WHEN DOES NEURAL WEIGHT GENERATION WORK? SUPPORT, COORDINATES, AND EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural weight generation promises to reuse trained checkpoints, but when does it produce reliable samples rather than a strong best result? We study performance-weighted diffusion on CIFAR-10 CNN and ViT model zoos, varying checkpoint support, generation coordinates, and denoiser capacity. At fixed checkpoint count, replacing one trajectory with three independent runs removes 79–90% of generated accuracy gain above chance under raw-weight coefficient diffusion, despite comparable aggregate denoising loss. Dimensional sweeps expose a normalization artifact that suppresses off-span perturbations, allowing functional outputs despite loss near the zero-predictor baseline. Greater capacity improves higher-dimensional generation, but unequal denoiser fit and exhausted training budgets leave an independent dimensionality penalty unresolved. Whitened single-trajectory span diffusion is the strongest learned custom-zoo generator evaluated, although simple baselines remain competitive and steering generally weakens with parameter-space novelty. On aligned public-zoo supports spanning hundreds of runs, full-latent U-Net diffusion recovers TiltDiff's reported accuracy scale using our own pipeline, with condition-mean Top-1@108 of 0.593–0.598 across 18 fresh-seed confirmation runs. However, distance-matched noise achieves higher mean accuracy in every run, while exponential weighting improves the upper tail at the cost of lower mean accuracy and yield. These results identify pipeline-specific sensitivities and show why weight-generation evaluation must assess typical-sample quality alongside best-of-samples performance.

1 INTRODUCTION

Generating neural-network weights could reuse computation already spent training existing models. This direction builds on hypernetworks (Ha et al., 2017; Knyazev et al., 2021; 2023), weight-space representations (Schürholt et al., 2021; Navon et al., 2023; Zhou et al., 2023), and generative models of neural parameters (Peebles et al., 2022; Schürholt et al., 2022a; Wang et al., 2024; Erkoç et al., 2023). Yet a generator's training collection may itself be expensive: independent runs require repeated optimization, whereas intermediate checkpoints are inexpensive to save. A collection with many checkpoints can therefore contain little independent variation. We ask: *what can a weight generator learn from limited trajectory support, and what constitutes successful generation?*

Three issues make this question harder than checkpoint count suggests. First, *support composition determines geometry*. Checkpoints from one run trace a correlated path; independent runs introduce variation between solutions whose connecting directions need not preserve network function. Increasing the number of source runs at a fixed checkpoint budget may broaden support while making it harder to model. Second, *dimensional comparisons depend on coordinates and achieved denoiser fit*. Parameter space, codec latents, and principal coordinates assign different scales to diffusion perturbations. Poor fitting can resemble a dimensionality penalty, while normalization can suppress unsupported directions and preserve accuracy despite high aggregate denoising loss. Third, *a strong best sample does not establish reliable generation*. An improved upper tail may coexist with lower average accuracy or fewer useful samples. Retrieval, interpolation, and weight averaging provide essential points of comparison, and distance from training checkpoints alone does not establish behavioral diversity.

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.

We study these issues through performance-weighted diffusion based on TiltDiff (Chuang et al., 2026). Custom CIFAR-10 CNN and ViT zoos allow controlled comparisons of support composition, alignment, coordinate scaling, and denoiser capacity. A larger public CNN zoo tests whether conclusions from coefficient diffusion persist under a different, full-latent U-Net pipeline. We evaluate typical accuracy, functional yield, best-of-samples performance, parameter-space novelty, and accuracy steering. Our contributions are threefold:

1. **We quantify support-composition effects at fixed checkpoint count.** Under raw-weight coefficient diffusion, replacing one trajectory with three independent runs removes 79–90% of generated accuracy gain above chance despite comparable aggregate denoising loss. The leading $M-1$ principal directions explain at least 90% of variance in the evaluated multi-run supports. These support constructions also vary run selection and temporal density, so the comparison does not isolate run count alone. Matched public-zoo controls corroborate the composition effect; alignment restores functionality more readily than high accuracy in the tested span pipeline.
2. **We identify normalization and fitting confounds in dimensional sweeps.** Per-coordinate normalization suppresses off-span perturbations, allowing functional outputs at losses near the zero-predictor baseline. Wider denoisers improve generation at higher dimension, but unequal achieved fit and exhausted training budgets leave an independent dimensionality penalty unresolved.
3. **We distinguish upper-tail gains from typical-sample quality.** Aligned public-zoo latent U-Net diffusion recovers TiltDiff's reported accuracy scale across 18 fresh-seed runs, with condition-mean Top-1@108 of 0.593–0.598. Exponential weighting improves the upper tail while lowering mean accuracy and yield; distance-matched noise achieves higher mean accuracy in every run. On custom-zoo supports, whitened single-trajectory span diffusion is the strongest learned generator tested, but simple baselines remain competitive, and steering generally weakens with parameter-space novelty.

The conclusions concern these CIFAR-10 supports and pipelines. The public-zoo U-Net establishes that aligned multi-run support can produce high-accuracy models, while changing several components relative to the span MLP; it is neither a capacity-only ablation nor a complete TiltDiff reproduction. Its fresh generator seeds also share retention pools and test images, following test-guided configuration selection. Together, the experiments show why support construction, coordinate scaling, achieved fit, and both typical and upper-tail performance must be reported to interpret a weight generator's success.

2 Study Design and Evaluation

We vary checkpoint support, representation, coordinates, and denoiser capacity, evaluating accuracy, yield, upper-tail performance, novelty, and steering. Appendix B gives implementation details.

Data and model zoo. The custom CIFAR-10 zoo uses 5,000 validation images for selection, conditioning, and development. The 10,000-image test split was opened only for the preregistered evaluation in Section 3.3; it does not independently validate other comparisons. The zoo contains 216 checkpoints: ConvNet and CIFAR-ViT at three scales, nine runs per cell, and four checkpoints per run. Representations and generators use the four tiny/small cells; medium models are reserved for scale transfer. Each cell's validation-best final run defines its reference trajectory. Nine-stage and 320-stage acquisitions support steering and composition studies, including dense re-executions of the second- and third-ranked runs. Shared checkpoints reproduce tensor-identically. Appendix C specifies architectures; Section 4 describes the public zoo.

Support composition. At $N = 36$ checkpoints, we compare $M \in \{1, 3, 9\}$ runs with $T = 36/M$ stages per run over updates 8,800–35,200, excluding nine shared targets. The arms select 36 evenly spaced reference-run stages, the same 12 update indices from each top-three run, or all four base-zoo checkpoints from every run. A one-run control covers the full trajectory. These constructions also vary source-run selection and temporal density. Each arm fits its own $K = 35$ principal basis, conditioning statistics, and novelty reference in role-standardized raw weights, without a codec. The aligned three-run control matches final checkpoints to the reference and applies each permutation throughout its run.

Representations and generators. The structural codec encodes standardized weight chunks with metadata, losing at most one percentage point of round-trip accuracy on training cells. Nine-stage steering, dense-frontier, and dimensional experiments use its latents; fixed-N composition uses raw weights. With K retained principal directions and generation dimension d, span generation uses $d = K$ and empirically whitened coefficients unless stated otherwise. Dimensional sweeps add off-span directions and compare global scaling with per-coordinate normalization.

Coefficient diffusion uses accuracy-conditioned MLP denoisers and performance-weighted loss following Chuang et al. (2026); separate latent-diffusion controls operate directly on codec latents. Baselines comprise retrieval, interpolation, coefficient-box sampling, perturbation, and weight averaging. Public-zoo experiments compare MLP controls with full-latent U-Net diffusion using a separate $1\times$ per-chunk codec. The U-Net is unconditional: performance enters through uniform, sigmoid, or exponential training-loss weights. Weighting comparisons fix architecture and budgets; MLP–U-Net comparisons change multiple components.

Steering and target exclusion. Nine-stage leave-one-out refits the basis, conditioning statistics, and generator on eight stages per fold. The dense frontier excludes nine targets jointly, fitting once per cell on $N = 311$ checkpoints with $K = 310$. Fixed-N arms use a separate shared set of nine targets and their own 36-checkpoint supports. Target validation accuracies supply requests; jointly excluded targets share a generator fit.

Exclusion is at the generator level under shared preprocessing: the structural codec saw all nine sparse targets and one dense target; fixed-N and per-chunk-control role statistics also include targets. Downstream support-derived quantities use retained checkpoints only. These protocols therefore do not establish whole-pipeline generalization to unseen checkpoints. Steering MAE and Spearman correlation use finite samples only; finite-output fractions are reported separately.

Accuracy and novelty. Evaluation uses the full relevant image split. Unconditional mean accuracy assigns 0.10 to non-finite predictions and public-zoo loading failures; the archived dimensional table is a finite-only exception. Yield is the fraction of all samples strictly above 0.40, with invalid outputs counted as failures, including in Appendix D. Public-CNN functionality additionally requires successful loading, finite predictions, and accuracy above $\tau_{\text{func}} \approx 0.141$, calibrated on 32 fresh initializations.

Novelty divides standardized weight distance to the nearest training checkpoint by a reference set's median leave-one-out nearest-neighbor distance. Fixed-N arms use their own supports. The dense frontier uses all 311 checkpoints in the numerator and either the full support or a nine-point subsample in the denominator. Novelty one denotes typical reference separation, not behavioral diversity. Defined distances remain included despite non-finite predictions; public-zoo norm filtering affects novelty only.

Sampling and uncertainty. Main coefficient comparisons use 5 generator seeds and 36 samples per seed; wider-trunk comparisons use 3 seeds. The frontier uses one seed per cell across nine targets. Other counts and stopping rules appear in Appendix B; reaching the training ceiling does not establish convergence.

Public-zoo U-Net selection inspected test accuracy. Confirmation freezes the configuration and uses three fresh seeds for two nested pools and three weighting schemes: 18 runs of 1,024 samples. Top-1@108 estimates the expected maximum test accuracy over random 108-sample subsets drawn without replacement within each run. Validation-selected accuracy evaluates the validation-best finite candidate among all 1,024 samples. Controls use retrieval and Gaussian perturbations calibrated to each run's median nearest-support distance. Validation selection does not undo test-guided configuration selection. Across-seed summaries quantify generator variability under fixed supports and shared images, not uncertainty over supports or datasets.

3 Support Geometry and Weight Generation

We examine support composition, coordinates, and denoiser fit in the four custom-zoo training cells, then assess generation through baselines, held-out image evaluation, and steering.

Support	CNN-S	CNN-T	ViT-S	ViT-T
$M = 1, T = 36$	**0.734**	**0.581**	**0.626**	**0.530**
$M = 3, T = 12$	0.165	0.153	0.191	0.188
$M = 3, T = 12$, aligned	0.231	0.268	0.174	0.201
$M = 9, T = 4$	0.117	0.111	0.137	0.178

Table 1: **Support composition at fixed $N = 36$.** Raw-weight coefficient diffusion uses M source runs and T checkpoints per run. Entries are mean unconditional validation accuracies over five generator seeds, with 36 samples per seed; invalid outputs receive 0.10. Bold marks the highest mean per column. S/T denote small/tiny. Table 4 reports across-seed variation; Appendix B details support selection and alignment.

3.1 Support Composition

Dense reference trajectories concentrate variance in codec space: two principal directions explain 90% in every cell, and 5–7 explain 99% except in VIT-SMALL, which requires 53. Tested adjacent-stage interpolants vary monotonically in accuracy; trajectory means and sampled convex combinations remain functional. Some longer chords reduce accuracy, so this concentration does not imply a uniformly high-performing convex hull.

Within-run midpoints retain approximately endpoint-level accuracy, whereas cross-seed midpoints achieve 0.13–0.20 and multi-run means approach chance. Weight, activation, and whole-head matching improve some CNN interpolants without restoring endpoint accuracy; ViT midpoints remain near chance. These linear-path results do not exclude nonlinear connectivity.

At fixed $N = 36$, raw-weight coefficient diffusion uses $M \in \{1, 3, 9\}$ runs with $T = 36/M$ stages per run (Section 2). For multi-run supports, the leading $M - 1$ directions explain at least 90% of variance in every cell. This is consistent with dominant between-run displacement: the centered means of M runs span at most $M - 1$ directions.

Replacing one-run support with three-run support removes 79–90% of generated accuracy gain above chance despite comparable aggregate weighted denoising loss (Table 1). Alignment improves three-run generation in three cells, but all remain far below one-run results. Source-run selection and temporal density also vary, so the comparison measures the specified support constructions rather than an isolated causal effect of run count.

Signed affine coefficients provide a complementary, method-dependent association between concentration on one source run and accuracy (Appendix D). These findings concern the tested raw-weight pipeline; the public-zoo U-Net demonstrates successful multi-run generation (Section 4).

3.2 Dimension, Capacity, and Normalization

Structural-codec latent sweeps use randomly rotated frames with a common root-mean-square scale recomputed at each dimension (Figure 1). Standard and wider trunks use five and three seeds. All wider-trunk runs at $d = 310$ and $d = 1024$ reach the 30,000-update ceiling, and some do not meet the plateau criterion; achieved denoiser fit is therefore unmatched.

At $d = 1024$, widening the trunk from 512 to 2,048 raises accuracy from $0.100/0.101/0.152/0.122$ to $0.597/0.409/0.629/0.488$ (CNN-S, CNN-T, ViT-S, ViT-T) and reduces loss from about 0.66 to 0.10–0.11. Accuracy approaches or exceeds that of the standard trunk at $d = 310$. Within the wider trunk, increasing d from 310 to 1024 lowers CNN accuracy by 0.067–0.092, while ViT accuracy is stable or improves; loss rises from 0.04–0.05 to 0.10–0.11 (Table 5). Capacity mitigates the decline, but unequal fit, scaling, and budget preclude isolating a dimensionality penalty.

Per-coordinate normalization creates a separate artifact. In a span-aligned frame, off-span directions have zero empirical variance; the 10^{-6} standard-deviation floor scales their generated components by 10^{-6} before decoding. Models retain predictive accuracy at loss near 1.0, the zero-predictor baseline, with orthogonal residual approximately 10^{-4}. Functionality can therefore partly reflect the reconstruction map. No literal zero-denoiser sampling control was run, and near-baseline aggregate loss does not exclude useful denoising within the support span.

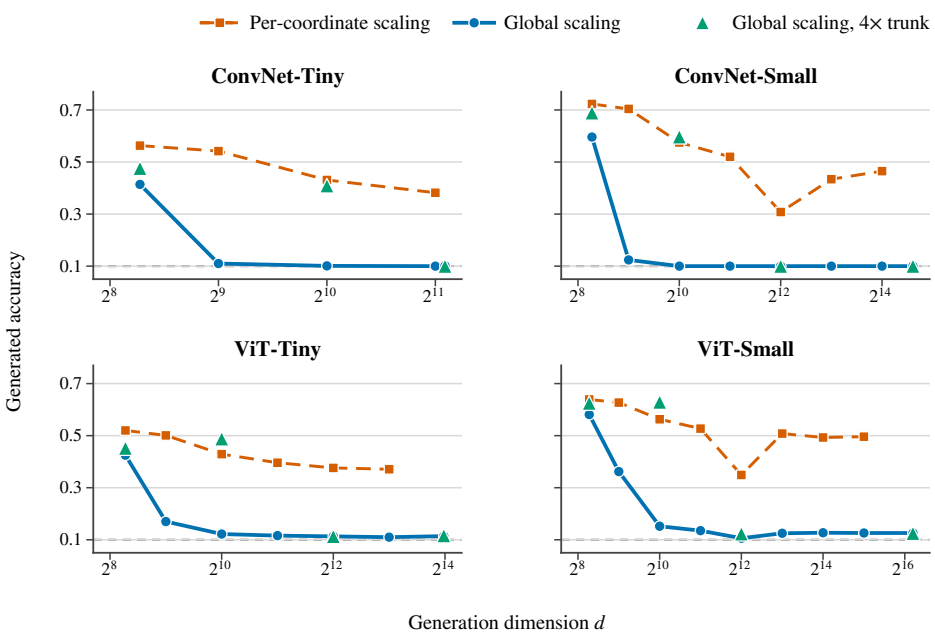

Figure 1: **Dimension, capacity, and normalization.** A $4\times$ wider trunk (triangles) improves globally scaled codec-latent generation at $d = 1024$ over the standard trunk (solid lines), but fit and budget confound dimensional comparisons. Per-coordinate normalization (dashed) suppresses off-span noise, preserving accuracy at near-baseline loss. Gray marks chance (0.10).

3.3 Whitened Span Generation

On the $N = 311$ single-trajectory codec-latent support ($d = K = 310$), whitened diffusion achieves $0.723/0.563/0.639/0.520$ accuracy, versus $0.596/0.414/0.581/0.425$ under global scaling and $0.689/0.476/0.625/0.452$ with global scaling and a $4\times$ wider trunk. Whitening rescales noise relative to observed coordinate variation; losses across these scalings are not directly comparable.

A separate fixed-architecture control achieves 0.721–0.731 accuracy using raw-weight PCA on 36 checkpoints ($K = 35$), exceeding the learned representations tested there. Generated weights nevertheless remain close to support, limiting claims about compression or new functional regions.

Following Zeng et al. (2026), we compare baselines under matched support, standardization, and novelty calibration. Retrieving the best checkpoint gives slightly higher accuracy at zero novelty. Averaging ten checkpoints comes within 0.002–0.012 of the generator's accuracy and yields greater novelty in three cells. No tested perturbation or averaging configuration exceeds it in both metrics, but its accuracy advantage over averaging is small.

We specified a separate evaluation before opening the 10,000-image test split. Regenerated models reproduce development accuracy; test–validation differences are at most 0.009, averaging 0.005 in absolute value. This supports held-out image accuracy for the selected models, not a held-out advantage over baselines or an independently held-out representation.

3.4 Novelty and Steering

Figure 2 evaluates nine jointly excluded targets using one generator fit on 311 checkpoints per cell. Steering metrics use finite samples and retain the target-exposed preprocessing of Section 2.

Interpolation provides the strongest steering within the sparse-reference distance scale. Added noise generally increases novelty and weakens steering; diffusion produces more distant samples with weaker control, and box sampling gives still greater novelty but little control. With the dense reference, lightly perturbed interpolants can exceed one distance unit while retaining strong steering in three

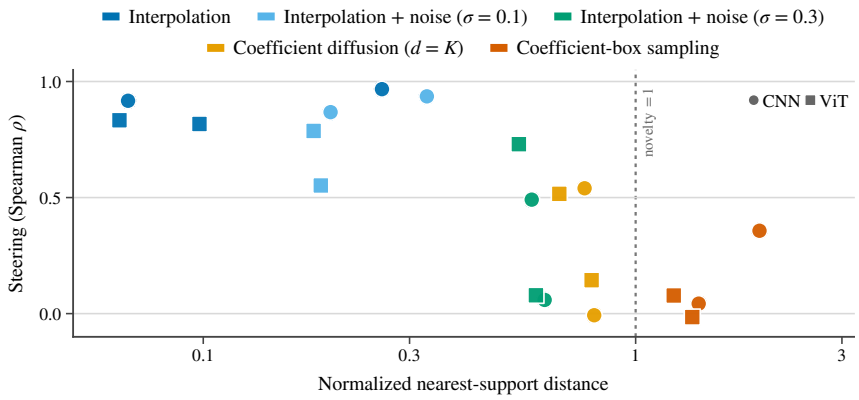

Figure 2: **Novelty and performance steering.** Nearest-support distance uses all 311 checkpoints, normalized by the median leave-one-out distance in a nine-point subsample. Spearman correlation pools finite samples across nine requests. Points denote method–architecture–scale combinations (circles: CNNs; squares: ViTs), using one generator seed per cell. The dotted line marks novelty one.

cells. Thus, the trade-off provides neither a universal novelty threshold nor evidence of behavioral diversity.

Conditional interpolation yields MAE 0.007–0.031 in nine-stage leave-one-out and 0.002–0.008 under dense exclusion, with Spearman correlations 0.82–0.97. Nearest-accuracy retrieval is comparable, and raw-weight interpolation matches the codec route. Matching a requested accuracy thus need not recover the excluded checkpoint or reach an unobserved functional region.

Scope across scales. On held-out medium models, codec reconstruction produces non-finite ConvNet predictions and chance-level ViT accuracy; two codec-free mappings also fail. These failures delimit the tested representations, not cross-scale generation generally (Appendix A).

4 Public-Zoo Generation and Evaluation

We extend the study to the public CIFAR-10 zoo of Schürholt et al. (2022b): 1,000 independently seeded runs of a 10,853-parameter CNN, with 51 checkpoints per run. We first test support composition and alignment with raw-weight coefficient diffusion, then evaluate full-latent U-Net diffusion under TiltDiff's checkpoint-retention protocol (Chuang et al., 2026). This separates controlled comparisons within a pipeline from changes across pipelines (Appendix B).

Representations and evaluation. Weights are flattened in a fixed order and standardized by parameter role using pool statistics. Initial MLP controls use raw weights or our per-chunk Transformer codec, with span or full-coordinate diffusion and coordinate standard deviations floored at 10^{-6}. The subsequent U-Net uses the full $1\times$ codec latent with global scaling. It is unconditional: performance enters through training-loss weights, not sampling-time accuracy requests.

Evaluation uses the zoo's original test split. Invalid outputs receive chance accuracy (0.10); functionality requires successful loading, finite predictions, and accuracy above $\tau_{\text{func}} \approx 0.141$. For each U-Net run, Top-1@108 estimates the expected maximum test accuracy over 108-model subsets drawn without replacement from 1,024 generated candidates. Validation-selected accuracy instead evaluates the validation-best finite candidate from the full generated pool. These metrics use different selection rules and candidate budgets.

4.1 Support and Alignment

We first use performance-weighted MLP diffusion in the whitened span of standardized raw weights, without a codec or alignment. At fixed $N = 36$, support comprises epochs 15–50 of one run or final

checkpoints from 36 runs; a further condition uses final checkpoints from 51 runs. Each condition produces 108 samples across three generator seeds.

Single-trajectory support yields mean accuracy 0.601, with every sample exceeding 0.40. The functional fraction falls to 45.4% with 36 runs and 25.9% with 51. The fixed-N comparison extends the support-composition result of Section 3 to this zoo under raw-weight coefficient diffusion; it does not isolate run count from the change in stage coverage.

With the representation and generator fixed, Git Re-Basin matches each run's epoch-50 checkpoint to a reference and applies the permutation to its retained stages. Four checks verify function preservation and recovery of planted permutations. Alignment raises functionality to 99.1% for both multi-run supports, but yields above 0.40 remain 13.0% and 1.8%, respectively. In this pipeline, alignment restores above-initialization performance more readily than high accuracy. These controls cannot separate imperfect matching, residual between-run variation, and generator limitations.

4.2 Generation from Retained Checkpoints

Following TiltDiff, we retain the top 10, 20, or 30% of checkpoints by validation accuracy from epochs 21–25 of all 1,000 runs. The pools contain 500, 1,000, or 1,500 checkpoints from 227, 379, or 520 runs, respectively. All use the alignment procedure above; epoch-50 checkpoints supply auxiliary alignment information outside the retained epochs.

Initial MLP controls. Across 13 aligned raw-weight and codec-latent conditions with span or full-coordinate MLP diffusion, no sample exceeds 0.40 accuracy. On top-10%, the $1\times$ codec with whitened latent-span diffusion reaches best test accuracy 0.264 among 108 samples pooled across three seeds, versus 0.240 for raw-weight span diffusion. Poor generation therefore persists without a codec in these configurations.

Full-latent U-Net. The U-Net configuration was selected post hoc after inspecting test accuracy on top-10%. We then freeze the configuration and evaluate top-20% and top-30% with uniform, sigmoid, and exponential loss weighting, using three fresh generator seeds per condition. These 18 runs share fixed training and sampling budgets; within each pool, the weighting comparison uses the same codec and denoiser architecture. Mean codec round-trip accuracy loss over all retained checkpoints is 0.0007 and 0.0008 for the two pools.

All 18 runs achieve Top-1@108 within TiltDiff's reported range, with condition means of 0.593–0.598. Aligned supports containing hundreds of runs can therefore yield high-accuracy samples with this latent U-Net. The result qualifies the MLP failures but does not identify their cause: representation, normalization, architecture, conditioning, and optimization change together. We recover the reported accuracy scale using our own pipeline; this is not a complete reproduction, and the unspecified original sampling budget prevents a budget-matched comparison. Moreover, the nested pools and shared test images make these fresh-seed confirmations of a frozen configuration, not evaluation on an untouched test set.

4.3 Upper-Tail Gains and Typical-Sample Quality

Table 2 shows that upper-tail and typical-sample criteria give different assessments. Uniform weighting produces mean accuracy 0.549 and 0.530, with yields of 94.1% and 91.0% on top-20% and top-30%. Across all 18 runs, Gaussian checkpoint perturbations scaled to match each generator run's median nearest-support distance achieve higher mean accuracy, while diffusion achieves higher Top-1@108. This baseline matches an aggregate distance, not the full distance distribution (Appendix B.4).

Exponential weighting improves seed-mean Top-1@108 over uniform by 0.0023 and 0.0045, and exceeds retrieval Top-1@108 in all six corresponding runs. However, mean accuracy falls to 0.476 and 0.413, and yield drops by 17.2 and 27.6 percentage points. The top-30% gain refutes the frozen prediction that improvement would not exceed 0.003. Exponential tilting therefore gives a small, repeated upper-tail benefit alongside substantially worse typical-sample quality under these settings.

Under exponential weighting, validation-selected models beat the pool's validation-best checkpoint in all six runs. Test-guided configuration selection and shared images limit inference across image

Pool	Weighting	Generator mean	Noise mean	Yield	Top-1@108	Val.-selected @1024
20%	Uniform	0.549	0.573	94.1%	0.5957 ± 0.0002	0.596
20%	Sigmoid	0.429	0.558	68.2%	0.5954 ± 0.0011	0.590
20%	Exponential	0.476	0.563	76.9%	0.5980 ± 0.0012	0.599
30%	Uniform	0.530	0.569	91.0%	0.5930 ± 0.0013	0.592
30%	Sigmoid	0.397	0.545	61.3%	0.5935 ± 0.0017	0.594
30%	Exponential	0.413	0.548	63.4%	0.5975 ± 0.0013	0.600

Table 2: **Upper-tail performance and typical-sample quality.** Public-zoo results averaged over three fresh generator seeds. Noise is calibrated separately for each run to match its median nearest-support distance (Appendix B.4). All other metrics describe the U-Net generator. Mean accuracy and yield above 0.40 include invalid samples as specified in Section 2. Top-1@108 reports the within-run expected test-ranked maximum, with across-seed SD; validation selection uses finite candidates among 1,024 generated models. The validation-best pool checkpoint achieves 0.5961 test accuracy in both pools. Seeds share evaluation images, and test accuracy informed earlier configuration selection.

populations. Accuracy decreases with nearest-support distance in all 18 runs, with Spearman correlations from -0.94 to -0.64. The association is consistent with better samples remaining closer to support; it does not establish copying or an absence of behavioral diversity.

Representation checks. Separate custom-zoo controls compare raw weights with per-chunk codec latents in the whitened $K = 35$ span of one-run, 36-checkpoint supports. Changing codec compression has modest effects in three cells, while the fourth varies substantially across codec seeds. Two codec conditions with MSE 1.31×10^{-4} and 1.35×10^{-4} nevertheless differ by an order of magnitude in round-trip accuracy loss. Weight MSE alone therefore does not establish functional fidelity. Separate SANE tokenization and reconstruction audits appear in Appendix B; no SANE-based generator is evaluated, so those audits do not explain either generation pipeline's results.

5 Related Work

Neural-weight generation. Hypernetworks predict parameters from conditioning variables (Ha et al., 2017), with graph-based variants extending prediction to unseen architectures (Knyazev et al., 2021; 2023). Model-zoo methods instead learn representations of trained checkpoints for downstream prediction and generative sampling (Schürholt et al., 2022a); SANE extends this approach through sequential encoding of weight tokens (Schürholt et al., 2024). Diffusion methods predict parameter updates (Peebles et al., 2022), generate autoencoded weights (Wang et al., 2024), and synthesize neural implicit fields (Erkoç et al., 2023). D2NWG conditions latent diffusion on dataset embeddings for task transfer (Soro et al., 2025), while RPG uses recurrent token modeling to condition diffusion at larger parameter scales (Wang et al., 2025). DeepWeightFlow combines permutation-aware canonicalization with flow matching directly in weight space (Gupta et al., 2026). Our study examines how checkpoint support, generation coordinates, and denoiser fitting affect the behavior of specific pipelines.

Weight-space geometry. Low-loss nonlinear paths can connect solutions separated by linear interpolation barriers (Garipov et al., 2018), and linear connectivity depends on stability to SGD noise (Frankle et al., 2020). Permutation symmetries account for part of this separation (Entezari et al., 2022), motivating weight matching, optimal-transport alignment, and activation repair (Ainsworth et al., 2023; Singh & Jaggi, 2020; Jordan et al., 2023). Model soups show that averaging models fine-tuned from a shared initialization can improve accuracy (Wortsman et al., 2022). Successful optimization in restricted subspaces (Li et al., 2018) and gradient concentration in leading Hessian eigenspaces (Gur-Ari et al., 2018) provide further evidence of structure below ambient parameter dimension. We test the usefulness of support-derived coordinates for generation, controlling support composition, alignment, and normalization. These empirical checkpoint spans need not coincide with the optimization subspaces studied in prior work.

Novelty and memorization. Zeng et al. (2026) report that evaluated weight generators often produce approximate copies or interpolations of training checkpoints, with perturbation and weight-

averaging baselines matching or exceeding their accuracy–novelty trade-offs. Extraction and replication have also been documented in image diffusion models (Carlini et al., 2023; Somepalli et al., 2023). We relate nearest-support distance to accuracy and performance steering, using retrieval, interpolation, and perturbation controls. Parameter-space distance measures separation from reference checkpoints; it does not by itself establish memorization or behavioral diversity.

Relation to TiltDiff. TiltDiff combines learned weight representations with latent diffusion and performance-weighted denoising (Chuang et al., 2026). We adopt its denoising objective and use its public-zoo checkpoint-retention protocol to examine support geometry and evaluation criteria. Our full-latent U-Net recovers the reported accuracy scale using our own codec and implementation, whereas the tested MLP configurations perform poorly (Section 4). This contrast limits conclusions drawn from any one pipeline. Controlled weighting comparisons distinguish effects on mean accuracy and yield from effects on best-of-samples performance. These experiments provide protocol-level controls rather than a complete reproduction of TiltDiff; its unspecified sampling budget also precludes a budget-matched cross-paper comparison.

6 Limitations

Our conclusions concern the evaluated pipelines. Most controlled experiments use MLP coefficient diffusion; the public-zoo U-Net succeeds on aligned multi-run support with a different configuration. Dimensional sweeps do not match achieved denoiser fit, and wider trunks reach the training ceiling. An independent dimensionality penalty remains unresolved. Without another learned generator tested on identical supports, we cannot establish a general barrier to sparse-support generation or rank diffusion against alternative methods.

The U-Net uses our codec and implementation under TiltDiff's retention protocol. Its original sampling budget is unspecified, precluding a budget-matched comparison. U-Net configuration selection inspected test accuracy, and fresh-seed runs reuse those images; Top-1@108 is test-ranked, whereas validation-selected accuracy uses a separate selection rule. Across-seed variation is conditional on fixed pools and images. The SANE audit evaluates reconstruction, including on checkpoints from unseen runs, but does not test a SANE-based generator.

All experiments use CIFAR-10 and a limited set of architectures. Custom reference runs are validation-selected, the novelty–steering frontier uses one generator seed, and sealed testing covers only the selected positive result. Checkpoints are excluded at the generator level; codec training and standardization are not uniformly target-disjoint. Most custom-zoo comparisons use validation images also used for conditioning and development. Longer or branched trajectories, other scale-transfer methods, and substantially wider target networks trained at full budget remain unexplored. Parameter-space distance alone does not establish useful behavioral diversity.

7 Conclusion

Our study shows that neural weight generation depends on support construction, coordinates, denoiser fit, and the criterion used to define success. At fixed checkpoint count, mixing independent runs sharply reduces accuracy in the tested raw-weight coefficient-diffusion pipeline despite comparable aggregate denoising loss. Dimensional sweeps expose normalization and fitting confounds, leaving an independent dimensionality penalty unresolved. Whitened single-trajectory span diffusion performs best among the learned custom-zoo generators evaluated, but simple baselines remain competitive and steering generally weakens with parameter-space novelty. An aligned public-zoo U-Net nevertheless recovers TiltDiff's reported accuracy scale, showing that multi-run support can yield high-accuracy models under a different pipeline. Exponential weighting improves its upper tail while lowering mean accuracy and yield; distance-matched noise achieves higher mean accuracy in every confirmatory run. These findings support evaluating weight generators through explicit support and coordinate controls, functional representation checks, and simple baselines, while assessing typical-sample quality and controllability alongside best-of-samples performance.

AI USE STATEMENT

In preparing this work, the authors used OpenAI and Anthropic language models to assist with manuscript editing and reference checking, figure and table preparation, experiment scripts, and supplementary documentation. The tools also provided feedback on experimental design, presentation, and interpretation of results. The authors reviewed and revised the assisted work, checked references and reported results against their source records, and take full responsibility for the methods, data, claims, and final manuscript.

ETHICS STATEMENT

This work studies neural weight generation using CIFAR-10 and model checkpoints from our training runs and a public model zoo. It does not involve human-subject experiments or the collection of new personal data. Generated models are evaluated in a controlled research setting and are not deployed in operational systems. We report unsuccessful and non-finite outputs alongside positive results and distinguish predictive accuracy from weight-space novelty. We identify no additional ethical concerns specific to the experiments conducted here beyond those associated with reusing datasets and pretrained models.

REPRODUCIBILITY STATEMENT

Section 2 describes the study design, data splits, and evaluation metrics. Appendix C specifies the custom architectures, and Appendix B details checkpoint acquisition, support construction, alignment, normalization, codec and generator configurations, training budgets, seeds, and baseline procedures. Sections 3 and 4 identify the pipelines and controls used in each comparison. We distinguish generator-level checkpoint exclusion from representation exposure, validation-based development from sealed test evaluation, and fresh-seed confirmation from independent image evaluation.

REFERENCES

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *USENIX Security Symposium*, 2023.

En-Ni Chuang, Hanjuan Huang, Hao-Jia Song, Hsing-Kuo Pao, and Tyng-Luh Liu. TiltDiff: Tilted weight-space diffusion for neural network generation. In *European Conference on Computer Vision (ECCV)*, 2026.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. HyperDiffusion: Generating implicit neural fields with weight-space diffusion. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Saumya Gupta, Scott Biggs, Moritz Laber, Zohair Shafi, Robin Walters, and Ayan Paul. Deep-WeightFlow: Re-basined flow matching for generating neural network weights. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2601.05052.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2017.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: REnormalizing permuted activations for interpolation repair. In *International Conference on Learning Representations (ICLR)*, 2023.

Boris Knyazev, Michal Drozdzal, Graham W. Taylor, and Adriana Romero-Soriano. Parameter prediction for unseen deep architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Boris Knyazev, Doha Hwang, and Simon Lacoste-Julien. Can we scale transformers to predict parameters of diverse ImageNet models? In *International Conference on Machine Learning (ICML)*, 2023.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning (ICML)*, 2023.

William Peebles, Ilija Radosavovic, Tim Brooks, Alexei A. Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*, 2022.

Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. Self-supervised representation learning on neural network weights for model characteristic prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Konstantin Schürholt, Boris Knyazev, Xavier Giró i Nieto, and Damian Borth. Hyper-representations as generative models: Sampling unseen neural network weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Konstantin Schürholt, Diyar Taskiran, Boris Knyazev, Xavier Giró i Nieto, and Damian Borth. Model zoos: A dataset of diverse populations of neural network models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Konstantin Schürholt, Michael W. Mahoney, and Damian Borth. Towards scalable and versatile weight space learning. In *International Conference on Machine Learning (ICML)*, 2024.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? Investigating data replication in diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Bedionita Soro, Bruno Andreis, Hayeon Lee, Wonyong Jeong, Song Chong, Frank Hutter, and Sung Ju Hwang. Diffusion-based neural network weights generation. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2402.18153.

Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion. *arXiv preprint arXiv:2402.13144*, 2024.

Kai Wang, Dongwen Tang, Wangbo Zhao, Konstantin Schürholt, Zhangyang Wang, and Yang You. Scaling up parameter generation: A recurrent diffusion approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2501.11587.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.

Boya Zeng, Yida Yin, Zhiqiu Xu, and Zhuang Liu. Generative modeling of weights: Generalization or memorization? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2506.07998.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J. Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A TRANSFER ACROSS MODEL SCALES

We test whether within-cell representations remain functional at unseen scales, separating codec reconstruction from direct weight transfer.

Codec reconstruction. The custom-zoo $1\times$ codec loses at most one accuracy point on its four training cells (Section 2), but fails on held-out medium models. Encoding and decoding turn CONVNET-MEDIUM from 0.79 accuracy to non-finite predictions and VIT-MEDIUM from 0.67 to chance; reconstruction MSE is 0.22 and 0.28, respectively. Errors affect several tensor types and vocabulary indices, including components seen at smaller scales. Tested noise-robust fine-tuning does not improve perturbation tolerance, while fixed-noise training sacrifices clean reconstruction accuracy. Codec failure thus prevents attributing downstream failure to the denoiser.

Direct weight transfer. Canonical-field kNN transfers values and moment transfer maps statistics, both in topological coordinates and without a codec. Neither preserves function on any tested scale pair, including tiny↔small within a family. Transfers to medium models yield chance-level ViTs and non-finite CNNs. These outcomes limit the tested representations and mappings; they do not establish a general barrier to cross-scale weight generation.

B EXPERIMENTAL PROTOCOLS

Custom-zoo coefficient and latent diffusion, public-zoo MLP controls, and public-zoo full-latent U-Net confirmation differ in representation, normalization, training, and sampling. Only comparisons within a specified pipeline isolate its varied factor.

B.1 DATA AND CHECKPOINT ACQUISITION

Data splits. The custom zoo uses CIFAR-10 identifier `r1-reviewed-cifar10` and version prefix `e71629e215c9`. Frozen, checksummed indices define disjoint 45,000-image training and 5,000-image validation splits. Images are scaled to $[0, 1]$ and normalized with per-channel means $(0.4914, 0.4820, 0.4463)$ and standard deviations $(0.2470, 0.2434, 0.2614)$ computed on the training split; the same constants apply to validation and test images, without augmentation. Validation supports selection, conditioning, and development evaluation. The 10,000-image test split was used only for the finalized preregistered analysis of Section 3.3; it does not independently validate the other development comparisons. The public zoo retains its shipped 42,000/8,000/10,000 splits and preprocessing. Top-10% U-Net exploration inspected test accuracy; confirmation uses fresh seeds on those same images.

Base zoo. Custom-zoo networks are trained with AdamW ($\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-4}), constant learning rate 10^{-3}, batch size 256, float16 mixed precision with a dynamic loss scaler, PyTorch default initialization seeded per run, and 35,200 updates without augmentation. Batch order is fixed for each seed–epoch pair. Each architecture–scale cell contains nine independently seeded runs, saved at 25, 50, 75, and 100% of training (Table 3).

Reference trajectories and dense acquisition. Within each training cell, runs are ranked by final validation accuracy, with ties broken by ascending seed identifier. The highest-ranked run defines the reference trajectory. Dense re-execution records stage k at update $110k$, for $k = 1, \ldots, 320$; the second- and third-ranked runs are re-executed at the same cadence. All re-executions use the frozen training recipe. The four reference trajectories reproduce all 16 shared-stage checkpoints tensor-identically, and the eight additional trajectories reproduce all 32. The sparse nine-stage supports use checkpoints at 2, 5, 10, 15, 20, 25, 50, 75, and 100% of training (updates 704, 1,760, 3,520, 5,280, 7,040, 8,800, 17,600, 26,400, and 35,200).

B.2 SUPPORTS AND TARGET EXCLUSION

All target checkpoints are excluded from generator training and subsequent support-derived fitting (basis, coordinate scales, conditioning statistics, and novelty references). Their validation accuracies

provide evaluation requests. Upstream preprocessing has limited exposure: the structural codec was trained on all nine sparse-stage targets and one of the nine dense-frontier targets (stage 160, the 50% base checkpoint); the fixed-support and per-chunk codec controls use role statistics pooled over 356 checkpoints, including the nine targets. The structural codec was initialized from the 144 training-cell base checkpoints and adapted on each reference trajectory's nine sparse stages. Thus these are *generator-level*, not end-to-end representation-level, checkpoint hold-outs. Image-split hold-out is separate.

Nine-stage leave-one-out. Each fold removes one of the nine stages and refits the $K = 7$ basis, conditioning statistics, and generator on the remaining eight, in the structural-codec latent space. There are nine fits per cell, with overlapping training supports. The held-out stage is a codec training checkpoint in every fold.

Dense frontier. The following nine stages are excluded jointly:

$$18,\ 53,\ 89,\ 124,\ 160,\ 196,\ 231,\ 267,\ 302,$$

the rounded midpoints of nine equal trajectory intervals, spanning updates 1,980–33,220. The remaining $N = 311$ checkpoints define one support with principal dimension $K = 310$ in the structural-codec latent space. Its mean, basis, coordinate scales, conditioning statistics, novelty references, and generator are fitted without the targets. The frontier uses one generator seed per cell; the same fit serves all nine target evaluations, with four diffusion samples, four box samples, two samples per noise level, one interpolant, and one nearest-accuracy retrieval per target. Steering statistics pool the samples of all nine targets and are computed over finite samples only. Novelty references comprise either all 311 support checkpoints or the uniform nine-point subsample at support positions $0, 39, 78, 116, 155, 194, 232, 271, 310$ (rounded from nine equally spaced positions), excluding all targets.

Fixed-support comparison. The comparison covers updates 8,800–35,200. Nine reference-run target updates are jointly excluded: 10,230, 13,200, 16,170, 19,030, 22,000, 24,970, 27,830, 30,800, and 33,770. These are rounded midpoints of nine intervals among the 241 in-range stages, none coinciding with a base-zoo checkpoint. The one-run arm selects 36 evenly spaced remaining stages; the three-run arm takes 12 shared indices from each of the three highest-ranked runs (updates 8,800, 11,220, 13,640, 15,950, 18,370, 20,790, 23,210, 25,630, 28,050, 30,360, 32,780, 35,200); the nine-run arm uses four base-zoo checkpoints from every run. Each arm has $N = MT = 36$, with M source runs and T stages per run. A full-range one-run control selects 36 stages from all 320 after excluding these same targets. The one- and three-run primary supports share endpoints at updates 8,800 and 35,200. Source-run selection and temporal density also vary, so the comparison does not identify run count alone.

Each arm uses role-standardized raw weights without a codec and fits its own $K = 35$ basis, conditioning statistics, and novelty reference. Five generator seeds train for 3,000 updates each; per seed, four samples at each of nine targets yield 36 samples. Targets share the generator fitted for that arm.

Custom-zoo alignment. Git Re-Basin weight matching performs deterministic coordinate descent over permutation variables, each a linear assignment that maximizes the affected weight inner products, stopping at a fixed point or after 50 sweeps. CNNs permute stage channels consistently through convolution, normalization parameters and statistics, and the next layer. ViTs permute the residual stream, MLP hidden units, and whole attention heads, consistently across connected tensors. Contract tests confirm functional equivalence to tolerance 10^{-5} and exact recovery of planted permutations for both families; no per-run contract test occurs in the experiment. Each non-reference run is matched at final stage 320 to the reference final checkpoint; its permutation then applies to all selected stages. The nine-run analysis of Appendix D uses the same procedure on base-zoo final checkpoints. The aligned three-run accuracies are reported with seed variation in Table 4.

B.3 Representations and Generators

Structural codec and latent diffusion. The structural codec encodes role-standardized weights in 64-value chunks with topological metadata. Its local, tensor, and global Transformer stages each

have depth two, with eight heads, local windows of eight, and latent width 64 per chunk. It is trained on the 144 training-cell base checkpoints for 18,000 epochs, with one full-batch AdamW update per epoch ($\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01) at a constant learning rate of 10^{-3}. Its weights are then adapted separately to each reference trajectory's nine sparse stages for 6,000 epochs at 3×10^{-4}, with the optimizer state reinitialized. Before principal analysis, codec latents are standardized per coordinate using training-latent means and population standard deviations floored at 10^{-8}. On the four training cells, round-trip weight MSE is $(3\text{–}10) \times 10^{-5}$ and accuracy loss is at most one percentage point.

The separate latent-diffusion control of Section 3.3 predicts ϵ with a 1,000-step linear schedule ($\beta : 10^{-4} \to 0.02$), DDIM-100 sampling, clean-latent thresholding at ± 6, EMA decay 0.999, and cosine learning-rate decay from 3×10^{-4} over 300,000 full-batch updates. It uses uniform loss and accuracy conditioning in the timestep embedding, without dropout or guidance. Custom-zoo coefficient diffusion instead uses an MLP ϵ-predictor, 100 steps, DDIM-50, conditioning dropout 0.2, and guidance $\epsilon_u + 1.5(\epsilon_c - \epsilon_u)$. The public-zoo U-Net follows a separate protocol below.

Principal coordinates and normalization. Let $x \in \mathbb{R}^P$ denote a role-standardized weight vector or a standardized codec latent, and let μ be its support mean. Role standardization pools one mean and one population standard deviation per parameter role over all elements of the training checkpoints (floor 10^{-12}). For the dimensional sweeps, x is the codec latent with $P = 2{,}176$ (ConvNet-tiny), $24{,}960$ (ConvNet-small), $16{,}128$ (ViT-tiny), and $75{,}136$ (ViT-small); the fixed-support comparison uses raw weights. An orthonormal frame $B_d \in \mathbb{R}^{P \times d}$ contains the empirical support span, whose first $K = N - 1$ directions are the right singular vectors of the centered support (no numerical rank tolerance is applied). For diagonal positive scales D, coordinates and reconstruction are

$$z = D^{-1} B_d^\top (x - \mu), \qquad \widehat{x} = \mu + B_d D \widehat{z}.$$

Whitened span generation sets $d = K$ and scales principal coordinates by their sample standard deviations over the N support points (floor 10^{-6}). Dimensional sweeps extend the span with $d - K$ Gaussian directions projected off the span and orthonormalized by QR. Geometry-preserving sweeps apply a QR-orthogonalized random rotation to all d coordinates and use one common scale, $D = sI_d$, with s the root-mean-square of all $N \cdot d$ coordinates, recomputed for each dimension; at $d = P$ the native latent coordinates are used. The span-aligned, per-coordinate-normalized construction instead places the support span in the first K coordinates and floors coordinate standard deviations at 10^{-6}. For a fixed support, random extensions and rotations depend only on d and are shared across arms and generator seeds within a cell.

Exactly off-span coordinates have zero empirical variance in the span-aligned frame. Reconstruction therefore multiplies their generated components by 10^{-6}, suppressing off-span perturbations. Predictive accuracy can consequently persist when aggregate denoising loss is near the zero-predictor baseline. This observation concerns the reconstruction map and does not establish that the denoiser learned nothing in the functionally relevant directions. No literal zero-denoiser sampling control was run; the statement rests on trained per-coordinate conditions whose weighted and unweighted losses both equal the trivial value to three decimals.

Dimensional sweeps and optimization. The standard MLP trunk has width 512; the $4\times$ trunk has width 2,048 (a width ratio, not a parameter-count ratio). Main arms use five generator seeds and wide trunks three. The frame, normalization, budget, and loss-restricted ablations use three seeds each; the bridge uses one and the converged per-coordinate arm five. Updates use all N support points. Training stops when an EMA of weighted loss (decay 0.99), checked every 2,000 steps, is below 0.98 and improves by less than 1%, or reaches 30,000 updates; budget and bridge arms instead run 3,000. All wide $d = 310$ and $d = 1024$ runs reach the ceiling. At its final check one seed per cell has not met the plateau criterion, and the other two meet it only there. Accuracy at $d = 1024$ improves with width, but wide-trunk loss remains $1.9\text{–}2.6\times$ the loss at $d = 310$ (Table 5). These runs do not isolate a dimensionality effect at matched fit.

Denoising losses. Reported losses are not on one scale. Sweep losses are the moving average of the sigmoid-weighted training objective in the arm's own coordinate scaling, so they compare across dimensions within the geometry-preserving arms but not with the per-coordinate arms. Fixed-support losses are the final full-batch weighted loss after 3,000 updates in the $K = 35$ raw-weight

frame. U-Net losses are the weighted losses of the final mini-batches and are not comparable across weighting schemes. The only common unweighted diagnostic is the sweep's fresh-noise re-evaluation of the unweighted objective on the training support, which agrees with the weighted average for the trivial-loss conditions.

Per-chunk codec controls. A separate per-chunk Transformer autoencoder uses model width 256, eight attention heads, and two layers. Latent widths 64, 28, 16, and 8 per 64-value chunk define the nominal $1\times$, $2.26\times$, $4\times$, and $8\times$ conditions; the nominal $2.26\times$ label is the reference method's token compression ratio ($289/128$), realized here as $64/28 \approx 2.29\times$. Each codec is fitted to its own support with reconstruction MSE, 4,000 AdamW updates, learning rate 10^{-3} with cosine decay, and batch size 12. Public-zoo codec controls use three codec seeds per compression level; custom-zoo controls use five; every codec seed is paired with every generator seed.

The reported custom-zoo compression comparison uses the one-run 36-checkpoint support. MLP diffusion operates in the whitened $K = 35$ span of raw weights or codec latents, with sigmoid tilting on standardized validation accuracy. Five generator seeds each produce four samples per jointly held-out target. Round-trip accuracy is evaluated separately for each codec. The reported comparison excludes the nine-run arm contained in the same records.

Performance weighting and conditioning. Custom-zoo coefficient diffusion standardizes each support's validation accuracy as z and applies $w = \sigma(z)$ normalized to unit mean. It conditions on z using the architecture-specific embeddings described below, drops conditioning with probability 0.2, and samples at the requested z with guidance 1.5. Every support checkpoint enters every update. Public-zoo MLP controls use the same guidance and unnormalized $w = \sigma(z)$; training uses uniform mini-batches, and sampling requests the pool's best validation accuracy. The public-zoo U-Net has no accuracy condition, dropout, or guidance. It weights the loss by 1, $\sigma(10(v - 0.5))$, or $\exp(5v)$, each normalized to unit mean, where v is pool min–max-normalized validation accuracy; mini-batches are uniform with replacement. Sigmoid and exponential weighting each have a maximum-to-minimum ratio of 148. Their post hoc effective sample sizes $(\sum_i w_i)^2 / \sum_i w_i^2$ are respectively 299 and 245 on top-20% ($N = 1000$), and 455 and 407 on top-30% ($N = 1500$).

Custom coefficient-diffusion configuration. Coefficient diffusion uses an ϵ-predicting multilayer perceptron with SiLU activations and no residual connections: a linear input projection of the d coordinates, a learned embedding of the 100 timesteps, and a linear embedding of the standardized accuracy request are summed, followed by two hidden linear layers and a linear output layer. The hidden width is 512, or 2,048 in the wider-trunk arm; the nine-stage leave-one-out study uses a smaller variant of width 128 in which the timestep and accuracy embeddings are concatenated with the input. The accuracy condition is the z-score of validation accuracy over the training support and is dropped with probability 0.2 during training; sampling uses classifier-free guidance $\epsilon = \epsilon_u + 1.5(\epsilon_c - \epsilon_u)$ at the standardized target accuracy. The forward process has 100 steps with β increasing linearly from 10^{-4} to 0.02; sampling is deterministic DDIM over 50 evenly spaced steps with no clipping or thresholding. Training uses every support checkpoint at every update, sigmoid tilting on the standardized accuracy normalized to unit mean, and AdamW at learning rate 10^{-3} with cosine decay over the update horizon and library-default momentum, epsilon, and weight decay ($0.9/0.999$, 10^{-8}, 0.01); no gradient clipping, parameter averaging, or mixed precision is used. The nine-stage, dense-frontier, fixed-support, and per-chunk-codec experiments train for 3,000 updates on whitened coordinates; the dimensional sweeps train to the plateau criterion or 30,000 updates with a cosine horizon of 30,000, so early-stopped runs end before the decay completes, and use one global scale in the geometry-preserving arms. Each fitted generator produces four samples per accuracy request.

Custom-zoo baselines. Steering baselines use the generator's support and principal representation, with the same evaluation and novelty calibration. *Nearest-accuracy retrieval* returns the support checkpoint whose validation accuracy is closest to the request. *Conditional interpolation* sorts support checkpoints by validation accuracy and interpolates between those bracketing the request a, with $w = (a - a_j)/(a_{j+1} - a_j)$. For unwhitened principal coefficients, the interpolant is $c = (1 - w)c_j + wc_{j+1}$ and reconstructs as $\widehat{x} = \mu + B_K c$. Requests outside the support's accuracy range return its lowest- or highest-accuracy checkpoint. Reconstructed vectors are decoded through the codec in the nine-stage and dense-frontier protocols, or detokenized from standardized raw weights in the fixed-support comparison. The nine-stage audit also tests interpolation directly in raw

weights and between stage-order neighbours. *Interpolation plus noise* adds $\sigma\, s \odot \epsilon$ to the interpolant's unwhitened coefficients, where s contains the support coefficients' per-coordinate standard deviations and $\epsilon \sim \mathcal{N}(0, I_K)$. Noise remains within the span. We evaluate $\sigma \in \{0.1, 0.3, 1.0\}$ with two draws per level and target; Figure 2 shows 0.1 and 0.3. *Coefficient-box sampling* draws each coefficient independently and uniformly between its support minimum and maximum, independently of the request, with four draws per target.

The Section 3.3 comparison uses the dense 311-checkpoint support in role-standardized weight space with frozen codec statistics. *Copying* returns the validation-best checkpoint. *Averaging* takes the equal-weight mean of $k \in \{2, 3, 5, 10\}$ checkpoints drawn uniformly without replacement, independently of accuracy, with 36 draws per k. *Noise addition* perturbs the validation-best checkpoint with isotropic Gaussian noise whose standard deviation is σ times the global standard deviation of the support entries, using $\sigma \in \{0.01, 0.02, 0.05, 0.1\}$ and 36 draws per level. Novelty in this comparison divides nearest-support distance by the full support's median leave-one-out distance. No baseline applies clipping, rejection, or renormalization.

Public-zoo MLP controls. The public CNN's 10,853 parameters are flattened in a fixed layer order, role-standardized using pool statistics, and padded to 170 chunks of width 64. Raw-weight conditions bypass the codec. Span conditions use $K = N - 1$ principal coordinates; full-coordinate conditions use padded weights or codec latents. Coordinate standard deviations are floored at 10^{-6} before normalization.

The MLP has hidden width 512 and uses 100 diffusion steps, DDIM-50 sampling, and classifier-free guidance of 1.5. Training uses 3,000 AdamW updates, learning rate 10^{-3}, batch size 64, and sigmoid tilting on standardized validation accuracy as specified above. Sampling requests the pool's best validation accuracy. Three generator seeds each produce 36 samples, for 108 samples per condition; the reported MLP best accuracy is the maximum over these 108 pooled samples.

B.4 Public-Zoo Protocols

Support and alignment. The public CIFAR-10 zoo contains 1,000 independently seeded runs of a 10,853-parameter CNN, with 51 checkpoints per run. The matched $N = 36$ comparison uses epochs 15–50 of one run or final checkpoints from 36 runs; an additional condition uses final checkpoints from 51 runs. Reported results use raw-weight span diffusion.

Alignment controls retain the representation and generator while applying Git Re-Basin weight matching over the CNN's four hidden-layer channel permutations (16, 32, 15, and 20 units; output units are not permuted), maximizing adjacent-layer inner products by linear assignment with coordinate descent over the four permutations in a seeded random order, at most 64 sweeps, stopping at a fixed point. Each run's epoch-50 checkpoint is matched to the epoch-50 checkpoint of the pool's first run (the run of the top-ranked checkpoint for the retention pools), and its permutation is applied to every retained stage from that run. On four verification runs a planted permutation is recovered exactly and accuracy is unchanged. Permutations are recomputed deterministically by each experiment from the same matcher rather than stored, and all arms and baselines within an experiment use the same aligned checkpoints. For the retention pools below, epoch-50 checkpoints provide auxiliary alignment information outside the retained epochs 21–25.

Retention and initial MLP controls. From epochs 21–25 of all 1,000 runs, retaining the top 10, 20, or 30% of checkpoints by validation accuracy gives 500, 1,000, or 1,500 checkpoints from 227, 379, or 520 distinct runs, respectively. Both MLP and U-Net experiments use aligned pools. Across 13 aligned MLP conditions spanning raw weights and codec latents, with span or full-coordinate generation, no sample exceeds 0.40 accuracy.

On the aligned top-10% pool, whitened latent-span MLP diffusion with the $1\times$ codec (codec-seed index zero) reaches best test accuracy 0.264 among 108 samples pooled across three generator seeds; raw-weight span diffusion reaches 0.240. The codec's signed round-trip accuracy loss, defined as original minus decoded accuracy, averages -0.0006 over the first eight pool checkpoints. Per-checkpoint losses were not retained; this probe provides neither a maximum-loss bound nor a full-pool estimate.

Full-latent U-Net configuration. The $1\times$ per-chunk codec produces a 170×64 latent, with one codec (seed index zero) fitted per pool on all retained checkpoints. Its Transformer has width 256, eight heads, two layers, and latent width 64; it minimizes reconstruction MSE for 4,000 AdamW steps at 10^{-3} with cosine decay and batch size 12. Fixed-order raw weights are role-standardized and zero-padded to 170 chunks before encoding. One global SD of all training latents sets their scale; no coordinate whitening is used.

The unconditional one-dimensional U-Net has 71.7M parameters and three levels of widths 256, 512, and 1,024. Each level has two residual blocks, with strided-convolution downsampling and nearest-neighbor upsampling. The bottleneck uses eight-head attention; timesteps use sinusoidal embeddings. It uses ϵ-prediction, 1,000 linear-β steps (10^{-4} to 0.02), uniform timesteps, and AdamW at 10^{-4} with 1,000 warm-up steps and cosine decay. Training draws batches of 64 with replacement, clips gradients at 1.0, uses bfloat16 autocast and EMA decay 0.999, and runs for 100,000 updates. From final EMA weights, deterministic DDIM-100 ($\eta = 0$) generates 1,024 models per run, without guidance or thresholding; padded positions are zeroed after decoding. There is no early stopping. Only the loss weights differ among uniform, sigmoid, and exponential conditions. Compared with the span MLP, representation, scaling, architecture, conditioning, schedule, optimizer, precision, and budgets all change. Executed configurations and code versions are retained in the reproducibility records.

Exploration and confirmation. Exploration on the top-10% pool selects the U-Net configuration after inspection of generated test accuracy; the preregistered selection rule, held-out denoising loss, had selected a Transformer denoiser whose samples were at chance, and the U-Net was adopted as a recorded post hoc choice. Confirmation fixes the configuration and evaluates top-20% and top-30% pools with three fresh generator seeds per weighting scheme, giving 18 runs. The retention pools are nested, so these seeds do not represent disjoint checkpoint populations.

Mean round-trip accuracy loss (original minus decoded test accuracy, evaluated for every retained checkpoint) is 0.0007 on top-20% and 0.0008 on top-30%. These measurements concern the U-Net codecs and cover the full pools, unlike the eight-checkpoint MLP codec probe above.

Sampling and comparison scope. Top-1@108 is the expected maximum test accuracy of a random 108-model subset of each run's 1,024 generated models. It uses 2,000 fixed-seed Monte Carlo subsets without replacement, with invalid samples assigned chance accuracy; reported means and population SDs aggregate three generator seeds. Validation-selected accuracy instead scores the highest-validation finite candidate among all 1,024 (ties by sample order). The pool reference is the zoo-recorded test accuracy of its validation-best checkpoint. Retrieval uses the same Top-1 estimator on 108 checkpoints sampled without replacement from the aligned pool; it is shared across weighting schemes.

For each run, distance-matched noise selects 1,024 anchors uniformly with replacement and perturbs role-standardized raw weights with isotropic Gaussian SD $m/\sqrt{P}$ per coordinate, where $P = 10{,}853$ and m is that generator run's median nearest-support distance. Matching uses aggregate distance, not accuracy or per-sample distance; all perturbed models are evaluated without rejection. Achieved generator medians in standardized units are $6.27/10.63/9.40$ (top-20%) and $7.38/13.29/12.52$ (top-30%) for uniform/sigmoid/exponential weighting; noise medians agree to two decimals. With exact reference distances 2.7732 and 2.7820, these correspond to normalized novelties $2.26/3.83/3.39$ and $2.65/4.78/4.50$ for both generator and matched noise (Table 7). Tables 2, 7, and 8 report generation quality, baselines, and frozen predictions. Prediction P1 uses the pre-confirmation TiltDiff Top-1 range $[0.558, 0.615]$ across variants and retention fractions. The original TiltDiff sampling budget is unspecified, so these experiments are neither a full reproduction nor a budget-matched comparison; they do not isolate the cause of the MLP deficit.

B.5 SANE Representation Audit

The SANE tokenizer produces 93 tokens of width 289. At latent width 128 per token, the representation has 11,904 dimensions for a network with 10,853 parameters. It compresses the padded token array, but not the unpadded parameter vector. Tokenizer inversion is exact on four audited checkpoints, without a trained autoencoder or generator.

The separate autoencoder experiment uses the released CIFAR-10 CNN configuration: input width 289, latent width 128, model width 1,024, eight heads, eight layers, and window size 64. Training uses 3,500 public-zoo samples with reconstruction and SimCLR contrastive objectives. Optimization uses AdamW at 10^{-4}, OneCycle scheduling, and 50 epochs. On full 93-token canonical sequences, mean round-trip accuracy loss is 0.3135, exceeding the preregistered 0.02 gate despite reconstruction $R^2 = 0.76$. No generator is trained on these latents.

A re-run using the audited reference preprocessing gives mean accuracy drops of 0.3135 on the 24 seen checkpoints and 0.3325 on checkpoints from 150 unseen runs after 50 epochs. Extending training to 2,000 epochs raises reconstruction R^2 to 0.9935: the mean drop falls to 0.0015 on seen checkpoints but remains 0.4016 on the unseen runs. None of the 150 unseen evaluations has a drop within 0.02. The preparation stores five views per sample, and the reference preprocessing uses canonical views for both contrastive branches; the earlier augmentation-mismatch explanation is therefore withdrawn. These results show that accurate reconstruction of the evaluated training checkpoints does not ensure preservation of function on the evaluated unseen runs. The reported plateau concerns reconstruction R^2, not all aspects of optimization or other SANE training recipes. No SANE-based generator is evaluated, and this audit does not explain the MLP or U-Net generation results.

Evaluation protocol. The 1,000 runs at epochs 21–25 are split by sorted run index into 700/150/150 runs for training, validation, and test. We report fixed 50- and 2,000-epoch endpoints on 24 training checkpoints and checkpoints from 150 unseen runs. The longer run meets its reconstruction-R^2 plateau criterion $|R^2(E) - R^2(0.8E)| < 0.005$ by epoch 1,600. No intermediate epoch was chosen using unseen-run accuracy. All decoded networks load and give finite predictions.

B.6 Evaluation Conventions

Accuracy and invalid outputs. All images in the relevant split are scored in evaluation mode, without fine-tuning, recalibration, or repair. Strict loading requires matching keys and tensor shapes. Custom-zoo CNN BatchNorm affine parameters and floating-point running statistics are generated like other tensors (statistics use the normalization-bias role); only the integer batch counter is copied from the last support checkpoint. ViTs use LayerNorm and the public CNN has no normalization layers. No custom-zoo loading failure occurred. Public-zoo loading failures and any non-finite logits receive 0.10 accuracy in unconditional means; yield above 0.40 counts them as failures.

The archived sweep (Table 6) instead reports finite-only means among 36 samples per condition, with finite fractions. Its six nominal $d = 4096$ ConvNet-tiny entries are excluded: $P = 2{,}176$, but an unchecked overcomplete frame produced $K + 2{,}176$ coordinates (2,184–2,486 across $N = 9$–311), rank $2{,}176$, with span–extension inner products as high as 0.29–0.83. These were neither 4,096-dimensional orthogonal conditions nor valid span extensions. Audited records retain their summaries; the remaining entries and main $d = 310, 1024$ comparisons have no identified frame error.

Public-zoo functionality. We evaluate 32 fresh initializations (seeds 1000–1031), using Kaiming-uniform convolutional and linear weights and biases 0.01. For their test accuracies $a_1, \ldots, a_{32}$, define

$$\mu_{\text{init}} = \frac{1}{32}\sum_{i=1}^{32} a_i, \qquad \sigma_{\text{init}} = \sqrt{\frac{1}{32}\sum_{i=1}^{32}(a_i - \mu_{\text{init}})^2},$$

$$\tau_{\text{func}} = \max\{\mu_{\text{init}} + 3\sigma_{\text{init}}, \max_i a_i\}.$$

The mean, population SD, and maximum are 0.1011, 0.0133, and 0.1283; unrounded values give $\tau_{\text{func}} = 0.140913$. A functional model must load, produce finite predictions, and exceed this accuracy. The empirical threshold applies only to the public CNN; it differs from the 0.40 yield cutoff and has no statistical false-positive guarantee.

Steering and uncertainty. Steering is measured by MAE and Spearman correlation between requested and achieved accuracy, computed over finite samples only in every protocol (nine-stage, dense frontier, and fixed-support); non-finite samples are counted in the finite fraction. Unconditional accuracy and yield assign chance accuracy to non-finite samples.

Across-seed means and SDs quantify generator variability conditional on the selected supports and evaluation images. They do not quantify uncertainty over source-run selection or datasets. Jointly held-out targets share a generator fit, and confirmatory seeds share retention pools and test images. Consistent validation-selected gains across seeds therefore do not establish significance over new image populations. Test-ranked Top-1 describes the upper tail and is distinct from selecting a model by validation accuracy.

Weight-space novelty. Let $\mathcal{S}$ denote the training support in standardized weight space, and $\mathcal{R}$ the reference set used to define its distance scale. For a generated weight vector $\widehat{w}$, novelty is

$$\nu(\widehat{w}; \mathcal{S}, \mathcal{R}) = \frac{\min_{w \in \mathcal{S}} \|\widehat{w} - w\|_2}{\operatorname{median}_{u \in \mathcal{R}} \min_{v \in \mathcal{R} \setminus \{u\}} \|u - v\|_2}.$$

The numerator always uses the full training support: each fixed-support arm uses its own 36 checkpoints, the dense frontier all 311, and the public-zoo experiments the retained pool. Each fixed-support arm uses its own support as the reference. The dense frontier uses either its full support or the uniform nine-point subsample above to define the denominator; the numerator is unchanged. Denominators are floored at 10^{-9}; a zero numerator (exact retrieval) is reported as zero. Novelty one represents typical reference-set separation, not a threshold for behavioral diversity. Samples with non-finite predictions remain in weight-novelty summaries when their distances are defined. In the public-zoo U-Net runs the reference distance was computed under reduced-precision matrix multiplication, which inflated the denominator by 11% (top-20%) and 5% (top-30%); Table 7 reports values corrected by the exactly recomputed denominators, and per-sample distances, accuracies, and predictions are unaffected.

Prediction disagreement and norm filtering. Prediction disagreement is the fraction of test images (validation images for the dense frontier) on which a sample's predicted labels differ from a reference checkpoint's; the raw value is the minimum over all support checkpoints, and the band is the median over the support of the nearest-other disagreement. The prediction-nearest checkpoint is not, in general, the weight-nearest checkpoint: in the U-Net runs the two coincide for 42–50% of samples under uniform weighting and 16–30% under tilted weighting (post hoc count). Undefined predictions are excluded from disagreement summaries.

Public-zoo MLP weight-novelty summaries exclude original-space weight norms above three times the maximum among 600 reference checkpoints, comprising 200 runs at epochs 21, 25, and 50. The reference maximum is approximately 8.5759, giving a bound of 25.7276. Excluded counts and the fraction within the bound are reported; mean novelty is undefined if no sample remains. In the U-Net runs every generated and noise sample lay within the bound, so no exclusion was applied. Accuracy and functionality retain all samples. Custom-zoo experiments use no such norm filter.

B.7 Reproducibility

Configurations, RNG states, data indices, and evaluation code are retained; shared-stage checkpoints from the reference and ranked runs reproduce tensor-identically. Experiments used single H200 GPUs with Python 3.9, PyTorch 2.6.0, CUDA 12.6, NumPy 1.25.2, and SciPy 1.9.3. The planned release includes code, configurations, checkpoint indices, evaluation records, and model checkpoints. The nine-stage leave-one-out source is preserved, but only aggregate results are retained in the research record; per-run output files were not archived.

C Model Zoo and Supporting Results

The base zoo contains 216 checkpoints under the frozen recipe with hash prefix `d5cad8669e0d`. Table 3 specifies its six cells; dense acquisitions are described above. Table 4 reports generator-seed variability, and Table 5 gives the corrected dimension and trunk-width comparison. The archived sweep in Table 6 is retained separately and does not establish an independent dimensionality effect.

Cell	Architecture	Parameters
CNN-T	2 stages, widths 8/16	1,586
CNN-S	3 stages, widths 16/32/64	24,346
CNN-M†	4 stages, widths 24/48/96/128	164,962
ViT-T	1 block, width 32, 2 heads, MLP 2×, patch 8	15,690
ViT-S	2 blocks, width 64, 4 heads, MLP 2×, patch 4	75,082
ViT-M†	4 blocks, width 128, 8 heads, MLP 3×, patch 4	677,770

Table 3: **Custom model zoo.** CNN/ViT denote ConvNet/CIFAR-ViT; T/S/M denote tiny/small/medium. Each cell has nine runs and four checkpoints per run. †Medium models are excluded from representation and generator fitting.

Condition	Seeds	CNN-S	CNN-T	ViT-S	ViT-T
$N = 311$, $d = K = 310$*; 36 samples per seed*					
Whitened	5	0.723 ± 0.002	0.563 ± 0.001	0.639 ± 0.001	0.520 ± 0.002
Geometry-preserving	5	0.596 ± 0.011	0.414 ± 0.007	0.581 ± 0.002	0.425 ± 0.003
Geometry-preserving, wide	3	0.689 ± 0.010	0.476 ± 0.007	0.625 ± 0.001	0.452 ± 0.011
$N = 36$*; 9 shared targets × 4 samples per seed*					
M=1, T=36	5	0.734 ± 0.000	0.581 ± 0.000	0.626 ± 0.000	0.530 ± 0.001
M=3, T=12	5	0.165 ± 0.020	0.153 ± 0.003	0.191 ± 0.016	0.188 ± 0.016
M=3, aligned	5	0.231 ± 0.016	0.268 ± 0.011	0.174 ± 0.026	0.201 ± 0.011
M=9, T=4	5	0.117 ± 0.006	0.111 ± 0.007	0.137 ± 0.013	0.178 ± 0.012

Table 4: **Generator-seed variability.** Mean unconditional accuracy ± SD across the stated seeds; non-finite samples receive chance accuracy. The upper block uses $N = 311$ and $d = K = 310$; the lower varies source-run count at $N = 36$. Wide denotes the $4\times$ trunk width. SDs of 0.000 reflect rounding. Target evaluations share a fitted generator, and uncertainty is conditional on the selected support.

Trunk / dimension	CNN-S	CNN-T	ViT-S	ViT-T
Standard, $d = 310$	0.596	0.414	0.581	0.425
Standard, $d = 1024$	0.100	0.101	0.152	0.122
$4\times$, $d = 310$	0.689	0.476	0.625	0.452
$4\times$, $d = 1024$	0.597	0.409	0.629	0.488

Table 5: **Dimension and trunk width in geometry-preserving coordinates.** Mean unconditional accuracy over five standard-trunk or three wider-trunk seeds. Compare dimensions within each trunk: widening improves $d = 1024$, but CNN accuracy remains below the same trunk at $d = 310$. Wider-trunk losses are approximately 0.04–0.05 at $d = 310$ and 0.10–0.11 at $d = 1024$, and all these runs reach the 30,000-update ceiling. The comparison is not matched in achieved denoiser fit.

Cell	N	d=K	d=512	d=4096	d=full
convnet-small	9	0.718	0.545 (0.53)	0.519 (0.72)	n.f.
	20	0.710	0.598 (0.69)	0.542 (0.86)	n.f.
	40	0.723	0.598 (0.81)	0.567 (0.89)	n.f.
	80	0.718	0.620 (0.97)	0.552 (0.89)	n.f.
	160	0.717	0.610	0.564 (0.89)	n.f.
	311	0.705	0.654	0.576 (0.92)	n.f.
convnet-tiny	9	0.550	0.288 (0.86)	—	0.115 (0.36)
	20	0.553	0.373 (0.86)	—	0.127 (0.44)
	40	0.560	0.418 (0.94)	—	0.126 (0.50)
	80	0.557	0.454 (0.94)	—	0.140 (0.47)
	160	0.559	0.474 (0.97)	—	0.139 (0.53)
	311	0.543 (0.97)	0.505 (0.97)	—	0.142 (0.47)
vit-small	9	0.619	0.427	0.476	0.151
	20	0.632	0.502	0.517	0.179
	40	0.635	0.547	0.528	0.195
	80	0.638	0.563	0.532	0.199
	160	0.633	0.583	0.540	0.204
	311	0.631	0.606	0.513	0.205
vit-tiny	9	0.504	0.301	0.344	0.131
	20	0.511	0.380	0.379	0.142
	40	0.517	0.403	0.386	0.147
	80	0.514	0.426	0.384	0.148
	160	0.511	0.464	0.391	0.149
	311	0.506	0.478	0.285	0.150

Table 6: **Archived dimension sweep.** Mean validation accuracy over finite-output models among 36 samples per condition, using axis-aligned coordinates with per-coordinate normalization. Here N is the support size, $K = N - 1$ is the retained span dimension, and $d = \text{full} = P$ denotes the full representation dimension. Parentheses report finite-output fractions when below one; unparenthesized numerical entries indicate all samples are finite, and n.f. indicates none are finite. Dashes mark six excluded ConvNet-tiny conditions nominally labeled $d = 4096$, whose overcomplete frames were not valid orthogonal extensions (Appendix B). These finite-only means are not directly comparable with unconditional means elsewhere. Normalization and unequal denoiser fit prevent attributing accuracy differences to dimension alone.

C.1 Public-Zoo Confirmation

Table 2 summarizes full-latent U-Net confirmation over three fresh generator seeds per condition. Table 7 adds novelty measurements and noise and retrieval controls. All 18 runs recover TiltDiff's reported Top-1 accuracy scale, while mean accuracy remains below distance-matched noise. These comparisons neither reproduce the complete TiltDiff pipeline nor match its sampling budget.

Frozen predictions. Table 8 scores the frozen predictions, including the refuted prediction limiting tilting's Top-1 gain. The prediction identifiers are local to this protocol.

Interpretation. Exponential weighting improves seed-mean Top-1@108 over uniform by 0.0023 and 0.0045 while reducing yield by 17.2 and 27.6 percentage points on top-20% and top-30%, respectively. It beats retrieval Top-1@108 and the validation-best pool checkpoint under validation selection in all six confirmatory runs. Accuracy–distance Spearman correlations are -0.94 to -0.64 across the 18 runs. These shared-image runs establish neither significance across new image populations nor that generated models are copies.

		Novelty		Top-1@108	
Pool	Weighting	Weight	Prediction	Matched noise	Retrieval
20%	Uniform	2.26	0.125 (1.01)	0.5925	0.5956
20%	Sigmoid	3.83	0.197 (1.60)	0.5862	0.5956
20%	Exponential	3.39	0.175 (1.42)	0.5882	0.5956
30%	Uniform	2.65	0.144 (1.18)	0.5898	0.5949
30%	Sigmoid	4.78	0.248 (2.03)	0.5791	0.5949
30%	Exponential	4.50	0.234 (1.92)	0.5807	0.5949

Table 7: **Novelty and baseline controls.** Weight novelty is the mean across three generator seeds of each run's median normalized nearest-support distance, using corrected reference distances 2.7732 (top-20%) and 2.7820 (top-30%). Prediction novelty reports disagreement, with reference-band-normalized values in parentheses; summaries average the same three seeds as Table 2. Noise mean accuracy is reported there. Noise matches the generator's median nearest-support distance to two decimals, not its full distance distribution (Appendix B.4). Retrieval is shared across weighting schemes within a pool. Reference-distance units do not establish behavioral diversity.

ID	Frozen prediction	Top-20%	Top-30%
P1	Top-1@108 lies in the frozen range $[0.558, 0.615]$.	9/9	9/9
P2	Mean accuracy is below matched noise; yield is not higher.	9/9; 9/9	9/9; 9/9
P3	Top-1@108 exceeds matched noise in at least eight of nine runs.	9/9	9/9
P4a	Performance tilting reduces yield.	Supported	Supported
P4b	Tilting improves Top-1@108 over uniform by at most 0.003.	Supported $+0.0023$	Refuted $+0.0045$
P5	Exponential weighting's validation-selected sample outperforms the pool's best-by-validation checkpoint.	3/3	3/3
P6	Accuracy decreases with checkpoint distance.	9/9	9/9

Table 8: **Confirmatory predictions and outcomes.** Counts refer to fresh generator seeds across the relevant conditions, not independent test sets. Top-10% exploratory results are excluded. The largest Top-1 increase in P4b is obtained with exponential weighting.

D Run-Level Affine Coefficients

Samples and success criterion. We analyze nine-run supports with four checkpoints per run, with and without alignment, across the four custom-zoo cells. Each of these eight conditions contributes 9 nearest-checkpoint copies, 9 interpolants, 180 coefficient-diffusion samples, and 36 uniform box samples, giving 1,872 samples. Diffusion uses five seeds and four samples per seed at each of nine jointly held-out targets; each generator is shared across targets. Success requires validation accuracy strictly above 0.40. All 117 non-finite samples receive chance accuracy and count as failures. The public-zoo functionality threshold is not used.

Signed affine decomposition. Let $C \in \mathbb{R}^{36\times 35}$ contain the support checkpoints in coefficient coordinates, and let $c \in \mathbb{R}^{35}$ represent a sample. We obtain $w \in \mathbb{R}^{36}$ by solving

$$\begin{bmatrix} C^\top \\ \mathbf{1}^\top \end{bmatrix} w = \begin{bmatrix} c \\ 1 \end{bmatrix}$$

with float64 least squares. The last row imposes the affine sum-to-one condition. The solution is unique for affinely independent support checkpoints; original records do not retain per-sample residuals.

For run r, with checkpoint indices $\mathcal{I}_r$, define

$$m_r = \sum_{j\in\mathcal{I}_r} w_j, \qquad m_{\max} = \max_r m_r.$$

The coefficients are signed: $m_{\max}$ is not a probability and may exceed one. All samples have recorded coefficients; 48 have $m_{\max} > 1$. The upper group therefore includes all $m_{\max} \geq 0.95$, without an upper cutoff.

Results. Table 9 shows an association between larger $m_{\max}$ and higher accuracy, pooled across sampling methods.

$m_{\max}$ interval	Successes / total	Yield (> 0.40)	Mean accuracy
$(-\infty, 0.40)$	0 / 991	0.0%	0.120
$[0.40, 0.60)$	1 / 351	0.3%	0.156
$[0.60, 0.80)$	8 / 242	3.3%	0.186
$[0.80, 0.95)$	23 / 116	19.8%	0.214
$[0.95, \infty)$	126 / 172	73.3%	0.458

Table 9: Generated accuracy versus the maximum run-level signed affine coefficient sum. Results pool four sampling methods across aligned and unaligned nine-run supports in four architecture cells. Non-finite samples receive chance accuracy and count as failures. The $m_{\max} \geq 0.95$ group comprises 72 retrieved checkpoints, 36 same-run interpolants, 19 diffusion samples, and 45 box samples; its pooled yield is not a diffusion-specific rate.

Among samples with $m_{\max} \geq 0.95$, all 72 retrieved checkpoints and 36 same-run interpolants succeed; 18 of 19 diffusion samples succeed and all 45 box samples fail. The pooled yield is method-dependent, not a diffusion-specific rate, and a large signed sum does not ensure function.