GENERATE TO TEACH: ADAPTING FLOW MODELS WITH CLASS-CONTRASTIVE LORA FOR FEW-SHOT CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

For synthetic data to help few-shot classification, generated images must preserve the distinctions among the target classes. Yet standard generator adaptation fits each support image only to its assigned prompt, leaving these relative distinctions implicit. We introduce TASKREL, a class-contrastive LoRA method for flow-matching generators that turns the generator's own conditional prediction errors into task supervision. For each support example, TASKREL evaluates the correct and competing class prompts on the same noisy latent and target velocity, so the class condition is the only variable. A finite-margin hinge favors the correct condition alongside positive flow-matching fitting. An evolving relation map selects hard competitors, while a scheduler-derived constant controls comparison strength independently of selection. The resulting shared generator is trained solely from labeled supports and its own predictions, without downstream-classifier feedback or inference-time filtering. Across six benchmarks with Stable Diffusion 3.5 Medium, TASKREL reaches 67.50% mean accuracy with a scratch ResNet-50, improving over the matched flow-matching control by 3.36 points and LoFT, the strongest external comparator, by 3.76 points. Its frozen synthetic pools retain gains across scratch ViT and CLIP learners and across 25–100 generated images per class. Thus, class relations provide a direct adaptation signal for reusable synthetic training data while preserving ordinary class-conditioned sampling.

1 INTRODUCTION

Few-shot image classification requires learning discriminative representations from a small labeled support set (Chen et al., 2019; Tian et al., 2020). In this data-limited regime, pretrained generative models can expand this set using visual knowledge learned from large image collections (He et al., 2023; Azizi et al., 2023). For classification, useful synthetic images must preserve the attributes that separate each target class from the alternatives in the task. Standard generator adaptation provides only indirect pressure toward this goal because each support image is fitted under its assigned class condition. The objective does not explicitly compare that condition with the competing classes that determine the classification problem.

Recent methods incorporate task-specific information through several adaptation mechanisms. DataDream learns compact weight updates for the target dataset (Kim et al., 2024). LoFT fuses image-specific adapters during generation, while ChimeraLoRA combines shared and instance-specific factors (Kim et al., 2025; 2026). Contextual augmentation varies backgrounds, poses and other factors surrounding the subject (Yang et al., 2026). These approaches improve the fidelity and diversity of few-shot synthetic data. Their ordinary class-conditioned fitting objectives still supervise each observed image–prompt pair in isolation from competing class conditions.

This omission is important because a classification label acquires meaning relative to the other labels in the task (Fig. 1). An aircraft image, for example, must convey the structural details that distinguish its type from visually similar variants. A recognizable subject, plausible background and varied viewpoint can coexist with weak evidence for those differences. With few supports and limited adaptation capacity, positive fitting can leave fine class distinctions underconstrained. We

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.

therefore ask whether comparisons across class conditions can provide the missing supervision while preserving the objective that fits real examples from every class.

A downstream classifier can provide such a signal by exposing failures that guide data generation (Ma et al., 2025). This couples generator adaptation to a particular learner and its evolving decision function. We seek a signal from the generator itself so that one adapted model can produce a reusable training set before the downstream learner is chosen. MBTI provides a relevant precedent by comparing inter-class denoiser predictions when learning class-specific prompt embeddings (Chae et al., 2026). Our setting adapts shared generator weights across all classes. Each comparison can therefore alter the same parameters used to fit every class, which makes the interaction between positive fitting and class-relative supervision a central design issue.

We introduce TASKREL, which turns the generator's prediction errors into class-relative supervision for low-rank adaptation (LoRA) (Hu et al., 2022). Flow matching provides a direct way to construct these comparisons (Lipman et al., 2023; Liu et al., 2023). Each training state interpolates a real image representation and Gaussian noise, yielding a known target velocity from the two endpoints. TaskRel evaluates the correct and competing class conditions on the same noisy state and against the same target. The class condition is the only variable within a comparison. The resulting error gap measures how much better the correct condition predicts a common regression target, giving shared generator weights a direct signal about class distinctions without image generation or downstream-classifier feedback during adaptation.

TaskRel combines this paired comparison with generative fitting. A finite-margin hinge encourages an error advantage for the correct condition and stops pairwise pressure once the margin is satisfied. A directed relation map tracks prediction-error gaps to sample competing classes during adaptation. A scheduler-derived coefficient sets the comparison-loss strength. This design separates comparison selection from optimization strength. The relation statistics, margin and weighting rule are derived from labeled supports, the training scheduler and the generator's own predictions.

After adaptation, the relation map is discarded and TaskRel uses ordinary class-conditioned sampling with no classifier-based ranking or rejection. The resulting image pools can be reused across recognition models without further generator adaptation. Across six datasets with Stable Diffusion 3.5 Medium, TaskRel reaches 67.50% mean accuracy with a scratch ResNet-50. This improves the matched positive-only flow-matching control by 3.36 percentage points and LoFT, the strongest external comparator, by 3.76 points. Controlled experiments separate the effect of the class-comparison objective from the allocation of competing conditions. The same frozen pools also retain gains with scratch ViT and CLIP learners and across synthetic-data budgets from 25 to 100 images per class.

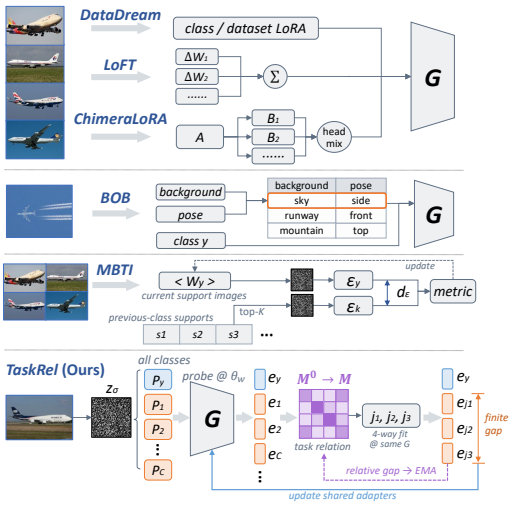


Figure 1: Positioning TaskRel among few-shot synthesis methods. Prior work improves synthetic data through generator adaptation, instance-level fusion, contextual variation, or class-token optimization. TaskRel compares class-conditioned prediction errors on a shared noisy state and target to supervise shared generator weights. Observed error gaps also guide the selection of competing classes during adaptation. All supervision comes from labeled supports and generator predictions, without feedback from a downstream learner.

We make three contributions. (1) We formulate a class-contrastive adaptation objective that compares correct and alternative conditions on a shared noisy state and target, using paired prediction errors to supervise shared generator weights. (2) We provide a conditional-regression interpretation of these comparisons and a training design that combines a finite-margin objective, adaptive class selection and separately calibrated comparison strength. (3) We conduct a controlled evaluation across visual domains, objective variants, learner architectures and synthetic-data budgets, testing the recognition utility and reusability of the resulting training sets under consistent protocols.

2 Related work

Generative modeling and adaptation. Diffusion and score models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b), efficient sampling (Song et al., 2021a), latent diffusion and transformers (Rombach et al., 2022; Peebles & Xie, 2023), and flow matching (Lipman et al., 2023; Liu et al., 2023; Esser et al., 2024; Stability AI, 2024) provide the generative foundations. Textual inversion learns prompt embeddings, DreamBooth adapts model weights, and LoRA parameterizes low-rank updates (Gal et al., 2023; Ruiz et al., 2023; Hu et al., 2022). TaskRel studies the class supervision used to adapt these weights.

Synthetic training data. Generated images support classification and representation learning (He et al., 2023; Azizi et al., 2023; Tian et al., 2023). SDEdit preserves input structure through noising and denoising (Meng et al., 2022). DA-Fusion learns concept tokens, GeNIe generates challenging cross-class examples, and Diff-II interpolates inverted concepts (Trabucco et al., 2024; Koohpayegani et al., 2025; Wang & Chen, 2025). DataDream adapts shared weights, LoFT fuses image-specific adapters, and ChimeraLoRA combines shared/instance factors with semantic boosting (Kim et al., 2024; 2025; 2026). BOB varies background and pose (Yang et al., 2026). TaskRel adds class-comparison supervision to dataset-shared adaptation.

Comparison scores and supervision sources. Representation contrast and negative selection shape discriminative learning (Chen et al., 2020; He et al., 2020; Khosla et al., 2020; Wu et al., 2017; Robinson et al., 2021). MBTI compares matched-state denoiser predictions while learning class-specific prompt embeddings (Chae et al., 2026). TaskRel adapts shared weights using a finite prediction-error margin and evolving class relations. Contrastive Flow Matching uses an independently sampled negative velocity target, while VeCoR regularizes velocity directions (Stoica et al., 2025; Hong et al., 2026). TaskRel changes the condition on a common noisy input and target. Reward optimization (Black et al., 2024; Clark et al., 2024), external representation alignment (Yu et al., 2025), failure-directed adaptation (Ma et al., 2025) and DCS's classifier–diffusion boosting loop (Wu et al., 2025) introduce additional supervision. Diffusion Classifier derives class scores from denoising errors (Li et al., 2023). TaskRel uses prediction-error comparisons to train the data generator from labeled supports, without external rewards or downstream-classifier feedback.

3 Class-contrastive adaptation with TaskRel

TaskRel adapts a shared conditional flow model through positive-only warm-up, class-relation probing and class-contrastive continuation (Fig. 2). The probe initializes the comparison sampler and calibration statistics. Continuation jointly optimizes generative fitting and a finite-margin comparison loss. The complete procedure is given in Sec. A.

3.1 Conditional prediction task

Let S_y contain n labeled images from class y among C classes, and let p_y be its class prompt. We adapt one generator to $S = \bigcup_y S_y$ using shared LoRA parameters θ in its transformer and two CLIP text encoders. The pretrained weights remain fixed. After adaptation, the generator supplies labeled images for classifier training.

A fixed variational autoencoder encodes each image as a latent z (Kingma & Welling, 2014; Rombach et al., 2022). Flow matching constructs a regression task by interpolating z with Gaussian noise ϵ. At noise level σ, the noisy input, known target velocity and prediction error under condition p_j are

$$z_\sigma = (1-\sigma)z + \sigma\epsilon, \qquad u = \epsilon - z, \qquad e_j = \frac{1}{d}\|v_\theta(z_\sigma, t, p_j) - u\|^2. \tag{1}$$

Here v_θ predicts velocity, t is the scheduler timestep associated with σ, and d is the number of latent elements. The same input, time and target are used for every condition evaluated on an anchor.

Our Stable Diffusion 3.5 Medium adaptation implementation forms a clean-latent estimate $\hat{z}_y = z_\sigma - \sigma v_\theta(z_\sigma, t, p_y)$ and compares it with z (Esser et al., 2024; Stability AI, 2024). In exact arithmetic,

$$\ell_+ = \frac{1}{d}\|\hat{z}_y - z\|^2 = \sigma^2 e_y. \tag{2}$$

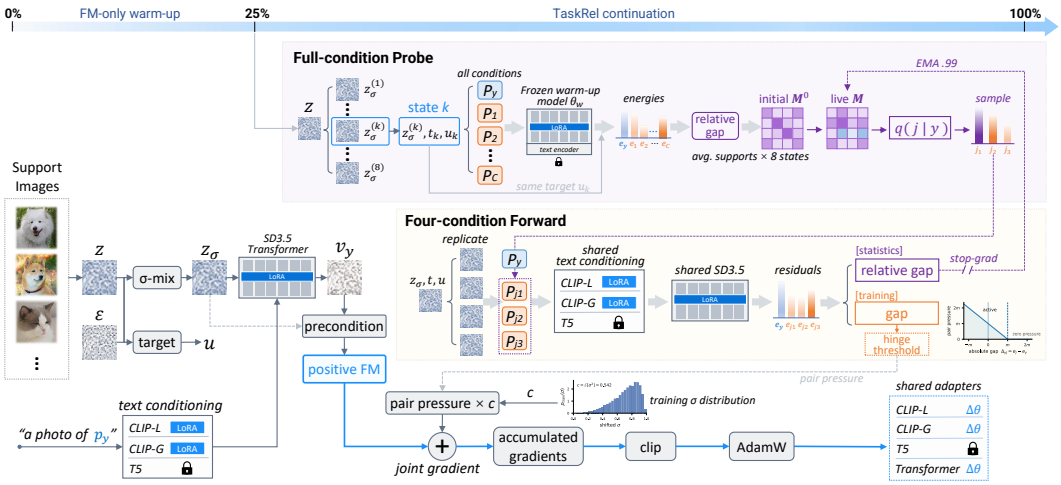


Figure 2: TaskRel training overview. Positive-only warm-up establishes the reference generator. A shared-state probe initializes the directed class-relation map and calibration statistics. During continuation, the relation map selects competing conditions whose paired prediction errors provide finite-margin supervision alongside generative fitting. Relation statistics are updated from detached error gaps. After adaptation, the relation map is discarded and the generator uses ordinary class-conditioned synthesis. Classifier training occurs only during downstream evaluation.

Training evaluates the clean-latent expression directly. The identity exposes its σ^2 weighting of velocity error, which provides the reference scale for the auxiliary objective.

3.2 Class-relative comparison objective

For an example of class y, the gap $\Delta_{yj} = e_j - e_y$ measures the correct condition's error advantage over an alternative j. Evaluating both conditions on the same regression task isolates their responses to the class prompt. Related shared-corruption comparisons extract class information from diffusion models (Li et al., 2023).

We require a finite error margin between the correct and competing conditions. For anchor i with label y_i, let N_i contain $K_n = 3$ distinct alternative labels. The auxiliary loss is

$$R_i = \frac{1}{K_n} \sum_{j \in N_i} [m - (e_{ij} - e_{iy_i})]_+. \tag{3}$$

Here $[a]_+ = \max(a, 0)$ and m is fixed by the warm-up calibration in Sec. 3.3. Both prediction branches participate in backpropagation when their gap is below m. For an active term, the derivative with respect to the alternative error is $-1/K_n$. The hinge derivative with respect to the correct-condition error is the active-comparison fraction in $[0, 1]$. Satisfied comparisons have zero hinge derivative. These energy derivatives specify the supervision, while shared parameters and the optimizer determine the model update.

The error gap also admits a conditional-regression interpretation. Let H, U and Y denote the random noisy state, target velocity and class label. At $h = (z_\sigma, t)$, define $\mu_y(h) = \mathbb{E}[U \mid H = h, Y = y]$. Writing $v_j = v_\theta(h, p_j)$ gives

$$\mathbb{E}[e_j - e_y \mid h, y] = \frac{\|v_j - \mu_y(h)\|^2 - \|v_y - \mu_y(h)\|^2}{d}. \tag{4}$$

The conditional target variance cancels, so both predictions are compared with the same class-specific mean. At the unrestricted population optimum of positive-only conditional regression, $v_j = \mu_j(h)$ for each class. The expected gap then equals $\|\mu_j(h) - \mu_y(h)\|^2/d$ on their common support. This population reference characterizes the comparison signal. Its assumptions and proof appear in Sec. B, and downstream evaluation measures the utility of the resulting synthetic data.

3.3 Adaptive selection of competing classes

With three alternatives per anchor, the choice of conditions determines which class distinctions receive supervision. We sample from all other classes using a directed matrix of observed error gaps. Small gaps indicate close competitors on the prediction task, and negative gaps identify alternatives with lower error than the correct condition.

Positive-only warm-up produces a reference generator θ_w. At this checkpoint, all class prompts are evaluated on $K_g = 8$ shared corruption states per support image. We normalize each gap by its correct-condition error and average within the source class:

$$M^0_{yj} = \frac{1}{nK_g} \sum_{x \in S_y} \sum_{k=1}^{K_g} \frac{e_j(x,k;\theta_w) - e_y(x,k;\theta_w)}{e_y(x,k;\theta_w)}. \tag{5}$$

Thus M^0_{yj} summarizes the relative advantage of y over j. The sampler uses these normalized gaps, while the hinge uses absolute gaps. The probe also fixes $m = 0.03 \operatorname{median} e_y$, pooling true-condition errors over all supports and corruption states.

Initialize $M = M^0$ and draw three distinct alternatives without replacement using weights

$$q(j \mid y; M) \propto \exp(-M_{yj}/\tau_g), \quad j \neq y, \qquad \tau_g = \max\big(\operatorname{median}_{y \neq j} |M^0_{yj}|, 10^{-4}\big). \tag{6}$$

The temperature τ_g is fixed by the initial matrix. After each anchor, we update its observed directed entries with an exponential moving average of the detached relative gaps, using decay 0.99. Other entries retain their previous values. The updated matrix governs subsequent draws. Backpropagation differentiates the mean hinge over the sampled labels, treating the sampling decisions and relation statistics as constants. The auxiliary objective therefore follows the adaptive sampling distribution. All initialization and updates use the supports and generator predictions.

3.4 Loss weighting and optimization

The sampler controls which conditions are compared. Their influence on adaptation is set by the auxiliary coefficient. Since the positive objective weights velocity error by σ^2, we choose its mean as the constant multiplier of the hinge:

$$\boxed{L = \frac{1}{B} \sum_{i=1}^{B} \ell_{+,i} + \mathbf{1}_{s \geq T_w} \frac{c}{B} \sum_{i=1}^{B} R_i, \qquad c = \mathbb{E}[\widetilde{\sigma}^2] \simeq 0.542.} \tag{7}$$

Here B is the minibatch size, s is the zero-based update index, and $T_w = \lfloor T/4 \rfloor$ defines the warm-up boundary for T total updates. The coefficient $\widetilde{\sigma}$ is the numerical scheduler value used in training. Its second moment is computed once under the training timestep-sampling law, as specified in Sec. A.2.

Weighting each comparison by its own $\widetilde{\sigma}^2$ would couple its contribution to the noise level. A constant separates this dependence from average scalar strength. At fixed model and sampler state,

$$\mathbb{E}[\widetilde{\sigma}^2 R] - c\mathbb{E}[R] = \operatorname{Cov}(\widetilde{\sigma}^2, R). \tag{8}$$

The covariance describes the difference in expected loss despite equal mean multipliers. Gradient differences likewise depend on their coupling to noise level. The calibration concerns scalar strength, with gradient norms and AdamW proposals also determined by model state and loss geometry. The weighting-rule comparison under shared settings is reported in Tab. 15.

Within each minibatch, positive and auxiliary gradients accumulate before one clipped AdamW step (Loshchilov & Hutter, 2019). At generation, we discard the class-pair matrix and use the adapted model's ordinary sampler. Adapter, optimizer and sampling settings are listed in Sec. A.

4 Experiments

4.1 Evaluation setup

Benchmarks. We evaluate Oxford-IIIT Pets (Parkhi et al., 2012), DTD (Cimpoi et al., 2014), RGB EuroSAT (Helber et al., 2019), FGVC-Aircraft (Maji et al., 2013), Caltech101 (Li et al., 2004) and

Flowers102 (Nilsback & Zisserman, 2008), covering fine-grained categories, textures, land use and general objects. We use DISEF's released splits (Dall'Asen et al., 2025) for five datasets and the official Aircraft partitions, retaining 100 Caltech classes. Methods share 16 real training images per class, and each generator supplies 100 synthetic images per class. Evaluation uses the full real-test split, excluded from fitting. Class lists and split sizes are in Sec. C.

Baselines. DA-Fusion (Trabucco et al., 2024), DataDream (Kim et al., 2024), LoFT (Kim et al., 2025) and ChimeraLoRA (Kim et al., 2026) represent token-based augmentation, dataset-shared adaptation, image-adapter fusion and shared/instance factorization. DataDream denotes DataDream-dset. Real-only uses the 16 supports, while FM control matches TaskRel's positive fitting and adaptation horizon. All generator ports use SD3.5 Medium, retaining their defining mechanisms and method-specific parameterization. LoFT/Chimera share our class prompts, while DataDream generation follows its own training prompts. BOB (Yang et al., 2026) is evaluated separately under its five-shot context protocol (Sec. D.6). MBTI (Chae et al., 2026) lacks a verified author implementation for our port. Full baseline recipes and checkpoint choices are in Tab. 7 and Sec. C.3.

Implementation and evaluation. TaskRel and FM train rank-16 transformer/two-CLIP LoRA for 50 epochs, batch 8, using AdamW learning rates $10^{-4}/5 \times 10^{-6}$. TaskRel uses 25% positive warm-up, three competing labels and EMA decay 0.99, with the calibration in Sec. 3. Synthesis uses 512 pixels, a nominal 24-step Euler schedule and guidance 2, without image filtering. DA-Fusion uses image-to-image strength 0.5, giving 12 effective steps. Scratch ResNet-50 (He et al., 2016) trains for 100 epochs on the shuffled real/synthetic union using SGD, batch 32, peak learning rate 0.01 and 15-epoch warm-up followed by cosine decay. Real-only uses 100 ordinary epochs on its supports. Tables 1 and 2 report terminal real-test micro accuracy as mean $\pm$ sample SD over student seeds 0/1/2 on fixed pools. Learner-transfer and full-six budget studies use seed 0. All within-learner recipes are fixed across methods, with full precision, augmentation and optimizer details in Secs. A and C.5.

4.2 Main comparison

Tab. 1 compares all 42 method–dataset combinations under one student protocol. TaskRel reaches 67.50%, exceeding LoFT, the strongest external comparator, by 3.76 points and the matched FM control by 3.36 points. It has the highest mean on Pets, DTD, EuroSAT and Aircraft, while LoFT leads on Caltech and Flowers.

The matched FM comparison is positive in every domain. The largest gains are 7.39 points on Aircraft, 4.89 on DTD and 4.84 on EuroSAT, covering fine-grained shape, local texture and land-cover structure. TaskRel also exceeds each domain's strongest external method by 2.63, 5.38, 4.21 and 3.76 points on Pets, DTD, EuroSAT and Aircraft, respectively. These gains span several forms of visual discrimination under one adaptation rule. Its differences from DataDream are positive in all 18 paired student runs. LoFT retains the lead on Caltech and Flowers by 0.55 and 1.72 points, showing that the ranking still depends on the domain.

The FM control holds the supports, positive fitting objective and adaptation horizon fixed. This comparison measures the effect of the added class-comparison procedure at a common positive-update budget, while the external comparisons test it against other adaptation mechanisms. For example, LoFT and Chimera already introduce instance-specific variation, yet TaskRel has the stronger six-domain average. The result supports adding class-relative supervision to a shared adapter. A new support, generator and synthesis realization on three datasets retains positive FM-relative mean gains (Tab. 9).

4.3 Design choices and reusability

Tab. 2 separates the comparison objective from the allocation of alternative conditions. Dynamic uniform samples from all competing classes uniformly. Frozen soft retains the warm-up relation matrix, and TaskRel updates its observed entries during adaptation. These three rules share the hinge, three negatives per anchor and coefficient c. The FM control keeps positive fitting alone, while independent target uses an alternative target construction and loss. Supports, warm-up and learner training are held fixed, with terminal accuracy measured over three seeds.

Table 1: Six-domain test accuracy (%) with scratch ResNet-50. Each augmented training set contains 16 real supports and 100 synthetic images per class. Real-only uses the 16 supports. All students train for 100 epochs. Entries are mean ± SD over three student seeds. Avg. gives equal weight to the six datasets, and bold marks the highest mean in each column.

Method	Pets	DTD	EuroSAT	Aircraft	Caltech	Flowers	Avg.
Real only	14.15 ±0.59	17.38 ±1.05	50.84 ±1.17	10.28 ±1.85	53.02 ±0.96	44.20 ±1.02	31.64
DA-Fusion (ICLR2024)	35.90 ±0.93	32.72 ±1.18	51.88 ±0.62	38.77 ±0.57	72.75 ±0.36	83.98 ±0.33	52.67
DataDream (ECCV2024)	61.58 ±1.29	39.87 ±1.47	65.11 ±0.40	43.59 ±0.63	81.46 ±2.48	85.18 ±0.65	62.80
ChimeraLoRA (CVPR2026)	63.85 ±1.41	40.39 ±1.19	59.03 ±1.04	44.49 ±0.29	83.64 ±0.27	86.28 ±0.11	62.95
LoFT (BMVC2025)	61.91 ±0.78	39.46 ±0.60	59.15 ±1.73	49.39 ±0.41	**84.16** ±0.66	**88.40** ±0.40	63.75
FM control	65.59 ±0.87	40.88 ±0.71	64.48 ±0.39	45.76 ±0.55	83.35 ±0.55	84.82 ±0.35	64.15
TASKREL	**66.48** ±0.47	**45.76** ±0.76	**69.32** ±1.26	**53.16** ±0.80	83.61 ±0.29	86.68 ±0.49	**67.50**

Table 2: Class-comparison objective and competitor allocation. FM control uses positive fitting alone. Dynamic uniform, frozen soft and TaskRel use the same class-relative hinge, three competing conditions and coefficient c, while changing how competitors are selected. Independent target uses a different comparison objective. Entries are scratch-ResNet-50 test accuracy (%), mean ± SD over three seeds with shared supports and training duration. Avg. is the four-domain mean, Rank is the mean within-domain rank, and ΔFM is the mean difference from FM in percentage points.

Adaptation rule	Avg.↑	Rank ↓	ΔFM ↑	Pets	DTD	EuroSAT	Aircraft
Independent target	52.84	4.75	-1.34	64.59 ± 0.44	38.04 ± 1.78	62.41 ± 4.69	46.30 ± 0.41
FM control	54.18	3.75	+0.00	65.59 ± 0.87	40.88 ± 0.71	64.48 ± 0.39	45.76 ± 0.55
Dynamic uniform,c	57.48	2.75	+3.30	64.60 ± 2.33	42.75 ± 1.26	71.00 ± 0.92	51.58 ± 0.29
Frozen soft,c	58.27	2.25	+4.09	65.45 ± 1.04	42.18 ± 1.21	**73.29** ± 0.75	52.18 ± 0.94
TASKREL,c	**58.68**	**1.50**	**+4.50**	**66.48** ± 0.47	**45.76** ± 0.76	69.32 ± 1.26	**53.16** ± 0.80

Across all four datasets, TaskRel reaches 58.68%, a 4.50-point gain over FM, with the best mean rank of 1.50. It exceeds dynamic uniform by 1.20 points, frozen soft by 0.41 points and independent target by 5.85 points in the four-domain average. On Aircraft, the dynamic-uniform and frozen-soft controls reach 51.58 ± 0.29 and 52.18 ± 0.94, compared with 53.16 ± 0.80 for TaskRel. TaskRel leads on Pets, DTD and Aircraft, while frozen soft leads on EuroSAT at 73.29% versus 69.32%. The common objective improves the average over FM under all three allocation rules, and the sampler determines how these gains are distributed. The independent-target comparison tests both target construction and loss shape, with details in Sec. E.1.

The controlled variants separate two findings. Uniform negative conditions already raise the four-domain mean by 3.30 points over FM, so an adaptive relation map is not required to obtain a benefit from class comparisons. Retaining the warm-up map raises that mean further, and updating observed relations gives the highest aggregate and lowest mean rank. On DTD, TaskRel exceeds dynamic uniform and frozen soft by 3.01 and 3.59 points. On Aircraft the corresponding gains are 1.58 and 0.98 points. The EuroSAT ordering reverses, which shows that the evolving sampler redistributes useful comparisons differently across tasks.

Transfer to different learners. Across six datasets, both panels in Tab. 3 retain each method's fixed synthetic pool and the same supports, with LoFT as the strongest external scratch-ResNet comparator. ViT-Tiny/16 starts from random weights, while pretrained CLIP ViT-B/16 adapts its visual and text encoders using rank-16 LoRA.

To test reuse, we train CLIP ViT-B/16 and scratch ViT-Tiny/16 on the same frozen pools with a common 100-epoch recipe within each learner and seed 0. TaskRel raises the six-domain CLIP mean from 82.52% to 84.57%, with gains on all six domains. The scratch-ViT mean rises from 43.71% to 45.75%. LoFT reaches 80.49% with CLIP and 43.30% with scratch ViT. The repeated CLIP anchor study also has positive TaskRel–FM differences in 9 of 9 paired student runs. The generator was never optimized against either learner, so these results test the portability of its synthetic supervision (Tabs. 3 and 10).

For scratch ViT, TaskRel exceeds LoFT's six-domain mean by 2.45 points and leads on Pets, DTD, EuroSAT and Aircraft. Caltech favors the FM pool and Flowers favors LoFT, so changing the learner

Table 3: Reuse of frozen synthetic pools across learner architectures. Each method's synthetic pool is reused without generator retuning. ViT-Tiny/16 is trained from scratch, while CLIP ViT-B/16 is adapted with LoRA. Within each learner, all methods use the same 100-epoch training recipe and seed 0. Entries are real-test accuracy (%). Avg. gives equal weight to the six datasets, and bold marks the highest value in each column.

Method	Pets	DTD	EuroSAT	Aircraft	Caltech	Flowers	Avg.
ViT-Tiny/16, scratch							
LoFT	24.26	35.05	59.44	15.27	54.89	**70.89**	43.30
FM control	27.01	34.63	62.85	12.87	**58.82**	66.06	43.71
TaskRel	**28.40**	**35.99**	**68.36**	**16.83**	57.40	67.52	**45.75**
CLIP ViT-B/16, LoRA fine-tuning							
LoFT	89.07	63.06	81.19	57.52	95.46	96.67	80.49
FM control	90.81	65.19	90.60	56.95	96.43	95.13	82.52
TaskRel	**91.66**	**68.56**	**91.62**	**62.38**	**96.47**	**96.75**	**84.57**

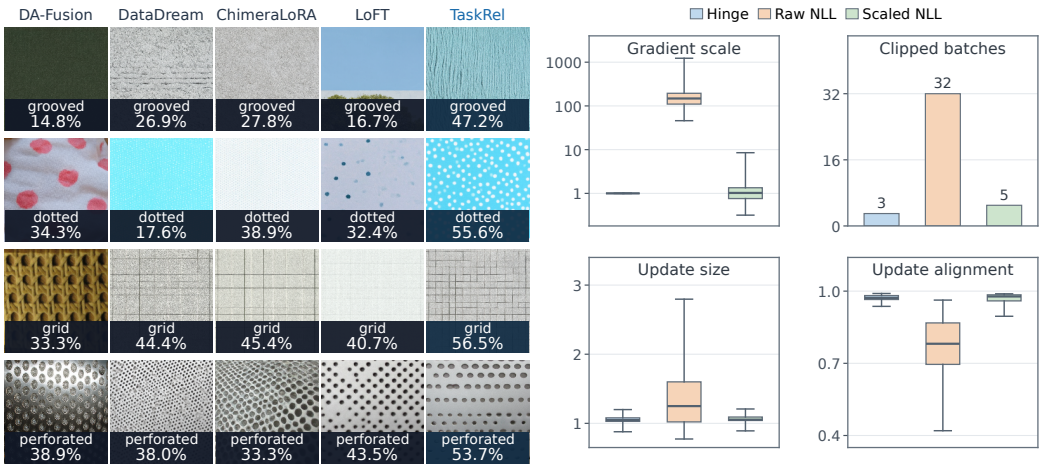
Table 4: Synthetic-data budget across all six domains. Entries are terminal scratch-ResNet-50 test accuracy (%) at seed 0. Every setting retains 16 real supports per class and trains for 100 epochs. Methods are ordered by six-domain average within each synthetic-data budget. Bold marks the highest value in each column.

Method	Pets	DTD	EuroSAT	Aircraft	Caltech	Flowers	Avg.
25 synthetic images/class							
LoFT	35.19	25.65	54.73	36.15	71.12	**79.50**	50.39
FM control	36.28	27.96	62.42	38.13	70.43	78.68	52.32
TaskRel	**41.26**	**29.26**	**63.40**	**41.40**	**73.79**	79.29	**54.73**
50 synthetic images/class							
LoFT	53.18	33.92	55.99	45.00	80.28	**85.51**	58.98
FM control	52.09	38.12	**63.27**	43.35	**80.32**	83.60	60.13
TaskRel	**59.42**	**42.08**	59.64	**49.68**	80.20	84.00	**62.51**
100 synthetic images/class							
LoFT	62.69	39.72	58.16	49.44	**84.50**	**88.59**	63.85
FM control	65.44	41.55	64.93	46.08	83.85	84.98	64.47
TaskRel	**67.02**	**45.21**	**68.27**	**53.56**	83.37	86.76	**67.37**

changes some domain-level rankings. The gain with a scratch transformer extends the evidence beyond the convolutional learner in Table 1. With CLIP, TaskRel has the highest seed-0 accuracy on all six datasets and exceeds LoFT's mean by 4.08 points. The gain shows that the same pools also complement a pretrained image–text representation. Since pool identity is fixed within each method, these experiments test reuse of the generated supervision without another generator-adaptation cycle. The single-seed breadth results and the repeated CLIP anchor study are reported separately to preserve their different levels of replication.

Utility at smaller synthetic-data budgets. Tab. 4 tests whether the gain depends on using the full 100-image pool. We retain all 16 real supports and take canonical prefixes of 25, 50 or 100 generated images per class, without image selection or generator retraining. Each scratch ResNet-50 trains for 100 epochs at seed 0. TaskRel has the highest six-domain mean at every budget, exceeding FM by 2.42, 2.38 and 2.89 points and LoFT by 4.35, 3.52 and 3.51 points, respectively. At 25 images, it leads on five of six domains. The benefit therefore persists with one quarter of the synthetic examples. These are fixed-epoch operating points: reducing the pool also reduces optimizer updates and increases the real-data fraction. Three-seed anchor results are reported in Tab. 11.

Class cues and the strength of contrastive updates. Fig. 3(a) illustrates class-defining ridges, dots, grids and openings in DTD. The four selected classes have recall gains of 10.19–19.44 points over their strongest external comparators, measured with three students trained on the full synthetic pools. These posthoc examples suggest useful class cues, without attributing utility to an individual generated image.



(a) Class cues in generated data

(b) Controlling contrastive updates

Figure 3: Class cues in generated data and the effect of contrastive-loss scale. (a) Selected DTD examples for four classes. Percentages report three-student mean test recall for the entire class and do not score the displayed image. (b) Fixed-state loss audit on Pets. The four plots use the same method order. Clockwise from top left, they report auxiliary-gradient norm relative to the hinge, batches invoking gradient clipping, update cosine with FM, and update norm relative to FM. Boxes show quartiles and medians over 32 evaluation batches, with whiskers spanning the observed range. NLL scaling is calibrated on 32 separate batches. Update proposals include clipping and AdamW state and are evaluated without committing an optimizer step. Details are in Secs. E.3 and E.4.

Fig. 3(b) asks how strongly a class-comparison loss should alter generative fitting. At a Pets warm-up checkpoint, we fix the model, optimizer, noisy states, targets and negative labels, then compare unit-weight hinge and residual-NLL losses. Across 32 evaluation batches, raw NLL has a median gradient norm 147.90 times the hinge's. Raw NLL triggers clipping in all 32 batches versus three for the hinge. Its median update norm is 1.249 times the FM-only proposal, with cosine 0.781. Generator-only scaling reduces clipping to five batches and restores the update norm/cosine to 1.053/0.978, close to the hinge's 1.047/0.972. This shows loss scale changes the update with the same comparison task, supporting the separation of comparison selection and strength in Sec. 3.4. This coefficient differs from the final scheduler-derived c. It neither selects c nor establishes that the hinge is uniquely effective. Protocol and optimizer diagnostics are given in Sec. E.3.

Breadth of classwise gains. Mean accuracy can conceal whether an improvement spans many classes. We examine all 100 Aircraft variants using the same three student seeds and fixed pools (Fig. 4). TaskRel improves mean recall in 62 classes over LoFT, 75 over ChimeraLoRA, 82 over DataDream and 85 over DA-Fusion. It improves 76 classes over the matched FM control. Against LoFT, the strongest external comparator on Aircraft, its paired accuracy gain is $+3.760 \pm 0.437$ points, with sample SD from three within-seed differences. Every class and comparator is retained, including ties and decreases. The aggregate benefit extends across many aircraft variants. Classwise counts describe observed differences without per-class significance testing.

5 Conclusion

TaskRel adapts a flow-matching generator by comparing class-conditioned prediction errors on shared noisy states and targets. A finite-margin objective, adaptive competitor sampling and scheduler-derived scaling turn these gaps into supervision for shared LoRA weights. Across six domains, one fixed configuration improves the matched flow-matching control and produces synthetic pools that transfer across scratch and pretrained learners. Controlled variants isolate the roles of class comparison and competitor allocation. These results show that class relations provide a generator-internal signal for reusable few-shot training data without downstream-classifier feedback or inference-time filtering.

REPRODUCIBILITY STATEMENT

The adaptation equations and complete training order are specified in Sec. 3 and Alg. 1. Proofs and the exact scheduler calculation are given in Secs. A and B. Data identities, port settings, student recipes, and seed-level interpretation are specified in Secs. C and D. The experimental record binds supports, checkpoints, generated image indices, and final student metrics to fixed configurations. The dataset partitions are those specified in Sec. 4.1.

ETHICS STATEMENT

This study uses pretrained generative models to create labeled images for benchmark classification. Synthetic images may reproduce biases or errors in the pretrained model and may be incorrectly labeled despite a valid conditioning prompt. The measured benefits do not establish suitability for medical, legal, or other high-stakes use. Dataset and model terms, image provenance, and restrictions on redistributing generated or original material must be respected in any redistribution. The experiments use existing benchmark labels and automated generation, without collecting new participant data or commissioning human annotations.

AI USE STATEMENT

Generative AI tools assisted with methodological refinement, experimental planning, code debugging, analysis of research records, and manuscript editing. The research motivation, foundational premises and core methodological ideas were conceived by human researchers. All experimental data were derived from executed, human-reviewed code. The authors created the technical workflow figures and bear full responsibility for the final content and scientific claims.

REFERENCES

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves ImageNet classification. *Transactions on Machine Learning Research*, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *MLSys*, 2021.

Byungkwan Chae, Youngjae Choi, and Heewon Kim. MBTI: Metric-based textual inversion for fine-grained image generation. In *WACV*, 2026.

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

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *ICLR*, 2024.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

Nicola Dall'Asen, Victor G. Turrisi da Costa, Yiming Wang, Nicu Sebe, and Elisa Ricci. Diversified in-domain synthesis with efficient fine-tuning for few-shot classification. In *ICIAP*, 2025.

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

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023.

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

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Zong-Wei Hong, Jing-lun Li, Lin-Ze Li, Shen Zhang, and Yao Tang. VeCoR — velocity contrastive regularization for flow matching. In *Findings of CVPR*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Hoyoung Kim, Minwoo Jang, Jabin Koo, Sangdoo Yun, and Jungseul Ok. ChimeraLoRA: Multi-head LoRA-guided synthetic datasets. In *CVPR*, 2026.

Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. DataDream: Few-shot guided dataset generation. In *ECCV*, 2024.

Jae Myung Kim, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. LoFT: LoRA-fused training dataset generation with few-shot guidance. In *BMVC*, 2025.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Soroush Abbasi Koohpayegani, Anuj Singh, K. L. Navaneet, Hamed Pirsiavash, and Hadi Jamali-Rad. GeNIe: Generative hard negative images through diffusion. *Transactions on Machine Learning Research*, 2025.

Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *ICCV*, 2023.

Fei-Fei Li, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Weijian Ma, Ruoxin Chen, Keyue Zhang, Shuang Wu, and Shouhong Ding. Instruct where the model fails: Generative data augmentation via guided self-contrastive fine-tuning. In *AAAI*, 2025.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022.

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

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

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

Stability AI. Stable diffusion 3.5 medium. Hugging Face model card, 2024.

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching. In *ICCV*, 2025.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *ICLR*, 2016.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *ECCV*, 2020.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. StableRep: Synthetic images from text-to-image models make strong visual representation learners. In *NeurIPS*, 2023.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *ICLR*, 2024.

Yanghao Wang and Long Chen. Inversion circle interpolation: Diffusion-based image augmentation for data-scarce classification. In *CVPR*, 2025.

Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

Ruitao Wu, Yifan Zhao, Guangyao Chen, and Jia Li. Diffusion-classifier synergy: Reward-aligned learning via mutual boosting loop for FSCIL. In *NeurIPS*, 2025.

William Yang, Xindi Wu, Zhiwei Deng, Esin Tureci, and Olga Russakovsky. Beyond objects: Contextual synthetic data generation for fine-grained classification. In *CVPR*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

APPENDIX

A TRAINING ALGORITHM AND IMPLEMENTATION

A.1 FROM LABELED SUPPORTS TO A REUSABLE GENERATOR

Alg. 1 gives the training procedure. Alg. 2 specifies its one-time calibration. The three stages are positive-only warm-up, generator-only probing, and class-contrastive continuation. The probe holds model parameters fixed. Continuation retains the warm-up weights, optimizer moments, learning-rate state, and training-data position. Only the adapted generator is needed for synthesis.

Here θ denotes the trainable transformer and CLIP LoRA parameters. All pretrained weights remain fixed. The core task has $C \geq 4$ classes, with $n = 16$ supports and one prompt per class. It uses $B = 8$, $K_n = 3$, $K_g = 8$, EMA decay $\beta = 0.99$, and $T = 100C$ optimizer updates. The update index s counts completed updates before the current one: $s = 0, \ldots, T_w - 1$ is warm-up and $s = T_w, \ldots, T - 1$ is continuation. Warm-up and continuation share clipped AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019), constant learning rates and optimizer state. Write sg for stop-gradient and $\mathrm{Backprop}$ for accumulation into the current parameter gradients, without an optimizer step.

Algorithm 1 TASKREL: class-contrastive LoRA adaptation

Require: Pretrained generator and fixed VAE, supports S, prompts $\{p_y\}$, horizon T
Initialize shared LoRA parameters θ and AdamW, $T_w \leftarrow \lfloor T/4 \rfloor$
Compute $c = \mathbb{E}[\tilde{\sigma}^2]$ from the training scheduler (Sec. A.2)
Perform T_w positive-only updates using $B^{-1} \sum_i \ell_{+,i}$
$(M^0, m, \tau_g) \leftarrow \textsc{Probe}(\theta, S, \{p_y\})$ ▷ Alg. 2
$M \leftarrow M^0$, initialize the private negative-sampling RNG
for $s = T_w, \ldots, T - 1$ **do**
    Clear parameter gradients, take the next support minibatch
    Sample each anchor's latent z_i, noise ϵ_i, and training (t_i, σ_i)
    Form z_{σ_i} and u_i, evaluate native $\ell_{+,i}$ (Eqs. (1) and (2))
    $\mathrm{Backprop}\big(B^{-1} \sum_i \ell_{+,i}\big)$
    for $i = 1, \ldots, B$ in minibatch order **do**
        Set $q_j \propto \exp(-M_{y_i j}/\tau_g)$ for every $j \neq y_i$
        Draw K_n distinct labels N_i with weights q, sort their indices
        Re-evaluate $\{y_i\} \cup N_i$ on the saved (z_{σ_i}, t_i, u_i)
        Compute energies $e_{iy_i}, \{e_{ij}\}_{j \in N_i}$ and R_i (Eq. (3))
        for $j \in N_i$ **do**
            $M_{y_i j} \leftarrow \beta M_{y_i j} + (1 - \beta)\, \mathrm{sg}[(e_{ij} - e_{iy_i})/e_{iy_i}]$
        end for
        $\mathrm{Backprop}(cR_i/B)$ ▷ Both prediction branches receive gradients
    end for
    Clip the accumulated LoRA gradient to norm 1, take one AdamW step
end for
return Final transformer/CLIP adapters, discard M for ordinary synthesis

What is differentiated. The true-condition prediction is recomputed in each auxiliary forward, with current CLIP adapters and the same saved corruption state. Neither it nor the wrong-condition predictions is detached. Only the relative gaps used for EMA bookkeeping are detached. Sampled label sets are treated as fixed when taking the gradient: there is no score-function term or gradient through the sampler, and no importance weighting toward a uniform full-class objective. Updating M after an anchor can change the next anchor's draw, but never its already constructed loss. Serial backward passes free each anchor's activations while accumulating gradients for one optimizer step.

Algorithm 2 PROBE: initialize class relations and frozen calibration statistics

Require: Warm-up parameters θ, equal-sized support sets S_y, class prompts $\{p_y\}$

Save module modes, use evaluation mode and disable autograd
Encode all class prompts at θ, initialize $A \leftarrow 0 \in \mathbb{R}^{C\times C}$, $\mathcal{E} \leftarrow [\,]$
for each class y and support $x \in S_y$ in canonical order **do**
    Center-crop x, draw and scale one VAE posterior latent z using its fixed seed
    for $k = 0, \ldots, K_g - 1$ **do**
        Draw $a \sim \mathcal{N}(0,1)$ and $\epsilon \sim \mathcal{N}(0, I)$ with the state seed
        $q \leftarrow \mathrm{clip}(\mathrm{sigmoid}(a), 0.02, 0.98)$, $b \leftarrow \lfloor 1000q \rfloor$
        Read (t_b, σ_b), form one shared $z_\sigma = (1-\sigma_b)z + \sigma_b\epsilon$, $u = \epsilon - z$
        Evaluate $e_j = d^{-1}\|v_\theta(z_\sigma, t_b, p_j) - u\|^2$ for all C prompts, in chunks of 16
        $A_{yj} \leftarrow A_{yj} + (e_j - e_y)/e_y$ for all j, append e_y to $\mathcal{E}$
    end for
end for
$M^0 \leftarrow A/(nK_g)$, require finite complete probe statistics
$m \leftarrow 0.03\ \mathrm{median}(\mathcal{E})$, $\tau_g \leftarrow \max(\mathrm{median}_{y\neq j}\,|M^0_{yj}|, 10^{-4})$
Restore each module's previous mode, retain all training and optimizer state
return (M^0, m, τ_g)

A.2 EXACT SCHEDULER-DERIVED COEFFICIENT

The native scheduler has 1,000 entries and static shift 3. Training draws $a \sim \mathcal{N}(0,1)$, sets $q = \mathrm{sigmoid}(a)$ and $i = \lfloor 1000q \rfloor$, then reads its timestep and shifted coefficient (t_i, σ_i). The index fraction q and physical noise coefficient σ_i are distinct. The continuous shift maps an unshifted value s to $3s/(1+2s)$. The instantiated scheduler table supplies the exact endpoint and rounding conventions used by the implementation.

Define

$$F_0 = 0, \quad F_{1000} = 1, \quad F_i = \Phi\left(\log \frac{i}{1000 - i}\right) \quad (1 \le i < 1000). \tag{9}$$

Here Φ is the standard normal cumulative distribution function. Under the specified logit-normal sampling law, the bin mass is $F_{i+1} - F_i$. Casting the native sigma values to BF16, then converting them to FP64 before squaring and summing, yields

$$c = \sum_{i=0}^{999} (F_{i+1} - F_i)\big(\mathrm{BF16}(\sigma_i)\big)^2 = 0.54196956370547. \tag{10}$$

The coefficient uses the unclamped training distribution. Graph probes clamp the index fraction to $[0.02, 0.98]$ before scheduler lookup. Using that probe distribution to calculate c would change the method. This analytic expectation uses the training sampling law and contains no classifier result or fitted per-dataset scale.

A.3 ARCHITECTURE, PREPROCESSING, AND OPTIMIZATION

Tab. 5 specifies the adaptation and generation settings shared across datasets, including adapter capacity and the optimization budget. The implementation uses PyTorch (Paszke et al., 2019). The pretrained pipeline combines CLIP text encoders (Radford et al., 2021) with a frozen T5 encoder (Raffel et al., 2020). Only the specified CLIP and transformer adapters are optimized. The VAE posterior sample is shifted and scaled as $z = (z_{\mathrm{VAE}} - 0.0609)1.5305$. At 512 pixels, the latent has shape $16 \times 64 \times 64$. Training samples the posterior rather than using its mean. EXIF-corrected RGB images are resized on the shorter side to 512, randomly cropped to 512×512, and normalized with mean and standard deviation 0.5. Graph probes use a center crop instead. Generator horizontal flipping and prior-preservation images are disabled.

Numerical realization of the losses. Training casts the scaled VAE latent to BF16, draws same-dtype noise, and constructs the noisy latent using BF16 scheduler coefficients. The native positive branch computes its clean-latent prediction before converting the residual calculation to FP32. The

auxiliary branch forms $u = \epsilon - z$ in the latent dtype, then computes prediction residuals and their mean square in FP32. Thus Eq. (2) is an exact-arithmetic identity, not a claim that replacing the native positive branch with a velocity-error expression is bitwise equivalent. During the separate graph pass, latent, noise, scheduler coefficients, and target construction use FP32. Transformer inputs/forward use BF16, residual means use FP32, and graph accumulation uses FP64. The graph supplies a fixed initialization statistic without updating the model.

Table 5: Frozen core training and generation configuration. The same rules apply across datasets. The support-derived m, τ_g are recomputed from each registered pipeline rather than selected by test accuracy.

Component	Setting
Backbone	SD3.5 Medium, fixed VAE and pretrained backbone weights
LoRA	Dataset-shared, rank 16, alpha 16, dropout 0, Gaussian A, zero B
Transformer targets	Added and ordinary attention Q/K/V and matching output projections
Text adapters	CLIP-L and CLIP-G Q/K/V/out, T5 is frozen
Training precision	BF16 frozen weights/forward, FP32 optimized adapters and moments, FP32 VAE
Transformer AdamW	LR 10^{-4}, decay 10^{-4}
Each CLIP AdamW group	LR 5×10^{-6}, decay 10^{-3}
Shared optimizer settings	Betas $(0.9, 0.999)$, epsilon 10^{-8}, global gradient clip 1
Schedule	50 epochs, batch 8, constant LR, first 25% of updates positive-only
Graph	16 supports/class, 8 states/support, all class conditions, chunks of 16
Comparison	Three wrong labels without replacement, mean absolute-gap hinge
Statistics	Relative gaps, observed-pair EMA decay 0.99, frozen temperature
Core token budgets	CLIP length 77, T5 training 77 and generation 256
Generation	100/class, 512 pixels, 24 native Euler steps, CFG 2, batch 2
Core seeds	Generator and synthesis 42, sampler 20260917, graph base 20260901

The eight transformer suffixes are `attn.add_k_proj`, `attn.add_q_proj`, `attn.add_v_proj`, `attn.to_add_out`, `attn.to_k`, `attn.to_q`, `attn.to_v`, and `attn.to_out.0`. No block restriction is applied. CLIP targets are `q_proj`, `k_proj`, `v_proj`, and `out_proj`. The two current CLIP encoders are re-evaluated for sampled prompts during training. Cached warm-up embeddings are not reused after their adapters change. Frozen T5 retains its inference mode when the graph pass restores module states.

The tested execution uses one process, gradient accumulation one and BF16 without a GradScaler. The objective warm-up keeps the learning rate constant. Only LoRA tensors are optimized, although checkpointing can require gradients on non-optimized embedding tensors.

A.4 Sampler, support identity, and replay

The relation matrix is directed and can contain negative entries. The weights in Eq. (6) specify weighted sampling without replacement, rather than set inclusion probabilities. Anchor-by-anchor sampling and EMA updates follow Alg. 1 and are required for exact replay.

Support preprocessing is deterministic for a tuple of run seed, epoch, and support index, with an independent epoch order. The exact support manifest defines which files are used. Consuming the first files in an already prepared view is not a claim that every dataset used the first 16 files of its original training directory. Nested directory labels are retained as full relative class identities.

Graph VAE seeds are $20260901 + 10^6 + g$, and state/noise seeds are $20260901 + 2 \times 10^6 + 100g + k$, where g is the global support index and $0 \leq k < 8$. Row sharding preserves this global indexing and evaluates all conditions for each source state. Margin calibration uses the combined raw positive energies, not the mean of shard medians. Resumption saves model, optimizer, scheduler, CPU/CUDA RNG, data position, sampler RNG, current matrix, and pair-exposure counts.

A.5 Generation and classifier training

Generation uses classifier-free guidance (Ho & Salimans, 2022) at scale 2, with the same sampler settings across the matched comparisons. Generation seeds derive deterministically from generation

seed, class key, image index, and a noise-purpose tag. Original and repeated pipelines use their registered seeds 42 and 43. Exact batch size, prompt, class order, image index, and adapter identity are retained. No image-quality selection is applied. The core uses ordinary class prompts, with the registered satellite template for EuroSAT. The five-shot extension uses a different, explicitly matched context protocol (Sec. D.6).

Table 6: Common ResNet-50 (RN50) evaluation. "Pretrained" uses fully fine-tuned torchvision RN50. The scratch regime uses a custom implementation, as detailed below.

Item	Setting
Data	Shuffled union of $16 + 100$ images/class, no enforced 50/50 balance
Duration / seeds	100 terminal epochs, seeds 0, 1, 2
Optimizer	SGD, momentum 0.9, weight decay 10^{-4}
Learning rate	Peak 0.01, 15-epoch ramp from 0.0001, then cosine decay toward 0.0001
Loss	Cross-entropy with label smoothing 0.1, no MixUp/CutMix
Batch / precision	Training 32, evaluation 64, FP32 deterministic execution
Training image view	224 pixels, RandAugment, random resized crop, flip, color/grayscale and blur/solarization augmentation
Evaluation image view	Resize to 256, center crop 224, ImageNet normalization
Metric	Correct test examples divided by total test examples, at epoch 100

The classifier recipe in Tab. 6 retains RandAugment (Cubuk et al., 2020) and label smoothing (Szegedy et al., 2016). MixUp (Zhang et al., 2018) and CutMix (Yun et al., 2019) are disabled for all compared synthetic pools. The cosine schedule follows the decay form of SGDR (Loshchilov & Hutter, 2017), using a single decay after warm-up without restarts.

The scratch Bottleneck downsamples in its first 1×1 convolution. The pretrained branch uses torchvision's RN50 implementation and local ImageNet1K weights with a newly initialized classifier head. Each regime is fixed across synthesis methods. The pretrained branch is fully fine-tuned, not a frozen-feature linear probe. Its ImageNet100 results are transfer within its pretraining class domain, not unseen-class learning without pretraining exposure.

B Why same-state class comparisons provide supervision

B.1 Conditional regression and paired comparisons

We make the probability space explicit. Let Z be a latent drawn from the class-conditional real distribution, Y its label, and ϵ independent Gaussian noise. Let $H = (Z_\sigma, t)$ and $U = \epsilon - Z$. All conditional expectations below are for a fixed model and a fixed noise sampling distribution, with finite second moments. At a fixed state h, the model output $v_j = v_\theta(h, p_j)$ is deterministic.

Proposition 1 (Conditional risk of a class comparison)**.** *Fix the model and noise-sampling law. Let $H = (Z_\sigma, t)$, $\mu_y(h) = \mathbb{E}[U \mid H = h, Y = y]$, and assume finite conditional second moments. For deterministic predictions $v_j = v_\theta(h, p_j)$, the identity in Eq. (4) holds. If every conditional predictor attains its population squared-error optimum, the expected gap is $d^{-1}\|\mu_j(h) - \mu_y(h)\|^2$, on the common domain of the conditional means.*

Proof of Prop. 1. Write $U = \mu_y + \eta$, where $\mathbb{E}[\eta \mid H = h, Y = y] = 0$. For any candidate condition j,

$$\mathbb{E}[\|v_j - U\|^2 \mid h, y] = \mathbb{E}[\|v_j - \mu_y - \eta\|^2 \mid h, y] \tag{11}$$

$$= \|v_j - \mu_y\|^2 + \mathbb{E}[\|\eta\|^2 \mid h, y]. \tag{12}$$

Subtract the expression for $j = y$ and divide by d. The conditional variance term is identical and cancels. Squared conditional flow matching is minimized by $v_y^*(h) = \mu_y(h)$. A strictly positive state-dependent weight preserves this pointwise minimizer. Substituting $v_j^*(h) = \mu_j(h)$ gives the stated squared distance, wherever both conditional means are defined. This substitution describes unrestricted population optima. A few-shot LoRA can depart from them. □

The same-state sample-level identity is

$$d\Delta_{yj} = \|v_j\|^2 - \|v_y\|^2 - 2u^\top(v_j - v_y). \tag{13}$$

The common $\|u\|^2$ term cancels exactly. Target uncertainty nevertheless remains in the inner product, so pairing does not make a single comparison noise-free. When the energy variances are finite,

$$\mathrm{Var}(e_j - e_y) = \mathrm{Var}(e_j) + \mathrm{Var}(e_y) - 2\,\mathrm{Cov}(e_j, e_y). \tag{14}$$

Relative to independent draws of the two errors with the same marginals, pairing reduces this variance when the covariance is positive. The same statement applies conditionally on a fixed latent anchor when comparing shared versus independently drawn corruptions. Common corruption is also used in diffusion-based classification (Li et al., 2023). The variance identity supplies the established statistical rationale for pairing.

Scope of the normalized graph. The graph averages $(e_j - e_y)/e_y$. An expectation of ratios is generally not a ratio of expectations, and the proposition does not identify the graph entries as squared distances or probabilities. Each normalized gap is invariant to a common positive rescaling of the energies at that state. Division introduces small-denominator sensitivity. The specified method uses the measured denominator without an added epsilon and rejects nonfinite probes. Eight corruption states per support reduce Monte Carlo noise while retaining the same number of independent real supports.

B.2 The two contributions to a class-relative update

Let $r = v_y - u$ and $\delta v = v_j - v_y$. Then

$$e_j - e_y = d^{-1}\big(\|\delta v\|^2 + 2r^\top \delta v\big). \tag{15}$$

The gap can increase through improved positive fitting, a change in the wrong-condition prediction, or both. If $v_y = u$ and $v_j = u + a$, increasing $\|a\|$ increases the gap without improving v_y. This example does not prove that wrong-condition repulsion is harmful, since rejecting an incompatible condition can be useful. It establishes that an internal margin alone cannot measure the quality of generated examples.

For one active pair, write $g_j = \nabla_\theta e_j$. The exact-arithmetic positive-plus-hinge objective gives

$$\nabla_\theta[\sigma^2 e_y + c(m + e_y - e_j)] = (\sigma^2 + c)g_y - cg_j. \tag{16}$$

For multiple negatives, each active pair receives its c/K_n weight. Shared parameters permit these components to interact with the fit of other classes. However, the sign of such an interaction depends on the corresponding Jacobians and optimizer state. A negative gradient cosine alone is not evidence of finite-step class damage. The native optimizer audit evaluates update proposals after clipping and AdamW preconditioning. Recognition experiments measure the resulting training pipelines.

B.3 Energy sensitivity of the hinge and residual-NLL

Let A denote the active negative set. Away from hinge kinks,

$$\frac{\partial R}{\partial e_y} = \frac{|A|}{K_n}, \qquad \frac{\partial R}{\partial e_j} = -\frac{\mathbf{1}_{j\in A}}{K_n}. \tag{17}$$

At a kink, the corresponding subgradient lies between these endpoint values. The implemented ReLU chooses zero for that pair. The bound concerns derivatives with respect to energy coordinates, not parameter gradients, which additionally depend on $\nabla_\theta e_j$. The hinge loss can grow without bound.

For the residual-NLL control, define

$$a_j = -\frac{e_j - e_y}{\tau e_y}, \qquad N = \log\left(1 + \sum_j e^{a_j}\right), \qquad \pi_j = \frac{e^{a_j}}{1 + \sum_l e^{a_l}}. \tag{18}$$

Here $\tau > 0$ is fixed, and the denominator $e_y > 0$ is differentiated in the tested implementation. Consequently,

$$\frac{\partial N}{\partial e_j} = -\frac{\pi_j}{\tau e_y}, \qquad \frac{\partial N}{\partial e_y} = \sum_j \frac{\pi_j e_j}{\tau e_y^2}. \tag{19}$$

Small temperatures or energies can amplify sensitivity. The native calibration experiment tests this quantitatively. It does not imply that all NLL formulations fail. These energy-derived scores carry no established calibration to the class posterior induced by the generator's terminal density. An uncorrected sampled log-softmax is generally biased relative to the full-class NLL.

B.4 Average coefficient versus state-dependent allocation

At fixed parameters, condition on the detached sampler state. For $w = \widetilde{\sigma}^2$ and $c = \mathbb{E}[w]$,

$$\mathbb{E}[wR] - c\mathbb{E}[R] = \mathrm{Cov}(w, R). \tag{20}$$

If differentiation and expectation can be interchanged, the corresponding vector relation is

$$\mathbb{E}[w\nabla_\theta R] - c\mathbb{E}[\nabla_\theta R] = \mathrm{Cov}(w, \nabla_\theta R). \tag{21}$$

This is a fixed-state identity. During training the sampler statistics, parameters, and optimizer moments evolve, so it is not an equality between two complete training trajectories. Even equality of expected raw gradients would not generally imply equality after nonlinear clipping and AdamW preconditioning. The constant coefficient isolates a design axis. Downstream accuracy remains an empirical question, and the positive/auxiliary balance can vary with noise level.

C Evaluation data and comparison protocols

C.1 Benchmarks and support identity

We use Oxford-IIIT Pets (Parkhi et al., 2012), DTD (Cimpoi et al., 2014), RGB EuroSAT (Helber et al., 2019), FGVC-Aircraft (Maji et al., 2013), Caltech101 (Li et al., 2004) and Oxford Flowers102 (Nilsback & Zisserman, 2008). The retained class counts are 37, 47, 10, 100, 100 and 102. Caltech excludes Faces_easy and is reported as Caltech100. Pets, DTD, EuroSAT, Caltech and Flowers use the train/validation/test split files released with DISEF (Dall'Asen et al., 2025). Aircraft uses its official variant-level partitions. In this dataset order, the available train/test partitions contain 2,944/3,669, 2,820/1,692, 13,500/8,100, 3,334/3,333, 4,128/2,465 and 4,093/2,463 images.

A frozen manifest selects 16 real training images per class. Every matched generator and classifier uses the same support content. Validation and test images are excluded from fitting. Each generator supplies 100 images per class, and recognition is evaluated on the full real-test partition. The additional 100-class ImageNet subset (Deng et al., 2009) appears separately in Tab. 8. Class lists, subset identities and support manifests define the comparison. Exact byte/RGB support–test overlap was checked. This check covers exact duplicates, while near-duplicates and exposure in foundation-model pretraining remain outside its scope.

Pets and DTD informed method development, and EuroSAT informed the strength study. The final configuration was fixed before the reported ImageNet100 outcomes. The independent pipelines in Tab. 9 vary support selection, generator training and synthesis together. The five-shot context study in Sec. D.6 uses a separate author support draw and reports one generator/noise realization, seed 42.

C.2 Baseline mechanisms and matched settings

The comparison spans shared and instance-adapter fine-tuning, textual inversion and an internal positive-only FM control. All main pools use SD3.5 Medium, 512×512 output, CFG 2 and a nominal 24-step schedule. Algorithm-specific optimization and conditioning are retained as specified in Tab. 7. These are SD3.5 ports with shared supports and downstream evaluation, rather than exact reproductions of the original backbones and training budgets. The endpoint is downstream real-test accuracy, consistent with the intended use of generated training data (Theis et al., 2016).

LoFT uses center crops, cosine LR 10^{-3}, 10 warm-up updates and frozen text encoders. Its original 600-update recipe lies outside this budget. ChimeraLoRA uses dual LR $10^{-4}/10^{-3}$, cosine decay, 100 warm-up updates and frozen text encoders. Its 50-epoch port changes the original 400-epoch budget and detector backend while retaining full semantic boosting. DA-Fusion uses 200 batch-4 updates per class at LR 0.002 with native velocity loss. Image-to-image strength 0.5 gives 12 effective steps from the nominal 24-step schedule.

Table 7: Defining baseline mechanisms and SD3.5 port scope. The common support and student protocols are fixed. Method-specific optimization details follow below and in the referenced sections.

Method	Mechanism and comparison recipe
FM / TASKREL	Shared rank-16 transformer and two-CLIP adapters, 50 epochs. TaskRel adds a class-relation graph and contrastive continuation (Tab. 5).
DataDream (2024)	Dataset-wise rank-16 adaptation, approximately 50 epochs. Checkpoint, prompt and optimization details appear in Sec. C.3.
LoFT (2025)	Independent rank-2 image adapters, 50 updates per image. Synthesis averages two complete adapter updates with equal weights.
ChimeraLoRA (2026)	Rank-4 shared-A/instance-B adapters with full semantic boosting, 50 epochs. GroundingDINO Base supplies boxes.
DA-Fusion (2024)	Class-specific textual inversion in both CLIPs and T5, followed by image-to-image synthesis. Generator weights remain frozen (Sec. D.5).
Diff-II (2025)	Joint concept/LoRA learning and contextual interpolation. No certified SD3.5 numerical result is included.
MBTI (2026)	Token adaptation with inter-class comparisons. Reproduction is deferred until official source code is verified.
Independent target	Loss-only control from Stoica et al. (2025), retaining the native warm-up and class condition (Sec. E.1).
BOB (2026)	Five-shot author supports and paired background/pose prompts, 1,200 matched updates (Sec. D.6).

The 50-update LoFT image adapters and 50-epoch batch-8 shared adapters receive the same total positive support-image exposure, with different parameter banks and optimizer-update counts. LoFT generates one image per adapter pair per call, while FM and TaskRel use batch size 2. All three use the common per-image seed rule, 24 steps, CFG 2, 512 pixels and generation T5 length 256. These settings match the data/evaluation budget without equating compute.

BOB uses paired author background/pose resources and therefore has its own matched five-shot comparison in Sec. D.6. These captions add pretrained-model-derived information from the supports. Their historical extraction cost is unavailable. Diff-II's inversion-circle interpolation and contextual conditioning require a dedicated SD3.5 port outside the completed comparison. MBTI is deferred until official source code is verified. Neither has a numerical entry. TaskRel compares velocity fitting errors under different class conditions at one shared target. Independent-target contrast tests a different objective family (Stoica et al., 2025), as specified in Sec. E.1.

C.3 DATADREAM CHECKPOINT AND PROMPT PROTOCOL

DataDream denotes the dataset-wise variant, named DataDream-dset in its implementation. The comparison uses approximately 50 support epochs, rank 16, batch 8, constant transformer LR 10^{-4}, CLIP LR 5×10^{-6}, random crops and frozen T5. Compatible supplied checkpoints were selected nearest to one quarter of their logged 200-epoch run before recognition accuracy was examined, with ties assigned to the earlier checkpoint. Pets uses update 3500 of 14800, corresponding to 47.30 support epochs and 3497 applied AdamW updates after three FP16 GradScaler skips. DTD uses update 4500, or 47.87 epochs. Caltech uses update 10000, or 50.00 epochs. All three checkpoints passed support, parameter and optimizer-state checks. These supplied runs use FP16 forward with FP32 learned weights and moments. Their training seeds are retained from the original runs. EuroSAT, Aircraft and Flowers use completed fresh 50-epoch adaptation on the frozen supports, with BF16 forward, FP32 learned weights/moments and seed 42.

Generation follows each checkpoint's training prompts. The supplied DTD texture template and Caltech normalized/title-cased names are retained. This is an explicit DataDream recipe difference from the matched FM/TaskRel class prompts. Images were generated afresh at 24 steps, CFG 2, 512 pixels and 100 images per class using deterministic per-image seeds. Pets uses batch 2. The other pools use batch 4 after a full-model numerical and memory check, with per-image VAE decoding.

Every admitted pool passed complete-image and terminal-student verification. Supplied image pools were excluded.

C.4 Prompt consistency

LoFT and ChimeraLoRA use exactly the FM/TaskRel class prompts in adaptation and synthesis, including capitalization and class-name mapping. EuroSAT uses "a satellite photo of {class}." Flowers retains lowercase class names. The supplementary FM/TaskRel ImageNet100 study maps WordNet folders to human-readable labels. Earlier EuroSAT runs with generic-photo prompts are excluded from matched tables and aggregates. The completed corrected cohorts supply the reported cells. Their replacement establishes protocol consistency, without attributing any accuracy change solely to prompts, since the inference implementation was also updated. Original diagnostic records remain archived.

C.5 Additional learners and image budgets

The six-domain transfer extension fixes student seed 0 before new outcomes. CLIP (Radford et al., 2021) and scratch ViT (Dosovitskiy et al., 2021; Touvron et al., 2021) use the same frozen pools as the main scratch-ResNet comparison. Scratch ViT uses the transformer architecture (Dosovitskiy et al., 2021) at the Tiny size (Touvron et al., 2021), implemented by timm's `vit_tiny_patch16_224` with `pretrained=False`: 12 blocks, 192-dimensional embeddings, 3 attention heads, 16-pixel patches and approximately 5.54M parameters for 100 classes. All parameters are trained with AdamW, LR 5×10^{-4}, weight decay 0.05, batch 64, 10-epoch linear warm-up, cosine decay toward 10^{-6}, 100 epochs and FP32. The common image augmentations and label smoothing 0.1 are retained. No distillation, MixUp/CutMix or pretrained initialization is used. CLIP ViT-B/16 uses rank-16 LoRA in the attention layers of its image and text encoders, with alpha 32 and dropout 0.1. All other CLIP parameters remain frozen. It uses the 100-epoch SGD schedule in Tab. 6, with the same augmentations and CLIP-specific channel normalization. This recipe is shared by FM, LoFT and TaskRel on all six datasets. All LoFT transfer evaluations use the same certified pools as the main scratch-ResNet comparison. The completed anchor seed 0 runs are reused only after verifying exact support and synthetic-pool identity.

The budget extension retains all 16 supports and the canonical first 25 or 50 images of each frozen 100-image class pool. At 100 fixed epochs, smaller pools imply fewer optimizer updates and a larger real-data fraction.

D Supplementary recognition results

Unless stated otherwise, entries report terminal real-test micro accuracy and sample SD over student seeds 0, 1 and 2. Each three-seed group shares one support set, generator and synthetic pool. The SD uses denominator $n - 1$ with $n = 3$ and measures conditional student variation. Pipeline realizations are reported separately (Bouthillier et al., 2021). Paired effects use within-seed differences before computing their mean and SD. Dataset averages give equal weight to the stated domains. Averages and paired differences are calculated from unrounded accuracies. Accuracy gains are absolute differences in percentage points. Per-dataset SDs are never averaged to construct aggregate uncertainty. Bold identifies observed maxima within a stated comparison block and carries no significance-test claim. Appendix tables are monochrome.

D.1 Additional learner and domain coverage

Tab. 8 complements the six-domain scratch results in Tab. 1 with fully fine-tuned pretrained RN50 on every completed domain and both learner regimes on ImageNet100. The ImageNet100 paired TaskRel–FM gains are +1.547 $\pm$ 0.666 points for scratch RN50 and +0.720 $\pm$ 0.586 for pretrained RN50. The latter includes ImageNet pretraining exposure, as specified in Sec. A.

Table 8: Additional matched FM and TaskRel evaluations. Pretrained RN50 is fully fine-tuned. IN100 also includes scratch RN50. Entries are mean ± sample SD over three students on each fixed pool. The six-domain scratch comparison remains in Tab. 1.

Dataset	RN50 initialization	FM	TaskRel
Pets	Pretrained	88.217 ± 0.103	**89.579** ± 0.247
DTD	Pretrained	58.708 ± 0.549	**62.076** ± 0.292
EuroSAT	Pretrained	86.000 ± 0.504	**87.947** ± 0.236
Aircraft	Pretrained	55.196 ± 0.613	**61.166** ± 0.355
Caltech100	Pretrained	**95.132** ± 0.247	94.699 ± 0.183
Flowers102	Pretrained	93.842 ± 0.407	**95.602** ± 0.299
IN100	Scratch	39.960 ± 0.314	**41.507** ± 0.352
IN100	Pretrained	71.007 ± 0.412	**71.727** ± 0.439

Table 9: A new support/generator/generation realization on three datasets. All four arms share the new pipeline within each dataset. Each method is evaluated without changing its configuration between realizations.

Dataset	Method	Scratch RN50	Pretrained RN50
Pets	FM	65.776 ± 1.214	87.853 ± 0.175
	TaskRel	66.140 ± 1.164	89.225 ± 0.238
	EMA, σ^2	**67.257** ± 0.582	89.270 ± 0.491
	EMA, coefficient 1	65.440 ± 1.022	**89.370** ± 0.047
DTD	FM	41.745 ± 2.212	59.555 ± 0.356
	TaskRel	**44.582** ± 0.402	61.485 ± 0.595
	EMA, σ^2	44.484 ± 1.866	62.727 ± 0.090
	EMA, coefficient 1	43.322 ± 0.639	**62.786** ± 0.839
EuroSAT	FM	61.930 ± 2.960	85.280 ± 0.879
	TaskRel	68.638 ± 0.215	87.296 ± 0.597
	EMA, σ^2	67.967 ± 1.289	86.988 ± 0.398
	EMA, coefficient 1	**72.691** ± 1.665	**88.399** ± 0.296

D.2 Independent support and generation pipelines

The repeated pipeline in Tab. 9 changes support selection, generator training and synthesis together. Every arm retains its configuration and shares the new supports within each dataset. For dataset d, pipeline p and student seed s, define $\Delta_{dps} = A^{\text{TaskRel}}_{dps} - A^{b}_{dps}$ for comparator b. Its mean and sample SD use the three paired student differences within that pipeline. The two pipeline realizations are reported separately.

Across Pets, DTD and EuroSAT, respectively, the repeated-pipeline TaskRel–FM effects are +0.363 ± 1.288, +2.837 ± 2.033 and +6.708 ± 3.001 points for scratch RN50, and +1.372 ± 0.395, +1.931 ± 0.828 and +2.016 ± 1.042 for pretrained RN50. Against EMA with coefficient 1, they are +0.700 ± 1.722, +1.261 ± 0.536 and -4.053 ± 1.852 for scratch RN50, and -0.145 ± 0.253, -1.300 ± 0.313 and -1.103 ± 0.471 for pretrained RN50. Both pipeline realizations retain positive mean gains over FM on all three anchors and both learners. The coefficient-one comparisons show that rankings depend on the pipeline and learner.

D.3 Reuse with a CLIP learner

In Tab. 10, the same frozen pools improve CLIP ViT-B/16 accuracy on Pets, DTD and EuroSAT by 0.75, 1.99 and 1.21 points, respectively. The equal-domain mean gain is 1.32 points, and all nine

Table 10: Transfer of a fixed generated pool to CLIP ViT-B/16. Three-seed final accuracy under one fixed image/text-LoRA student recipe. No generator retuning. All six groups are complete. The paired-gain column reports mean ± SD of differences between matching student seeds.

Dataset	FM	TaskRel	Paired gain (pp)
Pets	90.71 ± 0.11	**91.47** ± 0.40	0.754 ± 0.443
DTD	65.90 ± 0.71	**67.89** ± 0.59	1.990 ± 1.217
EuroSAT	90.46 ± 0.42	**91.66** ± 0.10	1.206 ± 0.346

Table 11: Synthetic-data budget curve on the same pools. Entries are mean ± sample SD over three student seeds. S/P denotes scratch/pretrained RN50. Budgets 25/50 use the registered prefixes of the 100-image pools. All learners train 100 epochs, so smaller pools imply fewer updates and a larger real-data fraction: $16/41$ at 25 synthetic images versus $16/116$ at 100. The curve includes these changes in exposure and mixture. Every cell uses all three seeds.

Dataset	Method	Learner	25/class	50/class	100/class
Pets	FM control	S	37.10 ± 1.16	52.80 ± 1.27	65.59 ± 0.87
		P	89.45 ± 0.30	88.89 ± 0.38	88.22 ± 0.10
	TaskRel	S	41.56 ± 0.55	56.85 ± 2.38	66.48 ± 0.47
		P	89.80 ± 0.18	89.87 ± 0.55	89.58 ± 0.25
DTD	FM control	S	30.28 ± 2.28	36.51 ± 1.66	40.88 ± 0.71
		P	60.34 ± 0.36	59.81 ± 0.56	58.71 ± 0.55
	TaskRel	S	30.46 ± 1.47	40.96 ± 1.30	45.76 ± 0.76
		P	62.75 ± 0.29	62.16 ± 0.71	62.08 ± 0.29
EuroSAT	FM control	S	61.91 ± 3.20	61.85 ± 2.70	64.48 ± 0.39
		P	85.82 ± 0.58	85.56 ± 0.40	86.00 ± 0.50
	TaskRel	S	60.29 ± 5.80	62.25 ± 2.71	69.32 ± 1.26
		P	87.39 ± 0.30	87.59 ± 0.69	87.95 ± 0.24

paired student differences are positive. This three-seed study complements the six-domain seed-0 comparison in the main text. Neither study retunes the generator for CLIP.

D.4 Synthetic-data budget

In the three-seed study in Tab. 11, TaskRel's paired scratch-RN50 gains over FM at 50 synthetic images per class are +4.052 ± 2.850 points on Pets and +4.452 ± 0.428 on DTD, positive for all three seeds. At 25 images, the corresponding gains are +4.452 ± 1.478 on Pets and +0.177 ± 0.973 on DTD. EuroSAT's 25-image scratch difference is -1.621 ± 8.602, while its pretrained difference is +1.572 ± 0.309. These fixed-epoch operating points include the changes in update count and real/synthetic fraction described in Sec. C.5. Tab. 4 provides the complete six-domain seed-0 comparison, including LoFT, at all three budgets.

D.5 Real-only optimization controls

The main real-only reference uses 16 supports per class for 100 ordinary epochs, with shuffling and no replacement sampling. Augmentation uses the same supports plus 100 synthetic images per class for 100 epochs. Each real image is presented 100 times in both settings. The augmented loader has more optimizer updates: 13,500 versus 1,900 on Pets and 17,100 versus 2,400 on DTD. Equal epochs thus preserve real-image exposure while allowing the classifier to process the additional synthetic data.

DA-Fusion uses the same optimizer, model, loss, support set, pool size and student schedule as every other synthesis row. Its original classifier uses synthetic replacement probability 0.5. Our common pooled protocol uses synthetic fraction $100/116$. This standardizes evaluation across generators while preserving their adaptation and synthesis mechanisms. The update-matched real-only controls in Tab. 12 resample the same supports to match mixed-loader length. They quantify the effect of

additional classifier optimization without adding labeled images. TaskRel exceeds both update-matched controls with a scratch learner on Pets and DTD, so these additional real-only updates leave a remaining accuracy gap.

Table 12: Real-data learning under ordinary and update-matched schedules. Update-matched references sample the same real supports with replacement to match the corresponding mixed-data loader length. They increase optimization on the same labeled images. TaskRel uses the same 16 supports plus 100 synthetic images/class. Bold is the actual maximum within each dataset, including both real-only exposure controls.

Dataset	Training data / schedule	Scratch RN50	Pretrained RN50
Pets	Real only, 100 epochs	14.146 ± 0.590	**89.697** ± 0.216
	Update-matched 50/class	39.684 ± 1.744	88.689 ± 0.335
	Update-matched 100/class	44.108 ± 2.688	87.744 ± 0.773
	TaskRel, 100/class	**66.485** ± 0.469	89.579 ± 0.247
DTD	Real only, 100 epochs	17.376 ± 1.051	61.367 ± 0.503
	Update-matched 50/class	33.570 ± 0.768	61.958 ± 0.448
	Update-matched 100/class	38.672 ± 1.660	60.757 ± 0.682
	TaskRel, 100/class	**45.764** ± 0.760	**62.076** ± 0.292

D.6 A matched five-shot context study

The study uses exactly 185 author-resource Pets supports (five/class), T5 length 256 during both training and generation, 1,200 total generator updates, and a 300-update common positive warm-up for FM/TaskRel. BOB receives paired author context at training and synthesis, as its mechanism requires. TaskRel is trained with class-only prompts. The context extension adds a sampled global author background/pose row at synthesis. The row's background and pose remain paired. No context sampling innovation is claimed by TaskRel.

Table 13: Five-shot context comparisons at generator/noise seed 42. Entries are mean $\pm$ sample SD over three student seeds on the same author support draw. Rows ascend by their two-learner mean. "Class" and "context" denote generation prompts. BOB/class is an ablation.

Policy	Scratch RN50	Pretrained RN50
BOB (CVPR2026) / class (ablation)	45.262 ± 0.545	82.075 ± 0.882
FM / class	56.955 ± 0.576	84.210 ± 0.622
TaskRel / class	58.944 ± 0.438	84.737 ± 0.897
FM / context	60.016 ± 0.553	**84.773** ± 0.339
BOB (CVPR2026) / context	60.861 ± 0.268	84.165 ± 0.288
TaskRel / context	**61.425** ± 0.820	84.474 ± 0.382

At this fixed generator realization, Tab. 13 shows that TaskRel/context reaches 61.425% with scratch RN50, compared with 60.861% for BOB/context and 60.016% for FM/context. Pretrained RN50 favors FM/context at 84.773%, versus 84.474% for TaskRel/context. These results describe the interaction of conditioning and learner initialization within this support/pool realization.

E Objective controls and classwise utility

E.1 Final-strength controls

All five rules in the main design table share the supports, positive warm-up and terminal scratch-learner schedule. Dynamic uniform, frozen soft and TaskRel use the same class-relative hinge, three alternative conditions per anchor and $c = \mathbb{E}[\widetilde{\sigma}^2]$. They isolate the allocation rule at the final strength. The independent-target control changes both the negative construction and the objective. It therefore tests a neighboring intervention rather than one isolated factor.

Aircraft extends the three development anchors to fine-grained manufactured objects. This follow-up was chosen after its existing FM/TaskRel results were observed. Every registered rule is retained.

TaskRel, FM, dynamic uniform, frozen soft and independent target obtain 53.16%, 45.76%, 51.58%, 52.18% and 46.30%, respectively. The main four-domain averages use complete scratch rows. Tab. 14 reports both scratch and pretrained results on the three anchors, allowing the allocation effects to be compared across learner initializations.

Table 14: Final-strength controls with conditional student uncertainty. All cells are mean ± sample SD over three students on a fixed pool. S/P denotes scratch/pretrained RN50. The three development anchors and both learner regimes are included.

Control	Learner	Pets	DTD	EuroSAT
Independent target	S	64.59 ± 0.44	38.04 ± 1.78	62.41 ± 4.69
	P	88.67 ± 0.53	60.62 ± 0.86	87.33 ± 0.12
FM control	S	65.59 ± 0.87	40.88 ± 0.71	64.48 ± 0.39
	P	88.22 ± 0.10	58.71 ± 0.55	86.00 ± 0.50
Dynamic uniform, c	S	64.60 ± 2.33	42.75 ± 1.26	71.00 ± 0.92
	P	89.18 ± 0.25	63.10 ± 0.76	88.10 ± 0.24
TASKREL	S	66.48 ± 0.47	45.76 ± 0.76	69.32 ± 1.26
	P	89.58 ± 0.25	62.08 ± 0.29	87.95 ± 0.24
Frozen soft, c	S	65.45 ± 1.04	42.18 ± 1.21	73.29 ± 0.75
	P	89.40 ± 0.12	62.14 ± 0.22	88.86 ± 0.53

The independent-target control uses $L_{IT} = B^{-1}\sum_i \sigma_i^2[e_i - 0.05\|v_i - \widetilde{u}_i\|^2/d]$, where $\widetilde{u}_i$ is a non-self minibatch trajectory target that may come from the same class. It retains the native positive warm-up and uses one prediction per image. TaskRel additionally evaluates alternative conditions. This loss-only control follows the objective family of Stoica et al. (2025), without implementing its complete guidance/inference system.

On DTD scratch, TaskRel exceeds dynamic uniform by +3.014 ± 1.858 points and frozen soft by +3.586 ± 0.889. Both differences are positive at all three student seeds. Paired gains over independent target are +1.899 ± 0.885 on Pets, +7.723 ± 1.744 on DTD and +6.909 ± 3.487 on EuroSAT. Their equal-domain mean is 5.51 points. The complete allocation study also identifies dataset and learner dependence, including stronger EuroSAT means for frozen soft at the same final coefficient.

E.2 COEFFICIENT CONTROLS WITH FIXED EMA ALLOCATION

Tab. 15 compares auxiliary coefficients 1, $\widetilde{\sigma}^2$ and $c = \mathbb{E}[\widetilde{\sigma}^2]$ under the same EMA sampling rule. FM supplies the positive-only reference. Every arm uses 50 generator epochs, 100 synthetic images/class and 100 student epochs. The coefficient comparison complements the final-strength allocation study in Tab. 14. EuroSAT favors the unit coefficient, while the constant yields the strongest scratch mean on Pets and DTD. The benchmark uses the same constant on every dataset.

Table 15: Coefficient controls with a fixed EMA allocation rule. Three-student mean ± sample SD. S/P denotes scratch/pretrained RN50. Bold marks the largest mean in each dataset/learner block.

Control	Learner	Pets	DTD	EuroSAT
FM	S	65.595 ± 0.869	40.879 ± 0.712	64.477 ± 0.391
	P	88.217 ± 0.103	58.708 ± 0.549	86.000 ± 0.504
EMA, coefficient 1	S	64.477 ± 0.585	42.691 ± 0.181	**72.206** ± 2.065
	P	89.307 ± 0.110	61.131 ± 0.181	**88.700** ± 0.359
EMA, $\widetilde{\sigma}^2$	S	66.167 ± 1.406	43.105 ± 1.109	70.922 ± 1.974
	P	**89.906** ± 0.582	**62.707** ± 0.473	88.354 ± 0.453
TASKREL	S	**66.485** ± 0.469	**45.764** ± 0.760	69.321 ± 1.258
	P	89.579 ± 0.247	62.076 ± 0.292	87.947 ± 0.236

E.3 Native loss-scale audit

Panel (b) of Fig. 3 visualizes the completed native Pets audit at the common warm-up checkpoint. The hinge and residual-NLL use the same three uniformly sampled wrong conditions per anchor, with model parameters, optimizer moments, noisy inputs, targets and prompt RNG held fixed. Hinge and raw NLL have unit multipliers. A separate 32-batch calibration sets the NLL multiplier to $\lambda = 0.006919095043579208$ before the plotted 32 batches are evaluated. The four square plots report weighted auxiliary-gradient norm divided by the same batch's hinge-gradient norm, batches invoking global clipping, combined-update norm divided by the FM-only update norm, and cosine between those two update proposals. Proposals use the analytic AdamW step, including clipping, weight decay and existing moments. Boxes show the 25th/75th percentiles and median over all 32 evaluation batches. Whiskers span the observed minimum and maximum, with no excluded outliers. The clipping plot reports counts out of 32. All plots use the same method order: hinge, raw NLL, then frozen-scale NLL. Fresh evaluation states reuse the training supports, so they are not a held-out-image test. No optimizer update or downstream feedback occurs. This λ diagnoses loss-scale effects and is unrelated to the scheduler-derived constant c used by final TaskRel. Alignment with FM alone is not a downstream quality metric or a preservation guarantee.

Raw NLL/hinge gradient norm has evaluation median 147.90. Scaling reduces its median to 1.023, but the 90th percentile remains 2.001 and the maximum 8.514. A global match therefore does not uniformly align batches, parameter groups, or directions. The median proposal-norm ratios to FM are 1.047, 1.249 and 1.053 for hinge, raw NLL and scaled NLL, respectively. Their median proposal cosines are 0.972, 0.781 and 0.978, with clipping on 3, 32 and 5 of the 32 evaluation batches.

E.4 Qualitative selection and classwise utility

Panel (a) of Fig. 3 is a posthoc case study. Four DTD classes were selected for large measured recall gains over the strongest of the four external comparators and interpretable structural cues. We inspected indices $\{0, 25, 50, 75\}$ for every generator and chose one common index per class: grooved 25, dotted 75, grid 50 and perforated 0. The common-index rule prevents independent selection of each baseline image. Every frame is shown without enhancement, retouching or image-specific cropping. Candidate sheets and source hashes are retained. No quantitative pool was filtered by this display selection.

Each overlaid percentage is whole-class final test recall, averaged over student seeds 0/1/2 using all synthetic training images and 36 test images per DTD class. It describes the class, rather than the displayed image. These selected outcomes illustrate class cues associated with measured utility. Tab. 1 supplies the complete comparison. The panel contains external methods, while FM remains an internal positive-only control. Shared image indices preserve the display rule without implying identical noise paths across algorithms.

Fig. 4 measures how widely the Aircraft gain is distributed across variants. TaskRel improves mean class recall on 62 classes relative to LoFT, 76 relative to FM and 85 relative to DA-Fusion. The corresponding overall accuracy gains are 3.76, 7.39 and 14.38 points. The remaining classes, including ties, are shown in the same bars. Class counts describe the direction of measured differences, without per-class significance testing. The analysis complements the selected DTD examples with complete classwise downstream outcomes on Aircraft.

F Computational cost and scope

F.1 One-time adaptation and reusable synthesis

TaskRel concentrates its additional computation in generator adaptation. The graph is discarded for ordinary synthesis, which requires the same adapted backbone and sampler as FM, with no additional scorer or filtering stage. Each generated pool is then reused across classifiers and image budgets. This reuse is demonstrated with scratch RN50, pretrained RN50, scratch ViT-Tiny and CLIP, without further generator adaptation.

On an isolated RTX 4090, four native generation profiles on Pets and EuroSAT measure 1.484–1.487 seconds per image for FM and TaskRel at batch 2, 24 steps and CFG 2. Each has a peak allocated

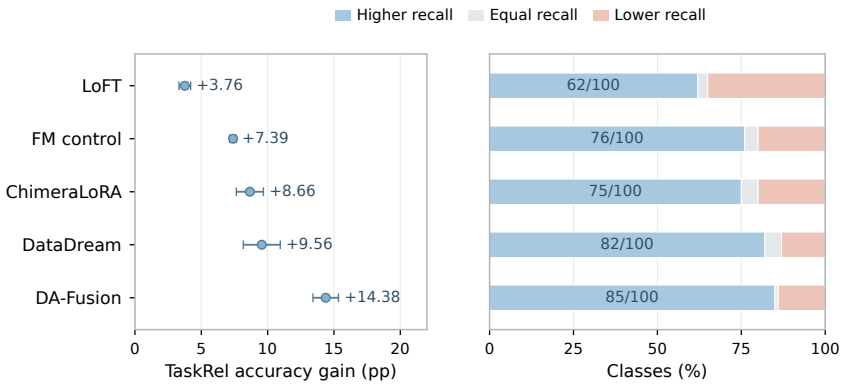


Figure 4: Downstream gains across all 100 Aircraft classes. Left: TaskRel minus each comparator in final scratch-RN50 test accuracy, with mean and sample SD over three paired student seeds on fixed pools. Right: class counts with higher, equal or lower three-seed mean recall for TaskRel. Filled blue segments label the improved-class count. Every synthesis comparator from Tab. 1 is included, ordered by its Aircraft accuracy. No classes or seeds are selected.

memory of 16.46 GiB. These profiles exclude model loading and file writing. The practical trade-off is additional one-time adaptation followed by ordinary-cost generation and repeated use of the resulting training set.

Table 16: Representative adaptation budgets and reusable-pool accuracy. Adaptation includes graph construction and is calibrated from native CUDA-event profiles on an RTX 4090. Accuracy is mean $\pm$ sample SD over three students. Calibrated-compute FM extends the original FM checkpoint to TaskRel's estimated adaptation budget. Rows retain both FM budgets.

Dataset	Operating point	Adaptation (h)	Scratch RN50	Pretrained RN50
Pets	FM, calibrated compute	8.284	60.898 ± 0.931	88.035 ± 0.992
	FM, 50 epochs	1.504	65.595 ± 0.869	88.217 ± 0.103
	TaskRel, 50 epochs	8.284	$\mathbf{66.485} \pm 0.469$	$\mathbf{89.579} \pm 0.247$
EuroSAT	FM, 50 epochs	0.415	64.477 ± 0.391	86.000 ± 0.504
	FM, calibrated compute	1.978	68.786 ± 1.928	87.930 ± 0.616
	TaskRel, 50 epochs	1.978	$\mathbf{69.321} \pm 1.258$	$\mathbf{87.947} \pm 0.236$

Profiles use the median of 12 native update timings after discarding the first two. For FM/TaskRel update times f, r and one graph source-row time g, the budget is $T_w f + (T - T_w)r + Cg$. FM continuation inherits its 50-epoch cost Tf and stops at the first update crossing the remaining allowance. Budgets exclude loading, checkpointing and whole-server overhead. Graph-row extrapolation repeats setup overhead and conservatively enlarges the FM allowance.

In Tab. 16, TaskRel's paired scratch gains over compute-budget FM are $+5.587 \pm 0.891$ points on Pets and $+0.535 \pm 0.989$ on EuroSAT. The latter gap is small relative to conditional student variation. Longer FM training reduces Pets accuracy relative to FM50. Thus the table compares specified operating points, without establishing an optimized compute frontier.

The 16-shot graph evaluates $128C^2$ condition predictions and stores $O(C^2)$ statistics. Each continuation update adds 32 condition sequences to the positive batch of eight. Serial per-anchor backward passes limit live activation memory. The first quarter of updates is positive-only.

F.2 Scope of the evidence

The experiments evaluate one frozen rule on SD3.5 Medium across six main domains, an additional ImageNet100 study and four learner regimes. Three-seed uncertainty describes students on fixed pools. Independent pipeline repetitions cover three anchors, while six-domain ViT/CLIP breadth uses seed 0. The full tables retain dataset-dependent rankings and distinguish equal-update adaptation

from calibrated-compute controls. The squared-error analysis concerns conditional regression. It supplies neither calibrated class probabilities nor a guarantee of downstream accuracy. The five-shot context study changes conditioning and support identity together, so it supplies context-transfer evidence rather than an isolated shot-count ablation. Scaling the quadratic graph and testing additional backbones remain open directions.