TRANSFERABLE STABILITY PREDICTION FOR BUDGETED DIFFUSION CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models classify images by comparing class-conditional denoising losses, but stabilizing predictions with repeated noise draws is expensive. Adaptive sampling prioritizes images with unstable predictions, but stability varies with the backbone and observed draws, making it challenging to transfer an policy without refitting. We introduce TAPS (Transferable Allocation via Predicted Stability), which uses class-permutation-invariant features of relative class competition to predict disagreement with a five-draw reference prediction and allocate a fixed sampling budget. Its predictors, feature normalization, and source-selected schedule remain frozen across tasks and backbones, while allocation decisions adapt to observed losses. Across eight datasets, ImageNet-trained TAPS policies transferred between Stable Diffusion 1.5 and 2.0 improve macro-averaged normalized accuracy–budget AUC over uniform sampling by 1.718 and 1.945 percentage points on SD1.5 and SD2.0 receivers, respectively. They also outperform the evaluated PETS-OKG and Blend-ASC adaptations. These results support relative class competition as a transferable signal for sampling budget allocation.

1 INTRODUCTION

Pretrained diffusion models enable zero-shot classification by comparing class-conditional denoising losses, without training a task-specific classifier (Clark & Jaini, 2023; Li et al., 2023). Averaging these noise-dependent losses over repeated draws stabilizes class scores but makes inference expensive. To use a shared inference budget more efficiently, adaptive sampling can allocate further draws selectively, prioritizing images whose predictions remain unstable. We study whether this allocation rule can be learned on one task and backbone, then reused across tasks and backbones without refitting.

Learning such a rule connects our problem to adaptive inference, where confidence estimates and learned stopping rules assess intermediate predictions and have been studied across tasks and models (Schuster et al., 2022; Dong et al., 2026). Within diffusion classification, existing methods reduce cost by eliminating unlikely labels or selecting informative timesteps (Clark & Jaini, 2023; Li et al., 2023; Yoon et al., 2025). Complementing these within-image strategies, we study complete-policy transfer for allocating repeated draws across images while retaining the candidate set and mean-loss rule.

Such transfer is challenging because stability depends on the receiving backbone and observed draws. For example, different initial draws for the same image and backbone can produce predictions that agree or disagree with a fixed finite-draw reference. A reusable policy must therefore adapt its decisions—not its parameters—to these observations (Figure 1). This motivates our central question: *Can relative denoising-loss patterns provide a transferable signal for assessing stability and allocating draws across tasks and backbones?*

To answer this question, we examine relative class competition underlying the mean-loss scores. These averages can conceal distinct patterns: similar gaps between two leading classes may reflect a consistent preference for one class or rankings that change across timesteps and draws. Because these patterns may distinguish stable predictions from those sensitive to further sampling, we encode them through normalized losses, relative class ranks, and agreement statistics. By encoding relative scores and ranks rather than class identities, these features provide a class-permutation-invariant representation shared across tasks.

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

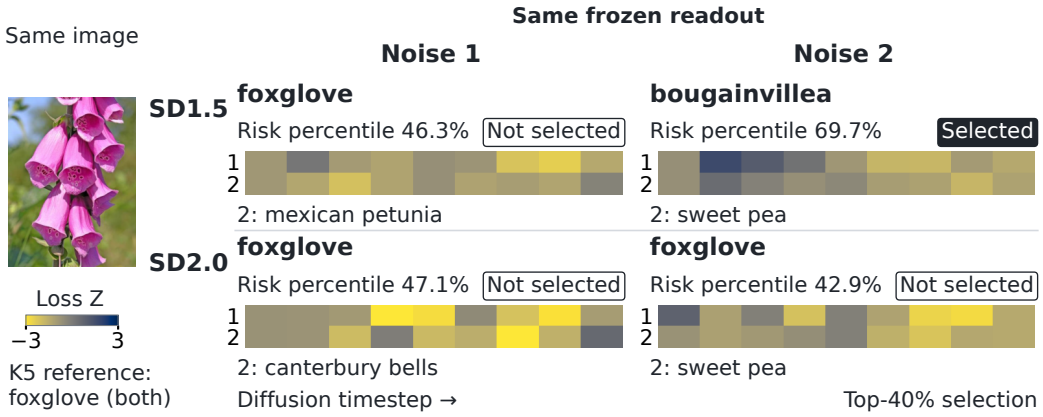

Figure 1: **Different observations, one frozen stability readout.** A frozen SD1.5/ImageNet readout, with its fitted feature normalization, assesses the same image on two backbones under matched initial noise draws. For each observation, heatmap rows show standardized losses for the two classes with the lowest aggregate scores; lower is better. The corresponding bold labels give current predictions, while risk percentiles determine membership in the top-40% continuation group. For comparison, K5 gives each backbone's five-draw reference prediction.

Using this representation, we introduce TAPS (Transferable Allocation via Predicted Stability). Its lightweight stagewise stability predictors, or *readouts*, rank images for continuation, while a source-selected schedule determines how many continue at each stage. These readouts are trained on source images using disagreement between the donor backbone's partial-observation and five-draw predictions, without ground-truth labels. The fitted readouts and feature normalization are then frozen alongside a schedule selected from the same source data. At inference, this frozen policy updates priorities from the receiving backbone's observed losses to allocate a fixed budget, while retaining mean-loss classification. Reusing these losses requires no additional denoiser evaluations for feature extraction.

To evaluate this frozen policy, we transfer ImageNet-trained TAPS between Stable Diffusion 1.5 and 2.0 across eight target datasets. At matched denoiser budgets, these transferred policies improve macro-averaged normalized accuracy–budget AUC over uniform sampling by 1.718 and 1.945 percentage points on the SD1.5 and SD2.0 receivers, respectively. Their macro AUCs also approach those of native policies fitted on receiver-side ImageNet losses and exceed those of the evaluated quadratic controls and adapted allocation baselines. Beyond these aggregate results, controlled comparisons show that frozen readouts track stability changes across observations and backbones, that updating priorities improves allocation, and that local class competition contributes beyond aggregate loss information.

In summary, our contributions are threefold:

- We identify relative class competition in diffusion losses as an empirically transferable signal for sampling allocation across tasks and backbones.
- We introduce TAPS, a complete sampling policy combining class-permutation-invariant features, finite-reference supervision, and a source-selected schedule, without receiver-side fitting or additional denoiser evaluations for feature extraction.
- We demonstrate observation-dependent allocation with frozen readouts and establish the contributions of stagewise priority updates and local class-competition features through controlled comparisons.

2 Related Work

Diffusion classification. Text-to-image diffusion models support zero-shot recognition by comparing class-conditional denoising losses (Clark & Jaini, 2023; Li et al., 2023). GDC instead fits

class distributions in DINOv2 (Oquab et al., 2024) feature space using diffusion-generated images (Qi et al., 2024). For loss-based classification, shared randomness improves comparisons, while statistical elimination reduces candidate evaluations and permits stopping once one candidate remains (Clark & Jaini, 2023). Other efficiency strategies prune label hierarchies (Shanbhag et al., 2024), select discriminative timesteps (Yoon et al., 2025), or compress denoiser tokens (Liu et al., 2026a). Complementing these within-image strategies, TAPS allocates repeated noise draws across images, retaining the candidate set and mean-loss rule. Its features reuse classification losses without an auxiliary semantic encoder or extra denoiser evaluations.

Consistency-based adaptive inference. Prediction consistency can guide adaptive computation. For example, CATs and CALM use agreement with full-model predictions for early exit (Schuster et al., 2021; 2022), while Certaindex uses answer stabilization for early exit and compute allocation (Fu et al., 2025). For stochastic sample allocation, PETS targets agreement with the infinite-budget majority vote (Liu et al., 2026b), while Blend-ASC combines adaptive consistency and mode estimation confidence rankings for joint allocation under a fixed total budget (Feng et al., 2026). To compare these allocation rules, we adapt PETS and Blend-ASC to TAPS's draw cap and mean-loss classifier. Within this setting, TAPS predicts disagreement with a finite mean-loss reference prediction from observed denoising losses.

Budget allocation and transfer. Adaptive inference also studies model selection and early exit (Bolukbasi et al., 2017), with MSDNet addressing budgeted batch classification (Huang et al., 2018). Related language-model methods predict computation-dependent reward (Damani et al., 2025), frame cross-query allocation as a bandit problem (Zuo & Zhu, 2026), or estimate reward and remaining cost during decoding (Manvi et al., 2026). Regarding transfer, Adaptive Thinking predicts the fraction of correct sampled answers from query hidden states using model-specific predictors that transfer across tasks (Li et al., 2026). LearnStop also evaluates transfer across tasks and models, learning stopping rules from observed reasoning prefixes (Dong et al., 2026). TAPS studies such transfer for a complete diffusion-classification policy, freezing readouts, feature normalization, and the source-selected schedule while receiving-backbone losses guide allocation.

3 METHOD

TAPS allocates repeated noise draws across images under a fixed batch budget using relative class competition in already-computed denoising losses (Figure 2). Its stagewise *Full* readouts use the complete feature representation to rank images for further sampling, while a source-selected schedule determines how many continue. Final predictions retain the original mean-loss rule.

3.1 OBSERVED LOSSES AND SELF-SUPERVISION

For an image x with C candidate classes, let $L_{i,j,c}(x)$ denote the denoising loss for noise draw i, timestep τ_j, and class c. Gaussian noise tensors are independent across draws, with each tensor shared across all J timesteps and C classes. A complete draw evaluates all JC timestep–class combinations. After k draws, the mean-loss classifier predicts

$$s_{k,c}(x) = \frac{1}{kJ} \sum_{i=1}^{k} \sum_{j=1}^{J} L_{i,j,c}(x), \qquad \hat{y}_k(x) = \underset{c \in \{1,\dots,C\}}{\arg\min}\ s_{k,c}(x). \tag{1}$$

We use $J = 9$ timesteps and at most $K = 5$ draws, aggregating cached scalar losses in FP32. Appendix A.1 details loss computation, precision, and cache construction. Completed source rollouts also supply supervision. For a permutation $\pi = (\pi_1, \dots, \pi_K)$ of the cached draws, let $L_k^\pi(x)$ denote the prefix tensor with entries $(L_k^\pi(x))_{i,j,c} = L_{\pi_i,j,c}(x)$ for $i = 1, \dots, k$, and let $\hat{y}_k^\pi(x)$ be its prediction under Eq. 1. The stage-k target is

$$u_k^\pi(x) = \mathbf{1}\{\hat{y}_k^\pi(x) \neq \hat{y}_K(x)\}, \qquad k = 1, \dots, K-1. \tag{2}$$

The reference $\hat{y}_K(x)$ is computed once from all K cached draws and shared across permutations. This label-free target measures finite-reference disagreement, not semantic error or the benefit of the

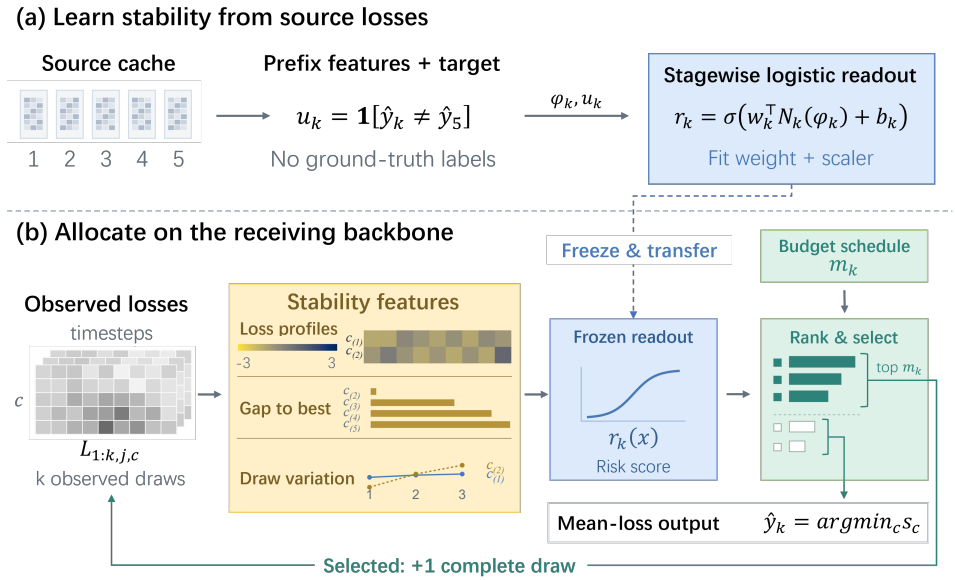

Figure 2: **TAPS training and budget allocation.** On the donor backbone's ImageNet data, disagreement between prefix and five-draw predictions supervises stagewise readouts. At inference, receiving-backbone loss features are scored using frozen donor normalization and readouts. A donor-selected schedule determines how many top-ranked images receive one additional complete draw; stopped images return their current mean-loss prediction.

next draw, and serves as a proxy for sampling priority. At inference, these priorities are computed from observed losses alone, without a separate reference computation.

3.2 Class-permutation-invariant loss features

Two observed prefixes can have similar mean-loss margins while differing in how consistently the aggregate winner is favored across noise levels and draws. We therefore retain both aggregate ambiguity and the underlying patterns of class competition. The feature map ϕ_k summarizes normalized loss gaps, leading-class profiles and ranks, winner agreement, and variation across observed draws. For a stage-k prefix, we suppress x and π and index draws in their observed order. Let $M_{j,c}$ be the draw-averaged loss, and let μ_j and σ_j be its mean and population standard deviation across classes at timestep τ_j. With $\delta = 10^{-6}$ and $\bar{\sigma}_j = \max(\sigma_j, \delta)$, define

$$M_{j,c} = \frac{1}{k}\sum_{i=1}^{k} L_{i,j,c}, \qquad Z_{j,c} = \frac{M_{j,c} - \mu_j}{\bar{\sigma}_j}, \qquad V_{j,c} = \frac{\mathrm{std}_{i=1,\dots,k}(L_{i,j,c})}{\bar{\sigma}_j} \quad (k > 1). \tag{3}$$

Here, Z expresses relative class evidence at each timestep, while V measures cross-draw variation using population standard deviation. Full combines the groups in Table 1, using the two classes with the lowest aggregate scores $s_{k,c}$ as leaders. Appendix A.2 gives complete definitions and concatenation order.

The features use class-symmetric statistics and score-ranked leading classes, without parameters tied to class identities. This construction allows the same feature definition to be applied across tasks with different candidate vocabularies. For $C \geq 5$ and $J = 9$, the feature dimension is independent of C: 67 at $k = 1$ and $103 + k$ at stages $2 \leq k < K$. Per-draw indicators retain their observed order. Appendix A.3 states the class-permutation-invariance property and its tie conditions.

Table 1: **Feature groups in the Full readout.** All features use observed classification losses. Leading classes are selected by aggregate score.

Group	Retained information
Aggregate ambiguity	Normalized gaps between the lowest aggregate score and the next four scores.
Local gaps and scales	Normalized gaps between each timestep's two lowest draw-averaged losses and relative class-loss scales across timesteps.
Leading-class profiles	Normalized losses and fractional ranks of the two leading classes; agreement between timestep-wise and aggregate winners.
Cross-draw variation	Loss dispersion summaries and ordered indicators of agreement between per-draw and aggregate winners ($k > 1$).

3.3 Stagewise risk learning and budget allocation

For each stage $k = 1, \dots, K-1$, we fit an ℓ_2-regularized logistic model to the target in Eq. 2:

$$r_k(x, \pi) = \mathrm{sigmoid}\big(w_k^\top \mathcal{N}_k(\phi_k(L_k^\pi(x))) + b_k\big), \tag{4}$$

where $\mathcal{N}_k$ standardizes features using training-partition statistics. Fitting uses inverse-frequency weights for agreement and disagreement targets, and r_k serves as a ranking score. Regularization is selected in the inner folds of three-by-three nested cross-validation, with all permutations grouped by image and both normalization and readouts fitted only on training folds. Final readouts and normalization are refitted on all source images and frozen for target inference. Benchmark class metadata is used for source-cache stratification, but not to define stability targets. Appendix A.4 provides training details.

To allocate samples using these scores, consider a batch $\mathcal{B}$ of N images sharing a candidate set and timestep grid, with average draw budget $B \in [1, K]$. A schedule specifies integer counts m_k of images receiving a $(k+1)$th draw, satisfying

$$N \geq m_1 \geq \cdots \geq m_{K-1} \geq 0, \qquad \sum_{k=1}^{K-1} m_k = \lfloor N(B-1) \rfloor. \tag{5}$$

Appendix A.5 gives the conversion from fractional schedules to integer counts.

All images receive one initial draw, with $A_0 = \mathcal{B}$. At stage k, active images in A_{k-1} have k draws, and the m_k images ranked highest by priority rule ρ_k continue:

$$A_k = \mathrm{Top}_{m_k}\big(A_{k-1}; \rho_k\big). \tag{6}$$

For Full, $\rho_k = r_k$ is recomputed from all observed draws using the frozen stage-specific readout. Selected images receive one additional draw; others stop. Each image returns its current mean-loss prediction upon stopping or reaching K draws.

Under these quotas, every priority rule uses exactly $N + \sum_{k=1}^{K-1} m_k = \lfloor NB \rfloor$ draws per batch, including initial observations, equivalent to $JC\lfloor NB \rfloor$ logical denoiser evaluations. Features require no additional denoiser evaluations; Appendix A.8 discusses runtime overhead.

3.4 Source-selected schedules

For each source backbone and evaluated budget, we select a schedule on its ImageNet source data using the Margin priority rather than a learned readout:

$$r_k^M(x) = -\frac{s_{k,(2)}(x) - s_{k,(1)}(x)}{\mathrm{std}_c(s_{k,c}(x)) + \delta}, \tag{7}$$

where $s_{k,(1)}$ and $s_{k,(2)}$ are the two lowest aggregate scores; larger priorities indicate greater ambiguity. We search feasible schedules whose continuation fractions, measured relative to the full batch, lie on a 0.1 grid. The objective is to maximize agreement between final predictions and the source backbone's five-draw reference, averaged over source images and draw orders under the source batching protocol (Appendix A.5). The selected schedule is frozen before target inference. For each target batch, its continuation fractions are converted to integer counts satisfying Eq. (5).

3.5 FROZEN POLICY TRANSFER

To transfer TAPS from backbone d to backbone m, we compute features from m's currently observed losses, standardize them using d's fitted normalization, and score them using d's stagewise weights and intercepts. We also retain the continuation fractions selected for each budget on d's ImageNet source data (Section 3.4). These donor components remain frozen, while m supplies all denoising losses and final mean-loss predictions. Native policies use the same procedure with $d = m$. Separately, the SDXL extension evaluates readout transfer under a fixed SD2.0/ImageNet Margin schedule shared by all readouts (Appendix C.4).

4 EXPERIMENTS

We evaluate TAPS's sampling efficiency and transfer, then examine observation-dependent stability and the contributions of class-competition features and stagewise priority updates.

4.1 EXPERIMENTAL SETUP

Models and controls. We evaluate frozen SD1.5 and SD2.0 (Rombach et al., 2022) on eight target tasks (12,023 query images per backbone). TAPS's Full readout and the Quad control are fitted on ImageNet-1K (Deng et al., 2009; Russakovsky et al., 2015) losses from the receiver (native) or other backbone (transferred), using finite-reference disagreement targets. Quad uses aggregate margins, loss scale, cross-draw statistics, and quadratic terms.

In Table 2, transferred TAPS uses the donor's readout, normalization, and ImageNet-selected Margin schedule. Native TAPS, both Quad variants, Margin, and random ranking use the receiver's Margin schedule. The shared-donor comparison below also evaluates transferred Quad under the donor schedule. Uniform-balanced distributes draws as evenly as possible, while unweighted PETS-OKG (Liu et al., 2026b) with a calibrated prior and a five-draw Blend-ASC adaptation (Feng et al., 2026) retain their own allocation rules (Appendices D.3 and D.4). All methods share target loss caches, draw budgets, and FP32 mean-loss classification; target labels enter neither readout fitting nor schedule selection.

Budgets and uncertainty. Unless stated otherwise, each task's query set forms one batch with $\lfloor NB \rfloor$ total draws at $B \in \{1.5, 2, \ldots, 4.5\}$, including initial observations. Results average over 20 orders of the same five cached draws. The primary metric is trapezoidal accuracy–budget AUC over $[1.5, 4.5]$, normalized by interval width and macro-averaged across tasks, with differences in percentage points (pp). We report 95% bootstrap intervals. Main transferred-TAPS comparisons use 2,000 image-bootstrap replicates, paired across policies and draw orders, with allocations recomputed on each resampled batch (*rerouting*). Separately reported *fixed-route* analyses retain saved allocations, with 5,000 replicates for update and stage-contribution intervals. Appendix B details the protocol and scope of uncertainty.

4.2 SAMPLING EFFICIENCY AND TRANSFER

We first test whether a source-trained TAPS policy remains useful when both the task and backbone change. Without receiver-side fitting, transferred TAPS improves eight-task macro AUC over Uniform-balanced by 1.718 pp on SD1.5 and 1.945 pp on SD2.0. It also exceeds Margin, native Quad, and the evaluated PETS-OKG and Blend-ASC adaptations at matched budgets, with macro AUCs close to native TAPS (Table 2; Figure 3).

Readout contribution under a shared donor schedule. Because complete-policy comparisons allow schedules to differ, we isolate the readout contribution by comparing transferred TAPS and transferred Quad under the same donor schedule, fixing all stagewise continuation counts. TAPS improves macro AUC by 0.131 $[0.023, 0.244]$ pp on SD1.5 and 0.213 $[0.127, 0.326]$ pp on SD2.0 (paired 95% rerouting intervals). The gains under fixed schedules support the Full readout's contribution to selecting which images receive further samples.

Table 2: **Sampling efficiency and task variation.** Normalized accuracy–budget AUC (%); higher is better. Left: eight-task macro average. Right: native Quad vs. transferred TAPS by task. Best values are bold; macro runners-up are underlined.

(a) Macro AUC

Method	SD1.5	SD2.0
Uniform-balanced	50.382	55.540
Random ranking	49.112	54.191
Margin	51.809	57.095
Quad (native)	51.960	57.206
Quad (transferred)	51.968	57.271
PETS-OKG (adapt.)	50.556	55.890
Blend-ASC (adapt.)	49.813	55.076
TAPS (native)	**52.116**	**57.515**
TAPS (transferred)	<u>52.100</u>	<u>57.485</u>

(b) Task AUC

	SD1.5		SD2.0	
Task	Quad	TAPS	Quad	TAPS
CIFAR-10	80.043	**80.246**	85.357	**85.663**
CIFAR-100	47.868	**48.481**	55.271	**55.781**
EuroSAT	**17.311**	17.212	28.130	**28.589**
Flowers-102	47.178	**47.204**	**57.266**	57.065
STL-10	89.643	**89.664**	91.799	**91.908**
Pets	**78.267**	78.092	77.349	**77.643**
DTD	35.803	**35.870**	43.469	**43.851**
Aircraft	19.565	**20.029**	19.007	**19.380**

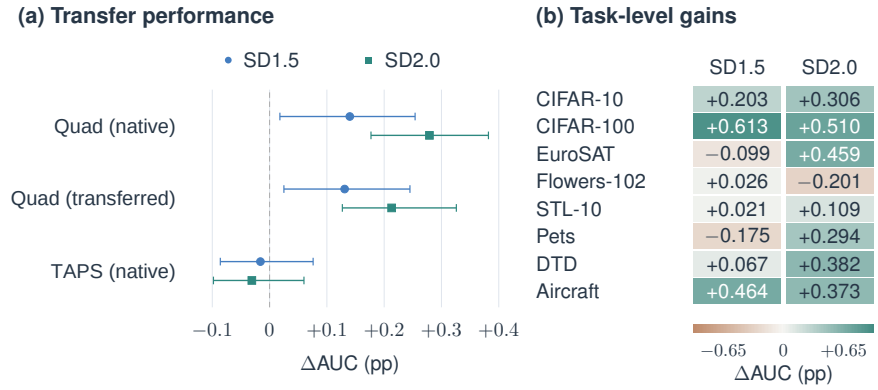

Figure 3: **Transfer performance and task-level gains.** Transferred TAPS minus (a) native Quad, transferred Quad, and native TAPS in eight-task macro AUC, with paired 95% rerouting intervals, and (b) native Quad per task (point estimates). Differences are in normalized accuracy–budget AUC (pp). Circles/squares denote SD1.5/SD2.0 receivers. Transferred TAPS uses the donor's readout, normalization, and schedule; all comparators shown here use the receiver schedule. Panel (b) uses native Quad AUCs reported to three decimal places.

Stability of the gains. To assess sensitivity to source sampling and fitting, we refit both readouts and their normalization on three random subsets, each containing 80% of the source images, and reselect a shared donor Margin schedule for each subset. On the same target loss caches, transferred TAPS exceeds transferred Quad in all three refits on both receivers (Table 3), supporting the consistency of its gains across the evaluated source fits.

Readout transfer to SDXL. To test transfer beyond the SD1.5–SD2.0 pair, we apply TAPS and Quad readouts trained on each donor's ImageNet losses to SDXL (Podell et al., 2024) on 320 fixed CIFAR-10 images. All four readouts and Margin share the preselected SD2.0/ImageNet Margin schedule, without receiver-side fitting or schedule selection. Both TAPS readouts exceed Uniform-balanced, but only the SD2.0-trained one exceeds Margin and its corresponding Quad; the SD1.5-trained one trails both. Thus, the advantage over stronger controls depends on the donor under this fixed schedule. Appendix C.4 provides the full setup and results.

4.3 Tracking stability with a frozen readout

We test whether frozen readouts track within-image stability changes as the observed noise or receiving backbone changes.

Table 3: **Stability of the gains.** Transferred TAPS minus transferred Quad in eight-task macro normalized accuracy–budget AUC (pp), across three refits on random subsets containing 80% of the source images. Within each refit, both methods use the same source subset and share a donor schedule reselected on that subset. Mean and sample standard deviation (SD) are computed across the three refits.

Receiver	Refit 1	Refit 2	Refit 3	Mean $\pm$ SD
SD1.5	$+0.1257$	$+0.0846$	$+0.1304$	$+0.1136 \pm 0.0252$
SD2.0	$+0.2403$	$+0.2601$	$+0.2272$	$+0.2425 \pm 0.0166$

Table 4: **Within-image stability ranking.** Task-macro pairwise ranking accuracy (%); entries are point estimates. The last row uses a shared three-draw reference excluding both compared observations; all other rows use the five-draw reference.

Input/readout	SD1.5	SD2.0
Any fixed image–backbone score	50.00	50.00
Margin	67.33	68.09
Quad (native)	69.86	70.09
TAPS readout (native)	67.98	69.08
TAPS readout (transferred)	68.14	69.01
TAPS readout (native), shuffled within image	49.99	50.01
TAPS readout (native), pair-excluded reference	64.19	64.87

Changing the observed noise. For each image–backbone pair, we compare the five single-draw predictions from its cache against a shared five-draw reference. Averaged across tasks, 74.65% of images on SD1.5 and 72.53% on SD2.0 have both agreeing and disagreeing predictions. To test whether readouts distinguish these states, we compare agreeing–disagreeing observation pairs within each image, scoring higher risk for the disagreeing observation as correct and ties as one half. This diagnostic ranks observations within an image, rather than images competing for a shared sampling budget.

Native and transferred TAPS readouts exceed the 50% ranking accuracy of any fixed image–backbone score, while shuffling the native readout's scores within each image reduces accuracy to chance (Table 4). To check dependence on including the compared draws in the reference, we recompute each pair's reference from the other three draws; the native TAPS readout remains above chance, with 95% intervals above 50% (Appendix C.2). These controls support tracking the current observation's stability beyond a fixed per-image difficulty score.

Changing the receiving backbone. We apply the same frozen ImageNet-trained TAPS readout and normalization to both backbones on matched images and initial noise, selecting the top 40% by risk after one draw. Let G_{15} and G_{20} denote images selected exclusively under SD1.5 and SD2.0, respectively, and let $u_{15}(x)$ and $u_{20}(x)$ indicate disagreement with each backbone's own five-draw reference. To test whether selection changes track backbone-specific instability beyond a common overall shift, we measure

$$I = \mathbb{E}[u_{15}(x) - u_{20}(x) \mid x \in G_{15}] - \mathbb{E}[u_{15}(x) - u_{20}(x) \mid x \in G_{20}]\,. \tag{8}$$

The interaction is 42.897 $[39.873, 45.522]$ pp for the SD1.5-trained readout and 42.903 $[40.103, 45.289]$ pp for the SD2.0-trained readout (95% intervals; 2,000 bootstrap replicates with groups reselected). Both positive interactions support using a frozen readout for backbone-specific sampling decisions.

4.4 Design choices for budget allocation

We evaluate how class-competition features, updated stability assessments, and sampling schedules contribute to allocation performance. For feature and priority-update comparisons, native and transferred readouts share the receiver-selected schedule within each receiving backbone, holding continuation counts fixed.

Table 5: **Feature and update contributions to allocation.** Macro normalized accuracy–budget AUC differences (pp) under shared receiver-selected schedules. The feature-block ablation retrains only stage one. U_Q and U_F are update gains for Quad and TAPS's Full readout over their fixed initial risk controls. Brackets are paired fixed-route 95% intervals. Update interactions use 5,000 replicates with shared image-bootstrap weights; individual update-gain intervals appear in Table 10. Differences are computed before rounding.

Source → receiver	Full minus ablated readout	U_Q	U_F	$U_F - U_Q$
SD1.5 → SD1.5	$+0.072$ $[0.013, 0.131]$	$+0.803$	$+0.929$	$+0.127$ $[0.050, 0.204]$
SD2.0 → SD1.5	$+0.082$ $[0.025, 0.141]$	$+0.784$	$+0.861$	$+0.077$ $[-0.015, 0.163]$
SD1.5 → SD2.0	$+0.119$ $[0.056, 0.183]$	$+0.743$	$+0.897$	$+0.154$ $[0.078, 0.234]$
SD2.0 → SD2.0	$+0.142$ $[0.078, 0.208]$	$+0.745$	$+0.885$	$+0.140$ $[0.049, 0.227]$

Contribution of class-competition features. We test whether local class competition provides allocation value beyond aggregate margins and normalized loss profiles. We remove the 36 first-stage local-gap, leading-class-rank, and winner-agreement features from TAPS's Full readout, reducing its first-stage feature dimension from 67 to 31. The ablated readout retains aggregate margins, local scales, and normalized leading-class loss profiles. Only the first-stage readout is retrained; later donor readouts, receiver schedules, classification, and budgets remain fixed. Removing the block lowers AUC in all four source–receiver settings (Table 5). Appendix C.3 reports a complementary ablation with quadratic controls.

Value of updating the stability assessment. We assess whether revising stability scores as samples accumulate improves allocation over retaining the initial assessment. The updated policy recomputes features from all observed draws and uses frozen stage-specific readouts. The fixed initial risk control retains its first-draw risk score for all later decisions. Both policies use the same schedule and update their mean-loss predictions as draws arrive, so the comparison isolates the value of updating the allocation priorities. For readout $a \in \{Q, F\}$ (Quad or TAPS's Full), let U_a denote the updated policy's AUC minus that of its fixed initial risk control.

Both readouts benefit from updating in all four settings (Table 5). Full has larger update-gain point estimates in all four settings, although the paired bootstrap interval for $U_F - U_Q$ includes zero for SD2.0→SD1.5. The gains over the fixed initial risk controls support stagewise stability reassessment for allocation, beyond updating the mean-loss prediction alone.

Role of the sampling schedule. The readout determines which images continue, while the schedule determines how many continue at each stage. To assess the schedule's contribution, we retain the Full readout's priority rule and replace the receiver-selected schedule with a balanced schedule. This reduces macro AUC by 1.127 pp on SD1.5 and 1.308 pp on SD2.0. Thus, the receiver-selected distribution of sampling effort across stages improves allocation over the balanced schedule even with the same priority rule. The allocation performance depends on both the readout's image ranking and the schedule's distribution of sampling depth.

5 Conclusion

TAPS learns stability predictors from class-permutation-invariant diffusion-loss features and uses them to allocate a shared sampling budget without changing the mean-loss classifier. Across eight tasks, TAPS policies transferred between SD1.5 and SD2.0 improve budgeted classification over uniform allocation, native quadratic controls, and the evaluated PETS-OKG and Blend-ASC adaptations. Transferred Full also improves over transferred Quad under the same donor schedule. Controlled feature ablations under shared receiver schedules show that local class competition contributes to allocation performance, while observation and backbone diagnostics show that a frozen readout can track changes in the stability of the receiving backbone's current predictions.

AI USE STATEMENT

We used AI tools to obtain critical feedback on the research methodology, experimental evaluation, and interpretation of results, and to assist with manuscript revision and the drafting of disclosure statements. AI-generated feedback served as suggestions for author consideration, rather than as evidence supporting the scientific claims. The authors take responsibility for the final manuscript, including the accuracy of its claims, analyses, references, and any content developed with AI assistance.

REPRODUCIBILITY STATEMENT

Section 3 describes the TAPS feature representation, self-supervised targets, stagewise readouts, budget allocation, and frozen-readout transfer protocol. Appendix A provides the loss construction and draw permutation procedure, complete Full feature definitions, conditions for class-permutation invariance, readout fitting and model selection, and schedule search with exact integer budget accounting. Section 4 and Appendix B describe the evaluation protocol, metrics, and bootstrap procedures. Appendices C and D provide additional transfer results, feature ablations, complete-policy comparisons, and the PETS-OKG and Blend-ASC adaptations.

REFERENCES

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 527–536. PMLR, 2017. URL https://proceedings.mlr.press/v70/bolukbasi17a.html.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3606–3613, 2014. URL https://openaccess.thecvf.com/content_cvpr_2014/html/Cimpoi_Describing_Textures_in_2014_CVPR_paper.html.

Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58921–58937. Curran Associates, Inc., 2023. doi: 10.52202/075280-2571. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b87bdcf963cad3d0b265fcb78ae7d11e-Abstract-Conference.html.

Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223. PMLR, 2011. URL https://proceedings.mlr.press/v15/coates11a.html.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ff414825df833edb8b1839e3d5d495e9-Abstract-Conference.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Zhe Dong, Fang Qin, and Manish Shah. When does learning to stop help? A cost-aware study of early exits in reasoning models. *arXiv preprint arXiv:2606.30852*, 2026. URL https://arxiv.org/abs/2606.30852.

Austin Feng, Marius Alonso, Ambroise Odonnat, Vasilii Feofanov, and Ievgen Redko. Optimal self-consistency for efficient reasoning with large language models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.12309.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, and Hao Zhang. Efficiently scaling LLM reasoning with Certaindex. In *Advances in Neural Information Processing Systems*, volume 38, pp. 157566–157604. Curran Associates, Inc., 2025. doi: 10.52202/085713-4751. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d037fd021c9aace128b8ce25001cdb6c-Paper-Conference.pdf.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242. URL https://doi.org/10.1109/JSTARS.2019.2918242.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk2aImxAb.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2206–2217, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Li_Your_Diffusion_Model_is_Secretly_a_Zero-Shot_Classifier_ICCV_2023_paper.html.

Pingzhi Li, Bairu Hou, Yun Zhu, Yihao Feng, Ke Ye, Tao Lei, Zhifeng Chen, Tianlong Chen, and Xianzhi Du. Adaptive thinking: Large language models know when to think in latent space. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/96dd7b14cde7e0a5194eceedafb2f249-Abstract-Conference.html.

Jiacheng Liu, Shengkun Tang, Jiacheng Cui, Dongkuan Xu, and Zhiqiang Shen. BiGain: Unified token compression for joint generation and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31931–31940, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_BiGain_Unified_Token_Compression_for_Joint_Generation_and_Classification_CVPR_2026_paper.html.

Zhangyi Liu, Huaizhi Qu, Xiaowei Yin, He Sun, Yanjun Han, Tianlong Chen, and Zhun Deng. PETS: A principled framework towards optimal trajectory allocation for efficient test-time self-consistency. *arXiv preprint arXiv:2602.16745*, 2026b. URL https://arxiv.org/abs/2602.16745.

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

Rohin Manvi, Joey Hong, Tim Seyde, Maxime Labonne, Mathias Lechner, and Sergey Levine. Zero-overhead introspection for adaptive test-time compute. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b22c5b6a87d259769fd186b016412f6a-Abstract-Conference.html.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47. URL https://www.robots.ox.ac.uk/~vgg/publications/2008/Nilsback08/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092. URL https://www.robots.ox.ac.uk/~vgg/publications/2012/parkhi12a/.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Zipeng Qi, Buhua Liu, Shiyan Zhang, Bao Li, Zhiqiang Xu, Haoyi Xiong, and Zeke Xie. A simple and efficient baseline for zero-shot generative classification. *arXiv preprint arXiv:2412.12594*, 2024. URL https://arxiv.org/abs/2412.12594.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://link.springer.com/article/10.1007/s11263-015-0816-y.

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4962–4979. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.406. URL https://aclanthology.org/2021.emnlp-main.406/.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17456–17472. Curran Associates, Inc., 2022. doi: 10.52202/068431-1269. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6fac9e316a4ae75ea244ddcef1982c71-Abstract-Conference.html.

Arundhati S. Shanbhag, Brian B. Moser, Tobias C. Nauen, Stanislav Frolov, Federico Raue, and Andreas Dengel. Just leaf it: Accelerating diffusion classifiers with hierarchical class pruning. *arXiv preprint arXiv:2411.12073*, 2024. URL https://arxiv.org/abs/2411.12073.

Jee Seok Yoon, Junghyo Sohn, Wootaek Jeong, and Heung-Il Suk. ExpertDiff: Head-less model reprogramming with diffusion classifiers for out-of-distribution generalization. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*, pp. 6866–6874, 2025. doi: 10.24963/ijcai.2025/764. URL https://www.ijcai.org/proceedings/2025/764.

Bowen Zuo and Yinglun Zhu. Strategic scaling of test-time compute: A bandit learning approach. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d8d2c5f79c53a2a685dbdf93f78c5695-Abstract-Conference.html.

Table 6: **Class-conditional prompts.** Each candidate uses one prompt. Class phrases retain the dataset-specific names and descriptions, including breed descriptions for Pets and variant names for Aircraft.

Dataset	Prompt form
ImageNet / STL-10	a photo of a {class}.
CIFAR-10 / CIFAR-100	a blurry photo of a {class}.
EuroSAT	a centered satellite photo of a {class phrase}.
Flowers-102	a photo of a {class}, a type of flower.
Pets	Dataset-specific breed description followed by ", a type of pet."
DTD	a photo of a {class} texture.
Aircraft	a photo of a {variant}, a type of aircraft.

A Additional method details

A.1 Coherent loss construction and draw permutations

Let z_0 be the fixed latent encoding of image x, and let e_c be the text embedding of its candidate-class prompt. For each image, let $\epsilon_i \sim \mathcal{N}(0, I)$ be independent noise draws. The cached coherent loss construction is

$$\begin{aligned} z_{i,j} &= \sqrt{\bar{\alpha}_{\tau_j}} z_0 + \sqrt{1 - \bar{\alpha}_{\tau_j}} \epsilon_i, \\ L_{i,j,c}(x) &= \frac{1}{D} \left\| \epsilon_i - \epsilon_\theta(z_{i,j}, \tau_j, e_c) \right\|_1, \end{aligned} \tag{9}$$

where D is the number of noise elements, ϵ_θ is the frozen denoiser, and $\bar{\alpha}_{\tau_j}$ is the forward-process coefficient. One noise tensor is shared across all timesteps and candidate classes within a draw. The timestep axis therefore represents evaluations at different noise levels, not successive states of a reverse-denoising chain.

Backbones and preprocessing. The main study uses `runwayml/stable-diffusion-v1-5` and `stabilityai/stable-diffusion-2-base`, denoted SD1.5 and SD2.0. Both use 512×512 inputs and ϵ-prediction. Images are resized bicubically to a short side of 512, center-cropped to 512×512, converted to RGB, and mapped to $[-1, 1]$. We use $z_0 = 0.18215\, \bar{z}(x)$, where $\bar{z}(x)$ is the VAE posterior mean, yielding a $4 \times 64 \times 64$ latent. The forward noise schedule has 1,000 steps, scaled-linear betas from 0.00085 to 0.012, and evaluated timesteps $\{100, 200, \ldots, 900\}$. Model evaluation and L1-loss generation follow an FP16 tensor path; the resulting scalar losses are stored in FP32, and all methods use the same FP32 mean-loss classification rule.

Text conditioning. Each candidate class has one fixed prompt, with dataset-specific forms listed in Table 6. Prompts are padded or truncated to 77 tokens, and token-level text-encoder outputs condition the denoiser without classifier-free guidance. Candidate columns follow the class indices in the prompt lists.

Noise draws and replay. Each image has $K = 5$ draws. Its noise stream is initialized with seed $42 + 1{,}000{,}003p + 97{,}409$, where p is its global position within the dataset cache. The same position-based seed rule is used across dataset caches and backbones. The main allocation experiments use 20 distinct permutations sampled without replacement from the $5! = 120$ possible orders with seed 130363. The same permutation bank is used across compared policies. For $\pi = (\pi_1, \ldots, \pi_K)$, an observed prefix satisfies $(L^\pi_k(x))_{i,j,c} = L_{\pi_i,j,c}(x)$ for $i \le k$. The complete-cache reference is computed once in a fixed reduction order and reused. This avoids changing the reference through floating-point reduction order.

Permutations supply different prefix states from the same cache. Cached replay exposes to a routing policy only losses already allocated to each image. Future losses construct source targets or evaluate outcomes, but do not enter deployed routing features.

Checkpoint metadata. Checkpoints are identified by the repository names above; exact repository revisions were not recorded.

A.2 COMPLETE FULL FEATURE MAP

We describe a stage-k prefix with $1 \le k < K$, suppressing image and draw-order indices. All standard deviations in this Full feature map use population normalization. Set $\delta = 10^{-6}$, $M_{j,c} = k^{-1}\sum_{i=1}^{k} L_{i,j,c}$, and $s_c = s_{k,c} = J^{-1}\sum_{j=1}^{J} M_{j,c}$. Let $c_{(1)}, \dots, c_{(C)}$ index classes in increasing aggregate-score order. The fixed-dimensional configuration assumes $C \ge 5$.

Aggregate ambiguity. With $\bar{\sigma}_s = \max(\mathrm{std}_c(s_c), \delta)$, retain

$$g_r = \frac{s_{c_{(r)}} - s_{c_{(1)}}}{\bar{\sigma}_s}, \qquad r = 2, 3, 4, 5. \tag{10}$$

These four features also define the learned Gap-only baseline in the additional-policy experiments.

Within-timestep geometry. Define

$$\begin{aligned} \mu_j &= \frac{1}{C}\sum_{c=1}^{C} M_{j,c}, \quad \sigma_j^2 = \frac{1}{C}\sum_{c=1}^{C}(M_{j,c} - \mu_j)^2, \\ Z_{j,c} &= \frac{M_{j,c} - \mu_j}{\max(\sigma_j, \delta)}, \quad a_j = \frac{\sigma_j}{\max(J^{-1}\sum_{\ell=1}^{J}\sigma_\ell, \delta)}. \end{aligned} \tag{11}$$

Let $Z_{j,[1]} \le \cdots \le Z_{j,[C]}$ be the within-timestep order statistics. Retain the local gap $d_j = Z_{j,[2]} - Z_{j,[1]}$ and relative scale a_j. The two entries defining d_j are selected separately at each timestep; they need not correspond to the aggregate leaders $c_{(1)}$ and $c_{(2)}$. The scale feature retains differences across timesteps after class-wise normalization.

Leading-class profiles and competition. For each globally leading class $c_{(r)}$, $r \in \{1, 2\}$, retain $Z_{j,c_{(r)}}$ and its fractional rank

$$\rho_{j,r} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{1}\{Z_{j,c} < Z_{j,c_{(r)}}\}. \tag{12}$$

Also retain $h_j = \mathbf{1}\{\arg\min_c M_{j,c} = c_{(1)}\}$. These features distinguish agreement across timesteps from an aggregate margin produced by conflicting local evidence.

Variation across observed draws. For $k > 1$, define

$$\begin{aligned} V_{j,c} &= \frac{[k^{-1}\sum_{i=1}^{k}(L_{i,j,c} - M_{j,c})^2]^{1/2}}{\max(\sigma_j, \delta)}, \\ \bar{V}_j &= \frac{1}{C}\sum_{c=1}^{C} V_{j,c}, \qquad V_j^{\max} = \max_c V_{j,c}. \end{aligned} \tag{13}$$

At each timestep retain $\bar{V}_j$, $V_j^{\max}$, $V_{j,c_{(1)}}$, and $V_{j,c_{(2)}}$. For each observed draw also retain

$$b_i = \mathbf{1}\left\{ \arg\min_c \frac{1}{J}\sum_{j=1}^{J} L_{i,j,c} = c_{(1)} \right\}, \qquad i = 1, \dots, k. \tag{14}$$

Concatenation. Using brackets for vectors ordered by timestep, the base vector is

$$\begin{aligned} \phi_k^{\mathrm{base}} = \mathrm{concat}\big([g_r]_{r=2}^{5}, [d_j]_j, [a_j]_j, \\ [Z_{j,c_{(1)}}]_j, [Z_{j,c_{(2)}}]_j, [\rho_{j,1}]_j, [\rho_{j,2}]_j, [h_j]_j\big). \end{aligned} \tag{15}$$

For $k = 1$, set $\phi_k = \phi_k^{\mathrm{base}}$. For $k > 1$, append $[\bar{V}_j]_j$, $[V_j^{\max}]_j$, $[V_{j,c_{(1)}}]_j$, $[V_{j,c_{(2)}}]_j$, and $[b_i]_{i=1}^{k}$, in that order. Consequently,

$$\dim(\phi_k) = \begin{cases} 4 + 7J, & k = 1, \\ 4 + 11J + k, & 2 \le k < K. \end{cases} \tag{16}$$

For $J = 9$ and $K = 5$, the dimensions are 67, 105, 106, and 107. The per-draw indicators retain observation order. No raw image, latent encoding, noise tensor, or class-specific embedding enters the readout.

A.3 CLASS-PERMUTATION INVARIANCE AND ITS SCOPE

Let P permute the class axis, with $(L_k P)_{i,j,c} = (L_k)_{i,j,p(c)}$ for a bijection p. Assume unique aggregate ranks for the leading classes and unique minima for the winner indicators. Then

$$\phi_k(L_k P) = \phi_k(L_k). \tag{17}$$

Class-wise moments, order-statistic values, and pooled dispersions do not change under permutation. The score-ranked leaders identify the same candidates after relabeling, so their loss profiles, fractional ranks, and winner-equality indicators also remain unchanged. Every component of the concatenation is therefore invariant.

This property concerns relabeling a fixed candidate set. Adding or removing candidates can change moments, leaders, and ranks. If finite-precision ties are resolved by numerical class indices, strict invariance need not hold on tied inputs; Eq. (17) is asserted under the uniqueness conditions above.

A.4 RISK FITTING AND SOURCE-ONLY MODEL SELECTION

Training examples are image/order pairs with features $\phi_k(L_k^\pi(x))$ and target $u_k^\pi(x)$. For coordinate ℓ, the fitted feature normalization $\mathcal{N}_k$ (StandardScaler) is

$$\widetilde{\phi}_{k,\ell} = (\mathcal{N}_k(\phi_k))_\ell = \frac{\phi_{k,\ell} - \mu_{k,\ell}^{\text{train}}}{s_{k,\ell}^{\text{train}}}. \tag{18}$$

Here $s_{k,\ell}^{\text{train}}$ is the scaler's fitted scale parameter, rather than the timestep normalization in Eq. (11). The scaler is fitted inside each training fold and transferred with the final readout.

At a fixed stage, consider a training partition with n examples and $n_a > 0$ examples of each target $a \in \{0, 1\}$, where $a = 0$ denotes agreement and $a = 1$ disagreement with the finite reference. We use inverse-frequency target weights $\alpha_a = n/(2n_a)$. With $\ell(u, r) = -u \log r - (1 - u) \log(1 - r)$, fitting uses

$$\min_{w,b} \sum_{i=1}^{n} \alpha_{u_i} \ell\big(u_i, \text{sigmoid}(w^\top \widetilde{\phi}_i + b)\big) + \frac{\|w\|_2^2}{2\gamma}. \tag{19}$$

The intercept is unpenalized. We fit each readout with L-BFGS using at most 1,000 iterations and select $\gamma \in \{0.003, 0.01, 0.03, 0.1, 0.3\}$ separately for each stage.

Grouped nested cross-validation. Three outer folds hold out images and all their permutations together. Each outer training partition selects γ using three-fold inner GroupKFold, ordering candidates by higher validation ROC-AUC, then lower log loss, then stronger regularization (smaller γ). Normalization and the readout are refitted on the outer training images before predicting the held-out images. Repeating this process produces an out-of-fold (OOF) score $\widetilde{r}_k(x, \pi)$ for each source pair. Grouped cross-validation on the full source set subsequently selects regularization for the final source refit.

OOF scores support readout-dependent schedule selection in the additional-policy studies. The main transferred policy uses a schedule selected from the donor backbone's ImageNet Margin scores in Eq. (7). Margin does not require a learned readout. Native baselines use Margin schedules selected on the receiving backbone's ImageNet losses. The readout-transfer controls in Appendix C.1 hold this receiver schedule fixed, whereas the shared-donor comparison in Section 4.2 holds the donor schedule fixed for transferred Full and Quad. Semantic labels define neither the instability target nor the schedule objective, although source-cache stratification uses benchmark class metadata.

Ranking scores versus probabilities. For fixed positive target weights and true disagreement probability p at a given feature value, the minimizer of the conditional, unregularized weighted log loss is

$$r^*(p) = \frac{\alpha_1 p}{\alpha_1 p + \alpha_0 (1 - p)}. \tag{20}$$

For positive weights this is increasing in p, but generally differs from p. We therefore use the fitted sigmoid as a ranking score without assuming probability calibration.

A.5 SCHEDULE SEARCH, INTEGER QUOTAS, AND DEPLOYMENT

For each evaluated budget, schedule search uses

$$\mathcal{Q}_B = \left\{ q \in \{0, 0.1, \ldots, 1\}^{K-1} : q_1 \geq \cdots \geq q_{K-1}, \quad \sum_{k=1}^{K-1} q_k = B - 1 \right\}. \tag{21}$$

Let Π denote the draw-order bank used for source schedule search. Each candidate is evaluated on the source batch for every $\pi \in \Pi$. The objective averages agreement between the resulting final predictions and the source backbone's five-draw reference over source images and orders. The main transferred policy evaluates candidates with donor-source Margin priorities and retains the selected schedule at deployment. For the separately reported own-schedule variants, learned priorities use the OOF scores from Appendix A.4. Schedule ties favor the lexicographically larger continuation vector, preferring broader early coverage.

Exact integer accounting. Let $T_N = \lfloor N(B-1) \rfloor$, $a_k = \lfloor N q_k \rfloor$, and $f_k = N q_k - a_k$. There are $R = T_N - \sum_k a_k$ remaining draws after assigning the floor quotas. Assign one extra unit to the R largest fractional remainders, with ties favoring earlier stages, and call the selected stage indices $\mathcal{R}$. Then

$$m_k = a_k + \mathbf{1}\{k \in \mathcal{R}\}, \qquad \sum_{k=1}^{K-1} m_k = T_N. \tag{22}$$

The counts remain nested: if adjacent floor quotas are equal, the earlier stage has at least as large a remainder and receives priority in a tie; if they differ, incrementing the smaller by one cannot reverse their order. Thus selecting m_k images from the previous active set is feasible.

For a fixed batch and draw order, let a denote the priority rule used in Eq. (6), and let $\kappa_{q,a}(x)$ be the total number of draws allocated to image x under schedule q. Writing A_k for the resulting active sets gives

$$\kappa_{q,a}(x) = 1 + \sum_{k=1}^{K-1} \mathbf{1}\{x \in A_k\}.$$

Since $|A_k| = m_k$, the total draw count is

$$\sum_{x \in \mathcal{B}} \kappa_{q,a}(x) = N + \sum_{k=1}^{K-1} |A_k| = N + T_N = \lfloor NB \rfloor. \tag{23}$$

This applies separately to each batch, including a smaller final batch.

Deployment. Every image first receives one coherent draw, counted toward the budget. At stage k, each active image has exactly k draws; its current risk determines whether it is among the top m_k eligible images. Selected images receive one additional complete draw. Images leaving the active set return their current mean-loss prediction; images remaining after the final continuation return the five-draw prediction. If the next active set is empty, no further draws are evaluated. Routing uses only allocated losses and requires no separate reference computation. Batch membership and risk-tie resolution can affect which images continue, but do not change the prescribed cardinalities.

A.6 ROLE OF THE FINITE-REFERENCE TARGET

Disagreement with the K-draw reference is a surrogate for deciding which images should receive more samples. To see its endpoint interpretation, consider N images with k observed draws each, of which exactly m must continue to K draws at equal per-image cost. Let $p_{n,k} = \Pr\{\hat{y}_k(x_n) \neq \hat{y}_K(x_n) \mid \mathcal{H}_k\}$, where $\mathcal{H}_k$ contains the currently observed information. For a continuation set $A \subseteq \{1, \ldots, N\}$ with $|A| = m$, selected images return their K-draw predictions, while unselected images retain their k-draw predictions. Let $\mathrm{Agree}(A)$ denote the number of these final predictions that agree with their respective K-draw references. Then

$$\mathbb{E}[\mathrm{Agree}(A) \mid \mathcal{H}_k] = N - \sum_{n=1}^{N} p_{n,k} + \sum_{n \in A} p_{n,k}. \tag{24}$$

Thus the largest disagreement probabilities are optimal for this restricted endpoint objective. TAPS instead allocates one draw at a time and is evaluated by semantic accuracy. The finite reference can be wrong, so endpoint disagreement identifies where a prediction may change, not whether that change will improve accuracy or justify the next draw.

A.7 Baseline readouts

The learned Gap-only readout uses the four gaps in Eq. (10). The Margin priority is defined in Eq. (7). Its additive stabilizer differs from the clipped denominator of the learned gap features.

Quadratic control. Quad uses the four aggregate gaps together with loss scale and observed cross-draw statistics. With $\mu_s = C^{-1}\sum_c s_c$, its first six base features are

$$(v_1, \dots, v_6) = \big(g_2, g_3, g_4, g_5, \log \bar{\sigma}_s, \log \max(\mu_s, 10^{-12})\big). \tag{25}$$

For $k > 1$, fix the current leading classes $a = c_{(1)}$ and $b = c_{(2)}$, and define their per-draw gap and its sample standard deviation:

$$\begin{aligned} \Delta_i &= \frac{1}{J}\sum_{j=1}^{J}(L_{i,j,b} - L_{i,j,a}), \qquad \bar{\Delta} = \frac{1}{k}\sum_{i=1}^{k}\Delta_i, \\ \sigma_\Delta &= \left[\frac{1}{k-1}\sum_{i=1}^{k}(\Delta_i - \bar{\Delta})^2\right]^{1/2}. \end{aligned} \tag{26}$$

The remaining base features are

$$\begin{aligned} v_7 &= \sigma_\Delta/\bar{\sigma}_s, \qquad v_8 = \frac{1}{k}\sum_{i=1}^{k}\mathbf{1}\{\Delta_i < 0\}, \\ v_9 &= \mathrm{clip}\left(\frac{\bar{\Delta}}{\max(\sigma_\Delta/\sqrt{k}, \delta\bar{\sigma}_s)}, -20, 20\right), \\ v_{10} &= \frac{1}{k}\sum_{i=1}^{k}\mathbf{1}\left\{\arg\min_c \frac{1}{J}\sum_{j=1}^{J} L_{i,j,c} \neq a\right\}. \end{aligned} \tag{27}$$

Writing $p_k = 6$ for $k = 1$ and $p_k = 10$ for $k > 1$, the feature vector is

$$\psi_k = \mathrm{concat}\big([v_i]_{i=1}^{p_k}, [v_i v_j]_{1 \le i \le j \le p_k}\big), \tag{28}$$

where quadratic terms follow lexicographic order in (i, j). This gives 27 features at stage one and 65 at each of stages two to four. StandardScaler is applied to the expanded vector before fitting the logistic readout using Appendix A.4. The readout includes an intercept; no constant feature is appended.

A.8 Computational cost and protocol scope

With fixed candidate count C and timestep count J, a coherent draw requires JC logical image–class–timestep denoiser evaluations. The total allocated work is

$$JC\sum_{x\in\mathcal{B}}\kappa_{q,a}(x) = JC\lfloor NB\rfloor. \tag{29}$$

These are logical evaluations, not necessarily separate forward calls. Image and class batching can combine evaluations, so draws are proportional to denoiser work only within a fixed configuration, not universally to runtime across tasks or hardware.

Feature construction uses already observed losses and no additional denoiser evaluations. For a readout with d_k input features at stage k, scoring an image costs $O(d_k)$ arithmetic; sorting n active images costs $O(n \log n)$. Feature extraction, sorting, and smaller denoiser batches may nevertheless affect latency. Offline construction of a complete evaluation cache is separate from the simulated inference work. The reported cached experiments do not establish end-to-end speedup or source-training amortization.

Table 7: **Target query subsets.** Counts refer to the shared image set evaluated by each backbone. EuroSAT uses the full RGB image pool; DTD uses partition 1, and Aircraft uses variant labels.

Dataset	Classes	Sampling pool	Queries
CIFAR-10	10	train	320
CIFAR-100	100	train	3,200
EuroSAT	10	full RGB pool	320
Flowers-102	102	test	2,351
STL-10	10	train	320
Oxford-IIIT Pets	37	trainval	1,184
DTD	47	train, partition 1	1,128
FGVC Aircraft	100	trainval	3,200
Total			12,023

B Evaluation protocol

B.1 Evaluation scope

Source images. Source fitting uses 1,000 images from the ImageNet-1k validation set, one sampled without replacement from each of its 1,000 classes. Sampling uses NumPy `default_rng`, with class seed 2025 and image seed 42. Each source image is evaluated against all 1,000 candidate classes. Image-grouped cross-validation keeps all draw orders of an image in the same fold.

Target queries. The main study evaluates fixed query subsets from eight datasets. The targets are CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), EuroSAT (Helber et al., 2019), Flowers-102 (Nilsback & Zisserman, 2008), STL-10 (Coates et al., 2011), Oxford-IIIT Pets (Parkhi et al., 2012), DTD (Cimpoi et al., 2014), and FGVC Aircraft (Maji et al., 2013). Table 7 specifies the data pools and query counts. For each dataset, the NumPy random seed is reset to 42 and classes are processed in increasing class-index order. For a class with at least 48 images, 48 are sampled without replacement; the first 16 form a reserved subset and the remaining 32 form the query subset. For smaller classes, all images are shuffled, up to 16 are reserved, and the remainder form the queries. Reserved images precede queries in the cache and are excluded from the main evaluation and from fitting the transferred readouts, fitting their feature normalization, or selecting their schedules. Both backbones use the same 12,023 query images. Each complete query subset forms one allocation batch in the main protocol. Reference disagreement supplies source supervision; semantic labels are used for classification accuracy and post-hoc correction/degradation analyses.

Transfer protocol. The main transferred Full policy uses the donor's readout weights, intercepts, fitted feature normalization, and ImageNet Margin schedule. The receiving backbone supplies the observed losses and mean-loss classifier. Native controls fit their readouts and feature normalization and select their Margin schedules on the receiving backbone's ImageNet losses. Transferred Quad is evaluated with either the receiver or donor schedule, as specified in each comparison. The shared-donor comparison in Section 4.2 gives transferred Full and Quad the same donor schedule. By contrast, the readout-transfer controls in Appendix C.1 and the feature and update analyses in Appendix C.3 share the receiver's Margin schedule. PETS-OKG and Blend-ASC use their own budget-matched allocation rules.

For the source-image sensitivity analysis, each of three refits uses a random subset of 800 source images. Within a refit, Full and Quad are fitted on the same subset, including feature standardization, and share the donor Margin schedule reselected on that subset. Evaluation uses the fixed target query caches.

SDXL extension. The SDXL extension applies Full and Quad trained on ImageNet losses from each of SD1.5 and SD2.0, with their fitted feature normalization, to 320 fixed CIFAR-10 target images. All four readouts and Margin use the preselected SD2.0/ImageNet Margin schedule; Uniform-balanced matches their total draw budget. No SDXL source cache is constructed, and no receiver-side readout training or schedule selection is performed. This extension evaluates frozen readout reuse under the specified schedule.

B.2 Accuracy–budget AUC and uncertainty

For task d, let $a_d(B_j)$ be classification accuracy in percent at budget B_j, averaged over the common draw-order bank. The seven budgets range from $B_1 = 1.5$ to $B_7 = 4.5$ in increments of 0.5. The normalized area is

$$\mathrm{AUC}_d = \frac{1}{B_7 - B_1} \sum_{j=1}^{6} (B_{j+1} - B_j) \frac{a_d(B_j) + a_d(B_{j+1})}{2}. \tag{30}$$

Macro results average AUC_d equally across tasks. Unless otherwise noted, differences are calculated before rounding and reported in percentage points. This measure differs from ROC-AUC for ranking reference disagreement.

Rerouting uncertainty. The main transferred-Full contrasts against readout and allocation controls, and the shared-receiver-schedule contrasts in Appendix C.1, use 2,000 paired image-bootstrap replicates with routing recomputed on each resampled query batch. Within each task, each replicate samples N_d images with replacement, retaining each sampled image's paired outcomes across methods, budgets, both backbones, and all 20 draw orders. Repeated images occupy repeated slots in the resampled batch. Each policy recomputes its allocation on the resampled batch; fitted readouts, feature standardization, source-selected schedules, and source-calibrated allocation parameters remain frozen. Each method and order retains the prescribed $\lfloor N_d B \rfloor$ total draws. Within each replicate, results first average over draw orders and then equally over the eight tasks; tasks themselves are not resampled.

Fixed-route analyses. Update-gain and stage-contribution analyses use 5,000 paired-image bootstrap replicates that resample image contributions while retaining the saved allocations. These intervals condition on the saved routes and complement the rerouting analyses, which recompute allocations after resampling.

For the transfer, update, stage-contribution, and pair-excluded analyses, bootstrap intervals use the 2.5th and 97.5th percentiles with linear quantile interpolation. They are pointwise and unadjusted, conditional on the evaluated tasks, caches, fitted readouts, and schedules; fixed-route intervals additionally condition on saved routes. The 20 orders reuse five cached draws per image and are not independent cache replications. These intervals include neither retraining uncertainty nor independent-noise replication, and do not quantify generalization to an arbitrary new dataset.

Uniform-balanced allocation. For $h = \lfloor B \rfloor$, Uniform-balanced assigns h draws to every image and one additional draw to $R = \lfloor N_d B \rfloor - N_d h$ images selected uniformly without replacement. Random selection uses the fixed task seed $26091402+$ task index and does not use observed losses. Thus each draw order uses exactly $\lfloor N_d B \rfloor$ draws. The main comparison and the Uniform-balanced column in Table 15 use this protocol; the normalized AUC is calculated from unrounded accuracies.

B.3 Observation and selection diagnostics

Within-image ranking. For each image–backbone pair, the within-image diagnostic compares its five single-draw predictions against one fixed five-draw reference. Here an observation is stable if its prediction agrees with the reference and unstable if it disagrees. For a stable/unstable pair from the same image, higher risk for the unstable observation scores one, a tie scores one half, and a reversal scores zero. Any fixed image–backbone score necessarily ties within these pairs.

Pair-excluded reference. The pair-excluded check enumerates the ten pairs of initial draws and constructs each pair's common reference from the other three draws using FP32 mean-loss aggregation. Only pairs with one agreeing and one disagreeing observation contribute. Scores are averaged first over eligible pairs within each image, then over eligible images within each task, and finally equally across tasks. For this pair-excluded analysis, there are 8,814/8,859 eligible images and 34,103/35,086 eligible pairs on SD1.5/SD2.0, respectively, out of 12,023 query images per backbone. Intervals use 5,000 whole-image multinomial bootstrap replicates, retaining all pairs within each sampled image and recomputing the eligible-image denominator. This is a frozen-score ranking analysis without budget allocation.

Backbone-dependent selection. The fixed-readout backbone diagnostic applies the same donor readout and fitted feature normalization to matched images and initial noise across SD1.5 and SD2.0. After one draw, the top 40% by risk are selected under each backbone. Each image's disagreement is measured against that backbone's own five-draw reference. The mutually exclusive selection groups are G_{15} and G_{20} in Eq. (8). The interaction I compares the backbone-specific disagreement differences between these groups and is reported as $100I$ in percentage points. The reported intervals use 2,000 bootstrap replicates, with the selection groups recomputed in each replicate.

Table 8: **Controlled readout-transfer contrasts on SD1.5.** Transferred Full minus the indicated readout, in normalized accuracy–budget AUC pp. Brackets are 95% intervals from 2,000 paired-image rerouting replicates. Macro equally weights eight tasks; all readouts share the receiver's ImageNet Margin schedule.

Task	− native Quad	− transferred Quad	− native Full
	$+0.203$	$+0.118$	-0.020
CIFAR-10	$[-0.189, 0.478]$	$[-0.191, 0.432]$	$[-0.271, 0.190]$
	$+0.611$	$+0.630$	-0.001
CIFAR-100	$[0.426, 0.761]$	$[0.443, 0.771]$	$[-0.125, 0.077]$
	-0.120	-0.154	-0.049
EuroSAT	$[-0.807, 0.479]$	$[-0.750, 0.431]$	$[-0.520, 0.482]$
	$+0.027$	$+0.017$	$+0.046$
Flowers-102	$[-0.125, 0.189]$	$[-0.126, 0.184]$	$[-0.052, 0.145]$
	$+0.021$	$+0.051$	-0.060
STL-10	$[-0.217, 0.300]$	$[-0.204, 0.318]$	$[-0.234, 0.177]$
	-0.175	-0.157	-0.054
Pets	$[-0.349, 0.070]$	$[-0.363, 0.073]$	$[-0.158, 0.089]$
	$+0.062$	$+0.040$	-0.003
DTD	$[-0.285, 0.413]$	$[-0.292, 0.375]$	$[-0.195, 0.202]$
	$+0.464$	$+0.477$	-0.014
Aircraft	$[0.272, 0.675]$	$[0.283, 0.691]$	$[-0.113, 0.133]$
	$+0.137$	$+0.128$	-0.019
Macro	$[0.014, 0.251]$	$[0.020, 0.241]$	$[-0.091, 0.073]$

Table 9: **Controlled readout-transfer contrasts on SD2.0.** Transferred Full minus the indicated readout, in normalized accuracy–budget AUC pp. Brackets are 95% intervals from 2,000 paired-image rerouting replicates. Macro equally weights eight tasks; all readouts share the receiver's ImageNet Margin schedule.

Task	− native Quad	− transferred Quad	− native Full
	$+0.306$	$+0.267$	-0.051
CIFAR-10	$[0.042, 0.617]$	$[0.008, 0.591]$	$[-0.234, 0.225]$
	$+0.512$	$+0.409$	$+0.005$
CIFAR-100	$[0.386, 0.656]$	$[0.271, 0.560]$	$[-0.093, 0.086]$
	$+0.473$	$+0.223$	$+0.040$
EuroSAT	$[-0.081, 1.102]$	$[-0.275, 0.884]$	$[-0.404, 0.581]$
	-0.201	-0.209	$+0.012$
Flowers-102	$[-0.345, -0.041]$	$[-0.338, -0.048]$	$[-0.101, 0.121]$
	$+0.108$	$+0.082$	$+0.020$
STL-10	$[-0.010, 0.254]$	$[-0.040, 0.228]$	$[-0.104, 0.163]$
	$+0.293$	$+0.273$	-0.011
Pets	$[0.076, 0.506]$	$[0.072, 0.506]$	$[-0.135, 0.146]$
	$+0.383$	$+0.322$	-0.120
DTD	$[0.077, 0.638]$	$[0.037, 0.652]$	$[-0.290, 0.080]$
	$+0.376$	$+0.359$	-0.120
Aircraft	$[0.144, 0.600]$	$[0.114, 0.556]$	$[-0.236, -0.003]$
	$+0.281$	$+0.216$	-0.028
Macro	$[0.179, 0.385]$	$[0.129, 0.328]$	$[-0.095, 0.064]$

C Additional analyses of stability and readout transfer

C.1 Readout transfer under a shared receiver schedule

Tables 8 and 9 compare readouts while holding the receiver's ImageNet Margin schedule fixed for every method. Here transferred Full also uses the receiver schedule, unlike the donor-schedule configuration in Table 2; its donor readouts and feature normalization remain frozen. These task-level and macro contrasts therefore isolate readout transfer under a shared receiver schedule. On SD2.0, transferred Full trails native and transferred Quad on Flowers-102, and trails native Full on Aircraft, with the corresponding pointwise rerouting intervals below zero.

Table 10: **Update gains and their paired interaction.** Macro normalized accuracy–budget AUC differences (pp) under shared receiver-selected schedules, with 95% intervals from 5,000 paired-image fixed-route replicates. U_F and U_Q are Full and Quad AUC gains from priority updates relative to their fixed initial risk controls. The interaction is computed within each replicate, using shared image-resampling weights.

Source → receiver	U_F	U_Q	$U_F - U_Q$
SD1.5 → SD1.5	$+0.929$ $[0.814, 1.054]$	$+0.803$ $[0.694, 0.915]$	$+0.127$ $[0.050, 0.204]$
SD2.0 → SD1.5	$+0.861$ $[0.742, 0.988]$	$+0.784$ $[0.674, 0.899]$	$+0.077$ $[-0.015, 0.163]$
SD1.5 → SD2.0	$+0.897$ $[0.767, 1.024]$	$+0.743$ $[0.620, 0.864]$	$+0.154$ $[0.078, 0.234]$
SD2.0 → SD2.0	$+0.885$ $[0.760, 1.009]$	$+0.745$ $[0.617, 0.878]$	$+0.140$ $[0.049, 0.227]$

C.2 Within-image observation controls

Table 4 reports the within-image point estimates. Stable and unstable observations are compared within each fixed image and backbone; ties contribute one half. Native Full obtains 67.98% and 69.08%, with paired 95% intervals $[66.96, 68.96]$ and $[68.12, 70.01]$. Native Quad obtains 69.86% and 70.09%, exceeding native Full on this diagnostic. The allocation and within-image ranking experiments answer different questions: one measures classification accuracy after distributing a batch budget, while the other measures ordering among different observations of an individual image.

Within-image score shuffling reduces native Full's point estimates to 49.99% and 50.01%, showing that its correspondence with the current observation matters. For the pair-excluded reference, native Full scores 64.186% on SD1.5, with a 95% interval of $[63.059, 65.278]\%$. On SD2.0 it scores 64.867%, with interval $[63.766, 65.954]\%$. Both whole-image bootstrap intervals lie above chance. Static image information may still predict average instability; the structural 50% reference concerns its inability to distinguish observations of the same image when its score is held fixed.

C.3 Feature ablations, stagewise priority updates, and allocation gains

The allocation analyses in this subsection use the receiving backbone's ImageNet Margin schedule, held fixed across the compared readouts.

Ablation of the deployed representation. The feature-block ablation in Table 5 removes the 36 first-stage local-gap, leading-class-rank, and winner-agreement features. It retains aggregate margins, local scales, and normalized leading-class loss profiles. Only the first-stage readout is retrained; later donor readouts, receiver schedules, classification, and budgets remain fixed.

Ablation with quadratic controls. A complementary ablation retains the 27 first-stage Quad features and 27 local-scale/normalized leading-class profile features in both models. Adding the same 36 local-gap, rank, and winner-agreement features increases the first-stage dimension from 54 to 90. Only the first-stage readouts are retrained. Adding the block yields small positive task-macro AUC differences in both transfer directions. For native SD1.5, the interval for this AUC difference includes zero; some task-level differences are negative.

Updating allocation priorities. The updated policy recomputes features from the accumulated observations and applies the corresponding frozen stage-specific readout. The fixed initial risk control instead retains each image's first-draw risk for every later selection. Both policies use the same schedule and update their mean-loss predictions as draws arrive. For $a \in \{F, Q\}$ (Full or Quad), U_a is the updated policy's AUC minus that of its fixed initial risk control. Table 10 reports these update gains and their paired interaction $U_F - U_Q$. Both readouts benefit from priority updates in all four settings, but the interaction interval includes zero for SD2.0→SD1.5.

Table 11: **Accuracy gains under budgeted routing.** Stage contributions for transferred Full minus native Quad under shared receiver-selected schedules, in task-macro accuracy pp with complete-query denominators. Totals use unrounded contributions. Stagewise 5,000-replicate fixed-route intervals appear in Tables 12 and 13.

Receiver	B	$1 \to 2$	$2 \to 3$	$3 \to 4$	$4 \to 5$	Total
SD1.5	2	$+0.112$	$+0.138$	$+0.047$	$+0.033$	$+0.329$
SD1.5	3	$+0.034$	$+0.032$	$+0.015$	$+0.006$	$+0.086$
SD2.0	2	$+0.211$	$+0.114$	$+0.090$	$+0.104$	$+0.519$
SD2.0	3	$+0.101$	$+0.065$	$+0.079$	$+0.062$	$+0.306$

From continuation choices to classification gains. At a fixed first-draw top-40% cutoff, we compare groups selected exclusively by transferred Full or a Quad control. Their net gain is the correction rate minus degradation rate after a second draw, computed within each task and then macro-averaged using semantic labels only for evaluation. Against transferred Quad, Full-only versus Quad-only gains are 7.405 versus 5.199 pp on SD1.5 and 9.001 versus 4.901 pp on SD2.0; native-Quad comparisons have the same point-estimated direction. Some task-level gains are negative.

We separately decompose budgeted routing at $B = 2, 3$, comparing transferred Full (F) with native Quad (Q) under the shared receiver-selected schedule. For a fixed task and draw order, let $A_k^a(B)$ be the images continuing from k to $k+1$ draws under policy $a \in \{F, Q\}$. Define the stage contribution in accuracy percentage points as

$$G_k^a(B) = \frac{100}{N} \sum_{x \in A_k^a(B)} \left[\mathbf{1}\{\hat{y}_{k+1}(x) = y(x)\} - \mathbf{1}\{\hat{y}_k(x) = y(x)\} \right], \tag{31}$$

where $y(x)$ is the semantic label and N is the complete query size, not the size of the continuing set. Task and draw-order indices are suppressed here. Images not continued at stage k contribute zero. Writing $\mathrm{Acc}^a(B)$ for final accuracy in percent and Acc_1 for the shared one-draw accuracy, telescoping the changes along each image's allocated prefix gives

$$\mathrm{Acc}^a(B) - \mathrm{Acc}_1 = \sum_{k=1}^{K-1} G_k^a(B).$$

Consequently, the stagewise contrasts sum exactly to the final accuracy difference, including after averaging over orders and tasks.

For task d, draw order o, stage k, and budget B, let $C_{d,o,k}^a(B)$ and $D_{d,o,k}^a(B)$ count, respectively, incorrect-to-correct and correct-to-incorrect transitions among images continued under policy a. Each task's denominator is its complete query size N_d. Averaging over the 20 draw orders and equally across eight tasks gives the macro stage contrast

$$\Delta_k(B) = \frac{100}{8} \sum_{d=1}^{8} \frac{1}{20 N_d} \sum_{o=1}^{20} \left[(C_{d,o,k}^F(B) - D_{d,o,k}^F(B)) - (C_{d,o,k}^Q(B) - D_{d,o,k}^Q(B)) \right]. \tag{32}$$

Here F and Q retain the transferred-Full and native-Quad meanings defined above. Stage intervals resample paired image contributions on saved routes.

The advantage accumulates across stages, with its distribution varying by backbone and budget. Several stage intervals include zero, including all four for SD1.5 at $B = 3$.

C.4 Frozen readout transfer to SDXL on CIFAR-10

This extension evaluates SDXL at its native 1024 resolution on 320 fixed CIFAR-10 query images. Each image has five cached draws, each covering nine timesteps and all candidate classes. Results average over 20 draw orders. The metric is normalized accuracy–budget AUC across the same seven budgets $B \in \{1.5, 2, \ldots, 4.5\}$. Full and Quad trained on each donor's ImageNet losses give four transferred readouts; CIFAR-10 supplies only the target queries. Their weights, intercepts, and fitted feature normalization remain frozen. All four readouts and Margin use the preselected

Table 12: **Stage contributions on SD1.5.** $\Delta_k(B)$ is transferred Full minus native Quad under the shared receiver-selected schedule, in task-macro accuracy pp; intervals use 5,000 paired-image fixed-route replicates.

B	Stage	$\Delta_k(B)$ [95% interval]
2	$1 \to 2$	$+0.112$ $[-0.013, 0.239]$
2	$2 \to 3$	$+0.138$ $[0.022, 0.256]$
2	$3 \to 4$	$+0.047$ $[-0.049, 0.139]$
2	$4 \to 5$	$+0.033$ $[-0.039, 0.102]$
3	$1 \to 2$	$+0.034$ $[-0.079, 0.146]$
3	$2 \to 3$	$+0.032$ $[-0.042, 0.106]$
3	$3 \to 4$	$+0.015$ $[-0.066, 0.092]$
3	$4 \to 5$	$+0.006$ $[-0.092, 0.098]$

Table 13: **Stage contributions on SD2.0.** $\Delta_k(B)$ is transferred Full minus native Quad under the shared receiver-selected schedule, in task-macro accuracy pp; intervals use 5,000 paired-image fixed-route replicates.

B	Stage	$\Delta_k(B)$ [95% interval]
2	$1 \to 2$	$+0.211$ $[0.069, 0.347]$
2	$2 \to 3$	$+0.114$ $[-0.008, 0.235]$
2	$3 \to 4$	$+0.090$ $[-0.010, 0.190]$
2	$4 \to 5$	$+0.104$ $[0.007, 0.205]$
3	$1 \to 2$	$+0.101$ $[0.006, 0.203]$
3	$2 \to 3$	$+0.065$ $[-0.007, 0.137]$
3	$3 \to 4$	$+0.079$ $[0.003, 0.154]$
3	$4 \to 5$	$+0.062$ $[-0.028, 0.150]$

SD2.0/ImageNet Margin schedule. No SDXL source cache is constructed, and no receiver-side readout fitting or schedule selection is performed.

Table 14 gives mixed results across donor readouts. SD2.0/ImageNet Full exceeds Margin by 0.117 pp and its corresponding Quad by 0.159 pp, whereas SD1.5/ImageNet Full trails Margin by 0.234 pp and its corresponding Quad by 0.300 pp. Both Full readouts exceed Uniform-balanced, but their ordering against the stronger controls differs. These are descriptive point estimates from one target task under one preselected schedule.

Table 14: **Frozen ImageNet-readout transfer to SDXL on CIFAR-10.** Normalized accuracy–budget AUC (%) over seven budget points, averaged across 20 draw orders on 320 fixed target images. All readouts and Margin use the preselected SD2.0/ImageNet Margin schedule; Uniform-balanced matches total cost. Entries are point estimates, and all methods use the same FP32 mean-loss decision rule.

Method	AUC (%)
Uniform-balanced	22.526
Margin	23.526
SD1.5/ImageNet Quad → SDXL	23.592
SD1.5/ImageNet Full → SDXL	23.292
SD2.0/ImageNet Quad → SDXL	23.484
SD2.0/ImageNet Full → SDXL	**23.643**

Table 15: **Complete-policy accuracy across budgets.** Eight-task macro accuracy (%), averaged over 20 draw orders. Each complete target query set forms one allocation batch. Gap-only and Full (own schedule) use schedules selected with their source OOF scores rather than Margin priorities. PETS-OKG (adapt.) uses a source-calibrated total prior concentration.

Backbone	B	Uniform-balanced	Margin	Gap-only	Full (own schedule)	PETS-OKG (adapt.)
SD1.5	1.5	45.091	47.282	45.865	46.315	44.523
	2.0	48.162	49.756	49.796	49.610	47.076
	2.5	49.522	51.176	51.384	51.722	49.475
	3.0	51.038	52.431	52.637	52.934	51.383
	3.5	51.835	53.331	53.319	53.567	52.763
	4.0	52.606	53.629	53.607	53.686	53.518
	4.5	53.164	53.781	53.747	53.780	53.723
SD2.0	1.5	49.957	52.348	52.200	52.896	49.447
	2.0	53.205	55.032	54.825	55.444	52.275
	2.5	54.719	56.629	56.592	57.189	54.992
	3.0	56.262	57.750	57.756	58.333	56.919
	3.5	57.028	58.638	58.518	58.796	58.203
	4.0	57.847	58.845	58.830	59.049	58.754
	4.5	58.397	59.001	58.998	59.041	58.949

Table 16: **Macro accuracy (%) of the fixed-draw classifier.** Each column uses exactly k draws per image, with maximum $K = 5$.

Backbone	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
SD1.5	41.962	48.162	51.038	52.606	53.793
SD2.0	46.683	53.205	56.262	57.847	58.989

D COMPLETE-POLICY COMPARISONS

This section examines readout-dependent schedule selection and gives implementation details for the PETS-OKG and Blend-ASC allocation comparators. Gap-only is defined in Appendix A.7. The Gap-only and Full (own schedule) variants select schedules using their respective source OOF scores. The main transferred Full policy uses the donor's Margin-selected schedule, as described in Appendix B.

D.1 ACCURACY ACROSS BUDGETS

Table 15 reports all seven budgets, with each complete target query set forming one allocation batch. Uniform-balanced follows Appendix B. PETS-OKG (adapt.) denotes the calibrated-prior, unweighted adaptation used in the main comparison. Its seven budget accuracies integrate to the macro AUCs in Table 2.

Table 16 reports the underlying fixed-draw classifier. The five-draw prediction supplies the finite reference for disagreement.

D.2 EFFECT OF SCHEDULE SELECTION

On SD1.5 at $B = 1.5$, Margin selects $q = (0.4, 0.1, 0, 0)$, whereas Full (own schedule) selects $q = (0.2, 0.1, 0.1, 0.1)$; each fraction refers to the entire batch. Replacing only Full's own schedule with Margin's increases macro accuracy by 1.355 points. Full with its own schedule gains 2.958, 0.791, 0.402, and 0.202 accuracy points at its four continuation stages; Margin gains 4.439, 0.880, 0, and 0 points. Each contribution is normalized by the complete query size within its task, then averaged over draw orders and equally across tasks, as in Eq. (32). The positive deeper-stage gains do not compensate for the reduction in second-draw coverage.

On the same backbone at $B = 3$, replacing Margin's ranker with Full under the same schedule adds 0.341 accuracy points, while adopting Full's own schedule adds a further 0.162 points. The schedule effect therefore depends on the available budget. These comparisons show how readout-dependent schedule selection changes the distribution of sampling effort across stages.

D.3 PETS ADAPTATION AND SOURCE CALIBRATION

The PETS comparator adapts unweighted offline OKG (Liu et al., 2026b), retaining its optimistic posterior-gain criterion, sequential image selection, and Dirichlet updates. Posterior integration is evaluated numerically rather than with Monte Carlo estimates. Every image receives one charged initial draw and has a five-draw cap. Following the preliminary evaluation, we perform a retrospective source calibration of the total prior concentration $\eta \in \{0.1, 1, 10, 100, 1000\}$. Both backbones select $\eta = 1$, giving symmetric target-class Dirichlet parameters $\alpha_c = \eta/C = 1/C$. At $B = 4$, calibration raises PETS-OKG accuracy from 52.415% to 53.518% on SD1.5 and from 57.335% to 58.754% on SD2.0. Allocation follows the sequential OKG rule, while final labels are computed from allocated losses using the original FP32 mean-loss rule, matching TAPS and the other controls.

D.4 BLEND-ASC ADAPTATION TO THE FIVE-DRAW BUDGET

Blend-ASC (Feng et al., 2026) jointly allocates samples across inputs by combining rankings from adaptive consistency and mode estimation confidence scores. We evaluate a five-draw adaptation under the same total draw budgets as TAPS. It uses its own allocation rule and returns labels using the same FP32 mean-loss classifier as every main-study method. The resulting macro AUCs are 49.813% on SD1.5 and 55.076% on SD2.0.