Factor Imbalance and Plasticity Loss in Low-Rank Factorized Networks

Seungwon Oh [1] **Kyung-Joong Kim** [1]

Abstract

Loss of plasticity is a critical challenge in continual and non-stationary learning, where neural networks gradually lose their ability to adapt to new data. We study this phenomenon in low-rank factorized neural networks. Across task-streaming, sample-incremental, and class-incremental learning, low-rank factorized MLPs adapt worse to later tasks or distributions than full-matrix counterparts. This degradation is not explained by the rank constraint alone: frozen-factor controls preserve the low-rank constraint while substantially changing the degradation pattern, implicating the coupled optimization of the two factors. We identify factor imbalance, measured by the Gram mismatch between the factors, as a factor-level diagnostic associated with plasticity loss. Finally, we show that directly regularizing this mismatch reduces imbalance and partially recovers plasticity.[1]

1. Introduction

Deep neural networks are increasingly expected to learn from non-stationary data streams, where the data distribution, task identity, or label space may change over time. A central challenge in such settings is the loss of plasticity: as training proceeds, a model can become progressively less capable of acquiring new knowledge (Zilly et al., 2021; Abbas et al., 2023; Lyle et al., 2023; Dohare et al., 2024). This phenomenon is distinct from catastrophic forgetting, as it concerns future learnability rather than only the retention of past knowledge. Prior work has linked plasticity loss to dormant neurons, loss-landscape curvature, spectral collapse, degraded trainability, and interactions among multiple mechanisms (Sokar et al., 2023; Lyle et al., 2023; Lewandowski et al., 2025; Lyle et al., 2025). Warm-starting studies further show that plasticity can degrade even without distribution shift, depending on the training protocol and the model state induced by prior training (Ash & Adams, 2020; Berariu et al., 2021; Lee et al., 2024; Shin et al., 2024; Ahn et al., 2025). Yet comparatively less is known about how parameterization itself affects plasticity.

Low-rank factorization is a standard parameterization for efficient learning. A factorized layer represents $W \in \mathbb{R}^{m\times n}$ as $W = UV^\top$, where $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$, and r is the bottleneck rank. Such structures appear in representation learning, parameter-efficient fine-tuning, model compression, and Transformer attention (Jing et al., 2020; Hu et al., 2022; Idelbayev & Carreira-Perpinán, 2020; Guo et al., 2024; Kobayashi et al., 2024). Prior analyses show that factorized optimization can learn spectral modes at different rates and induce low-rank, nuclear-norm-like, or rank-minimizing biases (Saxe et al., 2014; Gidel et al., 2019; Gunasekar et al., 2017; Arora et al., 2019; Huh et al., 2023; Timor et al., 2023). However, these works largely focus on the implicit bias and optimization dynamics of factorization, leaving open how low-rank factorized networks behave under continual learning.

We study plasticity loss in low-rank factorized MLPs under non-stationary learning. Across task-, sample-, and class-incremental protocols, factorized models adapt worse to later stages than their full-matrix counterparts. Frozen-factor controls show that this gap is not explained by the rank constraint alone, but is tied to the coupled optimization of the two factors. We then identify Gram mismatch between factors as a diagnostic of this coupled dynamics and show that directly controlling it partially restores plasticity.

2. Low-Rank Factorization Exacerbates Plasticity Loss

Low-rank factorization can affect plasticity through both reduced capacity and altered optimization dynamics. Replacing a dense weight matrix with two factors can reduce the number of trainable parameters and constrains the effective matrix rank. If plasticity loss were mainly capacity-driven, increasing the bottleneck rank should make factorized models approach their full-matrix counterparts. In this section, we test this expectation by comparing full-matrix and low-rank factorized MLPs across three continual learning set-

[1]Department of AI Convergence, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. Correspondence to: Kyung-Joong Kim <kjkim@gist.ac.kr>.

Presented at the ICML 2026 Workshop "Continual Adaptation at Scale: Towards Sustainable AI". Copyright 2026 by the author(s).

[1]Our source code is available at https://github.com/CILAB-RL-TEAM/plasticity-low-rank-factorization.git.

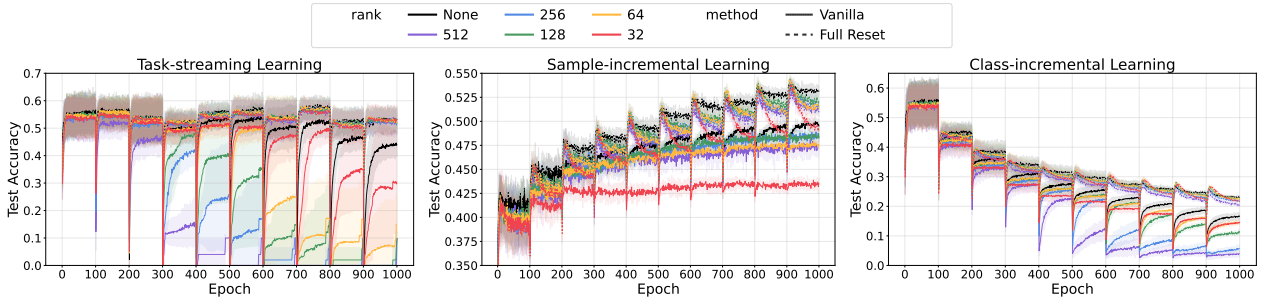

Figure 1. **Low-rank factorization exacerbates plasticity loss across continual learning settings.** We compare full-matrix MLPs with low-rank factorized MLPs under fixed depth and width, varying only the bottleneck rank r. **Left.** Task-streaming learning on CIFAR-100. **Middle.** Sample-incremental learning on CIFAR-10. **Right.** Class-incremental learning on CIFAR-100. Across all three settings, low-rank factorized models show stronger degradation than their full-matrix counterparts.

tings.

2.1. Experimental Setup

We compare full-matrix and low-rank factorized MLPs across three continual learning protocols. All models use a two-hidden-layer ReLU MLP with width 512 and a dense classification head; only hidden layers are factorized as $W_l = U_l V_l^\top$ with bottleneck rank r, while the full-matrix model is denoted by $r = \text{None}$. We evaluate task-streaming learning, where CIFAR-100 is split into 10 disjoint 10-class chunks and each stage trains/evaluates on the current chunk; sample-incremental learning, where CIFAR-10 is split into 10 sample chunks with a fixed label space and evaluation uses the full test set; and class-incremental learning, where CIFAR-100 chunks are accumulated and stage t trains/evaluates on all classes observed so far.

Each experiment has 10 stages with the same per-stage training budget, and results are averaged over five seeds. Following Lee et al. (2024), we reset the optimizer state at each stage. *Vanilla* continual training carries model parameters forward, whereas the *Full Reset* reference reinitializes parameters at each stage and trains on the data available at that stage. Additional optimization details are provided in Appendix A.

2.2. Results

Figure 1 compares low-rank factorized networks with their full-matrix counterparts across the three non-stationary learning protocols. Low-rank factorized networks suffer larger later-stage performance degradation across all three settings. Additional plasticity-related diagnostics are reported in Appendix L: feature rank tends to decrease over training, while the dormant neuron ratio tends to increase. Together, these results indicate that low-rank factorized networks experience stronger loss of plasticity across the considered continual learning settings.

Moreover, Appendix D.2 shows a broadly rank-dependent pattern in these diagnostics: larger bottleneck ranks often exhibit lower feature rank and higher dormant neuron ratio at later stages. This tendency is difficult to reconcile with a purely capacity-based explanation, since increasing r should alleviate the bottleneck if representational capacity were the main cause. Instead, the observed pattern suggests that the vulnerability of low-rank factorized networks is tied not only to the rank constraint itself, but also to the optimization dynamics introduced by jointly training two factors. This motivates the controlled experiment in the next section, where we keep the rank constraint while removing simultaneous co-adaptation between the two trainable factors.

3. Rank Constraint Alone Does Not Explain the Gap

A natural explanation for the performance gap is that factorized layers are rank-constrained. To test whether this constraint alone explains the degradation, we conduct a frozen-factor control experiment at $r = 256$ (Figure 2). We freeze one factor, either U_l or V_l, at initialization and train only the other. This keeps the composed matrix $W_l = U_l V_l^\top$ rank-constrained, while removing the simultaneous co-adaptation of two trainable factors.

If the degradation were primarily caused by the rank constraint, then frozen-factor models and fully trainable factorized models with the same rank should exhibit similar degradation patterns. Figure 2 shows that this is not the case. In the more severe task-streaming and class-incremental settings, freezing either factor substantially reduces later-stage degradation and yields performance close to the *Full Reset* baseline. Notably, under *Full Reset*, fully trainable and frozen-factor variants perform similarly, indicating that coupled factor optimization is not inherently unable to fit the stage data. The sample-incremental case is more nuanced: at $r = 256$, freezing one factor can slightly underperform *Vanilla* training, suggesting that factor co-adaptation can

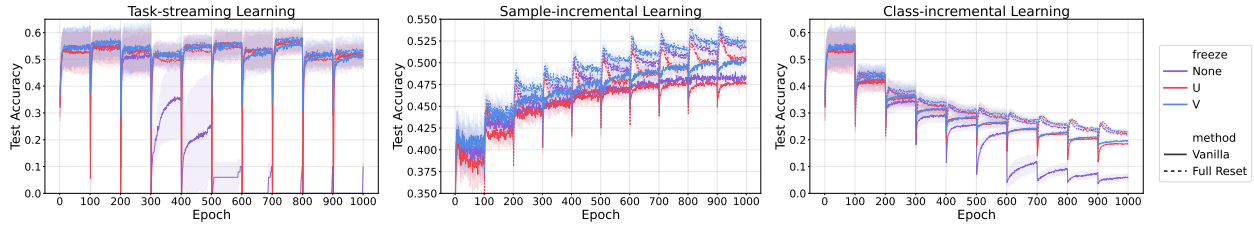

Figure 2. **Rank constraint alone does not explain the plasticity gap.** We conduct a freeze-one-factor control experiment with fixed rank $r = 256$, where either U or V is frozen at initialization and only the other factor is trained. **Left:** task-streaming CIFAR-100. **Middle:** sample-incremental CIFAR-10. **Right:** class-incremental CIFAR-100. Freezing one factor changes the degradation pattern compared to fully trainable low-rank factorization, suggesting that the coupled optimization dynamics between the two factors, rather than the rank constraint alone, contribute to plasticity loss.

remain useful when the distribution shift is mild. Overall, severe plasticity loss can be avoided even when the composed matrices remain rank-constrained, suggesting that the rank constraint alone is insufficient to explain the degradation.

These findings motivate a more mechanistic analysis of factorized optimization. We next examine factor imbalance as a factor-level diagnostic of the coupled dynamics. Appendix H.1 shows that the frozen-factor control remains qualitatively different from fully trainable factorization at $r = 512$.

4. Imbalanced Factor Dynamics in Non-stationary Learning

The preceding section shows that the degradation of low-rank factorized networks cannot be explained by reduced representational rank alone. Since freezing one factor preserves the rank constraint but substantially changes the degradation pattern, we now examine the coupled optimization dynamics of the two trainable factors.

A factorized layer represents a composed matrix $W = UV^\top$ through two matrices that live in different ambient spaces but share the same rank-r latent coordinates. This motivates comparing the two factors through their geometry in this shared latent space, which leads to the balancedness measure introduced below.

4.1. Coupled Factor Geometry

Consider a factorized weight matrix $W = UV^\top$, with $W \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times r}$, and $V \in \mathbb{R}^{n\times r}$. Writing u_k and v_k for the k-th columns of U and V, respectively, we have

$$W = \sum_{k=1}^{r} u_k v_k^\top .$$

Thus, the two factors are coupled through r paired rank-one components. This coupling also appears in the gradients: for $G = \nabla_W \ell(W)$,

$$\nabla_U \ell(UV^\top) = GV, \qquad \nabla_V \ell(UV^\top) = G^\top U.$$

Hence, the update of each factor is scaled and directed by the other factor. To analyze this coupled optimization, we track how the two sides of the factorization use the paired components.

We call a factorization balanced when the paired rank-r components have matching Gram geometry, and quantify imbalance by

$$\Delta(U, V) = \|U^\top U - V^\top V\|_F .$$

The resulting imbalance $\Delta(U, V)$ captures a relational property of the factorization, rather than of the composed matrix W alone. Since the factorization $W = UV^\top$ is not unique, the same composed matrix can have different internal factor geometries, which can lead to different updates when gradient descent is applied to U and V.

4.2. Imbalance Drift and Factor-Dependent Updates

We next show that the balanced geometry defined above is not necessarily preserved by practical finite-step training. Let $W_t = U_t V_t^\top$, and let $G_t = \nabla_W \ell_{\mathcal{B}_t}(W_t)$ denote the mini-batch gradient at step t. A gradient descent step on the two factors is

$$U_{t+1} = U_t - \eta G_t V_t, \qquad V_{t+1} = V_t - \eta G_t^\top U_t.$$

Define the Gram mismatch matrix

$$D_t = U_t^\top U_t - V_t^\top V_t.$$

A direct expansion gives

$$\begin{aligned} D_{t+1} &= U_{t+1}^\top U_{t+1} - V_{t+1}^\top V_{t+1} \\ &= D_t + \eta^2 \left(V_t^\top G_t^\top G_t V_t - U_t^\top G_t G_t^\top U_t \right). \end{aligned}$$

Since the additional term is generally nonzero, finite-step training can change the Gram mismatch through the interaction between the mini-batch gradient and the current

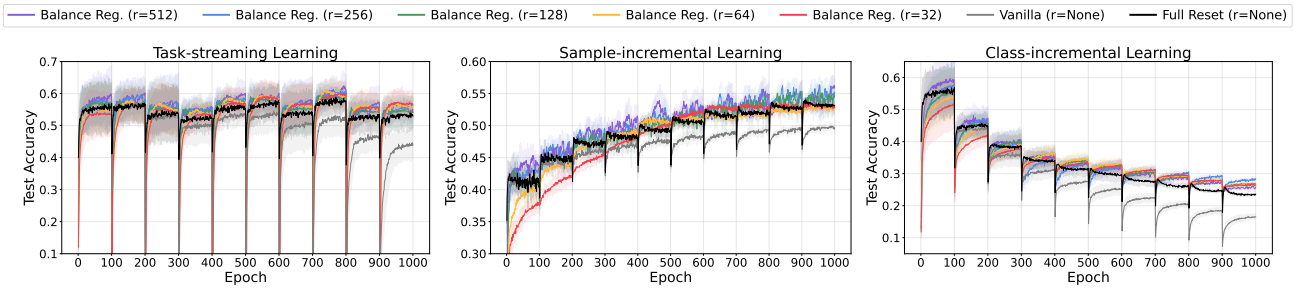

Figure 3. **Balance regularization mitigates plasticity loss in low-rank factorized networks.** We train low-rank factorized MLPs with an explicit balance regularizer, which penalizes mismatch between the two factor Gram matrices. **Left.** Task-streaming learning on CIFAR-100. **Middle.** Sample-incremental learning on CIFAR-10. **Right.** Class-incremental learning on CIFAR-100. Across all three settings, enforcing balance improves later-stage adaptation; corresponding imbalance trajectories are shown in Appendix J.

factor geometry. These changes can accumulate over training, and under non-stationary training the resulting factor state may hinder subsequent adaptation. We track $\Delta(U_t, V_t) = \|D_t\|_F$ as a diagnostic of imbalance drift.

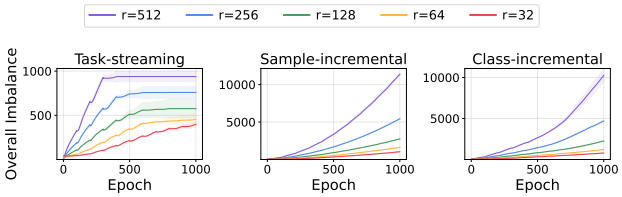

Figure 4. **Imbalance drift across continual learning settings.** Imbalance grows over the course of training in all three continual learning settings, and this drift becomes steeper as the bottleneck rank r increases.

As shown in Figure 4, imbalance indeed grows over training and empirically shows more pronounced drift at larger bottleneck ranks. The relevance of this drift is that factor imbalance can change the update induced by the same composed matrix. This can already be seen in a rank-one case:

$$W = uv^\top = (\alpha u)(\alpha^{-1} v)^\top,$$

which represents the same matrix for any $\alpha > 0$. Although the represented matrix is unchanged, the factorized update induced by a full-matrix gradient G changes with the factor scaling:

$$\mathcal{A}_\alpha(G) = \alpha^{-2} G v v^\top + \alpha^2 u u^\top G.$$

Thus, the balance between the two factors controls the relative strength of the two update channels: large α amplifies updates through $uu^\top G$ while suppressing updates through $Gvv^\top$, whereas small α has the opposite effect. This illustrates that imbalance is a factorization-level state variable that can drift under finite-step training and alter the effective update induced by a given full-matrix gradient.

Appendix C provides the corresponding first-order update analysis. It shows that factorized progress depends on gradient-factor alignment and, in a single-mode theorem, that larger Gram imbalance suppresses the weaker update channel under a fixed represented scale.

4.3. Empirical Results

To test whether factor imbalance is related to plasticity degradation, we add an explicit balance regularizer,

$$L(U, V) = \ell(UV^\top) + \lambda \|U^\top U - V^\top V\|_F^2.$$

This penalty directly constrains the Gram mismatch between the two factors. Figure 3 shows that balance regularization improves later-stage adaptation across all three continual learning settings. It also reduces factor imbalance, with log-scale imbalance trajectories reported in Appendix J and additional diagnostics in Appendix L.

We further extend this intervention beyond explicitly factorized MLP layers in Appendix I, showing that related geometry regularization can improve continual learning in Transformers and LoRA-style adapter settings.

5. Conclusion

We studied how low-rank factorized parameterizations affect plasticity in non-stationary learning. Across three continual learning settings, low-rank MLPs lose plasticity faster than full-matrix counterparts. Frozen-factor controls show that this degradation is not due to rank constraint alone, but instead implicates coupled factor optimization. We identify factor imbalance, measured by the Gram mismatch $U^\top U - V^\top V$, as a diagnostic associated with this degradation, and show that directly regularizing this mismatch reduces imbalance and partially restores plasticity. Appendix experiments further suggest that related geometry regularization can influence continual adaptation in Transformer and LoRA-style adapters, leaving large-scale pretrained PEFT, normalization, and stochastic regularization as important future directions.

Acknowledgements

This work was supported by the Ministry of Trade, Industry and Energy (MOTIE), Korea, through the Global Industrial Technology Cooperation Center program, supervised by the Korea Institute for Advancement of Technology (KIAT) (Grant No. P0028435).

Impact Statement

This work studies plasticity loss in low-rank factorized neural networks under non-stationary learning. Understanding this phenomenon can help improve the reliability of parameter-efficient and continual learning systems, which are increasingly relevant for models that must adapt to changing data distributions. Such systems may be useful in dynamic environments, including robotics, autonomous systems, and personalized AI services. This paper is primarily empirical and analytical, and we do not identify specific societal or ethical risks beyond those generally associated with advances in machine learning.

References

Abbas, Z., Zhao, R., Modayil, J., White, A., and Machado, M. C. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

Ahn, H., Hyeon, J., Shin, H., and Moon, T. Revisiting warm-start training: No generalization loss under standard training schemes. In *AI That Keeps Up: NeurIPS 2025 Workshop on Continual and Compatible Foundation Model Updates*, 2025. URL https://openreview.net/forum?id=243zzBgRLm.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. *Advances in neural information processing systems*, 32, 2019.

Ash, J. and Adams, R. P. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Berariu, T., Czarnecki, W., De, S., Bornschein, J., Smith, S., Pascanu, R., and Clopath, C. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021.

Ceron, J. S. O., Bellemare, M. G., and Castro, P. S. Small batch deep reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=wPqEvmwFEh.

Dohare, S., Hernandez-Garcia, J. F., Lan, Q., Rahman, P., Mahmood, A. R., and Sutton, R. S. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. *Advances in neural information processing systems*, 30, 2017.

Guo, Y., Wang, G., and Kankanhalli, M. Pela: Learning parameter-efficient models with low-rank approximation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15699–15709, 2024.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, W., Xiao, L., and Pennington, J. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgqN1SYvr.

Huh, M., Mobahi, H., Zhang, R., Cheung, B., Agrawal, P., and Isola, P. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=bCiNWDmlY2.

Idelbayev, Y. and Carreira-Perpinán, M. A. Low-rank compression of neural nets: Learning the rank of each layer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8049–8059, 2020.

Jing, L., Zbontar, J., et al. Implicit rank-minimizing autoencoder. *Advances in Neural Information Processing Systems*, 33:14736–14746, 2020.

Khodak, M., Tenenholtz, N. A., Mackey, L., and Fusi, N. Initialization and regularization of factorized neural layers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KTlJT1nof6d.

Kobayashi, S., Akram, Y., and Von Oswald, J. Weight decay induces low-rank attention layers. *Advances in Neural Information Processing Systems*, 37:4481–4510, 2024.

Kumar, A., Agarwal, R., Ghosh, D., and Levine, S. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O9bnihsFfXU.

Lee, H., Cho, H., Kim, H., Kim, D., Min, D., Choo, J., and Lyle, C. Slow and steady wins the race: Maintaining plasticity with hare and tortoise networks. In *International Conference on Machine Learning*, pp. 26416–26438. PMLR, 2024.

Lewandowski, A., Bortkiewicz, M., Kumar, S., György, A., Schuurmans, D., Ostaszewski, M., and Machado, M. C. Learning continually by spectral regularization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Hcb2cgPbMg.

Lu, A., Yuan, H., Feng, T., and Sun, Y. Rethinking the stability-plasticity trade-off in continual learning from an architectural perspective. In *International Conference on Machine Learning*, pp. 40888–40902. PMLR, 2025.

Lyle, C., Zheng, Z., Nikishin, E., Pires, B. A., Pascanu, R., and Dabney, W. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

Lyle, C., Zheng, Z., Khetarpal, K., Hasselt, H. v., Pascanu, R., Martens, J., and Dabney, W. Disentangling the causes of plasticity loss in neural networks. In Lomonaco, V., Melacci, S., Tuytelaars, T., Chandar, S., and Pascanu, R. (eds.), *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 750–783. PMLR, 29 Jul–01 Aug 2025. URL https://proceedings.mlr.press/v274/lyle25a.html.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *The Second International Conference on Learning Representations*, 2014. URL https://openreview.net/forum?id=_wzZwKpTDF_9C.

Shin, B., Oh, J., Cho, H., and Yun, C. Dash: Warm-starting neural network training in stationary settings without loss of plasticity. *Advances in Neural Information Processing Systems*, 37:43300–43340, 2024.

Sokar, G., Agarwal, R., Castro, P. S., and Evci, U. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 32145–32168. PMLR, 2023.

Timor, N., Vardi, G., and Shamir, O. Implicit regularization towards rank minimization in relu networks. In *International Conference on Algorithmic Learning Theory*, pp. 1429–1459. PMLR, 2023.

Zhang, L., Zhang, L., Shi, S., Chu, X., and Li, B. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023.

Zhu, J., Greenewald, K., Nadjahi, K., Borde, H. S. D. O., Gabrielsson, R. B., Choshen, L., Ghassemi, M., Yurochkin, M., and Solomon, J. Asymmetry in low-rank adapters of foundation models. In *International Conference on Machine Learning*, pp. 62369–62385. PMLR, 2024a.

Zhu, Z., Wu, Y., Gu, Q., and Cevher, V. Imbalance-regularized loRA: A plug-and-play method for improving fine-tuning of foundation models. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024b. URL https://openreview.net/forum?id=hQfJ7nyDeK.

Zilly, J., Achille, A., Censi, A., and Frazzoli, E. On plasticity, invariance, and mutually frozen weights in sequential task learning. *Advances in neural information processing systems*, 34:12386–12399, 2021.

A. Experimental Details

All models use a two-hidden-layer ReLU MLP with hidden width 512, followed by a dense linear classification head. The classification head is not factorized and remains a full linear layer for both full-matrix and low-rank models. For low-rank models, only the hidden linear layers are factorized: each hidden weight matrix W_l is represented as $W_l = U_l V_l^\top$ with bottleneck rank r. We keep the depth and hidden width fixed across experiments and vary only r. The full-matrix model is denoted by $r = \mathrm{None}$.

We evaluate three continual learning protocols. In task-streaming learning, CIFAR-100 is split into 10 disjoint class chunks, each containing 10 classes. At each stage, the model is trained only on the current class chunk, and evaluation is performed on the test examples from that chunk. This setting measures adaptation to a sequential stream of tasks rather than retention of previous tasks. In sample-incremental learning, CIFAR-10 is split into 10 disjoint sample chunks while keeping the label space fixed. Training starts from the first chunk, one additional chunk is added at each subsequent stage, and evaluation is performed on the full CIFAR-10 test set. In class-incremental learning, CIFAR-100 is split into 10 disjoint 10-class chunks; at stage t, the model is trained on the union of all chunks observed up to that stage and evaluated on the test examples from the observed classes.

All models are trained with AdamW using a learning rate of 10^{-3}, batch size 256, and zero decoupled weight decay. We use gradient clipping with maximum norm 0.5 in all experiments. Each continual learning experiment consists of 10 stages with 100 training epochs per stage, where an epoch is defined with respect to the training data available at the current stage. Following Lee et al. (2024), we reset the optimizer state at the beginning of each stage. For *Vanilla* continual training, model parameters are carried forward across stages while the optimizer state is reset. For the *Full Reset* reference, model parameters are reinitialized at the beginning of each stage and trained on the data available at that stage using the same per-stage training budget. All reported results are averaged over five random seeds.

B. Hyperparameter Search Space

For each regularized experiment, we select the coefficient that gives the highest final-stage test accuracy. For balance regularization, the coefficient denotes the strength of the Gram-mismatch penalty. Table 1 reports the search grid and the selected coefficients.

Table 1. Hyperparameter search space and selected coefficients for each experiment.

Experiment	**Method**	**Search Space**	**r=512**	**r=256**	**r=128**	**r=64**	**r=32**
Task-streaming Learning	Balance Reg.	$1e{+}0, 1e{+}1, 1e{+}2, 1e{+}3$	$1e{+}2$	$1e{+}2$	$1e{+}1$	$1e{+}1$	$1e{+}1$
Sample-incremental Learning	Balance Reg.	$1e{+}0, 1e{+}1, 1e{+}2, 1e{+}3$	$1e{+}3$	$1e{+}3$	$1e{+}2$	$1e{+}2$	$1e{+}1$
Class-incremental Learning	Balance Reg.	$1e{+}0, 1e{+}1, 1e{+}2, 1e{+}3$	$1e{+}3$	$1e{+}3$	$1e{+}3$	$1e{+}3$	$1e{+}2$

C. Effective Update Geometry of Factorized Layers

In this section, we provide the first-order update calculation used to interpret factor imbalance in Section 4.2. The goal is not to show that the imbalance measure $\Delta(U, V)$ alone determines optimization difficulty, but to make explicit how the update induced by a factorized parameterization depends on the current factors.

C.1. Factor Gradients and Induced Updates

Consider a factorized weight matrix

$$W = UV^\top, \qquad U \in \mathbb{R}^{m\times r}, \qquad V \in \mathbb{R}^{n\times r}, \qquad W \in \mathbb{R}^{m\times n}.$$

Let $\ell(W)$ be a differentiable loss and let

$$G = \nabla_W \ell(W) \in \mathbb{R}^{m\times n}.$$

For infinitesimal perturbations dU and dV, the corresponding perturbation of the composed matrix is

$$dW = dUV^\top + UdV^\top.$$

Therefore,

$$\begin{aligned} d\ell &= \langle G, dW\rangle \\ &= \langle G, dUV^\top\rangle + \langle G, UdV^\top\rangle \\ &= \langle GV, dU\rangle + \langle G^\top U, dV\rangle. \end{aligned}$$

Thus the factor gradients are

$$\nabla_U \ell(UV^\top) = GV, \qquad \nabla_V \ell(UV^\top) = G^\top U.$$

A gradient descent step on the two factors gives

$$U^+ = U - \eta GV, \qquad V^+ = V - \eta G^\top U.$$

The induced update on the composed matrix is

$$\begin{aligned} W^+ &= U^+(V^+)^\top \\ &= (U - \eta GV)(V - \eta G^\top U)^\top \\ &= (U - \eta GV)(V^\top - \eta U^\top G) \\ &= UV^\top - \eta GVV^\top - \eta UU^\top G + \eta^2 GVU^\top G. \end{aligned}$$

Hence

$$W^+ - W = -\eta\left(GVV^\top + UU^\top G\right) + \eta^2 GVU^\top G.$$

To first order in η, the factorized parameterization therefore induces the effective update

$$\Delta W_{\text{fac}} = -\eta \mathcal{A}_{U,V}(G), \qquad \mathcal{A}_{U,V}(G) = GVV^\top + UU^\top G.$$

This shows that the effective update is not determined by the full-matrix gradient G alone. It also depends on the current factor covariances $UU^\top$ and $VV^\top$, which are themselves determined by the particular factorization used to represent W.

C.2. First-Order Progress Under a New Loss

We next relate the induced update to progress after a distribution shift. Let $\ell_{\text{new}}(W)$ denote the loss on a newly introduced distribution, task, or set of classes, and let

$$G = \nabla_W \ell_{\text{new}}(W).$$

A full-matrix parameterization updates W directly along $-\eta G$. In contrast, the factorized parameterization updates W, to first order, along

$$-\eta \mathcal{A}_{U,V}(G) = -\eta\left(GVV^\top + UU^\top G\right).$$

Assume that ℓ_{new} is locally L-smooth. Then

$$\ell_{\text{new}}(W - \eta \mathcal{A}_{U,V}(G)) \leq \ell_{\text{new}}(W) - \eta \langle G, \mathcal{A}_{U,V}(G) \rangle + \frac{L\eta^2}{2} \|\mathcal{A}_{U,V}(G)\|_F^2.$$

The first-order decrease term is

$$\begin{aligned}
\langle G, \mathcal{A}_{U,V}(G) \rangle &= \langle G, GVV^\top \rangle + \langle G, UU^\top G \rangle \\
&= \operatorname{tr}(G^\top GVV^\top) + \operatorname{tr}(G^\top UU^\top G) \\
&= \operatorname{tr}(V^\top G^\top GV) + \operatorname{tr}(G^\top UU^\top G) \\
&= \|GV\|_F^2 + \|U^\top G\|_F^2.
\end{aligned}$$

Thus, first-order progress under the factorized update depends on the alignment between the new gradient G and the current factors U and V. If the new gradient has components that are weakly represented through both factor geometries, then those components can contribute little first-order progress even when $\|G\|_F$ is large. Conversely, components aligned with highly scaled factor directions can be amplified.

This calculation does not imply that the imbalance measure $\Delta(U, V) = \|U^\top U - V^\top V\|_F$ alone determines adaptation speed. Progress also depends on the direction of the new gradient G relative to the current factor directions. Rather, the calculation shows that factorized updates are inherently factor-dependent: the same full-matrix gradient can produce different effective updates depending on the internal geometry of U and V.

C.3. A Mode-Wise Illustration

We next give a simplified mode-wise illustration of how factor geometry can create non-uniform effective learning rates. This calculation is not intended as a complete spectral characterization of the nonlinear network dynamics; it only describes gradient modes aligned with the current factor directions.

Suppose locally that

$$U = Q \operatorname{diag}(a_i), \qquad V = P \operatorname{diag}(b_i),$$

where $Q = [q_1, \ldots, q_r] \in \mathbb{R}^{m \times r}$ and $P = [p_1, \ldots, p_r] \in \mathbb{R}^{n \times r}$ have orthonormal columns. Consider a rank-one gradient mode

$$G_{ij} = q_i p_j^\top.$$

Then

$$G_{ij} VV^\top = q_i p_j^\top P \operatorname{diag}(b_k^2) P^\top = b_j^2 q_i p_j^\top = b_j^2 G_{ij},$$

and

$$UU^\top G_{ij} = Q \operatorname{diag}(a_k^2) Q^\top q_i p_j^\top = a_i^2 q_i p_j^\top = a_i^2 G_{ij}.$$

Therefore,

$$\mathcal{A}_{U,V}(G_{ij}) = G_{ij} VV^\top + UU^\top G_{ij} = (a_i^2 + b_j^2) G_{ij}.$$

For this factor-aligned mode, the factorized update behaves as if it used the mode-dependent effective learning rate

$$\eta_{ij}^{\text{eff}} = \eta(a_i^2 + b_j^2).$$

This illustration shows how the current factor geometry can make the induced update anisotropic across gradient modes. If the quantities $a_i^2 + b_j^2$ vary widely, then some modes are updated with much larger effective step sizes than others. In a non-stationary setting, adaptation to a new distribution may require movement along modes that are not strongly scaled by the current factors. In such cases, factor-dependent anisotropy can slow adaptation along those directions. Again, this is not a statement that $\Delta(U, V)$ alone determines plasticity; rather, it clarifies why changes in the internal factor geometry can matter for future updates.

C.4. Gram Imbalance and Suppressed Update Channels

We finally connect the latent Gram imbalance to the anisotropy of the update channels induced by a factorized parameterization. The goal is not to show that imbalance reduces progress for every gradient direction, but to show that, under a fixed represented scale, imbalance necessarily suppresses one of the two channels.

Recall that the first-order update induced by a factorized matrix $W = UV^\top$ is

$$\mathcal{A}_{U,V}(G) = GVV^\top + UU^\top G.$$

The imbalance measure $\|U^\top U - V^\top V\|_F$ is computed in the shared latent coordinate space, whereas the update operator depends on the ambient covariances $UU^\top$ and $VV^\top$. These quantities are nevertheless spectrally linked: $U^\top U$ and $UU^\top$ have the same nonzero eigenvalues, and the same holds for $V^\top V$ and $VV^\top$. Thus, latent Gram imbalance reflects a mismatch between the singular-value geometries of the two factors, which in turn set the spectral scales of the two update channels.

We make this connection explicit in a single-mode setting.

Theorem C.1 (Balanced factors maximize the weaker update channel)**.** *Consider a single paired factor mode*

$$U = q\sqrt{a}, \qquad V = p\sqrt{b},$$

where $q \in \mathbb{R}^m$, $p \in \mathbb{R}^n$, $\|q\|_2 = \|p\|_2 = 1$, and $a, b > 0$. Suppose the represented scale is fixed:

$$ab = \sigma^2$$

for some $\sigma > 0$. Then the balanced factorization

$$a = b = \sigma$$

uniquely maximizes the weaker update-channel scale

$$s(a, b) = \min(a, b).$$

Moreover, under the same constraint, $s(a, b)$ is a strictly decreasing function of the mode-wise Gram imbalance

$$\delta(a, b) = |a - b|.$$

Proof. For the single-mode factors,

$$U^\top U = a, \qquad V^\top V = b,$$

so the mode-wise Gram imbalance is

$$\delta(a, b) = |a - b|.$$

The composed matrix is

$$W = UV^\top = \sqrt{ab}\, qp^\top.$$

Thus, fixing $ab = \sigma^2$ fixes the represented scale of this mode.

The factor-induced update operator is

$$\mathcal{A}_{U,V}(G) = GVV^\top + UU^\top G = b\, Gpp^\top + a\, qq^\top G.$$

Hence the two update channels have scale coefficients b and a, respectively. The weaker channel is therefore

$$s(a, b) = \min(a, b).$$

Using the constraint $ab = \sigma^2$, write

$$b = \frac{\sigma^2}{a}.$$

Then

$$s(a, b) = \min\left(a, \frac{\sigma^2}{a}\right).$$

If $a \leq \sigma$, then

$$s(a, b) = a \leq \sigma.$$

If $a \geq \sigma$, then

$$s(a, b) = \frac{\sigma^2}{a} \leq \sigma.$$

Therefore,

$$s(a, b) \leq \sigma,$$

with equality if and only if $a = \sigma$, equivalently $a = b = \sigma$. Thus the balanced factorization uniquely maximizes the weaker update channel.

It remains to show the monotone relation between imbalance and the weaker channel. Parameterize the fixed-scale constraint by

$$a = \sigma e^{c}, \qquad b = \sigma e^{-c}$$

for some $c \in \mathbb{R}$. Let $x = |c|$. Since $|\sinh c| = \sinh x$,

$$\delta(a, b) = |a - b| = 2\sigma \sinh x,$$

while

$$s(a, b) = \min(\sigma e^{c}, \sigma e^{-c}) = \sigma e^{-x}.$$

The quantity $2\sigma \sinh x$ is strictly increasing in $x \geq 0$, whereas σe^{-x} is strictly decreasing in $x \geq 0$. Therefore, along the fixed-scale constraint, $s(a, b)$ decreases monotonically as $\delta(a, b)$ increases. □

Remark C.2. Theorem C.1 does not imply that imbalance reduces progress for every gradient direction. If a new gradient aligns with the amplified channel, imbalance can increase the corresponding update scale. The result instead shows that, for a fixed represented mode, imbalance necessarily suppresses one of the two update channels and makes the update more anisotropic. In a non-stationary setting, where future gradients are not known in advance, this suppression can make adaptation more sensitive to the current factor state. Thus, the theorem should be interpreted as a channel-wise mechanism rather than as a monotone prediction of plasticity from $\Delta(U, V)$ alone.

D. Initialization Experiments

In this section, we compare three initialization schemes for low-rank factorized networks: Kaiming, spectral, and orthogonal initialization. Spectral initialization was proposed as a principled way to initialize the factors so that training remains close to the optimization behavior of a well-tuned unfactorized layer, and it was shown to provide empirical gains across multiple factorized-model settings (Khodak et al., 2021). Orthogonal initialization was proposed because prior theoretical work showed that, in deep linear networks, it can accelerate convergence relative to standard Gaussian initialization and requires a width for efficient convergence that does not grow with depth, highlighting its potential to preserve more stable optimization geometry (Hu et al., 2020). Although the orthogonal result is derived for deep linear networks, it still provides a useful motivation for testing whether a better-conditioned initialization can alleviate the optimization difficulties of low-rank factorization in our setting. The goal is to test whether the main conclusion of the paper—that low-rank factorization is disadvantaged in non-stationary learning—depends strongly on the choice of initialization.

D.1. Stationary Setting

We first compare the three initialization schemes in a stationary setting. For this experiment, we randomly sample 10% of the CIFAR-10 training set and train each model with batch size 256 for a total of 20K update steps. All other architectural and optimization choices are kept fixed so that the observed differences can be attributed primarily to initialization.

The results suggest that low-rank factorization is not by itself a major obstacle in the stationary regime. Across the three initialization schemes, low-rank models remain broadly competitive when the data distribution is fixed throughout training. However, when the bottleneck rank r is too small, the model can suffer from insufficient capacity, which limits the achievable generalization performance. This indicates that poor performance in the stationary setting is mainly associated with an overly restrictive rank bottleneck rather than with the factorized parameterization itself.

We also track the balancedness metric over training and observe that it increases gradually over time. This trend appears consistently across initialization schemes, suggesting that even in the stationary regime the two factors do not remain perfectly balanced during optimization.

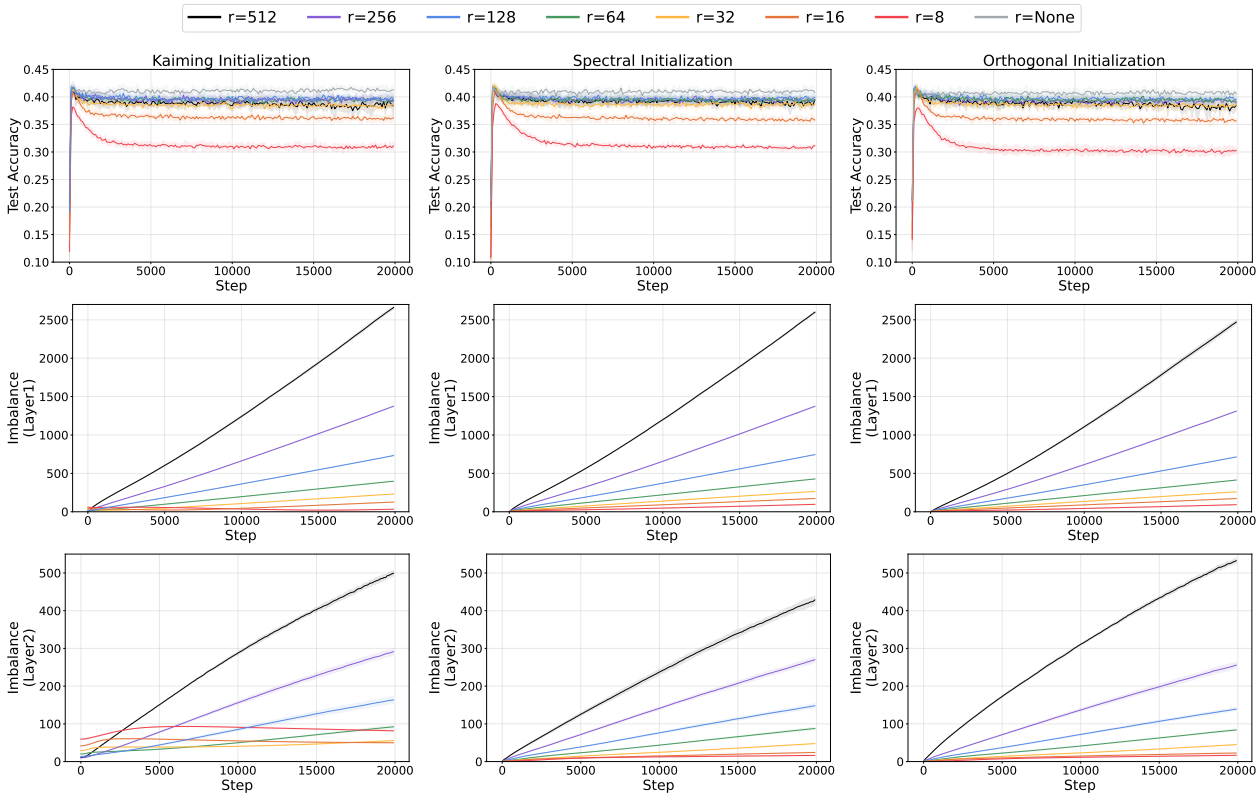

Figure 5. **Supervised Learning with different initialization methods.**

D.2. Continual Learning Setting

We next compare the same three initialization schemes in the continual learning experiments used in the main paper. Here the qualitative picture changes substantially. Relative to their full-matrix counterparts, low-rank factorized models show consistently worse performance under non-stationary training. This degradation appears under Kaiming, spectral, and orthogonal initialization alike.

Therefore, the main conclusion is robust to the choice of initialization: although different schemes can affect early optimization behavior or the precise magnitude of the gap, none of them removes the disadvantage of low-rank factorization in continual learning. These results support the interpretation that the degradation is a property of the factorized dynamics under distributional change, rather than an artifact of a particular initialization recipe.

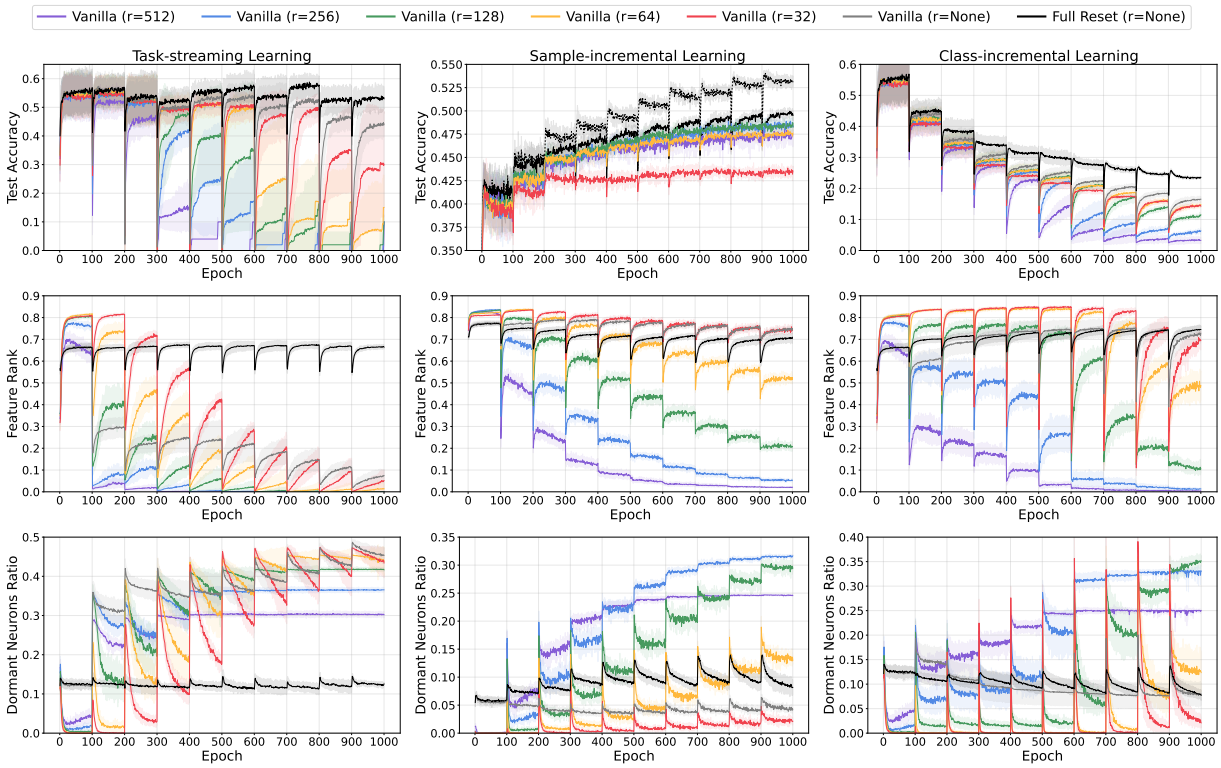

Figure 6. **Continual learning with Kaiming initialization.**

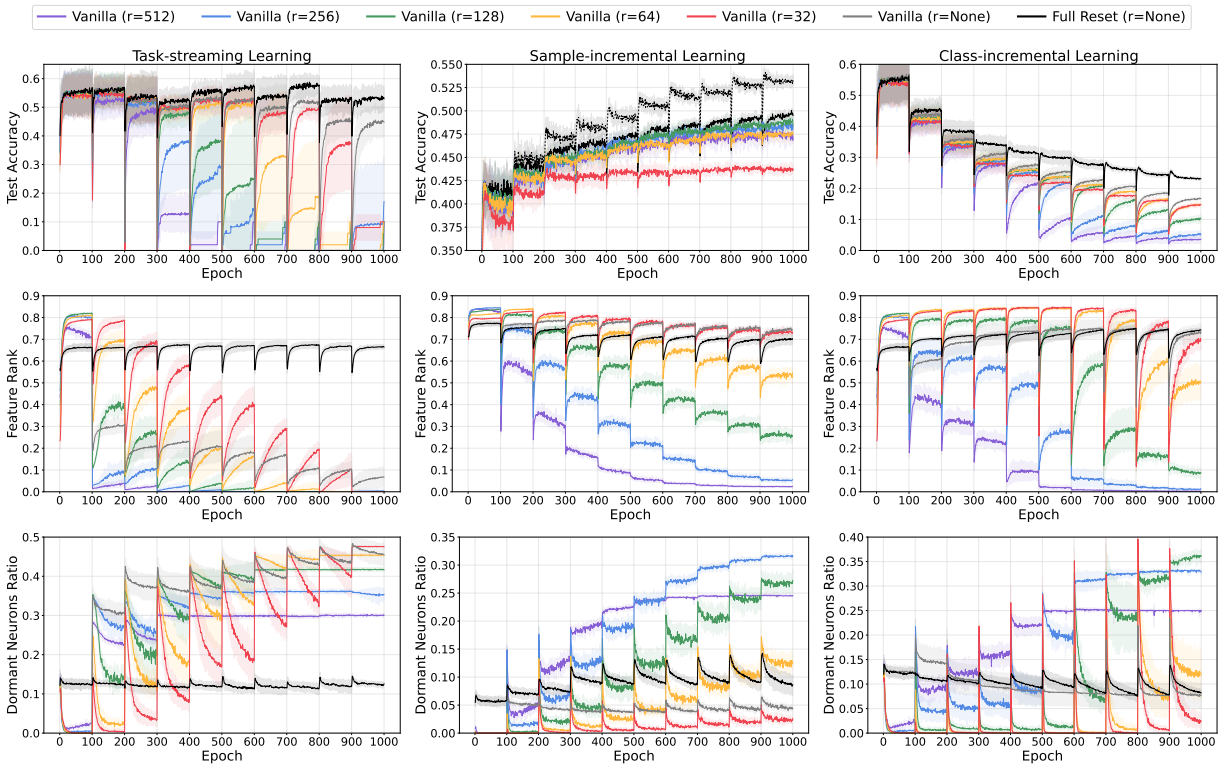

Figure 7. **Continual learning with spectral initialization.**

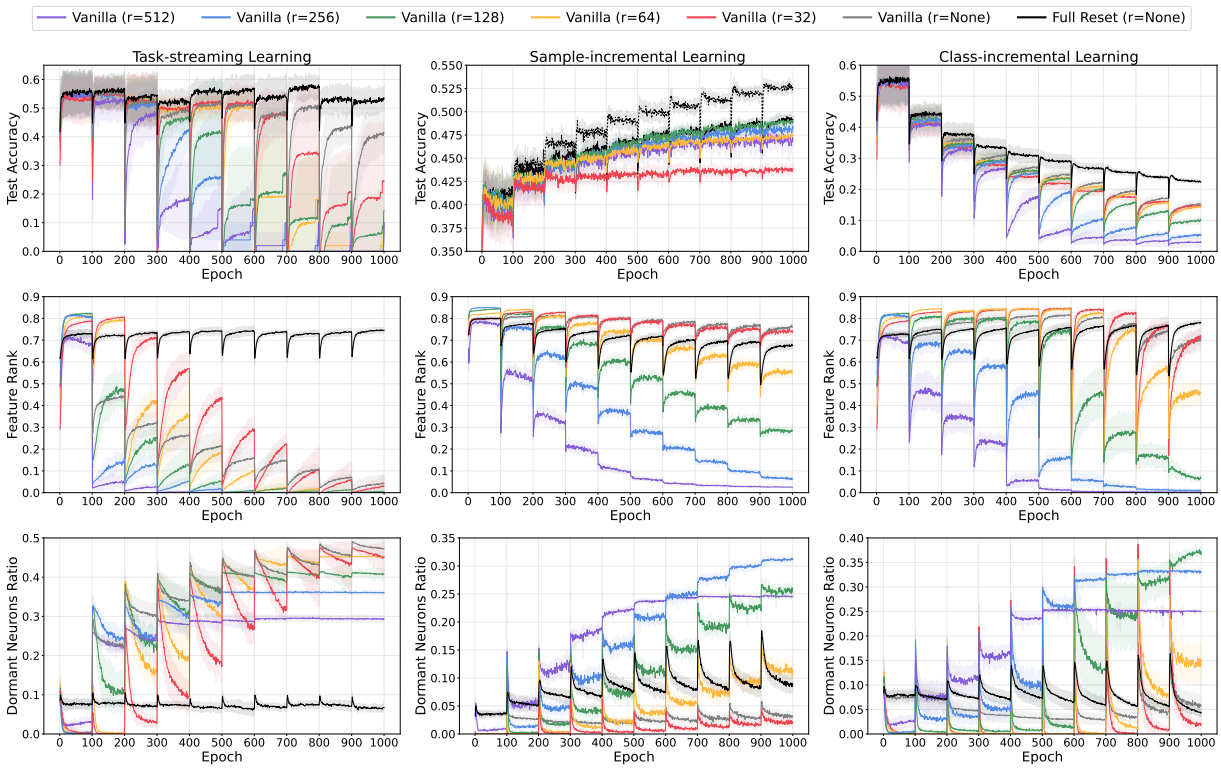

Figure 8. **Continual learning with orthogonal initialization.**

E. Effect of Network Width

Lyle et al. (2025) reported that wider networks can suffer less plasticity loss under non-stationary training, motivating us to test whether increasing width mitigates the degradation of low-rank factorized networks. Figures 9–11 summarize these width-scaling experiments across the three continual learning settings. In addition to the default hidden width of 512, we repeat the same experiments with hidden widths 256 and 1024, keeping all other settings fixed. Across widths, low-rank factorized networks still perform worse than their full-matrix counterparts, indicating that the observed plasticity gap is not eliminated by increasing model capacity. The same diagnostic trends also persist: feature rank decreases and dormant neuron ratio increases more strongly for low-rank factorized networks, especially at larger bottleneck ranks r, consistent with the main results.

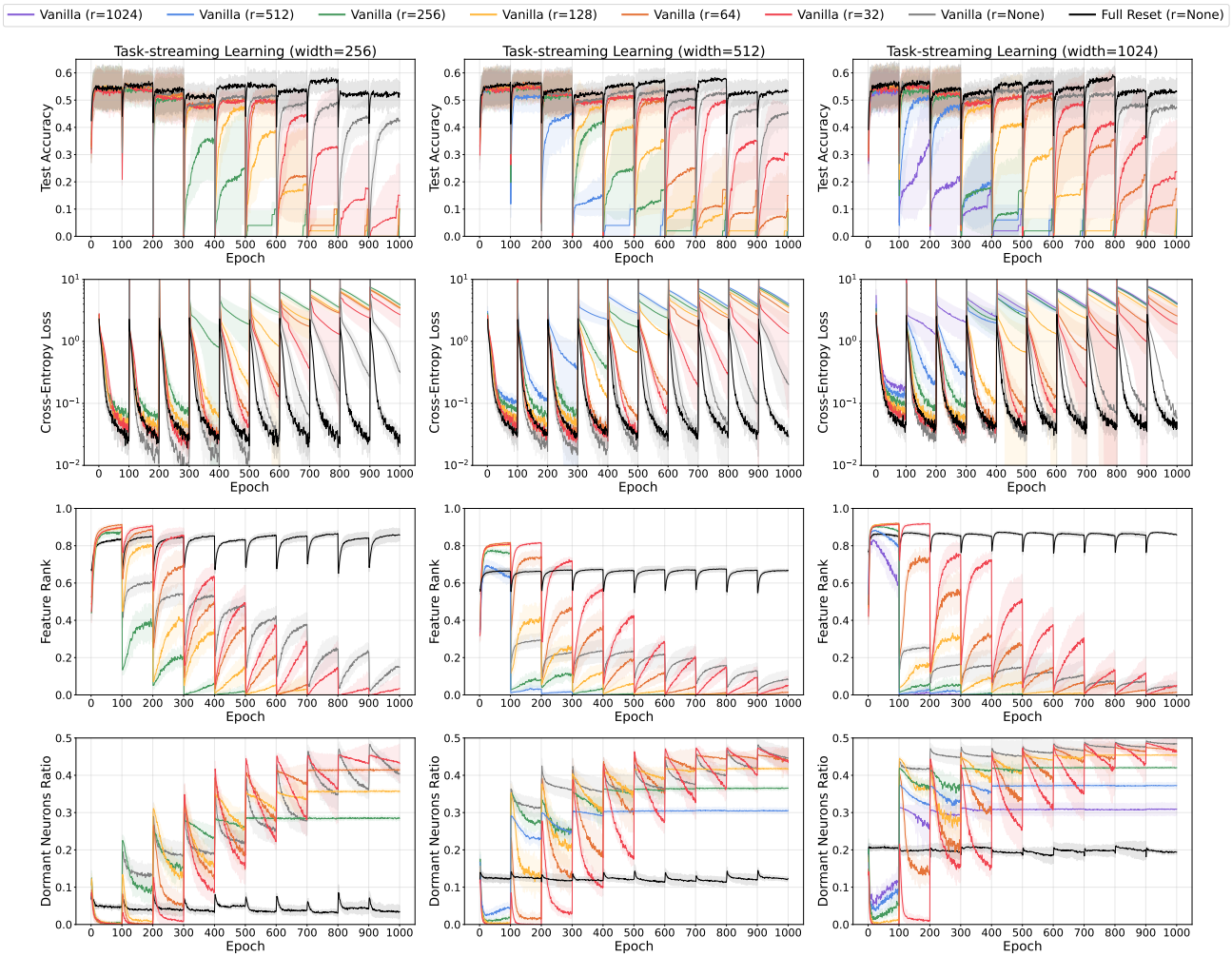

Figure 9. **Effect of network width on task-streaming learning.** Columns show hidden widths 256, 512, and 1024, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Increasing width does not eliminate the plasticity gap: low-rank factorized networks still underperform their full-matrix counterparts, with lower feature rank and higher dormant neuron ratio at larger bottleneck ranks r.

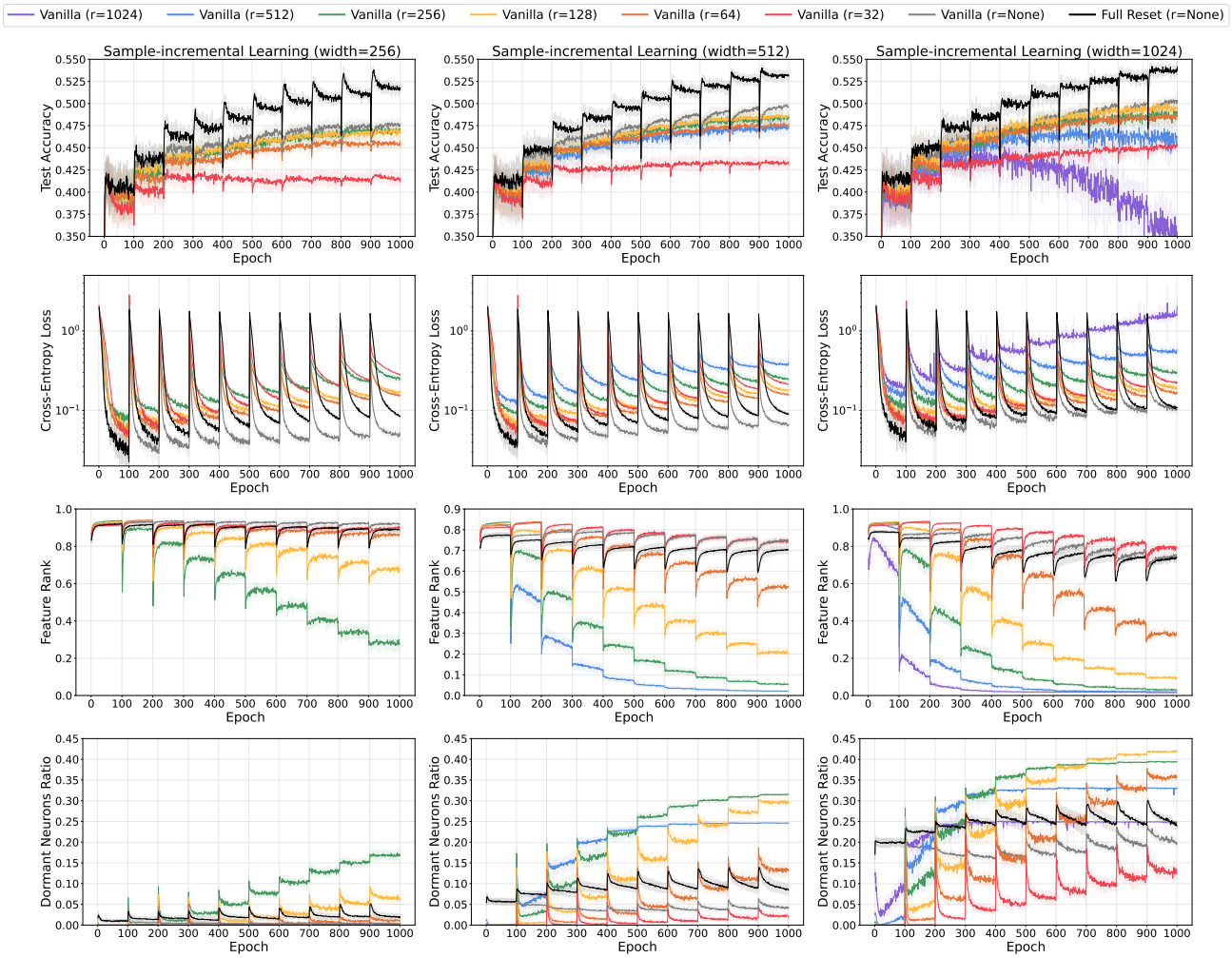

Figure 10. **Effect of network width on sample-incremental learning.** Columns show hidden widths 256, 512, and 1024, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Increasing width does not eliminate the plasticity gap: low-rank factorized networks still underperform their full-matrix counterparts, with lower feature rank and higher dormant neuron ratio at larger bottleneck ranks r.

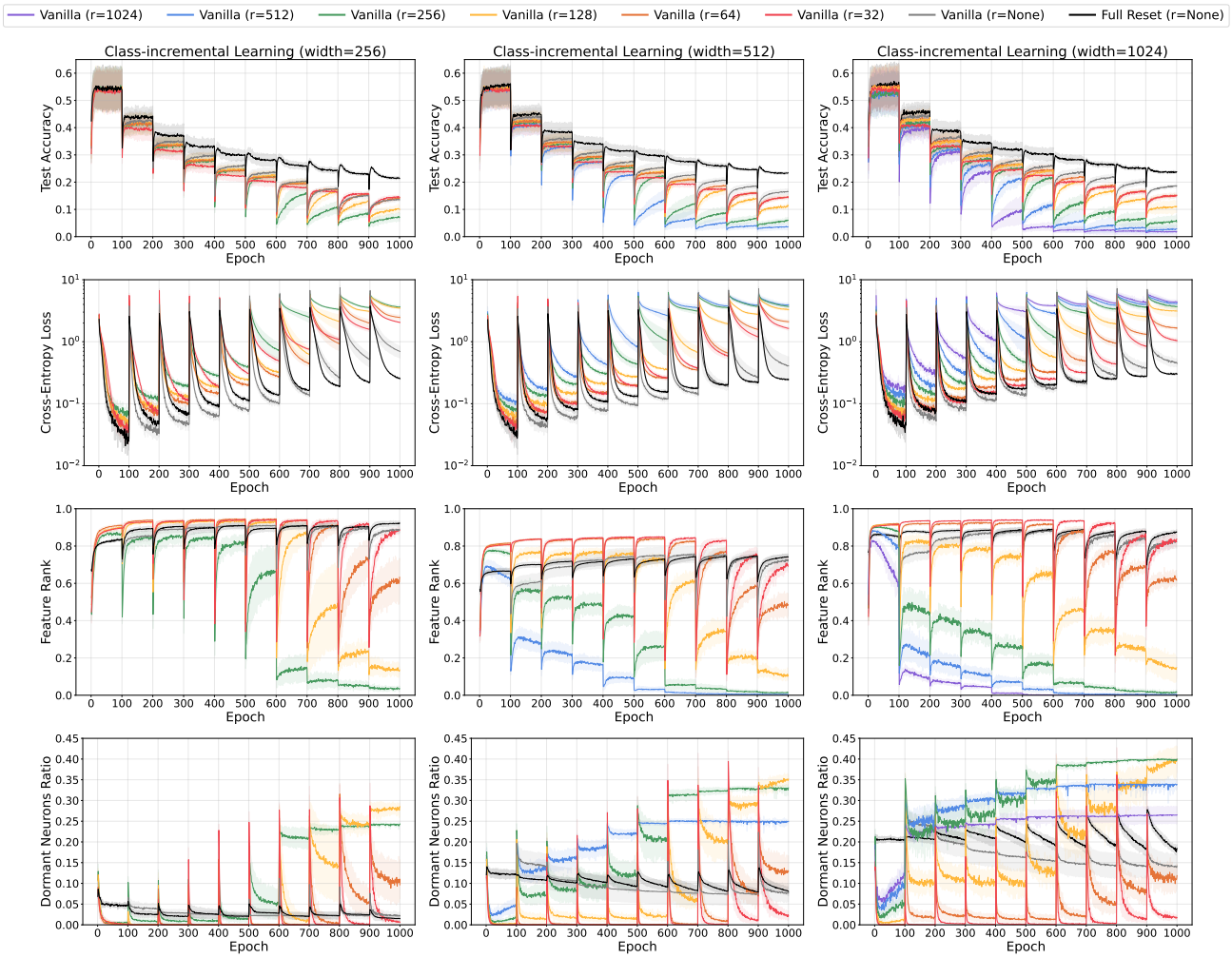

Figure 11. **Effect of network width on class-incremental learning.** Columns show hidden widths 256, 512, and 1024, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Increasing width does not eliminate the plasticity gap: low-rank factorized networks still underperform their full-matrix counterparts, with lower feature rank and higher dormant neuron ratio at larger bottleneck ranks r.

F. Effect of Network Depth

Prior work on plasticity loss has shown that the effect of depth is task-dependent: while deeper networks can help in some structured nonstationary tasks, they may also reduce performance and accelerate plasticity loss in harder settings without additional hyperparameter tuning (Lyle et al., 2025). We therefore use depth variation as a stress test for low-rank factorization rather than assuming that depth should improve plasticity. As shown in Figures 12–14, increasing depth does not eliminate the plasticity gap. Although deeper full-matrix MLPs can also show degraded accuracy, lower feature rank, and increased neuron dormancy in our fixed-width setting, low-rank factorized models consistently underperform their full-matrix counterparts across depths. These results suggest that factorized parameterization adds a distinct source of degradation on top of any depth-induced plasticity loss. This is consistent with the broader architectural perspective that depth and width can shape continual learning behavior (Lu et al., 2025), but shows that depth-induced benefits reported under other architectural regimes do not directly carry over to fixed-width low-rank MLPs.

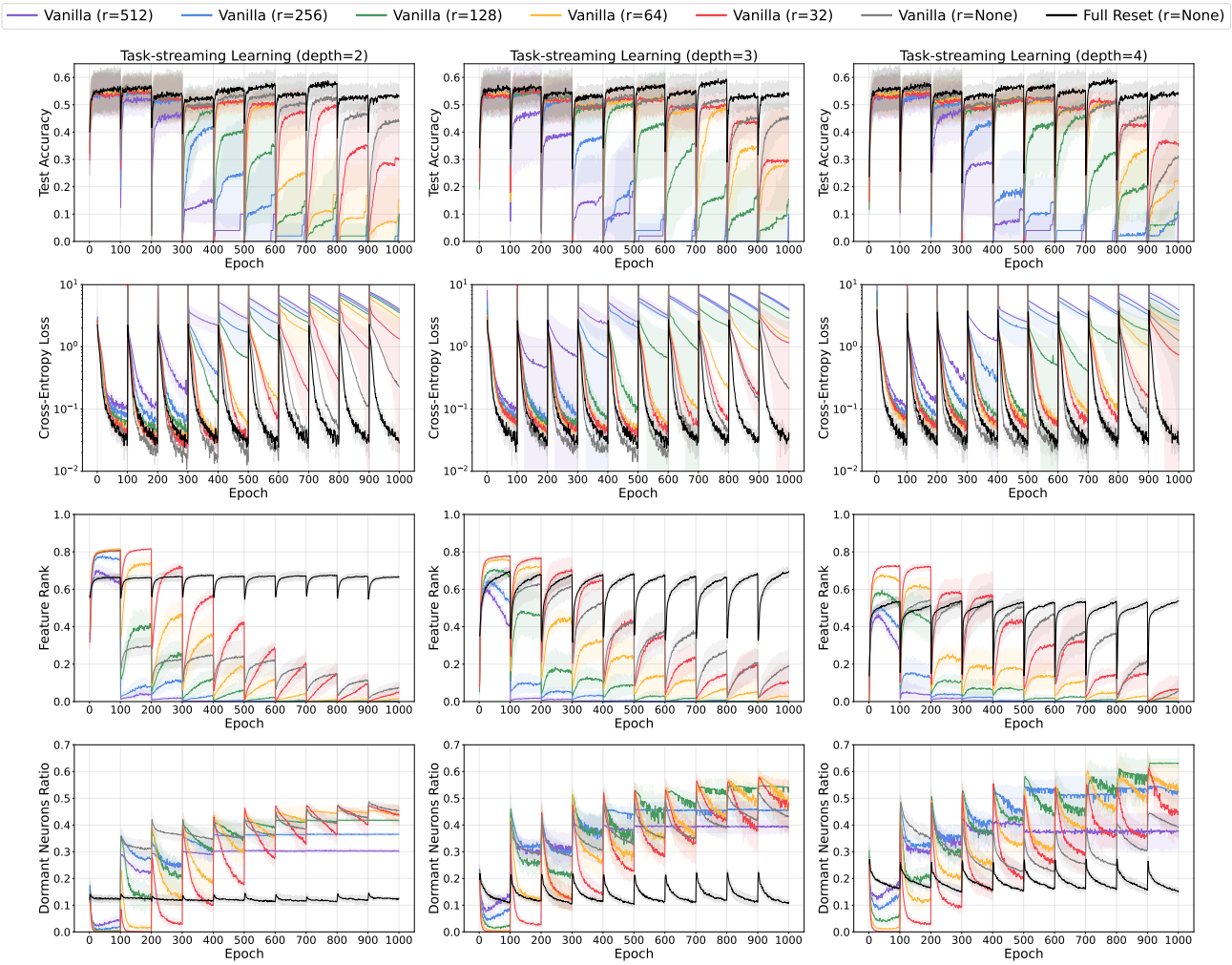

Figure 12. **Effect of network depth on task-streaming learning.** Columns show network depths 2, 3, and 4, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Deeper low-rank networks may delay the onset of degradation, but they still suffer substantial later-stage plasticity loss relative to full-matrix counterparts.

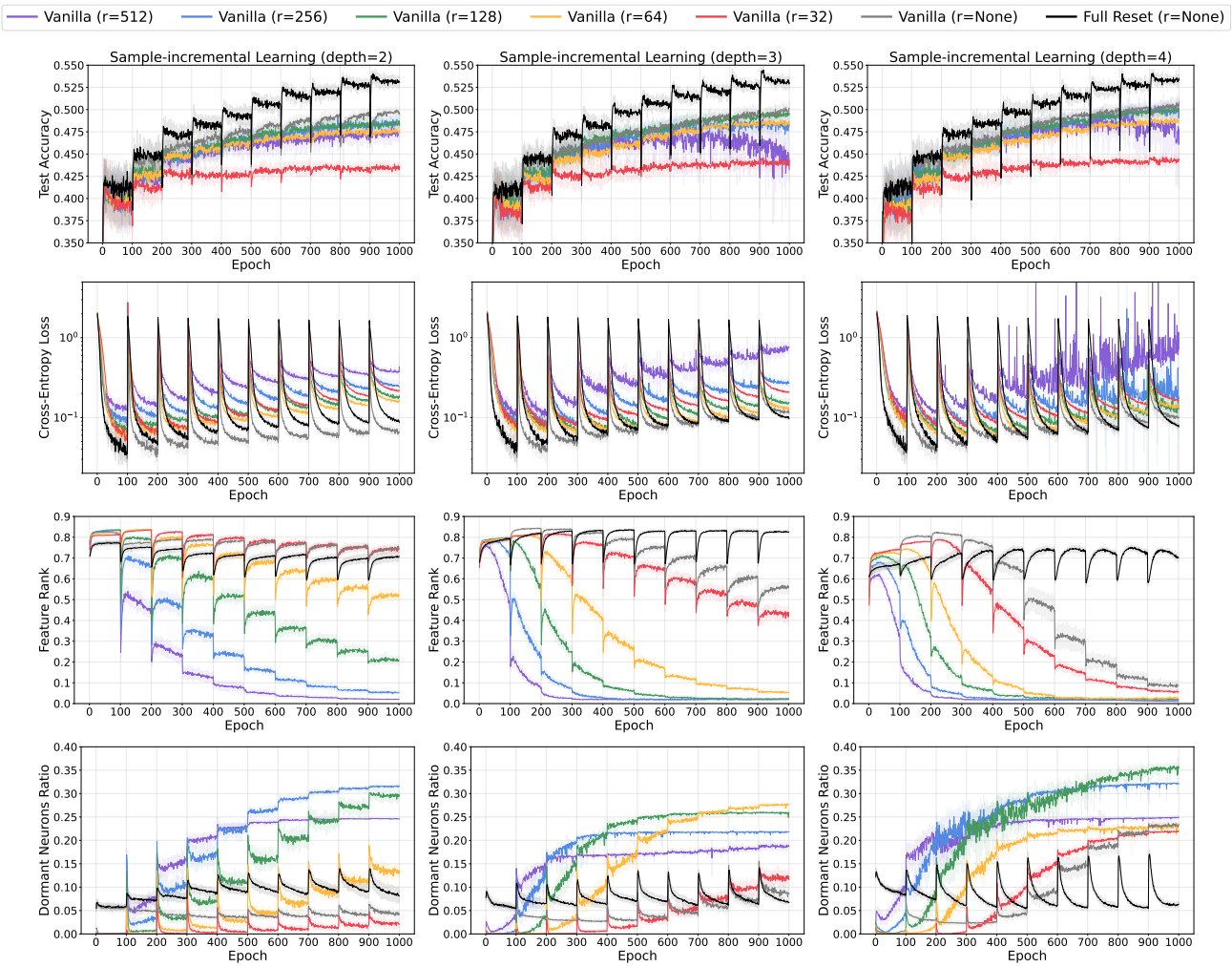

Figure 13. **Effect of network depth on sample-incremental learning.** Columns show network depths 2, 3, and 4, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Deeper low-rank factorized networks lose plasticity more rapidly, showing faster feature-rank decay and increased neuron dormancy.

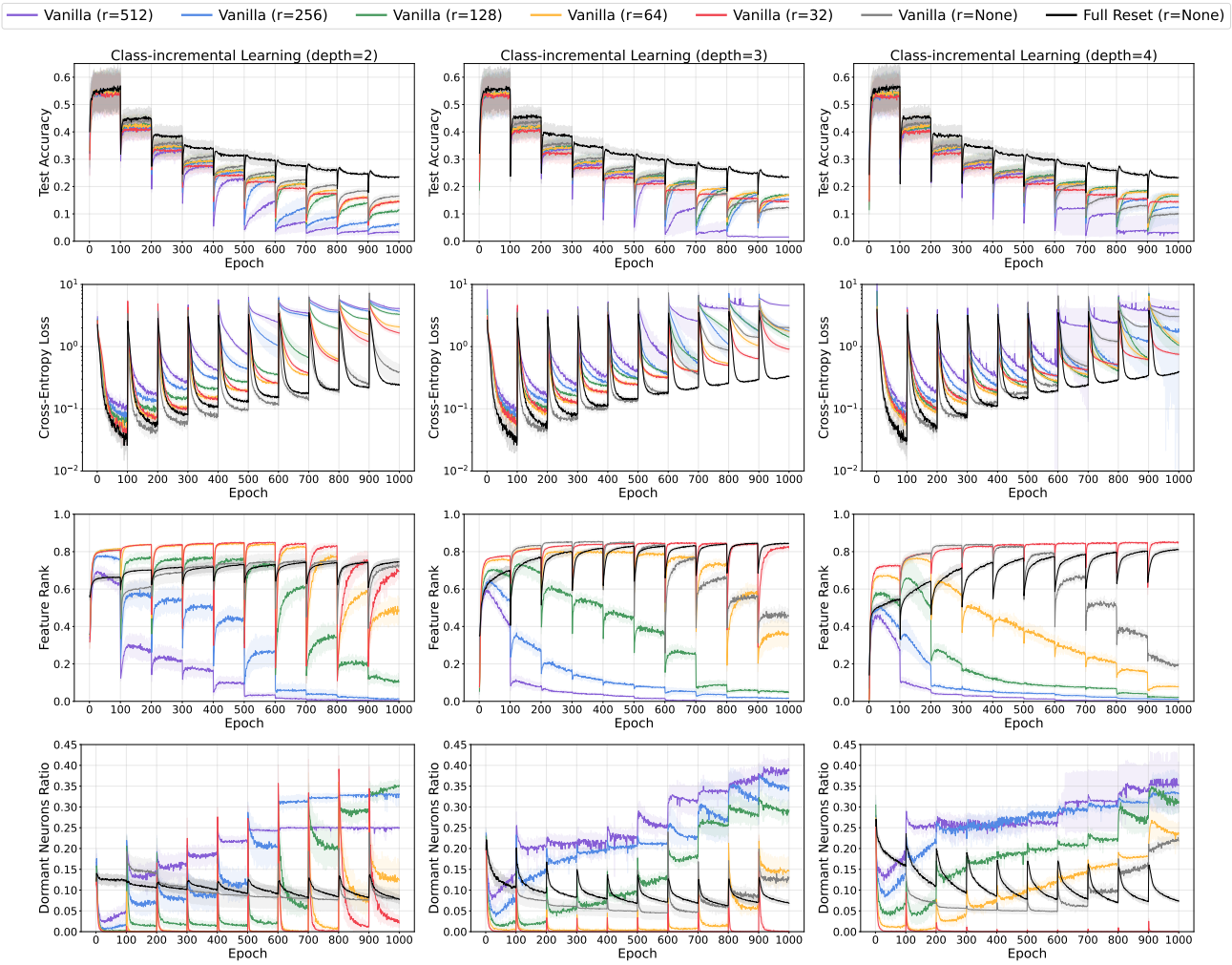

Figure 14. **Effect of network depth on class-incremental learning.** Columns show network depths 2, 3, and 4, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Deeper low-rank factorized networks lose plasticity more rapidly, showing faster feature-rank decay and increased neuron dormancy.

G. Effect of Batch Size

Motivated by the observation of Ceron et al. (2023) that smaller batch sizes can improve performance in non-stationary reinforcement learning, we examine whether batch size also affects plasticity loss in our supervised continual learning settings. Starting from the main setting with batch size 256 and learning rate 10^{-3}, we additionally test batch sizes 128 and 64. Because our experiments are organized by epochs, smaller batches increase the number of optimizer updates per epoch; therefore, to approximately preserve the average update scale per epoch, we reduce the learning rate in proportion to the batch size, using 10^{-3}, 5×10^{-4}, and 2.5×10^{-4} for batch sizes 256, 128, and 64, respectively. All other experimental settings are kept fixed.

Figures 15–17 summarize the results. Smaller batch sizes consistently mitigate plasticity loss: as the batch size decreases from 256 to 128 and 64, later-stage degradation becomes weaker and continual training retains better adaptation ability. This trend is consistent with the observation of Ceron et al. (2023) that small-batch training can improve performance under non-stationary reinforcement learning, suggesting a similar benefit in supervised continual learning. However, the main conclusion remains unchanged: across all tested batch sizes, low-rank factorized models still underperform their full-matrix counterparts. Thus, smaller batches reduce the overall severity of plasticity loss, but do not eliminate the additional vulnerability introduced by low-rank factorized parameterization.

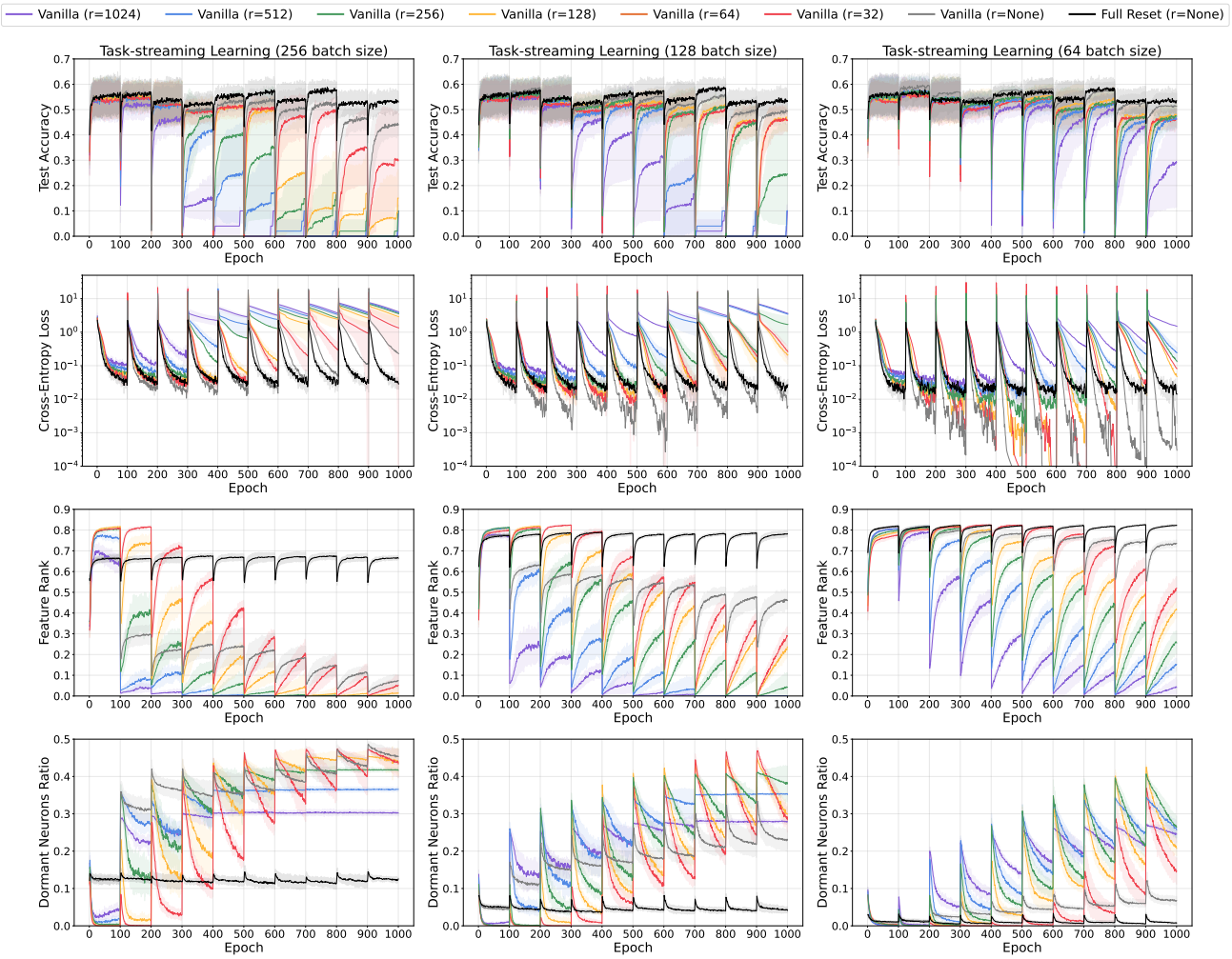

Figure 15. **Effect of batch size on task-streaming learning.** Columns show batch sizes 256, 128, and 64, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Smaller batch sizes mitigate plasticity loss, yielding weaker later-stage degradation and improved plasticity-related diagnostics. Results are averaged over five random seeds.

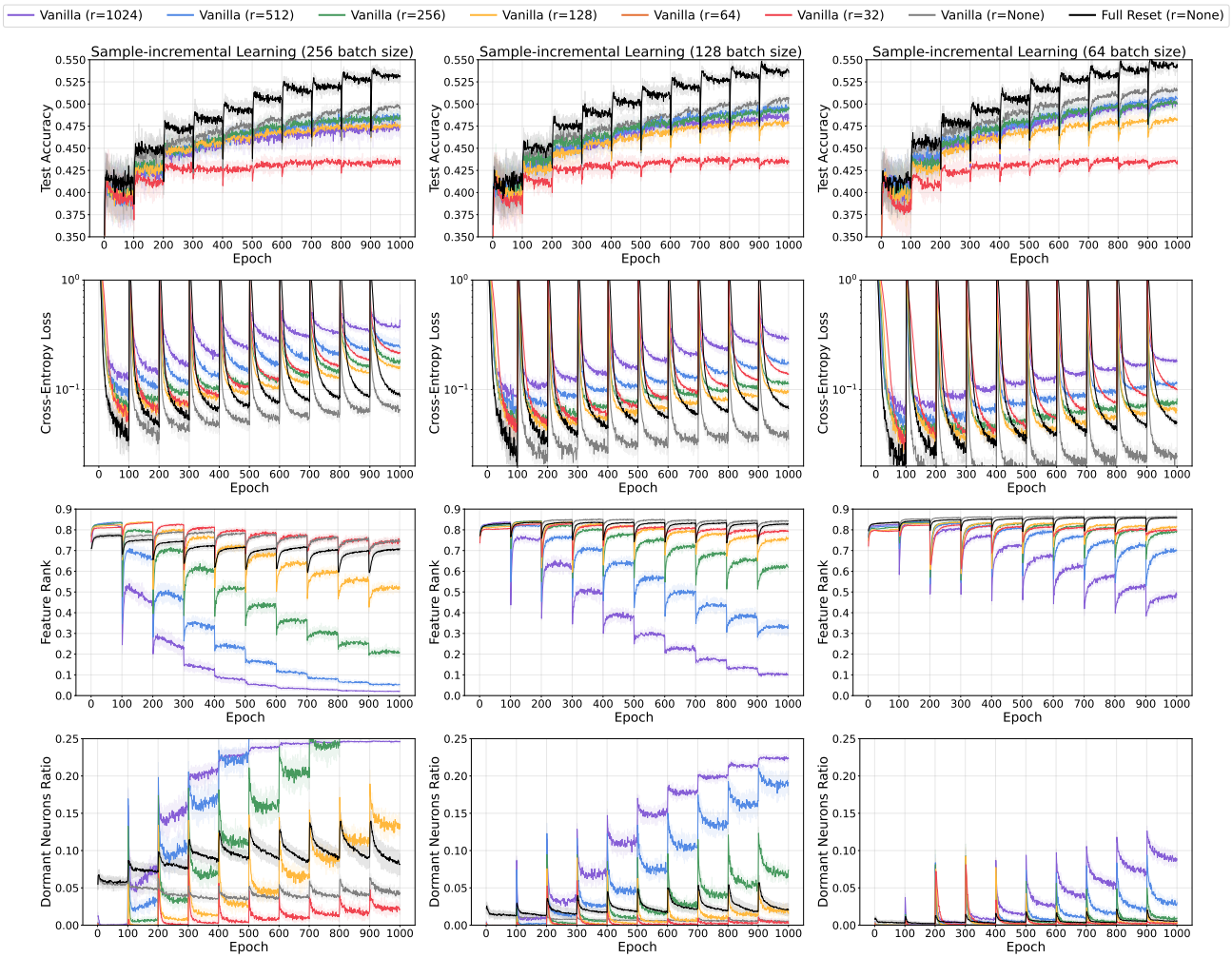

Figure 16. **Effect of batch size on sample-incremental learning.** Columns show batch sizes 256, 128, and 64, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Smaller batch sizes mitigate plasticity loss, yielding weaker later-stage degradation and improved plasticity-related diagnostics. Results are averaged over five random seeds.

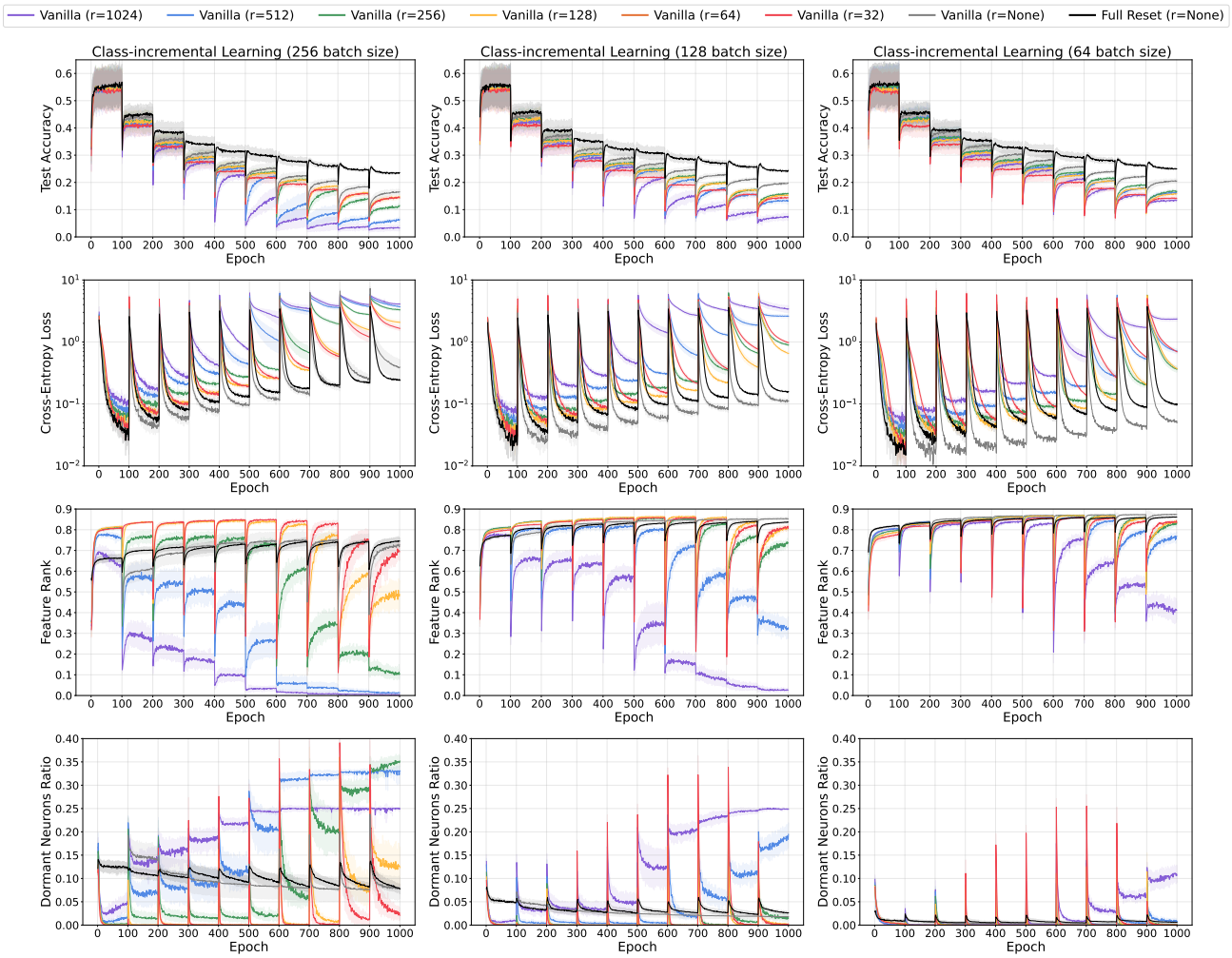

Figure 17. **Effect of batch size on class-incremental learning.** Columns show batch sizes 256, 128, and 64, and rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio. Smaller batch sizes mitigate plasticity loss, yielding weaker later-stage degradation and improved plasticity-related diagnostics. Results are averaged over five random seeds.

H. Additional Controls for Coupled Factor Optimization

The frozen-factor control in Section 3 suggests that the plasticity gap of low-rank factorized networks is not explained by the rank constraint alone. In this section, we provide additional experiments that further examine this point. Unless otherwise specified, all experiments in this section follow the experimental setup of the main experiments.

H.1. Full-Rank Frozen-Factor Control

We extend the frozen-factor control in Section 3 to $r = 512$, matching the hidden width of the MLP. For the hidden-to-hidden layers, this setting removes the bottleneck rank constraint while retaining the factorized parameterization. As before, we freeze either U_l or V_l at initialization and train only the other factor. This removes simultaneous co-adaptation between the two factors while keeping one side of the factor geometry fixed throughout training. Figure 18 summarizes the results across the three continual learning settings.

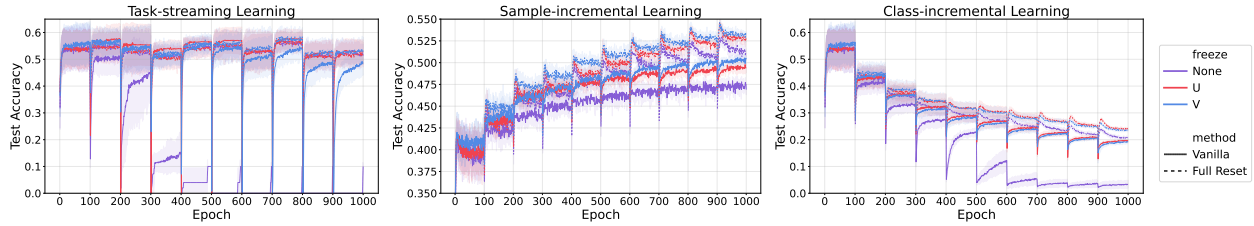

Figure 18. **Rank constraint alone does not explain the plasticity gap.** We conduct a freeze-one-factor control experiment with fixed rank $r = 512$, where either U or V is frozen at initialization and only the other factor is trained. **Left:** task-streaming CIFAR-100. **Middle:** sample-incremental CIFAR-10. **Right:** class-incremental CIFAR-100.

The results are consistent with the $r = 256$ experiment: the degradation of fully trainable low-rank factorized networks is not explained by the rank constraint alone. In task-streaming learning, freezing either factor substantially mitigates later-stage degradation relative to jointly training both factors. The freeze-V_l variant shows a mild drop in the final stages, but remains clearly separated from the fully trainable factorized model. Thus, even at $r = 512$, removing simultaneous factor co-adaptation preserves substantially more plasticity.

The sample-incremental setting shows a more factor-dependent pattern. Since a factorized layer computes $h = XU_lV_l^\top$, freezing U_l fixes the intermediate representation XU_l, leaving only the output-side mixing $V_l^\top$ trainable. This can be restrictive when newly added samples expand the support of the input distribution, although the restriction becomes weaker at higher rank because the fixed latent projection is larger. Nevertheless, the fully trainable factorized model still exhibits substantial degradation at $r = 512$, showing that increasing the bottleneck rank alone does not remove the sample-incremental plasticity loss.

In class-incremental learning, both frozen-factor variants again suffer much less degradation than the fully trainable factorized model, matching the qualitative behavior observed at $r = 256$. This further supports the view that severe plasticity loss can be avoided despite the rank constraint, as long as the two factors are not jointly co-adapted throughout training.

We also observe that, under full reset, frozen-factor variants sometimes generalize better than fully trainable factorized models in the sample- and class-incremental settings. We interpret this as a possible fixed-random-projection regularization effect rather than as evidence that frozen factors are uniformly superior. This interpretation is consistent with recent LoRA-style results showing that freezing one projection factor can retain competitive or improved generalization (Zhang et al., 2023; Zhu et al., 2024a).

Taken together, the $r = 512$ results reinforce the main conclusion of the frozen-factor control: increasing the bottleneck rank does not eliminate the qualitative gap between frozen-factor and jointly trained factorized models. Thus, plasticity loss in low-rank factorized networks depends not only on the rank constraint, but also on the coupled optimization dynamics induced by jointly training the two factors.

H.2. Overparameterized Fixed-Basis Control

The full-rank frozen-factor control above removes the bottleneck rank constraint for the hidden-to-hidden layers, but it still trains only one factor. Thus, one possible concern is that the improvement of frozen-factor controls may be partly due to the restricted form of the trainable parameterization rather than the removal of coupled factor optimization alone. To further separate these effects, we consider an overparameterized fixed-basis control.

For each factorized layer, we replace the standard two-factor parameterization

$$W_l = U_l V_l^\top$$

with a three-factor form

$$W_l = U_l A_l V_l^\top,$$

where U_l and V_l are frozen throughout training, and only the square middle matrix $A_l \in \mathbb{R}^{r\times r}$ is optimized. This construction fixes the input and output bases of the factorized layer, thereby removing simultaneous co-adaptation and scaling drift of the outer factors. At the same time, the trainable middle matrix A_l provides a richer set of degrees of freedom within the r-dimensional latent space than the one-sided frozen-factor control.

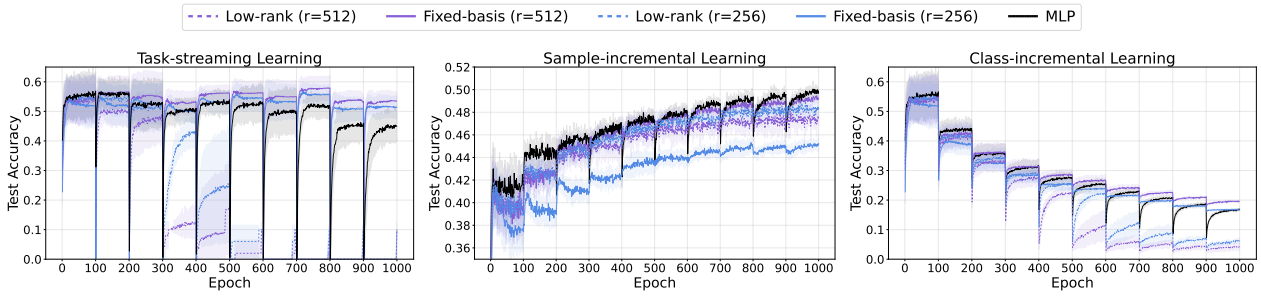

Figure 19. **Overparameterized fixed-basis control.**

Figure 19 reports representative high-rank cases, $r = 512$ and $r = 256$. In particular, the $r = 512$ fixed-basis control substantially mitigates the severe late-stage degradation of jointly trained factorized models, especially in task-streaming and class-incremental learning. Since the trainable component is a full middle matrix A_l, this improvement is unlikely to be explained merely by the restricted one-sided parameterization of the frozen-factor control. Together with the full-rank frozen-factor control in Figure 18, this suggests that the key stabilizing effect comes from preventing the outer factors from co-adapting and drifting in geometry over time.

At smaller ranks, the fixed-basis control can still be limited by reduced latent dimensionality and frozen random subspaces. Thus, this experiment does not rule out rank-related limitations, but it supports the conclusion that rank constraints alone do not explain the severe collapse of jointly trained factorized networks.

H.3. Fixed Projections in Vision Transformers

We next examine whether related fixed-projection effects appear in Transformer-style architectures, motivated by the fact that attention mechanisms contain multiplicative parameter matrices, such as $W_K^\top W_Q$ and PW_V, which can be viewed as factorized structures (Kobayashi et al., 2024). We conduct sample-incremental learning experiments on CIFAR-100 using Vision Transformers with either single-head self-attention or multi-head self-attention with 3 heads, following the setting of Lee et al. (2024). We compare *Vanilla*, *Full Reset*, and two fixed-projection controls, *freeze Q* and *freeze K*, where either the query projection or the key projection is frozen at initialization. These controls do not exactly match the low-rank factorization $UV^\top$ studied in the main text, but they serve a similar diagnostic role by restricting part of a multiplicative attention pathway and reducing simultaneous co-adaptation among attention projections.

Models are trained with dropout 0.1 and decoupled weight decay 0.05. For optimization, we use a linear learning-rate warmup schedule that increases the learning rate to 3×10^{-3} over the first 10 epochs and then keeps it fixed.

Figure 20 summarizes the results. The ViT experiments show that continual training exhibits later-stage degradation relative to the Full Reset reference, and that freezing attention projections changes the degradation pattern relative to fully trainable Vanilla training. This suggests that the relevance of fixed-projection controls is not limited to factorized MLP layers, but can also extend to Transformer-style architectures with multiplicative attention interactions. We interpret these results as an architectural robustness check rather than a complete analysis of Transformer plasticity.

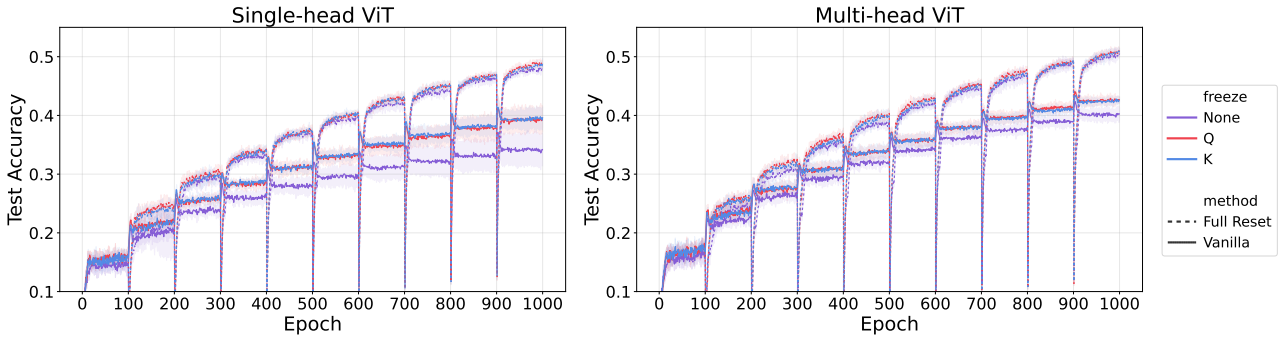

Figure 20. **Fixed-projection controls in Vision Transformers.** We conduct sample-incremental learning experiments on CIFAR-100 using Vision Transformers with single-head and multi-head self-attention. **Left:** single-head attention. **Right:** multi-head attention. We compare *Vanilla*, *Full Reset*, and fixed-projection controls where either the query projection or key projection is frozen at initialization. Freezing attention projections changes the degradation pattern relative to fully trainable Vanilla training, suggesting that projection co-adaptation can affect plasticity in Transformer-style architectures. Results are averaged over five random seeds.

H.4. Frozen-Factor Controls in LoRA-Style Continual Adaptation

Finally, we evaluate a LoRA-style adaptation setting across the three continual learning protocols used in the main paper: task-streaming, sample-incremental, and class-incremental learning. We first train the corresponding MLP backbone on the first stage with weight decay 1.0, which we found in full-fine-tuning trials to yield a strong and stable initial representation. For the remaining stages, this learned backbone is kept fixed and only the LoRA adapters are trained. This setup allows us to study the plasticity of low-rank adaptation modules on top of a fixed learned representation. For LoRA adapter training, we use the same learning rate as in the main experiments, 10^{-3}, for all variants; lower adapter learning rates led to substantially slower optimization and an unconverged *Full Reset* reference under the fixed 100-epoch stage budget.

We compare *Vanilla*, *Full Reset*, and a frozen-factor control. In the *Vanilla* setting, both LoRA factors are trained during the adaptation stages. In the *Full Reset* reference, the learned backbone is kept fixed, while the LoRA adapters are reinitialized at each stage and trained from scratch with the same per-stage budget. For the frozen-factor control, denoted by *freeze A*, the A factor is fixed while the other LoRA factor is trained. We do not freeze B because LoRA commonly initializes the output-side factor B to zero; freezing this zero-initialized factor would keep the adapter output at zero and prevent the LoRA module from learning.

Figures 21 and 22 summarize the results. Across task-streaming, sample-incremental, and class-incremental learning, LoRA-style continual adaptation also shows a plasticity gap between continual training and the *Full Reset* reference. Moreover, freezing one LoRA factor changes the degradation pattern relative to *Vanilla* training, consistent with the main observation that jointly adapting two low-rank factors can affect later-stage learnability. These results suggest that factorized adaptation modules can inherit similar plasticity vulnerabilities under non-stationary training, even when the base network is fixed after the initial stage.

Overall, these fixed-projection and frozen-factor experiments support the robustness of the main conclusion. Across higher-rank MLPs, Vision Transformers, and LoRA-style adaptation, restricting simultaneous co-adaptation changes the plasticity behavior relative to fully trainable factorized or multiplicative parameterizations. Thus, the plasticity gap studied in the main text is not only a consequence of limited representational rank, but is also tied to the optimization dynamics induced by jointly adapting coupled parameter components.

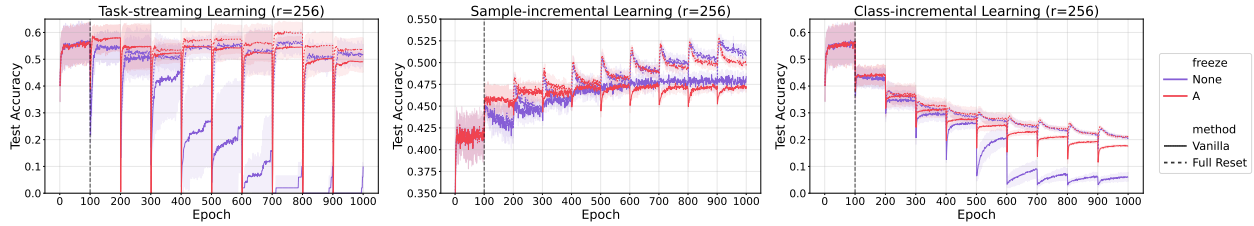

Figure 21. **Frozen-factor controls in LoRA-style continual adaptation at rank $r = 256$. Left:** task-streaming CIFAR-100. **Middle:** sample-incremental CIFAR-10. **Right:** class-incremental CIFAR-100. We compare *Vanilla*, *Full Reset*, and a frozen-factor control where the LoRA A factor is frozen while the other factor is trained. Freezing one LoRA factor changes the degradation pattern relative to fully trainable LoRA adapters, suggesting that joint adaptation of the two low-rank factors can affect plasticity under non-stationary training. Results are averaged over five random seeds.

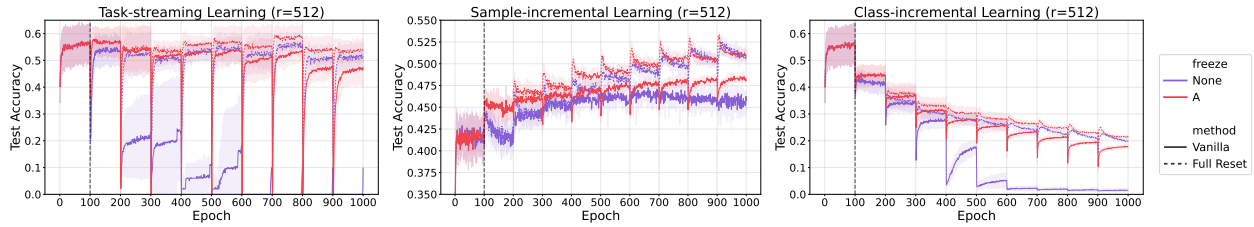

Figure 22. **Frozen-factor controls in LoRA-style continual adaptation at rank $r = 512$. Left:** task-streaming CIFAR-100. **Middle:** sample-incremental CIFAR-10. **Right:** class-incremental CIFAR-100. The fully trainable LoRA adapters show stronger later-stage degradation relative to the frozen-A control, making the effect of joint factor adaptation more visible than in the $r = 256$ setting. This pattern is consistent with the main low-rank MLP results, where increasing rank does not merely alleviate a capacity bottleneck but can amplify the plasticity gap through additional degrees of freedom for coupled factor dynamics. Results are averaged over five random seeds.

I. Balance Regularization Beyond Factorized MLPs

In this section, we extend the balance-regularization intervention beyond the factorized MLP layers studied in the main text. The goal is to test whether directly regularizing the geometry of the coupled projections or factors also improves later-stage adaptation in these settings.

I.1. Balance Regularization in Vision Transformers

We use the same Vision Transformer architecture as in Appendix H.3. Specifically, we evaluate multi-head self-attention with three heads across the three continual learning protocols used in the main paper. Detailed hyperparameters are provided in Table 2.

Balance regularization is added between the query and key projections,

$$\lambda \left\| W_Q^\top W_Q - W_K^\top W_K \right\|_F^2 .$$

Although this penalty is not identical to the factor-balance regularizer for an explicit low-rank layer $W = UV^\top$, it serves an analogous role in the attention module: the query and key projections jointly determine the multiplicative attention scores, and the penalty discourages their latent geometries from drifting apart during non-stationary training.

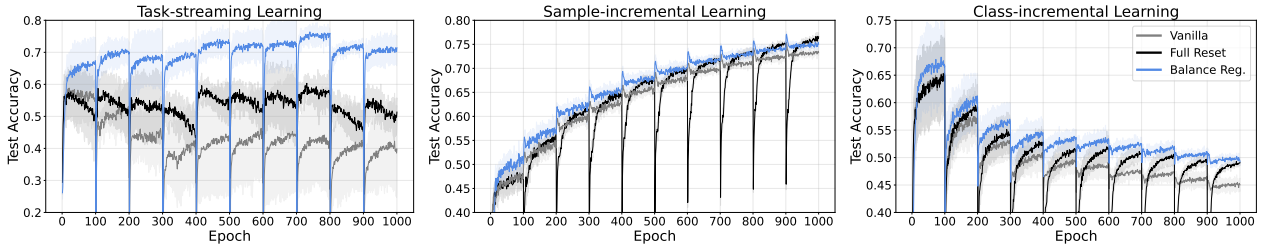

Figure 23. **Balance regularization in Vision Transformers.**

Figure 23 shows that balance regularization clearly improves performance over Vanilla continual training. Except for sample-incremental learning, the regularized model also reaches or surpasses the Full Reset reference. These results suggest that the benefit of balance regularization is not limited to explicitly factorized MLP layers, but can also appear in Transformer-style architectures with coupled multiplicative projections.

Table 2. Hyperparameters for Vision Transformers. All experiments use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, batch size 256, dropout 0.1, decoupled weight decay 0.05, linear learning-rate warmup, and gradient clipping with maximum norm 0.5.

Experiment	Learning rate	Balance coefficient λ
Task-streaming Learning	3×10^{-3}	1e+0
Sample-incremental Learning	2×10^{-3}	1e+0
Class-incremental Learning	2×10^{-3}	1e+1

I.2. Balance Regularization in LoRA-Style Continual Adaptation

We next evaluate whether factor balance remains useful in a LoRA-style continual adaptation setting. After the first stage, the backbone weights W_0 are frozen, and each adapted hidden layer is written as $W = W_0 + \Delta W$, with $\Delta W = BA$, where only the LoRA factors are trained in subsequent stages. We apply LoRA only to the hidden layers and leave the classifier head unfactorized to avoid artificial scaling effects from the small output dimension of the classifier.

In addition to standard balance regularization, we include the LoRA-specific imbalance regularizer of Zhu et al. (2024b). This regularizer was motivated by forward variance propagation and encourages a controlled imbalance by scaling the $B^\top B$ term by r/m. Hyperparameters are reported in Table 3.

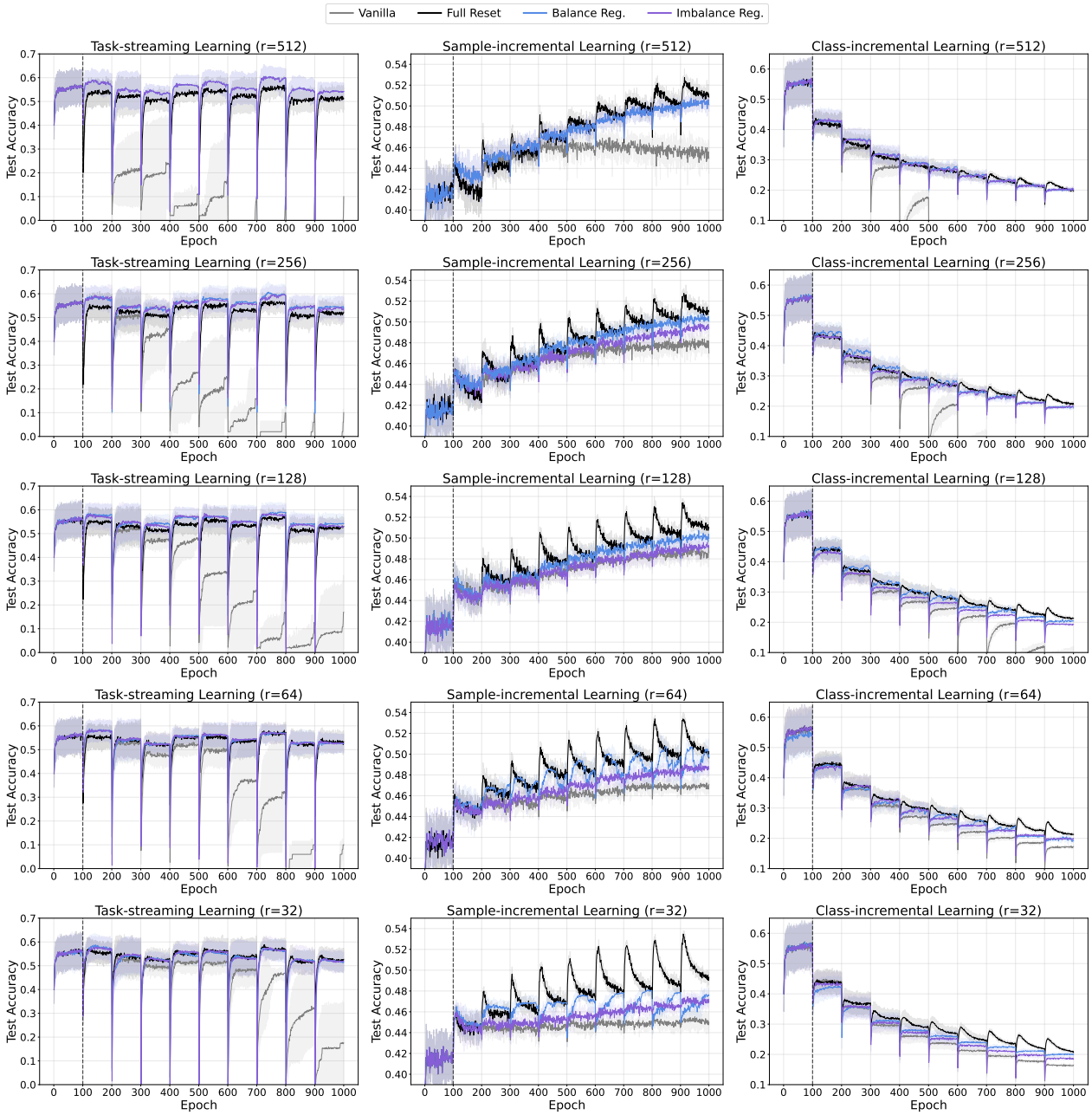

Figure 24. **Balance regularization in LoRA-style continual adaptation.**

Figure 24 shows that Balance Reg. generally improves over vanilla LoRA, indicating that controlling factor geometry

remains useful even when only residual adapter factors are trained. To interpret the comparison, recall that standard Balance Reg. penalizes

$$R_{\text{bal}}(A, B) = \lambda \left\| AA^\top - B^\top B \right\|_F^2 ,$$

whereas the LoRA-specific imbalance regularizer uses

$$R_{\text{imb}}(A, B) = \lambda \left\| AA^\top - \frac{r}{m} B^\top B \right\|_F^2 ,$$

where r is the LoRA rank and m is the output dimension of the adapted layer. When $r = m = 512$, the two regularizers are mathematically identical because $r/m = 1$, and indeed yield the same performance. For lower ranks, however, Balance Reg. generally outperforms Imbalance Reg. Since $r < m$, the target of R_{imb} implies

$$AA^\top \approx \frac{r}{m} B^\top B \qquad \Longleftrightarrow \qquad B^\top B \approx \frac{m}{r} AA^\top .$$

Thus, as the rank decreases, the imbalance regularizer imposes increasingly strong output-side scaling. From the perspective of forward variance propagation, such controlled imbalance can be a reasonable geometry for LoRA factors. However, in continual adaptation, especially at small ranks, the adapter must reuse a limited set of latent directions across changing data distributions. Standard Balance Reg. instead keeps the two LoRA factors on comparable Gram scales, providing a more stable factor geometry for continual adaptation.

Overall, these results support our main conclusion that strict factor balance is a useful regularization target for continual low-rank adaptation. The comparison with the forward-variance-motivated imbalance regularizer of Zhu et al. (2024b) suggests that the most appropriate notion of factor geometry depends on the learning regime: controlled imbalance may be suitable for variance-preserving LoRA fine-tuning, whereas continual adaptation benefits more consistently from strict Gram balance. However, Balance Reg. still falls short of Full Reset at smaller ranks, indicating that factor-geometry control alone cannot fully compensate for the limited adaptation capacity of low-rank LoRA adapters.

Table 3. Hyperparameter search space and selected coefficients for LoRA-style continual adaptation.

Experiment	**Method**	**Search Space**	**r=512**	**r=256**	**r=128**	**r=64**	**r=32**
Task-streaming Learning	Balance Reg.	1e+0, 1e+1, 1e+2, 1e+3	1e+2	1e+1	1e+1	1e+1	1e+1
	Imbalance Reg.	1e+0, 1e+1, 1e+2, 1e+3	1e+3	1e+2	1e+2	1e+1	1e+1
Sample-incremental Learning	Balance Reg.	1e−1, 1e+0, 1e+1, 1e+2	1e+1	1e+1	1e+0	1e+0	1e+0
	Imbalance Reg.	1e−1, 1e+0, 1e+1, 1e+2	1e+1	1e+0	1e−1	1e+0	1e+0
Class-incremental Learning	Balance Reg.	1e+0, 1e+1, 1e+2, 1e+3	1e+3	1e+1	1e+1	1e+1	1e+2
	Imbalance Reg.	1e+0, 1e+1, 1e+2, 1e+3	1e+0	1e+2	1e+3	1e+2	1e+2

J. Log-scale visualization of imbalance reduction

We provide log-scale visualizations of the factor imbalance trajectories to show the low-magnitude dynamics that are difficult to distinguish on a linear scale. Since balance regularization reduces the Gram mismatch by a large margin, the remaining imbalance dynamics are more clearly visible on a logarithmic y-axis.

Figure 25 compares vanilla and balance-regularized imbalance trajectories across the three continual learning settings. The log-scale view shows that balance regularization keeps the imbalance near a uniformly low level throughout training. These plots complement the main-text results by showing that the accuracy improvements in Figure 3 are accompanied by a strong reduction in factor imbalance.

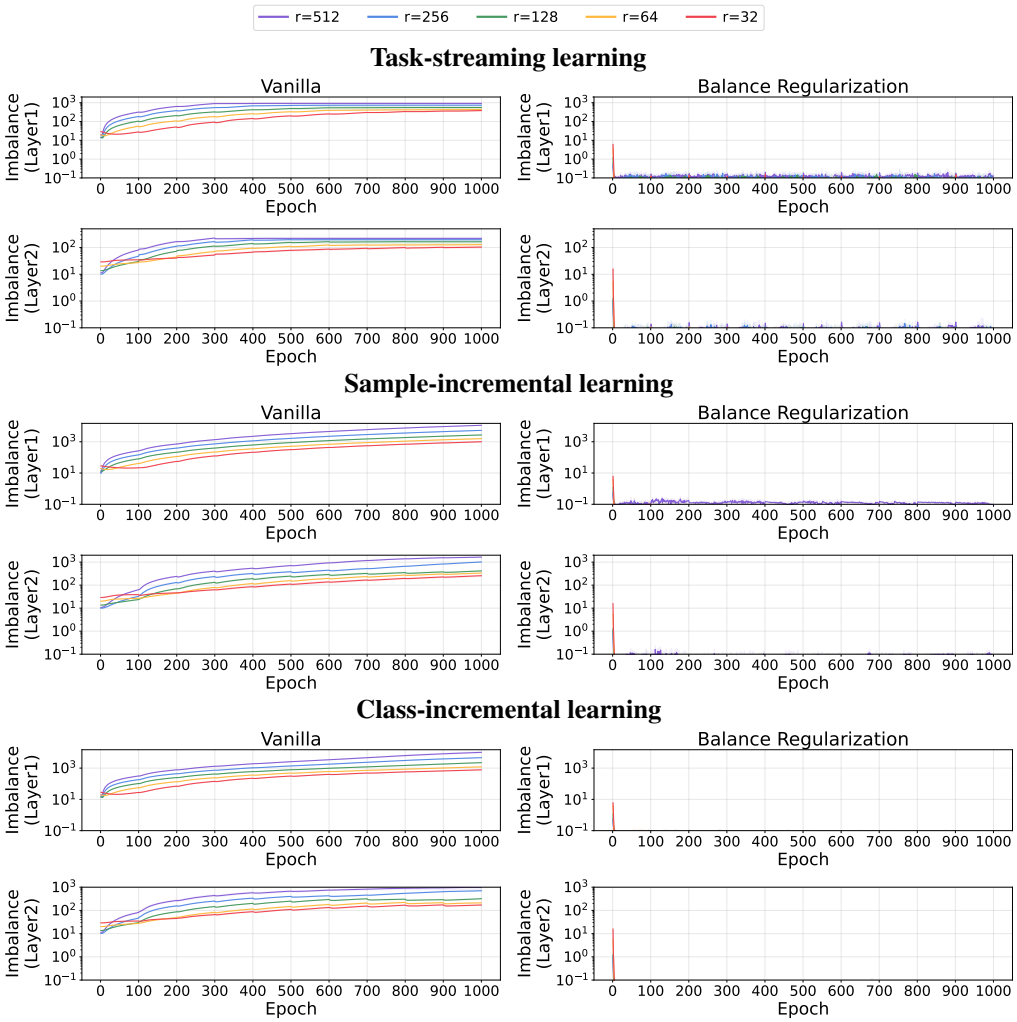

Figure 25. **Log-scale visualization of factor imbalance across continual learning settings.** Rows show task-streaming, sample-incremental, and class-incremental learning. In each row, the left column shows vanilla training and the right column shows balance regularization, with the two plotted rows corresponding to the two factorized layers. Balance regularization keeps the Gram mismatch near a uniformly low level throughout training. Results are averaged over five random seeds.

K. Relating Accumulated Factor Imbalance to Degradation Metrics

We further analyze the relationship between accumulated factor imbalance and several degradation metrics across 16 rank configurations, $r \in \{32, 64, \ldots, 512\}$. Rather than relying only on the final checkpoint of each stage, we summarize each metric by the area under its within-stage trajectory. This provides a more stable measure of adaptation, since all models are trained for a fixed number of epochs per stage and the final checkpoint can be affected by overfitting, saturation, or late-stage fluctuations.

For each rank, we compare vanilla continual training with the corresponding *Full Reset* counterpart at the same rank. We define the AUC-based degradation gap for metric q as

$$\mathrm{Gap}_q(r) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} s_q \left(\mathrm{AUC}^q_{\textit{Full Reset},r,t} - \mathrm{AUC}^q_{\textit{Vanilla},r,t} \right),$$

where $s_q = 1$ for metrics where larger values are better, namely test accuracy and feature rank, and $s_q = -1$ for metrics where smaller values are better, namely cross-entropy loss and dormant neuron ratio. With this convention, larger values of $\mathrm{Gap}_q(r)$ consistently indicate stronger degradation of vanilla continual training relative to its *Full Reset* counterpart. We compute accumulated factor imbalance by averaging $\|U^\top U - V^\top V\|_F$ over epochs, stages, and factorized layers, and then compare it with these degradation gaps across ranks.

Figure 26 and Table 4 summarize how accumulated factor imbalance co-varies with AUC-based degradation across ranks. In task-streaming learning, imbalance strongly tracks the test accuracy and cross-entropy loss gaps ($\rho = 0.929$ for both), indicating that high-rank degradation appears in both predictive performance and the training objective, although the feature rank and dormant neuron ratio gaps show weaker or opposite trends. Sample-incremental learning is more mixed: the test accuracy gap is weakly related to imbalance, while the cross-entropy loss and feature rank gaps increase with imbalance, suggesting that optimization- and representation-level degradation can emerge even when test accuracy is less sensitive. Class-incremental learning shows the most consistent pattern, with strong associations for test accuracy ($\rho = 0.974$), cross-entropy loss ($\rho = 0.956$), feature rank ($\rho = 0.991$), and dormant neuron ratio ($\rho = 0.750$). Overall, accumulated imbalance is most consistently associated with degradation under stronger stage shifts, while its relationship to individual diagnostics remains protocol-dependent.

These correlations are rank-wise associations across the 16 rank configurations and should not be interpreted as causal evidence by themselves, since rank and accumulated imbalance are also strongly coupled. Rather, together with the frozen-factor controls and balance-regularization experiments, these results support the view that factor imbalance is an important factor-level mechanism associated with plasticity loss in jointly trained low-rank factorized networks, while the strength of this association depends on the metric and learning protocol.

Table 4. **Rank-wise correlations between accumulated factor imbalance and AUC-based degradation gaps.** Spearman correlations are computed across 16 rank-level means, $r \in \{32, 64, \ldots, 512\}$. Imbalance is strongly associated with test accuracy and training loss gaps in task-streaming and class-incremental learning, while sample-incremental learning shows weaker accuracy association but positive associations for loss and feature rank gaps.

Experiment	Metric	Spearman ρ	p-value
Task-streaming Learning	Test Accuracy Gap	0.929	1.94×10^{-7}
	Cross-Entropy Loss Gap	0.929	1.94×10^{-7}
	Feature Rank Gap	0.032	0.905
	Dormant Neuron Ratio Gap	-0.709	0.002
Sample-incremental Learning	Test Accuracy Gap	-0.135	0.617
	Cross-Entropy Loss Gap	0.838	5.01×10^{-5}
	Feature Rank Gap	0.541	0.030
	Dormant Neuron Ratio Gap	0.321	0.226
Class-incremental Learning	Test Accuracy Gap	0.974	2.28×10^{-10}
	Cross-Entropy Loss Gap	0.956	7.76×10^{-9}
	Feature Rank Gap	0.991	1.09×10^{-13}
	Dormant Neuron Ratio Gap	0.750	8.20×10^{-4}

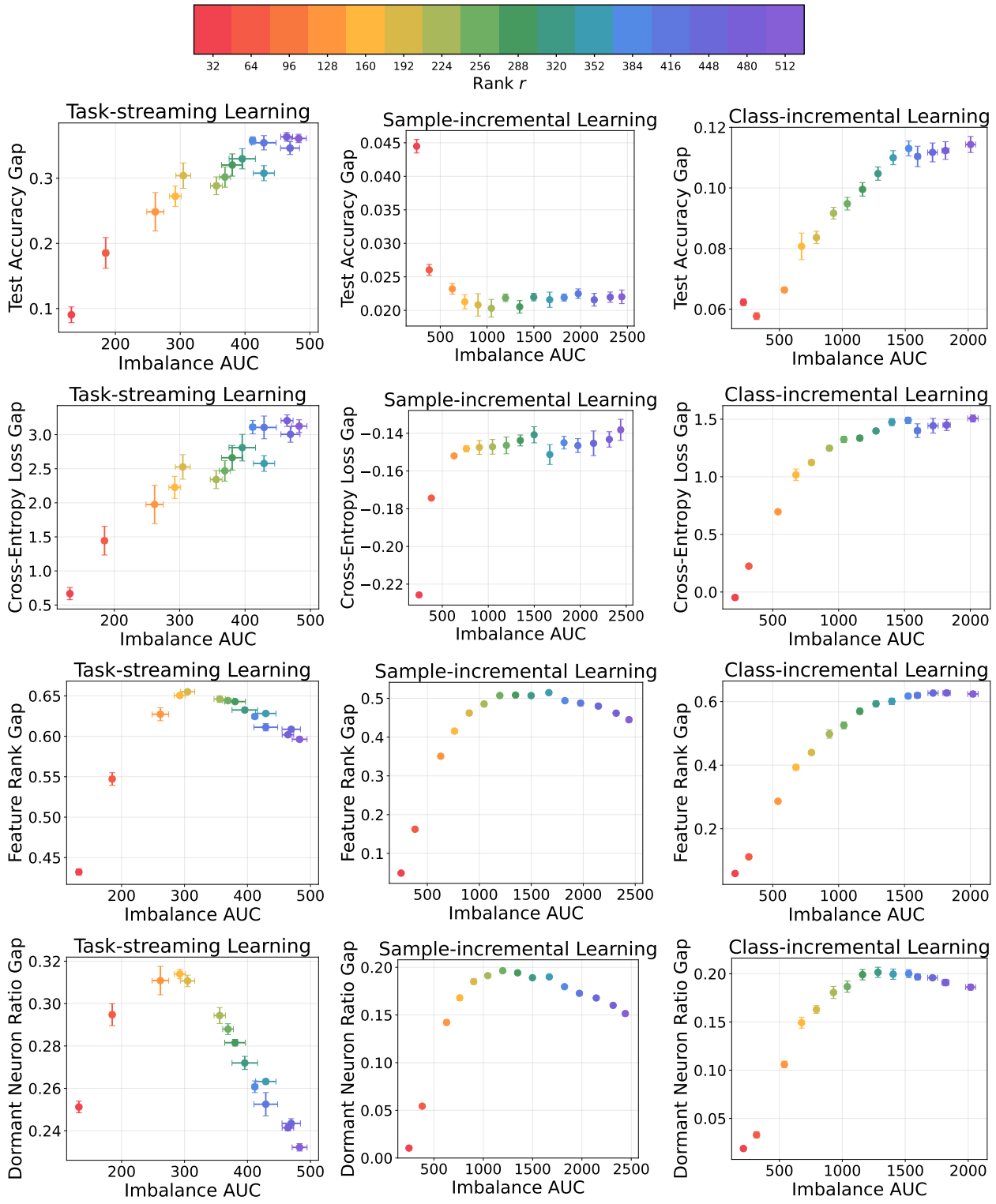

Figure 26. **Accumulated factor imbalance versus AUC-based degradation gaps across ranks.** Rows show test accuracy, cross-entropy loss, feature rank, and dormant neuron ratio gaps; columns show task-streaming, sample-incremental, and class-incremental learning. Each point denotes one rank, averaged over seeds, with error bars indicating standard errors.

L. Diagnostic Statistics

In this section, we report additional diagnostic statistics to examine whether the improvements from balance regularization can be explained by standard indicators of plasticity loss. We track cross-entropy loss, feature rank, dormant neuron ratio, effective weight rank, spectral norm, and condition number. Dormant neuron ratio is measured following Sokar et al. (2023), while feature rank and effective weight rank are computed following Kumar et al. (2021); the former is computed from learned representations, and the latter from the composed matrix $W = UV^\top$. Spectral norm and condition number are computed separately for the individual factors U and V. We summarize these diagnostics for vanilla low-rank training and balance regularization in Figures 27 and 28, respectively.

Figure 28 shows that balance regularization generally improves feature rank and reduces dormant neurons relative to the vanilla low-rank baseline in Figure 27, although these quantities are not perfectly controlled across all settings. We therefore treat them as auxiliary diagnostics rather than oracle measures of plasticity. The effective weight-rank curves also suggest that effective rank alone does not provide a sufficient explanation of the recovery. Likewise, we do not observe comparably large or systematic changes in the spectral norm or condition number of the individual factors that would explain the recovery purely through factor scale changes or generic improvements in individual-factor conditioning.

Overall, the patterns in Figures 27–28 suggest that the observed improvements are not fully explained by any single diagnostic statistic measured here. This motivates analyzing the interaction between the two factors directly, rather than relying only on diagnostics computed from the representations, the composed matrix, or each factor in isolation.

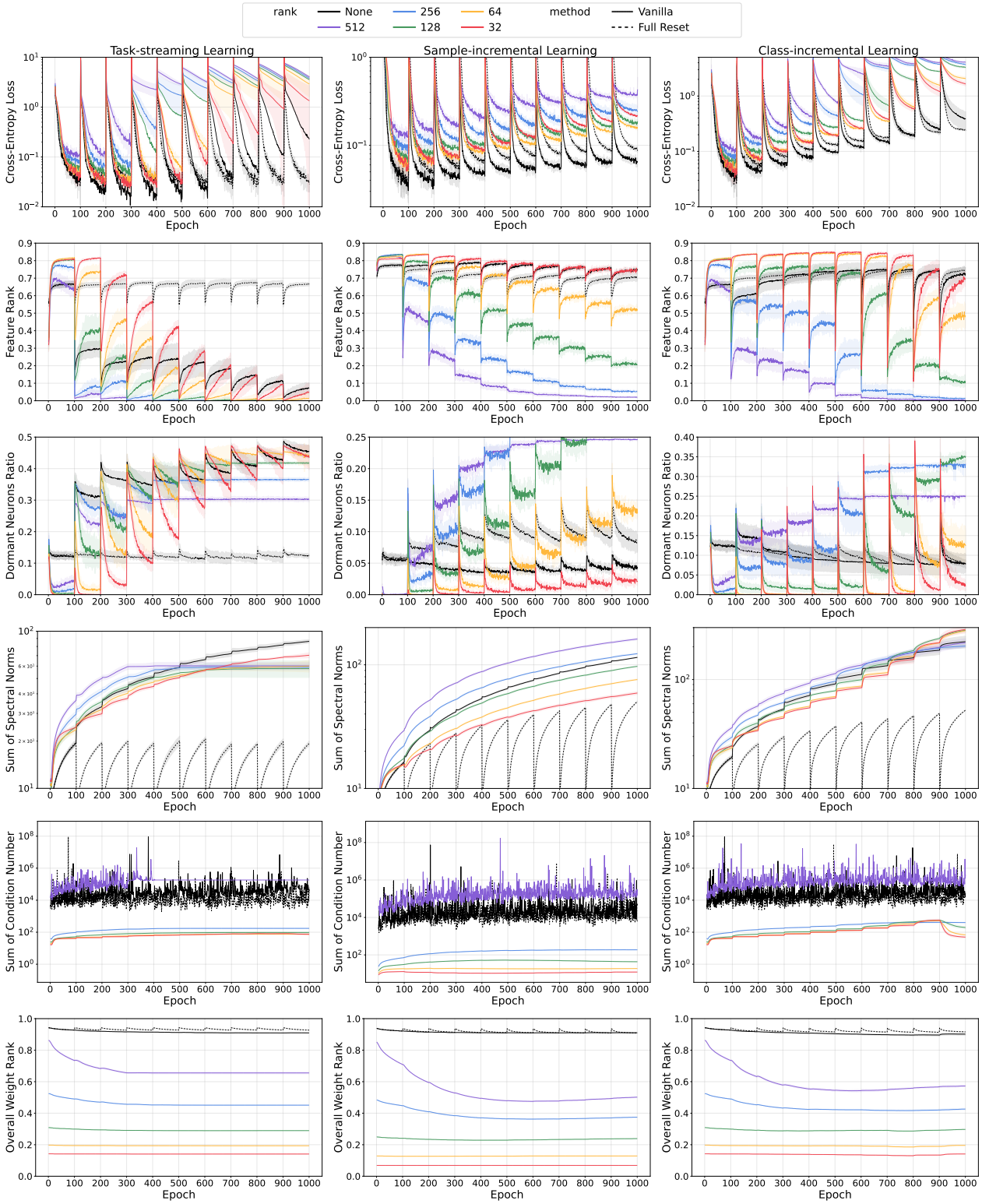

Figure 27. **Diagnostic Statistics of Vanilla across continual learning settings.**

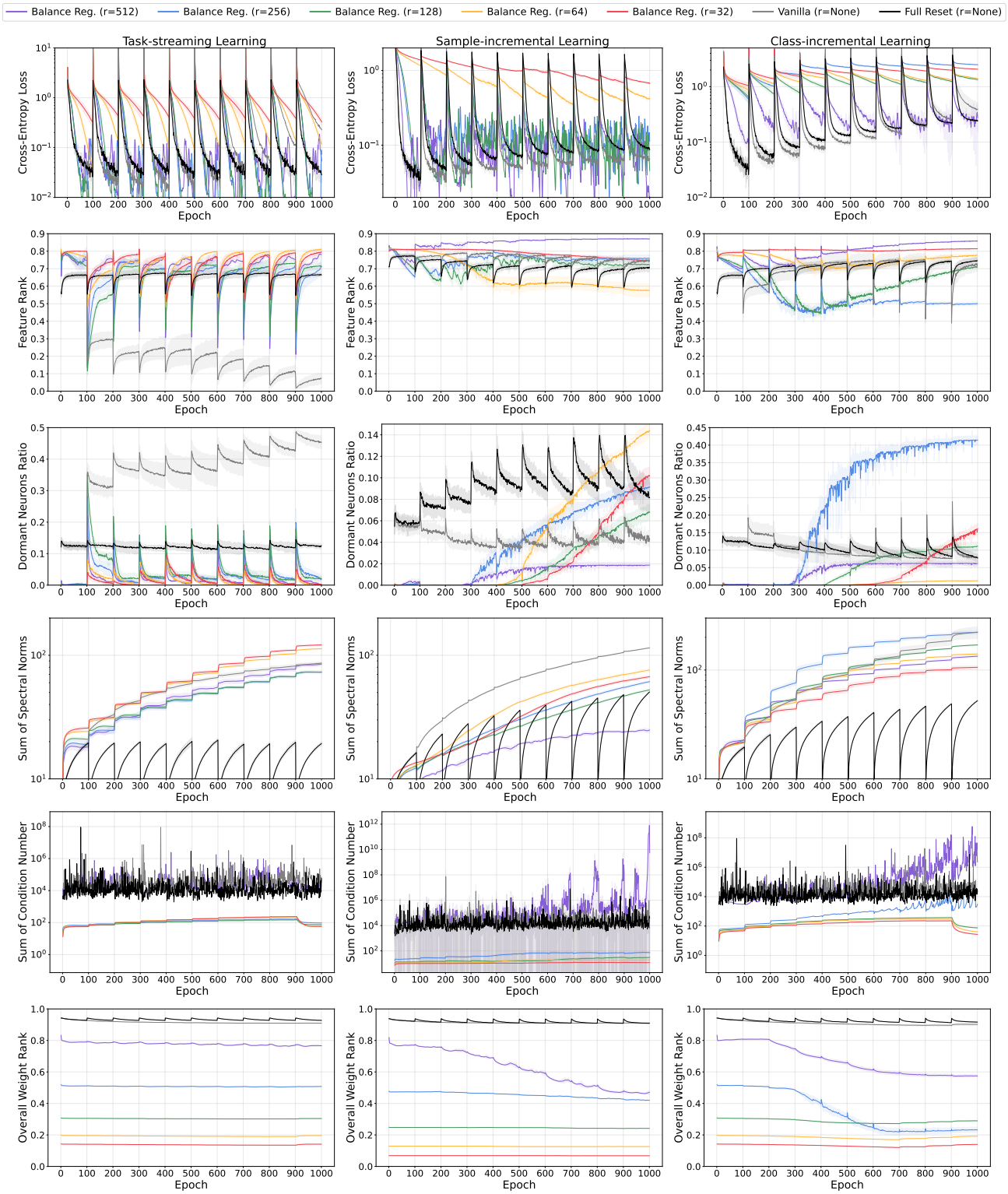

Figure 28. **Diagnostic Statistics of balance regularization across continual learning settings.**

M. Domain-streaming Learning

To further evaluate plasticity under input-domain distribution shifts, we construct a corruption-based domain-streaming protocol on CIFAR-10. Following the common corruption benchmark (Hendrycks & Dietterich, 2019), we use image corruptions that change the input distribution while preserving the semantic label space. The CIFAR-10 training set is randomly partitioned into 10 disjoint sample chunks, and each stage applies a distinct corruption type to the current chunk. We use a fixed severity level of 5 for all corruptions, and apply the following stage-wise corruption sequence: `snow`, `frost`, `fog`, `brightness`, `contrast`, `spatter`, `elastic_transform`, `jpeg_compression`, `pixelate`, and `saturate`. The model is trained only on the corrupted data from the current stage, without access to previous chunks, and evaluated on the CIFAR-10 test set corrupted with the same stage-specific corruption. This protocol isolates plasticity under sequential input-domain shifts while keeping the label space fixed.

Figure 29 shows that corruption-based domain streaming also induces plasticity loss, even though the label space remains fixed throughout training. *Vanilla* falls below the corresponding *Full Reset* reference, and this gap is most pronounced for high-rank factorized models, particularly at $r = 512$. Since the *Full Reset* models remain competitive, the degradation is difficult to explain by rank capacity alone.

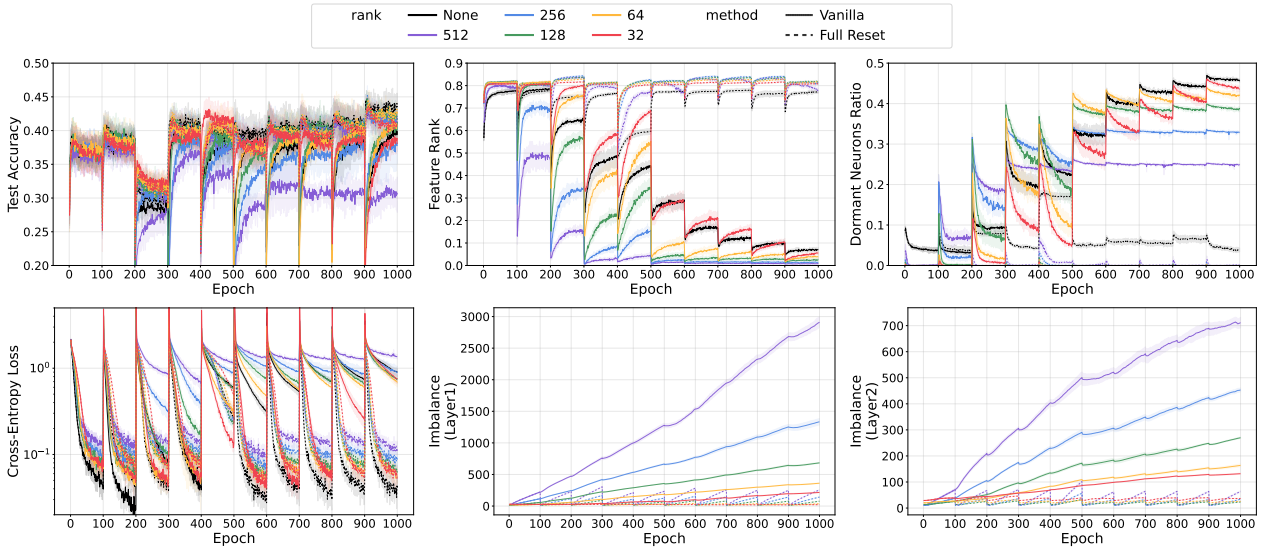

Figure 29. **Low-rank factorization exacerbates plasticity loss in domain-streaming learning.** We compare full-matrix MLPs with low-rank factorized MLPs under fixed depth and width, varying only the bottleneck rank r. Low-rank factorized models show stronger degradation than their full-matrix counterparts.

Figure 30 further shows that balance regularization substantially mitigates this degradation and improves the associated plasticity diagnostics, including feature rank and dormant neuron ratio. These results suggest that the plasticity gap of jointly trained low-rank factors is not restricted to class-split continual learning, but also appears under sequential input-domain shifts induced by common corruptions.

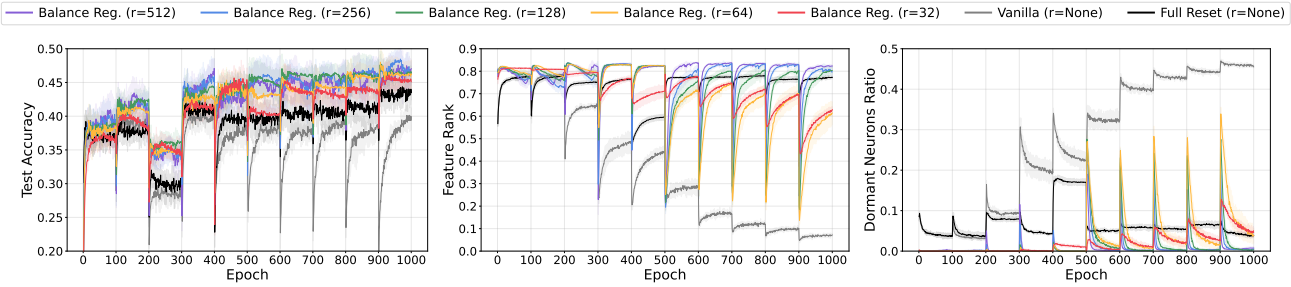

Figure 30. **Balance regularization mitigates plasticity loss in domain-streaming learning.** We train low-rank factorized MLPs with an explicit balance regularizer, which penalizes mismatch between the two factor Gram matrices.