95}_{\pm 0.37}$	$\mathbf{89.51}_{\pm 0.27}$

We further examine the sensitivity of GTS to the two guidance strengths in Eq. 7. Throughout the main experiments, we use a fixed default configuration of $(\beta_{\text{mode}}, \beta_{\text{var}}) = (112.5, 60)$ across synthetic budgets, without budget-specific tuning. To assess the robustness of this choice, we vary one guidance strength around its default value while keeping the other fixed, and evaluate three representative synthetic budgets, $K \in \{100, 200, 1000\}$. As shown in Fig. 4, the default configuration is the best-performing setting at all three budgets in both sweeps, and every tested alternative stays within 1.01 points of it. The fixed configuration therefore provides a stable operating point that transfers across synthetic budgets without budget-specific tuning.

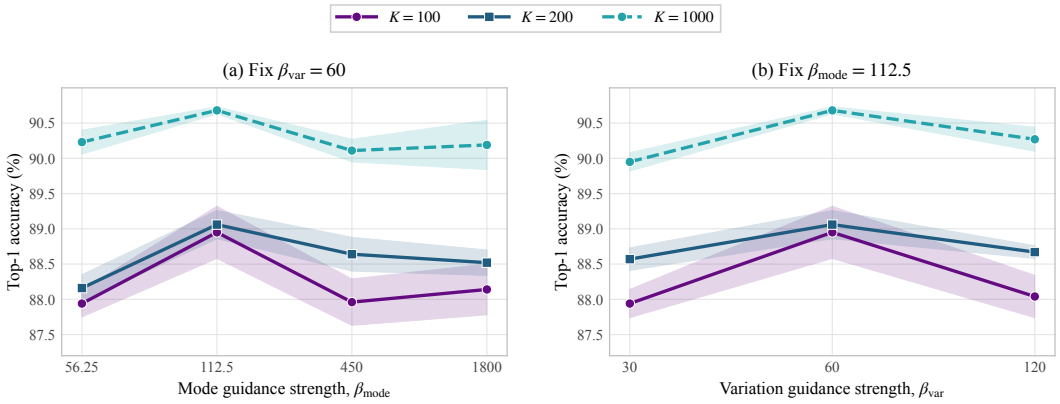

Figure 4: Sensitivity analysis of the guidance strengths with ResNet-18 on ImageWoof2. Left: β_{mode} is varied while $\beta_{\text{var}} = 60$. Right: β_{var} is varied while $\beta_{\text{mode}} = 112.5$. Lines report means over three student seeds, and shaded bands indicate one standard deviation.

6 Conclusion

Most synthetic-data research scales the amount of generated data or replaces real data with compact synthetic sets, and the complementary question of where a limited synthetic budget should be spent when the complete real training set is retained has been left open. We formulate the centralized full-real additive setting as fixed-budget placement over class-conditional feature-space targets, and we propose Gap-Targeted Synthesis (GTS), which diagnoses under-covered modes and shared-variation intervals in the real data, allocates the fixed budget by their measured deficits, and guides diffusion toward the resulting targets. Across the comparisons we report, covering three student architectures on ImageWoof2 and two further ImageNet subsets, GTS attains the highest mean top-1 accuracy. Where a synthetic budget is placed therefore deserves attention alongside how large it is.

AI USE STATEMENT

In this work, generative AI tools were used for prose polishing and coding assistance during manuscript and codebase preparation. The research ideas, experimental design, and result analysis were carried out by the authors. The authors reviewed all AI-assisted text and code and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses public ImageNet subsets and involves no human subjects and no personally identifying data. The synthetic images used in our experiments are produced by a pre-trained class-conditional diffusion model and do not depict real individuals. Augmenting a training set with synthetic images carries the generative model's own distributional properties into the downstream classifier, and the approach introduced in this paper determines the generation budget from coverage deficits measured on the real data itself.

REPRODUCIBILITY STATEMENT

Sec. 4 gives the formal construction of GTS in full, and App. B, App. C, and App. D describe its implementation-level configuration for the gap-analysis, quota-allocation, and guidance stages, respectively. The datasets used are public ImageNet subsets, and the evaluation protocol, student architectures, synthetic budgets, and the compared baselines are listed in Sec. 5. See App. G for the controlled mode-deletion protocol; Sec. 5 reports the distribution-shift evaluation. Code will be released with the camera-ready version.

REFERENCES

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go MAD. In *ICLR*, 2024. URL https://arxiv.org/abs/2307.01850.

Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *TMLR*, 2024. URL https://arxiv.org/abs/2310.00158.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves ImageNet classification. *TMLR*, 2023. URL https://arxiv.org/abs/2304.08466.

Jeffrey A. Chan-Santiago, Praveen Tirupattur, Gaurav Kumar Nayak, Gaowen Liu, and Mubarak Shah. MGD^3: Mode-guided dataset distillation using diffusion models. In *ICML*, 2025. URL https://arxiv.org/abs/2505.18963.

Mingyang Chen, Jiawei Du, Bo Huang, Yi Wang, Xiaobo Zhang, and Wei Wang. Influence-guided diffusion for dataset distillation. In *ICLR*, 2025. URL https://openreview.net/forum?id=0whx8MhysK.

Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E. Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f99f7b22ad47fa6ce151730cf8d17911-Abstract-Conference.html.

Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *CVPR*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Gu_Efficient_Dataset_Distillation_via_Minimax_Diffusion_CVPR_2024_paper.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip H. S. Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023. URL https://arxiv.org/abs/2210.07574.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from ImageNet, 2019. URL https://github.com/fastai/imagenette.

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

Sangwoo Lee, Sunghwan Park, and Jaewoo Lee. WHERE to generate matters: Budget-aware synthetic augmentation for label skewed federated learning. *arXiv preprint arXiv:2607.06616*, 2026. URL https://arxiv.org/abs/2607.06616.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. URL https://arxiv.org/abs/2210.02747.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. URL https://arxiv.org/abs/2304.07193.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, pp. 5389–5400, 2019. URL https://proceedings.mlr.press/v97/recht19a.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Joonghyuk Shin, Minguk Kang, and Jaesik Park. Fill-up: Balancing long-tailed data with generative models. *arXiv preprint arXiv:2306.07200*, 2023. URL https://arxiv.org/abs/2306.07200.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024. URL https://www.nature.com/articles/s41586-024-07566-y.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. D^4M: Dataset Distillation via Disentangled Diffusion Model. In *CVPR*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Su_D4_Dataset_Distillation_via_Disentangled_Diffusion_Model_CVPR_2024_paper.html.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *ICLR*, 2024. URL https://arxiv.org/abs/2302.07944.

Chenru Wang, Yunyi Chen, Zijun Yang, Joey Tianyi Zhou, and Chi Zhang. IMS3: Breaking distributional aggregation in diffusion-based dataset distillation. In *CVPR*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_IMS3_Breaking_Distributional_Aggregation_in_Diffusion-Based_Dataset_Distillation_CVPR_2026_paper.html.

Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3eefceb8087e964f89c2d59e8a249915-Abstract.html.

Haoxuan Wang, Zhenghao Zhao, Junyi Wu, Yuzhang Shang, Gaowen Liu, and Yan Yan. CaO_2: Rectifying inconsistencies in diffusion-based dataset distillation. In *ICCV*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_CaO2_Rectifying_Inconsistencies_in_Diffusion-Based_Dataset_Distillation_ICCV_2025_paper.html.

Zhicai Wang, Longhui Wei, Tan Wang, Heyu Chen, Yanbin Hao, Xiang Wang, Xiangnan He, and Qi Tian. Enhance image classification via inter-class image mixup with diffusion model. In *CVPR*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_Enhance_Image_Classification_via_Inter-Class_Image_Mixup_with_Diffusion_Model_CVPR_2024_paper.html.

Yifan Zhang, Daquan Zhou, Bryan Hooi, Kai Wang, and Jiashi Feng. Expanding small-scale datasets with guided imagination. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f188a55392d3a7509b0b27f8d24364bb-Abstract-Conference.html.

APPENDIX CONTENTS

- Appendix A: Complete GTS procedure 14
- Appendix B: Gap-analysis implementation details 14
- Appendix C: Quota-allocation implementation details 15
- Appendix D: Guidance implementation details 15
- Appendix E: Experimental details 16
- Appendix F: Multi-architecture budget scaling 17
- Appendix G: Controlled mode deletion 17
- Appendix H: Training-set regimes 19
- Appendix I: Qualitative inspection of shared variation axes 20
- Appendix J: Qualitative comparison of gap-guided samples 21
- Appendix K: Generation cost 22

A COMPLETE GTS PROCEDURE

Algorithm 1 summarizes structural gap analysis, quota allocation, and gap-guided diffusion sampling.

Algorithm 1 Gap-Targeted Synthesis. Mode and Variation refer to the mode gap and variation gap diagnostics, respectively.

Require: Real set $\mathcal{D}_{\text{real}}$; frozen encoder ϕ and decoder Dec; class-conditional noise predictor ϵ_θ; total budget K_{tot}
Ensure: Targeted synthetic set $\mathcal{D}_{\text{GTS}}$

$z_i \leftarrow \text{norm}(\phi(x_i))$ for all $(x_i, y_i) \in \mathcal{D}_{\text{real}}$
For each class c, fit Euclidean Lloyd k-means to $\{z_i : y_i = c\}$ with $R_c = 10$, 20 initializations, and random seed 2026080501; unit-normalize the fitted centers
Set $\pi_{c,r}$ to each cluster's fraction of class-c samples; construct Mode targets a_j from positive masses in Eq. (4)
Fit shared axes and construct Variation targets (v_k, I_j)
Compute gap masses $M_j^{(c)}$ and calibrated masses $\overline{M}_j^{(c)} \leftarrow \lambda_{s(j)} M_j^{(c)}$
Allocate integer class and target quotas $K_c, K_j^{(c)}$ using Eq. (11) and App. C
$\mathcal{D}_{\text{GTS}} \leftarrow \varnothing$
for each (c, j) and repetition $k = 1, \ldots, K_j^{(c)}$ **do**
    Sample $x_T \sim \mathcal{N}(0, I)$ and instantiate target j
    For Variation, run the paired unguided reference trajectory
    for $t = T, \ldots, 1$ in the DDIM reverse process **do**
        $\epsilon_t \leftarrow \epsilon_\theta(x_t, t, c)$
        $\epsilon'_t \leftarrow \epsilon_t$
        if t is a guidance step **then**
            $\widehat{x}_0 \leftarrow (x_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_t)/\sqrt{\bar{\alpha}_t}$
            $z_t \leftarrow \phi(\text{Dec}(\widehat{x}_0))$
            if j is Mode **then**
                $\mathcal{L}_s \leftarrow (1 - \text{norm}(z_t)^\top a_j)^2$
            else
                $u_{t,k} \leftarrow v_k^\top (z_t - \mu_c)/s_k$; $\mathcal{L}_s \leftarrow d(u_{t,k}, I_j)^2$
            end if
            $\epsilon'_t \leftarrow \epsilon_t + \text{Cap}_\gamma(\beta_s \sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t}\mathcal{L}_s; \epsilon_t)$
            if j is Variation and the reverse iteration is in the repair window **then**
                $\epsilon'_t \leftarrow \text{VariationRepair}(\epsilon'_t, x_t, j)$ (App. D)
            end if
        end if
        $x_{t-1} \leftarrow \text{DDIMStep}(x_t, \epsilon'_t, t)$
    end for
    $\mathcal{D}_{\text{GTS}} \leftarrow \mathcal{D}_{\text{GTS}} \cup \{(\text{Dec}(x_0), c)\}$
end for
return $\mathcal{D}_{\text{GTS}}$

B GAP-ANALYSIS IMPLEMENTATION DETAILS

We use frozen DINOv2 ViT-B/14 CLS features. Cosine measurements and mode gap clustering use unit-normalized features. Each class is clustered independently with Euclidean Lloyd k-means ($R_c = 10$, 20 initializations, random seed 2026080501); fitted centers are unit-normalized after clustering. The mode proportions $\pi_{c,r}$ in Eq. (4) are the fractions of class-c images assigned to the respective clusters. To construct the shared variation gap axes, we remove each class mean and pool the residuals,

$$\widetilde{\mathcal{Z}} = \bigcup_{c=1}^{C} \{z - \mu_c : z \in \mathcal{Z}_c\}. \tag{8}$$

A shared PCA basis $\{v_k\}$ is fitted to this pool. We standardize projection coordinates with the robust pooled scale $s_k = 1.4826\,\mathrm{MAD}(\widetilde{\mathcal{Z}}v_k)$:

$$u_{i,k} = \frac{v_k^\top (z_i - \mu_{y_i})}{s_k}. \tag{9}$$

For each class and axis, the pooled 1st–99th percentile range is divided into $B = 20$ equal-width bins. A bin is retained when $q_{-c,k}(b) \geq \tau = 0.02$ and $p_{c,k}(b) < \rho q_{-c,k}(b)$ with $\rho = 0.5$. Each retained bin remains a separate variation gap interval; adjacent bins are not merged.

For the mode gap, each allocated sample receives a deterministic target derived from its initial-noise seed. If $m_{c,r}$ is its mode center and $z_{c,r,n}$ is the selected real member, the target is

$$a_{c,r,n} = \mathrm{norm}(m_{c,r} + 0.3(z_{c,r,n} - m_{c,r}))\,. \tag{10}$$

This construction varies targets within an observed mode without adding Gaussian feature noise.

C Quota-allocation implementation details

Each target belongs to $s(j) \in \{\mathrm{mode}, \mathrm{var}\}$. We calibrate its raw gap mass with $\lambda_{\mathrm{mode}} = 1$ and $\lambda_{\mathrm{var}} = 2$, giving $\overline{M}_j^{(c)} = \lambda_{s(j)} M_j^{(c)}$. Let $G_c = \sum_j \overline{M}_j^{(c)}$ and $K = K_{\mathrm{tot}}/C$. We compute a continuous class quota q_c by scaling G_c and clipping it to the interval $[0.5K, 2K]$:

$$q_c = \mathrm{clip}(\eta G_c, 0.5K, 2K), \qquad \sum_{c=1}^{C} q_c = K_{\mathrm{tot}}. \tag{11}$$

The scale η is found by 100 iterations of bisection. We floor the q_c values and allocate the remaining images in descending order of fractional remainder, with class name as the deterministic tie-breaker. This gives integer class budgets K_c within the bounds and summing to K_{tot}. Within each class, the continuous quota of target j is $K_c \overline{M}_j^{(c)}/G_c$. We again floor and assign the remainder in descending order, using the sorted target key as the tie-breaker. Hence, every class and target count is an integer and the global budget is preserved exactly.

D Guidance implementation details

At a guidance step, the Tweedie estimate is decoded by the VAE and passed through the frozen DINOv2 encoder. We first derive the schedule-dependent factor in Eq. (7). Let $\sigma_t = \sqrt{1 - \bar{\alpha}_t}$. Under the noise-prediction parameterization, the model score and predicted noise satisfy $s_\theta(x_t, t) = -\epsilon_\theta(x_t, t)/\sigma_t$. To decrease the selected feature-space loss, we guide the score in the negative-gradient direction, $s'_t = s_\theta(x_t, t) - \beta_s \nabla_{x_t} \mathcal{L}_s$. Converting the guided score back to the noise prediction used by DDIM gives

$$\begin{aligned} \epsilon'_t &= -\sigma_t s'_t \\ &= -\sigma_t \left(s_\theta(x_t, t) - \beta_s \nabla_{x_t} \mathcal{L}_s\right) \\ &= \epsilon_t + \beta_s \sqrt{1 - \bar{\alpha}_t}\, \nabla_{x_t} \mathcal{L}_s. \end{aligned} \tag{12}$$

Thus, $\sqrt{1 - \bar{\alpha}_t}$ is fixed by the diffusion schedule and is required to express a score correction as a noise-prediction correction; it is not a tunable parameter. The cap in Eq. (7) is applied to this resulting correction.

For the variation gap, a detected interval $[\ell_j, r_j]$ has midpoint $a_j = (\ell_j + r_j)/2$. The main guidance pass uses the centered inner interval

$$I_j = [a_j - \delta_j, a_j + \delta_j], \qquad \delta_j = \tfrac{1}{2} \min(a_j - \ell_j, r_j - a_j). \tag{13}$$

The noise correction in Eq. (7) is capped without changing its direction. For the uncapped correction $\Delta\epsilon_t$, the implementation first computes its norm ratio to the current noise prediction,

$$r_t = \frac{\|\Delta\epsilon_t\|_2}{\max(\|\epsilon_t\|_2, \varepsilon)}, \qquad \overline{\Delta\epsilon}_t = \Delta\epsilon_t \min\left(1, \frac{\gamma}{\max(r_t, \varepsilon)}\right). \tag{14}$$

The primary configuration uses 50 deterministic DDIM steps. Guidance is evaluated at every reverse iteration from 25 through 49, with $\beta_{\text{mode}} = 112.5$, $\beta_{\text{var}} = 60$, and $\gamma = 0.5$. For every variation gap sample, a no-gap-guidance trajectory with the same class and initial noise provides same-time reference features. At iterations 45–49, a constrained repair is attempted after the guided feature enters both the detected bin and its inner interval. The main interval loss has zero off-axis penalty; orthogonal correction is confined to this repair. Its direction is normalized to $0.01\|\epsilon_t\|_2$ before backtracking. A candidate is accepted only when it remains inside both intervals and reduces displacement orthogonal to the target axis. The mode gap does not use this repair. Every allocated sample enters the synthetic set without post-generation rejection. In the audited pool the repair is active at the 90 configured step slots; the two interval conditions are met at 64 of them, and every one of those 64 candidates passes the acceptance test, so the backtracking search never exhausts its scales. The remaining 26 slots are skipped because the sample has not yet entered the target or control region, or carries no off-axis displacement left to reduce.

E Experimental details

Dataset and comparison protocol. We use the official training and validation partitions for each source benchmark. Exact class lists, split manifests, and corruption settings will be released with the configuration files. Random uses the same generator, reverse schedule, and initial-noise distribution as GTS, with the gap gradient disabled. Additive comparisons retain the full real training set and match the total synthetic budget. Untargeted methods use the nominal K images per class, while GTS redistributes the same total budget using the allocation in Sec. 4.2. ImageIDC is the ten-class ImageNet subset of Kim et al. (2022), used to test a less dog-centric class structure; its WordNet IDs are `n02869837`, `n01749939`, `n02488291`, `n02107142`, `n13037406`, `n02091831`, `n04517823`, `n04589890`, `n03062245`, and `n01773797`.

Distribution-shift protocol. For Tab. 3, each checkpoint is trained on clean ImageWoof2 and evaluated without adaptation; only the evaluation distribution changes. All evaluation sets use the same ten ImageWoof2 WordNet IDs and label order. These IDs are `n02086240`, `n02087394`, `n02088364`, `n02089973`, `n02093754`, `n02096294`, `n02099601`, `n02105641`, `n02111889`, and `n02115641`. ImageNet-Sketch-10 filters ImageNet-Sketch to these classes, and the two ImageNet-V2-10 sets similarly filter the Threshold0.7 and TopImages variants. ImageWoof2-C applies the 15 ImageNet-C corruption types at severities 1–5 to the ImageWoof2 validation images after resizing the short side to 256 and center-cropping to 224×224; its reported average pools all corruption–severity cells. The same normalization and evaluation transform are used for every method, and no shifted image is used for training, model selection, or gap diagnosis.

Student training and budget grids. We evaluate ConvNet-6, ResNet-AP-10, and ResNet-18 (He et al., 2016). The main multi-architecture comparison uses $K \in \{10, 100, 200, 450, 1000\}$, adding exactly $K_{\text{tot}} = 10K$ images on ImageWoof2. Synthetic datasets are frozen before student training so that all student architectures consume the same images. Hyperparameters, augmentations, epoch count, and evaluation preprocessing are held fixed within each architecture. The best mean result in a comparison block is typeset in bold. Fig. 1 additionally reports Random sampling and GTS at $K = 5000$ for all three architectures; this extension is not part of the main multi-method comparison.

Target attainment and mode-deletion metrics. We extract DINOv2 features from each generated image after sampling. A mode gap hit falls within the assigned mode's real-feature 90th-percentile radius around its anchor; a variation gap hit falls in its assigned shared-axis bin. Guided and unguided runs use paired classes, targets, and initial noises. The mode-deletion experiment reports accuracy on the full evaluation set and on the subset affected by the deleted training modes.

Generation settings. Real images are represented by frozen DINOv2 ViT-B/14 CLS features. The mode gap diagnostic uses cosine distance, while the variation gap uses standardized coordinates on shared variation axes estimated from pooled class-centered residuals. In the primary configuration, each class uses ten local modes, and the variation gap diagnostic inspects 12 frozen PCA axes. The two diagnostic masses are calibrated before quota allocation. We use a 50-step DDIM sampler and compute guidance through the full VAE-decoder and frozen-DINOv2 path. Every allocated image enters the frozen synthetic set without post-generation rejection. Apps. B–D give the gap

thresholds, allocation rules, guidance schedule, and stabilization details. All generation settings, student configurations, seeds, and dataset manifests will be released with the code.

F MULTI-ARCHITECTURE BUDGET SCALING

Fig. 5 visualizes the mean-accuracy trends for Random, the four prior diffusion-based methods, and GTS at the five budgets reported in Tab. 1; the table provides the corresponding standard deviations. The GTS curve lies above every other method at all five budgets for each of the three student architectures, and the ordering is established at the smallest budget rather than emerging only once the budget grows.

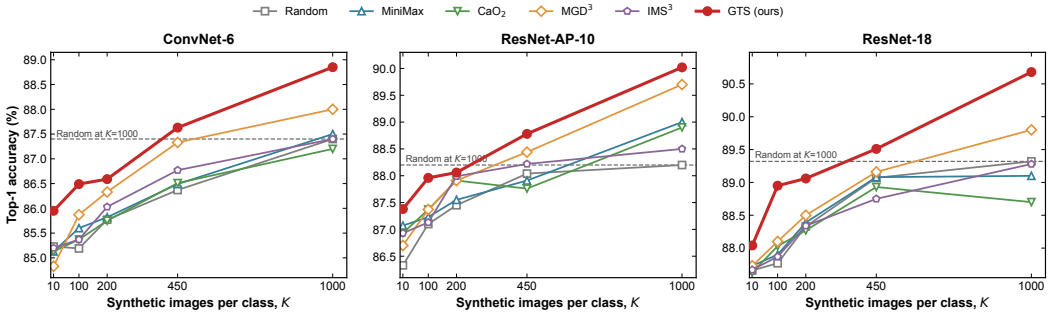

Figure 5: ImageWoof2 top-1 accuracy as the number of added synthetic images per class increases. Lines reproduce the means at $K \in \{10, 100, 200, 450, 1000\}$ from Tab. 1; the corresponding standard deviations over three student seeds are reported in the table. Every method retains the complete real training set and adds the same total synthetic budget at each K. Dashed horizontal lines mark Random accuracy at $K = 1000$.

G CONTROLLED MODE DELETION

This controlled test asks whether our method can find and replenish a class mode that is missing from the available training data. We select one or two intra-class clusters and remove all training samples belonging to the selected clusters before gap analysis. The detected-anchor variant must infer its targets from the remaining real images; the oracle-anchor variant is given the centers of the deleted clusters, separating target detection from the subsequent generation and student-training stages. Both variants add a total of 1,000 synthetic images (nominal $K = 100$) to the remaining real training set. We compare them with the remaining-real and Random controls, as well as a full-real reference that retains the removed images. The affected evaluation subset consists of held-out images whose nearest neighbor in the full real training set belongs to a deleted cluster. Tab. 5 reports ResNet-18 accuracy on both the full evaluation set and this affected subset.

Deleting the clusters costs 4.03 points on the affected subset. Adding the same budget without guidance recovers 0.23 of them, the detected-anchor variant 1.73, and the oracle-anchor variant 2.19, so roughly 43% and 54% of the deleted mode's contribution is restored when the budget is aimed at it. The detected-anchor variant runs without any knowledge of what was removed and recovers most of what the oracle-anchor variant does, which indicates that the gap diagnostic locates a missing mode from the remaining real images rather than depending on privileged information.

Table 5: Controlled mode-deletion evaluation on ImageWoof2 with ResNet-18 and $K = 100$ (1,000 added synthetic images in total for each augmentation method). Full-set and affected-subset top-1 accuracy assess downstream utility. Entries are mean $\pm$ standard deviation over three student seeds.

Training data	Full eval. acc.	Affected acc.
Remaining real	$85.65_{\pm 0.10}$	$84.23_{\pm 0.17}$
Remaining + Random	$86.21_{\pm 0.31}$	$84.46_{\pm 0.81}$
Remaining + detected-anchor GTS	$86.77_{\pm 0.17}$	$85.96_{\pm 0.22}$
Remaining + oracle-anchor GTS	$87.17_{\pm 0.24}$	$86.42_{\pm 0.44}$
Full real	$87.00_{\pm 0.32}$	$88.26_{\pm 0.33}$

H TRAINING-SET REGIMES

Tab. 6 distinguishes common training-set regimes. Here, D denotes a reference number of images per class (IPC). The experiments in the main paper use the additive regime: the complete real training set is retained and synthetic images are added to it.

Table 6: Training-set regimes used to separate total-data budget from real and synthetic contributions. D denotes a reference IPC, and the additive regime is the setting used in the main experiments.

Setting	Total IPC	Real IPC	Synthetic IPC	Purpose
Real-only	D	D	0	Full-real reference
Synthetic-only	D	0	D	Stand-alone utility of synthetic or distilled data
Fixed-budget mix	D	$D/2$	$D/2$	Complementarity under a matched total budget
Additive mix	$2D$	D	D	Gain from adding synthetic data to the complete real set
Real-subset baseline	$D/2$	$D/2$	0	Contribution of the real half of the fixed-budget mix
Synthetic-subset baseline	$D/2$	0	$D/2$	Contribution of the synthetic half of the fixed-budget mix

Tab. 7 reports the synthetic-only comparison that motivated this choice. Its Full real accuracies reproduce the three-student-seed means of Tab. 1, rounded to one decimal. Every other accuracy in that table comes from a single archived run reported as a point estimate, so no standard deviation accompanies it.

Table 7: Auxiliary ImageWoof2 synthetic-only results at IPC 1,000. DINO and CLIP diversity characterize each training set. Full real accuracies reproduce the three-seed means in Tab. 1, rounded to one decimal. Synthetic-method accuracies are single-run point estimates (%). Bold marks the strongest synthetic method in each column.

Training data	DINO diversity	CLIP diversity	ConvNet-6	ResNet-AP-10	ResNet-18
Full real (9,025 images)	0.6116	0.2743	84.2	85.8	87.0
DiT direct	0.3670	0.1476	68.9	73.1	75.1
MiniMax	0.3441	0.1525	75.1	77.5	78.3
MGD^3	**0.4522**	**0.1988**	**81.1**	**82.7**	**84.4**
CaO_2	0.3325	0.1404	70.0	73.2	75.3
IMS^3	0.3773	0.1553	71.7	74.4	76.5

At IPC 1,000 the ten-class synthetic-only set contains 10,000 images, already 975 more than the complete real training set. The strongest synthetic method, MGD^3, still trails the full-real reference by 3.1, 3.1, and 2.6 percentage points on ConvNet-6, ResNet-AP-10, and ResNet-18, and its DINO and CLIP diversity remain well below the real set. Increasing synthetic cardinality beyond real-set cardinality therefore leaves a gap in the information and coverage that the real data carries. Dataset distillation remains a well-posed low-IPC compression problem under this reading, and what the comparison bounds is the use of synthetic count as a proxy for effective coverage. When the complete real set is available, a synthetic replacement that is larger yet less accurate offers neither compression nor improved utility, which is what motivates the additive setting: we retain every real example and spend only a marginal synthetic budget on diagnosed under-covered regions.

I QUALITATIVE INSPECTION OF SHARED VARIATION AXES

To examine whether the PCA basis fitted to pooled class-centered residuals captures recurring within-class semantics, we visualize the two endpoints of a high-confidence shared axis. Each row in Fig. 6 is an anonymized class; columns A and B contain four real images nearest the two endpoint regions of the same axis. The A/B orientation is arbitrary because the sign of a PCA axis is arbitrary. A consistent transition across class rows provides qualitative evidence that the direction captures variation shared across classes. This visualization validates the shared-axis construction; it does not by itself assert that any particular class contains a variation gap.

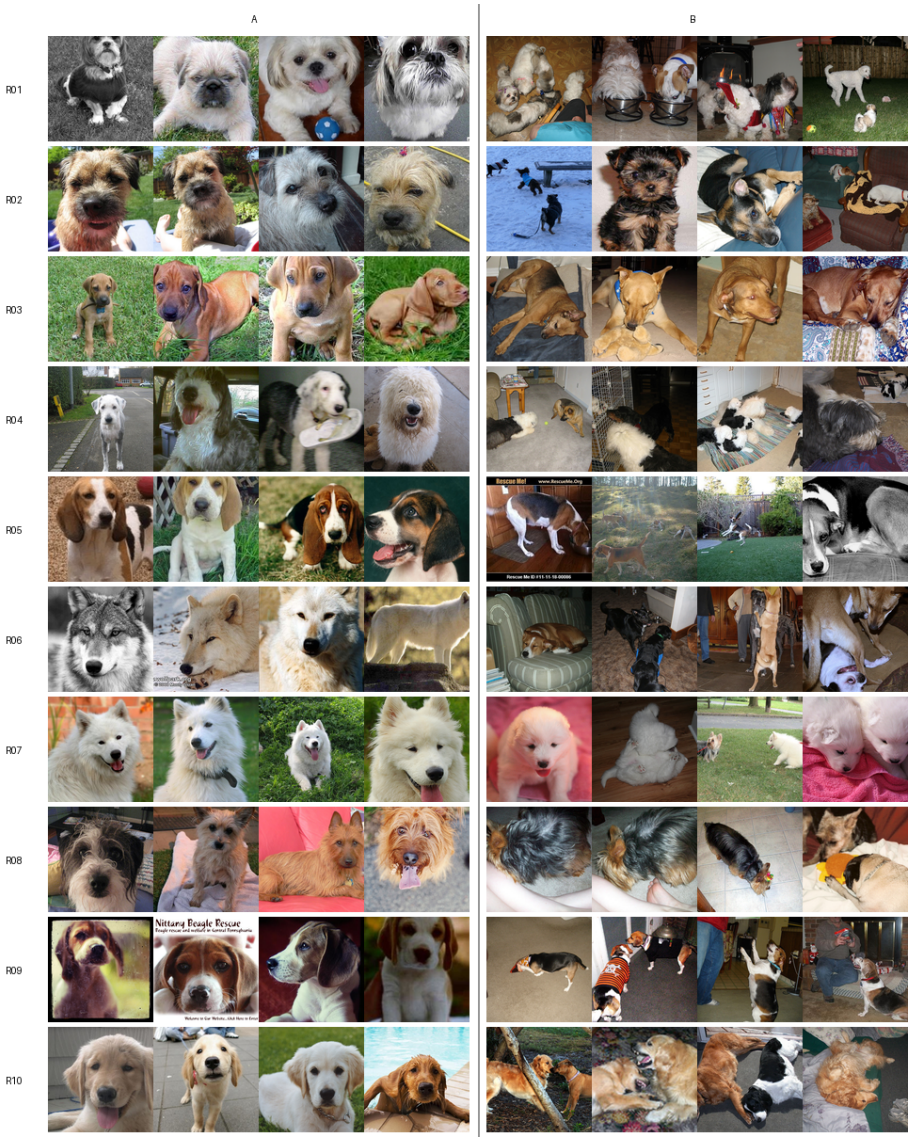

Figure 6: Cross-class shared variation along PCA axis 6. Endpoint A emphasizes isolated close-up portraits, whereas endpoint B emphasizes social interaction, daily-life context, and activity or resting scenes. The consistent portrait-to-context transition across class rows supports its interpretation as a shared within-class variation direction.

J QUALITATIVE COMPARISON OF GAP-GUIDED SAMPLES

This qualitative comparison shows images generated for ImageWoof2 at $K = 450$ with and without gap guidance. Each pair in Fig. 7 uses the same class label, quota position, and initial noise, so the two images differ only in whether the gap-guidance correction is applied during sampling. Both arms produce recognizable images of the labelled class, and the guided column reaches a different point within that class rather than a different class, which is the intended effect of a within-class correction: the guidance moves a sample toward its assigned target while the class label it was conditioned on is retained.



Figure 7: Paired ImageWoof2 samples generated without gap guidance (left) and with GTS gap guidance (right) at $K = 450$. Corresponding columns share the same initial noise and quota position. The class is marked at the left of each row.

K GENERATION COST

Gap guidance leaves the sampling schedule untouched and adds computation only inside the guided steps. As stated in Sec. 5, both arms run 50 deterministic DDIM steps with a single DiT evaluation per reverse step; we verify this from the per-image sampler audit written by the generator, where every record reports `reverse_steps` $= 50$, `dit_forward_count` $= 50$, and `nfe_per_reverse_step` $= 1.0$ in both arms. The measured cost difference therefore isolates the guidance path, namely the VAE decode and the DINOv2 forward and backward pass evaluated on the 25 guided steps.

Table 8 reports wall-clock time per image at $K = 100$ on a single RTX 3080 Ti with batch size 1, separated by the guidance objective a sample carries. The per-image cost is set by that objective rather than by the dataset: across the three datasets the unguided entries agree to within 0.04 s and the guided ones to within 0.45 s.

Two features of the table are worth stating explicitly. First, variation gap sampling is slightly more expensive than mode gap sampling in the guided arm, because the variation objective also evaluates the shared-axis coordinate and may invoke the constrained repair. Second, the unguided arm is itself more expensive for variation gap rows. This does not reflect a different sampler: the unguided arm runs with $\beta = 0$ in both cases. For variation gap rows it additionally records a reference trace, extracting and storing one frozen DINOv2 feature at each of the 25 guided step indices, so that the guided sample can later be compared against its own unguided trajectory. That bookkeeping accounts for the difference and is not part of the sampling cost.

Taking the realised mixture of objectives into account, generating a $K = 1000$ pool of $10{,}000$ images requires roughly 22 GPU-hours with guidance, compared with 7 GPU-hours without guidance, on a single RTX 3080 Ti. Guidance is applied once during dataset construction and adds no overhead to subsequent student training whenever a generated pool is reused across architectures and seeds.

Table 8: Wall-clock generation cost per image at $K = 100$, batch size 1, on a single RTX 3080 Ti, separated by guidance objective. Both arms use the same 50-step DDIM schedule and the same number of DiT evaluations per image. The unguided variation gap column includes writing the reference trace described in the text.

	mode gap (s)		variation gap (s)	
Dataset	Unguided	Guided	Unguided	Guided
ImageNette	2.00	7.73	3.38	8.46
ImageIDC	2.01	7.76	3.39	8.63
ImageWoof2	2.01	7.70	3.35	8.18

The guidance path also has a limited memory footprint at batch size 1, as it retains only one decoded image and one encoder activation graph at a time.