BEYOND ORTHOGONALITY: ROTATION PARAMETERIZATION SHAPES SUSTAINED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning requires networks to keep learning as tasks change, yet prolonged training can erode this ability. Spectral constraints such as orthogonality can support adaptation, but leave open how the constrained weights should be represented and optimized. Motivated by rotation constructions shared by classical and quantum learning models, we examine this question while maintaining hidden-weight orthogonality throughout training. Across 1,000-task permutation streams on MNIST, CIFAR-10, CIFAR-100, and Tiny ImageNet, increasing rotation repetition from one to nineteen improves mean continual accuracy by 3.58–14.42 percentage points and increases the benefit of training history relative to networks trained anew on each task. Most of this improvement is reached by eight repetitions and persists under learning-rate selection and matched initialization; further repetition yields diminishing returns, while comparisons at a fixed parameter count reveal a complementary benefit from broader coordinate interactions. Beyond the number and arrangement of rotations, constraining their optimization coordinates also changes adaptation within the same nominal representable family. More broadly, these findings show that favorable weight spectra alone do not determine sustained learning: the structure and optimization of the underlying parameterization remain consequential.

1 INTRODUCTION

Continual learning depends on a network's ability to keep learning as its experience changes, yet prolonged training can erode this plasticity even when optimization continues (Lyle et al., 2023; Dohare et al., 2024; Abbas et al., 2023). Efforts to address this problem range from renewing parts of the learner through partial resets, continual backpropagation, and function-preserving plasticity injection (Nikishin et al., 2022; Dohare et al., 2024; Nikishin et al., 2023) to controlling how learned weights transform signals. Within the latter approach, spectral regularization targets the largest singular value (Lewandowski et al., 2025), Parseval regularization encourages orthogonality (Chung et al., 2024), and FIRE approximately restores weight isometry while minimizing changes to learned weights (Han et al., 2026). AdamO instead decouples isometry-promoting regularization from Adam's adaptive task update, with the same study also examining strictly orthogonal operators and GroupSort activations whose Jacobians are permutation matrices almost everywhere (Rosseau et al., 2026; Anil et al., 2019). Related quantum-learning evidence associates sustained adaptation in the models studied with unitary constraints and compact, periodic parameter manifolds (Chen & Zhang, 2026). Together, these findings motivate studying the conditions under which norm-preserving transformations support continued learning. An important question remains even when orthogonality is guaranteed throughout training. For the square rotation matrices studied here, every singular value stays at one and the represented weights remain in the compact group $\mathrm{SO}(n)$, but these shared properties leave considerable freedom in how the learner represents and updates its transformations. Different parameterizations can make different rotations available, expose different directions of change, or alter the effect of a parameter update on the represented matrix. Nor does compactness of the weight matrices bound their stored coordinates or biases, and hidden-weight orthogonality alone does not make the full ReLU network dynamically isometric. This distinction leads to our central question: *How do the representation and optimization of rotations affect continued learning when hidden-weight orthogonality is held fixed?*

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 investigate this question using compositions of elementary plane rotations, drawing on constructions established in tunable unitary networks and orthogonal finetuning (Jing et al., 2017; Liu et al., 2024). Because these compositions remain orthogonal for every parameter value, they allow us to vary the number and arrangement of rotations without changing the hidden-weight spectra. We increase Butterfly repetition to study rotation budget, replace its pairing schedule with matched Givens constructions, and use block-Givens to vary coordinate coupling at a fixed angle count. A complementary comparison constrains exponential optimization coordinates while retaining the nominal representable rotation family (Lezcano-Casado & Martínez-Rubio, 2019). The effects of these choices can involve initialization, expressivity, redundancy, and effective updates together. Together, these comparisons allow us to examine how the structure and optimization of rotation parameterizations affect continued adaptation when hidden-weight orthogonality is maintained.

2 Studying adaptation under orthogonal parameterizations

2.1 A common spectral constraint

We study networks with two width-128 hidden layers and a linear classifier, written under the row-vector convention as

$$h_\ell = \mathrm{ReLU}(h_{\ell-1}W_\ell(\theta_\ell) + b_\ell)\,, \qquad \ell \in \{1,2\}, \qquad f(x) = h_2 H + d. \tag{1}$$

Here, h_0 is the input representation, b_ℓ are trainable biases, and the classifier (H, d) follows a common training policy within each comparison. Rotation models maintain $W_\ell \in \mathrm{SO}(128)$ in exact arithmetic, while dense networks provide an unconstrained reference.

Orthogonal initialization and Jacobian conditioning have long been linked to learning dynamics (Saxe et al., 2014; Pennington et al., 2017; 2018). We maintain the hidden-weight constraint throughout training, fixing every singular value at one while allowing the rotations to evolve. However, the full network Jacobian also contains activation derivatives and the classifier, so hidden-weight orthogonality does not guarantee dynamical isometry. Likewise, compactness of the represented matrix family does not bound its optimization coordinates or biases.

2.2 Shared rotations and parameterization geometry

Classical evidence that spectral control can support continued learning has a counterpart in quantum models, where sustained adaptation has been associated with unitary constraints and compact, periodic parameter manifolds (Chen & Zhang, 2026). Orthogonal and unitary transformations share unit singular values and compact matrix domains, yet these properties alone do not settle trainability. Noiseless unitary circuits can exhibit barren plateaus (McClean et al., 2018), while local noise channels can also suppress gradients exponentially (Wang et al., 2021). Separately, rearranging data-encoding gates can affect classification performance at a fixed trainable parameter count (Periyasamy et al., 2022). These observations motivate examining the construction and optimization of norm-preserving transformations.

Givens rotations provide a concrete connection between the two settings. Within the single-excitation subspace, each quantum reconfigurable beam-splitter (RBS) gate rotates two amplitudes, giving its circuit an equivalent classical description as a composition of Givens rotations (Landman et al., 2022). Favorable gradient-variance scaling for Hamming-weight-preserving RBS circuits further motivates this family, although the results assume specified distributions and squared state-matching losses rather than our ReLU classification losses (Monbroussou et al., 2025). We choose Butterfly as a structured starting point, drawing on orthogonal finetuning and tunable unitary networks (Liu et al., 2024; Jing et al., 2017). The correspondence motivates the construction without implying quantum computational advantage. To understand what its parameterization leaves free, let $W(\theta) \in \mathrm{SO}(n)$ depend smoothly on p coordinates and define

$$\mathcal{B}_\theta = \frac{\partial\ \mathrm{vec}\, W(\theta)}{\partial \theta} \in \mathbb{R}^{n^2 \times p}. \tag{2}$$

A small update then gives

$$\mathrm{vec}(W(\theta + \Delta\theta) - W(\theta)) = \mathcal{B}_\theta \Delta\theta + O(\|\Delta\theta\|^2). \tag{3}$$

The column space of $\mathcal{B}_\theta$ describes locally accessible weight changes, while its singular values determine their scaling relative to coordinate perturbations. Neither is fixed by the singular values of W, so additional angles may introduce new directions or alter the scaling of existing ones.

Differentiating $W^\top W = I$ makes $W^\top \mathrm{d}W$ skew-symmetric and therefore yields

$$\operatorname{rank}(\mathcal{B}_\theta) \le \min\left\{p, \frac{n(n-1)}{2}\right\}. \tag{4}$$

For block-Givens,

$$W = Q^\top \operatorname{diag}(B_1, \ldots, B_{n/b})Q, \qquad B_i \in \mathrm{SO}(b), \tag{5}$$

a fixed orthogonal Q preserves the block family's dimension, tightening the bound to $\min\{p, n(b-1)/2\}$. At width 128, this bound rises from 448 to 8,128 as b increases from 8 to 128, despite a shared budget of 8,512 angles per layer.

These distinctions motivate varying rotation budget, pairing schedule, and coordinate coupling separately. The bounds describe geometric restrictions, not ranks measured during training or guarantees of better adaptation; the realized weight update also depends on the optimizer through $\Delta\theta$. Appendix A specifies the constructions.

2.3 Adaptation and the benefit of training history

We compare a continual learner that retains its parameters across tasks with a fresh reference that restarts its network on each task under the same task and training budget. The continual learner carries optimizer state unless a reset is specified, while the fresh reference's optimizer policy is stated for each comparison. For terminal test accuracies expressed as proportions, we define

$$G_{s,t} = 100(C_{s,t} - F_{s,t}), \qquad \overline{G}_{s,\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} G_{s,t}, \tag{6}$$

where $C_{s,t}$ and $F_{s,t}$ denote continual and fresh accuracy for seed s and task t.

A positive gap indicates a benefit from accumulated training history relative to the named reference, but a larger gap can reflect either stronger continual performance or weaker fresh performance. We therefore report absolute accuracies alongside paired gaps and their trajectories, characterizing sustained adaptation without equating the benefit of history with preservation of intrinsic plasticity or isolating parameter history, transfer, and optimizer history.

3 Rotation structure and continued adaptation

3.1 Same orthogonality, different adaptation

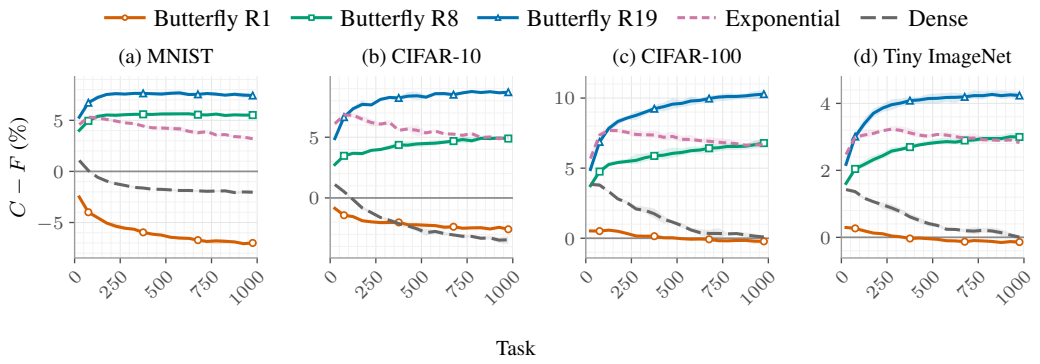

Figure 1: Training-history gaps with continual-optimizer carry. Figure 1a: MNIST; Figure 1b: CIFAR-10; Figure 1c: CIFAR-100; Figure 1d: Tiny ImageNet. Curves show nonoverlapping 50-task means; bands span the minimum and maximum of three environment means. Vertical scales are independent. $C - F$ (%) denotes percentage-point differences.

Our starting point is a common training budget under which models with the same hidden-weight spectra develop markedly different responses to continued training. We use 1,000-task pixel-permutation streams on MNIST, CIFAR-10, CIFAR-100, and Tiny ImageNet, with a frozen random projection to width 128 and a trainable classifier. Each task receives one epoch with batch size 128 and Adam (Kingma & Ba, 2015) at learning rate 10^{-3}; the continual learner carries its parameters and Adam state, while the fresh reference resets both. Performance summaries include all 1,000 tasks, averaging ten seeds within three equally weighted environments as specified in Appendix B.

Figure 1 reveals a consistent change with Butterfly repetition. From R1 to R19, continual accuracy rises from 78.01% to 92.43% on MNIST and from 30.26% to 40.42% on CIFAR-10, turning negative average gaps of -5.93 and -2.12 percentage points into positive gaps of $+7.39$ and $+8.09$ points. Fresh accuracy changes comparatively little on these datasets, so the larger advantage primarily accompanies stronger continual performance. CIFAR-100 and Tiny ImageNet also benefit from greater repetition, although their absolute accuracies remain modest and the increased gaps partly reflect weaker fresh references. Appendix C.1.1 gives the complete absolute outcomes, including the dense comparison.

The exponential model adds a temporal distinction to this picture: although continued training helps on average on every dataset, its advantage recedes during later tasks after an initial increase. These differences emerge while all orthogonal models maintain task-end hidden-weight deviations below 3.1×10^{-5}, showing that the shared spectral constraint leaves substantial freedom in both the magnitude and persistence of the benefit from history. Optimizer history is part of this comparison. Prior studies retain momentum in Continual ImageNet (Dohare et al., 2024) or reset optimizer state between Atari environments (Lyle et al., 2024), so we also evaluate separately trained reset counterparts. The improvement with Butterfly repetition persists under resets (Appendices C.1.2 and C.2.2), motivating a closer look at what changes as the rotation budget grows.

3.2 Rotation budget and pairing schedule

Between R1 and R19, increasing repetition adds both trainable angles and stages through which coordinates interact. We examine this progression over $R \in \{1, 2, 4, 8, 16, 19\}$, giving $7R$ stages and $448R$ angles per hidden layer, and ask whether its benefit depends on Butterfly's dyadic arrangement by replacing those pairings with fixed random perfect matchings. The resulting Givens models match the stage and angle counts and retain the preceding training and reference policies. Both schedules show increasing mean continual accuracy and gap across the six budgets (Figure 2; Table 1). For Butterfly on CIFAR-10, the gap moves from -2.12 points at R1 through $+1.31$ at R4 to $+8.09$ at R19; on CIFAR-100, continual accuracy rises from 8.74% to 16.94% as the gap grows from $+0.08$ to $+9.13$ points. The trajectories give this ordering further meaning: smaller-budget models can lose accuracy and relative advantage as training proceeds, whereas larger budgets sustain higher performance, with diminishing gains per additional angle toward the top of the ladder.

Table 1: Rotation budgets with continual-optimizer carry. Task-averaged (1–1,000) continual (C) and fresh (F) test accuracy (%) and paired gap $G = C - F$ (pp); $\pm$ is sample SD across three environment means.

	CIFAR-10			CIFAR-100		
Model	C	F	G	C	F	G
Butterfly R1	30.26	32.38	-2.12 ± 0.09	8.74	8.66	$+0.08 \pm 0.08$
Butterfly R2	32.07	32.57	-0.51 ± 0.04	10.20	8.57	$+1.63 \pm 0.08$
Butterfly R4	34.02	32.72	$+1.31 \pm 0.05$	11.84	8.43	$+3.41 \pm 0.18$
Butterfly R8	37.03	32.72	$+4.31 \pm 0.17$	14.15	8.22	$+5.92 \pm 0.25$
Butterfly R16	39.88	32.46	$+7.42 \pm 0.15$	16.52	7.92	$+8.60 \pm 0.23$
Butterfly R19	40.42	32.32	$+8.09 \pm 0.06$	16.94	7.81	$+9.13 \pm 0.24$
Givens R1	30.22	32.38	-2.16 ± 0.10	8.74	8.67	$+0.07 \pm 0.08$
Givens R2	32.05	32.68	-0.62 ± 0.00	10.23	8.58	$+1.64 \pm 0.09$
Givens R4	34.10	33.03	$+1.07 \pm 0.01$	11.87	8.51	$+3.36 \pm 0.16$
Givens R8	37.29	33.30	$+3.99 \pm 0.11$	14.33	8.46	$+5.88 \pm 0.25$
Givens R16	40.21	33.28	$+6.92 \pm 0.07$	16.74	8.33	$+8.41 \pm 0.15$
Givens R19	40.71	33.20	$+7.51 \pm 0.06$	17.19	8.25	$+8.94 \pm 0.15$

Random pairings preserve the budget response, but they also illustrate why the absolute outcomes matter. At the largest matched budget, Givens slightly exceeds Butterfly's continual accuracy on

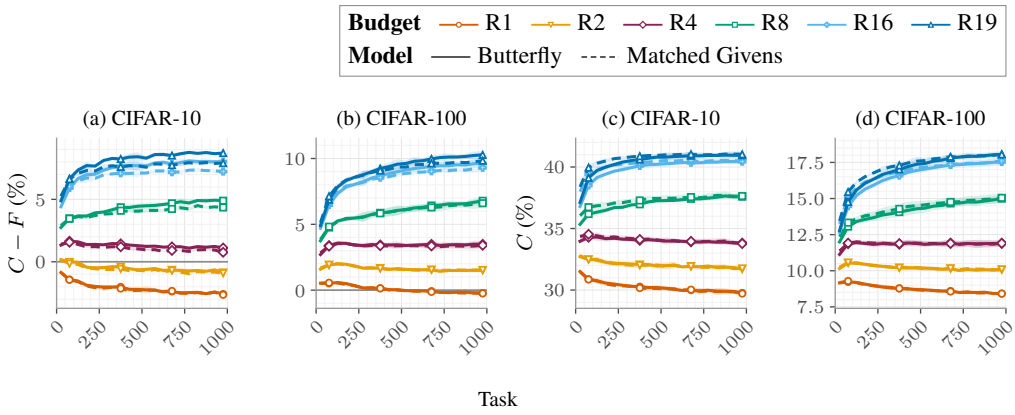

Figure 2: Rotation budgets with continual-optimizer carry. Figures 2a and 2b: CIFAR-10/CIFAR-100 gaps; Figures 2c and 2d: corresponding continual accuracy. Butterfly repetition R matches $7R$ Givens rounds. Curves show 50-task means; bands span one sample SD across three environment means. Vertical scales are independent; gaps labeled % are in percentage points.

both CIFAR datasets while producing smaller gaps, because its fresh reference improves as well. On CIFAR-10, for example, continual and fresh accuracies reach 40.71% and 33.20%, leaving a $+7.51$-point gap. The additional datasets and optimizer-reset comparisons retain the improvement with budget for both schedules (Appendices C.2 and C.2.2).

Greater repetition therefore helps beyond the particular dyadic construction, without establishing a plateau or identifying additional independent directions as the explanation. Since increasing the budget also changes initialization and parameterization geometry, a complementary comparison holds the stored angle count fixed and varies the coordinate interactions it supports.

3.3 Coordinate coupling at a fixed angle count

The block construction from Section 2.2 allows this comparison by restricting rotations to blocks of size $b \in \{8, 16, 32, 64, 128\}$, while retaining 133 rounds and 8,512 angles per hidden layer. We use a common fixed orthogonal change of basis and the same CIFAR-100 training protocol, so the comparison varies the organization of the rotation budget within the family already described.

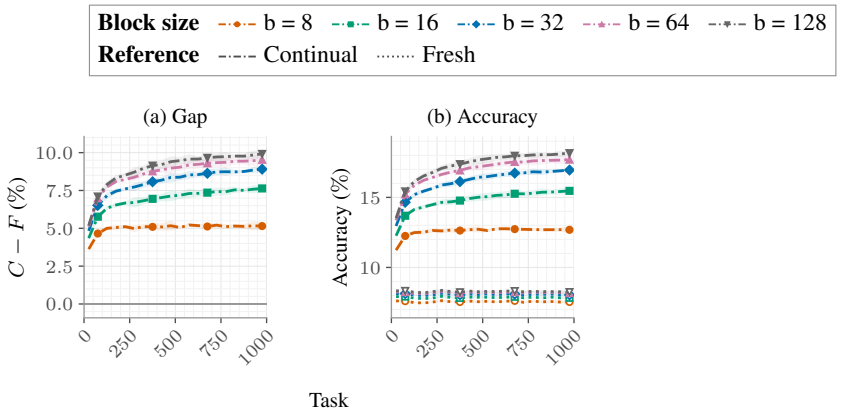

Figure 3: Figure 3a: paired gaps (pp, labeled %). Figure 3b: continual accuracy (dash-dot, filled markers) and fresh-reset accuracy (dotted, open markers). Block-Givens uses the R19 angle budget; colors and markers identify block sizes. Curves show 50-task means; bands span one sample SD across three environment means.

Table 2: Block-Givens with optimizer carry, averaged over tasks 1–1,000. C, F: continual and fresh-reset test accuracy (%); $G = C - F$: paired gap (pp). $\pm$ denotes sample SD across three environment means.

Block size b	C	F	G
8	12.58	7.56	$+5.02 \pm 0.17$
16	14.79	7.86	$+6.93 \pm 0.22$
32	16.11	8.08	$+8.03 \pm 0.20$
64	16.86	8.21	$+8.65 \pm 0.22$
128	17.26	8.28	$+8.97 \pm 0.23$

As the blocks widen, both continual accuracy and the benefit of history increase at every tested size (Figure 3; Table 2). From $b = 8$ to $b = 128$, continual accuracy rises from 12.58% to 17.26%, outpacing the increase in fresh accuracy from 7.56% to 8.28% and enlarging the gap from $+5.02$ to $+8.97$ points. Their trajectories separate as well: the smallest block levels off near five points after its initial rise, while larger blocks continue improving over much of the stream, with final-bin gaps of $+5.15$ and $+9.92$ points at the two extremes. The improvement becomes smaller as coupling broadens; moving from $b = 64$ to $b = 128$ adds only 0.40 points of mean continual accuracy and 0.32 points of gap.

Together with the budget ladders, this result shows that continued adaptation depends on how the stored rotations are organized as well as how many are available. Broader coupling improves absolute performance and the benefit of history under this protocol, but it changes the attainable transformations and optimization geometry together, leaving the mechanism unresolved rather than establishing a measured local-rank explanation.

4 Optimization and initialization within orthogonal parameterizations

The improvements in Section 3 leave open how much of the response to rotation structure depends on the conditions under which learning begins and proceeds. A common learning rate may suit some parameterizations better than others, while increasing repetition also changes the initial predictor. We examine these possibilities by allowing separate rate selection and matching initial functions, before turning to exponential coordinates to ask whether constraining optimization can change adaptation within the same nominal rotation family.

4.1 Learning-rate sensitivity with a frozen classifier

We first revisit the budget comparison with a frozen classifier, selecting Adam's learning rate separately for each Butterfly repetition by continual validation accuracy over tasks 901–1,000. The chosen rate is shared with the fresh-reset reference, while the frozen input projection and continual-optimizer carry policy remain as before. Under these conditions, increasing repetition from R1 to R19 raises CIFAR-100 continual accuracy from 3.20% to 4.95% and the paired gap from $+0.13 \pm 0.19$ to $+1.83 \pm 0.09$ percentage points (Figure 4). Although the intermediate budgets do not give a strictly monotonic progression, the broader improvement persists without classifier updates and after separate rate selection. Reported accuracies and gaps are averaged over all 1,000 tasks and four seeds; $\pm$ denotes the sample standard deviation across the four seed averages.

Alongside these performance differences, the trained networks develop different sensitivities to parameter and input changes. Following the diagnostic perspective of Rosseau et al. (2026), we examine the empirical NTK (Jacot et al., 2018) and nonlinear trunk Jacobian. Between R1 and R19, mean normalized effective rank rises from 0.055 to 0.080 for the NTK and from 0.461 to 0.787 for the trunk Jacobian, with smaller gains at high repetition. These increases indicate that parameter and input sensitivities are distributed across more effective directions, accompanying the improvement in continual accuracy. The diagnostics therefore connect rotation budget with differences in trained-network sensitivity, while leaving open how these differences interact with the optimizer and other mechanisms of plasticity (Lyle et al., 2025). Appendix D.1.1 gives the complete rate grids, Appendix D.1.2 the CIFAR-10 results, and Appendix D.1.6 extensions to VAE features, SGD, and fresh references that retain their own optimizer state.

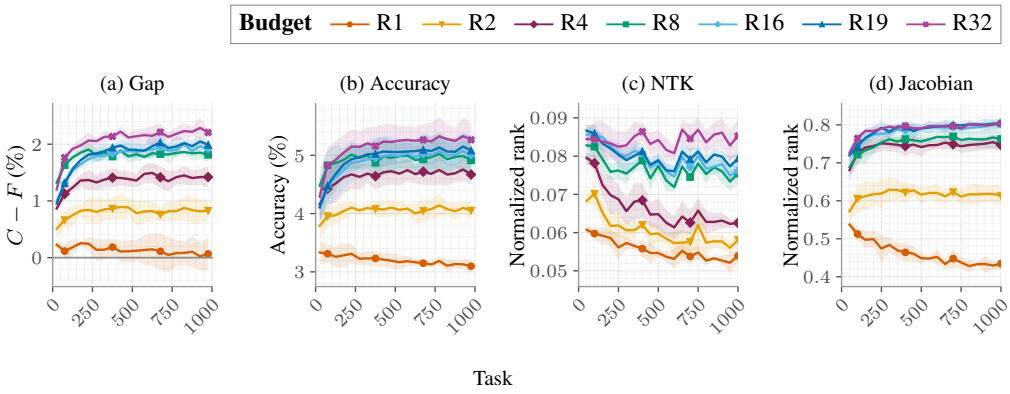

Figure 4: CIFAR-100 at selected Adam rates: Figure 4a, fresh-reset gap (pp, labeled %); Figure 4b, continual accuracy; Figure 4c, normalized NTK effective rank; Figure 4d, normalized trunk-Jacobian effective rank. Performance uses 50-task bins; ranks use 50-task checkpoints. Bands show one sample SD across four seeds.

The fixed-angle block comparison carries this observation beyond increasing parameter count. With the classifier frozen, all five block sizes select the same Adam rate, 10^{-3}, from the five-rate grid. Widening the blocks from 8 to 128 raises continual accuracy from 3.41% to 5.01%, exceeding the accompanying increase in fresh accuracy from 2.84% to 3.61% and enlarging the gap from $+0.57 \pm 0.10$ to $+1.40 \pm 0.08$ points (Figure 5). Thus, the benefit of broader coupling remains when both the stored angle count and selected rate are shared; the full comparison appears in Appendix D.1.3.

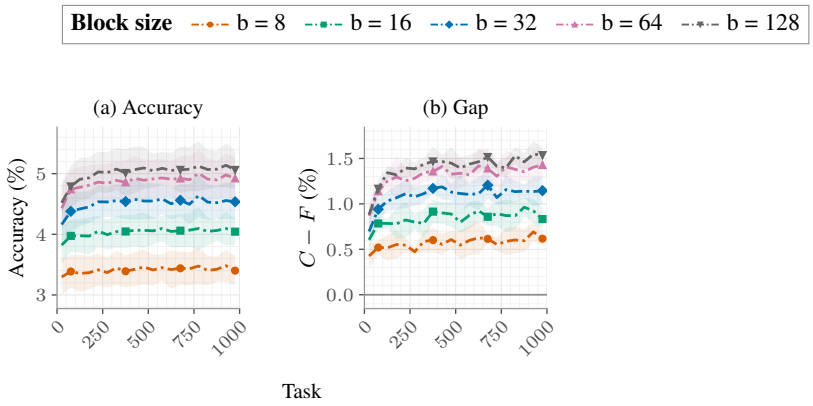

Figure 5: Selected-rate block-Givens with a frozen classifier: Figure 5a, continual accuracy; Figure 5b, fresh-reset gap (pp, labeled %). Colors and markers identify block sizes. Curves show 50-task means; bands span one sample SD across four seeds.

4.2 Matched initial functions across rotation budgets

Separate rate selection still leaves open whether larger budgets benefit from more favorable initial predictors. We therefore construct R16 and R19 by extending an R8 initialization with zero-angle identity blocks, leaving the added angles trainable. Shared biases and a frozen classifier make the initial functions identical, including each task-specific fresh draw, while allowing their subsequent evolution to differ. We retain the preceding CIFAR-100 protocol at a common Adam rate of 10^{-3}.

Starting from the same function does not remove the performance separation. Continual accuracy reaches 4.19%, 4.88%, and 4.94% for R8, matched R16, and matched R19, with corresponding gaps of 1.30 ± 0.07, 1.80 ± 0.09, and 1.84 ± 0.05 points (Figure 6). Both larger models improve continual accuracy in every seed, and their task-averaged accuracies remain within 0.02 percentage points of

ordinary initialization at the same rate. Their advantage therefore does not require different initial predictions.

The sensitivity measurements also separate as training proceeds, although more clearly in the trunk Jacobian than in the NTK. Averaged over 20 checkpoints, normalized trunk-Jacobian effective ranks are 0.724, 0.784, and 0.788, compared with NTK ranks of 0.078, 0.081, and 0.082. These differences characterize the networks reached from the common starting function without establishing which aspect of their evolution explains the accuracy gain. Appendix D.2 provides the ordinary-initialization controls and additional dataset, classifier, and optimizer-history comparisons.

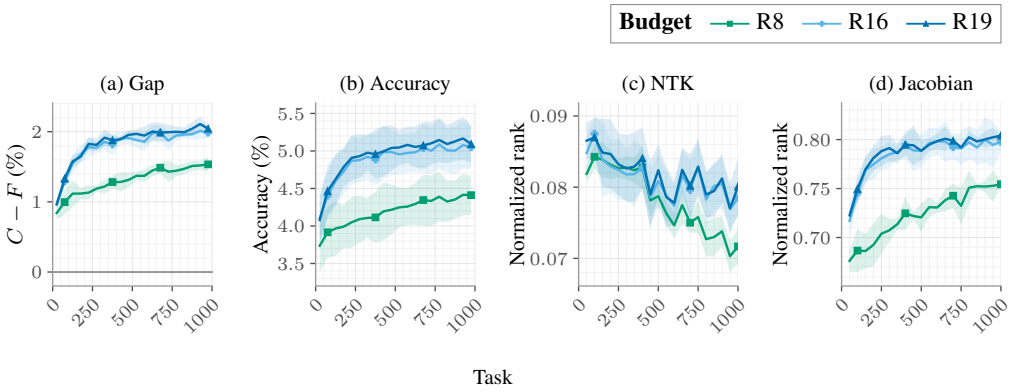

Figure 6: CIFAR-100 with R8-matched initialization and a frozen classifier: Figure 6a, fresh-reset gap (pp, labeled %); Figure 6b, continual accuracy; Figure 6c, normalized NTK effective rank; Figure 6d, normalized trunk-Jacobian effective rank. Performance uses 50-task bins; ranks use 50-task checkpoints. Bands show one sample SD across four seeds.

4.3 Constraining exponential optimization coordinates

Matching initial functions narrows the role of initialization, but the rotation-budget comparisons still change the parameterization itself. Exponential coordinates offer a complementary comparison: we can constrain the stored coordinates while retaining the nominal representable family. For $W(A) = \exp(A - A^{\mathsf{T}})$ (Lezcano-Casado & Martínez-Rubio, 2019), we project A onto a Frobenius ball after each Adam update, retaining Adam's moments. The radius preserves representability of $\mathrm{SO}(128)$ in exact arithmetic; Appendix D.3.1 specifies the constraint and numerical implementations.

Under the frozen-classifier CIFAR-100 protocol at the common rate 10^{-3}, the two trajectories diverge. Between the first and last 50 tasks, unconstrained continual accuracy falls from 4.64% to 4.26%, whereas constrained accuracy rises from 4.68% to 5.16%. Fresh accuracy changes similarly in both cases, so the gaps follow continual performance, falling from $+1.04$ to $+0.75$ points without the constraint and rising from $+1.09$ to $+1.66$ with it (Figure 7). Both gap directions hold in all four seeds.

Allowing each model its selected rate qualifies this advantage. At 5×10^{-3} for the unconstrained model and 10^{-3} for the constrained model, the constraint accompanies higher mean continual accuracy, 5.07% versus 4.70%, but also higher fresh accuracy, 3.55% versus 3.11%. Its gap is consequently slightly smaller, $+1.52 \pm 0.12$ versus $+1.60 \pm 0.09$ points. The constraint can also hurt performance: at 10^{-2}, continual accuracy drops from 4.32% to 3.65%. These comparisons show that coordinate constraints can change the course of continued adaptation, while their benefit remains dependent on learning rate. Appendix D.3.2 gives the full grids and additional settings, Appendix D.3.4 the VAE extension, and Appendix D.3.3 the spectral diagnostics; the higher effective ranks at selected rates also compare different learning rates.

5 Scope and practical performance

The aim of this paper is to analyze how rotation parameterization affects continued adaptation, rather than to propose a state-of-the-art method. Comparisons with established plasticity methods and ad-

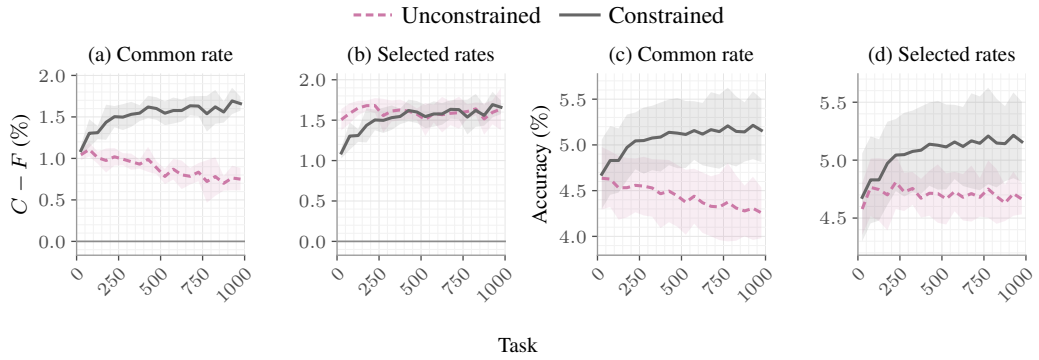

Figure 7: Frozen-classifier CIFAR-100: Figures 7a and 7b, fresh-reset gaps (pp, labeled %); Figures 7c and 7d, continual accuracy. Figures 7a and 7c use common LR 10^{-3}; Figures 7b and 7d use selected rates, 5×10^{-3} unconstrained and 10^{-3} constrained, reusing the constrained history.

ditional task settings nevertheless help assess the practical relevance and scope of the findings. The evaluated interventions include adaptations of Shrink-and-Perturb (Ash & Adams, 2020), L2-init regularization toward initial weights (Kumar et al., 2025), and ReDo neuron recycling (Sokar et al., 2023). We also evaluate LayerNorm (Ba et al., 2016) and spectral regularization (Lewandowski et al., 2025). Rankings against established plasticity methods vary between random-projection and VAE settings. In random-label relearning, Butterfly R8/R16/R19 sustain memorization while dense Adam deteriorates, although continual backpropagation achieves the highest absolute accuracy. In cumulative class-incremental CIFAR-100, R16/R19 achieve higher final accuracy than R8 at transferred learning rates, with all earlier training data retained. Appendix E reports the protocols, tuning procedures, quantitative comparisons, and reference-state policies for these evaluations.

6 Conclusion

Maintaining hidden-weight orthogonality leaves open whether a network benefits from continued training, and our results show that orthogonal parameterizations are not interchangeable in this respect. With identical hidden-weight spectra, the outcome ranges from below the fresh reference at Butterfly R1 to well above it at larger budgets. The response is clearest between R1 and R8: continual accuracy and the history gap rise steeply over this range on all four datasets, random pairings retain it, and it survives optimizer resets. Beyond R8 the returns diminish, although the response persists even with a completely frozen classifier and separately selected learning rates: continual accuracy rises from 28.08% at R1 to 31.39% at R8 on CIFAR-10 and from 3.20% to 4.92% on CIFAR-100, whereas R32 adds only 0.48 and 0.23 points more. Matched initialization confirms that these differences do not come from different initial predictions. The number of rotations is only part of the picture: widening the CIFAR-100 blocks raises continual accuracy from 12.58% to 17.26% at the same angle count, with the smallest step from $b = 64$ to $b = 128$, and constraining exponential optimization coordinates changes adaptation within the same nominal rotation family. These comparisons establish a role for parameterization beyond the shared spectral constraint, while leaving the contributions of expressivity, redundant coordinates, and optimizer dynamics unresolved. We do not propose a state-of-the-art method: the absolute accuracies of these small networks are modest, and several established interventions surpass the rotation models on frozen VAE features. The shared rotation construction also motivates future studies of continued adaptation in quantum models. Our classical results show that, even with identical hidden-weight spectra, the choice of parameterization can change whether accumulated experience helps or hinders subsequent learning. Orthogonality alone therefore does not deliver sustained learning; the parameterization that realizes it, and the way its coordinates are optimized, must be identified as deliberately as the spectral properties it preserves.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, literature searches and knowledge retrieval, discussing experimental results, and preparing and checking code, figures, and supplementary documentation. The authors take full responsibility for research ideation, experimental design, methodological decisions, scientific claims, interpretation of results, and all submitted materials.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix A describe the parameterizations and mathematical constructions. Appendix B documents the experimental protocols, endpoint definitions, learning-rate selection, and statistical reporting, while Appendices C, D, and E provide experiment-specific settings and supporting results. The supplementary material includes source code, environment specifications, experiment configurations, random seeds, frozen encoder and classifier weights, ordered data splits, and the saved per-run and diagnostic measurements. Supplied scripts download the public benchmark images and verify them against recorded hashes, and reconstruct the task streams, splits, and encoder features; reconstructed streams are checked against the original array hashes, and features are checked numerically. Its reproduction guide provides commands for running experiments and regenerating the reported figures and tables, and an artifact manifest connects each figure and table to its supporting data and scripts. All supplied figures and tables were regenerated from the packaged measurements without downloads or training.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pp. 620–636. PMLR, 2023. URL https://proceedings.mlr.press/v232/abbas23a.html.

Cem Anil, James Lucas, and Roger Grosse. Sorting out Lipschitz function approximation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 291–301. PMLR, 2019. URL https://proceedings.mlr.press/v97/anil19a.html.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/288cd2567953f06e460a33951f55daaf-Abstract.html.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

Yu-Qin Chen and Shi-Xin Zhang. Intrinsic preservation of plasticity in continual quantum learning. *PRX Quantum*, 7(3):033003, 2026. doi: 10.1103/ry52-85ll. URL https://doi.org/10.1103/ry52-85ll.

Wesley Chung, Lynn Cherif, David Meger, and Doina Precup. Parseval regularization for continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 127937–127967, 2024. doi: 10.52202/079017-4062. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e6df4efa20adf8ef9acb80e94072a429-Abstract-Conference.html.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632:768–774, 2024. doi: 10.1038/s41586-024-07711-7. URL https://www.nature.com/articles/s41586-024-07711-7.

Isaac Han, Sangyeon Park, Seungwon Oh, Donghu Kim, Hojoon Lee, and Kyung-Joong Kim. FIRE: Frobenius-isometry reinitialization for balancing the stability-plasticity tradeoff. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CfZLxT3zIZ.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1733–1741. PMLR, 2017. URL https://proceedings.mlr.press/v70/jing17a.html.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 410–430. PMLR, 2025. URL https://proceedings.mlr.press/v274/kumar25a.html.

Jonas Landman, Natansh Mathur, Yun Yvonna Li, Martin Strahm, Skander Kazdaghli, Anupam Prakash, and Iordanis Kerenidis. Quantum methods for neural networks and application to medical image classification. *Quantum*, 6:881, 2022. doi: 10.22331/q-2022-12-22-881. URL https://doi.org/10.22331/q-2022-12-22-881.

Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, András György, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C. Machado. Learning continually by spectral regularization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5565ab682d6c7f8d9da34ba0919974b0-Abstract-Conference.html.

Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3794–3803. PMLR, 2019. URL https://proceedings.mlr.press/v97/lezcano-casado19a.html.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7NzgkEdGyr.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23190–23211. PMLR, 2023. URL https://proceedings.mlr.press/v202/lyle23b.html.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, James Martens, Hado van Hasselt, Razvan Pascanu, and Will Dabney. Normalization and effective learning rates in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 106440–106473, 2024. doi: 10.52202/079017-3378. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c04d37be05ba74419d2d5705972a9d64-Abstract-Conference.html.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 750–783. PMLR, 2025. URL https://proceedings.mlr.press/v274/lyle25a.html.

Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9: 4812, 2018. doi: 10.1038/s41467-018-07090-4. URL https://doi.org/10.1038/s41467-018-07090-4.

Léo Monbroussou, Eliott Z. Mamon, Jonas Landman, Alex B. Grilo, Romain Kukla, and Elham Kashefi. Trainability and expressivity of Hamming-Weight preserving quantum circuits for machine learning. *Quantum*, 9:1745, 2025. doi: 10.22331/q-2025-05-15-1745. URL https://doi.org/10.22331/q-2025-05-15-1745.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37142–37159, 2023. doi: 10.52202/075280-1614. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/75101364dc3aa7772d27528ea504472b-Abstract-Conference.html.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d9fc0cdb67638d50f411432d0d41d0ba-Abstract.html.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1924–1932. PMLR, 2018. URL https://proceedings.mlr.press/v84/pennington18a.html.

Maniraman Periyasamy, Nico Meyer, Christian Ufrecht, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. Incremental data-uploading for full-quantum classification. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 31–37. IEEE, 2022. doi: 10.1109/QCE53715.2022.00021. URL https://doi.org/10.1109/QCE53715.2022.00021.

Andries Rosseau, Robert Müller, and Ann Nowé. Preserving plasticity in continual learning via dynamical isometry. In *The Forty-Third International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/60923.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *The Second International Conference on Learning Representations*, 2014. URL https://iclr.cc/archive/2014/conference-proceedings/.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168. PMLR, 2023. URL https://proceedings.mlr.press/v202/sokar23a.html.

Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12:6961, 2021. doi: 10.1038/s41467-021-27045-6. URL https://doi.org/10.1038/s41467-021-27045-6.

APPENDIX

A PARAMETERIZATIONS AND MATHEMATICAL DETAILS

A.1 ELEMENTARY GIVENS ROTATIONS AND STAGE COMPOSITION

All three angle-based parameterizations use the same elementary operation. With the row-vector convention of Eq. 1, let $G_{ij}(\theta)$ equal the identity except for its rows and columns indexed by the ordered pair (i, j), where it has the block

$$G_{ij}(\theta)\big|_{(i,j)} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \qquad (x_i, x_j) \mapsto (x_i \cos\theta - x_j \sin\theta,\ x_i \sin\theta + x_j \cos\theta). \quad (7)$$

Up to the angle-sign convention, this is the two-coordinate rotation underlying the RBS correspondence discussed in Section 2.2 (Landman et al., 2022). A stage consists of rotations on disjoint coordinate pairs, each with its own angle. The pairs within a stage can be evaluated together because no coordinate participates twice, while stages are applied sequentially. If $\mathcal{P}_t$ is the ordered list of pairs in stage t, the represented matrix is

$$S_t = \prod_{(i,j)\in\mathcal{P}_t} G_{ij}(\theta_{t,ij}), \qquad W = S_0 S_1 \cdots S_{T-1}, \qquad x \mapsto xW. \quad (8)$$

Products here follow application order from left to right. Each factor is a rotation, so the product belongs to $\mathrm{SO}(n)$ in exact arithmetic. Bias addition and ReLU occur after the complete hidden-layer product, not between its stages. Thus, additional rotation stages change the parameterization of a hidden linear map without adding nonlinear network layers. All angle counts below are per hidden layer and exclude biases and the classifier.

A.2 BUTTERFLY PAIRING SCHEDULE

For width $n = 2^L$, a Butterfly pass has stages $s = 0, \ldots, L-1$. Stage s pairs coordinate i with $i \oplus 2^s$ for every index i whose sth binary bit is zero, where $\oplus$ denotes bitwise exclusive or and indices start at zero. Equivalently, the pairs are

$$(2^{s+1}q + j,\ 2^{s+1}q + j + 2^s), \qquad 0 \le q < \frac{n}{2^{s+1}}, \quad 0 \le j < 2^s. \quad (9)$$

The separation doubles at successive stages. For example, at width eight the three stages use $(0,1), (2,3), (4,5), (6,7)$; then $(0,2), (1,3), (4,6), (5,7)$; then $(0,4), (1,5), (2,6), (3,7)$. Repeating this ordered sequence R times produces RL stages and $K = RnL/2$ angles. Each repetition has independent trainable angles; only the pairing schedule is reused. The stored angle vector is reshaped to $(R, L, n/2)$, indexing repetition, stage, and coordinate pair; passes are applied sequentially. Within a stage, reshaping the activation axis to $(n/2^{s+1}, 2, 2^s)$ brings paired coordinates together for Eq. 7, avoiding construction of a dense matrix. At width 128, $L = 7$ and $K = 448R$, giving the stored counts used in the main text. A single pass can mix information across all coordinates while still representing a restricted family of rotations; such mixing does not establish complete coverage of $\mathrm{SO}(n)$.

A.3 MATCHED RANDOM GIVENS SCHEDULES

The matched Givens construction replaces the dyadic schedule by a fixed sequence of random perfect matchings at even width n. For each stage, a permutation p_t of $\{0, \ldots, n-1\}$ is sampled once, and its adjacent entries define the pairs $(p_t(0), p_t(1)), \ldots, (p_t(n-2), p_t(n-1))$. The forward pass gathers coordinates in this order, applies Eq. 7 to adjacent pairs, and restores the original coordinate order before the next stage. The permutations are generated from the structural pairing seed and remain fixed throughout training and across tasks. For width $n = 2^L$, setting $T = RL$ matches Butterfly's number of stages and its $K = Tn/2$ stored angles. This matches the amount of trainable rotation computation at the stage level, but does not assert equal wall-clock cost, attainable matrices, or effective optimization directions. The pairing seed is separate from the seed used to initialize trainable angles; hidden layers with the same structural seed and pairing rule reuse the same schedules while receiving independently initialized parameters.

A.4 Block-Givens schedules and change of basis

Block-Givens partitions the coordinates into n/b contiguous groups of even size b, where b divides n. Each stage samples a perfect matching independently within each group, so all $n/2$ rotations remain inside their assigned blocks. The pair slots within each stage are also randomly reordered; with all slots active, this changes only the storage order of disjoint rotations. The fixed-angle comparison uses all $n/2$ slots and $T = 133$ stages for every block size, giving $K = Tn/2 = 8{,}512$ at $n = 128$. There is no sharing of angle values between stages or blocks. Before conjugation, the complete product is $B = \mathrm{diag}(B_1, \ldots, B_{n/b})$, with each B_a an element of $\mathrm{SO}(b)$. The family is consequently restricted by block size even though the number of stored angles is unchanged; repeated rotations can supply redundant coordinates rather than new independent directions.

The common change of basis in Eq. 5 is implemented as $x \mapsto xQ^\mathsf{T} \mapsto (xQ^\mathsf{T})B \mapsto ((xQ^\mathsf{T})B)Q$. The matrix Q is generated from a Gaussian matrix by QR decomposition with column signs corrected using the diagonal of the triangular factor. It is fixed, stored in FP32, and excluded from the trainable parameter count. Its generation depends on width and the structural seed, so changing block size at a fixed seed retains the same Q. Conjugation places the block restrictions in a common rotated basis and can yield a dense matrix in the original coordinates. Thus, block size controls which coordinates the trainable rotations couple in that basis, without implying that groups of original features remain disconnected. Evaluating the conjugated transformation also requires two fixed matrix multiplications, which are not included in the trainable angle budget.

A.5 Parameter storage and initialization

For the angle-based parameterizations, each hidden layer stores an FP32 angle vector and a separate bias vector. Ordinary initialization draws independent $\mathcal{N}(0, 0.05^2)$ angles, with hidden biases initialized to zero. The matched-initialization controls instead retain an R8 initialization and append trainable zero-angle identity blocks, as specified in Section 4.2 and Appendix D.2. Under ordinary initialization, increasing repetition changes the composition of randomly initialized rotations even when the per-angle distribution is unchanged. During optimization, the stored angles are neither clipped nor reduced modulo 2π: the forward map evaluates their sines and cosines without projecting the stored coordinates. Each rotation is periodic in its angle, but optimization updates the angles as unconstrained real-valued parameters. The represented matrices therefore lie in a compact orthogonal group, even though the trainable coordinates need not remain bounded. Structural formulas and stored parameter counts do not substitute for measurements of local Jacobian rank or learning performance.

A.6 Parameter counts and exponential representability

At width 128, a Butterfly trunk contains $2(448R + 128) = 896R + 256$ trainable parameters, including the two hidden biases, while the fixed-angle block trunk contains $2(8512 + 128) = 17{,}280$. The exponential trunk stores two full 128×128 generators and two hidden biases, totaling 33,024 coordinates, including redundant generator coordinates. A trainable classifier adds $129N_{\mathrm{class}}$ parameters for N_{class} output classes; frozen classifier parameters remain part of the model but are excluded from its trainable count. These counts describe storage rather than independent degrees of freedom, which is also relevant when interpreting the generator constraint. To see why the radius in Eq. 12 retains the nominal rotation family, write an arbitrary $W \in \mathrm{SO}(128)$ in a real orthogonal basis as 64 planar rotations, with angles $\phi_j \in [-\pi, \pi]$. A corresponding real skew-symmetric logarithm S satisfies

$$\exp(S) = W, \qquad \|S\|_F^2 = 2\sum_{j=1}^{64} \phi_j^2 \le 128\pi^2. \tag{10}$$

Choosing $A = S/2$ gives $A - A^\mathsf{T} = S$ and $\|A\|_F \le \pi\sqrt{128}/2$. This is a representability argument in exact arithmetic, not a statement about numerical conditioning, uniqueness of coordinates, or the optimization trajectory. In the exponential comparison on CIFAR-100 VAE features, both variants evaluate the matrix exponential using an order-10 Taylor kernel with 12 squarings and FP64 internal calculations; stored parameters and forward activations remain FP32.

B EXPERIMENTAL PROTOCOLS AND STATISTICAL REPORTING

Our experiments examine continued adaptation through two complementary sets of comparisons. Sections 3 and 4 study classification under input permutations: each task rearranges input coordinates while preserving the labels. Section 3 varies rotation budget, pairing schedule, and coordinate coupling using pixel permutations followed by a frozen random projection. Section 4 examines learning-rate sensitivity, matched initial functions, and constraints on exponential optimization coordinates. Appendices C and D provide the corresponding additional results and protocols, including experiments that permute the coordinates of frozen VAE representations.

Appendix E places these comparisons in a broader context. It evaluates established plasticity methods under the permutation protocols and examines two additional forms of sequential learning. In random-label relearning, the same images receive newly sampled labels on each task, testing the ability to repeatedly learn changing targets. In cumulative class-incremental learning, new classes are introduced in stages and all earlier training data remain available, testing adaptation as the classification problem expands. The following subsections explain the shared protocol choices, treatment of training history, selection procedures, and statistical conventions; experiment-specific settings are given alongside their results.

B.1 TASK STREAMS AND REPRESENTATION POLICIES

The opening experiments compare adaptation under a common training budget using 1,000-task pixel-permutation streams on MNIST, CIFAR-10, CIFAR-100, and Tiny ImageNet. Labels remain unchanged across tasks. Writing the task permutation as P_t, each input is permuted before a frozen random projection to width 128, giving $h_0 = xP_tQ_{\text{in}}$. The output classifier in Eq. 1 remains trainable. Each task receives one training epoch with batch size 128 and Adam at the fixed learning rate 10^{-3}. The continual learner retains its hidden parameters, classifier, and Adam state across task boundaries. Its fresh reference reinitializes the trainable network and optimizer on every task and receives the same task, minibatch sequence, and training budget. Their performance difference measures the benefit or cost of accumulated training state under this reference. The corresponding experiments with continual-optimizer resets are reported in Appendices C.1.2 and C.2.2.

The CIFAR-10 and CIFAR-100 random-projection experiments use all 50,000 official training images for training. The official test partition is divided into 2,000 validation images and 8,000 test images, with disjoint validation and test subsets within each environment. Section 4 retains pixel permutations and frozen random input projections while introducing frozen-classifier comparisons, learning-rate selection, and matched-initialization controls. These comparisons therefore vary the stated classifier, optimization, or initialization policy within the random-projection setting. For the CIFAR-10 and CIFAR-100 random-projection experiments, pixels are scaled by $1/255$ without image augmentation. At batch size 128, each training epoch comprises 390 updates, with the remaining 80 examples omitted. The Adam comparisons use $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and no weight decay; learning rates are specified with each comparison.

The latent-permutation experiments provide a complementary setting on CIFAR-10 and CIFAR-100. A frozen ResNet-VAE encoder produces the deterministic 128-dimensional posterior mean $\mu(x)$, which is used without clipping, rescaling, centering, or normalization. Each task permutes these latent coordinates, giving $h_0 = \mu(x)P_t$, while retaining all ten CIFAR-10 labels or all 100 CIFAR-100 labels. Random classifier weights and biases remain frozen and are shared across matched methods and learner roles within each model/head seed. This setting examines adaptation on a fixed learned representation with a fixed classifier. Representation, permutation placement, and classifier policy jointly define the comparison with the opening experiments. For each latent-permutation dataset, the official training partition supplies 45,000 training and 5,000 validation images, and evaluation uses all 10,000 official test images. A masked final minibatch at batch size 128 includes all training examples and gives 352 updates per one-epoch task. The frozen encoders use checkpoints from epoch 85 for CIFAR-10 and epoch 99 for CIFAR-100. The same 5,000-image validation split supports encoder selection and subsequent learner hyperparameter selection; encoder fitting did not include image deduplication.

B.2 Parameters and optimizer history

The three-role frozen-classifier permutation comparisons distinguish retention of model parameters from retention of optimizer state. Continual retains its learned trunk and its own optimizer state. Fresh-reset and fresh-carry start each task from identical task-specific fresh trunks; fresh-reset initializes a new optimizer state, whereas fresh-carry retains its own state from preceding tasks. Neither fresh role receives continual's optimizer state. All three roles share the same frozen classifier and, within a comparison, the same learning rate and training budget.

Comparing fresh-reset with fresh-carry measures the effect of optimizer-state retention at matched fresh parameters. Comparing continual with fresh-carry measures the benefit of retaining learned parameters under a shared optimizer-carry policy. Their optimizer-state values nevertheless reflect their respective training trajectories, so this comparison includes the interaction between parameter and optimizer histories. Continual versus fresh-reset measures the combined effect of retaining both forms of state. Experiments using a different set of roles or a trainable classifier specify those policies alongside their results, including the matched-initialization controls in Appendix D.2 and the additional task settings in Appendix E.

B.3 Learning-rate selection and reporting windows

Task-averaged performance summaries cover the full stated task horizon unless a particular window is explicitly identified. For the 1,000-task experiments, these summaries include tasks 1–1,000 without discarding an initial period. Early/late comparisons, tuning summaries, stagewise results, and final-stage outcomes use their stated windows or endpoints. In particular, final-stage cumulative classification accuracy describes performance after all classes have been introduced, rather than an average over the preceding stages. Where learning rates are selected, selection uses continual validation performance. Test accuracy and continual-minus-fresh gaps do not enter the selection criterion, and fresh references use the continual-selected rate. The candidate grid, selection window, and relationship between selection and reporting are specified with each experiment in Appendices D and E. Fixed-rate and transferred-rate comparisons retain their stated rates without a separate learning-rate search for each evaluated setting.

B.4 Endpoints and statistical summaries

Absolute accuracy and the continual-minus-fresh gap describe complementary aspects of adaptation. Accuracy is reported as a percentage, while differences between accuracies are reported in percentage points. For each task and seed, the gap pairs continual performance with the named fresh reference, as defined in Eq. 6. Paired differences are formed before averaging or calculating their variability. In the one-epoch experiments, terminal evaluation and evaluation at the validation-selected epoch coincide. Multi-epoch experiments distinguish terminal performance from performance at a validation-selected epoch; the random-label experiments instead report terminal and peak training accuracy. Each protocol also specifies which parameters and optimizer state are carried into the next task.

The opening experiments, primary rotation-budget comparisons, and coordinate-coupling experiments use ten model seeds distributed across three environments, with four, three, and three seeds, respectively. For task-averaged summaries, we first average tasks within each seed, then seeds within each environment, and finally give the three environment means equal weight. Trajectories apply the same aggregation within non-overlapping 50-task bins. Reported standard deviations are sample standard deviations across the three environment means. The opening figure instead uses bands spanning their minimum and maximum, while the primary rotation-budget and coordinate-coupling figures use one sample standard deviation. These quantities describe variation across environments and are not confidence intervals. The Section 4 comparisons and their Appendix D extensions use four seeds within one task environment, including both random-projection and VAE experiments. The matched R19/R32 comparison in Appendix C.2.3 uses the same four-seed, single-environment reporting structure. Task-averaged summaries first average within each seed and then report the mean and sample standard deviation across seeds; trajectory bands apply this calculation within each plotted task bin. In frozen-classifier comparisons, these seeds include the random classifier draws. VAE comparisons additionally share one encoder per dataset, so their seed variation does not measure variation across independently trained encoders. Throughout these summaries, repeated

tasks, epochs, and diagnostic checkpoints are observations within a run, rather than independent replications.

The quoted numerical orthogonality checks use the task-end deviation $\max_\ell \|W_\ell W_\ell^\mathsf{T} - I\|_F$, where ℓ ranges over the hidden layers. Each materialized weight matrix is converted to FP64 before computing this unnormalized Frobenius norm; layers are combined by their maximum. The reported bounds summarize recorded task endpoints, not every optimizer update.

C Rotation structure and continued adaptation

The comparisons in this appendix examine how the effects of rotation structure depend on optimizer-state retention, pairing schedule, and rotation budget. We first complement the main-text gap trajectories with absolute accuracies and compare training with and without Adam-state retention across tasks. We then extend the matched Butterfly and Givens comparisons to additional datasets and evaluate Butterfly at a larger rotation budget. The final subsection connects these budget comparisons to the fixed-angle experiment on coordinate coupling. Together, these results distinguish changes in absolute performance from changes in the advantage conferred by accumulated training.

C.1 Adaptation benefits

C.1.1 Absolute performance with optimizer carry

Table 3 reports the absolute continual and fresh test accuracies underlying Figure 1, using the task windows and aggregation defined in Appendix B. Among the five opening models, Butterfly R19 attains the highest mean continual accuracy on all four datasets. The absolute outcomes also clarify the interpretation of the gaps. On MNIST, for example, the dense model achieves 89.79% continual accuracy but a gap of -1.45 percentage points because its fresh reference reaches 91.24%. A negative gap therefore identifies an advantage for the fresh reference under the same training budget, while absolute accuracy describes the performance attained by each learner.

Table 3: Absolute performance in the opening comparison, with Adam state carried across tasks. C and F are continual and fresh-reset test accuracy (%); $G = C - F$ is their paired gap in percentage points. Values average tasks 1–1,000, with equal weight given to three environment means after averaging seeds within each environment. The uncertainty shown for G is one sample SD across those environment means.

Dataset	Model	C (%)	F (%)	G (pp)
MNIST	Butterfly R1	78.01	83.94	-5.93 ± 0.06
	Butterfly R8	90.27	84.83	$+5.44 \pm 0.08$
	Butterfly R19	92.43	85.04	$+7.39 \pm 0.14$
	Exponential	91.29	87.06	$+4.23 \pm 0.14$
	Dense	89.79	91.24	-1.45 ± 0.02
CIFAR-10	Butterfly R1	30.26	32.38	-2.12 ± 0.09
	Butterfly R8	37.03	32.72	$+4.31 \pm 0.17$
	Butterfly R19	40.42	32.32	$+8.09 \pm 0.06$
	Exponential	38.44	32.78	$+5.66 \pm 0.01$
	Dense	34.13	36.25	-2.12 ± 0.13
CIFAR-100	Butterfly R1	8.74	8.66	$+0.08 \pm 0.08$
	Butterfly R8	14.15	8.22	$+5.92 \pm 0.25$
	Butterfly R19	16.94	7.81	$+9.13 \pm 0.24$
	Exponential	14.94	7.89	$+7.05 \pm 0.22$
	Dense	11.23	9.81	$+1.42 \pm 0.13$
Tiny ImageNet	Butterfly R1	3.50	3.51	-0.01 ± 0.01
	Butterfly R8	6.01	3.35	$+2.66 \pm 0.09$
	Butterfly R19	7.07	3.14	$+3.93 \pm 0.12$
	Exponential	6.15	3.15	$+3.00 \pm 0.07$
	Dense	4.82	4.27	$+0.55 \pm 0.05$

C.1.2 Opening comparison with optimizer resets

To examine the role of optimizer-state retention, we train counterparts that retain learned parameters but reset Adam at every task boundary. The fresh reference resets both parameters and optimizer

under both policies. Carry and reset are applied throughout their respective training runs, allowing each policy to shape the subsequent parameter trajectory. Both use the same fixed learning rate, task budget, and aggregation, with ten seeds distributed across three environments for each model and dataset.

Figure 8 and Table 4 show that increasing Butterfly repetition from R1 to R19 improves task-averaged continual accuracy and the continual-minus-fresh gap under optimizer resets. On MNIST, continual accuracy rises from 75.02% to 91.89%, while the gap changes from -8.93 to $+6.85$ percentage points. On CIFAR-10, accuracy rises from 29.88% to 40.31% and the gap from -2.50 to $+7.98$ points. The same pattern holds on CIFAR-100, where accuracy increases from 7.77% to 16.68% and the gap from -0.89 to $+8.87$ points, and on Tiny ImageNet, where accuracy increases from 3.22% to 6.93% and the gap from -0.28 to $+3.79$ points. The improvement with rotation budget therefore persists under both optimizer policies. Adam-state retention affects the magnitude of performance and can change the sign of the gap: CIFAR-100 R1, for example, has a mean gap of $+0.08$ points with carry and -0.89 points with resets. Larger Butterfly budgets nevertheless improve both absolute continual performance and the benefit of accumulated training in either setting.

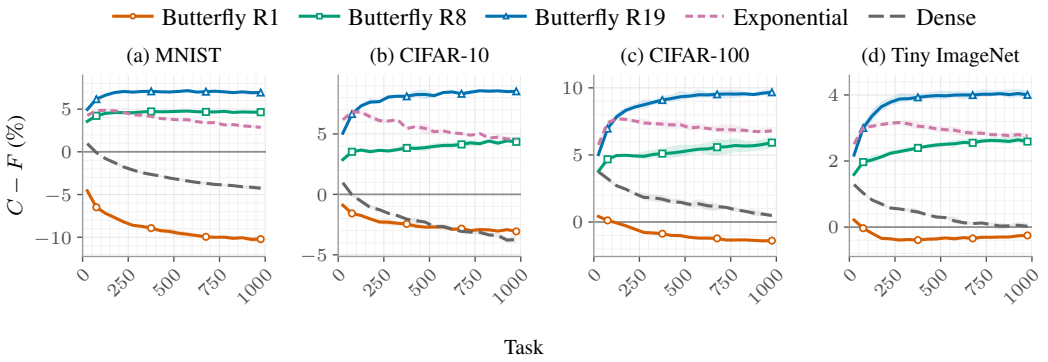

Figure 8: Training-history gaps when the continual learner retains its parameters but resets Adam between tasks. From left to right, the panels show MNIST, CIFAR-10, CIFAR-100, and Tiny ImageNet. The fresh reference resets both parameters and Adam. Each point averages 50 consecutive tasks; shading spans the minimum and maximum of three environment means. Vertical scales vary by dataset. The gap $C - F$, labeled %, is measured in percentage points.

Table 4: Absolute performance when the continual learner resets Adam between tasks. Continual (C) and fresh-reset (F) test accuracies are percentages, and their paired gap $G = C - F$ is in percentage points. Values average tasks 1–1,000 using the same equal-environment weighting as Table 3; the uncertainty on G is one sample SD across three environment means.

Dataset	Model	C (%)	F (%)	G (pp)
MNIST	Butterfly R1	75.02	83.94	-8.93 ± 0.15
	Butterfly R8	89.41	84.83	$+4.58 \pm 0.06$
	Butterfly R19	91.89	85.03	$+6.85 \pm 0.12$
	Exponential	90.90	87.06	$+3.84 \pm 0.11$
	Dense	88.53	91.24	-2.71 ± 0.12
CIFAR-10	Butterfly R1	29.88	32.38	-2.50 ± 0.11
	Butterfly R8	36.63	32.72	$+3.91 \pm 0.16$
	Butterfly R19	40.31	32.32	$+7.98 \pm 0.09$
	Exponential	38.29	32.78	$+5.51 \pm 0.05$
	Dense	34.07	36.25	-2.18 ± 0.01
CIFAR-100	Butterfly R1	7.77	8.66	-0.89 ± 0.10
	Butterfly R8	13.47	8.22	$+5.24 \pm 0.34$
	Butterfly R19	16.68	7.81	$+8.87 \pm 0.24$
	Exponential	14.96	7.89	$+7.07 \pm 0.21$
	Dense	11.42	9.81	$+1.61 \pm 0.16$
Tiny ImageNet	Butterfly R1	3.22	3.51	-0.28 ± 0.03
	Butterfly R8	5.74	3.35	$+2.39 \pm 0.08$
	Butterfly R19	6.93	3.14	$+3.79 \pm 0.12$
	Exponential	6.08	3.15	$+2.93 \pm 0.08$
	Dense	4.65	4.27	$+0.38 \pm 0.03$

C.2 Rotation budgets and pairing schedules

C.2.1 Additional datasets with optimizer carry

The MNIST and Tiny ImageNet comparisons extend the CIFAR results in Section 3.2 to datasets with substantially different absolute accuracies. Butterfly repetition $R \in \{1, 2, 4, 8, 16, 19\}$ is matched to $7R$ Givens rounds, giving the two pairing schedules the same stage and angle counts at each budget. Figure 9 and Table 5 show that task-averaged continual accuracy and the paired gap increase at every tested budget for both schedules.

For Butterfly on MNIST, continual accuracy rises from 78.01% at R1 to 92.43% at R19, while the gap changes from -5.93 to $+7.39$ percentage points. On Tiny ImageNet, the corresponding accuracies are 3.50% and 7.07%, with gaps of -0.01 and $+3.93$ points. The consistent ordering across both pairing schedules extends the budget response beyond the CIFAR datasets. All primary budget comparisons use ten seeds grouped 4/3/3 across three equally weighted environments.

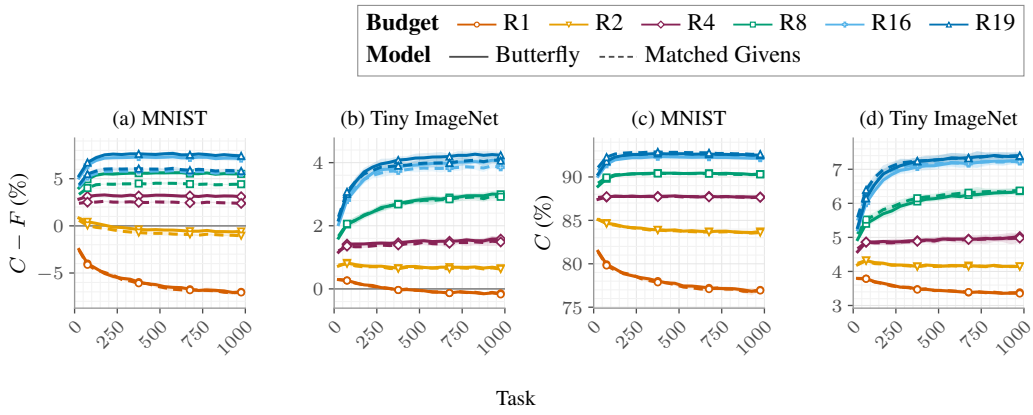

Figure 9: Rotation-budget comparisons on MNIST and Tiny ImageNet with Adam state carried across tasks. The left pair of panels shows continual-minus-fresh-reset gaps; the right pair shows continual test accuracy. Colors identify the budget, and solid and dashed lines distinguish Butterfly from matched Givens. At repetition R, both use $7R$ stages and $448R$ angles per hidden layer. Curves average nonoverlapping 50-task bins; shading shows one sample SD across three environment means. Accuracy is in percent and gaps are in percentage points, despite the % axis label. Each panel uses its own vertical scale.

Table 5: MNIST and Tiny ImageNet results for the matched Butterfly and Givens budgets with continual-optimizer carry. C and F are continual and fresh-reset test accuracy (%); $G = C - F$ is the paired gap in percentage points. All values average tasks 1–1,000 with equal weight across three environment means; the uncertainty on G is their sample SD.

	MNIST			Tiny ImageNet		
Model	C	F	G	C	F	G
Butterfly R1	78.01	83.94	-5.93 ± 0.06	3.50	3.51	-0.01 ± 0.01
Butterfly R2	83.98	84.26	-0.28 ± 0.03	4.18	3.48	$+0.70 \pm 0.02$
Butterfly R4	87.72	84.56	$+3.17 \pm 0.05$	4.91	3.43	$+1.48 \pm 0.04$
Butterfly R8	90.27	84.83	$+5.44 \pm 0.08$	6.01	3.35	$+2.66 \pm 0.09$
Butterfly R16	92.08	85.01	$+7.07 \pm 0.16$	6.92	3.20	$+3.73 \pm 0.10$
Butterfly R19	92.43	85.04	$+7.39 \pm 0.14$	7.07	3.14	$+3.93 \pm 0.12$
Givens R1	77.92	83.93	-6.01 ± 0.03	3.49	3.50	-0.01 ± 0.01
Givens R2	83.93	84.58	-0.65 ± 0.04	4.18	3.50	$+0.68 \pm 0.01$
Givens R4	87.72	85.23	$+2.48 \pm 0.04$	4.90	3.49	$+1.41 \pm 0.08$
Givens R8	90.27	85.91	$+4.36 \pm 0.05$	6.09	3.45	$+2.64 \pm 0.10$
Givens R16	92.26	86.63	$+5.64 \pm 0.12$	6.99	3.37	$+3.63 \pm 0.10$
Givens R19	92.65	86.79	$+5.86 \pm 0.11$	7.13	3.33	$+3.80 \pm 0.10$

C.2.2 ROTATION BUDGETS WITH OPTIMIZER RESETS

We apply the parameter-retention and optimizer-reset policy from Appendix C.1.2 to every matched Butterfly and Givens budget. Figures 10 and 11 and Tables 6 and 7 report the results for all four datasets. For both pairing schedules, task-averaged continual accuracy and the paired gap increase at every step across the six tested budgets.

The same ordering under carry and reset shows that the observed budget response persists when Adam state is reinitialized between tasks. The trajectories complement these averages by showing how the advantage evolves during training: different budgets produce different levels and temporal patterns of performance, even when their full-stream means follow a consistent ordering.

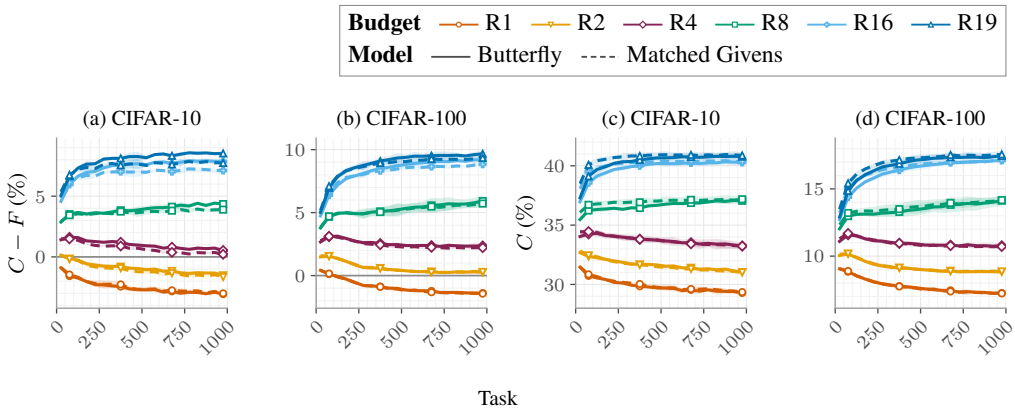

Figure 10: Rotation-budget comparisons on CIFAR-10 and CIFAR-100 when Adam resets between tasks. Continual learners retain their parameters; fresh references reset both parameters and Adam. Gaps are shown in the left pair of panels and continual accuracy in the right pair. Colors identify the budget; solid lines denote Butterfly and dashed lines matched Givens, with $7R$ stages and $448R$ angles per layer in either construction. Curves average nonoverlapping 50-task bins, with one sample SD across three environment means shaded. Accuracy is in percent; gap axes labeled % report percentage points. Vertical scales vary between panels.

Table 6: CIFAR-10 and CIFAR-100 results for the matched rotation budgets with continual-optimizer resets. C and F denote continual and fresh-reset test accuracy (%), and $G = C - F$ is their paired gap in percentage points. Values average tasks 1–1,000 with equal weight across three environment means. The uncertainty on G is one sample SD across those means.

	CIFAR-10			CIFAR-100		
Model	C	F	G	C	F	G
Butterfly R1	29.88	32.38	-2.50 ± 0.11	7.77	8.66	-0.89 ± 0.10
Butterfly R2	31.66	32.57	-0.91 ± 0.11	9.18	8.57	$+0.61 \pm 0.10$
Butterfly R4	33.73	32.72	$+1.01 \pm 0.08$	10.97	8.43	$+2.55 \pm 0.10$
Butterfly R8	36.63	32.72	$+3.91 \pm 0.16$	13.47	8.22	$+5.24 \pm 0.34$
Butterfly R16	39.81	32.46	$+7.35 \pm 0.14$	16.25	7.92	$+8.33 \pm 0.17$
Butterfly R19	40.31	32.32	$+7.98 \pm 0.09$	16.68	7.81	$+8.87 \pm 0.24$
Givens R1	29.96	32.38	-2.42 ± 0.17	7.76	8.67	-0.91 ± 0.10
Givens R2	31.61	32.68	-1.07 ± 0.08	9.20	8.58	$+0.61 \pm 0.08$
Givens R4	33.74	33.03	$+0.71 \pm 0.01$	10.96	8.51	$+2.45 \pm 0.08$
Givens R8	36.96	33.30	$+3.66 \pm 0.13$	13.64	8.46	$+5.18 \pm 0.31$
Givens R16	40.14	33.28	$+6.85 \pm 0.13$	16.46	8.33	$+8.14 \pm 0.15$
Givens R19	40.68	33.20	$+7.47 \pm 0.10$	16.93	8.25	$+8.68 \pm 0.20$

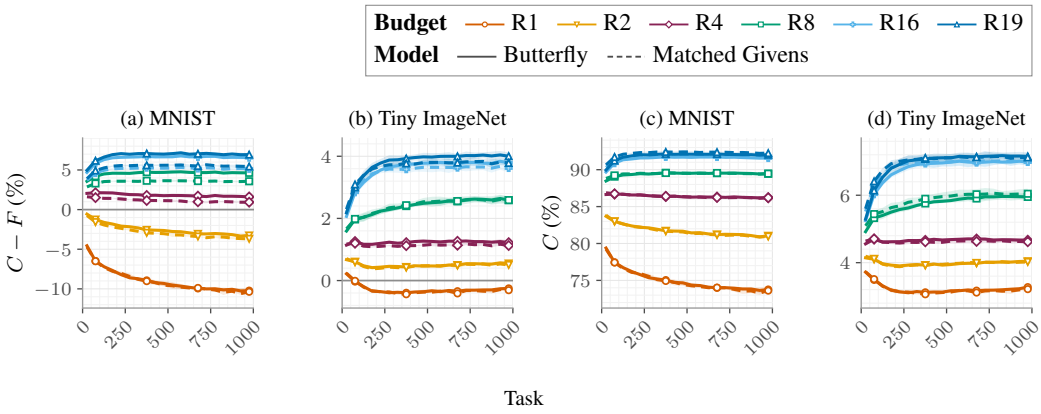

Figure 11: The optimizer-reset budget comparison on MNIST and Tiny ImageNet. The continual learner keeps its parameters between tasks, while the fresh reference resets both parameters and Adam. The left panels show their accuracy gap and the right panels continual accuracy. Butterfly (solid) and matched Givens (dashed) share $7R$ stages and $448R$ angles per layer at each color-coded budget. Curves average nonoverlapping 50-task bins; shading shows one sample SD across three environment means. Accuracy is in percent and gaps are in percentage points, although the gap axes use %. Panels have independent vertical scales.

Table 7: MNIST and Tiny ImageNet results for the matched rotation budgets with continual-optimizer resets. Continual (C) and fresh-reset (F) test accuracies are percentages; their paired gap $G = C - F$ is in percentage points. Values average tasks 1–1,000 with equal weight across three environment means, and the uncertainty on G is their sample SD.

	MNIST			Tiny ImageNet		
Model	C	F	G	C	F	G
Butterfly R1	75.02	83.94	-8.93 ± 0.15	3.22	3.51	-0.28 ± 0.03
Butterfly R2	81.76	84.26	-2.50 ± 0.07	3.99	3.48	$+0.50 \pm 0.02$
Butterfly R4	86.39	84.56	$+1.83 \pm 0.04$	4.67	3.43	$+1.23 \pm 0.03$
Butterfly R8	89.41	84.83	$+4.58 \pm 0.06$	5.74	3.35	$+2.39 \pm 0.08$
Butterfly R16	91.49	85.01	$+6.49 \pm 0.10$	6.77	3.20	$+3.58 \pm 0.09$
Butterfly R19	91.89	85.03	$+6.85 \pm 0.12$	6.93	3.14	$+3.79 \pm 0.12$
Givens R1	74.92	83.93	-9.02 ± 0.11	3.20	3.50	-0.30 ± 0.01
Givens R2	81.70	84.58	-2.88 ± 0.07	3.99	3.50	$+0.49 \pm 0.02$
Givens R4	86.39	85.23	$+1.16 \pm 0.04$	4.63	3.49	$+1.14 \pm 0.03$
Givens R8	89.44	85.91	$+3.54 \pm 0.05$	5.85	3.45	$+2.40 \pm 0.11$
Givens R16	91.72	86.63	$+5.09 \pm 0.10$	6.85	3.37	$+3.49 \pm 0.11$
Givens R19	92.21	86.79	$+5.42 \pm 0.10$	6.97	3.33	$+3.64 \pm 0.09$

C.2.3 BUTTERFLY R19 AND R32 WITH MATCHED REPLICATION

To examine larger Butterfly budgets, we compare R19 with R32, which uses 224 stages and 14,336 angles per hidden layer. Both models use the same four seeds in one environment. The R19 observations are the corresponding subset of the primary budget comparison, so the two models are compared under matched replication. This comparison extends the Butterfly budget range; the matched Butterfly–Givens comparisons cover R1–R19.

Figures 12 and 13 and Table 8 show higher task-averaged continual accuracy and a larger paired gap for R32 on all four datasets. Relative to matched R19, continual accuracy increases by 0.84 percentage points on MNIST, 1.00 on CIFAR-10, 0.91 on CIFAR-100, and 0.18 on Tiny ImageNet. The corresponding gap increases are 0.79, 1.60, 1.36, and 0.40 points. These outcomes show continued gains in mean performance beyond R19, with their magnitude depending on the dataset. On MNIST, fresh accuracy changes little, so the larger gap closely follows the improvement in continual accuracy. On both CIFAR datasets and Tiny ImageNet, fresh accuracy decreases, contributing to the larger gaps alongside the improvement in continual performance. Bands and reported standard deviations describe variation across the four seed averages. R19 already stores 8,512 angles per hidden layer, exceeding the 8,128-dimensional rotation space. Crossing this parameter-count threshold does not by itself establish full coverage of $\mathrm{SO}(128)$. Even when all local rotation directions are available, the parameterization determines how readily angle updates move along them, as described by Eq. 3. We hypothesize that the additional rotations in R32 improve access to useful directions by changing their scaling and providing alternative combinations of angle updates that produce similar weight changes.

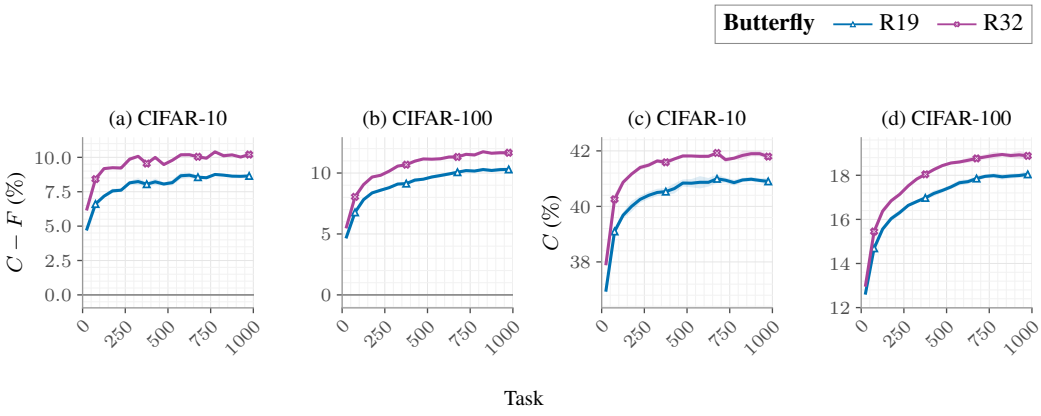

Figure 12: Butterfly R19 and R32 on CIFAR-10 and CIFAR-100 with continual-optimizer carry. The left panels compare continual-minus-fresh-reset gaps and the right panels continual test accuracy. Both models use the same four seeds and task environment. Curves average nonoverlapping 50-task bins; shading shows one sample SD across seeds. Accuracy is in percent; gaps labeled % are in percentage points. Vertical scales vary between panels.

Table 8: Matched R19/R32 results with continual-optimizer carry, averaged over tasks 1–1,000. C and F are continual and fresh-reset test accuracy (%); $G = C - F$ is the paired gap in percentage points. Both budgets use the same four seeds in one environment, and the uncertainty on G is one sample SD across seed averages.

Dataset	Model	C	F	G
CIFAR-10	R19	40.40	32.38	$+8.02 \pm 0.12$
	R32	41.40	31.79	$+9.62 \pm 0.06$
CIFAR-100	R19	16.94	7.78	$+9.16 \pm 0.09$
	R32	17.84	7.33	$+10.52 \pm 0.05$
MNIST	R19	92.47	85.21	$+7.26 \pm 0.12$
	R32	93.31	85.26	$+8.05 \pm 0.01$
Tiny ImageNet	R19	7.09	3.18	$+3.91 \pm 0.04$
	R32	7.27	2.96	$+4.31 \pm 0.01$

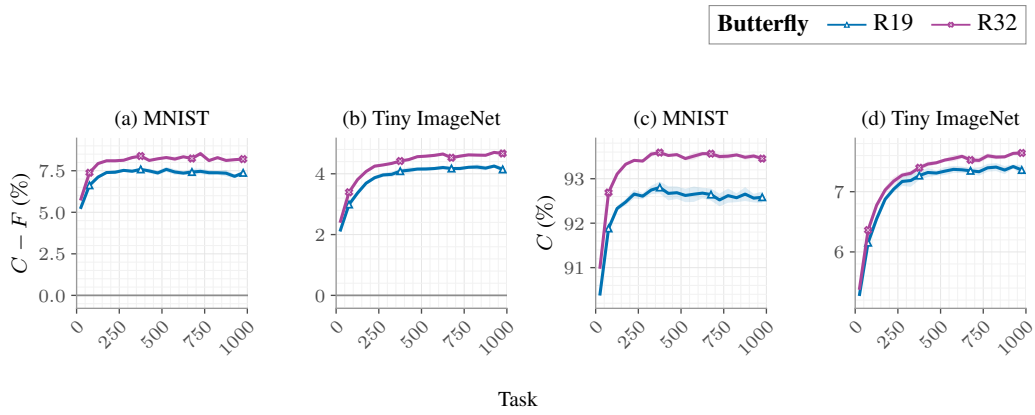

Figure 13: Butterfly R19 and R32 on MNIST and Tiny ImageNet with continual-optimizer carry. Continual-minus-fresh-reset gaps appear on the left and continual test accuracy on the right. Curves average nonoverlapping 50-task bins for the same four seeds in one environment; shading shows one sample SD across seeds. Accuracy is in percent and gaps are in percentage points, despite the % axis label. Each panel uses its own vertical scale.

C.3 Coordinate coupling

The block-Givens comparison complements the rotation-budget experiments by holding the number of stored angles fixed while varying how coordinates interact. Each hidden layer uses 133 rounds and 8,512 angles, with block sizes ranging from 8 to 128 and a common fixed orthogonal change of basis. Under the trainable-classifier protocol with continual-optimizer carry, broader coupling improves both mean continual accuracy and the paired gap on CIFAR-100. Figure 3 and Table 2 report the trajectories and task-averaged outcomes, while Appendix A specifies the pairing construction and change of basis. The frozen-classifier experiments examine this response with learning-rate selection and measurements of the trained networks' sensitivities. Appendices D.1.3 and D.1.4 report the CIFAR-100 Adam sweep and selected-rate diagnostics, respectively; Appendix D.1.5 reports the CIFAR-10 SGD sweep and diagnostics. Each comparison specifies its selection window, reporting horizon, and replication structure.

D Optimization and initialization

This appendix examines how optimization choices affect adaptation under orthogonal hidden weights. Appendix D.1 compares learning rates across rotation budgets and block sizes, then relates the selected-rate results to parameter and input sensitivities. Appendix D.2 matches initial functions across rotation budgets to examine the contribution of parameterization beyond initialization. Appendix D.3 studies constraints on the optimization coordinates of the exponential parameterization.

D.1 Learning-rate sensitivity

The random-projection comparisons show improvements in continual accuracy with increased rotation budget and broader coordinate coupling after learning-rate selection. We examine these improvements alongside continual-minus-fresh gaps, since learning rate affects both the continual learner and its fresh reference. The experiments begin with Butterfly networks on random-projection inputs, followed by block-Givens networks with a fixed number of angles, and conclude with Butterfly networks on VAE representations. All comparisons in this section use frozen classifiers, one epoch per task, batch size 128, and four model/head seeds in a shared environment. Learning rates are selected by continual validation accuracy and then shared with the fresh references. The random-projection Adam comparisons select over tasks 901–1,000; the CIFAR-10 block-Givens SGD and VAE comparisons use a 50-task tuning stream. Reported test summaries average tasks 1–1,000. Gaps are formed within each task and seed before averaging, and reported standard deviations describe variation across the four seed means.

D.1.1 Butterfly: complete random-projection learning-rate comparison

We evaluate Adam rates $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$ using frozen random input projections and frozen classifiers. The full search covers Butterfly budgets R1, R2, R4, R8, R16, R19, and R32. Figure 14 displays the common-rate comparisons for R1–R19; the selected-rate results below also include R32. Every rate–budget combination runs for 1,000 tasks. The continual learner carries its trunk parameters and Adam state between tasks, while fresh-reset reinitializes both. Each heatmap column compares rotation budgets at the same learning rate. Selecting a rate suited to the budget improves continual accuracy while retaining a positive advantage over fresh-reset. For CIFAR-10 R19, using 0.001 rather than 0.01 raises mean continual accuracy from 29.54% to 31.49%, with fresh-reset gaps of 5.33 and 3.08 percentage points, respectively. For CIFAR-100 R19, using 0.001 rather than 0.005 raises accuracy from 4.79% to 4.95%, with gaps of 2.37 and 1.83 points. These comparisons motivate selecting rates by continual validation accuracy and reporting both absolute accuracy and the fresh-reference gap.

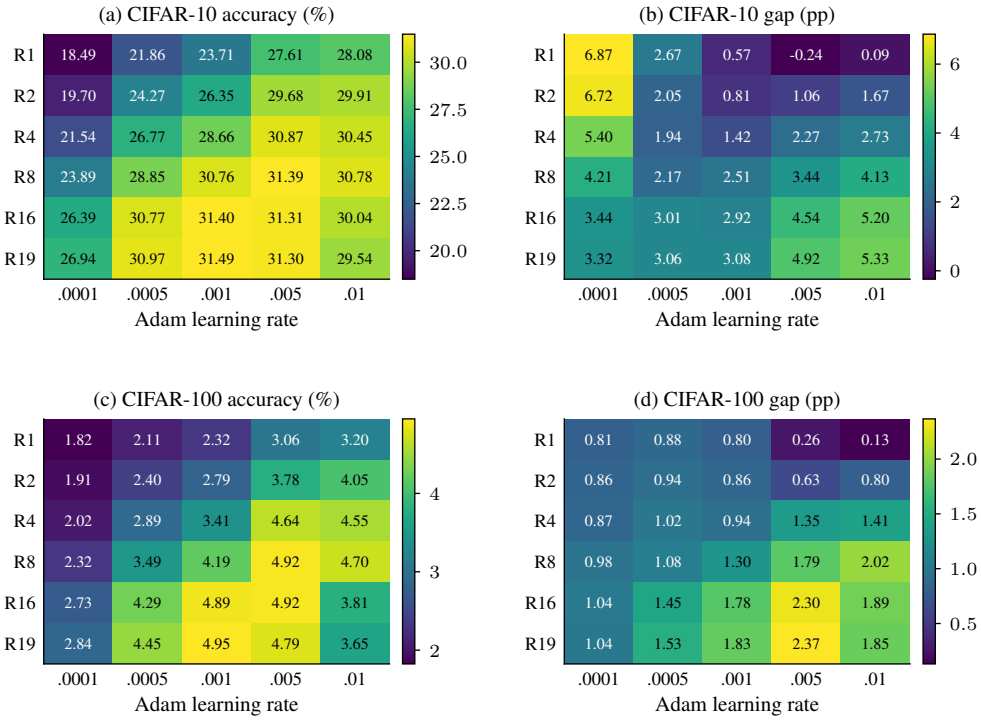

Figure 14: Butterfly learning-rate sensitivity with frozen random projections and classifiers. Rows show CIFAR-10 and CIFAR-100; columns show continual test accuracy (%) and its gap over fresh-reset (percentage points). Each cell averages tasks 1–1,000 within each of four seeds and then across seeds. The heatmaps use separate color scales.

D.1.2 BUTTERFLY: SELECTED-RATE CIFAR-10 RESULTS AND DIAGNOSTICS

For each budget, we select the rate by averaging continual validation accuracy over tasks 901–1,000 within each seed and then across seeds, resolving ties in favor of the smaller rate. Both datasets select 10^{-2} for R1 and R2, 5×10^{-3} for R4 and R8, and 10^{-3} for R16, R19, and R32. R1 and R2 select the largest rate tested. Table 9 reports test outcomes over all 1,000 tasks, including the tasks used for validation-based selection. On CIFAR-10, mean continual accuracy rises from 28.08% at R1 to 31.49% at R19, and the fresh-reset gap rises from 0.09 ± 0.12 to 3.08 ± 0.38 percentage points. R32 reaches 31.87% continual accuracy with a gap of 3.96 ± 0.23 points. The intermediate budgets illustrate why the two measures are useful together: R8 achieves 31.39% continual accuracy and a gap of 3.44 ± 0.14 points. R19 therefore has slightly higher accuracy but a smaller gap than R8 because its fresh reference also performs better.

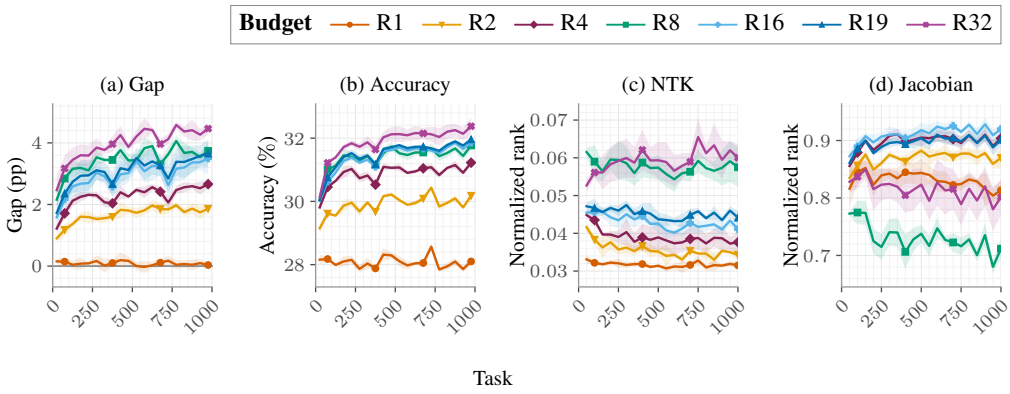

Figure 15: Butterfly performance and sensitivity on CIFAR-10 at the selected Adam rates. From left to right: fresh-reset gap (percentage points), continual test accuracy (%), normalized NTK effective rank, and normalized trunk-Jacobian effective rank. Performance is averaged in 50-task bins; diagnostics are measured every 50 tasks. Curves show four-seed means, with bands of one sample standard deviation.

Post-hoc NTK and Jacobian measurements. We measure parameter and input sensitivities at 20 continual-model checkpoints, after tasks 50, 100, ..., 1,000. Seven budgets and four seeds give 560 checkpoint measurements per dataset. The trunk Jacobian measures sensitivity to the 128-dimensional projected input. It includes the trunk's ReLU nonlinearities and excludes the input projection and classifier. We evaluate it on the first 64 training images after applying the checkpoint task's pixel permutation and frozen projection. Image identities and task transformations are shared across budgets. The empirical logit NTK measures sensitivity to trainable trunk parameters, including biases, by differentiating the logits through the frozen classifier. Its parameter Jacobian excludes the frozen projection and classifier parameters and uses no optimizer preconditioning. The CIFAR-10 kernel uses 64 images and ten logits per image, giving 640 logit rows; the CIFAR-100 kernel uses 16 images and 100 logits per image, giving 1,600 rows. Normalization is relative to the corresponding kernel dimension.

For both diagnostics, normalized effective rank is $r_{\text{eff}} = \exp[-\sum_i p_i \log p_i]/d$. For the trunk Jacobian, $p_i = s_i^2 / \sum_j s_j^2$, where s_i are its singular values, and $d = 128$. We average the resulting ranks across the 64 probe images. For the NTK, p_i are normalized eigenvalues after clipping negative numerical eigenvalues to zero, and d is the number of logit rows. These measures describe how broadly sensitivity is distributed across spectral directions; they differ from algebraic rank and from the rank of the parameterization differential studied in Appendix A. Summary values average the 20 task-end checkpoints within each seed and then across seeds. The two diagnostics capture different aspects of the trained model. On CIFAR-10, R8 has a higher mean normalized NTK effective rank than R19, 0.058 versus 0.045, but a lower trunk-Jacobian effective rank, 0.730 versus 0.896. Thus, sensitivity to parameter changes and sensitivity to input changes can have different budget orderings. Figure 15 places both alongside accuracy and the fresh-reset gap to show how these properties vary over the task stream.

Table 9: Butterfly test performance at the Adam rate selected for each budget using continual validation accuracy over tasks 901–1,000. C and F are continual and fresh-reset accuracy (%), averaged over all 1,000 tasks and four seeds. $G = C - F$ is the paired gap in percentage points; its uncertainty is one sample standard deviation across seed means.

Dataset	R	LR	C	F	G
CIFAR-10	1	0.01	28.08	27.99	$+0.09 \pm 0.12$
CIFAR-10	2	0.01	29.91	28.24	$+1.67 \pm 0.10$
CIFAR-10	4	0.005	30.87	28.60	$+2.27 \pm 0.10$
CIFAR-10	8	0.005	31.39	27.94	$+3.44 \pm 0.14$
CIFAR-10	16	0.001	31.40	28.48	$+2.92 \pm 0.34$
CIFAR-10	19	0.001	31.49	28.41	$+3.08 \pm 0.38$
CIFAR-10	32	0.001	31.87	27.91	$+3.96 \pm 0.23$
CIFAR-100	1	0.01	3.20	3.07	$+0.13 \pm 0.19$
CIFAR-100	2	0.01	4.05	3.24	$+0.80 \pm 0.17$
CIFAR-100	4	0.005	4.64	3.29	$+1.35 \pm 0.10$
CIFAR-100	8	0.005	4.92	3.13	$+1.79 \pm 0.04$
CIFAR-100	16	0.001	4.89	3.11	$+1.78 \pm 0.09$
CIFAR-100	19	0.001	4.95	3.12	$+1.83 \pm 0.09$
CIFAR-100	32	0.001	5.15	3.07	$+2.08 \pm 0.07$

D.1.3 BLOCK-GIVENS: FROZEN-CLASSIFIER CIFAR-100 SWEEP

The block-Givens comparison varies the block size $b \in \{8, 16, 32, 64, 128\}$ while holding the number of trainable angles at 8,512 per hidden layer. Inputs use frozen random projections, and the classifier remains frozen. We evaluate all 25 combinations of block size and the five Adam rates used above for 1,000 tasks. The continual learner carries its trunk and Adam state; fresh-reset reinitializes both on each task. Table 10 reports continual validation accuracy over tasks 901–1,000. All five block sizes select 10^{-3}. Table 11 and Figure 5 report test accuracy and paired gaps over all 1,000 tasks.

Table 10: Adam learning-rate selection for block-Givens on CIFAR-100 with frozen random projections and classifiers. Entries give mean continual validation accuracy (%) over tasks 901–1,000 and four seeds. Every block size selects 10^{-3}.

b	10^{-4}	5×10^{-4}	10^{-3}	5×10^{-3}	10^{-2}
8	2.779	3.499	3.637	2.775	1.000
16	2.996	4.159	4.330	3.570	2.408
32	3.049	4.657	4.881	4.245	2.951
64	3.139	4.999	5.281	4.706	3.412
128	3.155	5.060	5.445	5.100	3.711

Table 11: Block-Givens test performance on CIFAR-100 at the selected Adam rate 10^{-3}. C and F_r denote continual and fresh-reset accuracy (%), and $G = C - F_r$ is their paired gap in percentage points. Entries report the mean and sample standard deviation across four seeds after averaging tasks 1–1,000 within each seed.

b	Selected LR	C	F_r	G
8	0.001	3.41 ± 0.26	2.84 ± 0.24	0.57 ± 0.10
16	0.001	4.03 ± 0.26	3.19 ± 0.28	0.84 ± 0.11
32	0.001	4.51 ± 0.29	3.42 ± 0.30	1.09 ± 0.11
64	0.001	4.86 ± 0.34	3.55 ± 0.32	1.31 ± 0.12
128	0.001	5.01 ± 0.34	3.61 ± 0.33	1.40 ± 0.08

At this shared rate, both continual accuracy and the fresh-reset gap increase across the five block sizes. The comparison therefore connects broader coordinate coupling with improved adaptation while holding the angle count and learning rate fixed. The following diagnostics examine how parameter and input sensitivities vary across these models.

D.1.4 BLOCK-GIVENS: CIFAR-100 DIAGNOSTICS

We evaluate the selected-rate continual models after tasks 50, 100, . . ., 1,000, giving 400 checkpoint measurements across five block sizes and four seeds. The NTK uses 16 training images and 100

logits per image; the trunk Jacobian uses 64 images. Both follow the definitions and normalization in Appendix D.1.2. Larger blocks have higher normalized NTK effective rank, with little separation among $b \geq 32$ late in training. At task 1,000, the mean is 0.065 for $b = 8$ and 0.078 for $b = 128$, reaching its highest value of 0.079 at $b = 64$. Trunk-Jacobian effective rank increases across all five sizes, from 0.522 to 0.810. The distinction is clearest among the larger blocks: the NTK effective rank levels off, while the trunk Jacobian continues to distribute sensitivity more broadly across input directions. Figure 16 shows these diagnostics alongside the accuracy and gap trajectories.

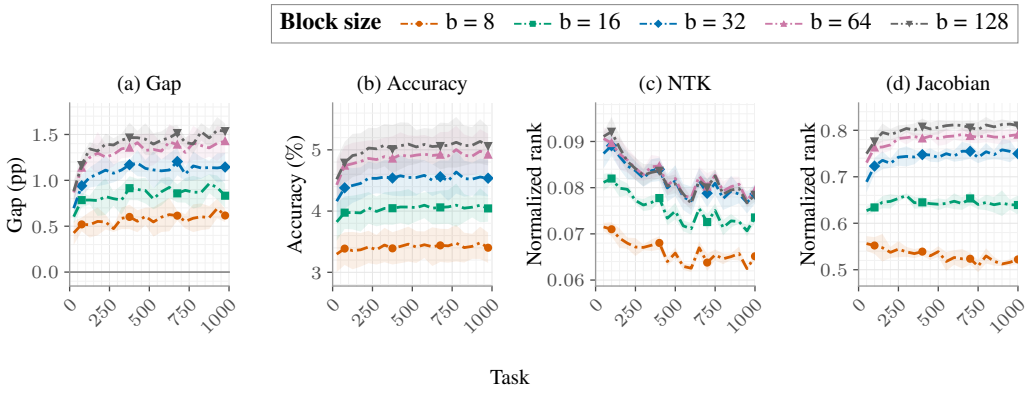

Figure 16: Block-Givens performance and sensitivity on CIFAR-100 at Adam rate 10^{-3}. From left to right: fresh-reset gap (percentage points), continual test accuracy (%), normalized NTK effective rank, and normalized trunk-Jacobian effective rank. Performance is averaged in 50-task bins; diagnostics are measured every 50 tasks from the continual models. Curves show four-seed means, with bands of one sample standard deviation.

D.1.5 Block-Givens: CIFAR-10 SGD sweep and diagnostics

The CIFAR-10 comparison uses SGD with momentum 0.99, frozen random projections and classifiers, and the same five block sizes with 8,512 angles per hidden layer. We evaluate the five candidate rates on a 50-task tuning stream. Selection averages continual validation accuracy over tasks 1–50 within each seed and then across seeds, with ties favoring the smaller rate. The selected rate is 10^{-3} for $b = 8$ and 5×10^{-3} for the other four sizes (Table 12). The selected-rate runs start from their initial parameters and empty optimizer state and train for 1,000 tasks. Their first 50 tasks reuse the tuning permutations and minibatch orders and are included in the reported averages. The continual learner carries its trunk and momentum between tasks; the reported gaps use fresh-reset, which reinitializes both.

Table 12: SGD learning-rate selection for block-Givens on CIFAR-10 with frozen random projections and classifiers. Entries give mean continual validation accuracy (%) over the 50-task tuning stream and four seeds. Block size $b = 8$ selects 10^{-3}; the other sizes select 5×10^{-3}.

b	10^{-4}	5×10^{-4}	10^{-3}	5×10^{-3}	10^{-2}
8	18.37	24.80	26.02	25.85	25.28
16	18.44	25.55	26.95	27.14	26.58
32	18.60	25.75	27.40	27.67	27.11
64	18.68	25.93	27.71	28.09	27.37
128	18.79	26.18	27.94	28.23	27.58

Over the full task stream, continual accuracy increases from 26.68% at $b = 8$ to 29.24% at $b = 128$, while fresh-reset accuracy increases from 23.86% to 27.84%. The gap is positive at every block size and largest at $b = 8$, 2.83 ± 0.52 percentage points, which selects the smaller rate. Among the four sizes sharing 5×10^{-3}, the gap increases from 1.15 ± 0.24 to 1.40 ± 0.25 points. Broader coupling therefore improves absolute continual accuracy throughout this comparison, while the relative advantage over fresh-reset also depends on the fresh learner's response.

The diagnostics use the same 20 task-end checkpoints and four seeds per block, giving 400 measurements. Both use 64 training images, producing 640 logit rows for the NTK. At task 1,000,

Table 13: Block-Givens test performance on CIFAR-10 at the selected SGD rates. C and F_r denote continual and fresh-reset accuracy (%), and $G = C - F_r$ is their paired gap in percentage points. Entries report the mean and sample standard deviation across four seeds after averaging tasks 1–1,000 within each seed.

b	Selected LR	C	F_r	G
8	0.001	26.68 ± 0.13	23.86 ± 0.54	2.83 ± 0.52
16	0.005	28.09 ± 0.04	26.94 ± 0.25	1.15 ± 0.24
32	0.005	28.75 ± 0.04	27.48 ± 0.29	1.27 ± 0.27
64	0.005	29.09 ± 0.06	27.77 ± 0.28	1.31 ± 0.24
128	0.005	29.24 ± 0.05	27.84 ± 0.28	1.40 ± 0.25

normalized NTK effective rank is 0.036 for $b = 8$ and 0.045–0.050 for the larger blocks. Trunk-Jacobian effective rank is highest at $b = 8$, 0.783, compared with 0.722–0.758 for the other sizes. At their selected rates, the larger blocks therefore distribute parameter sensitivity more broadly, whereas $b = 8$ has the most broadly distributed input sensitivity (Figure 17).

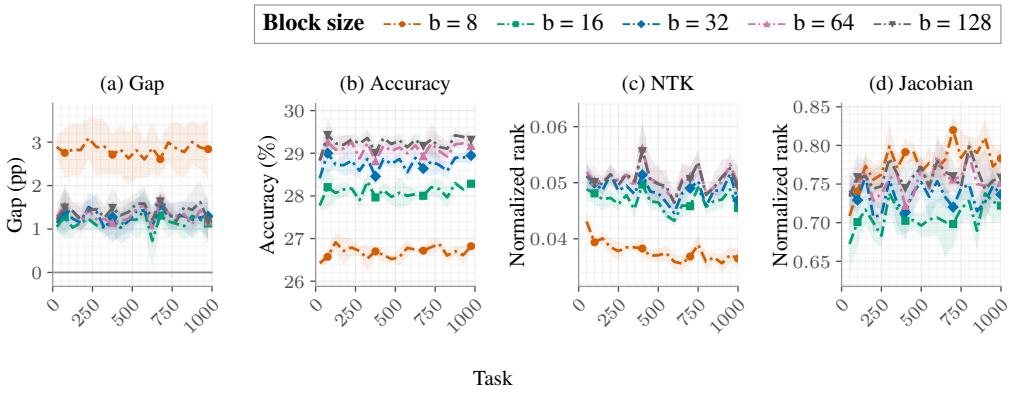

Figure 17: Block-Givens performance and sensitivity on CIFAR-10 at the selected SGD rates. From left to right: fresh-reset gap (percentage points), continual test accuracy (%), normalized NTK effective rank, and normalized trunk-Jacobian effective rank. Performance is averaged in 50-task bins; diagnostics are measured every 50 tasks. Curves show four-seed means, with bands of one sample standard deviation.

D.1.6 Butterfly: selected-rate comparisons on VAE representations

We next compare Butterfly budgets R1, R2, R4, R8, R16, and R19 using frozen 128-dimensional ResNet-VAE posterior means, latent-coordinate permutations, and frozen classifiers. Each task uses one epoch over 45,000 training images. Batch size 128 and a masked final minibatch give 352 updates per task; validation and test contain 5,000 and 10,000 images, respectively. These experiments extend the comparison to learned representations and to both Adam and SGD. For each dataset, budget, and optimizer, we select among the same five learning rates using continual validation accuracy over a 50-task tuning stream. Scores are averaged over tasks within each of four seeds and then across seeds, with ties favoring the smaller rate. The selected rate is shared by all three roles. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$; SGD uses momentum 0.99 without Nesterov acceleration or dampening.

Selected-rate runs restart from the original initialization and empty optimizer state and train for 1,000 tasks at a constant rate. Their first 50 tasks reuse the tuning permutations and minibatch orders and remain included in the reported averages. Continual carries its trunk and optimizer state between tasks. Fresh-reset reinitializes both, while fresh-carry receives the same fresh trunk initialization and retains its own optimizer state. All three roles share the frozen classifier. For R1, R2, R4, R8, R16, and R19, respectively, the selected CIFAR-10 Adam rates are $(0.005, 0.005, 0.01, 0.005, 0.005, 0.001)$, and the SGD rates are $(0.01, 0.01, 0.005, 0.01, 0.01, 0.01)$. On CIFAR-100, Adam

selects $(0.01, 0.01, 0.01, 0.005, 0.005, 0.005)$, while SGD selects 0.01 for every budget. The rate 0.01 is the largest tested.

Across rotation budgets and both optimizers, continual accuracy largely reaches an early plateau on the frozen VAE representations (Figure 18). A possible contributor is the concentration of variation in these features: across the training images, 106 of 128 latent coordinates on CIFAR-10 and 102 of 128 on CIFAR-100 have variance below 0.1. Thus, more than two-thirds of the coordinates vary little in both datasets, which may help explain the early saturation. Figures 19 and 20 show the corresponding gaps against fresh-reset and fresh-carry.

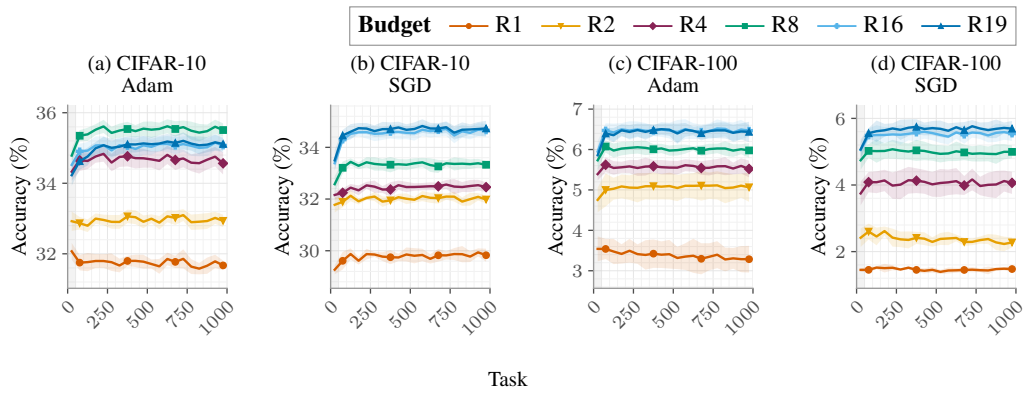

Figure 18: Butterfly continual test accuracy on frozen VAE representations. The first two panels show CIFAR-10 with Adam and SGD; the last two show CIFAR-100 in the same optimizer order. Curves show four-seed means of 50-task averages, with bands of one sample standard deviation. Gray shading marks the first 50 tasks, which reuse the tuning permutations and minibatch orders.

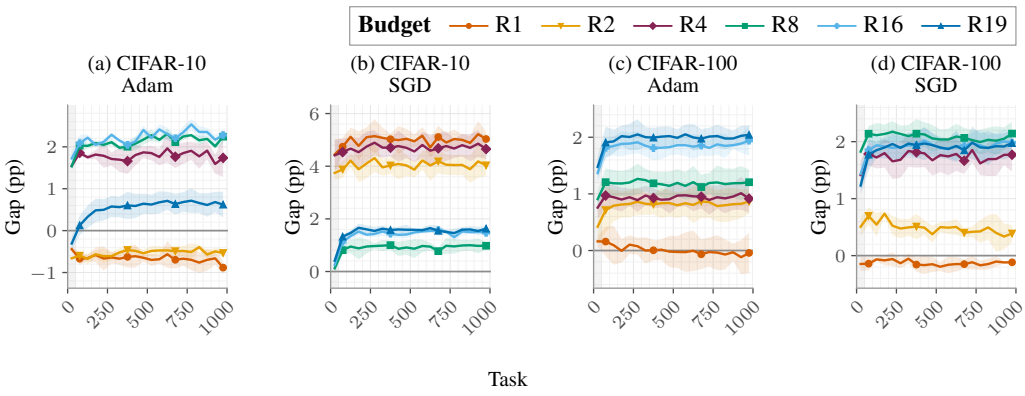

Figure 19: Butterfly accuracy gaps over fresh-reset on frozen VAE representations, in percentage points. Panels show CIFAR-10 Adam, CIFAR-10 SGD, CIFAR-100 Adam, and CIFAR-100 SGD. Curves show four-seed means of paired gaps averaged in 50-task bins; bands show one sample standard deviation. Gray shading marks the first 50 tasks, which reuse the tuning permutations and minibatch orders.

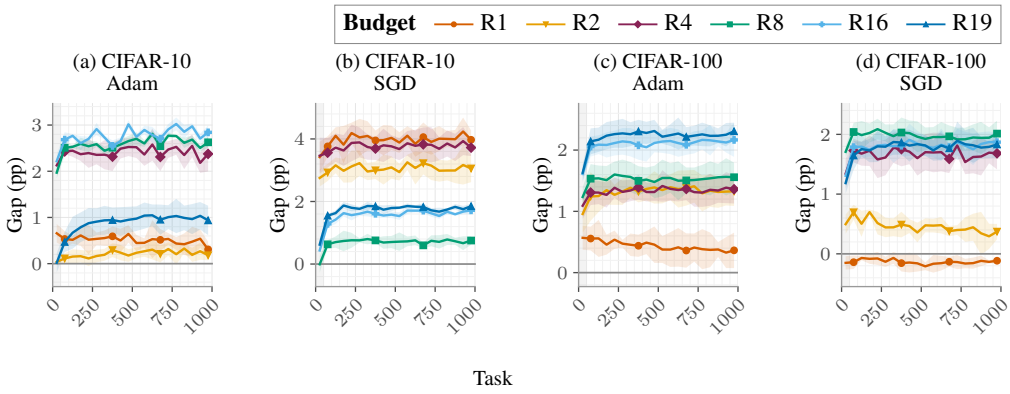

Figure 20: Butterfly accuracy gaps over fresh-carry on frozen VAE representations, in percentage points. Panels show CIFAR-10 Adam, CIFAR-10 SGD, CIFAR-100 Adam, and CIFAR-100 SGD. Curves show four-seed means of paired gaps averaged in 50-task bins; bands show one sample standard deviation. Gray shading marks the first 50 tasks, which reuse the tuning permutations and minibatch orders.

D.2 Matched initialization

D.2.1 Initialization and training protocol

Matching the initial function allows us to examine how rotation budget affects adaptation when networks begin with identical predictions. We extend the frozen-classifier CIFAR-100 comparison in Section 4.2 to CIFAR-10 and to trainable classifiers, and compare matched and ordinary initialization at the same budget and learning rate. For each hidden layer, matched R16 and R19 retain an R8 initialization and append zero-angle identity blocks. All added angles remain trainable. Shared initial biases and classifier parameters give the models the same initial function within each comparison. This construction applies both when initializing the continual learner and when drawing fresh parameters for each task. The shared starting function permits different parameter sensitivities and subsequent optimization trajectories. Both classifier settings follow the CIFAR random-projection protocol and four-seed reporting conventions in Appendix B. Frozen-classifier comparisons use the continual, fresh-reset, and fresh-carry roles defined there.

CIFAR-100 uses a common learning rate of 0.001. On CIFAR-10, R8 uses 0.005 and matched R16/R19 use 0.001, transferring the rates selected for ordinary initialization in Appendix D.1 without another search. Table 14 also includes CIFAR-10 R8 at 0.001 and ordinary-initialization controls at the same repetition and rate. The frozen CIFAR-100 results complement Section 4.2 by showing that the larger-budget advantage persists against fresh-carry: the corresponding gaps are 1.43, 1.95, and 2.01 percentage points for R8, matched R16, and matched R19. At fixed budget and rate, matched R16/R19 continual accuracies differ from ordinary initialization by less than 0.02 points.

Table 14: Test performance with frozen classifiers under matched and ordinary initialization. C, F_r, and F_c denote continual, fresh-reset, and fresh-carry accuracy (%); $G_r = C - F_r$ and $G_c = C - F_c$ are paired gaps in percentage points. Values average tasks 1–1,000 within each seed and report the mean $\pm$ one sample standard deviation across four seeds in one environment per dataset. C100 and C10 denote CIFAR-100 and CIFAR-10. Dashes indicate that fresh-carry results are unavailable.

Data	Model	Initialization	LR	C	F_r	F_c	G_r	G_c
C100	R8	R8	0.001	4.19 ± 0.31	2.88 ± 0.27	2.76 ± 0.25	1.30 ± 0.07	1.43 ± 0.10
C100	R16	matched	0.001	4.88 ± 0.30	3.08 ± 0.29	2.93 ± 0.27	1.80 ± 0.09	1.95 ± 0.10
C100	R16	ordinary	0.001	4.89 ± 0.32	3.11 ± 0.29	–	1.78 ± 0.09	–
C100	R19	matched	0.001	4.94 ± 0.28	3.09 ± 0.29	2.93 ± 0.26	1.84 ± 0.05	2.01 ± 0.06
C100	R19	ordinary	0.001	4.95 ± 0.22	3.12 ± 0.29	–	1.83 ± 0.09	–
C10	R8	R8	0.001	30.76 ± 0.06	28.25 ± 0.43	–	2.51 ± 0.43	–
C10	R8	R8	0.005	31.39 ± 0.13	27.94 ± 0.19	27.49 ± 0.21	3.44 ± 0.14	3.89 ± 0.14
C10	R16	matched	0.001	31.38 ± 0.08	28.40 ± 0.38	27.86 ± 0.41	2.98 ± 0.31	3.51 ± 0.34
C10	R16	ordinary	0.001	31.40 ± 0.05	28.48 ± 0.38	–	2.92 ± 0.34	–
C10	R19	matched	0.001	31.45 ± 0.11	28.30 ± 0.38	27.78 ± 0.41	3.15 ± 0.30	3.67 ± 0.34
C10	R19	ordinary	0.001	31.49 ± 0.03	28.41 ± 0.37	–	3.08 ± 0.38	–

D.2.2 Frozen-classifier CIFAR-10

At the transferred rates, matched R16 and R19 reach mean continual accuracies of 31.38% and 31.45%, close to R8's 31.39% at 0.005. Their fresh-reset gaps are smaller, at 2.98 and 3.15 percentage points compared with R8's 3.44, because their fresh references also perform better. At the common rate of 0.001, R8 instead reaches 30.76% continual accuracy with a 2.51-point gap, below both matched larger models. These comparisons distinguish performance at the transferred rates from the budget comparison at a common rate. Matching initialization changes mean continual accuracy by less than 0.04 points relative to ordinary initialization at the same budget and rate. The two sensitivity measures give different orderings in Figure 21. Averaged over the 20 diagnostic checkpoints, normalized NTK effective rank is 0.058 for R8 at 0.005 and 0.043/0.045 for matched R16/R19 at 0.001. Normalized trunk-Jacobian effective rank instead increases from 0.730 to 0.909/0.899. At these rates, similar continual accuracies therefore coexist with different distributions of sensitivity to parameter and input changes.

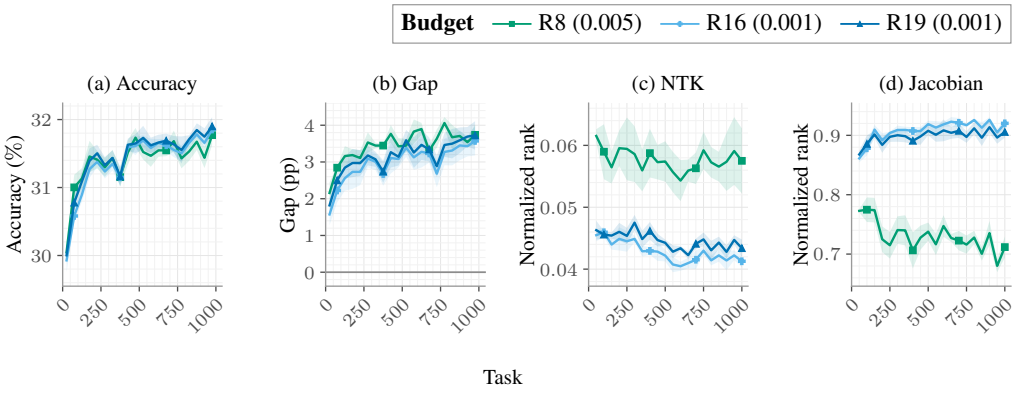

Figure 21: Matched initialization on CIFAR-10 with a frozen classifier. From left to right: continual test accuracy (%), its gap over fresh-reset (percentage points), normalized NTK effective rank, and normalized trunk-Jacobian effective rank. R8 uses learning rate 0.005; matched R16/R19 use 0.001. Performance curves average nonoverlapping 50-task bins, while diagnostics measure the continual learner every 50 tasks. Curves show means across four seeds in one environment, with bands of one sample standard deviation.

D.2.3 Trainable-classifier comparisons

With a trainable classifier, the continual learner retains the learned trunk, classifier, and Adam state between tasks. Its fresh-reset reference reinitializes the complete trainable network and optimizer for each task, using the same R8-matched construction for the larger trunks. The input projection

remains frozen. CIFAR-100 uses a common learning rate of 0.001. CIFAR-10 uses 0.005 for matched R16 and 0.001 for matched R19, with R8 references at both rates. These rates are retained without retuning for matched initialization. Table 15 includes both same-rate R8 references and ordinary-initialization controls.

Table 15: Test performance with trainable classifiers under matched and ordinary initialization. C and F_r are continual and fresh-reset test accuracies (%), and $G_r = C - F_r$ is their paired gap in percentage points. Entries summarize tasks 1–1,000 as the mean ± one sample standard deviation across four seed averages in one environment per dataset. C100 and C10 denote CIFAR-100 and CIFAR-10.

Data	Model	Initialization	LR	C	F_r	G_r
C100	R8	R8	0.001	14.08 ± 0.10	8.20 ± 0.01	5.88 ± 0.10
C100	R16	matched	0.001	16.49 ± 0.12	7.83 ± 0.01	8.66 ± 0.12
C100	R16	ordinary	0.001	16.46 ± 0.09	7.88 ± 0.01	8.58 ± 0.09
C100	R19	matched	0.001	16.83 ± 0.16	7.70 ± 0.01	9.13 ± 0.17
C100	R19	ordinary	0.001	16.94 ± 0.09	7.78 ± 0.01	9.16 ± 0.09
C10	R8	R8	0.001	36.92 ± 0.16	32.78 ± 0.01	4.14 ± 0.15
C10	R8	R8	0.005	40.35 ± 0.09	34.32 ± 0.02	6.03 ± 0.08
C10	R16	matched	0.005	40.06 ± 0.12	32.87 ± 0.01	7.20 ± 0.13
C10	R16	ordinary	0.005	40.01 ± 0.07	32.89 ± 0.00	7.13 ± 0.07
C10	R19	matched	0.001	40.40 ± 0.11	32.26 ± 0.01	8.14 ± 0.09
C10	R19	ordinary	0.001	40.40 ± 0.12	32.38 ± 0.01	8.02 ± 0.12

On CIFAR-100, mean continual accuracy increases from 14.08% for R8 to 16.49% for matched R16 and 16.83% for matched R19. Their fresh-reset gaps are 5.88, 8.66, and 9.13 percentage points. The improvement with budget thus also occurs when the classifier adapts and its learned parameters are retained across tasks. CIFAR-10 illustrates why absolute accuracy and the fresh-reference gap should be considered together. At 0.005, matched R16 has slightly lower continual accuracy than R8, 40.06% versus 40.35%, but a larger gap, 7.20 versus 6.03 points, because its fresh accuracy is lower. At 0.001, matched R19 improves both quantities over R8: continual accuracy increases from 36.92% to 40.40%, and the gap from 4.14 to 8.14 points. Across both datasets, matched and ordinary initialization differ in mean continual accuracy by less than 0.11 points at the same budget and rate.

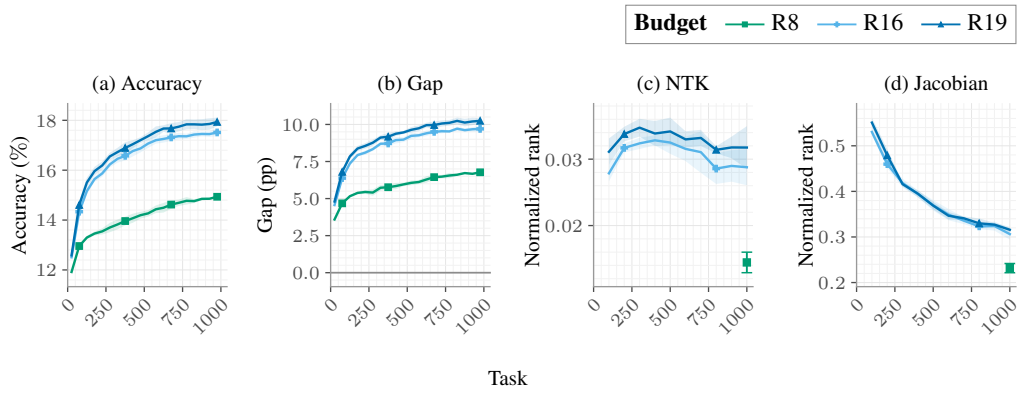

Figure 22: Matched initialization on CIFAR-100 with a trainable classifier and common Adam learning rate 0.001. From left to right: continual test accuracy (%), its gap over fresh-reset (percentage points), normalized NTK effective rank, and normalized trunk-Jacobian effective rank. Performance uses nonoverlapping 50-task bins. Continual R16/R19 diagnostics are measured every 100 tasks; R8 is shown only at task 1,000. Curves and points show four-seed means in one environment; bands and error bars show one sample standard deviation.

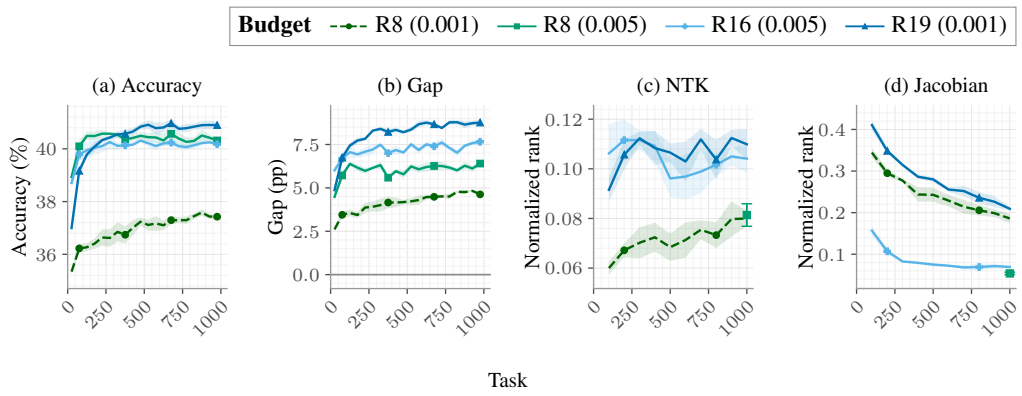

Figure 23: Matched initialization on CIFAR-10 with a trainable classifier. From left to right: continual test accuracy (%), its gap over fresh-reset (percentage points), normalized NTK effective rank, and normalized trunk-Jacobian effective rank. Matched R16 uses learning rate 0.005 and matched R19 uses 0.001, with R8 references at both rates as identified in the legend. Performance uses nonoverlapping 50-task bins. Continual R16/R19 and R8 at 0.001 diagnostics are measured every 100 tasks. R8 at 0.005 remains an endpoint-only measurement at task 1,000. The added diagnostics come from a deterministic replay matching all recorded test/validation accuracies and the original final diagnostic values; they are not independent replications.

Curves and points show four-seed means in one environment; bands and error bars show one sample standard deviation.

D.2.4 Diagnostic definitions and coverage

Diagnostic definitions, probe sizes, normalization, and averaging follow Appendix D.1.2. For trainable classifiers, the NTK additionally differentiates with respect to classifier parameters.

Frozen-classifier diagnostics are measured every 50 tasks for continual, fresh-reset, and fresh-carry learners; the figures show the continual learner. Trainable-classifier diagnostics cover continual R16/R19 and CIFAR-10 R8 at learning rate 0.001 every 100 tasks; the remaining R8 dataset/rate configurations have measurements only at task 1,000. Table 16 compares the common final checkpoint, whereas checkpoint-averaged values reported here and in Section 4.2 summarize the 20 frozen-model checkpoints. At task 1,000 on CIFAR-100 with a trainable classifier, normalized NTK effective rank is 0.014 for R8 and 0.029/0.032 for matched R16/R19. The corresponding trunk-Jacobian ranks are 0.232 and 0.306/0.316. Although both larger models finish above R8, their mean trunk-Jacobian ranks decline over the measured training history. The cross-budget ordering therefore coexists with declining input-sensitivity effective rank within the larger models.

Table 16: Normalized NTK and trunk-Jacobian effective ranks of the continual learners after task 1,000. R16/R19 use R8-matched initialization. Entries are means ± one sample standard deviation across four seeds in one environment per dataset. C100 and C10 denote CIFAR-100 and CIFAR-10. NTKs include classifier parameters only when the classifier is trainable.

Classifier	Data	Model	LR	NTK rank	Trunk-Jacobian rank
Frozen	C10	R8	0.005	0.058 ± 0.005	0.712 ± 0.018
Frozen	C10	R16	0.001	0.041 ± 0.001	0.920 ± 0.007
Frozen	C10	R19	0.001	0.043 ± 0.001	0.906 ± 0.010
Frozen	C100	R8	0.001	0.072 ± 0.002	0.754 ± 0.004
Frozen	C100	R16	0.001	0.078 ± 0.002	0.798 ± 0.010
Frozen	C100	R19	0.001	0.080 ± 0.004	0.804 ± 0.002
Trainable	C10	R8	0.001	0.080 ± 0.003	0.186 ± 0.011
Trainable	C10	R8	0.005	0.081 ± 0.005	0.054 ± 0.003
Trainable	C10	R16	0.005	0.104 ± 0.006	0.069 ± 0.003
Trainable	C10	R19	0.001	0.110 ± 0.006	0.209 ± 0.005
Trainable	C100	R8	0.001	0.014 ± 0.002	0.232 ± 0.010
Trainable	C100	R16	0.001	0.029 ± 0.003	0.306 ± 0.012
Trainable	C100	R19	0.001	0.032 ± 0.003	0.316 ± 0.006

D.3 EXPONENTIAL-COORDINATE CONSTRAINTS

D.3.1 CONSTRAINT AND NUMERICAL IMPLEMENTATION

The exponential models in Section 4.3 parameterize each hidden matrix as

$$W(A) = \exp(A - A^{\mathsf{T}}), \qquad A \in \mathbb{R}^{128\times 128}, \tag{11}$$

following the use of exponential coordinates for orthogonal optimization (Lezcano-Casado & Martínez-Rubio, 2019). The constrained variant, `expm_ball`, projects the stored generator after each optimizer update:

$$A \leftarrow A \min\left\{1, \frac{r}{\|A\|_F}\right\}, \qquad r = \frac{\pi\sqrt{128}}{2}, \tag{12}$$

with the zero matrix unchanged. Optimizer moments are retained without projection or reset.

Both variants store the full matrix A, so the constraint also acts on coordinates that are redundant in $A - A^{\mathsf{T}}$. The chosen radius preserves representability of $\mathrm{SO}(128)$ in exact arithmetic (Appendix A.6). The comparison therefore changes the allowed optimization coordinates while retaining the same nominal family of rotations. The random-projection experiments evaluate the matrix exponential using an order-10 Taylor approximation with FP64 internal arithmetic and FP32 stored parameters and outputs. The unconstrained and constrained implementations use 12 and seven squarings, respectively. This numerical difference accompanies the coordinate constraint. Across the frozen-classifier learning-rate grid, the maximum recorded task-end hidden-weight orthogonality deviation is below 4.7×10^{-7}; this verifies the recorded orthogonality accuracy but does not establish equality of numerical derivatives. The VAE experiments use 12 squarings for both variants.

D.3.2 RANDOM PROJECTIONS: COMPLETE LEARNING-RATE GRIDS AND ADDITIONAL SETTINGS

We use the CIFAR-10 and CIFAR-100 random-projection protocols and four-seed aggregation described in Appendix B, evaluating both frozen and trainable classifiers. Each model is trained for 1,000 tasks at each learning rate in $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$. These comparisons include continual and fresh-reset learners. With a trainable classifier, continual retains the classifier together with its hidden parameters and Adam state, whereas fresh-reset reinitializes all trainable parameters and optimizer state. Frozen classifiers are shared across models, rates, and learner roles within each seed. Learning-rate selection maximizes mean continual validation accuracy over tasks 901–1,000 for frozen classifiers and tasks 101–1,000 for trainable classifiers. On both datasets and under both classifier policies, the selected rates are 0.005 for `expm` and 0.001 for `expm_ball`. Fresh-reset uses the corresponding continual-selected rate. Tables 17 and 18 report all rates over tasks 1–1,000, including the tasks used for validation-based rate selection.

The additional trajectories in Figures 24–26 complement the frozen-classifier CIFAR-100 comparison in Section 4.3. With a trainable classifier on CIFAR-100, selected-rate continual accuracy increases from 17.17% to 17.35%, while fresh-reset accuracy decreases from 9.15% to 7.89%. Most of the larger continual-minus-fresh gap therefore reflects the weaker fresh reference. At the common rate 0.001, continual accuracy instead increases from 14.81% to 17.35%, with fresh-reset accuracy approximately 7.89% for both models. On CIFAR-10, selected-rate continual accuracy increases from 31.01% to 31.43% with a frozen classifier and from 39.03% to 40.74% with a trainable classifier. The frozen-classifier gap nevertheless decreases from 4.06 to 2.84 percentage points because fresh-reset accuracy improves more. The effect depends on learning rate: at 0.01, the constraint lowers continual accuracy in all four dataset/classifier settings.

Table 17: CIFAR-100 exponential-coordinate learning-rate grid under frozen and trainable classifiers. C and F denote continual and fresh-reset test accuracy (%); $G = C - F$ is the paired gap in percentage points. Accuracies are means over tasks 1–1,000 and four seeds; gaps show the mean $\pm$ sample standard deviation across the four seed averages.

		expm			expm_ball		
Classifier	LR	C	F	G	C	F	G
Frozen	0.0001	3.19	2.27	$+0.92 \pm 0.11$	3.21	2.27	$+0.93 \pm 0.11$
Frozen	0.0005	4.21	3.37	$+0.84 \pm 0.07$	4.65	3.37	$+1.28 \pm 0.12$
Frozen	0.001	4.44	3.55	$+0.89 \pm 0.10$	5.07	3.55	$+1.52 \pm 0.12$
Frozen	0.005	4.70	3.11	$+1.60 \pm 0.09$	4.89	3.11	$+1.78 \pm 0.10$
Frozen	0.01	4.32	2.86	$+1.46 \pm 0.16$	3.65	2.93	$+0.72 \pm 0.12$
Trainable	0.0001	10.10	3.61	$+6.49 \pm 0.06$	10.21	3.61	$+6.60 \pm 0.04$
Trainable	0.0005	13.65	7.11	$+6.54 \pm 0.04$	15.11	7.11	$+8.01 \pm 0.10$
Trainable	0.001	14.81	7.89	$+6.92 \pm 0.07$	17.35	7.89	$+9.46 \pm 0.14$
Trainable	0.005	17.17	9.15	$+8.02 \pm 0.37$	16.33	9.15	$+7.17 \pm 0.11$
Trainable	0.01	15.54	8.74	$+6.80 \pm 0.09$	12.01	8.75	$+3.26 \pm 0.29$

Table 18: CIFAR-10 exponential-coordinate learning-rate grid under frozen and trainable classifiers. C and F denote continual and fresh-reset test accuracy (%); $G = C - F$ is the paired gap in percentage points. Accuracies are means over tasks 1–1,000 and four seeds; gaps show the mean $\pm$ sample standard deviation across the four seed averages.

		expm			expm_ball		
Classifier	LR	C	F	G	C	F	G
Frozen	0.0001	27.53	25.86	$+1.67 \pm 0.46$	27.81	25.86	$+1.95 \pm 0.47$
Frozen	0.0005	30.46	28.79	$+1.68 \pm 0.43$	31.09	28.79	$+2.30 \pm 0.38$
Frozen	0.001	30.85	28.58	$+2.26 \pm 0.36$	31.43	28.59	$+2.84 \pm 0.31$
Frozen	0.005	31.01	26.95	$+4.06 \pm 0.10$	30.80	26.95	$+3.85 \pm 0.17$
Frozen	0.01	29.97	26.28	$+3.70 \pm 0.15$	28.20	26.29	$+1.92 \pm 0.12$
Trainable	0.0001	32.20	27.43	$+4.77 \pm 0.06$	32.24	27.43	$+4.81 \pm 0.04$
Trainable	0.0005	37.19	31.96	$+5.23 \pm 0.05$	38.72	31.96	$+6.76 \pm 0.05$
Trainable	0.001	38.46	32.81	$+5.65 \pm 0.03$	40.74	32.82	$+7.92 \pm 0.05$
Trainable	0.005	39.03	32.82	$+6.22 \pm 0.15$	38.98	32.79	$+6.19 \pm 0.04$
Trainable	0.01	36.97	31.56	$+5.42 \pm 0.29$	34.17	31.58	$+2.59 \pm 0.14$

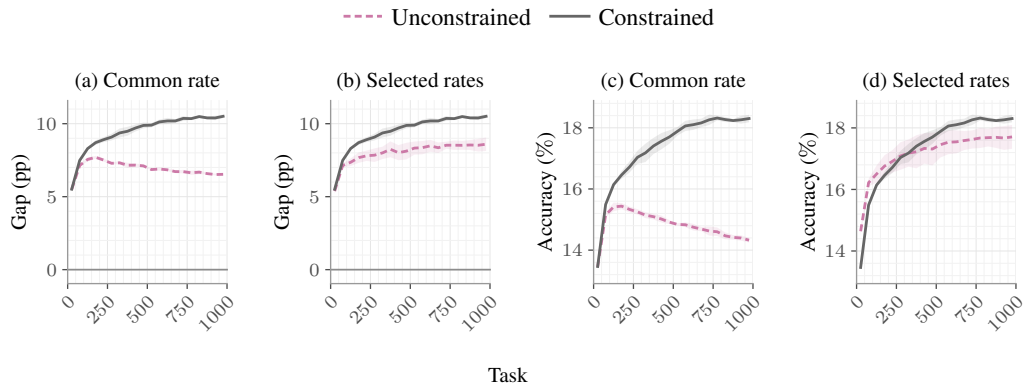

Figure 24: Exponential-coordinate comparisons on CIFAR-100 with a trainable classifier. From left to right: fresh-reset gaps at the common and selected rates, followed by continual test accuracies at the common and selected rates. The common rate is 0.001; selected rates are 0.005 unconstrained and 0.001 constrained, so the constrained history appears in both comparisons. Accuracies are percentages and gaps are percentage points. Curves show four-seed means of 50-task averages; bands show one sample standard deviation.

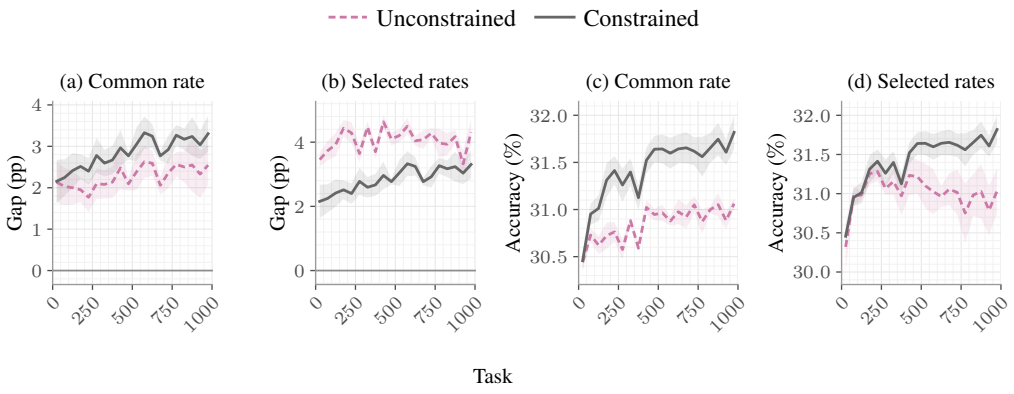

Figure 25: Exponential-coordinate comparisons on CIFAR-10 with a frozen classifier. From left to right: fresh-reset gaps at the common and selected rates, followed by continual test accuracies at the common and selected rates. The common rate is 0.001; selected rates are 0.005 unconstrained and 0.001 constrained, so the constrained history appears in both comparisons. Accuracies are percentages and gaps are percentage points. Curves show four-seed means of 50-task averages; bands show one sample standard deviation.

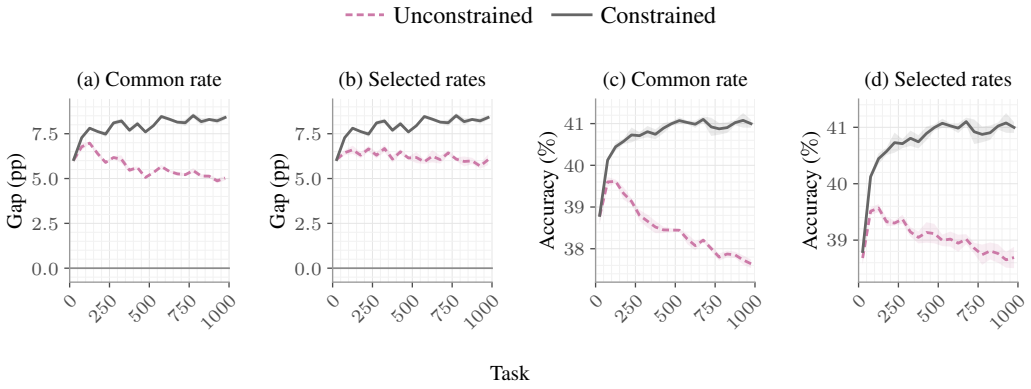

Figure 26: Exponential-coordinate comparisons on CIFAR-10 with a trainable classifier. From left to right: fresh-reset gaps at the common and selected rates, followed by continual test accuracies at the common and selected rates. The common rate is 0.001; selected rates are 0.005 unconstrained and 0.001 constrained, so the constrained history appears in both comparisons. Accuracies are percentages and gaps are percentage points. Curves show four-seed means of 50-task averages; bands show one sample standard deviation.

D.3.3 EMPIRICAL NTK GEOMETRY UNDER EXPONENTIAL-COORDINATE CONSTRAINTS

We use the empirical logit NTK and nonlinear trunk-Jacobian effective ranks defined in Appendix D.1.2. Measurements concern the continual learners at their selected rates: 0.005 for `expm` and 0.001 for `expm_ball`. Frozen-classifier measurements are available after tasks 50, 100, . . ., 1,000 for all four seeds; trainable-classifier measurements are available only at task 1,000. Classifier parameters enter the NTK differentiation only when trainable. Averaging over the measured checkpoints and seeds, frozen-classifier CIFAR-100 gives normalized NTK effective ranks of 0.153 and 0.174 for unconstrained and constrained coordinates, respectively. Corresponding trunk-Jacobian effective ranks are 0.652 and 0.789. On CIFAR-10, these values are 0.074 and 0.126 for the NTK and 0.803 and 0.874 for the trunk Jacobian. Figure 27 shows the frozen-classifier trajectories, and Table 19 reports the final measurements under both classifier policies. At their selected rates, the constrained models therefore distribute sensitivity across more effective directions on average. This association accompanies higher continual accuracy, but the comparison changes learning rate as well as the coordinate constraint. In particular, the unconstrained CIFAR-100 model at the common rate 0.001 is absent from this diagnostic collection, so these measurements do not explain the common-rate performance difference in Section 4.3.

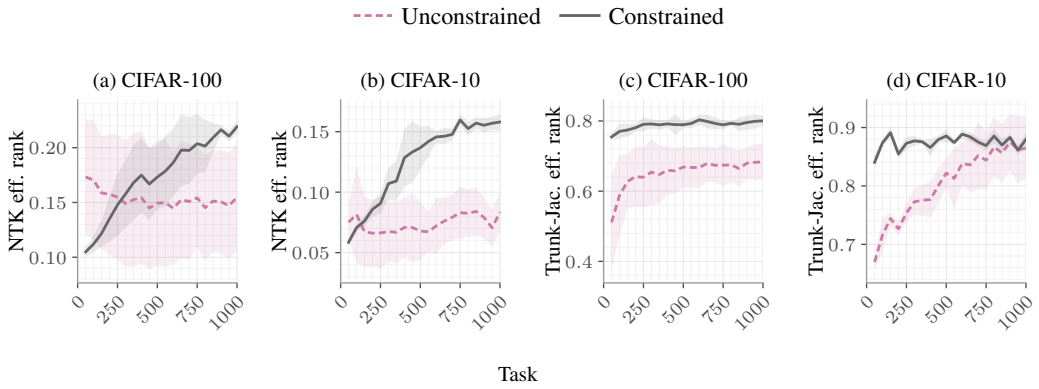

Figure 27: Sensitivity diagnostics for the frozen-classifier exponential models at their selected rates: 0.005 unconstrained and 0.001 constrained. From left to right: normalized NTK effective rank on CIFAR-100 and CIFAR-10, followed by normalized trunk-Jacobian effective rank on the same datasets. Measurements are taken every 50 tasks. Curves show four-seed means; bands show one sample standard deviation.

Table 19: Normalized NTK and trunk-Jacobian effective ranks at task 1,000 for the selected-rate exponential models. Entries show the mean $\pm$ sample standard deviation across four seeds. C100/C10 denote CIFAR-100/10. Classifier parameters enter the NTK only when trainable; trainable-classifier measurements are available only at this endpoint.

Classifier	Data	Model	LR	NTK rank	Trunk-Jacobian rank
Frozen	C100	expm	0.005	0.155 ± 0.049	0.683 ± 0.050
Frozen	C100	expm_ball	0.001	0.219 ± 0.005	0.800 ± 0.015
Frozen	C10	expm	0.005	0.083 ± 0.016	0.865 ± 0.053
Frozen	C10	expm_ball	0.001	0.158 ± 0.007	0.880 ± 0.011
Trainable	C100	expm	0.005	0.030 ± 0.004	0.131 ± 0.015
Trainable	C100	expm_ball	0.001	0.051 ± 0.009	0.296 ± 0.014
Trainable	C10	expm	0.005	0.053 ± 0.005	0.057 ± 0.004
Trainable	C10	expm_ball	0.001	0.184 ± 0.010	0.200 ± 0.011

D.3.4 VAE REPRESENTATIONS: SELECTED-RATE EXPONENTIAL COMPARISONS

We extend the comparison to CIFAR-100 using the frozen VAE representations and classifier described in Appendix B. The three learner roles follow Appendix B.2, and the optimizer settings and 50-task learning-rate selection follow Appendix D.1.6. The selected 1,000-task histories include the first 50 tasks used for tuning. Both variants select 0.001 under Adam and 0.01 under SGD.

The variants share initial generators inside the constraint ball. Both use an order-10 Taylor approximation with 12 squarings and FP64 internal arithmetic; stored parameters and optimizer state remain FP32. The constrained variant applies the projection defined above after every update. The constrained models reach an early accuracy plateau, whereas unconstrained accuracy declines with continued training (Figure 28). Between the first and last 50-task windows, constrained accuracy changes from 6.47% to 6.43% under Adam and from 6.03% to 6.09% under SGD. The corresponding unconstrained accuracies decrease from 6.39% to 5.17% and from 5.76% to 3.80%.

Over tasks 1–1,000, the constraint increases mean continual accuracy from 5.61% to 6.40% under Adam and from 4.55% to 6.07% under SGD. The constrained continual learners nevertheless remain below fresh-reset on average, with gaps of -0.41 and -0.04 percentage points. Unconstrained generator norms grow throughout training, while constrained norms remain near the projection radius (Figure 29). These experiments use the same frozen features as Appendix D.1.6. The concentration of feature variation discussed there is a possible contributor to early saturation, but does not establish an explanation for the differences between the exponential variants.

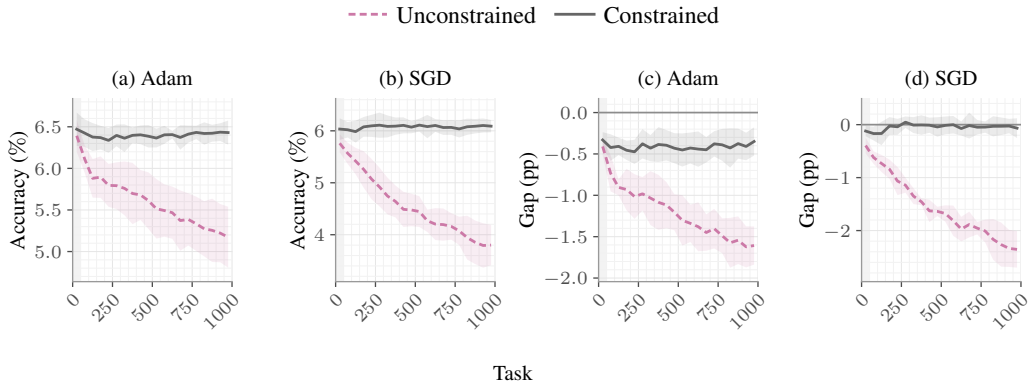

Figure 28: Exponential-coordinate comparisons on frozen CIFAR-100 VAE representations with a frozen classifier. From left to right: continual test accuracy under Adam and SGD, followed by fresh-reset gaps under Adam and SGD. Both models use learning rate 0.001 under Adam and 0.01 under SGD. Curves show four-seed means of 50-task averages; bands show one sample standard deviation. Gray shading marks the first 50 tasks used for tuning and included in the reported histories.

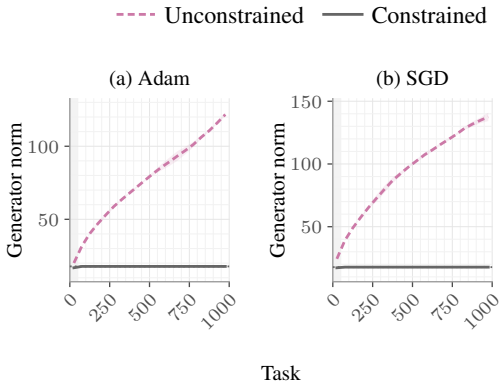

Figure 29: Stored-generator Frobenius norms for the VAE comparisons under Adam and SGD. Each task records the maximum $\|A\|_F$ over both hidden layers and all post-update steps, after projection where applicable. Curves average these task maxima in 50-task bins and across four seeds; bands show one sample standard deviation. The dotted line marks $r = \pi\sqrt{128}/2$, and gray shading marks the tuning prefix.

E PRACTICAL PERFORMANCE AND ADDITIONAL TASK SETTINGS

E.1 ESTABLISHED PLASTICITY METHODS

E.1.1 RANDOM PROJECTIONS: PROTOCOL, METHOD SETTINGS, AND SELECTION

We compare established plasticity methods with Butterfly R1, R8, and R19 on CIFAR-10 and CIFAR-100 pixel-permutation streams. Inputs pass through a frozen random projection into two width-128 ReLU hidden layers and a trainable classifier. The comparisons follow the one-epoch protocol in Appendix B, with Adam learning rate fixed at 10^{-3} for every method. Continual learners retain their parameters, optimizer state, and intervention state across tasks; method-matched fresh references reinitialize these states on every task. The Butterfly histories are the optimizer-carry histories used in the rotation-budget comparisons.

Method-specific settings were selected through a preceding validation search on each dataset. Each candidate received 300 one-epoch tasks in one environment with two model seeds. Selection maximized continual validation accuracy averaged over tasks 101–300 within each seed and then across seeds, with exact ties resolved by candidate identifier. The search jointly varied learning rate and intervention settings over the grids in Table 20, comprising 95 completed candidates per dataset. The subsequent 1,000-task evaluation retained the selected intervention settings but fixed every learning rate at 10^{-3}. It therefore compares methods at a common rate, rather than at their individually selected search rates. The common rate matches the Butterfly histories from Section 3, which were trained at 10^{-3} without a search, and was the selected rate for 14 of the 18 dataset/method pairs. The remaining four, CBP on CIFAR-10 and AdamO, L2, and L2-init on CIFAR-100, selected 3×10^{-3}, the top of $\mathcal{L}_3$. Because the search also contained their selected intervention settings at 10^{-3}, the cost of the common rate is measured: over the selection window, their validation accuracy was lower by 0.68, 0.79, 0.19, and 0.15 percentage points, respectively.

L2 and L2-init use penalty coefficient 10^{-4}. Shrink-and-Perturb applies a shrinkage factor of 0.9999 and Gaussian noise with standard deviation 0.001 after every optimizer update. Contribution-based continual backpropagation (CBP) uses replacement rate 0.001, maturity threshold 100 updates, and utility decay 0.99. ReDo uses threshold zero and a recycling interval of 500 updates. Recycling clears the affected optimizer moments; CBP also resets the affected per-coordinate Adam step counts, whereas ReDo retains Adam's global step count. Intervention state persists in continual learners and resets in fresh references. Because a fresh task contains 390 updates, its ReDo reference never reaches the recycling interval. AdamO uses coefficient 0.01 and an isometry-update rate equal to the Adam learning rate, with both hidden matrices and the trainable classifier included in its matrix regularization. AdamO and the orthogonal-initialization control use orthogonal hidden initialization. LayerNorm acts on hidden pre-activations. A dense control completes the comparison. Each baseline has six seeds distributed equally across three environments. Butterfly has ten seeds distributed 4/3/3 across three different environments. We average tasks within seeds and seeds within environments, then weight the three environment means equally. Reported uncertainty is sample SD across environment means. Continual-minus-fresh gaps are paired within task and seed; baseline–Butterfly comparisons are unpaired because their environments and seeds differ.

Table 20: Random-projection baseline search on each CIFAR dataset. Candidates are Cartesian products of the listed method settings and learning rates, with $\mathcal{L}_3 = \{10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ and $\mathcal{L}_5 = \{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. N counts candidates. Each candidate receives 300 tasks; selection uses continual validation accuracy over tasks 101–300. Final evaluation fixes the learning rate at 10^{-3}.

Method	LR grid	Other searched parameters	N
Dense	$\mathcal{L}_3$	None	3
Orthogonal-init	$\mathcal{L}_5$	None	5
LayerNorm	$\mathcal{L}_3$	None	3
L2	$\mathcal{L}_3$	$\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$	12
L2-init	$\mathcal{L}_3$	$\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$	12
Shrink-and-Perturb	$\mathcal{L}_3$	Shrink $\in \{0.999, 0.9999\}$; noise SD $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$	18
Continual BP	$\mathcal{L}_3$	Replacement rate $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$	9
ReDo	$\mathcal{L}_3$	Threshold $\in \{0, 0.025, 0.1\}$; period $\in \{500, 1000\}$ updates	18
AdamO	$\mathcal{L}_5$	$\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}\}$	15
Total			95

E.1.2 Random projections: absolute performance and fresh-reference outcomes

At the common Adam learning rate, Butterfly R19 has the highest mean continual test accuracy among the evaluated methods on both datasets: 40.42% on CIFAR-10 and 16.94% on CIFAR-100, averaged over tasks 1–1,000. CBP has the highest corresponding baseline means, 39.10% and 15.81%. The ordering on CIFAR-10 is sensitive to this choice: CBP's selected search rate was worth 0.68 validation points over its common-rate setting, a sizeable fraction of its 1.32-point deficit to R19. This ordering does not extend to smaller Butterfly budgets: R1 falls below every baseline on both datasets, while R8 falls below several interventions. Tables 21 and 22 report continual training and test accuracy alongside fresh-reference accuracy and paired history gaps. Figure 30 shows how continual accuracy and the gaps evolve. These comparisons place the rotation-budget response alongside established methods under the stated fixed-rate protocol; they do not establish a general ranking across representations or tuning policies.

Table 21: Fixed-rate CIFAR-10 comparison, averaged over tasks 1–1,000. C and F denote continual and fresh-reset test accuracy; Train denotes continual training accuracy, all in percent. $G = C - F$ is the paired gap in percentage points. Entries are mean ± sample SD across three equally weighted environment means. Baseline and Butterfly cohorts differ.

Method	C	Train	F	G
Dense	34.24 ± 0.12	34.34 ± 0.31	36.16 ± 0.03	-1.92 ± 0.11
Orthogonal-init dense	34.12 ± 0.11	34.20 ± 0.19	37.88 ± 0.10	-3.76 ± 0.02
LayerNorm	34.95 ± 0.10	35.16 ± 0.20	40.21 ± 0.02	-5.26 ± 0.10
L2	37.74 ± 0.03	37.92 ± 0.40	35.77 ± 0.03	1.97 ± 0.06
L2-init	38.90 ± 0.04	39.20 ± 0.33	36.22 ± 0.04	2.68 ± 0.04
Shrink and Perturb	38.27 ± 0.02	38.46 ± 0.38	36.04 ± 0.02	2.23 ± 0.02
Continual backpropagation	39.10 ± 0.10	39.45 ± 0.28	36.32 ± 0.05	2.79 ± 0.06
ReDo	38.53 ± 0.05	38.98 ± 0.44	36.16 ± 0.03	2.37 ± 0.07
AdamO	39.07 ± 0.11	39.35 ± 0.30	37.82 ± 0.10	1.25 ± 0.07
Butterfly R1	30.26 ± 0.15	30.40 ± 0.07	32.38 ± 0.20	-2.12 ± 0.09
Butterfly R8	37.03 ± 0.16	37.27 ± 0.11	32.72 ± 0.22	4.31 ± 0.17
Butterfly R19	40.42 ± 0.23	41.33 ± 0.06	32.32 ± 0.22	8.09 ± 0.06

Table 22: Fixed-rate CIFAR-100 comparison, averaged over tasks 1–1,000. C and F denote continual and fresh-reset test accuracy; Train denotes continual training accuracy, all in percent. $G = C - F$ is the paired gap in percentage points. Entries are mean ± sample SD across three equally weighted environment means. Baseline and Butterfly cohorts differ.

Method	C	Train	F	G
Dense	11.26 ± 0.27	11.16 ± 0.28	9.78 ± 0.27	1.48 ± 0.03
Orthogonal-init dense	11.12 ± 0.33	11.06 ± 0.33	11.21 ± 0.26	-0.09 ± 0.08
LayerNorm	12.32 ± 0.14	12.21 ± 0.08	13.11 ± 0.28	-0.79 ± 0.22
L2	13.89 ± 0.32	13.92 ± 0.36	9.46 ± 0.27	4.44 ± 0.08
L2-init	14.77 ± 0.22	14.93 ± 0.25	9.75 ± 0.26	5.02 ± 0.04
Shrink and Perturb	14.41 ± 0.17	14.57 ± 0.18	9.66 ± 0.26	4.75 ± 0.09
Continual backpropagation	15.81 ± 0.20	16.14 ± 0.26	9.94 ± 0.26	5.87 ± 0.07
ReDo	14.58 ± 0.19	14.72 ± 0.25	9.78 ± 0.27	4.79 ± 0.08
AdamO	14.76 ± 0.17	14.90 ± 0.20	11.14 ± 0.26	3.62 ± 0.09
Butterfly R1	8.74 ± 0.10	8.72 ± 0.03	8.66 ± 0.06	0.08 ± 0.08
Butterfly R8	14.15 ± 0.26	14.28 ± 0.07	8.22 ± 0.02	5.92 ± 0.25
Butterfly R19	16.94 ± 0.22	17.63 ± 0.08	7.81 ± 0.03	9.13 ± 0.24

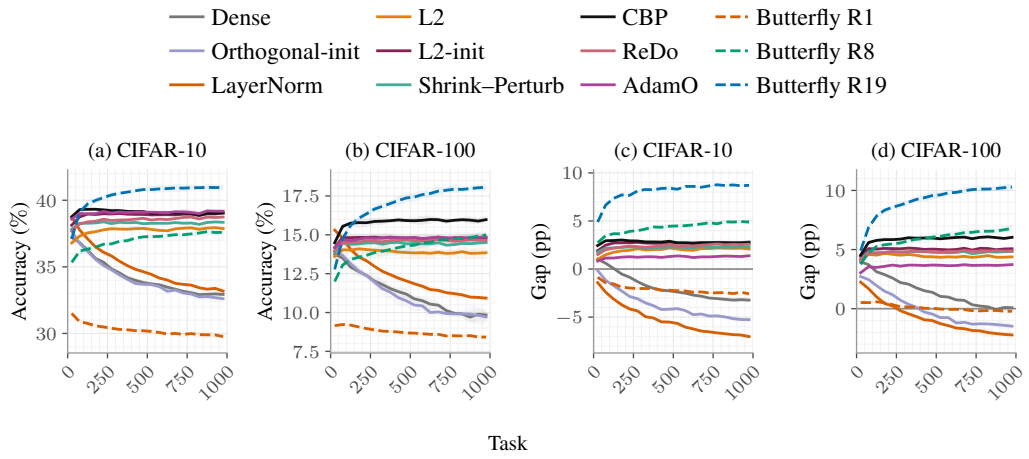

Figure 30: Fixed-rate baseline comparisons, from left to right: CIFAR-10 and CIFAR-100 continual test accuracy, followed by their corresponding paired continual-minus-fresh-reset gaps. Accuracy is in percent and gaps in percentage points. Curves summarize nonoverlapping 50-task bins; bands show one sample SD across three equally weighted environment means. Butterfly curves are dashed. Baseline and Butterfly cohorts are unpaired.

E.1.3 VAE REPRESENTATIONS: BASELINE COMPARISONS

Several established plasticity methods outperform Butterfly on frozen CIFAR-100 VAE representations, even when Butterfly benefits from training history. Over tasks 1–1,000, R19 reaches 6.42% continual test accuracy under Adam, compared with 17.25% for L2, and 5.65% under SGD, compared with 12.88% for spectral regularization. Its corresponding fresh-reset gaps remain positive, at 1.97 and 1.88 percentage points. Thus, a positive history gap can coexist with substantially lower absolute accuracy. Table 23 retains all twelve baselines and seven Butterfly budgets; Figure 31 shows representative histories.

The comparison follows Appendix B's latent-permutation protocol: each task permutes the coordinates of frozen 128-dimensional posterior-mean features while preserving the CIFAR-100 labels. All methods use two width-128 hidden layers and a frozen random classifier, with one epoch per task. Four model/head seeds share the encoder, task stream, classifier bank, and mini-batch protocol. Representation, permutation location, classifier policy, and tuning procedure differ from the random-projection comparison, so the change in rankings cannot be attributed to representation alone. Each method selects its learning rate separately under Adam and SGD from $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$. Selection requires all five 50-task candidates to complete and maximizes continual validation accuracy averaged over tasks and then seeds; exact ties favor the smaller rate. Intervention strengths remain fixed. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$; SGD uses momentum 0.99 without Nesterov or dampening.

The selected-rate histories restart from the original model/head seeds with fresh optimizer state. Their first 50 tasks repeat the tuning stream and reproduce its recorded training, validation, and test metrics; optimizer state is not continued from the tuning runs. Tables summarize all 1,000 tasks, including this repeated selection window, and figures shade the first 50 tasks. The remaining tasks extend the same environment and encoder. Boundary selections are marked in the table. L2, L2-init, Shrink-and-Perturb, CBP, ReDo, and AdamO retain the strengths specified in Appendix E.1.1, with interventions restricted to the trainable trunk. CBP and ReDo recycle only the first hidden layer to preserve the frozen classifier. Recycling clears affected optimizer moments or momentum; CBP additionally resets affected per-coordinate Adam step counts. ReDo's 500-update interval is not reached within a fresh task, which contains 352 updates.

Spectral regularization uses coefficient 10^{-4}, exponent $k = 2$, and one power iteration per update with persistent power-iteration state. FIRE applies ten cubic Newton–Schulz iterations at task boundaries after the initial task, retaining each learner's optimizer policy. Parseval uses coefficient $0.001(64/128)^2$, target Gram scale 2, one group, orthogonal initialization with gain $\sqrt{2}$, and a learned diagonal affine transformation. LayerNorm acts on hidden pre-activations. The AdamO-style update combined with SGD is denoted SGD-isometry. Continual learners retain their parameters, optimizer state, and intervention state. Fresh-reset and fresh-carry use identical task-specific fresh parameters and reset intervention state; fresh-carry alone retains its own optimizer state. All roles use the continual-selected rate. Reported means first average tasks within seeds, with sample SD across the four seed means. Gaps are paired within task and seed. The table reports continual accuracy and its fresh-reset gap.

Table 23: Selected-rate CIFAR-100 VAE comparisons under Adam and SGD, averaged over tasks 1–1,000, including the repeated 50-task tuning window. C denotes continual test accuracy (%) and $G = C - F_r$ the paired fresh-reset gap (percentage points). Accuracy and gap entries are mean ± sample SD across four seeds. An asterisk marks a selected rate at the search-grid boundary.

	Adam			SGD		
Method	LR	C	G	LR	C	G
Dense	0.005	12.57 ± 0.51	-0.28 ± 0.40	0.01^*	3.84 ± 0.09	-2.17 ± 0.28
Orthogonal-init	0.005	12.78 ± 0.49	-0.57 ± 0.42	0.01^*	3.89 ± 0.12	-3.80 ± 0.13
L2	0.005	17.25 ± 0.20	4.39 ± 0.11	0.01^*	10.61 ± 0.49	4.68 ± 0.73
L2-init	0.005	17.25 ± 0.17	3.98 ± 0.17	0.01^*	10.88 ± 0.38	4.89 ± 0.60
Shrink-and-Perturb	0.005	16.83 ± 0.22	4.05 ± 0.12	0.01^*	9.80 ± 0.30	3.83 ± 0.54
Continual BP	0.01^*	15.73 ± 0.32	2.29 ± 0.26	0.01^*	4.11 ± 0.23	-1.82 ± 0.24
ReDo	0.005	14.73 ± 0.46	1.89 ± 0.28	0.01^*	7.33 ± 0.34	1.32 ± 0.57
LayerNorm	0.005	10.07 ± 0.24	-3.00 ± 0.11	0.005	3.07 ± 0.09	-6.35 ± 0.28
Spectral	0.005	16.41 ± 0.30	3.08 ± 0.12	0.01^*	12.88 ± 0.46	6.82 ± 0.65
AdamO / SGD-isometry	0.005	15.97 ± 0.34	3.38 ± 0.09	0.01^*	11.75 ± 0.59	4.21 ± 0.59
FIRE	0.005	13.42 ± 0.23	0.10 ± 0.13	0.01^*	10.20 ± 0.20	2.67 ± 0.31
Parseval	0.005	14.28 ± 0.20	1.81 ± 0.17	0.01^*	10.96 ± 0.41	2.53 ± 0.30
Butterfly R1	0.01^*	3.39 ± 0.30	0.01 ± 0.24	0.01^*	1.46 ± 0.02	-0.13 ± 0.07
Butterfly R2	0.01^*	5.06 ± 0.33	0.80 ± 0.25	0.01^*	2.38 ± 0.15	0.49 ± 0.10
Butterfly R4	0.01^*	5.56 ± 0.17	0.93 ± 0.21	0.01^*	4.05 ± 0.35	1.73 ± 0.25
Butterfly R8	0.005	6.00 ± 0.14	1.18 ± 0.23	0.01^*	4.98 ± 0.25	2.07 ± 0.18
Butterfly R16	0.005	6.45 ± 0.20	1.83 ± 0.17	0.01^*	5.51 ± 0.22	1.91 ± 0.19
Butterfly R19	0.005	6.42 ± 0.17	1.97 ± 0.18	0.01^*	5.65 ± 0.21	1.88 ± 0.21
Butterfly R32	0.001	6.37 ± 0.14	1.29 ± 0.29	0.005	5.73 ± 0.20	1.94 ± 0.26

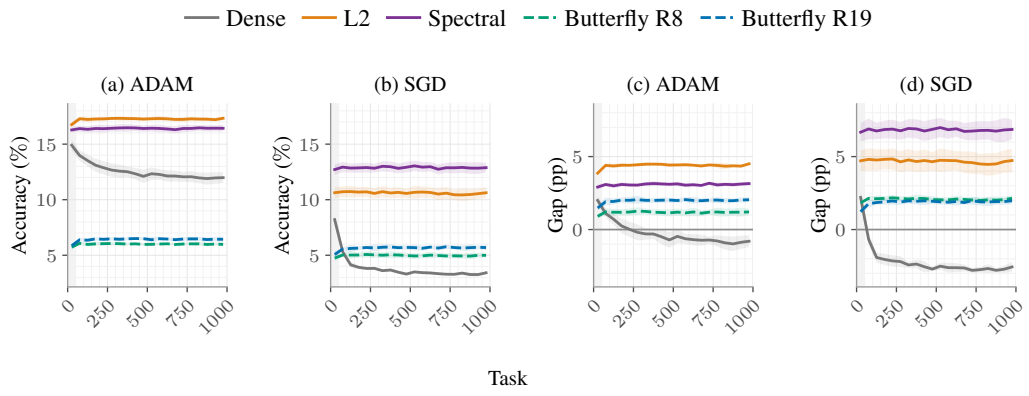

Figure 31: Selected CIFAR-100 VAE histories, from left to right: Adam and SGD continual test accuracy, followed by their corresponding paired fresh-reset gaps. Accuracy is in percent and gaps in percentage points. L2, spectral regularization, dense, and Butterfly R8/R19 illustrate the comparison; the table reports the complete method set. Curves summarize nonoverlapping 50-task bins, with one sample SD across four seeds. Gray shading marks the repeated 50-task tuning window.

E.2 RANDOM-LABEL RELEARNING

Butterfly models sustain high training accuracy across repeated random-label tasks, with larger rotation budgets improving mean memorization performance. Over all 50 tasks, mean terminal training accuracy rises from 92.10% for R8 to 94.86% for R16 and 94.95% for R19. Most of this increase occurs between R8 and R16, with little further change at R19. By comparison, dense Adam's mean terminal accuracy falls from 39.51% over tasks 1–10 to 11.58% over tasks 41–50. Figure 32 shows these histories, and Table 24 reports terminal and peak accuracy, training loss, and fresh-reference outcomes.

Each task assigns new independent random ten-class labels to the same 1,200 CIFAR-10 training images, comprising 120 images from each original class. These labels remain fixed throughout the task. Inputs are neither permuted nor augmented: pixels are scaled by $1/255$ and passed through a shared frozen random projection to width 128. All models have two width-128 ReLU hidden layers and a trainable classifier. Each task receives 100 epochs at batch size 16, totaling 7,500 optimizer updates. Adam uses learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and no weight decay. These runs use fixed settings without a hyperparameter search. Continual backpropagation (CBP) uses the contribution-based intervention and settings specified in Appendix E.1.1, including resets of affected Adam moments and per-coordinate step counts when units are recycled. Continual learners carry their terminal parameters, optimizer state, and intervention state into the next task. Method-matched fresh references reinitialize these states on every task and receive the same labels, minibatches, and training budget.

We evaluate memorization on all 1,200 training images before training and after each epoch. Terminal accuracy and loss use epoch 100; peak accuracy is the maximum over these recorded evaluations. Peak accuracy is a descriptive outcome and does not select the state carried forward. Four model seeds share the image subset, label stream, and input projection. Table entries first average tasks within each seed and then report the mean and sample SD across seeds; accuracy gaps are paired within task and seed. The distinction between peak and terminal accuracy further characterizes the Butterfly budget response. Mean peak accuracy increases from 92.92% for R8 to 97.87% for R16 and 98.02% for R19. The larger budgets therefore reach higher memorization levels within tasks, although some of that accuracy is lost before the terminal readout. All three budgets have positive terminal history gaps, alongside fresh-reference accuracies of 46.56–50.19%. These gaps describe the benefit of accumulated training history when relearning labels on the same images and should be read alongside absolute accuracy. CBP provides an additional benchmark, reaching 99.70% mean terminal accuracy under this protocol.

Table 24: Random-label memorization over tasks 1–50. C and F denote continual and method-matched fresh learners. Terminal accuracy and cross-entropy use epoch 100; peak accuracy is the maximum over the initial and epoch-end evaluations. Accuracies are percentages, and $G = C - F$ is the paired terminal gap in percentage points. Entries are mean $\pm$ sample SD across four seeds after averaging tasks within each seed.

Method	Terminal C	Terminal F	G	Peak C	Peak F	Loss C	Loss F
Dense	18.29 ± 2.23	98.58 ± 0.65	-80.28 ± 2.76	18.51 ± 2.36	100.00 ± 0.01	2.14 ± 0.04	0.06 ± 0.02
Continual BP	99.70 ± 0.05	99.79 ± 0.18	-0.10 ± 0.18	99.90 ± 0.02	100.00 ± 0.00	0.03 ± 0.00	0.02 ± 0.00
Butterfly R8	92.10 ± 0.36	46.56 ± 0.12	45.54 ± 0.43	92.92 ± 0.43	48.56 ± 0.12	0.42 ± 0.01	1.75 ± 0.00
Butterfly R16	94.86 ± 1.34	49.59 ± 0.37	45.27 ± 1.55	97.87 ± 0.06	52.13 ± 0.18	0.24 ± 0.05	1.71 ± 0.00
Butterfly R19	94.95 ± 1.23	50.19 ± 0.63	44.76 ± 1.23	98.02 ± 0.06	52.47 ± 0.34	0.23 ± 0.04	1.70 ± 0.00

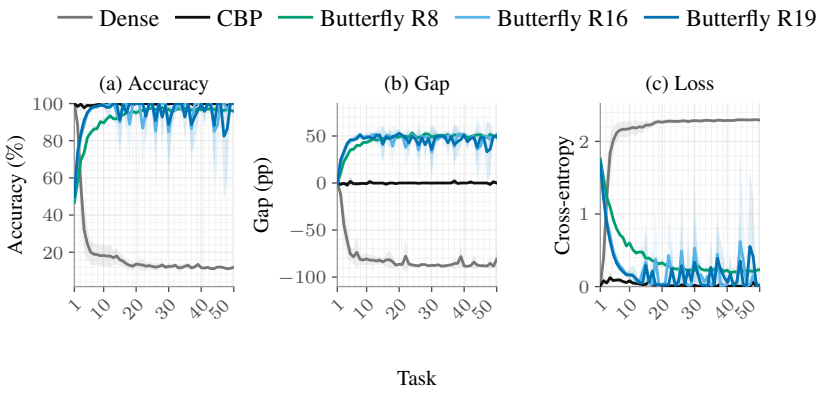

Figure 32: Random-label relearning, from left to right: terminal continual training accuracy (%), paired continual-minus-fresh terminal accuracy gap (percentage points), and terminal continual training cross-entropy. Lines show means across four seeds at each task; bands show one sample SD clipped to each metric's admissible range.

E.3 Cumulative class-incremental learning

In cumulative CIFAR-100 learning, R16 and R19 attain higher mean final accuracy than R8 under the transferred learning-rate policy. At the final 100-class stage, validation-selected continual accuracy is 9.24% for R8, 10.28% for R16, and 10.53% for R19. R19 exceeds its fresh-carry reference by 1.14 ± 0.25 percentage points, with accuracies of $10.53 \pm 0.35\%$ and $9.39 \pm 0.20\%$, respectively. R16 also has a positive mean gap, whereas R8's mean gap is slightly negative. Figure 33 shows the stagewise results, and Table 25 reports validation-selected and terminal outcomes at the final stage.

The protocol introduces five classes at each of 20 stages and trains on all images from the classes introduced so far. Earlier training data remain available throughout. Each class contributes 450 training and 50 validation images from the official training partition. Evaluation uses 80 official test images per class, excluding the 20 images per class previously used for validation in the permutation experiments. Training and evaluation use only the logits of introduced classes. Four seeds define paired class orders across models and learner roles. Butterfly R8, R16, and R19 use two width-128 ReLU hidden layers with trainable biases, a frozen random input projection, and a frozen 100-class linear classifier. Each stage receives 200 epochs at batch size 90. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and no weight decay. Learning rates transfer from the frozen-classifier CIFAR-100 random-projection permutation sweep, selected by continual validation accuracy over tasks 901–1,000: 0.005 for R8 and 0.001 for R16 and R19. These rates remain constant throughout cumulative training. Comparisons involving R8 therefore vary learning rate as well as rotation budget.

For this cumulative setting, training uses horizontal flips, random crops after four-pixel reflection padding, and nearest-neighbor rotations with angles sampled uniformly between $0°$ and $15°$, following the augmentation protocol of Dohare et al. (2024). Pixels are scaled by $1/255$ and channel-normalized with means $(0.5071, 0.4865, 0.4409)$ and standard deviations $(0.2673, 0.2564, 0.2762)$; rotations use zero fill in normalized coordinates. Evaluation is deterministic. Continual and fresh-carry learners receive identical minibatches and augmentation draws within each seed. At each stage, each model, seed, and learner role independently selects the first epoch attaining the highest validation accuracy among epochs 1–200. The continual learner carries these selected parameters into the next stage, while fresh-carry reinitializes its trainable trunk. Both retain their own end-of-stage Adam moments and update counts: restoring selected parameters does not restore optimizer state to the selected epoch. Primary test scores use the validation-selected parameters; terminal scores use epoch 200. Means and sample SD are computed across four seeds, with continual-minus-fresh-carry gaps paired within seed.

The absolute accuracies remain modest. Declining accuracy as classes are added does not by itself establish a loss of plasticity, because the classification problem also becomes harder. This experiment evaluates adaptation as the problem expands with earlier training data retained. At the first

stage, neither learner has prior class history, so any gap reflects different initializations and the resulting training trajectories.

Table 25: Final 100-class test accuracy for continual (C) and fresh-carry (F) learners, with paired gap $G = C - F$. Accuracies are percentages and gaps are percentage points. Entries are mean $\pm$ sample SD across four seeds. Validation-selected scores use each learner's first best-validation epoch within the final stage; terminal scores use epoch 200.

		Validation-selected			Terminal		
Model	LR	C	F	G	C	F	G
R8	0.005	9.24 ± 0.20	9.36 ± 0.35	-0.11 ± 0.21	8.67 ± 0.39	8.36 ± 0.57	0.31 ± 0.30
R16	0.001	10.28 ± 0.46	9.77 ± 0.37	0.50 ± 0.58	9.89 ± 0.65	9.11 ± 0.57	0.78 ± 0.32
R19	0.001	10.53 ± 0.35	9.39 ± 0.20	1.14 ± 0.25	10.15 ± 0.75	9.09 ± 0.51	1.06 ± 0.47

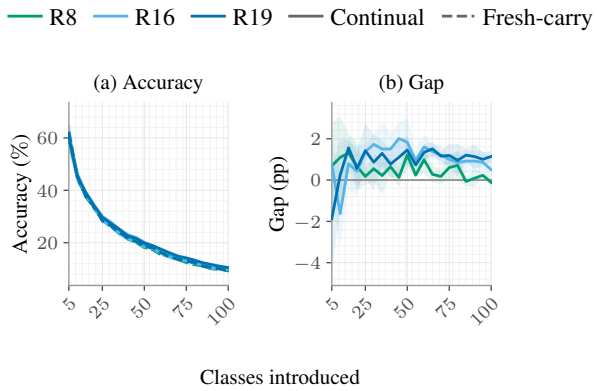

Figure 33: Cumulative class-incremental CIFAR-100: validation-selected accuracy (%) on the left and paired continual-minus-fresh-carry gaps (percentage points) on the right. Solid and dashed accuracy curves denote continual and fresh-carry learners, respectively. Lines show means across four seeds at each stage; bands show one sample SD clipped to each metric's admissible range.