TAIL-PRESERVING DIFFUSION VIA GEOMETRIC REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-conditional diffusion models trained on long-tailed data often generate minority-class samples that drift toward head-class semantics. We study this failure through the distinction between conditional geometry preservation and marginal coverage control. Our analysis shows that positive class-only reweighting preserves the population-optimal conditional denoising target, although it can alter shared-parameter optimization. Sampling-prior adjustment changes the marginal mixture while leaving each conditional distribution fixed. Under a local manifold model, we further bound deterministic reverse-trajectory departure in terms of initial distance and normal velocity, motivating geometric control of semantic drift. Guided by this analysis, we propose *Geometry-aware Tail-preserving Score Regularization* (GTSR). GTSR retains the standard denoising objective and adds a tail-specific regularizer on denoised estimates in a frozen semantic representation. A real-sample memory bank and rank-adaptive kNN-PCA define a local class reference, and the regularizer penalizes the normal-displacement energy fraction as a practical geometric proxy. Low-noise activation selects estimates that support reliable comparison with this reference. Evaluation on CIFAR10-LT, CIFAR100-LT, and Tiny-ImageNet-LT shows improved tail conditional accuracy, conditional fidelity, and semantic isolation, with consistent gains under DDPM and EDM parameterizations. These findings support geometry-preserving training complemented by sampling-time coverage control.

1 INTRODUCTION

Class-conditional diffusion models have become a standard approach for high-quality, controllable image generation (Ho et al., 2020; Song et al., 2021b; Dhariwal & Nichol, 2021). Their performance, however, depends on the availability of representative training data for each class. Under long-tailed distributions, minority classes provide limited support for learning class-specific structure, and their generated samples often drift toward visually or semantically similar head classes (Qin et al., 2023; Yan et al., 2024; Zhang et al., 2024; Hong et al., 2026). This headward semantic leakage compromises conditional fidelity even when tail classes are explicitly requested.

A common strategy for addressing imbalance is to increase the contribution of rare examples through loss reweighting or resampling (Cui et al., 2019; Kang et al., 2020; Qin et al., 2023). In diffusion models, these interventions affect shared denoising dynamics across classes and noise levels. Increasing the influence of tail examples can therefore change both the frequency of tail-like outputs and the fidelity of their class-specific structure. Existing observations suggest that training-time rebalancing can improve apparent tail coverage while degrading conditional fidelity and generation stability (Yan et al., 2024; Zhang et al., 2024). This trade-off motivates a closer examination of how imbalance remedies interact with the geometry learned from sparse tail data.

We examine this interaction by distinguishing the population denoising target from finite-sample learning dynamics. Positive class-only weights preserve the pointwise population-optimal conditional denoising target. In practical networks, shared parameters couple the learning of different classes, and reweighting changes how their gradient contributions influence optimization. With sparse tail support, estimation uncertainty can be direction-dependent: deviations away from locally supported class structure may remain weakly constrained by the available observations. We hypothesize that emphasizing these uncertain directions through reweighting can destabilize tail-class denoising and

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.

promote headward semantic drift. This perspective connects the observed fidelity–coverage trade-off to finite-sample geometry and shared-parameter optimization.

The resulting design principle is to *protect conditional geometry during training and adjust marginal coverage at sampling time*. Conditional geometry governs how faithfully generated samples follow the requested class, while the sampling prior controls how frequently each class is requested. Changing the prior adjusts the marginal mixture while preserving each learned conditional distribution. It consequently provides a separate mechanism for coverage control, with the quality of tail-conditioned generation depending on the learned denoising behavior.

Our geometric motivation further identifies which deviations warrant attention. In an idealized local manifold model, controlling the normal component of a reverse trajectory's velocity bounds its departure from the target manifold over a segment initialized nearby. When other class regions are sufficiently separated, this distance control supports semantic isolation. The analysis motivates regularizing departures from class-local structure while retaining flexibility along directions supported by within-class variation.

Guided by this principle, we propose *Geometry-aware Tail-preserving Score Regularization* (GTSR). GTSR retains the standard denoising objective and adds a geometric regularizer for tail classes. Using real training samples, it constructs a local class reference in a frozen semantic representation and penalizes the fraction of denoised-feature displacement energy orthogonal to the estimated class-local subspace. This displacement-based proxy translates the geometric motivation into a practical training objective. The regularizer focuses on low-noise timesteps, where denoised estimates support reliable comparisons with the real-sample reference. Together, tail-specific regularization and low-noise activation target sparsely supported classes where geometric supervision is informative. We further separate marginal tail coverage from conditional fidelity in evaluation, distinguishing training-time geometry preservation from sampling-time coverage control. Experiments on standard long-tailed image benchmarks show that GTSR improves tail fidelity and reduces headward semantic leakage while maintaining competitive coverage.

2 PRELIMINARIES

2.1 LONG-TAILED CONDITIONAL GENERATION

Let $\mathcal{D} = \{(x_0, y)\}$ denote a long-tailed dataset, where $y \in \mathcal{Y}$ is a class label and the empirical class prior $\hat{p}(y)$ is highly imbalanced. We denote the disjoint tail and head sets by $\mathcal{Y}_{\text{tail}}, \mathcal{Y}_{\text{head}} \subset \mathcal{Y}$, respectively, and write $\mathcal{Y}_{\text{mid}} = \mathcal{Y} \setminus (\mathcal{Y}_{\text{tail}} \cup \mathcal{Y}_{\text{head}})$ for the remaining classes. Our goal is class-conditional generation that simultaneously satisfies: (i) high within-class fidelity, (ii) sufficient coverage for tail classes, and (iii) semantic isolation, meaning that the samples generated for a tail class do not drift toward head-class semantics.

2.2 CONDITIONAL DIFFUSION MODELS

We consider a class-conditional diffusion model with T timesteps. The forward noising process maps a clean sample x_0 to a noisy observation x_t via

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I), \tag{1}$$

where $\{\bar{\alpha}_t\}_{t=1}^{T}$ is a fixed noise schedule. The model is trained using the standard DDPM noise-prediction objective

$$\mathcal{L}_{\text{ddpm}}(\theta) = \mathbb{E}_{(x_0,y)\sim\mathcal{D},\, t,\, \varepsilon}\big[\|\varepsilon - \epsilon_\theta(x_t, t, y)\|_2^2\big], \tag{2}$$

where $\epsilon_\theta(x_t, t, y)$ denotes the noise-prediction network. Its associated score estimate is $-\epsilon_\theta(x_t, t, y)/\sqrt{1 - \bar{\alpha}_t}$. Under this parameterization, the score field governs the reverse diffusion dynamics and, consequently, the geometry of the conditional generation distribution $p_\theta(x_0 \mid y)$.

2.3 A GEOMETRIC PROXY: TANGENT AND NORMAL DIRECTIONS

For a fixed class y, we represent its local geometry in the frozen feature space of a pretrained encoder $\phi(\cdot)$. Given $z = \phi(x)$, PCA over nearby real features from the same class defines an empirical tangent subspace, whose orthogonal complement serves as a normal subspace. The former captures

locally supported within-class variation, while the latter measures departure from the local class geometry. The accuracy of this local tangent estimate depends on neighborhood scale and sample support (Aamari & Levrard, 2019). This empirical decomposition serves as a local geometric proxy.

3 THEORETICAL ANALYSIS

We analyze the distinct roles of training-time reweighting and sampling-time prior adjustment, then develop a local geometric motivation for preserving tail-class semantics.

3.1 CLASS-ONLY REWEIGHTING: POPULATION TARGET VERSUS OPTIMIZATION DYNAMICS

Class-only reweighting increases the contribution of underrepresented classes during training by assigning each class y a positive loss weight $w(y)$, typically larger for rarer classes. Multiplying each example's denoising loss in Eq. (2) by its class weight yields

$$\mathcal{L}_w(\theta) = \mathbb{E}_{(x_0,y)\sim\mathcal{D},\, t,\, \varepsilon}\left[w(y)\,\|\varepsilon - \epsilon_\theta(x_t, t, y)\|_2^2\right], \tag{3}$$

where $w(y) > 0$ depends only on the class label and the sampling distribution remains unchanged.

Pointwise population target. Conditioning on (x_t, t, y) makes $w(y)$ a positive constant, so it scales the conditional regression loss without changing its minimizer:

$$\begin{aligned}\epsilon_w^*(x_t, t, y) &= \arg\min_h \mathbb{E}\left[w(y)\|\varepsilon - h\|_2^2 \mid x_t, t, y\right] \\ &= \mathbb{E}[\varepsilon \mid x_t, t, y] = \epsilon^*(x_t, t, y),\end{aligned} \tag{4}$$

where ϵ^* denotes the pointwise population minimizer of the unweighted DDPM objective. Thus, positive class-only reweighting preserves the conditional regression target. Appendix C provides the derivation.

Why reweighting can still hurt in practice. Finite-capacity models trained on long-tailed data introduce parameter-sharing and sample-support effects. In particular, the noise-prediction network shares parameters across classes, timesteps, and regions of the input space, while tail classes are learned from substantially fewer samples. Reweighting changes how optimization allocates gradient budget across these shared parameters, effectively emphasizing sparse tail regions during training. Its practical effect is therefore expressed through finite-sample estimation and shared-parameter optimization dynamics.

Finite-sample geometric uncertainty. Sparse tail support can increase uncertainty in the learned conditional denoiser, with its directional structure depending on the data and shared model. We hypothesize that deviations from locally supported class structure are weakly constrained and that reweighting can amplify their influence during optimization. The unchanged population target therefore coexists with possible changes in learned tail-class geometry. Sections 5.3 and 5.4 examine this hypothesis through leakage dynamics and geometric diagnostics.

3.2 SAMPLING-TIME PRIOR SHIFT

In contrast, modifying the sampling prior $\pi(y)$ changes only the marginal distribution

$$p_\theta^\pi(x_0) = \sum_y \pi(y)\, p_\theta(x_0 \mid y), \tag{5}$$

while leaving the trained conditional noise-prediction network $\epsilon_\theta(x_t, t, y)$ unchanged. Thus, inference-time prior adjustment can control class frequency without directly modifying the learned conditional geometry. This distinction motivates controlling requested-label frequencies through the sampling prior and preserving class-conditional geometry during training.

3.3 IDEALIZED GEOMETRIC MOTIVATION

The preceding analysis motivates controlling motion away from the local class geometry. The following proposition describes this effect for a deterministic reverse trajectory in semantic space.

Proposition 3.1 (Normal-direction suppression and semantic isolation). *Let $\mathcal{M}_y$ be a smooth embedded class manifold and let $z(\tau)$, $\tau \in [0, S]$, remain in a tubular neighborhood with a unique nearest-point projection onto $\mathcal{M}_y$. Decompose its complete velocity at the projected point as*

$$\dot{z}(\tau) = F(z(\tau), \tau, y) = F_T(z(\tau), \tau, y) + F_N(z(\tau), \tau, y), \tag{6}$$

where F_T and F_N are the tangent and normal components, respectively. Write $d(\tau) = \mathrm{dist}(z(\tau), \mathcal{M}_y)$ and assume

$$\|F_N(z(\tau), \tau, y)\|_2 \le C_\tau \delta + \kappa d(\tau), \qquad 0 \le C_\tau \le C, \quad \delta, \kappa \ge 0. \tag{7}$$

Then $d(\tau) \le e^{\kappa\tau} d(0) + C\delta(e^{\kappa\tau} - 1)/\kappa$, with the continuous limit $d(\tau) \le d(0) + C\delta\tau$ at $\kappa = 0$. If $d(0) = O(\delta)$, the trajectory remains within an $O(\delta)$ tube on this fixed local segment. If competing-class regions are separated from $\mathcal{M}_y$ by a distance greater than this bound, the segment stays outside those regions.

Proposition 3.1 supplies an idealized geometric motivation in terms of the complete reverse velocity. For a probability-flow trajectory, this velocity includes the schedule-dependent drift and, in feature space, the encoder Jacobian. GTSR implements a training-time displacement proxy, defined from the denoised estimate and real class features in Sec. 4. Its empirical connection to semantic drift is examined through trajectory, confusion, and displacement-pair analyses in Sec. 5.4 and Appendix B.7. Appendix D provides the local distance argument.

4 METHOD

4.1 OVERVIEW AND DESIGN PRINCIPLE

Guided by Sec. 3, GTSR retains the standard DDPM denoising objective and adds a geometric regularizer to discourage tail-class semantic drift:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{ddpm}}(\theta) + \lambda_N \mathcal{L}_N(\theta), \tag{8}$$

where λ_N controls the regularization strength. In a frozen feature space, $\mathcal{L}_N$ penalizes the fraction of displacement energy orthogonal to the estimated local class subspace. The displacement is measured from the denoised feature to the target-class prototype, with both the prototype and local subspace estimated from real samples.

The regularization is restricted to tail-class examples at low-noise timesteps, where denoised estimates support reliable comparisons with the class reference. The neighborhood size and tangent rank adapt to the number of unique real samples available in each class memory bank. The full training procedure is provided in Appendix E.

4.2 GEOMETRY-AWARE TAIL SCORE REGULARIZATION

Tail-only activation and low-noise gating. The regularizer is applied exclusively to samples whose labels belong to the tail-class set $\mathcal{Y}_{\mathrm{tail}}$. Furthermore, it is activated only at timesteps $t \le t_0$, corresponding to the low-noise regime of the forward diffusion process. Coarse layout and broad class identity can emerge early in the reverse trajectory. The gate selects the later, low-noise regime in which the denoised estimate $\hat{x}_0$ supports more reliable feature-space comparison with the real-data reference. The kNN-PCA basis itself is estimated around the real class prototype. Appendix B.14 relates this choice to the noise-prediction error and the existing gate ablation.

Representation-space geometry with frozen encoder. All geometric operations are performed in a representation space induced by a fixed, pretrained encoder $\phi(\cdot)$. Compared with pixel space, this representation provides a more stable notion of semantic similarity and class-conditional neighborhood structure. The parameters of ϕ are fixed throughout training, while gradients through $\phi(\hat{x}_0)$ to $\hat{x}_0$ and the denoiser are retained. Memory-bank features, prototypes, neighborhood selection, and PCA bases are stop-gradient references. This design decouples representation learning from score learning and prevents feedback effects that could destabilize training.

Per-class memory bank and prototype. For each class y, we maintain a memory bank $\mathcal{B}_y$ containing feature embeddings of real clean samples $z_{\mathrm{real}} = \phi(x_0)$. The memory bank serves as a local reference

for estimating class-conditional geometry. In addition, we maintain a class prototype μ_y, updated using an exponential moving average,

$$\mu_y \leftarrow m\,\mu_y + (1-m)\cdot \mathrm{Mean}\Big(\Big\{z^{(i)}_{\mathrm{real}} : y^{(i)} = y\Big\}\Big), \tag{9}$$

where $m \in [0,1)$ is the prototype momentum and the mean uses current minibatch examples with label y. The prototype provides a stable class-level reference in feature space. Using only real samples in this construction prevents self-reinforcing drift from model-generated estimates.

Rank-adaptive local tangent estimation. A potential difficulty is that the rarest classes may have fewer unique real samples than the nominal neighborhood size k. We therefore treat the hyperparameters k and r as nominal upper bounds. Let n_y denote the number of *unique* real training images from class y currently represented in $\mathcal{B}_y$, with each sample identity contributing once to the count. For class y, we use

$$k_y = \min(k, n_y), \qquad r_y = \min(r, \max(k_y - 1, 0), d), \tag{10}$$

where d is the feature dimension. If $k_y < 2$, the class does not provide enough local geometry and its geometric loss is skipped for that update. Otherwise, we select the k_y nearest unique real features to the prototype: $\mathcal{N}_y = \mathrm{kNN}(\mu_y, \mathcal{B}_y, k_y)$, and compute an orthonormal basis

$$U_y = \mathrm{PCA}_{r_y}(\mathcal{N}_y) \;\in\; \mathbb{R}^{d\times r_y}. \tag{11}$$

The effective rank is therefore never larger than what can be supported by the unique real samples. We also use a reliability weight

$$\rho_y = \min\Big(1, \frac{r_y}{r}\Big), \tag{12}$$

which assigns a smaller relative contribution to severely rank-limited examples within an active minibatch. We interpret $\mathrm{span}(U_y)$ as a local tangent proxy and its orthogonal complement as a normal proxy. The basis dimension and weight therefore reflect the support available in the real-data memory.

Denoised estimate and displacement decomposition. For a training sample $(x_0^{(i)}, y^{(i)})$ at timestep $t^{(i)}$, we compute the predicted noise $\hat{\varepsilon}^{(i)} = \epsilon_\theta(x_t^{(i)}, t^{(i)}, y^{(i)})$ and obtain the standard DDPM denoised estimate

$$\hat{x}_0^{(i)} = \frac{x_t^{(i)} - \sqrt{1-\bar{\alpha}_{t^{(i)}}}\,\hat{\varepsilon}^{(i)}}{\sqrt{\bar{\alpha}_{t^{(i)}}}}, \qquad z^{(i)} = \phi(\hat{x}_0^{(i)}). \tag{13}$$

For tail classes, we define the displacement from the denoised estimate to the class prototype

$$u^{(i)} = \mu_{y^{(i)}} - z^{(i)}, \tag{14}$$

and decompose it into tangential and normal components:

$$u_T^{(i)} = U_{y^{(i)}} U_{y^{(i)}}^\top u^{(i)}, \qquad u_N^{(i)} = u^{(i)} - u_T^{(i)}. \tag{15}$$

4.3 Normal-direction regularizer

Let $\mathcal{I}_{\mathrm{tail}}$ denote the set of tail-class samples in the current minibatch that satisfy the low-noise condition:

$$\mathcal{I}_{\mathrm{tail}} = \{\, i : y^{(i)} \in \mathcal{Y}_{\mathrm{tail}} \;\wedge\; t^{(i)} \le t_0 \;\wedge\; k_{y^{(i)}} \ge 2 \,\}. \tag{16}$$

For each $i \in \mathcal{I}_{\mathrm{tail}}$, we compute $u_N^{(i)}$ as defined in Eq. (15) using the rank-adaptive basis $U_{y^{(i)}}$. The normal-direction regularizer is

$$\mathcal{L}_N(\theta) = \frac{1}{\max\big(\eta, \sum_{i\in\mathcal{I}_{\mathrm{tail}}} \rho_{y^{(i)}}\big)} \sum_{i\in\mathcal{I}_{\mathrm{tail}}} \rho_{y^{(i)}} \frac{\|u_N^{(i)}\|_2^2}{\|u^{(i)}\|_2^2 + \eta}, \tag{17}$$

where $\eta > 0$ is a small constant for numerical stability. When $\mathcal{I}_{\mathrm{tail}} = \varnothing$, we set $\mathcal{L}_N = 0$ for that minibatch.

The ratio in Eq. (17) measures the fraction of displacement energy that lies in the estimated normal subspace, while ρ_y reflects how much empirical tangent rank is available for class y. The rank-adaptive basis restricts each local model to the directions supported by its real samples. The normalized loss favors a smaller normal-displacement energy fraction and assigns no separate penalty to tangential distance from the prototype. Its denominator couples the two components, so the optimization acts on their relative energies. The DDPM term remains active for every training example, and the diversity diagnostics in Appendix B.2 evaluate the resulting within-class variation.

Method	CIFAR10-LT			CIFAR100-LT		
	TAcc↑	cFID↓	Leakage↓	TAcc↑	cFID↓	Leakage↓
Vanilla / Prior Shift	0.1524±0.0021	1.5513±0.0284	0.3244±0.0062	0.0821±0.0017	3.9401±0.0516	0.4120±0.0083
Reweighting	0.1540±0.0024	2.5306±0.0413	0.3090±0.0071	0.0949±0.0020	4.8710±0.0662	0.4360±0.0096
Balanced Sampling	0.1580±0.0026	2.3140±0.0375	0.3015±0.0060	0.0970±0.0022	4.3103±0.0584	0.4017±0.0091
CFG	0.1661±0.0027	1.6800±0.0330	0.2985±0.0061	0.1013±0.0024	3.7611±0.0493	0.3655±0.0080
CORAL	0.1684±0.0023	1.5638±0.0309	0.2716±0.0057	0.1121±0.0021	3.6184±0.0487	0.3369±0.0077
CCUA	0.1691±0.0025	1.6047±0.0328	0.2869±0.0060	0.1134±0.0023	3.7026±0.0515	0.3498±0.0081
GTSR (Ours)	**0.1710±0.0020**	**1.5102±0.0347**	**0.2420±0.0052**	**0.1160±0.0019**	**3.5271±0.0460**	**0.3210±0.0072**

Table 1: Main results on CIFAR10-LT and CIFAR100-LT (IR=100), reported as mean ± standard deviation over five runs. Conditional metrics use the same sample budget per class and the same evaluation classifier. TAcc denotes exact tail conditional accuracy. Vanilla and Prior Shift share the conditional generator.

5 Experiments

5.1 Experimental setup

Datasets, backbone, and training. We evaluate CIFAR10-LT, CIFAR100-LT, and Tiny-ImageNet-LT with exponentially decaying training class sizes ($\mathrm{IR} = 100$) and balanced test sets. We use a class-conditional U-Net backbone for DDPM (Ho et al., 2020) and the additional EDM (Karras et al., 2022) comparison, matching architecture, noise schedules, and optimization settings across methods within each formulation. DDPM uses fixed-step DDIM sampling and five runs paired across methods by seed, while EDM uses three runs. Experiments use two A100 GPUs, with configurations and model selection detailed in Appendix F.

Metrics and sampling. Using equal per-class generation budgets and a fixed pretrained classifier, we measure exact tail conditional accuracy (TAcc, ↑), Tail→Head leakage (↓), and tail-to-tail confusion (↓). We additionally report class-conditional FID averaged over tail classes ($\mathrm{cFID}_{\mathrm{tail}}$, ↓) (Heusel et al., 2017). Prior Shift keeps the generator fixed and assigns tail labels $\alpha > 1$ times the sampling weight of other labels, with marginal tail coverage measured by TCov_{π}. Appendix B.2 provides the metric definitions, including the classifier-based quantities in Eqs. (18)–(21).

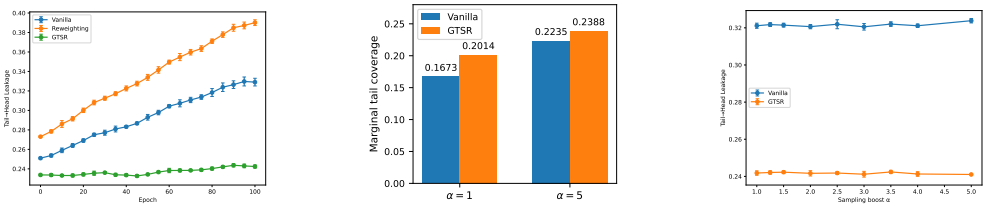
Baselines. We compare Vanilla (standard DDPM), Reweighting (class-weighted denoising losses) (Cui et al., 2019; Kang et al., 2020), Balanced Sampling (class-balanced minibatches) (Qin et al., 2023), Prior Shift, and classifier-free guidance (CFG) (Ho & Salimans, 2021). CORAL (Rodriguez et al., 2025) applies supervised contrastive regularization to U-Net bottleneck features, while CCUA (Chen et al., 2025) combines unsupervised contrastive learning with conditional–unconditional alignment. We report mean ± standard deviation over five runs, unless otherwise specified.

5.2 Main results

Table 1 shows that training-time rebalancing improves tail conditional accuracy while worsening tail cFID and leaving substantial Tail→Head leakage. CORAL and CCUA improve this trade-off, while GTSR achieves the highest mean TAcc and the lowest mean tail cFID and leakage. Prior Shift shares Vanilla's conditional performance, reflecting its role in marginal coverage control.

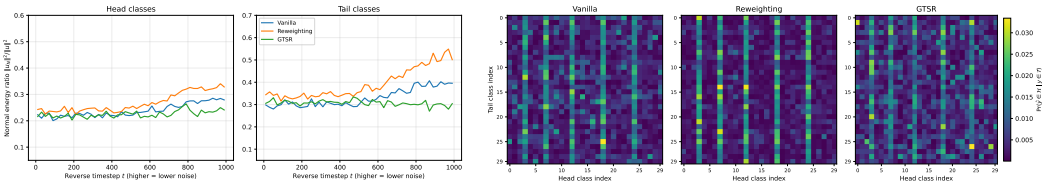
These findings are consistent with the finite-sample perspective in Sec. 3.1: increasing the contribution of sparse tail examples can leave class-specific geometry poorly constrained. GTSR complements the denoising objective with targeted regularization of deviations from locally supported tail-class structure. Its improved semantic isolation accords with the geometric motivation in Sec. 3.3.

Similar improvements hold on Tiny-ImageNet-LT and under EDM (Tables 15 and 17), supporting consistency across the evaluated datasets and diffusion parameterizations. Appendix B.2 provides complementary confusion, head/overall quality, and diversity diagnostics (Tables 7, 8, and 9). Appendix B.15 reports run-level effect sizes and conservative paired-inference bounds (Tables 21 and 22).



(a) (b) (c)

Figure 1: **CIFAR10-LT: training vs. inference effects.** (a) Tail→Head leakage vs. epoch. (b) Marginal tail coverage $\mathrm{TCov}_{\tilde{\pi}_\alpha}$ at boost factors $\alpha = 1$ and 5. (c) Tail→Head leakage vs. boost α.



(a) **Normal-displacement energy ratio**. (b) **Tail-to-head leakage structure**.

Figure 2: **Geometry evidence for leakage on CIFAR100-LT.**

5.3 Training dynamics of semantic leakage

The analysis in Sec. 3.1 suggests that Tail→Head leakage is linked to optimization-induced geometric instability in sparse tail regions, particularly in the low-noise regime where denoised estimates support reliable semantic comparison. To examine this effect, we track leakage on a held-out set throughout training.

Fig. 1(a) plots Tail→Head leakage over training on CIFAR10-LT. Training-time reweighting yields a clear monotonic increase, consistent with progressively accumulated head-biased attractors during optimization. In contrast, GTSR keeps leakage nearly flat, indicating that tail/low-noise normal suppression stabilizes the learned conditional behavior in these regions. We report the corresponding curves on CIFAR100-LT and Tiny-ImageNet-LT in Appendix B.9 and Appendix B.11.3, respectively. Similar leakage dynamics are also observed with an EDM backbone (Appendix B.12).

5.4 Geometry Analysis

Normal energy and semantic confusion. Fig. 2a shows that reweighting increases the tail normal-displacement energy ratio penalized in Eq. (17), especially at low noise, while GTSR lowers it. This contrast is consistent with the finite-sample perspective in Sec. 3.1: sparse tail observations leave some geometric directions weakly constrained, motivating the additional real-data reference used by GTSR. Correspondingly, Fig. 2b shows concentrated tail-to-head errors toward a few head classes under reweighting and reduced headward confusion under GTSR. Together, these observations associate lower relative normal displacement with improved semantic isolation. The normal-energy trend also holds under EDM (Fig. 9, Appendix B.12), while Appendix B.2 characterizes the remaining tail-to-tail errors.

Reverse trajectories. Table 2(a) shows that GTSR stays closer on average to the target tail prototype and maintains a larger minimum distance from head prototypes than Vanilla and Reweighting. This pattern accords with the semantic-isolation motivation in Sec. 3.3 and supports using real-class references to regularize headward drift. Representative trajectories, full statistics, and the association between normal-energy ratio and leakage are provided in Appendices B.5, B.6, and B.7.

Class-preserving versus cross-class displacement. Table 2(b) shows that class-preserving changes have substantially lower normal-energy fractions than tail-to-confusable-head changes. This contrast supports the semantic relevance of the local PCA decomposition used by GTSR to target relative

(a) Reverse-trajectory distances

Method	Avg. tail ↓	Min. head ↑
Vanilla	0.286	0.184
Reweighting	0.334	0.151
GTSR	**0.229**	**0.247**

(b) Semantic displacement

Image pair	r_N
Same-class neighbors	0.179
Class-preserving augmentation	0.214
Tail-to-confusable-head	0.659

Table 2: **Trajectory and semantic-displacement diagnostics.** (a) Average distance to the target tail prototype and minimum distance to head prototypes along reverse trajectories. (b) Normal displacement-energy fraction r_N for image pairs on CIFAR100-LT.

Variant	TAcc↑	Tail cFID↓	Leakage↓
GTSR (full)	**0.1160±0.0019**	**3.5271±0.0460**	**0.3210±0.0072**
$\lambda_N = 0$ (Vanilla)	0.0821±0.0017	3.9401±0.0516	0.4120±0.0083
w/o low-noise gating	0.1011±0.0024	3.8420±0.0524	0.3719±0.0085
w/o tail-only activation	0.0908±0.0020	3.7040±0.0496	0.3487±0.0081
GTSR reweighting	0.1096±0.0029	4.2900±0.0617	0.4198±0.0101

Table 3: **Ablation study on CIFAR100-LT (**$\mathrm{IR} = 100$**).** Mean ± standard deviation over five matched runs.

normal displacement. The reference rank adapts to available sample support, as described in Sec. 4.2. Pair definitions, sensitivity, and effective-rank diagnostics remain in Appendices B.13, B.8, and B.4.

5.5 Effect of inference-time prior shift

We vary the tail-label sampling weight while keeping the trained generator and conditional sampling procedure fixed. Figure 1(b–c) contrasts the reported increase in marginal tail coverage with nearly unchanged conditional leakage on CIFAR10-LT: Vanilla remains near 0.32, while GTSR remains near 0.24 across sampling boosts. This behavior agrees with Sec. 3.2: prior adjustment redistributes probability mass over requested labels while preserving each conditional distribution. Increasing the frequency of tail requests therefore leaves their class-conditioned semantic errors unchanged. GTSR retains its lower leakage across the tested priors, consistent with its training-time regularization of tail-class geometry. At a fixed prior, differences between Vanilla and GTSR reflect their learned conditional distributions, while coverage changes across sampling boosts within each model isolate the influence of the requested-label mixture. Together, these observations support the complementary roles of geometry preservation during training and marginal coverage control at inference. Appendices B.9 and B.11.3 provide additional sampling sweeps, while Appendix B.16 reports the coverage endpoints and their cross-metric consistency check.

5.6 Ablation Study

Table 3 evaluates GTSR's components on CIFAR100-LT, using matched Vanilla runs for the zero-weight control. Removing the geometric term increases headward leakage, while regularizing all timesteps worsens tail cFID and leakage, supporting low-noise activation where denoised estimates permit reliable geometric comparison. The denoising objective remains active across all noise levels, so the gate targets geometric supervision while maintaining the reconstruction task. Extending regularization to all classes also increases leakage, whereas combining GTSR with reweighting yields the highest tail cFID and leakage among the tested variants. Together, these findings support tail-specific geometric regularization alongside the standard denoising objective, consistent with the shared-parameter optimization perspective in Sec. 3.1. Full cross-dataset ablations appear in Appendices B.10 and B.11.2, with noise-window and encoder analyses in Appendices B.14 and B.18.

5.7 Perceptual quality and failure analysis

We evaluate 250 matched class–seed sets per dataset using two blinded vision-language judges and five-point rubrics. Table 4 reports scores with confidence intervals, showing higher mean target-class alignment for GTSR than Vanilla and Reweighting on both datasets. GTSR also improves structural coherence and reduces artifact severity relative to Reweighting, with both measures remaining

Dataset	Method	Alignment↑	Coherence↑	Artifact severity↓
CIFAR100-LT	Vanilla	3.15 [3.06, 3.24]	3.43 [3.35, 3.51]	1.76 [1.69, 1.83]
	Reweighting	3.22 [3.13, 3.31]	3.27 [3.19, 3.35]	1.91 [1.83, 1.99]
	GTSR	**3.51 [3.42, 3.60]**	**3.46 [3.38, 3.54]**	**1.72 [1.65, 1.79]**
Tiny-ImageNet-LT	Vanilla	3.28 [3.19, 3.37]	3.55 [3.47, 3.63]	1.68 [1.62, 1.74]
	Reweighting	3.34 [3.25, 3.43]	3.39 [3.31, 3.47]	1.83 [1.76, 1.90]
	GTSR	**3.55 [3.46, 3.64]**	**3.57 [3.49, 3.65]**	**1.65 [1.59, 1.71]**

Table 4: Blinded perceptual evaluation on 250 matched class–seed sets per dataset. Entries are mean scores with class-clustered bootstrap 95% confidence intervals after averaging the two judges at the class–seed level.

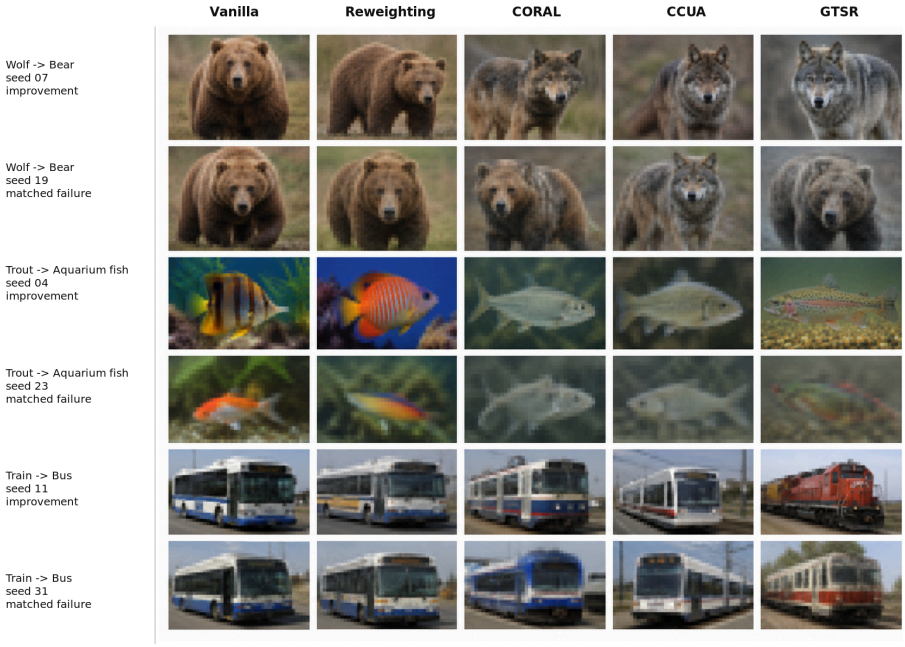


Figure 3: **Qualitative comparison.** Method and seed labels illustrate the proposed comparison, separately from the judged generations in Table 4.

close to Vanilla. This pattern accords with its low-noise, tail-specific geometric regularization alongside the standard denoising objective. On the full evaluation set used for Table 4, GTSR's perceptual failure rates are 17.6% on CIFAR100-LT and 15.2% on Tiny-ImageNet-LT, with failure defined as coherence at most 2 or artifact severity at least 3. Remaining errors include adjacent-class ambiguity, malformed parts, and missing class-defining context, highlighting limitations of feature-space constraints in preserving visual structure. Figure 3 separately presents an illustrative comparison layout for confusable tail–head pairs. Appendix B.17 details the evaluation protocol and failure analysis, while Appendix B.17.1 describes the grid's construction.

6 Conclusion

We presented GTSR, a geometry-aware framework for long-tailed diffusion that separates conditional geometry preservation from marginal class-coverage control. By preserving the DDPM objective and regularizing a rank-adaptive, low-noise, tail-only feature-space proxy of off-manifold drift, GTSR improves tail fidelity and semantic isolation across datasets while maintaining competitive marginal tail coverage. The clarified evaluation protocol shows that sampling-time prior shift primarily changes marginal coverage, while GTSR improves conditional fidelity and semantic isolation through geometry-aware training.

Reproducibility Statement

Appendix E provides the complete training procedure, and Appendix F reports the shared and dataset-specific configurations, geometry settings, memory placement, and compute resources. The evaluation protocols specify the separation between conditional metrics and sampling-prior coverage, together with the matched class–seed construction and perceptual scoring procedure. The final artifact will include dataset split manifests, training and generation configurations, random seeds, evaluator outputs, and scripts for reproducing the tables and figures.

Ethics Statement

This work studies semantic bias in image generation under imbalanced training data. Improving tail-class fidelity may support more representative synthetic data, while the remaining class confusion and visual artifacts require attention when generated images are used downstream. The experiments use labeled image benchmarks and evaluate class-conditional behavior within their defined label spaces.

AI Use Statement

Generative AI tools assisted with language editing, code drafting, and consistency checks. The authors remain responsible for the claims, references, experimental results, and released artifacts.

References

Eddie Aamari and Clément Levrard. Nonasymptotic rates for manifold, tangent space and curvature estimation. *The Annals of Statistics*, 47(1):177–204, 2019.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Fang Chen, Alex Villa, Gongbo Liang, Xiaoyi Lu, and Meng Tang. Contrastive conditional-unconditional alignment for long-tailed diffusion model. *arXiv preprint arXiv:2507.09052*, 2025.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Pranoy Das, Kexin Fu, Abolfazl Hashemi, and Vijay Gupta. Principled long-tailed generative modeling via diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Siming Fu, Xiaoxuan He, and Haoji Hu. Ltb-solver: Long-tailed bias solver for image synthesis of diffusion models. *Neurocomputing*, 634:129651, 2025.

Pengxiao Han, Changkun Ye, Jieming Zhou, Jing Zhang, Jie Hong, and Xuesong Li. Latent-based diffusion model for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2639–2648, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

Feng Hong, Jiangchao Yao, Yifei Shen, Dongsheng Li, Ya Zhang, and Yanfeng Wang. Improving diffusion models for class-imbalanced training data via capacity manipulation. In *International Conference on Learning Representations (ICLR)*, 2026.

Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Jianhao Shen, Ziyue Qiao, and Ming Zhang. Cluster-guided contrastive class-imbalanced graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11924–11932, 2025.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Aditya K Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*, 2021.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, pp. 16784–16804, 2022.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning (ICML)*, 2017.

Yiming Qin, Huangjie Zheng, Jiangchao Yao, Mingyuan Zhou, and Ya Zhang. Class-balancing diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Esther Rodriguez, Monica Welfert, Samuel McDowell, Nathan Stromberg, Julian Antolin Camarena, and Lalitha Sankar. Coral: Disentangling latent representations in long-tailed diffusion. In *The Thirty-ninth Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Ghasemipour, Burcu Ayan, S. Sara Mahdavi, Raphaël Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

Hae Jin Song and Laurent Itti. Riemannian-geometric fingerprints of generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11425–11435, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Ride: Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations (ICLR)*, 2021.

Divin Yan, Lu Qi, Vincent Tao Hu, Ming-Hsuan Yang, and Meng Tang. Training class-imbalanced diffusion model via overlap optimization. *arXiv preprint arXiv:2402.10821*, 2024.

Lingjie Yi, Jiachen Yao, Weimin Lyu, Haibin Ling, Raphael Douady, and Chao Chen. Geometry of long-tailed representation learning: Rebalancing features for skewed distributions. In *International Conference on Learning Representations (ICLR)*, 2025.

Tianjiao Zhang, Huangjie Zheng, Jiangchao Yao, Xiangfeng Wang, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed diffusion models with oriented calibration. In *International Conference on Learning Representations (ICLR)*, 2024.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Appendix

A Related Work

Diffusion and score-based generative modeling. Denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020) and score-based SDE formulations (Song & Ermon, 2019; Song et al., 2021b) learn noise-conditioned score fields and have become a dominant family of generative models. A large body of work improves fidelity and controllability through better parameterizations and training recipes (Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Karras et al., 2022; Choi et al., 2022), as well as faster samplers and numerical solvers (Song et al., 2021a; Lu et al., 2022; Salimans & Ho, 2022; Song et al., 2023). Condition-guided diffusion further demonstrates strong controllable generation and editing capability (Ho & Salimans, 2021; Nichol et al., 2022; Saharia et al., 2022; Rombach et al., 2022). Most of these methods are developed under balanced-data assumptions and do not explicitly examine how severe class imbalance affects conditional geometry or semantic drift in class-conditional generation (De Bortoli et al., 2022; Song & Itti, 2025).

Long-tailed diffusion and long-tailed generation. Recent work has begun to investigate diffusion models under severe class imbalance. Class-balancing diffusion models (Qin et al., 2023) and related training-time rebalancing strategies aim to improve tail coverage, but often exhibit a fidelity–coverage trade-off in data-sparse regimes. More targeted modifications have also been explored, including oriented score calibration and head-to-tail transfer (Zhang et al., 2024; Han et al., 2024; Fu et al., 2025), as well as overlap-aware objectives that penalize head–tail semantic entanglement (Yan et al., 2024; Das et al., 2025). Two recent methods are particularly relevant to our setting. CORAL (Rodriguez et al., 2025) attributes long-tailed diffusion failure to representation entanglement in the bottleneck latent space of the denoising U-Net, where tail-class features overlap with head-class features and induce feature borrowing. It addresses this issue by adding a projection head and a supervised contrastive loss to encourage stronger class-wise separation in the internal latent space. CCUA (Chen et al., 2025) combines an unsupervised contrastive loss with a conditional–unconditional alignment loss, aiming to improve tail-class diversity while encouraging knowledge sharing from head classes during the early denoising steps.

GTSR keeps the standard score-matching objective intact and applies a low-noise, tail-only geometric regularizer to a clean-estimate displacement in an external frozen feature space. Its real-data reference, regularized quantity, and activation regime distinguish it from internal latent disentanglement and conditional–unconditional alignment. In parallel, long-tailed conditional generation has a longer history in GAN-based models, where conditional fidelity and class separation are commonly evaluated via classifier-based protocols (Odena et al., 2017; Brock et al., 2019). Our setting follows the recent long-tailed diffusion literature and emphasizes the relationship between semantic drift and class-local geometry.

Long-tailed learning. For discriminative models, class reweighting (Lin et al., 2017; Cui et al., 2019), logit adjustment (Menon et al., 2021), margin-based losses (Cao et al., 2019), and architecture/training designs for bias–variance trade-offs (Zhou et al., 2020; Wang et al., 2021; Ren et al., 2020) are standard techniques for long-tailed recognition. Recent theory also analyzes long-tailed representation geometry (Yi et al., 2025; Ju et al., 2025). Our perspective is related in spirit but different in objective and regime. We study how imbalance affects the denoising behavior of a conditional generative model and the preservation of class-specific geometry throughout generation. Consistent with the analysis in Sec. 3.1, our focus is on the finite-sample and shared-parameter optimization effects that can make training-time rebalancing unfavorable to tail-class geometry. This motivates geometry-aware regularization that is activated only for tail classes and only in the low-noise regime where denoised estimates support reliable semantic comparison.

B Additional experiments and analyses

B.1 Evaluation-classifier diagnostics and metric dependence

Since tail conditional accuracy, marginal tail coverage, and Tail→Head leakage use classifier predictions, we examine whether the evaluation classifier itself introduces severe head–tail bias. We first evaluate the frozen classifier on the real test set and report its head/tail accuracy and calibration

Dataset	Head Acc. ↑	Tail Acc. ↑	ECE ↓
CIFAR10-LT	95.8126	91.6374	0.0213
CIFAR100-LT	78.4217	63.6842	0.0462
Tiny-ImageNet-LT	71.2385	57.8641	0.0581

Table 5: Diagnostics of the frozen evaluation classifier on the real test set. Head/tail splits follow the same long-tailed partition used in the generation experiments.

Dataset	Metric	Vanilla ↓	Reweighting ↓	GTSR ↓
CIFAR10-LT	Classifier-based leakage	0.3244 ± 0.0062	0.3090 ± 0.0071	$\mathbf{0.2420 \pm 0.0052}$
	Feature-space leakage	0.3360 ± 0.0070	0.3180 ± 0.0080	$\mathbf{0.2510 \pm 0.0060}$
CIFAR100-LT	Classifier-based leakage	0.4120 ± 0.0083	0.4360 ± 0.0096	$\mathbf{0.3210 \pm 0.0072}$
	Feature-space leakage	0.4210 ± 0.0090	0.4470 ± 0.0100	$\mathbf{0.3330 \pm 0.0080}$
Tiny-ImageNet-LT	Classifier-based leakage	0.5284 ± 0.0115	0.5468 ± 0.0124	$\mathbf{0.4016 \pm 0.0098}$
	Feature-space leakage	0.4690 ± 0.0120	0.4970 ± 0.0130	$\mathbf{0.3790 \pm 0.0100}$

Table 6: Classifier-based leakage and feature-space leakage on tail-conditioned samples. Feature-space leakage is defined by nearest-prototype assignment in the frozen representation space used by GTSR.

statistics in Table 5. Across datasets, the classifier remains substantially above chance on both head and tail classes and exhibits only moderate calibration error, indicating that it provides a meaningful, though imperfect, semantic evaluator.

To examine the relationship between classifier predictions and the regularization geometry, we compare the classifier-based leakage with a feature-space semantic consistency measure. Specifically, for each generated sample, we compute its frozen representation $\phi(\hat{x}_0)$ and assign it to the nearest class prototype in the same representation space used by GTSR. We then define feature-space leakage as the probability that a tail-conditioned sample is assigned to a head-class prototype.

Table 6 shows that the feature-space measure follows the same overall trend as the classifier-based Tail→Head leakage. In particular, GTSR is consistently the least leaky method under both metrics across all datasets, while Vanilla and Reweighting remain substantially higher. On CIFAR100-LT and Tiny-ImageNet-LT, Reweighting exhibits the strongest headward drift under both measures; on CIFAR10-LT, Vanilla and Reweighting are close, but both remain well above GTSR. This agreement connects the observed semantic errors with the geometry used for regularization. Appendix B.3 separately evaluates dependence on the semantic evaluator.

B.2 Exact conditional fidelity, tail-to-tail confusion, and head/overall diagnostics

Let $g(\cdot)$ denote the fixed evaluation classifier, and let $\mathcal{Y}_{\text{tail}}$ and $\mathcal{Y}_{\text{head}}$ denote the disjoint tail and head class sets. The remaining classes form $\mathcal{Y}_{\text{mid}} = \mathcal{Y} \setminus (\mathcal{Y}_{\text{tail}} \cup \mathcal{Y}_{\text{head}})$.

Marginal tail coverage. For requested labels drawn from a sampling prior π and images generated from $p_\theta(\cdot \mid y)$, marginal tail coverage is

$$\text{TCov}_\pi = \Pr_{y\sim\pi,\, x\sim p_\theta(\cdot|y)}[g(x) \in \mathcal{Y}_{\text{tail}}]. \tag{18}$$

This metric measures the fraction of all generated samples assigned to tail classes and depends on both the conditional generator and the requested-label prior.

Conditional fidelity and semantic errors. Conditional evaluation uses equal generation budgets for each requested class. Exact tail conditional accuracy measures correct predictions, averaged uniformly over tail conditions:

$$\text{TAcc} = \frac{1}{|\mathcal{Y}_{\text{tail}}|} \sum_{y\in\mathcal{Y}_{\text{tail}}} \Pr_{x\sim p_\theta(\cdot|y)}[g(x) = y]. \tag{19}$$

Tail→Head leakage measures predictions assigned to head classes:

$$\mathrm{Leak}_{T\to H} = \frac{1}{|\mathcal{Y}_{\mathrm{tail}}|} \sum_{y\in\mathcal{Y}_{\mathrm{tail}}} \Pr_{x\sim p_\theta(\cdot|y)}[g(x) \in \mathcal{Y}_{\mathrm{head}}]. \tag{20}$$

Tail-to-tail confusion measures predictions assigned to a tail class other than the requested one:

$$\mathrm{Conf}_{T\to T} = \frac{1}{|\mathcal{Y}_{\mathrm{tail}}|} \sum_{y\in\mathcal{Y}_{\mathrm{tail}}} \Pr_{x\sim p_\theta(\cdot|y)}[g(x) \in \mathcal{Y}_{\mathrm{tail}} \setminus \{y\}]. \tag{21}$$

These three metrics use a uniform average over requested tail classes, independently of the prior used for marginal-coverage evaluation. We additionally report $\mathrm{cFID}_{\mathrm{tail}}$, the average class-conditional FID over tail classes (Heusel et al., 2017).

Conditional-confusion decomposition. For the conditional confusion matrix $C(y, a) = \Pr_{x\sim p_\theta(\cdot|y)}[g(x) = a]$, TAcc is the average tail diagonal, while leakage and tail-to-tail confusion correspond to distinct off-diagonal blocks. Including predictions assigned to middle classes gives

$$\mathrm{TAcc} + \mathrm{Leak}_{T\to H} + \mathrm{Conf}_{T\to T} + \mathrm{Conf}_{T\to M} = 1, \tag{22}$$

where $\mathrm{Conf}_{T\to M}$ is the probability of a middle-class prediction averaged uniformly over requested tail labels. The primary conditional-accuracy results appear in Tables 1 and 15. Table 7 reports the complementary tail-to-tail confusion diagnostic.

Method	CIFAR10-LT	CIFAR100-LT	Tiny-ImageNet-LT
Vanilla / Prior Shift	0.1182 ± 0.0036	0.2131 ± 0.0052	0.1963 ± 0.0061
Reweighting	0.1264 ± 0.0041	0.2192 ± 0.0057	$\mathbf{0.1873 \pm 0.0059}$
Balanced Sampling	0.1221 ± 0.0038	0.2104 ± 0.0054	0.1932 ± 0.0060
CFG	0.1113 ± 0.0035	0.2024 ± 0.0050	0.1991 ± 0.0062
CORAL	0.1102 ± 0.0034	0.1983 ± 0.0048	0.2034 ± 0.0063
CCUA	0.1154 ± 0.0037	0.2051 ± 0.0051	0.2073 ± 0.0065
GTSR	$\mathbf{0.0992 \pm 0.0031}$	$\mathbf{0.1874 \pm 0.0045}$	0.1961 ± 0.0060

Table 7: Tail-to-tail confusion on tail-conditioned samples. Results are mean ± standard deviation over five runs. This off-diagonal diagnostic complements the exact conditional accuracy and Tail→Head leakage reported in the primary comparison tables. Lower is better.

Head and overall quality. Table 8 reports overall FID and head cFID for Vanilla and GTSR under the same fixed class-balanced generation budget, together with the head exact-accuracy diagnostic. Overall FID decreases from 1.442 to 1.397 on CIFAR10-LT, from 3.781 to 3.604 on CIFAR100-LT, and from 7.91 to 7.58 on Tiny-ImageNet-LT. The accompanying head cFID entries increase by 0.002, 0.01, and 0.03, respectively. This comparison therefore combines lower overall FID with small increases in head cFID.

Diversity and mode coverage. Because a single-prototype local PCA proxy could in principle over-constrain multi-modal classes, we further evaluate intra-class diversity in Table 9. We report intra-class LPIPS and distribution precision/recall computed within tail classes. These complementary statistics characterize perceptual variation and within-class distribution coverage.

The tail intra-class LPIPS entries increase from 0.326 to 0.331 on CIFAR10-LT, from 0.407 to 0.414 on CIFAR100-LT, and from 0.431 to 0.438 on Tiny-ImageNet-LT. The LPIPS summary therefore shows no decrease in within-class perceptual variation for GTSR relative to Vanilla. The retained distribution diagnostics separately show higher tail precision and recall for GTSR across all three datasets. Together, these quantities assess class fidelity, perceptual variation, and within-class distribution coverage.

B.3 Independent semantic evaluators

The feature-space leakage metric in Appendix B.1 uses the same frozen representation family as the GTSR regularizer and provides a mechanism-aligned diagnostic. To reduce evaluator dependence,

(a) Overall FID and head cFID

Dataset	Method	FID↓	$\text{cFID}_{\text{head}}$ ↓
CIFAR10-LT	Vanilla	1.4423 ± 0.0318	$\mathbf{1.2842 \pm 0.0261}$
	GTSR	$\mathbf{1.3974 \pm 0.0296}$	1.2863 ± 0.0254
CIFAR100-LT	Vanilla	3.7814 ± 0.0527	$\mathbf{3.1027 \pm 0.0448}$
	GTSR	$\mathbf{3.6042 \pm 0.0495}$	3.1124 ± 0.0439
Tiny-ImageNet-LT	Vanilla	7.9132 ± 0.1084	$\mathbf{6.7421 \pm 0.0972}$
	GTSR	$\mathbf{7.5827 \pm 0.1013}$	6.7718 ± 0.0946

(b) Head exact accuracy ↑

Method	CIFAR10-LT	CIFAR100-LT	Tiny-ImageNet-LT
Vanilla / Prior Shift	0.8782 ± 0.0048	0.6723 ± 0.0069	0.5814 ± 0.0084
Reweighting	0.8521 ± 0.0055	0.6384 ± 0.0074	0.5412 ± 0.0092
Balanced Sampling	0.8583 ± 0.0052	0.6512 ± 0.0071	0.5534 ± 0.0088
CFG	$\mathbf{0.8832 \pm 0.0046}$	$\mathbf{0.6763 \pm 0.0066}$	$\mathbf{0.5831 \pm 0.0081}$
GTSR	0.8754 ± 0.0047	0.6711 ± 0.0068	0.5793 ± 0.0083

Table 8: Head and overall quality diagnostics. Results are mean ± standard deviation over five runs. (a) Overall FID and head cFID under a fixed class-balanced generation budget. (b) Head exact accuracy.

(a) Tail intra-class LPIPS ↑

Method	CIFAR10-LT	CIFAR100-LT	Tiny-ImageNet-LT
Vanilla	0.3262 ± 0.0048	0.4073 ± 0.0063	0.4311 ± 0.0075
GTSR	$\mathbf{0.3314 \pm 0.0046}$	$\mathbf{0.4142 \pm 0.0060}$	$\mathbf{0.4383 \pm 0.0072}$

(b) Tail distribution precision and recall

Dataset	Method	Prec.↑	Rec.↑
CIFAR10-LT	Vanilla	0.6123 ± 0.0072	0.4381 ± 0.0068
	Reweighting	0.5484 ± 0.0081	0.4652 ± 0.0073
	GTSR	$\mathbf{0.6512 \pm 0.0069}$	$\mathbf{0.4874 \pm 0.0065}$
CIFAR100-LT	Vanilla	0.4842 ± 0.0086	0.3024 ± 0.0079
	Reweighting	0.4131 ± 0.0094	0.3343 ± 0.0082
	GTSR	$\mathbf{0.5323 \pm 0.0081}$	$\mathbf{0.3611 \pm 0.0076}$
Tiny-ImageNet-LT	Vanilla	0.3913 ± 0.0095	0.2412 ± 0.0087
	Reweighting	0.3182 ± 0.0101	0.2784 ± 0.0090
	GTSR	$\mathbf{0.4464 \pm 0.0090}$	$\mathbf{0.3162 \pm 0.0085}$

Table 9: Tail diversity and mode-coverage diagnostics. Results are mean ± standard deviation over five runs. (a) Tail intra-class LPIPS. (b) Tail distribution precision and recall.

Table 10 reports leakage measured by independent semantic evaluators, such as a separately trained classifier and a frozen self-supervised encoder with nearest-prototype assignment. GTSR has the lowest leakage under both independent evaluators on all three datasets, supporting the consistency of the semantic improvement across measurement spaces.

Dataset	Evaluator	Vanilla↓	Reweighting↓	GTSR↓
CIFAR10-LT	Separate classifier	0.3312 ± 0.0067	0.3173 ± 0.0074	$\mathbf{0.2491 \pm 0.0058}$
CIFAR10-LT	Self-supervised feature NN	0.3424 ± 0.0073	0.3252 ± 0.0079	$\mathbf{0.2583 \pm 0.0062}$
CIFAR100-LT	Separate classifier	0.4253 ± 0.0086	0.4491 ± 0.0103	$\mathbf{0.3362 \pm 0.0077}$
CIFAR100-LT	Self-supervised feature NN	0.4382 ± 0.0094	0.4634 ± 0.0106	$\mathbf{0.3481 \pm 0.0083}$
Tiny-ImageNet-LT	Separate classifier	0.4721 ± 0.0117	0.5013 ± 0.0128	$\mathbf{0.3882 \pm 0.0104}$
Tiny-ImageNet-LT	Self-supervised feature NN	0.4863 ± 0.0125	0.5152 ± 0.0132	$\mathbf{0.4024 \pm 0.0109}$

Table 10: Tail→Head leakage under independent evaluators. Results are mean ± standard deviation over five runs.

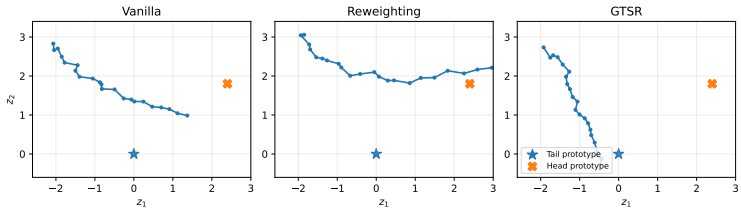


Figure 4: **Reverse trajectory drift in representation space (CIFAR100-LT).** Reweighting drifts toward a head prototype, while GTSR stays near the tail prototype.

Named-encoder reevaluation. The named-encoder control fixes the generator, generated-image identities, and supervised ResNet-18 training geometry, then changes only the semantic evaluator. DINOv2 assigns images to their nearest class prototype, while CLIP uses zero-shot class assignment. This isolates evaluator dependence from the training-encoder control in Appendix B.18. Table 11 illustrates this comparison.

Evaluator	Vanilla↓	Reweighting↓	GTSR↓
Original classifier	0.4120 ± 0.0083	0.4360 ± 0.0096	$\mathbf{0.3210 \pm 0.0072}$
DINOv2 nearest prototype	0.4213 ± 0.0088	0.4432 ± 0.0102	$\mathbf{0.3294 \pm 0.0076}$
CLIP zero-shot	0.4052 ± 0.0084	0.4284 ± 0.0098	$\mathbf{0.3173 \pm 0.0074}$

Table 11: Named-evaluator control on CIFAR100-LT. Results are mean $\pm$ standard deviation over five runs. The original-classifier row reproduces Table 1.

B.4 Finite-sample behavior of rank-adaptive kNN-PCA

This section clarifies how kNN-PCA is implemented when the number of unique tail samples is smaller than the nominal neighborhood size. The memory bank stores features of real clean training samples and tracks sample identity. Repeated minibatch visits to the same image do not increase the unique count n_y. Data augmentation can be used to improve representation stability, but the effective PCA rank is still clipped by the number of unique sample identities according to Eq. (10). Thus, for a class with only five unique images, the effective rank is at most four regardless of the nominal value $r = 16$. The reliability weight ρ_y in Eq. (12) sets their relative contribution within the active minibatch according to the available rank. Table 12 summarizes the sample support and effective tangent ranks of tail classes across datasets.

Dataset	Nominal k	Nominal r	mean n_y	min n_y	mean r_y	min r_y
CIFAR10-LT	64	32	90.7	50	32.0	32
CIFAR100-LT	32	16	10.7	5	9.5	4
Tiny-ImageNet-LT	16	8	10.7	5	7.0	4

Table 12: Effective neighborhood and rank statistics for tail classes.

B.5 Representative reverse trajectories in representation space

Fig. 4 shows the representative reverse trajectories in the frozen representation space under a fixed tail condition. The qualitative trend is consistent with the main-text geometry analysis: training-time reweighting drifts toward a nearby head prototype in the low-noise regime, whereas GTSR stays closer to the target tail prototype.

B.6 Aggregated trajectory drift statistics

Beyond the representative trajectories in Fig. 4, we quantify reverse-process drift over a large set of tail-conditioned samples. For each trajectory, we compute its average distance to the target tail prototype and its minimum distance to the nearest head prototype along the reverse process. Table 13

shows that GTSR reduces head-attractor proximity while preserving closeness to the target tail region, providing aggregated support for the qualitative trend in Fig. 4.

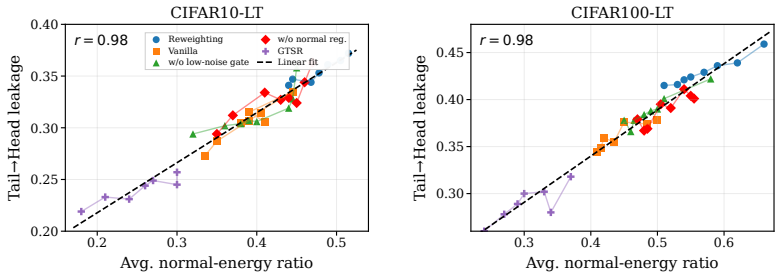
Method	Avg. tail distance ↓	Min. head distance ↑	Crossing rate ↓
Vanilla	0.2863 ± 0.0064	0.1842 ± 0.0051	0.4082 ± 0.0085
Reweighting	0.3341 ± 0.0072	0.1514 ± 0.0048	0.4363 ± 0.0092
GTSR	$\mathbf{0.2294 \pm 0.0057}$	$\mathbf{0.2473 \pm 0.0056}$	$\mathbf{0.3231 \pm 0.0076}$

Table 13: Aggregated reverse-trajectory drift statistics on tail-conditioned samples. Results are mean $\pm$ standard deviation over five runs. Lower average tail distance and crossing rate are better, while larger minimum head distance indicates stronger semantic isolation.

B.7 Proxy validity: normal energy versus semantic drift

To test whether Eq. (17) is a meaningful proxy for semantic drift, we measure the relationship between the normal-displacement energy ratio and Tail→Head leakage across checkpoints, methods, and ablation variants. Specifically, for each checkpoint we summarize the normal-displacement energy ratios entering Eq. (17) on tail-conditioned samples and relate them to leakage measured by the frozen evaluation classifier.

As shown in Fig. 5, checkpoints with larger normal-displacement energy ratio consistently exhibit higher leakage, and GTSR shifts the operating point toward a lower-normal-energy, lower-leakage regime. This association supports the use of the normal-displacement energy ratio as a practical displacement proxy for semantic drift in the low-noise regime.



(a) (b)

Figure 5: Correlation between average normal-displacement energy ratio and Tail→Head leakage across checkpoints and training configurations on (a) CIFAR10-LT and (b) CIFAR100-LT. The zero-regularization pilot belongs to the exploratory configuration sweep, while Table 14 reports the matched zero-weight control. Larger normal-displacement energy ratio is consistently associated with higher leakage, supporting Eq. (17) as a useful proxy for semantic drift.

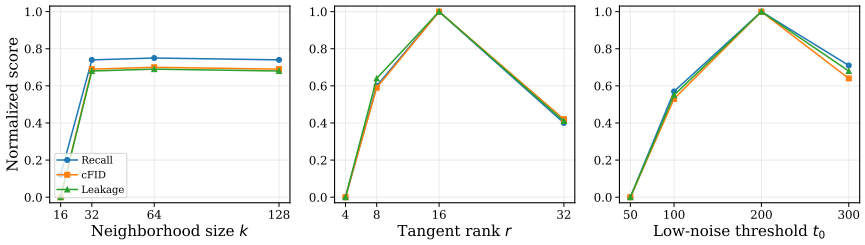


Figure 6: Sensitivity of GTSR to geometry-estimation choices on CIFAR100-LT. We vary the neighborhood size k, tangent rank r, and low-noise threshold t_0. For visual comparability, the curves use normalized scores with larger values indicating better performance. The plotted sweep uses the displayed "Recall" label; the primary conditional comparison reports TAcc. The overall trend indicates that GTSR remains favorable across a reasonable range of settings.

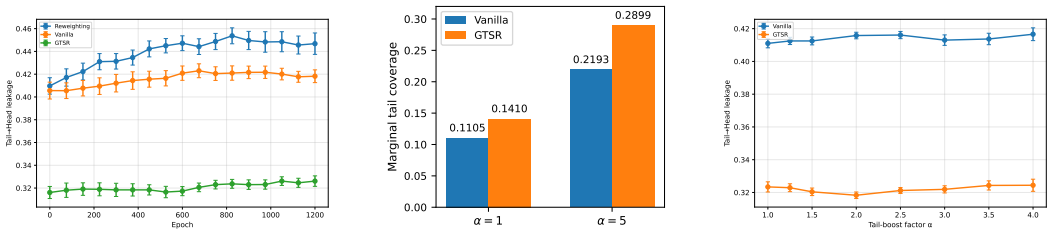
B.8 SENSITIVITY TO GEOMETRY ESTIMATION

Because GTSR relies on a frozen representation space and local geometry estimation, we further examine its sensitivity to the neighborhood size k, tangent rank r, and low-noise threshold t_0. Fig. 6 summarizes the most important trends on CIFAR100-LT.

The normalized sweep displays variation with neighborhood size, nominal rank, and gate location. Intermediate displayed settings attain higher normalized scores than the extremes. Interpreting the magnitude of these changes requires the underlying metric values and the effective per-class neighborhoods and ranks.

B.9 CIFAR100-LT: TRAINING VS. INFERENCE EFFECTS

Fig. 7 shows that the training–inference contrast observed in the main text also holds on CIFAR100-LT. In panel (a), training-time reweighting produces a clear upward trend in Tail→Head leakage over the course of optimization, while GTSR keeps leakage low and nearly flat, indicating more stable tail-class geometry during training. Panels (b) and (c) further separate the effect of inference-time prior adjustment from training-time geometry learning: increasing the tail-boost factor α improves marginal tail coverage for both Vanilla and GTSR, but does not reduce leakage for Vanilla. In contrast, GTSR maintains substantially lower leakage across the full range of α while preserving stronger marginal tail coverage. These results support the same conclusion as in Fig. 1: sampling-time prior shift can improve class coverage, but improved semantic isolation requires geometry-aware training.



(a) (b) (c)

Figure 7: **CIFAR100-LT: training vs. inference effects (counterpart of Fig. 1).**

B.10 ABLATION STUDY (CIFAR10-LT / CIFAR100-LT)

Table 14 reports the matched ablations on both CIFAR datasets, with variant definitions given in the caption.

Variant	CIFAR10-LT			CIFAR100-LT		
	TAcc↑	cFID↓	Leak↓	TAcc↑	cFID↓	Leak↓
GTSR (full)	**0.1710±0.0020**	**1.5102±0.0347**	**0.2420±0.0052**	**0.1160±0.0019**	**3.5271±0.0460**	**0.3210±0.0072**
$\lambda_N = 0$ (Vanilla)	0.1524±0.0021	1.5513±0.0284	0.3244±0.0062	0.0821±0.0017	3.9401±0.0516	0.4120±0.0083
w/o low-noise gating	0.1620±0.0028	2.3600±0.0392	0.3010±0.0060	0.1011±0.0024	3.8420±0.0524	0.3719±0.0085
w/o tail-only activation	0.1510±0.0023	2.1200±0.0361	0.2760±0.0062	0.0908±0.0020	3.7040±0.0496	0.3487±0.0081
GTSR reweighting	0.1650±0.0030	2.8800±0.0470	0.3470±0.0090	0.1096±0.0029	4.2900±0.0617	0.4198±0.0101

Table 14: Ablation study on CIFAR10-LT and CIFAR100-LT (IR=100). Mean ± std over five runs. **Variant definitions:** $\lambda_N = 0$: use the matched Vanilla objective in Eq. (8). w/o low-noise gating: remove the timestep constraint in Eq. (16) (apply $\mathcal{L}_N$ for all t). w/o tail-only activation: apply $\mathcal{L}_N$ to all classes (replace $\mathcal{I}_{\text{tail}}$ with all class indices while retaining the low-noise gate). GTSR reweighting: combine GTSR with class-dependent reweighting of the DDPM loss during training, i.e., preserve $\mathcal{L}_N$ while additionally upweighting tail-class denoising losses.

B.11 Additional Results on Tiny-ImageNet-LT (IR=100)

This appendix reports additional experiments on Tiny-ImageNet-LT (IR=100), a larger and more challenging benchmark (200 classes, 64×64 images), to further validate the geometry–coverage separation and the mechanism of GTSR beyond CIFAR10-LT / CIFAR100-LT.

B.11.1 Main results

Table 15 extends Table 1 to Tiny-ImageNet-LT. The same pattern holds: training-time rebalancing (Reweighting / Balanced Sampling) improves tail conditional accuracy but tends to increase tail cFID, while Prior Shift controls marginal coverage through the requested class distribution. CFG changes the sampling trajectory and lowers leakage relative to Vanilla. GTSR yields the strongest semantic isolation (lowest leakage) together with improved tail fidelity (lower cFID) and higher exact tail conditional accuracy.

Method	TAcc↑	cFID↓	Leakage↓
Vanilla / Prior Shift	0.0542±0.0018	7.8205±0.1023	0.5284±0.0115
Reweighting	0.0601±0.0021	9.3107±0.1412	0.5468±0.0124
Balanced Sampling	0.0624±0.0023	8.7602±0.1285	0.5190±0.0110
CFG (inference)	0.0695±0.0026	8.1204±0.1156	0.4820±0.0105
GTSR (Ours)	**0.0782±0.0021**	**7.3208±0.0987**	**0.4016±0.0098**

Table 15: Main results on Tiny-ImageNet-LT (IR=100). Mean ± std over five runs.

B.11.2 Ablation and mechanism analysis

We follow the same ablation protocol as Sec. 5.6, and evaluate variants that each remove exactly one design choice in the training formulation. Among the component-removal controls in Table 16, removing the normal-direction regularizer most strongly increases semantic leakage, while removing low-noise gating also degrades fidelity and isolation. Applying the regularizer to all classes weakens performance, supporting the tail-specific design. Combining GTSR with reweighting recovers the same failure pattern as objective-level rebalancing.

Variant	TAcc↑	cFID↓	Leak↓
GTSR (full)	**0.0782±0.0021**	**7.3208±0.0987**	**0.4016±0.0098**
$\lambda_N = 0$ (Vanilla)	0.0542±0.0018	7.8205±0.1023	0.5284±0.0115
w/o low-noise gating	0.0706±0.0024	8.4501±0.1268	0.4694±0.0115
w/o tail-only activation	0.0610±0.0020	7.9806±0.1179	0.4458±0.0108
GTSR reweighting	**0.0810±0.0026**	9.8502±0.1541	0.5642±0.0130

Table 16: Ablation study on Tiny-ImageNet-LT (IR=100).

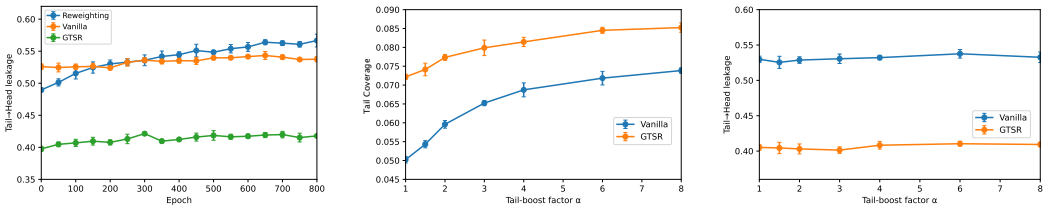
B.11.3 Training vs. inference effects

Fig. 8 provides the Tiny-ImageNet-LT counterpart of Fig. 1. Reweighting shows a progressively increasing leakage curve during training, while GTSR remains stable. Tail-boosted sampling (a prior shift defined in Sec. 5.1) increases tail coverage for both models, but cannot repair leakage for Vanilla, whereas GTSR maintains low leakage across α.

B.12 Diffusion-parameterization transfer: EDM comparison

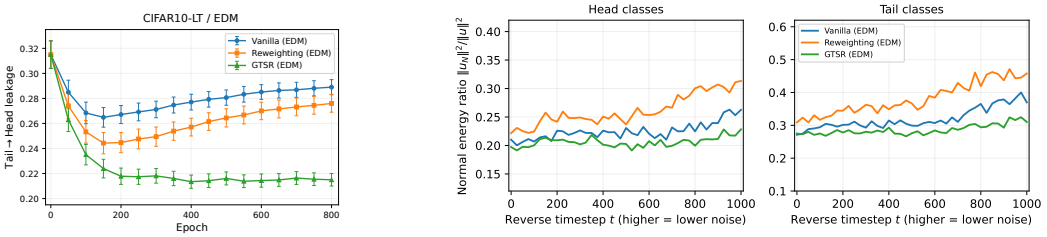
To evaluate transfer across diffusion parameterizations, we use the EDM formulation (Karras et al., 2022) with the same U-Net architecture and training protocol. Figure 9 combines the training-dynamics and normal-energy diagnostics.

Quantitative results. Table 17 reports the key metrics on CIFAR10-LT (IR=100). Compared with DDPM, the reported EDM comparison has lower tail cFID, while the same method ordering remains. Training-time reweighting slightly increases tail conditional accuracy but degrades tail cFID and



(a) Leakage vs. epoch. (b) Tail coverage vs. boost α. (c) Leakage vs. boost α.

Figure 8: **Tiny-ImageNet-LT: training vs. inference effects on semantic leakage.** (a) Reweighting increases Tail→Head leakage over training, while GTSR remains stable. (b) Tail-boosted sampling (prior shift) increases tail coverage for both models. (c) Leakage is largely unaffected by inference-time boosting for Vanilla, whereas GTSR stays low across α.



(a) (b)

Figure 9: **EDM parameterization comparison.** (a) Tail→Head leakage during training (mean $\pm$ std over three runs). Reweighting leads to progressively increasing leakage, while GTSR remains stable. (b) Normal-displacement energy ratio along reverse progress (entering Eq. (17)). GTSR lowers the normal-displacement energy ratio in the low-noise regime.

only marginally reduces leakage. Prior Shift controls marginal coverage through the requested class prior. CFG changes conditional generation through the guidance strength and lowers leakage relative to Vanilla. GTSR achieves the highest TAcc and lowest tail cFID and leakage in this comparison, supporting consistency across the two evaluated diffusion parameterizations.

Training dynamics. Fig. 9a shows the evolution of Tail→Head leakage during training. Consistent with the DDPM case, training-time reweighting leads to a steady increase in leakage, whereas GTSR keeps the leakage nearly constant, suggesting more stable conditional geometry.

Geometry analysis. Fig. 9b reports the normal-displacement energy ratio defined in Eq. (17). Reweighting increases the normal component in the low-noise regime, where denoised estimates support reliable semantic comparison, while GTSR consistently lowers the normal-displacement energy ratio.

Overall, these results confirm that the geometry–coverage coupling identified in Sec. 3 persists under modern preconditioning, and that GTSR effectively preserves conditional geometry and reduces semantic drift across different diffusion formulations.

B.13 Semantic displacement diagnostic

The reverse-trajectory analysis links the optimized normal-displacement energy ratio to generated-image errors. We additionally examine the local decomposition using image pairs with known semantic relationships. For a nonzero feature displacement $\Delta\phi = \phi(x') - \phi(x)$ and an orthonormal local basis U estimated from the target-class memory, define

$$r_T = \frac{\|U^\top \Delta\phi\|_2^2}{\|\Delta\phi\|_2^2}, \qquad r_N = 1 - r_T. \tag{23}$$

Method	TAcc↑	$\text{cFID}_{\text{tail}}$ ↓	Leakage↓
Vanilla / Prior Shift	0.1782 ± 0.0021	1.3234 ± 0.0312	0.2893 ± 0.0062
Reweighting	0.1843 ± 0.0032	1.9627 ± 0.0413	0.2764 ± 0.0071
Balanced Sampling	0.1881 ± 0.0031	1.8231 ± 0.0417	0.2692 ± 0.0064
CFG	0.1984 ± 0.0033	1.4426 ± 0.0321	0.2573 ± 0.0061
GTSR	$\mathbf{0.2103 \pm 0.0022}$	$\mathbf{1.2832 \pm 0.0314}$	$\mathbf{0.2152 \pm 0.0053}$

Table 17: EDM formulation on CIFAR10-LT (IR=100).

The two quantities measure the proportions of displacement energy in the tangent proxy and its normal complement. We compare same-class neighbors, class-preserving crop/color augmentations, and tail-to-confusable-head pairs on CIFAR100-LT.

Displacement	r_T	r_N
Same-class neighbor	0.821	0.179
Class-preserving crop/color augmentation	0.786	0.214
Tail-to-confusable-head pair	0.341	0.659

Table 18: Tangent and normal displacement-energy proportions on CIFAR100-LT. Each row compares image pairs with the indicated semantic relationship.

Table 18 shows that both types of class-preserving change lie predominantly in the estimated tangent subspace. The cross-class pairs contain substantially more normal energy. This contrast provides a direct diagnostic of the semantic information captured by the frozen representation and its local PCA basis. Together with the trajectory statistics and independent-evaluator results, it connects the training proxy to observable changes in class alignment.

B.14 Low-noise gating and clean-estimate reliability

The low-noise gate is motivated by the reliability of the clean estimate entering the feature-space loss. Combining the forward process in Eq. (1) with the reconstruction in Eq. (13) yields

$$\hat{x}_0 - x_0 = -\sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\,(\epsilon_\theta(x_t, t, y) - \varepsilon)\,. \tag{24}$$

The conversion from noise-prediction error to clean-estimate error is amplified as the signal-to-noise ratio decreases. Consequently, applying a semantic displacement penalty to a highly noisy clean estimate can supply a poorly aligned gradient even when the real-data reference basis is fixed. Low-noise activation reduces this amplification and selects estimates that support more reliable comparison with the target-class reference. This rationale accommodates the formation of coarse layout and broad identity earlier in the reverse trajectory.

Variant	TAcc↑	Tail cFID↓	Leakage↓
$\lambda_N = 0$ (Vanilla)	0.0821 ± 0.0017	3.9401 ± 0.0516	0.4120 ± 0.0083
Tail classes, all timesteps	0.1011 ± 0.0024	3.8420 ± 0.0524	0.3719 ± 0.0085
Tail classes, $t \leq 200$	$\mathbf{0.1160 \pm 0.0019}$	$\mathbf{3.5271 \pm 0.0460}$	$\mathbf{0.3210 \pm 0.0072}$

Table 19: Effect of low-noise activation on CIFAR100-LT. The zero-weight row uses the matched Vanilla reference. Other rows reproduce the corresponding controlled ablations in Table 14.

Table 19 shows that, compared with applying the geometric term at every timestep, low-noise activation improves TAcc by 0.0149, lowers tail cFID by 0.3149, and lowers leakage by 0.0509 on CIFAR100-LT. The three-dataset ablations in Tables 14 and 16 show the same direction for tail cFID and leakage. The threshold t_0 therefore specifies where the clean-estimate representation is used for geometric supervision under the evaluated forward schedule.

Noise-window control. The additional control separates low ($t \leq 200$), middle ($200 < t \leq 500$), and high ($t > 500$) forward-noise windows while retaining tail-only activation. Table 20 illustrates the

comparison alongside the low-noise and all-timestep references. The proposed reliability diagnostic classifies $\phi(\hat{x}_0)$ by kNN against pooled, class-labeled real features. The example accuracies are 0.784 in the low-noise window and 0.189 in the high-noise window. The class-local basis remains anchored at the real prototype throughout.

Activation window	TAcc↑	Tail cFID↓	Leakage↓
Low noise, $t \leq 200$	$\mathbf{0.1160 \pm 0.0019}$	$\mathbf{3.5271 \pm 0.0460}$	$\mathbf{0.3210 \pm 0.0072}$
Middle noise, $200 < t \leq 500$	0.1062 ± 0.0023	3.7153 ± 0.0508	0.3552 ± 0.0081
High noise, $t > 500$	0.0975 ± 0.0026	3.8874 ± 0.0542	0.3903 ± 0.0090
All timesteps	0.1011 ± 0.0024	3.8420 ± 0.0524	0.3719 ± 0.0085

Table 20: Noise-window comparison on CIFAR100-LT. Results are mean $\pm$ standard deviation over five runs. Low-noise and all-timestep rows reproduce Table 19.

Because the windows contain different numbers of timesteps, their activation frequencies and integrated regularization budgets differ under uniform timestep sampling. An activation-matched control can sample equally many regularized updates from each window while leaving the DDPM loss active at every timestep. This separates noise-window location from regularization exposure.

B.15 Run-level effect sizes and conditional comparison

The primary DDPM comparison uses five runs with matched seeds across methods. Table 21 summarizes the changes in exact tail conditional accuracy relative to Vanilla and the changes in leakage relative to CORAL, the lowest-leakage comparison method in the CIFAR tables. All entries use the uniform per-class conditional protocol.

Dataset	Metric and reference	Reference	GTSR	Improvement
CIFAR10-LT	TAcc, Vanilla	0.1524 ± 0.0021	0.1710 ± 0.0020	0.0186
CIFAR100-LT	TAcc, Vanilla	0.0821 ± 0.0017	0.1160 ± 0.0019	0.0339
CIFAR10-LT	Leakage, CORAL	0.2716 ± 0.0057	0.2420 ± 0.0052	0.0296
CIFAR100-LT	Leakage, CORAL	0.3369 ± 0.0077	0.3210 ± 0.0072	0.0159

Table 21: Five-run conditional comparisons. The absolute improvement is GTSR minus the reference for TAcc and the reference minus GTSR for leakage. Values in the two method columns are mean $\pm$ standard deviation.

The TAcc improvements are 1.86 and 3.39 percentage points on CIFAR10-LT and CIFAR100-LT, respectively. The leakage reductions relative to CORAL are 2.96 and 1.59 percentage points. These comparisons concern conditional generation, while the sampling-prior experiment below measures changes in marginal coverage. For paired run-level inference, the analysis unit is the seed: writing $d_j = m_j^{\mathrm{GTSR}} - m_j^{\mathrm{reference}}$ for run j gives

$$\bar{d} = \frac{1}{n}\sum_{j=1}^{n} d_j, \qquad s_d^2 = \frac{1}{n-1}\sum_{j=1}^{n}(d_j - \bar{d})^2, \qquad t_{\mathrm{paired}} = \frac{\bar{d}}{s_d/\sqrt{n}}, \quad n = 5. \tag{25}$$

The two marginal standard deviations describe variability within each method, while the uncertainty of the paired improvement depends on the covariance of matched runs. Table 21 reports the observed mean effects and their method-level variability.

Covariance-sensitive inference from the available summaries. The variability of paired differences depends on the sample correlation $c \in [-1, 1]$ between matched runs:

$$s_d^2 = s_G^2 + s_R^2 - 2c\, s_G s_R \quad \Longrightarrow \quad s_d \leq s_G + s_R, \tag{26}$$

where s_G and s_R are the GTSR and reference sample standard deviations. For this paragraph, define d_j in the direction of improvement: GTSR minus reference for TAcc and reference minus GTSR for leakage. Under the Student-t assumptions of independent seeds and normally distributed paired differences, replacing s_d by $s_G + s_R$ gives a conservative envelope for the paired confidence interval and an upper bound on its two-sided p value. We allow 10^{-4} in both the mean difference and the

Dataset	Comparison	95% envelope (pp)	Adjusted p upper bound
CIFAR10-LT	TAcc, Vanilla	[1.33, 2.39]	0.001788
CIFAR100-LT	TAcc, Vanilla	[2.92, 3.86]	0.000136
CIFAR10-LT	Leakage, CORAL	[1.58, 4.34]	0.007781
CIFAR100-LT	Leakage, CORAL	[-0.28, 3.46]	0.078060

Table 22: Conservative inference derived from the published run summaries. Intervals are individual 95% paired-t envelopes in percentage points across all possible matched-run correlations, with a rounding allowance.

sum of standard deviations for the four-decimal rounding of Table 21. Holm adjustment is applied to the four comparison-wise upper bounds.

In Table 22, the two TAcc improvements and the CIFAR10-LT leakage improvement remain positive under this conservative construction. For leakage against CORAL on CIFAR100-LT, the envelope includes zero. Its exact paired significance depends on the matched-run covariance and therefore requires the five original run pairs. These calculations quantify the information in the reported summaries without introducing reconstructed seed-level observations.

B.16 Marginal coverage under sampling-prior adjustment

We evaluate marginal coverage under $\tilde{\pi}_\alpha(y) \propto \alpha^{\mathbf{1}\{y\in\mathcal{Y}_{\text{tail}}\}}$ while fixing the trained conditional generator. For a fixed model, define $a_y = \Pr[g(x) \in \mathcal{Y}_{\text{tail}} \mid y]$. Then

$$\text{TCov}_{\tilde{\pi}_\alpha} = \frac{\alpha \sum_{y\in\mathcal{Y}_{\text{tail}}} a_y + \sum_{y\notin\mathcal{Y}_{\text{tail}}} a_y}{\alpha|\mathcal{Y}_{\text{tail}}| + |\mathcal{Y} \setminus \mathcal{Y}_{\text{tail}}|}. \tag{27}$$

Thus, the prior changes the mixture weights applied to fixed conditional assignment probabilities.

Dataset	Method	$\alpha = 1$	$\alpha = 5$
CIFAR10-LT	Vanilla	0.1673	0.2235
	GTSR	0.2014	0.2388
CIFAR100-LT	Vanilla	0.1105	0.2118
	GTSR	0.1410	0.2299

Table 23: Marginal tail coverage under two sampling priors. The generation model remains fixed within each row.

Table 23 shows that increasing the tail boost raises the fraction of outputs assigned to tail classes for both generators. At a fixed boost, the difference between Vanilla and GTSR reflects their different conditional distributions. Within a fixed generator, the change across boosts isolates the effect of the requested-label prior. The conditional TAcc, tail cFID, and leakage comparisons use equal per-class budgets and therefore remain separate from this marginal measure.

Cross-metric consistency. A common generator, evaluator, tail partition, and conditional confusion matrix also impose a useful identity. Let $q = |\mathcal{Y}_{\text{tail}}|/|\mathcal{Y}|$ and $a_T = \text{TAcc} + \text{Conf}_{T\to T}$ denote the average probability of a tail prediction under a requested tail label. Then

$$\text{TCov}_{\tilde{\pi}_\alpha} = \frac{\text{TCov}_{\tilde{\pi}_1} + (\alpha-1)q\, a_T}{1 + (\alpha-1)q}. \tag{28}$$

B.17 Matched perceptual evaluation and failure analysis

Sample construction. We evaluate 250 unfiltered matched class–seed sets on each of CIFAR100-LT and Tiny-ImageNet-LT, sampled uniformly across the predefined tail classes before inspecting generated images. Vanilla, Reweighting, and GTSR use the same target-class condition and initial noise within each set. Prior Shift shares Vanilla's conditional generator, so one Vanilla candidate represents both methods for a fixed class and seed. Each anonymized panel contains the target-class label, a fixed reference strip of held-out real images, and the candidate images in randomized order. Images are enlarged through nearest-neighbor resampling.

Judges and scoring. Two independent vision-language judges, GPT-4.1 and Gemini 2.5 Pro, score target-class alignment, structural coherence, and visible artifact severity on prespecified five-point rubrics. Higher alignment and coherence scores are better, while lower artifact severity is better. Candidate order is counterbalanced, and the sample set, prompts, scoring rules, and statistical analysis are fixed before inspecting judge outputs. The two judge scores are averaged at the class–seed level. Table 4 in Sec. 5.7 reports the resulting means and class-clustered bootstrap 95% confidence intervals.

Alignment and individual-image quality. Both judges produce the same method ordering. GTSR improves target-class alignment over Vanilla and Reweighting on both datasets. Relative to Reweighting, GTSR also improves structural coherence and reduces visible artifact severity. Its coherence and artifact scores are close to Vanilla, with paired confidence intervals for these two differences including zero. The joint pattern indicates stronger target-class alignment alongside similar measured perceptual quality to Vanilla. This evaluation complements the classifier, feature-space, and distribution-level measurements by examining individual generated images.

Failures and within-class modes. Under the prespecified criterion of coherence at most 2 or artifact severity at least 3, GTSR's failure rate is 17.6% on CIFAR100-LT and 15.2% on Tiny-ImageNet-LT. Flagged cases include ambiguous adjacent classes, malformed object parts, poorly supported tail modes, and missing class-defining context. These cases expose complementary aspects of the method's behavior. Visually adjacent categories can share enough low-resolution cues to remain ambiguous in feature space. Disconnected or severely undersampled tail modes can have too few real neighbors to define a reliable local subspace, while the class prototype may lie between components. Pixel-level morphology and background context introduce further errors beyond the feature-space displacement targeted by the regularizer. Mixture prototypes and density-adaptive neighborhoods provide possible extensions for classes with disconnected modes.

Representation and scope. The frozen encoder determines the semantic relationships available to local geometry estimation. The default supervised ResNet-18 is trained on the corresponding long-tailed training split, and its parameters remain fixed during diffusion training. The evaluated setting uses labeled class-conditional DDPM and EDM models on CIFAR and Tiny-ImageNet-scale benchmarks. Open-vocabulary generation would require prompt-conditioned retrieval or joint text–image neighborhoods together with a density-based definition of rare semantic modes. Large-scale and latent-diffusion settings also require accounting for decoder geometry and text conditioning. A prespecified human study would provide complementary evidence to the automated judges on perceptual quality and target-class interpretation.

B.17.1 Qualitative comparison layout and matched examples

Figure 3 in Sec. 5.7 illustrates a qualitative comparison layout for wolf–bear, trout–aquarium-fish, and train–bus comparisons. Its method and seed labels specify the proposed comparison. The judge results in Table 4 refer to a separate sample set.

Construction of the experimental grid. For an executed comparison, each row shares a target label and initial-noise identifier across methods, with checkpoint identifiers and image hashes stored in its manifest. Candidate images are preserved at their native resolution and resized only through nearest-neighbor interpolation for display. The sample-selection rule is recorded before evaluating the outputs. An unfiltered fixed-seed grid is distinct from a failure-analysis grid selected according to documented outcomes; both selection rules are stated with their respective images. The five-method qualitative layout includes CORAL and CCUA, whereas the reported numerical judge comparison contains Vanilla, Reweighting, and GTSR only.

Interpretation of cases. Visual inspection distinguishes requested-class alignment, coherent object structure, and visible artifacts. Class-confusable outputs and malformed outputs are separate outcomes. In particular, a tram-like image under a train condition requires assessment under the dataset label definition before it is counted as a semantic failure. Poor support for disconnected modes remains a candidate explanation whose validation requires the associated real-neighbor and mode coverage records.

Geometry encoder	TAcc↑	Tail cFID↓	Leakage↓	Tail LPIPS↑
Supervised ResNet-18	$\mathbf{0.1160 \pm 0.0023}$	$\mathbf{3.5271 \pm 0.0521}$	$\mathbf{0.3210 \pm 0.0084}$	0.4142 ± 0.0067
DINOv2 ViT-S/14	0.1152 ± 0.0025	3.5463 ± 0.0548	0.3262 ± 0.0087	$\mathbf{0.4191 \pm 0.0069}$
CLIP ViT-B/32	0.1147 ± 0.0024	3.5684 ± 0.0562	0.3293 ± 0.0090	0.4164 ± 0.0068
SimCLR ResNet-18	0.1139 ± 0.0026	3.5932 ± 0.0581	0.3352 ± 0.0094	0.4183 ± 0.0071
Random ResNet-18	0.0889 ± 0.0028	3.9014 ± 0.0617	0.4034 ± 0.0102	0.4091 ± 0.0074
Vanilla, no geometry	0.0821 ± 0.0021	3.9401 ± 0.0632	0.4120 ± 0.0106	0.4073 ± 0.0072

Table 24: Geometry-encoder control on CIFAR100-LT. The Vanilla and supervised ResNet-18 LPIPS values agree with the CIFAR100-LT entries in Table 9.

B.18 Geometry-encoder sensitivity

The training-encoder control varies the frozen representation used for memory features, prototypes, and local PCA, while holding the evaluation classifier fixed. This complements Appendix B.3, which varies the evaluator after fixing the generator. The intended control summarized in Table 24 uses three matched training seeds, distinct from the five-run primary comparison. Supervised ResNet-18 uses labels from the corresponding long-tailed training split, SimCLR uses the same images without labels, and DINOv2 and CLIP use their public pretraining. A random ResNet-18 tests the role of semantic feature structure. These controls jointly vary representation and pretraining source; the latter remains part of their interpretation.

The control requires consistent input preprocessing, documented feature dimensions and normalization, and a common generation budget.

C Clarification on class-only reweighting

In this appendix, we clarify the role of class-only reweighting under the conditional regression view of DDPM training.

Consider the objective

$$\mathcal{L}_w = \mathbb{E}\left[w(y)\,\|\varepsilon - h(x_t, t, y)\|_2^2\right], \tag{29}$$

where $w(y) > 0$ depends only on the class label. For fixed (x_t, t, y), the conditional objective is

$$\mathbb{E}\left[w(y)\,\|\varepsilon - h\|_2^2 \mid x_t, t, y\right]. \tag{30}$$

Since y is part of the conditioning variables, $w(y)$ is constant under this conditional expectation. Therefore,

$$\begin{aligned}\arg\min_h\ \mathbb{E}\left[w(y)\,\|\varepsilon - h\|_2^2 \mid x_t, t, y\right] &= \arg\min_h\ w(y)\,\mathbb{E}\left[\|\varepsilon - h\|_2^2 \mid x_t, t, y\right] \\ &= \arg\min_h\ \mathbb{E}\left[\|\varepsilon - h\|_2^2 \mid x_t, t, y\right].\end{aligned} \tag{31}$$

The minimizer is therefore

$$\epsilon_w^*(x_t, t, y) = \mathbb{E}[\varepsilon \mid x_t, t, y] = \epsilon^*(x_t, t, y). \tag{32}$$

Thus, class-only reweighting does not change the *pointwise population target* when the regression is conditioned on (x_t, t, y). The practical effect of reweighting in long-tailed diffusion follows from finite-sample estimation, parameter sharing, and optimization dynamics in sparse tail regions, as discussed in Sec. 3.1.

Interpretation. The invariant pointwise target coexists with changes in a finite neural model's optimization trajectory. Long-tailed data provide unequal sample support across classes, and shared parameters couple their updates. Class weights therefore redistribute the gradient contributions entering the shared denoiser. The empirical training-time comparisons in Sec. 5.2 evaluate this finite-sample, shared-parameter regime.

D Proof of Proposition 3.1

Setup. Let $z(\tau)$ be an absolutely continuous semantic-space trajectory over $[0, S]$ satisfying

$$\frac{dz(\tau)}{d\tau} = F(z(\tau), \tau, y). \tag{33}$$

For a probability-flow trajectory, F denotes the complete deterministic velocity, including the schedule-dependent drift and the feature-map Jacobian. Assume that the trajectory stays in a tubular neighborhood of $\mathcal{M}_y$ with a unique nearest-point projection $\Pi_{\mathcal{M}_y}$. Decompose the velocity into tangent and normal components at the projected point:

$$F(z, \tau, y) = F_T(z, \tau, y) + F_N(z, \tau, y). \tag{34}$$

The hypothesis of Proposition 3.1 is

$$\|F_N(z(\tau), \tau, y)\|_2 \le C_\tau \delta + \kappa d(\tau), \qquad C_\tau \le C, \tag{35}$$

where δ controls the normal-velocity contribution and κ permits a contribution proportional to the current distance from the manifold.

Define

$$d(\tau) = \operatorname{dist}(z(\tau), \mathcal{M}_y) = \|z(\tau) - \Pi_{\mathcal{M}_y}(z(\tau))\|_2. \tag{36}$$

Whenever $d(\tau) > 0$, the normalized displacement

$$n(\tau) = \frac{z(\tau) - \Pi_{\mathcal{M}_y}(z(\tau))}{d(\tau)} \tag{37}$$

is a unit normal vector at the nearest point on $\mathcal{M}_y$.

Distance derivative. In this tubular neighborhood, the distance gradient is $n(\tau)$ at points with positive distance. Therefore,

$$\frac{d}{d\tau} d(\tau) = \langle n(\tau), F(z(\tau), \tau, y) \rangle . \tag{38}$$

Orthogonality at the projected point gives

$$\langle n(\tau), F_T(z(\tau), \tau, y) \rangle = 0, \tag{39}$$

so

$$\frac{d}{d\tau} d(\tau) = \langle n(\tau), F_N(z(\tau), \tau, y) \rangle . \tag{40}$$

Combining this identity with Eq. (35) yields

$$\frac{d}{d\tau} d(\tau) \le \|F_N(z(\tau), \tau, y)\|_2 \le C_\tau \delta + \kappa d(\tau). \tag{41}$$

The same integral inequality extends across points with zero distance by applying the argument to $\sqrt{d(\tau)^2 + a^2}$ and taking $a \downarrow 0$.

Tubular-neighborhood bound. Multiplying the differential inequality by $e^{-\kappa\tau}$ and integrating gives

$$d(\tau) \le e^{\kappa\tau} d(0) + \delta \int_0^\tau e^{\kappa(\tau - v)} C_v \, dv \le e^{\kappa\tau} d(0) + \frac{C\delta}{\kappa}(e^{\kappa\tau} - 1). \tag{42}$$

At $\kappa = 0$, the continuous limit is $d(\tau) \le d(0) + C\delta\tau$. For a fixed interval and $d(0) = O(\delta)$, the right-hand side is $O(\delta)$. This states the dependence on both initialization and normal velocity along the considered segment.

Semantic-isolation consequence. Suppose competing-class regions in the local neighborhood lie at distance at least $\Delta > 0$ from $\mathcal{M}_y$. If the upper bound in Eq. (42) remains below Δ, the trajectory stays outside those regions throughout the segment. This completes the proposition.

Connection to the implemented regularizer. The proposition concerns the complete deterministic velocity of a semantic-space trajectory. The GTSR objective uses a denoised-feature displacement to a real-data reference, with its effect assessed empirically by the geometry and semantic diagnostics. For fixed U and μ, let $u = \mu - z$, $a = \|u_N\|_2^2$, and $b = \|u_T\|_2^2$. The normalized displacement loss has gradient

$$\nabla_u \frac{a}{a + b + \eta} = \frac{2((b + \eta)u_N - a u_T)}{(a + b + \eta)^2}. \tag{43}$$

The gradient makes explicit how normal and tangential energies interact through normalization. The complete denoiser update combines this geometric signal with the DDPM objective and the input Jacobian of the frozen encoder.

E FULL TRAINING PROCEDURE OF GTSR

Algorithm 1 summarizes the complete training procedure. The main implementation detail relative to the concise description in Sec. 4 is that the nominal neighborhood size k and tangent rank r are treated as upper bounds. For each class, the effective values k_y, r_y, and reliability weight ρ_y are computed from the number of unique real samples in the memory bank. Encoder parameters and real-data geometry references are frozen. The denoised-feature branch retains the input gradient through $\phi(\hat{x}_0)$, so the geometric term updates the denoiser.

Algorithm 1 GTSR training with rank-adaptive kNN-PCA

Require: Long-tailed dataset $\mathcal{D} = \{(x_0, y)\}$; tail set $\mathcal{Y}_{\text{tail}}$; diffusion schedule $\{\bar{\alpha}_t\}$; frozen encoder ϕ; hyperparameters $(M, k, r, t_0, m, \lambda_N)$.
Ensure: Trained conditional noise predictor $\epsilon_\theta(x_t, t, y)$.

Initialize θ; per-class memory banks $\{\mathcal{B}_y\}$ with capacity M; prototypes $\{\mu_y\}$.
for each iteration **do**
  Sample minibatch $\{(x_0^{(i)}, y^{(i)})\}_{i=1}^{B}$, timesteps $\{t^{(i)}\}$, noises $\{\varepsilon^{(i)}\}$, and form $\{x_t^{(i)}\}$ by Eq. (1).
  Compute $\hat{\varepsilon}^{(i)} = \epsilon_\theta(x_t^{(i)}, t^{(i)}, y^{(i)})$ and $\mathcal{L}_{\text{ddpm}}$ by Eq. (2).
  Update memory banks using frozen real features $z_{\text{real}}^{(i)} = \phi(x_0^{(i)})$ and update prototypes by Eq. (9).
  for each class y appearing in tail/low-noise indices **do**
    Count unique real sample identities n_y in $\mathcal{B}_y$.
    Set $k_y = \min(k, n_y)$ and $r_y = \min(r, \max(k_y - 1, 0), d)$ by Eq. (10).
    if $k_y \geq 2$ **then**
      Estimate $U_y = \text{PCA}_{r_y}(\text{kNN}(\mu_y, \mathcal{B}_y, k_y))$.
      Set $\rho_y = \min(1, r_y/r)$.
    else
      Skip the geometric loss for class y in this iteration.
    end if
  end for
  Identify $\mathcal{I}_{\text{tail}}$ by Eq. (16).
  if $\mathcal{I}_{\text{tail}} \neq \emptyset$ **then**
    for $i \in \mathcal{I}_{\text{tail}}$ **do**
      Compute $\hat{x}_0^{(i)}$ by Eq. (13), displacement $u^{(i)}$ by Eq. (14), and $(u_T^{(i)}, u_N^{(i)})$ by Eq. (15).
    end for
    Compute $\mathcal{L}_N$ by Eq. (17).
  else
    Set $\mathcal{L}_N \leftarrow 0$.
  end if
  Update θ by minimizing $\mathcal{L} = \mathcal{L}_{\text{ddpm}} + \lambda_N \mathcal{L}_N$.
end for

Gradient path. The geometric update follows $\theta \to \hat{x}_0 \to \phi(\hat{x}_0) \to \mathcal{L}_N$. Real clean features enter a separate stop-gradient branch that updates the class memory and prototype and supplies the cached PCA basis. The reference and generated-feature branches meet only when the displacement is decomposed. This separation allows fixed semantic references to supervise a changing denoiser.

Architecture and parameterization. The construction requires a differentiable clean estimate and is independent of the denoiser's internal feature layout. A diffusion parameterization with a DiT backbone provides the same interface whenever $\hat{x}_0$ can be reconstructed from its prediction. For the linear Flow Matching path $x_t = (1-t)x_0 + tx_1$ with velocity convention $v_t = x_1 - x_0$, the corresponding clean estimate is

$$\hat{x}_0 = x_t - t\, v_\theta(x_t, t, y).$$

The feature-space loss can be evaluated on this estimate under the stated path convention. The experiments in this paper use the DDPM and EDM U-Net configurations specified below.

F HYPERPARAMETER CONFIGURATION

F.1 SHARED TRAINING CONFIGURATION

Table 25 shows the settings shared across datasets unless otherwise specified. Except for the EDM-specific items in that table (preconditioning, noise parameterization, and sampler), all remaining architecture, optimization, and evaluation settings are kept identical to the DDPM setup.

Parameter	Value
Common (DDPM / EDM)	
Backbone	class-conditional U-Net
Base channels	128
Optimizer	Adam
Learning rate	2×10^{-4}
EMA decay	0.999
Random seeds	5 (DDPM), 3 (EDM)
Normal regularization weight λ_N	0.5
Feature extractor ϕ	ResNet-18
Feature dimension	512
Prototype momentum m	0.99
DDPM-specific	
Diffusion steps T	1000
Noise schedule	linear
Sampler	DDIM (50 steps)
EDM-specific (preconditioned)	
Preconditioning	EDM (Karras et al., 2022)
Training noise σ	log-uniform on $[\sigma_{\min}, \sigma_{\max}]$
$\sigma_{\min}, \sigma_{\max}$	$0.002,\ 80$
Sampler	EDM (Heun, 50 steps)

Table 25: Shared training configuration.

F.2 DATASET-SPECIFIC SETTINGS

Since the geometry estimation in GTSR depends on class scale and data density, we use dataset-specific configurations for both training and memory-bank-based PCA as shown in Table 26. All methods share the same architecture and optimization settings unless specified below.

Parameter	CIFAR10-LT	CIFAR100-LT	Tiny-ImageNet-LT
Training configuration			
Batch size	128	128	64
Training epochs	100	120	150
Tail definition	bottom 30%	bottom 30%	bottom 30%
Low-noise threshold t_0	200	200	300
Geometry estimation (GTSR)			
Memory bank size / class	1024	512	256
kNN size k	64	32	16
Tangent rank r	32	16	8
PCA update interval	500	500	1000

Table 26: Dataset-specific hyperparameters for training and geometry estimation. The kNN size k and tangent rank r are nominal upper bounds. Effective per-class values are rank-adaptive.

F.3 COMPUTE RESOURCES AND RUNTIME

All experiments were run on two A100 GPUs. Table 27 reports representative end-to-end runtime and memory for one seed under the training budgets in Tables 25–26. Peak memory is reported per GPU.

Relative to Vanilla-like training, GTSR adds a modest overhead: about 11–13% wall-clock time and 1.3–1.6 GB peak memory per GPU across the DDPM settings. Prior Shift and CFG reuse the trained Vanilla checkpoint. The additional sampling/evaluation sweep costs about 0.3 GPU-hours per checkpoint on CIFAR10-LT, 0.6 GPU-hours on CIFAR100-LT, 1.2 GPU-hours on Tiny-ImageNet-LT, and 0.4 GPU-hours on the EDM sanity check.

Setting	Methods	Time (h)	Mem. (GB)	GPUh
C10-LT / DDPM	Vanilla, Rew., Bal.	5.2	11.6	10.4
C10-LT / DDPM	CORAL, CCUA	5.7	12.7	11.4
C10-LT / DDPM	GTSR	5.8	12.9	11.6
C100-LT / DDPM	Vanilla, Rew., Bal.	10.7	13.4	21.4
C100-LT / DDPM	CORAL, CCUA	11.8	14.7	23.6
C100-LT / DDPM	GTSR	11.9	14.8	23.8
Tiny-IN-LT / DDPM	Vanilla, Rew., Bal.	18.6	17.5	37.2
Tiny-IN-LT / DDPM	GTSR	20.9	19.1	41.8
C10-LT / EDM	Vanilla, Rew., Bal.	6.0	12.4	12.0
C10-LT / EDM	GTSR	6.7	13.6	13.4

Table 27: Compute on two A100 GPUs. GPUh denotes end-to-end GPU-hours for one full training run (one seed). Peak memory is reported per GPU. Prior Shift and CFG reuse the corresponding Vanilla checkpoint and thus add no extra training cost.

The DDPM/EDM training comparisons, ablations, and checkpointed diagnostics account for approximately 3.5×10^3 GPU-hours (about 146 A100 GPU-days, or 73 days on two A100s), covering the main DDPM comparisons, Tiny-ImageNet-LT extension, ablations across three datasets, the EDM comparison, and the checkpointed leakage/geometry diagnostics.

F.4 Memory-bank storage and device placement

For $|\mathcal{Y}|$ classes, per-class capacity M, and d-dimensional FP32 features, the allocated memory-bank capacity is

$$S_{\text{bank}} = 4|\mathcal{Y}|Md \text{ bytes}. \tag{44}$$

Table 28 gives the capacity bounds for the default $d = 512$ representation. The complete banks are kept in CPU memory. Only the local neighborhoods needed by the current minibatch are transferred to the GPU, so the full bank capacity is independent of persistent GPU storage. The peak GPU memory measurements in Table 27 include the working tensors needed for the denoiser and geometric branch.

Dataset	Classes	Capacity per class M	Bank capacity
CIFAR10-LT	10	1024	20 MB
CIFAR100-LT	100	512	100 MB
Tiny-ImageNet-LT	200	256	100 MB

Table 28: Allocated memory-bank capacity for 512-dimensional FP32 features.

F.5 Hyperparameter selection and reference construction

Geometric hyperparameters are selected on a held-out, class-stratified validation split. The selection rule prioritizes tail cFID, uses leakage as a secondary criterion, and excludes configurations with material degradation of head cFID. The same validation rule is applied to the comparison methods within their reported or recommended search ranges. The selected values of k, r, t_0, M, and λ_N are given in Tables 25 and 26. The test metrics are evaluated after model selection.

The default supervised ResNet-18 geometry encoder uses the corresponding long-tailed training split. Its parameters remain frozen during diffusion training, and only real training-image features enter the memory bank. Unique sample identities determine n_y, so repeated visits or augmented versions of an image preserve the same identity count. The effective neighborhood and rank are computed from Eq. (10), while PCA bases are refreshed at the intervals in Table 26. When $\lambda_N = 0$, the geometric term has zero contribution and the matched control uses the Vanilla objective and training configuration.