Worker Disagreement Reveals Sharp Directions in Local SGD

Tolga Dimlioglu TD2249@NYU.EDU
Kristi Topollai KT2664@NYU.EDU
Anna Choromanska AC5455@NYU.EDU
New York University, New York, USA

Abstract

Deep neural network training often exhibits highly anisotropic loss geometry, where a few sharp dominant Hessian directions coexist with a large flatter bulk. Gradients tend to align disproportionately with these dominant directions, although stable progress often requires movement through flatter bulk directions [13]. Estimating the dominant subspace is therefore useful but costly with direct Hessian-based methods. We show that standard Local SGD exposes this geometry through worker disagreement. We theoretically show that the worker–average gap covariance is shaped by stochastic-gradient noise and Hessian curvature, causing workers to disagree along sharp, curvature-sensitive directions. Thus, worker–average gaps provide a cheap Hessian-free estimator of the dominant subspace. Experiments on MLPs, CNNs, and Transformers show that subspaces formed by worker–average gaps capture a substantial fraction of the gradient component lying in the dominant Hessian eigenspace.

1. Introduction

Deep neural networks are optimized in extremely high-dimensional parameter spaces, yet their training dynamics often exhibit low-dimensional structure [7, 17]. In particular, the training-loss Hessian is highly anisotropic: a few large eigenvalues define a sharp low-dimensional *dominant* subspace, while the remaining directions form a flatter high-dimensional *bulk* [2, 9–11]. Gradients along centralized SGD trajectories often concentrate in the dominant eigenspace [1, 3], but recent evidence shows that useful loss descent can depend substantially on movement through flatter bulk directions [13]. Thus, identifying the dominant subspace is important for diagnosing and controlling optimization, but direct Hessian-based estimation is computationally expensive.

We investigate whether distributed training dynamics can reveal the dominant subspace without explicitly estimating the Hessian. In Local SGD, workers independently take multiple stochastic-gradient steps before averaging [8, 14], naturally *disagreeing* on the next iterate. We quantify this disagreement via worker–average gaps. Prior analyses typically treat these gaps as consensus error to be bounded [4, 14]. We instead ask whether they reveal useful landscape directions.

We answer this question theoretically and empirically. We show that worker–average gap covariance is shaped by the interaction between stochastic-gradient noise and Hessian curvature, amplifying disagreement along directions that are both noisy and curvature-sensitive. Since SGD noise is highly anisotropic and curvature-aligned [15, 16, 18, 20], worker–average gaps can serve as a Hessian-free proxy for the dominant eigenspace. Experiments on MLPs, CNNs, and Transformers show that worker–average gap subspaces capture, and can be used to suppress, a substantial fraction of the gradient's dominant Hessian component. These results recast worker disagreement from a consensus error into a low-cost source of geometric tool for controlling optimization.

© T. Dimlioglu, K. Topollai & A. Choromanska.

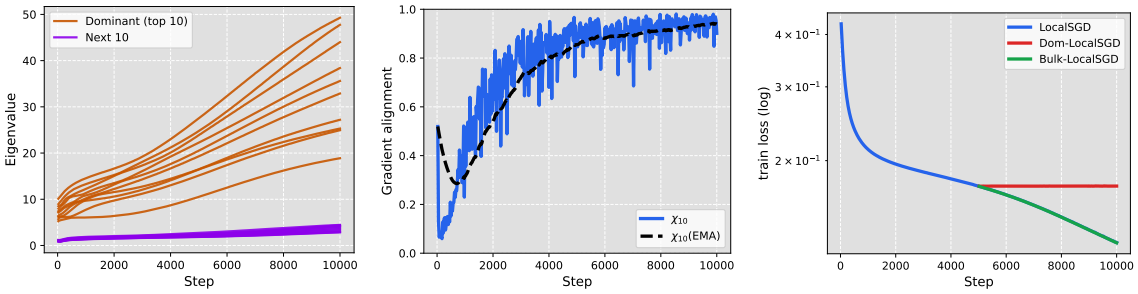


Figure 1: FC tanh network trained on MNIST-5k using Local SGD with $M = 4$ workers and $\tau = 5$. **Left:** the Hessian spectrum develops a clear separation between the top 10 eigenvalues and the bulk. **Middle:** the gradient direction increasingly aligns with the dominant Hessian subspace, measured by χ_{10}. **Right:** retaining only the dominant component stalls optimization, indicating that useful descent is largely carried by bulk directions.

2. Preliminaries and Initial Observations

Setup. We consider distributed empirical risk minimization with M workers minimizing $f(\theta) = \frac{1}{N}\sum_{n=1}^{N} \ell(h_\theta(x_n), y_n)$, where $h_\theta : \mathcal{X} \to \mathcal{Y}$ is a neural network with parameters $\theta \in \mathbb{R}^D$ and ℓ is the per-sample loss. The training set is partitioned IID across M workers. We focus on Local SGD training. At the beginning of each communication round, all workers start from a common parameter vector; each worker then performs τ local stochastic-gradient steps on its own minibatches; and the workers synchronize by averaging their parameters. Pseudo-code is provided in Appendix 1.

Notations. Here, we follow the dominant–bulk subspace setup of [13]. Let $\nabla f(\theta) \in \mathbb{R}^D$ denote the full-batch gradient and $H(\theta) = \nabla^2 f(\theta) \in \mathbb{R}^{D\times D}$ denote the loss Hessian. Let $\lambda_1(\theta) \geq \cdots \geq \lambda_D(\theta)$ be the eigenvalues of $H(\theta)$, with corresponding eigenvectors $u_1(\theta), \ldots, u_D(\theta)$. For a chosen dimension C, we define the *Dominant* subspace as $\mathcal{S}_C(\theta) = \mathrm{span}\{u_1(\theta), \ldots, u_C(\theta)\}$, and refer to its orthogonal complement $\mathcal{S}_C^{\perp}(\theta)$ as the *Bulk* subspace. Following [3, 13], we set C to the number of classes in the classification task. We denote the orthogonal projector onto $\mathcal{S}_C(\theta)$ by $P_C(\theta) = \sum_{k=1}^{C} u_k(\theta)u_k(\theta)^\top$, and define $P_C^{\perp}(\theta) = I - P_C(\theta)$.

For any vector $v \in \mathbb{R}^D$, we measure its relative alignment with the dominant subspace using

$$\chi_C(v;\theta) := \|P_C(\theta)v\|_2/\|v\|_2 \tag{1}$$

This metric is inherited from Song et al. [13]. Notice that, when $\chi_C(v;\theta)$ is close to one, we say v is aligned with $\mathcal{S}_C(\theta)$ as most of the norm of v lies in the dominant subspace; when it is close to zero, v lies mostly in the bulk subspace. Next, we verify that the dominant-subspace alignment observed in centralized SGD also appears along Local SGD trajectories.

Observation 1: Local SGD gradients align with the dominant subspace. At each synchronization round, after worker averaging, we compute the full-batch gradient $\nabla f(\bar{\theta})$ and Hessian $H(\bar{\theta})$ at the averaged parameter $\bar{\theta}$, form the dominant subspace $\mathcal{S}_C(\bar{\theta})$, and report $\chi_C(\nabla f(\bar{\theta}); \bar{\theta})$. Figure 1 shows the result for a fully-connected (FC) tanh network trained on MNIST [6] with $M = 4$ workers and communication period $\tau = 5$. The Hessian develops a clearly separated dominant subspace, and the gradient increasingly aligns with this subspace throughout training as shown in left and middle panels of Figure 1 respectively. We observe the same qualitative behavior for a CNN on CIFAR10 [5] and a Transformer on SST2 [12]; but these results are deferred to Appendix A.2 due to space constraints. Then, we test whether this dominant-space alignment implies that Local SGD can be continued using only the dominant component of its communication-round update.

Observation 2: Local SGD progress is not carried by the dominant component. Starting from a standard Local SGD checkpoint, we compare standard Local SGD with two projected variants. Let $\bar{p}^c := \frac{1}{M}\sum_{i=1}^{M}\theta_i^{c,\tau} - \bar{\theta}^c$ denote the average outer step at communication round c. Dom-Local SGD projects $\bar{p}^c$ onto the dominant subspace $\mathcal{S}_C(\bar{\theta}^c)$, whereas Bulk-Local SGD projects it onto the complementary bulk subspace $\mathcal{S}_C^{\perp}(\bar{\theta}^c)$. As shown in Figure 1 right panel, Dom-Local SGD stalls after the projection is introduced, whereas Bulk-Local SGD closely tracks standard Local SGD. Thus, although gradients are strongly aligned with the dominant Hessian subspace, the update component that sustains training lies primarily in the bulk. More details and results on CIFAR-10 and SST-2 are given in Appendix A.2. Next, we look beyond the average outer step of Local SGD and analyze the disagreement among local worker parameters, i.e., the gap vector between each worker parameter and their average, which Local SGD produces naturally during training.

3. Theoretical Characterization of Worker–Average Gaps

We analyze how worker disagreements, i.e., the worker–average gaps, arise within communication round c. At the beginning of the round, all workers are synchronized at $\bar{\theta}^c$, so that $\theta_i^{c,0} = \bar{\theta}^c$ for $i = 1, \dots, M$. Each worker then performs τ local SGD steps with step size η:

$$\theta_i^{c,t} = \theta_i^{c,t-1} - \eta g_i^{c,t}, \qquad g_i^{c,t} = \nabla f(\theta_i^{c,t-1}) + \epsilon_i^{c,t}, \qquad t = 1, \dots, \tau, \tag{2}$$

where $\epsilon_i^{c,t}$ is the stochastic-gradient noise satisfying $\mathbb{E}[\epsilon_i^{c,t} \mid \theta_i^{c,t-1}] = 0$. Now, define the within-round average and the worker–average gap i from this average as $\bar{\theta}^{c,t} := \frac{1}{M}\sum_{j=1}^{M}\theta_j^{c,t}$ and $d_i^{c,t} := \theta_i^{c,t} - \bar{\theta}^{c,t}$ respectively. Notice that, by construction, $\sum_i d_i^{c,t} = 0$, and since workers are synchronized at the start of the round, $d_i^{c,0} = 0$. After the final local step, the next synchronized model is $\bar{\theta}^{c+1} := \bar{\theta}^{c,\tau}$ and the worker–average gap is $\Delta_i^{c+1} := \theta_i^{c,\tau} - \bar{\theta}^{c+1} = d_i^{c,\tau}$. The following lemma describes the within-round evolution of the worker-average gaps.

Lemma 1 (Linearized worker–average gap dynamics) *Assume that f is twice differentiable. For each local step $t \in \{1, 2, \dots, \tau\}$ in communication round c, define the local Hessian around the within-round average by $H_{c,t} := \nabla^2 f(\bar{\theta}^{c,t-1})$. Let $\bar{\epsilon}^{c,t} := \frac{1}{M}\sum_{j=1}^{M}\epsilon_j^{c,t}$ and $\zeta_i^{c,t} := \epsilon_i^{c,t} - \bar{\epsilon}^{c,t}$. Then the within-round worker–average gap approximately satisfies*

$$d_i^{c,t} \approx (I - \eta H_{c,t}) d_i^{c,t-1} - \eta \zeta_i^{c,t}. \tag{3}$$

Lemma 1 shows that worker-average gaps are driven by centered stochastic-gradient noise, while the local Hessian determines how previous gaps propagate through the within-round dynamics.

Theorem 2 (Worker–average gap covariance as propagated stochastic noise) *Under the approximation in Lemma 1, suppose that the Hessian does not change substantially within one communication round, so that $H_{c,t} \approx H_c$ for $t = 1, \dots, \tau$. Then the final worker–average gap at the end of communication round c satisfies*

$$\Delta_i^{c+1} = d_i^{c,\tau} \approx -\eta \sum_{t=1}^{\tau} (I - \eta H_c)^{\tau - t} \zeta_i^{c,t} \tag{4}$$

Assume further that the stochastic-gradient noise is independent across workers and local steps, with $\mathrm{Cov}(\epsilon_i^{c,t}) = \Sigma_c$. Then the centered worker noise satisfies $\mathrm{Cov}(\zeta_i^{c,t}) = \left(1 - \frac{1}{M}\right)\Sigma_c$ and the worker-gap covariance is

$$\boxed{\mathrm{Cov}(\Delta_i^{c+1}) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sum_{q=0}^{\tau-1} (I - \eta H_c)^q \Sigma_c (I - \eta H_c)^q} \tag{5}$$

The takeaway from Theorem 2 is that the covariance of worker-average gaps at the end of the communication round, is shaped jointly by the stochastic-gradient noise covariance Σ_c and the local curvature H_c. We next project this covariance onto individual Hessian eigendirections to characterize where the worker disagreement is largest in the eigenbasis.

Proposition 3 (Directional gap variance under noise–curvature coupling) *Let $H_c = U_c \Lambda_c U_c^\top$ and $\Lambda_c = \mathrm{diag}(\lambda_{1,c}, \ldots, \lambda_{D,c})$, where $u_{r,c}$, the r-th column of U_c, is the Hessian eigenvector corresponding to eigenvalue $\lambda_{r,c}$. Suppose that the stochastic-gradient noise covariance is approximately diagonal in the Hessian eigenbasis: $U_c^\top \Sigma_c U_c \approx \mathrm{diag}\left(\sigma_{1,c}^2, \ldots, \sigma_{D,c}^2\right)$, where $\sigma_{r,c}^2$ denotes the noise variance along Hessian eigendirection $u_{r,c}$. Then*

$$\mathrm{Var}\left(\langle \Delta_i^{c+1}, u_{r,c} \rangle\right) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sigma_{r,c}^2 \sum_{q=0}^{\tau-1} (1 - \eta\lambda_{r,c})^{2q}. \tag{6}$$

We write $\mathrm{Var}\left(\langle \Delta_i^{c+1}, u_{r,c} \rangle\right) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sigma_{r,c}^2 \psi_\tau(\eta\lambda_{r,c})$ where $\psi_\tau(a) := \sum_{q=0}^{\tau-1} (1-a)^{2q}$. Furthermore, if the noise variance obeys the empirically supported noise–curvature scaling $\sigma_{r,c}^2 \propto \lambda_{r,c}^\gamma$ where $1 \le \gamma \le 2$ [18], then

$$\mathrm{Var}\left(\langle \Delta_i^{c+1}, u_{r,c} \rangle\right) \propto \lambda_{r,c}^\gamma \psi_\tau(\eta\lambda_{r,c}). \tag{7}$$

Proposition 3 links worker disagreement to Hessian curvature at the eigendirection level. The term $\sigma_{r,c}^2$ captures how much stochastic-gradient noise is injected along $u_{r,c}$, while $\psi_\tau(\eta\lambda_{r,c})$ captures how local SGD propagates that noise before synchronization. Recent evidence suggests that $\sigma_{r,c}^2$ itself grows with curvature, approximately as $\lambda_{r,c}^\gamma$ where $1 \le \gamma \le 2$ [18]. Under this coupling, worker–average gaps tend to become large along high-curvature Hessian directions. Since the dominant subspace is precisely the span of the leading Hessian eigendirections, i.e., the directions with the largest eigenvalues, this suggests that the span of worker–average gaps can be used as a data-driven estimator of the dominant subspace.

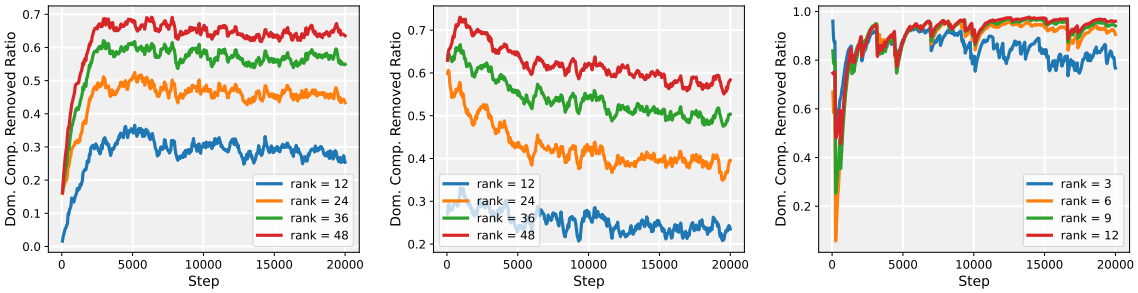
The detailed proofs of the theory presented here can be found in Section B of the Appendix.

4. Worker–Average Gaps as a Dominant Subspace Estimator

We test our theoretical claims in Section 3 by constructing a subspace from observed worker gaps during standard Local SGD and measuring its alignment with the top Hessian eigenspace. To form the gap-based proxy subspace, we maintain a FIFO buffer of worker–average gaps collected during standard Local SGD training. At synchronization round c, we insert the observed gaps $\{\Delta_i^c\}_{i=1}^{M-1}$ into the buffer (Not M due to linear dependence). We denote the current buffer by $Z_c = [z_1, \ldots, z_B] \in \mathbb{R}^{D \times B}$ where each buffer entry z_j is a previously observed worker–average gap Δ_i^c, B is the buffer capacity, and D is the number of model parameters. We construct the gap subspace from the Gram matrix $G_c = Z_c^\top Z_c$. Let $G_c = V_c \Omega_c V_c^\top$ be its eigendecomposition. We form the orthonormal basis $Q_c = Z_c V_c \Omega_c^{-1/2}$. Then $Q_c \in \mathbb{R}^{D \times B_c}$ (B_c is the retained rank) has orthonormal columns and spans the column space of the gap buffer. Additional implementation details are given in Appendix A.3.

Recall that P_C denotes the orthogonal projector onto the dominant Hessian subspace, obtained from the top-C Hessian eigendirections. Similarly, we define $P_{Q_c} = Q_c Q_c^\top$ as the projector obtained from the worker–average gap subspace. For a vector v, we measure how much of its true dominant component is removed by the proxy filter $I - P_{Q_c}$ using

$$\rho_c(v) := 1 - \frac{\|P_C(I - P_{Q_c})v\|_2}{\|P_C v\|_2}. \tag{8}$$



(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformer on SST2-5k

Figure 2: Dominant-component removal fraction achieved by worker-gap subspaces of different buffer capacities. The worker-gap subspace captures a large portion of the true dominant Hessian component across architectures, with coverage improving as the buffer capacity increases. (Curves are shown with exponential moving average smoothing for clarity.)

Thus, $\rho_c(v)$ measures the fraction of the true dominant component of v suppressed by the worker-gap proxy subspace Q_c. Larger values indicate that the gap subspace more effectively captures the Hessian-based dominant directions. In our experiments, we evaluate this quantity on the full-batch gradient $v = \nabla f(\bar{\theta}^c)$ at the averaged model $\bar{\theta}^c$ at communication round c, similar to the setup described in *Observation 2* of Section 2.

To assess effectiveness across architectures and data modalities, we run experiments on three model–dataset pairs: a tanh FC network on MNIST, a ReLU CNN on CIFAR10, and a 2-layer Transformer on SST2. For each dataset, we train on a subset of 5k samples using standard Local SGD with $M = 4$ workers and communication period $\tau = 5$. Additional experimental details are provided in Appendix A.3. Following [3, 13], we set $C = 10$ for MNIST and CIFAR10, and $C = 2$ for SST2. Accordingly, for MNIST and CIFAR10, we sweep FIFO buffer capacities $B \in \{12, 24, 36, 48\}$. For SST2, we sweep $B \in \{3, 6, 9, 12\}$. Results are shared in Figure 2.

Figure 2 shows that the worker-gap subspace removes a substantial fraction of the gradient's dominant Hessian component across all three settings. Larger buffer capacities consistently improve this removal, showing that having more recent worker–average gaps capture more of the dominant eigenspace. The effect is strongest for SST2, where even small buffers remove most of the dominant component, while MNIST and CIFAR10 improve more gradually with rank. Overall, these results support our theory that worker disagreement concentrates along sharp, curvature-sensitive directions, making worker–average gaps an effective Hessian-free proxy for the dominant subspace.

5. Conclusion and Future Work

In this work, we showed that the worker–average gaps naturally produced by Local SGD carry useful information about sharp directions in the loss landscape. Our theory links worker–average gaps to stochastic-gradient noise and Hessian curvature, showing that under a noise–curvature coupling [18], gaps become large along high-curvature Hessian directions. This motivates using their span as a Hessian-free dominant-subspace estimator, which empirically captures a substantial fraction of the dominant Hessian component across model–dataset pairs. Appendix C provides additional results on the effect of τ on the quality of the gap-based subspace; together with preliminary filtering experiments showing that suppressing the gap-estimated dominant component and amplifying its orthogonal complement can accelerate optimization. We view the design of optimization mechanisms that leverage worker–average gaps as a promising direction for future work.

References

[1] Gerard Ben Arous, Reza Gheissari, Jiaoyang Huang, and Aukosh Jagannath. High-dimensional sgd aligns with emerging outlier eigenspaces. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 47732–47778, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d10d6b28d74c4f0fcab588feeb6fe7d6-Paper-Conference.pdf.

[2] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

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

[4] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck R. Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems*, pages 11082–11094, 2019.

[5] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[7] Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3411–3420, 2023. doi: 10.1109/TPAMI.2022.3178101.

[8] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[9] Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

[10] Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

[11] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[12] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[13] Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does sgd really happen in tiny subspaces? In *International Conference on Learning Representations*, volume 2025, pages 8086–8120, 2025.

[14] Sebastian Urban Stich. Local sgd converges fast and communicates little. In *ICLR 2019-International Conference on Learning Representations*, 2019.

[15] Qian-Yuan Tang, Yufei Gu, Yunfeng Cai, Mingming Sun, Ping Li, zhou Xun, and Zeke Xie. Investigating the overlooked hessian structure: From CNNs to LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=o62ZzfCEwZ.

[16] Zeke Xie, Qian-Yuan Tang, Mingming Sun, and Ping Li. On the overlooked structure of stochastic gradients. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 66257–66276. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d0b2eda0386f477ab14d7e181e16c899-Paper-Conference.pdf.

[17] Can Yaras, Peng Wang, Laura Balzano, and Qing Qu. Compressible dynamics in deep overparameterized low-rank learning and adaptation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56946–56965. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yaras24a.html.

[18] Yikuan Zhang, Ning Yang, and Yuhai Tu. On the superlinear relationship between sgd noise covariance and loss landscape curvature. *arXiv preprint arXiv:2602.05600*, 2026.

[19] WenJie Zhou, Bohan Wang, Wei Chen, and Xueqi Cheng. BSFA: Leveraging the subspace dichotomy to accelerate neural network training. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18834–18849, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.952. URL https://aclanthology.org/2025.emnlp-main.952/.

[20] Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.

Appendix A. Experiment Details and More Results

Here, we provide more details of the experimental setup. To run the experiments, we use Python 3.11.15 programming language, PyTorch 2.8.0 framework with torchvision 0.23.0 and CUDA 12.6. We used 3 machines, each equipped with 4 $\times$ GTX 1080 GPUs to run the experiments.

A.1. Local SGD Training

We consider distributed empirical risk minimization with M workers minimizing

$$f(\theta) = \frac{1}{N} \sum_{n=1}^{N} \ell(h_\theta(x_n), y_n),$$

where $h_\theta : \mathcal{X} \to \mathcal{Y}$ is a neural network with parameters $\theta \in \mathbb{R}^D$, and ℓ denotes the per-sample loss. The training set is partitioned IID across the M workers. We denote the local dataset on worker i by $\mathcal{D}_i$, so that, for equal-size partitions,

$$f(\theta) = \frac{1}{M} \sum_{i=1}^{M} f_i(\theta), \qquad f_i(\theta) = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \ell(h_\theta(x), y).$$

Local SGD proceeds in communication rounds. We use c to index communication rounds and s to index local steps within a communication round. At the beginning of communication round c, all workers are initialized from the same averaged model $\bar{\theta}^c$. Each worker then independently performs τ stochastic-gradient steps on minibatches sampled from its local data:

$$\theta_i^{c,s+1} = \theta_i^{c,s} - \eta \nabla f_{i,\mathcal{B}_i^{c,s}}\left(\theta_i^{c,s}\right), \qquad s = 0, \ldots, \tau - 1,$$

where $\mathcal{B}_i^{c,s}$ is the minibatch sampled by worker i at local step s of communication round c. After τ local steps, the workers synchronize by averaging their local parameters:

$$\bar{\theta}^{c+1} = \frac{1}{M} \sum_{i=1}^{M} \theta_i^{c,\tau}.$$

Thus, compared with fully synchronized minibatch SGD, Local SGD communicates only once every τ local updates which reduces the time spent on communication by a factor of τ. In our experiments, we also record the worker–average gap vectors before synchronization,

$$\Delta_i^{c+1} = \theta_i^{c,\tau} - \bar{\theta}^{c+1}, \qquad i = 1, \ldots, M,$$

which quantify the disagreement accumulated by the local workers during the communication period. These gaps are the basic objects used to construct the gap-based subspace estimator analyzed in the main text. Pseudo-code is provided in Algorithm 1.

Algorithm 1 Local SGD with Worker–Average Gaps

Require: Number of workers M, communication period τ, number of communication rounds C, learning rate η

Initialize global model $\bar{\theta}^0$
for $c = 0, 1, \ldots, C-1$ **do**
  for each worker $i = 1, \ldots, M$ in parallel **do**
    Set $\theta_i^{c,0} \leftarrow \bar{\theta}^c$
    for $s = 0, 1, \ldots, \tau - 1$ **do**
      Sample minibatch $\mathcal{B}_i^{c,s}$
      $g_i^{c,s} \leftarrow \nabla_\theta f_{\mathcal{B}_i^{c,s}}(\theta_i^{c,s})$
      $\theta_i^{c,s+1} \leftarrow \theta_i^{c,s} - \eta g_i^{c,s}$
    end for
  end for
  Compute the worker average $\bar{\theta}^{c+1} = \frac{1}{M}\sum_{i=1}^{M} \theta_i^{c,\tau}$
  Compute worker–average gaps $\Delta_i^{c+1} = \theta_i^{c,\tau} - \bar{\theta}^{c+1}, \qquad i = 1, \ldots, M.$
end for
return $\bar{\theta}^C$

A.2. Dominant–Bulk Subspace Phenomenon in Local SGD Training

Hyperparameters. We train all models using Local SGD with $M = 4$ workers and communication period $\tau = 5$, i.e., each worker performs five local SGD steps between consecutive synchronizations. Training is run for 10,000 local steps, corresponding to 2,000 communication rounds. The local optimizer is vanilla SGD without momentum, and we do not use weight decay. The per-worker batch size is set to 50. For each dataset, we randomly select 5,000 training samples, which we denote by MNIST-5k, CIFAR10-5k, and SST2-5k. We use mean squared error (MSE) as the training loss f for all experiments. We follow the experimental protocol of [13], which covers multiple architectures and data modalities. Below, we provide the architectural details of the models used in our experiments.

FC tanh: We train a fully connected neural network with two hidden layers and tanh activations on MNIST-5k. Each hidden layer has width 200. Thus, the three linear weight matrices have sizes $[\texttt{input\_size}, 200]$, $[200, 200]$, and $[200, \texttt{output\_size}]$, with corresponding bias vectors of sizes $[200]$, $[200]$, and $[\texttt{output\_size}]$. Since the model is trained on MNIST, $\texttt{output\_size} = 10$.

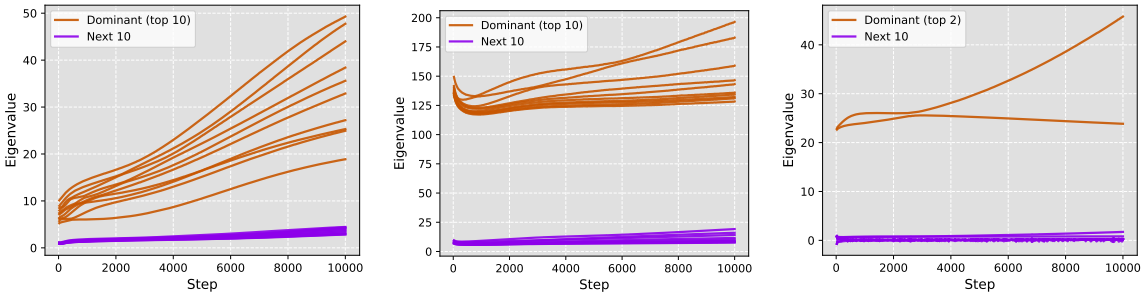
CNN ReLU: We train a convolutional neural network with ReLU activations on CIFAR10-5k. The network consists of two convolutional blocks, each using 32 output channels. Each block applies a 3×3 convolution with stride 1 and padding 1, followed by a ReLU activation and 2×2 max pooling. After the two convolutional blocks, the feature map is flattened and passed to a linear classifier. Since the model is trained on CIFAR10, the output dimension is set to 10.

Transformer: For SST2-5k, we train a small Transformer encoder for binary classification. The model uses token and positional embeddings with hidden dimension 64. The maximum sequence length is set to the dataset-specific value used in preprocessing. The encoder has two Transformer layers, each with two attention heads. Each layer contains a multi-head self-attention block followed by a residual connection and LayerNorm, and a feed-forward block consisting of two linear layers with a ReLU activation between them, again followed by a residual connection and LayerNorm. No

dropout is used. The final sequence representation is obtained by mean pooling over the sequence dimension, and a linear classifier maps the pooled representation to two output classes. The classifier head is initialized with zero weights and zero bias.

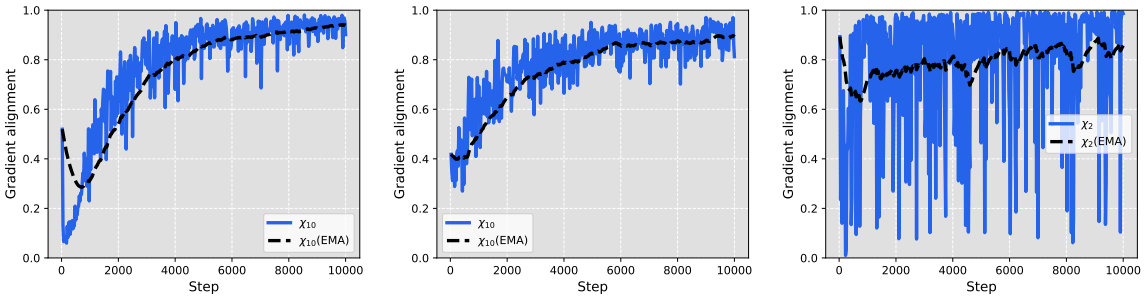
The learning rate is set to $\eta = 0.05$ for FC tanh and it is set to $\eta = 0.005$ for CNN ReLU and Transformer experiments.

To obtain the leading eigenvalue–eigenvector pairs of the Hessian, we use the Lanczos algorithm. The Hessian is evaluated on the full training subset, i.e., using all 5,000 samples from the corresponding dataset. We compute the Hessian spectrum and collect the relevant training statistics every 25 local steps, which corresponds to every 5 communication rounds since $\tau = 5$. Unless stated otherwise, both the leading Hessian eigenvalue–eigenvector pairs and the full-batch gradients are evaluated at the synchronized model $\bar{\theta}^c$ at the beginning of a communication round, before the workers perform their local updates. In other words, these quantities are computed at the common parameter vector obtained from the most recent synchronization, prior to the subsequent local exploration phase.



(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformers on SST2-5k

Figure 3: Evolution of Hessian eigenvalues during Local SGD training. Across all three settings, a small number of dominant eigenvalues separates from the remaining spectrum, confirming the dominant–bulk Hessian structure.

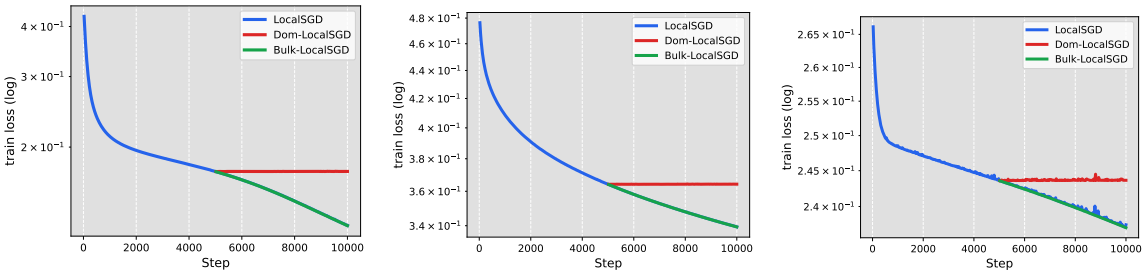


(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformers on SST2-5k

Figure 4: Alignment between the gradient and the dominant Hessian eigenspace over training. Gradients become increasingly aligned with the dominant subspace, consistent with the dominant-space concentration observed in centralized training.

Observation 1: Gradients along Local SGD iterates align with the dominant subspace. We first measure the alignment of the full-batch gradient with the Hessian's Dominant subspace along the Local SGD trajectory. Specifically, at each synchronization round, after parameter averaging, all workers share the averaged parameter vector $\bar{\theta}$. We compute the full-batch gradient $\nabla f(\bar{\theta})$ and Hessian $H(\bar{\theta})$, form the dominant subspace $\mathcal{S}_C(\bar{\theta})$, and report $\chi_C(\nabla f(\bar{\theta}); \bar{\theta})$. We perform this analysis on three model–dataset pairs: a fully-connected tanh network on MNIST, a ReLU CNN on CIFAR10, and a 2-layer Transformer on SST2.

Figure 4 plots $\chi_C(\nabla f(\bar{\theta}); \bar{\theta})$ over training for all three model–dataset pairs, and Figure 3 shows the corresponding evolution of the Hessian spectrum. Across all settings, the full-batch gradients remain strongly aligned with the dominant subspace, while the leading Hessian eigenvalues stay well separated from the rest of the spectrum. These observations mirror the dominant-subspace alignment previously reported for single-worker SGD, and motivate the next question: whether training can be carried out using only the Dominant or Bulk components of the Local SGD update.



(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformers on SST2-5k

Figure 5: Training loss under Local SGD and updates restricted to the dominant or bulk subspaces. Suppressing the bulk component slows or stalls progress, while retaining the bulk component preserves useful descent, indicating that effective learning relies strongly on the flatter bulk directions.

Observation 2: Local SGD training cannot be performed in the Dominant Subspace. Starting from checkpoints obtained during standard Local SGD training, we compare three continuation runs: standard Local SGD, Dom-Local SGD, and Bulk-Local SGD. Let $\bar{\theta}^c$ denote the synchronized average model at the beginning of communication round c. All workers are initialized from this point, $\theta_i^{c,0} = \bar{\theta}^c$, and perform τ local SGD steps to obtain $\theta_i^{c,\tau}$. We define the average outer step over the communication round as

$$\bar{p}^c := \frac{1}{M} \sum_{i=1}^{M} \theta_i^{c,\tau} - \bar{\theta}^c. \tag{9}$$

Thus, standard Local SGD updates the synchronized model as

$$\bar{\theta}^{c+1} = \bar{\theta}^c + \bar{p}^c. \tag{10}$$

To isolate which part of this displacement is responsible for training progress, we also consider projected continuation runs. Let $\mathcal{S}_C(\bar{\theta}^c)$ denote the dominant subspace at $\bar{\theta}^c$, and let P_C^c be the orthogonal projection matrix onto this subspace. Dom-Local SGD keeps only the component of the average outer step in the dominant subspace,

$$\bar{\theta}^{c+1} = \bar{\theta}^c + P_C^c \bar{p}^c, \tag{11}$$

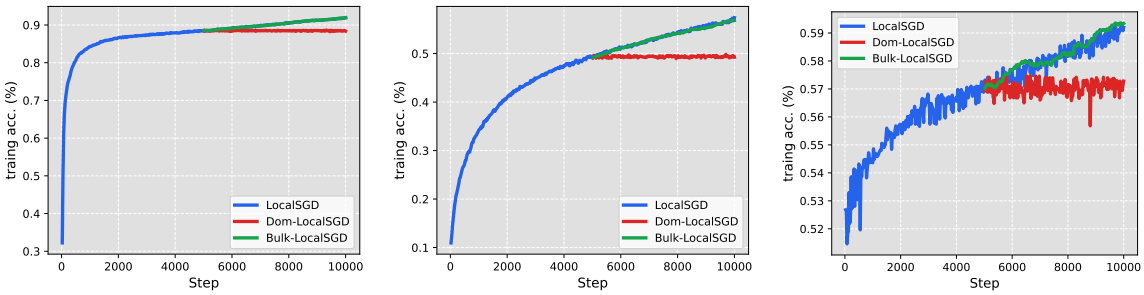
whereas Bulk-Local SGD keeps only the complementary bulk component,

$$\bar{\theta}^{c+1} = \bar{\theta}^{c} + (I - P_C^c)\bar{p}^c. \tag{12}$$

In all three cases, the local trajectories are generated in the same way; the only difference is whether the resulting communication-round displacement is used directly, projected onto the dominant subspace, or projected onto the bulk subspace.

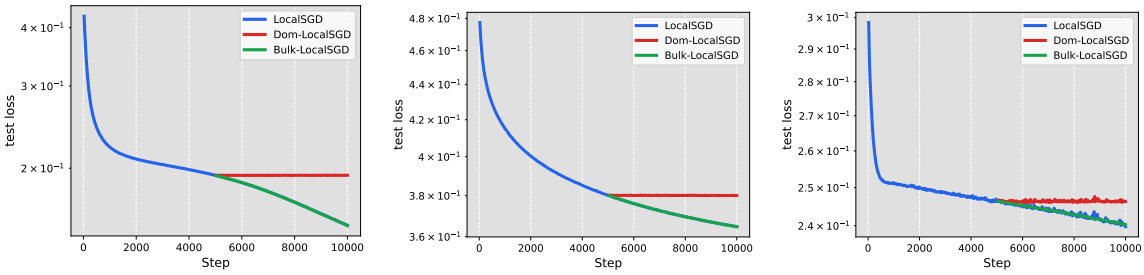
Figure 5 shows that Dom-Local SGD fails to sustain training, with the loss either stalling or diverging after the projection is introduced. In contrast, Bulk-Local SGD closely matches standard Local SGD and sometimes converges slightly faster. Thus, despite the strong dominant-subspace alignment of full-batch gradients, the communication-round updates that sustain training are primarily carried by the bulk component. This dominant–bulk separation is consistent with the phenomenon previously observed in single-worker SGD training.

For completeness, we also share how train accuracy, test loss and test accuracy changes for different model–dataset pairs in Figures 6, 7 and 8 respectively.



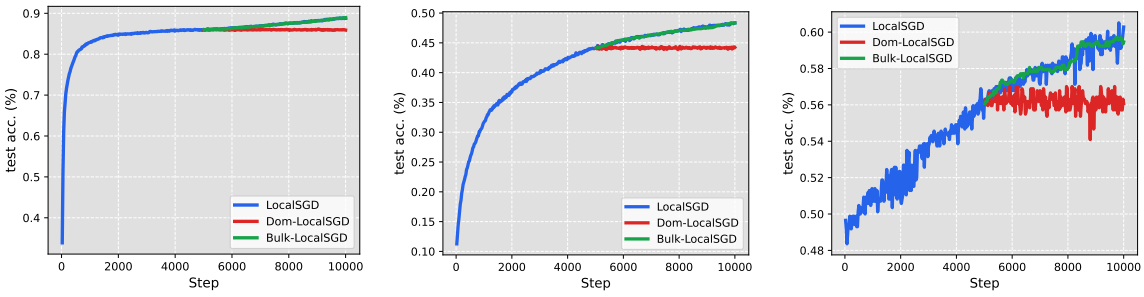
(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformers on SST2-5k

Figure 6: Training accuracy under Local SGD and updates restricted to the dominant or bulk subspaces.



(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformers on SST2-5k

Figure 7: Test loss under Local SGD and updates restricted to the dominant or bulk subspaces.



(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformers on SST2-5k

Figure 8: Test accuracy under Local SGD and updates restricted to the dominant or bulk subspaces.

A.3. Dominant Subspace Estimation using Worker-Average Gaps

To form a proxy subspace for estimating the dominant Hessian subspace, we maintain a FIFO buffer of worker–average gaps collected during standard Local SGD training. At synchronization round c, the worker–average gap is

$$\Delta_i^c := \theta_i^{c,\tau} - \frac{1}{M}\sum_{j=1}^{M} \theta_j^{c,\tau}.$$

Equivalently, since all workers start the round from the same synchronized parameter vector $\theta_i^{c,0} = \theta_j^{c,0}$, if $p_i^c := \theta_i^{c,\tau} - \theta_i^{c,0}$ denotes the outer displacement or progress of worker i, then

$$\Delta_i^c = p_i^c - \frac{1}{M}\sum_{j=1}^{M} p_j^c.$$

Thus, the worker–average gap is exactly the deviation of each worker's accumulated local update from the cross-worker average accumulated update.

Since $\sum_{i=1}^{M} \Delta_i^c = 0$, at most $M-1$ linearly independent gaps are obtained from each synchronization round. We insert these gaps into a FIFO buffer

$$Z_c = [z_1, \ldots, z_B] \in \mathbb{R}^{D\times B},$$

where each buffer entry z_j is a previously observed worker–average gap $\Delta_i^{c'}$, B is the buffer capacity, and D is the number of model parameters. We form the proxy subspace using the full effective rank of the current buffer. Concretely, we compute the Gram matrix

$$G_c := Z_c^\top Z_c,$$

keep the eigenvectors whose eigenvalues are above a small relative threshold, and write the retained eigendecomposition as

$$G_c V_c = V_c \Omega_c.$$

We then construct an orthonormal basis for the retained column space of Z_c as

$$Q_c := Z_c V_c \Omega_c^{-1/2}.$$

The resulting subspace

$$\widehat{\mathcal{S}}_c := \mathrm{span}(Q_c)$$

is our worker-gap proxy for directions of large recent worker disagreement. Its effective rank is $B_c = \dim(\widehat{\mathcal{S}}_c)$, which equals the number of retained eigenvalues of G_c. In practice, this is typically the buffer size B, except when some eigenvalues of G_c fall below the relative threshold 10^{-8}.

Algorithm 2 summarizes the construction of the worker–average gap subspace and the dominant component removal metric ρ used in our experiments. At each synchronization round, we first compute the deviations of the local worker parameters from their synchronized average and insert these vectors into a FIFO buffer. We then compute an orthonormal basis for the span of the buffered gaps using the Gram matrix $Z_c^\top Z_c$, which avoids explicitly forming a large $D \times D$ covariance matrix. The resulting basis Q_c defines the proxy projector $P_{Q_c} = Q_c Q_c^\top$. Given the dominant Hessian basis U_C, we measure how much of the true dominant component of the full-batch gradient $g_c = \nabla f(\bar{\theta}^c)$ is removed by the gap-subspace filter $I - P_{Q_c}$ using the ratio $\rho_c(g_c)$.

Algorithm 2 Worker–Average Gap Subspace and Dominant Component Removal

Require: Local worker parameters $\{\theta_i^{c,\tau}\}_{i=1}^M$ at synchronization round c, buffer capacity B, relative threshold ε, dominant Hessian basis U_C, full-batch gradient g_c

Compute the synchronized average $\bar{\theta}^c \leftarrow \frac{1}{M}\sum_{i=1}^M \theta_i^{c,\tau}$

Compute worker–average gaps $\Delta_i^c \leftarrow \theta_i^{c,\tau} - \bar{\theta}^c, \qquad i = 1, \ldots, M-1$ (only $M-1$ are linearly independent since $\sum_{i=1}^M \Delta_i^c = 0$)

Insert $\{\Delta_i^c\}_{i=1}^{M-1}$ into the FIFO buffer and discard the oldest entries if the buffer size exceeds B

Let the current buffer be $Z_c = [z_1, \ldots, z_{B_c}] \in \mathbb{R}^{D \times B_c}$, where $B_c \le B$ is the current number of stored gaps

Form the Gram matrix $G_c \leftarrow Z_c^\top Z_c$

Compute the eigendecomposition $G_c V_c = V_c \Omega_c$

Retain eigenpairs whose eigenvalues satisfy $\omega_r \ge \varepsilon \omega_{\max}(G_c)$

Let V_c^{ret} and Ω_c^{ret} denote the retained eigenvectors and eigenvalues

Construct the orthonormal gap-subspace basis $Q_c \leftarrow Z_c V_c^{\mathrm{ret}} \left(\Omega_c^{\mathrm{ret}}\right)^{-1/2}$

Compute the dominant Hessian projector $P_C \leftarrow U_C U_C^\top$

Compute the worker-gap projector $P_{Q_c} \leftarrow Q_c Q_c^\top$

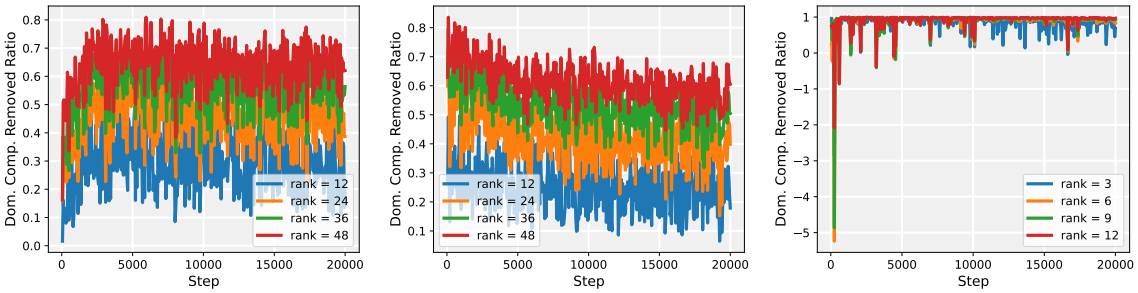
Compute the dominant component removal ratio $\rho_c(g_c) \leftarrow 1 - \frac{\|P_C(I - P_{Q_c})g_c\|_2}{\|P_C g_c\|_2}$

return Gap-subspace basis Q_c and dominant component removal ratio $\rho_c(g_c)$

In section 4, we share the EMA smoothed $\rho_c(\cdot)$ curves for clarity. For completeness, we also share the plots with the raw values in Figure 9.

Appendix B. Theoretical Characterization of Worker-Average Gaps

Here, we provide full proofs of the theory presented in the main body of the paper. The derivations of Lemma 1, Theorem 2 and Proposition 3 are presented in Subsections B.1, B.2 and B.3



(a) FC on MNIST-5k (b) CNN on CIFAR10-5k (c) Transformer on SST2-5k

Figure 9: Raw dominant-component removal fraction achieved by worker-gap subspaces of different buffer capacities. The worker-gap subspace captures a large portion of the true dominant Hessian component across architectures, with coverage improving as the buffer capacity increases.

B.1. Linearized worker–average gap dynamics

We consider Local SGD training with M workers. Let $\theta_i^{c,t}$ denote the parameter vector of worker i after t local steps in communication round c, where $t = 0, \dots, \tau$. At the beginning of each round, all workers are synchronized:

$$\theta_i^{c,0} = \bar{\theta}^c, \qquad i = 1, \dots, M.$$

Each worker performs local SGD updates using stochastic gradients of the form

$$g_i^{c,t} = \nabla f(\theta_i^{c,t-1}) + \epsilon_i^{c,t}, \qquad \mathbb{E}[\epsilon_i^{c,t} \mid \theta_i^{c,t-1}] = 0. \tag{13}$$

Thus,

$$\theta_i^{c,t} = \theta_i^{c,t-1} - \eta \left[\nabla f(\theta_i^{c,t-1}) + \epsilon_i^{c,t} \right]. \tag{14}$$

To describe the relative motion of the workers within a communication round, define the within-round average

$$\bar{\theta}^{c,t} := \frac{1}{M} \sum_{j=1}^{M} \theta_j^{c,t}, \tag{15}$$

and the worker deviation from this average

$$d_i^{c,t} := \theta_i^{c,t} - \bar{\theta}^{c,t}. \tag{16}$$

By construction,

$$\frac{1}{M} \sum_{i=1}^{M} d_i^{c,t} = 0, \qquad d_i^{c,0} = 0.$$

After τ local steps, the next synchronized model is

$$\bar{\theta}^{c+1} := \bar{\theta}^{c,\tau},$$

and the worker–average gap at communication round $c+1$ is

$$\Delta_i^{c+1} := \theta_i^{c,\tau} - \bar{\theta}^{c+1} = d_i^{c,\tau}. \tag{17}$$

We next linearize the population gradient around the within-round average $\bar{\theta}^{c,t-1}$. Define the local Hessian

$$H_{c,t} := \nabla^2 f(\bar{\theta}^{c,t-1}). \tag{18}$$

Then

$$\nabla f(\theta_i^{c,t-1}) = \nabla f(\bar{\theta}^{c,t-1}) + H_{c,t} d_i^{c,t-1} + \mathcal{R}_i^{c,t}, \tag{19}$$

where $\mathcal{R}_i^{c,t}$ denotes the higher-order Taylor remainder.

Averaging (14) across workers gives

$$\bar{\theta}^{c,t} = \bar{\theta}^{c,t-1} - \eta \left[\frac{1}{M} \sum_{j=1}^{M} \nabla f(\theta_j^{c,t-1}) + \bar{\epsilon}^{c,t} \right], \qquad \bar{\epsilon}^{c,t} := \frac{1}{M} \sum_{j=1}^{M} \epsilon_j^{c,t}. \tag{20}$$

Define the centered worker noise

$$\zeta_i^{c,t} := \epsilon_i^{c,t} - \bar{\epsilon}^{c,t}. \tag{21}$$

Subtracting (20) from (14) yields

$$d_i^{c,t} = d_i^{c,t-1} - \eta \left(\nabla f(\theta_i^{c,t-1}) - \frac{1}{M} \sum_{j=1}^{M} \nabla f(\theta_j^{c,t-1}) \right) - \eta \zeta_i^{c,t}. \tag{22}$$

Using (19) and $\frac{1}{M} \sum_{j=1}^{M} d_j^{c,t-1} = 0$, we obtain

$$d_i^{c,t} = (I - \eta H_{c,t})\, d_i^{c,t-1} - \eta \zeta_i^{c,t} - \eta \left(\mathcal{R}_i^{c,t} - \bar{\mathcal{R}}^{c,t} \right), \tag{23}$$

where

$$\bar{\mathcal{R}}^{c,t} := \frac{1}{M} \sum_{j=1}^{M} \mathcal{R}_j^{c,t}.$$

Ignoring the higher-order terms gives the approximate recurrence

$$d_i^{c,t} \approx (I - \eta H_{c,t})\, d_i^{c,t-1} - \eta \zeta_i^{c,t}. \tag{24}$$

This recurrence shows that the shared center gradient cancels across workers: worker deviations are driven by centered stochastic-gradient noise, while the local Hessian determines how existing deviations are propagated.

B.2. Worker–average gap covariance as propagated stochastic noise

We now specialize the recurrence (24) to obtain an explicit covariance expression. Assume that the Hessian does not change substantially within one communication round, so that

$$H_{c,t} \approx H_c, \qquad t = 1, \ldots, \tau.$$

Then

$$d_i^{c,t} \approx (I - \eta H_c) d_i^{c,t-1} - \eta \zeta_i^{c,t}. \tag{25}$$

For a noise vector injected at local step s, define the propagation matrix from step s to the end of the local phase as

$$A_{c,s} := (I - \eta H_c)^{\tau - s}. \tag{26}$$

Unrolling (25) from $d_i^{c,0} = 0$ gives

$$\Delta_i^{c+1} = d_i^{c,\tau} \approx -\eta \sum_{s=1}^{\tau} A_{c,s} \zeta_i^{c,s}. \tag{27}$$

This expression shows that worker–average gaps are not merely raw stochastic-gradient noise. Rather, the noise injected at each local step is propagated through the local optimization dynamics before synchronization. Assume that the stochastic-gradient noise is independent across workers and local steps, with

$$\mathrm{Cov}(\epsilon_i^{c,t}) = \Sigma_c.$$

Since

$$\bar{\epsilon}^{c,t} = \frac{1}{M} \sum_{j=1}^{M} \epsilon_j^{c,t},$$

we have

$$\zeta_i^{c,t} = \epsilon_i^{c,t} - \bar{\epsilon}^{c,t} = \left(1 - \frac{1}{M}\right) \epsilon_i^{c,t} - \frac{1}{M} \sum_{j \neq i} \epsilon_j^{c,t}.$$

Using independence across workers,

$$\begin{aligned} \mathrm{Cov}(\zeta_i^{c,t}) &= \left(1 - \frac{1}{M}\right)^2 \Sigma_c + \frac{M-1}{M^2} \Sigma_c \\ &= \left(1 - \frac{1}{M}\right) \Sigma_c. \end{aligned} \tag{28}$$

Starting from (27), and assuming $\mathbb{E}[\zeta_i^{c,s}] = 0$, the gap covariance is

$$\mathrm{Cov}(\Delta_i^{c+1}) \approx \eta^2 \sum_{s=1}^{\tau} \sum_{\ell=1}^{\tau} A_{c,s} \mathbb{E}\left[\zeta_i^{c,s} (\zeta_i^{c,\ell})^\top\right] A_{c,\ell}^\top. \tag{29}$$

By independence across local steps, the cross terms vanish for $s \neq \ell$, and therefore

$$\mathrm{Cov}(\Delta_i^{c+1}) \approx \eta^2 \sum_{s=1}^{\tau} A_{c,s} \, \mathrm{Cov}(\zeta_i^{c,s}) A_{c,s}^\top. \tag{30}$$

Substituting (28) gives

$$\mathrm{Cov}(\Delta_i^{c+1}) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sum_{s=1}^{\tau} A_{c,s} \Sigma_c A_{c,s}^\top. \tag{31}$$

Using $A_{c,s} = (I - \eta H_c)^{\tau - s}$, the symmetry of H_c, and reindexing with $q = \tau - s$, we obtain

$$\boxed{\mathrm{Cov}(\Delta_i^{c+1}) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sum_{q=0}^{\tau-1} (I - \eta H_c)^q \Sigma_c (I - \eta H_c)^q}. \tag{32}$$

Thus, the worker-gap covariance is shaped jointly by the stochastic-gradient noise covariance Σ_c and the local curvature H_c.

B.3. Directional gap variance under noise–curvature coupling

Equation (32) shows that the worker-gap covariance is determined by the interaction between the local curvature H_c and the stochastic-gradient noise covariance Σ_c. To interpret this expression directionally, let

$$H_c = U_c \Lambda_c U_c^\top, \qquad \Lambda_c = \mathrm{diag}(\lambda_{1,c}, \ldots, \lambda_{D,c}),$$

where $u_{r,c}$, the r-th column of U_c, is the Hessian eigenvector associated with eigenvalue $\lambda_{r,c}$. Assume that the noise covariance is approximately diagonal in this Hessian eigenbasis:

$$U_c^\top \Sigma_c U_c \approx \mathrm{diag}\left(\sigma_{1,c}^2, \ldots, \sigma_{D,c}^2\right).$$

Equivalently, $\sigma_{r,c}^2$ denotes the stochastic-gradient noise variance along Hessian eigendirection $u_{r,c}$.

Projecting (32) onto $u_{r,c}$ then gives

$$\mathrm{Var}\left(\langle \Delta_i^{c+1}, u_{r,c} \rangle\right) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sigma_{r,c}^2 \sum_{q=0}^{\tau-1} (1 - \eta\lambda_{r,c})^{2q}. \tag{33}$$

Define

$$\psi_\tau(a) := \sum_{q=0}^{\tau-1} (1-a)^{2q}.$$

Then (33) can be written as

$$\mathrm{Var}\left(\langle \Delta_i^{c+1}, u_{r,c} \rangle\right) \approx \eta^2 \left(1 - \frac{1}{M}\right) \sigma_{r,c}^2 \psi_\tau(\eta\lambda_{r,c}). \tag{34}$$

Thus, the worker deviation along a Hessian eigendirection is controlled by two factors: the stochastic-gradient noise strength $\sigma_{r,c}^2$ along that direction, and the curvature-dependent local-dynamics factor $\psi_\tau(\eta\lambda_{r,c})$.

Recent work on SGD noise and loss curvature suggests that the first factor is itself coupled to curvature. In particular, when the SGD noise covariance is expressed in the Hessian eigenbasis, its diagonal entries approximately obey

$$\sigma_{r,c}^2 \propto \lambda_{r,c}^\gamma, \qquad 1 \le \gamma \le 2,$$

with superlinear scaling $\gamma > 1$ observed in cross-entropy classification settings, while mean-squared error yields $\gamma \approx 1$. Under this empirically supported noise–curvature relation, (34) gives

$$\mathrm{Var}\left(\langle \Delta_i^{c+1}, u_{r,c} \rangle\right) \propto \lambda_{r,c}^\gamma \psi_\tau(\eta\lambda_{r,c}). \tag{35}$$

This expression directly links worker disagreement to Hessian curvature: directions with larger Hessian eigenvalues induce larger stochastic-gradient noise, and therefore larger worker-average deviations, up to the modulation introduced by $\psi_\tau(\eta\lambda_{r,c})$. Hence, under the observed noise–curvature coupling, worker-average gaps are expected to concentrate along high-curvature Hessian directions. This provides a theoretical motivation for using the span of worker-average gaps as a data-driven estimator of the dominant curvature subspace.

Appendix C. Additional Results

In this section, we provide additional experimental results.

C.1. Effect of the Communication Period on the Gap-Based Subspace

We investigate how the quality of the worker–average gap subspace changes with the communication period τ. To this end, we repeat the experimental setup described in Section 4 that uses three model–dataset pairs: a tanh fully connected network on MNIST-5k, a ReLU CNN on CIFAR-10-5k, and a 2-layer Transformer on SST-2-5k. As in the main experiments, Local SGD is performed with $M = 4$ workers.

For each setting, we construct the worker–average gap subspace using a FIFO buffer of recent gaps and evaluate its ability to capture the dominant Hessian subspace. We set the number of dominant Hessian directions to $C = 10$ for MNIST-5k and CIFAR-10-5k, and to $C = 2$ for SST-2-5k. For MNIST-5k and CIFAR-10-5k, we sweep buffer capacities $B \in \{12, 24, 48\}$, while for SST-2-5k, we sweep $B \in \{3, 6, 12\}$. We measure the effectiveness of the gap-induced subspace using the dominant component removal ratio $\rho_c(v)$ defined in Equation 8, evaluated throughout training. We repeat the analysis for communication periods $\tau \in \{2, 5, 10\}$. This way we can assess whether more frequent synchronization, which produces shorter and more locally linear worker trajectories, leads to a more accurate gap-based estimator of the dominant Hessian subspace.

We report the results in Figures 10, 11, and 12. In each plot, curves corresponding to the same communication period are grouped using the same color family, with different shades indicating different buffer ranks. Different communication periods are shown using distinct colors.

The effect of the communication period is consistent across FC-Tanh and CNN-ReLU. Smaller communication periods yield substantially higher dominant component removal. In particular, $\tau = 2$ consistently achieves the largest $\rho_c(v)$, followed by $\tau = 5$, while $\tau = 10$ gives the weakest removal. A striking observation is that the $\tau = 2$, $B = 24$ configuration is approximately as effective as the $\tau = 5$, $B = 48$ configuration in both settings. This suggests that more frequent synchronization can compensate for a smaller gap buffer, because shorter local trajectories produce worker–average gaps that more faithfully reflect the local curvature–noise geometry around the synchronized model. This trend is also consistent with the constant-Hessian approximation used in the gap covariance analysis in Section B.2: for smaller τ, the Hessian is more likely to remain approximately stable over the local trajectory, whereas for larger τ, nonlinear trajectory drift and Hessian variation can make the collected gaps a less accurate proxy for the dominant Hessian subspace at the synchronization point.

The Transformer on SST-2-5k shows a weaker dependence on the communication period. Across most buffer ranks and communication periods, the dominant component removal quickly becomes large and remains close to one throughout training. This is likely because the dominant subspace dimension in this setting is only $C = 2$, making it easier for even a small worker-gap buffer to capture the relevant directions. Overall, these results support the view that worker–average gaps provide an effective Hessian-free estimator of the dominant subspace, with the strongest and most consistent performance obtained when synchronization is frequent and the buffer has sufficient rank.

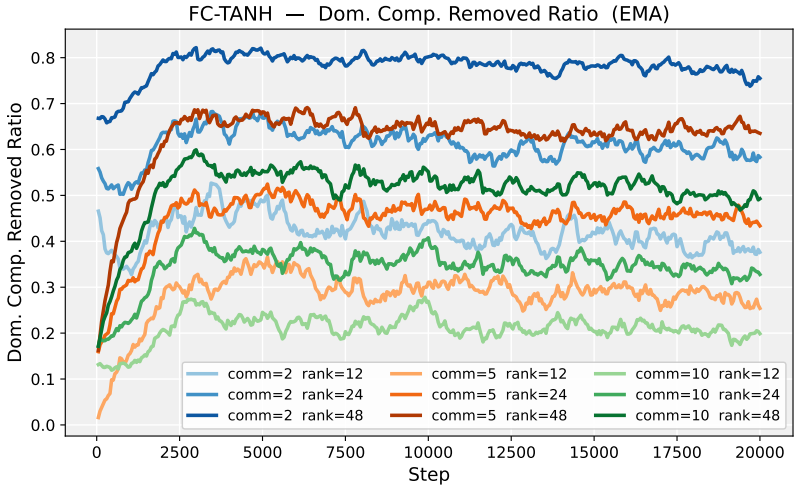


Figure 10: Communication period ablation for FC trained on MNIST-5k

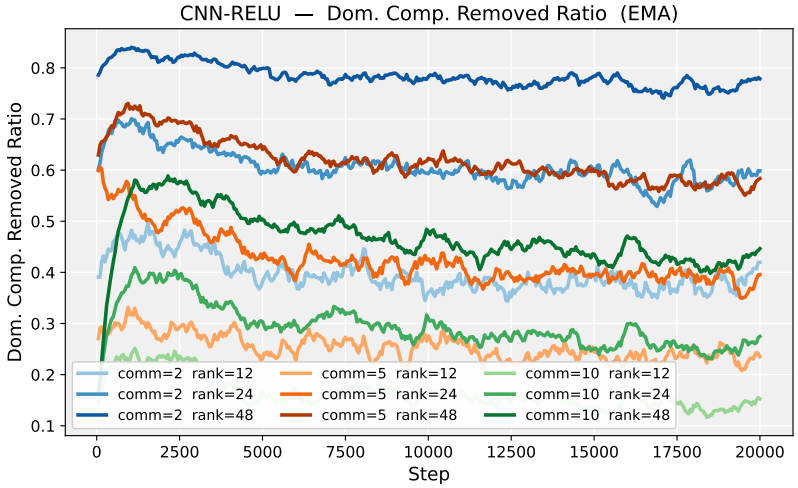


Figure 11: Communication period ablation for CNN trained on CIFAR10-5k

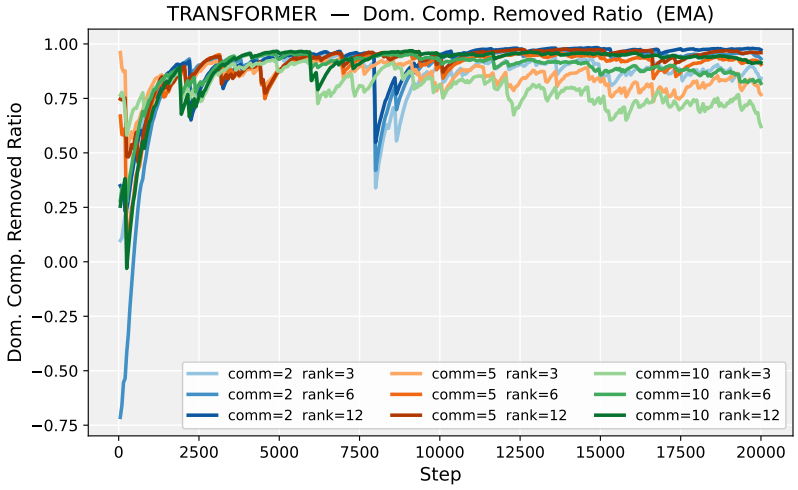


Figure 12: Communication period ablation for Transformer trained on SST2-5k

C.2. Can We Accelerate Optimization by Filtering Out the Dominant Component?

In this section, we operationalize the use of worker–average gap-based subspaces and ask whether training can be accelerated by suppressing the estimated dominant directions and amplifying the orthogonal component. Our focus here is not on generalization or test performance, but rather on training. We evaluate whether the proposed filtering changes the rate at which the training loss decreases by suppressing the estimated optimization steps along the high-curvature directions.

Inspired by the update modification in [19], we propose the following: Using the proxy subspace construction from Appendix A.3, we modify the Local SGD synchronization step as follows. At communication round c, let

$$\bar{p}^c := \frac{1}{M} \sum_{i=1}^{M} p_i^c$$

denote the standard Local SGD average outer displacement. Given the current gap-based basis Q_c, whose columns span the worker-gap proxy subspace $\widehat{\mathcal{S}}_c$, we decompose $\bar{p}^c$ into its component inside the estimated dominant subspace and its orthogonal complement:

$$p^c_{\text{dom}} := Q_c Q_c^\top \bar{p}^c, \qquad p^c_{\text{bulk}} := \bar{p}^c - p^c_{\text{dom}}.$$

We then replace the standard Local SGD synchronization update $\bar{p}^c$ with the filtered update

$$p^c_{\text{filt}} := \alpha p^c_{\text{dom}} + \gamma p^c_{\text{bulk}},$$

and synchronize all workers to

$$\theta^{c+1,0} = \theta^{c,0} + p^c_{\text{filt}}.$$

Here, α controls how strongly we retain the gap-estimated dominant component, while γ controls the scaling of the orthogonal component. Thus, choosing $\alpha < 1$ suppresses directions in $\widehat{\mathcal{S}}_c$, and choosing $\gamma > 1$ amplifies the component orthogonal to the worker-gap subspace. The special case $\alpha = \gamma = 1$ recovers standard Local SGD.

Similar to the experiment setup before, we train FC tanh on MNIST-5k, CNN ReLU on CIFAR-10-5k, and a 2-layer Transformer on SST-2-5k. In all settings, we perform training with Local SGD using $M = 4$ workers and communication period $\tau = 5$. Each stochastic mini-batch has size 50, and training is run for 10,000 local steps, corresponding to 2,000 communication rounds. Since our goal is to evaluate whether the Local SGD update modification leveraging gap-subspaces can improve optimization, we first perform a careful learning-rate sweep for each setting. We select the learning rate η that achieves the lowest final training loss. The sweep values are listed below, with the selected learning rate shown in bold.

- **FC tanh:** $[0.001, 0.005, 0.01, 0.05, 0.1, 0.2, \mathbf{0.3}, 0.4, 0.5]$
- **CNN ReLU:** $[0.001, 0.005, 0.01, \mathbf{0.02}, 0.03, 0.04, 0.05]$
- **Transformer:** $[0.01, 0.02, 0.03, 0.04, 0.06, 0.08, \mathbf{0.1}, 0.2]$

Next, we examine how the training loss evolves under two variants of the filtered Local SGD update: (i) fixing $\gamma = 1.0$ and sweeping α, and (ii) fixing $\alpha = 1.0$ and sweeping γ. For the FC-TANH and CNN-RELU experiments, we set the FIFO buffer capacity of the gap-based proxy subspace to $B = 24$. For the Transformer experiments, we set $B = 6$. Recall that the effective rank of the proxy subspace is determined by the number of retained directions in the buffer span.

i) Fixing $\gamma = 1.0$ and varying α. In this setting, we fix $\gamma = 1.0$ and sweep

$$\alpha \in \{0.0, 0.1, 0.25, 0.5, 1.0, 1.25, 1.5, 1.75, 2.0\}.$$

When $\alpha = 1.0$, the filtered update reduces to standard Local SGD, and we mark this baseline curve in black. We report the results in Figures 13, 14, and 15. In each figure, the left panel shows the training loss curves for $\alpha \leq 1.0$, corresponding to suppression of the gap-estimated dominant component, while the right panel shows the curves for $\alpha \geq 1.0$, corresponding to amplification of this component.

As can be seen from Figures 13, 14, and 15, across all three settings, suppressing the gap-estimated dominant component tends to improve the optimization trajectory relative to standard Local SGD. When $\gamma = 1.0$ and $\alpha < 1.0$, the training loss decreases faster and often reaches a lower final value than the $\alpha = 1.0$ baseline. This effect is especially visible in the Transformer experiment, where reducing the weight on p^c_{dom} leads to a noticeably faster late-stage decrease in training loss. Conversely, amplifying the same component by setting $\alpha > 1.0$ slows down optimization and leads to worse final training loss. The degradation becomes more pronounced as α increases, particularly for the Transformer model, where large α values also introduce visibly higher instability. These results support the view that the worker-gap subspace captures directions that are detrimental to fast optimization, and that suppressing the corresponding component of the Local SGD outer update can accelerate training.

ii) Fixing $\alpha = 1.0$ and varying γ. In this setting, we fix $\alpha = 1.0$ and sweep

$$\gamma \in \{0.0, 0.1, 0.25, 0.5, 1.0, 1.25, 1.5, 1.75, 2.0\}.$$

When $\gamma = 1.0$, the filtered update reduces to standard Local SGD, and we mark this baseline curve in black. We report the results in Figures 16, 17, and 18. In each figure, the left panel shows the

training loss curves for $\gamma \leq 1.0$, corresponding to suppression of the component orthogonal to the gap-estimated dominant subspace, while the right panel shows the curves for $\gamma \geq 1.0$, corresponding to amplification of this orthogonal, bulk component.

As can be observed in Figures 16, 17, and 18, suppressing the estimated bulk component by setting $\gamma < 1.0$ substantially slows down optimization and leads to a much higher final training loss, across all three settings. In particular, small values of γ prevent the training loss from decreasing effectively, suggesting that the orthogonal component p^{c}_{bulk} carries much of the useful descent signal. Conversely, amplifying this component with $\gamma > 1.0$ consistently accelerates optimization and reaches a significantly lower final training loss than standard Local SGD. The improvement is especially clear in the Transformer experiment, where larger γ values lead to a much faster late-stage drop in training loss. At the same time, very large values of γ can introduce additional instability, as seen from the noisier loss curves for the largest amplification factors. Overall, these results complement the α-sweep: suppressing the gap-estimated dominant component and amplifying the orthogonal bulk component both improve the rate at which the training loss decreases.

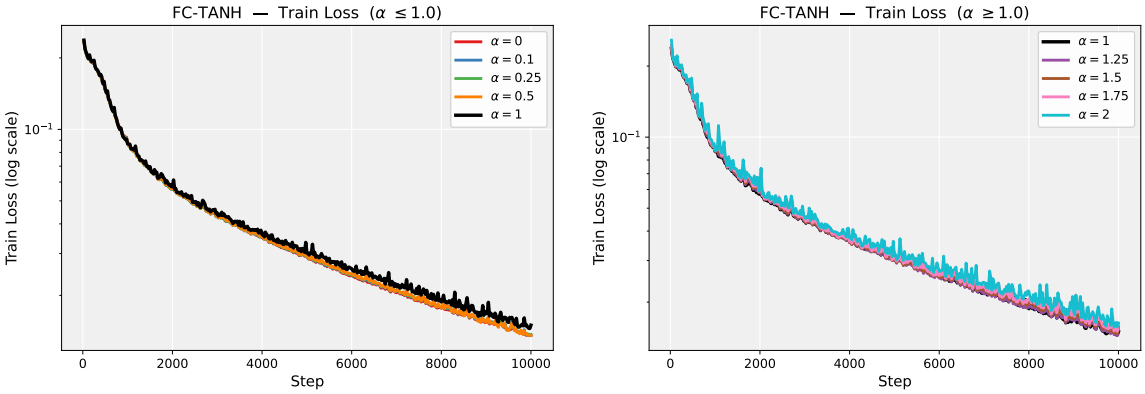


Figure 13: FC TANH model trained on MNIST-5k. We fix $\gamma = 1.0$ and vary α.

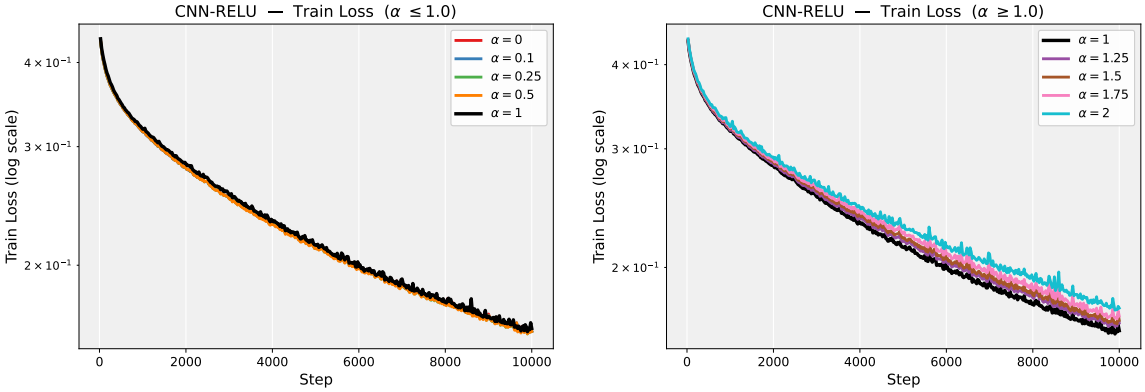


Figure 14: CNN ReLU model trained on CIFAR10-5k. We fix $\gamma = 1.0$ and vary α.

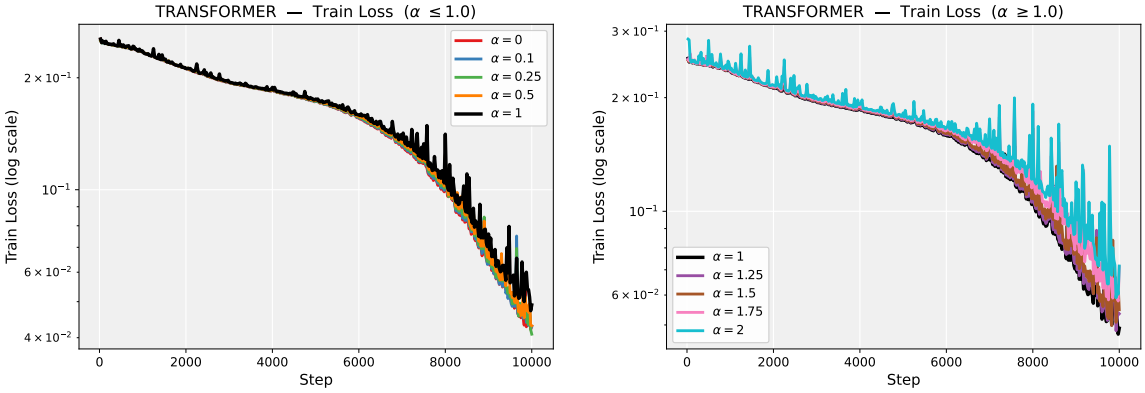


Figure 15: Transformer model trained on SST-5k. We fix $\gamma = 1.0$ and vary α.

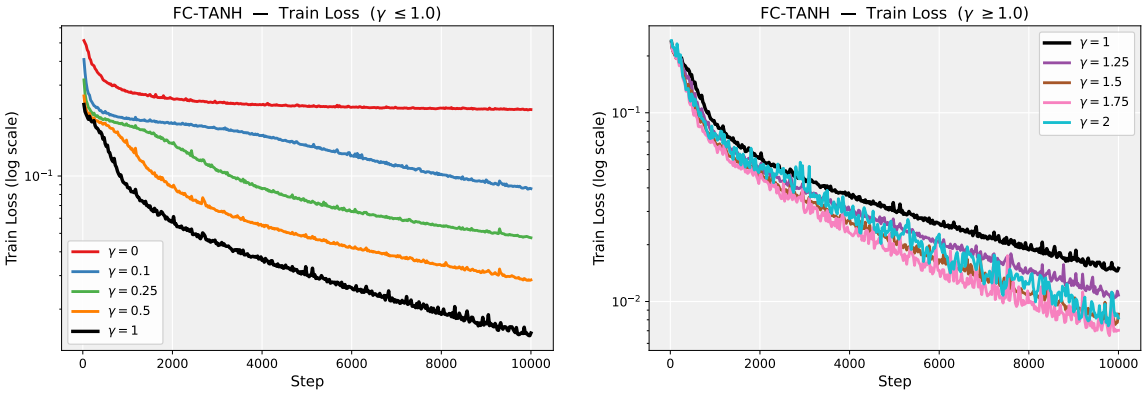


Figure 16: FC TANH model trained on MNIST-5k. We fix $\alpha = 1.0$ and vary γ.

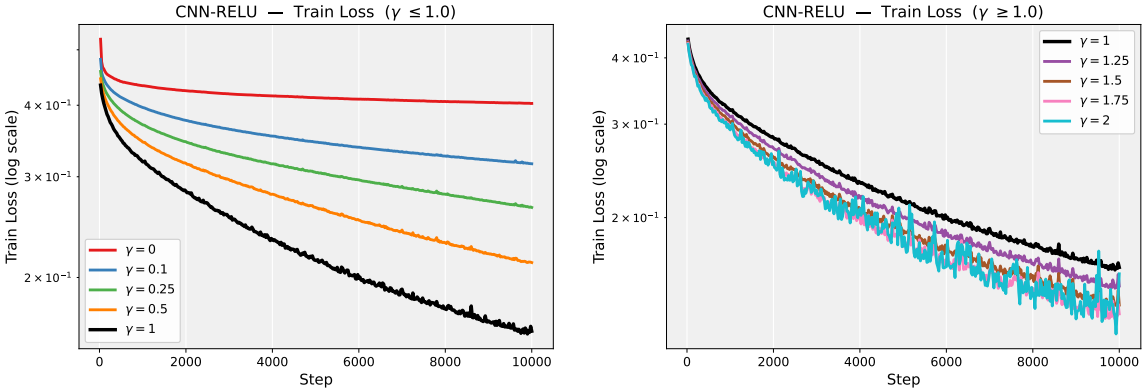


Figure 17: CNN ReLU model trained on CIFAR10-5k. We fix $\alpha = 1.0$ and vary γ.

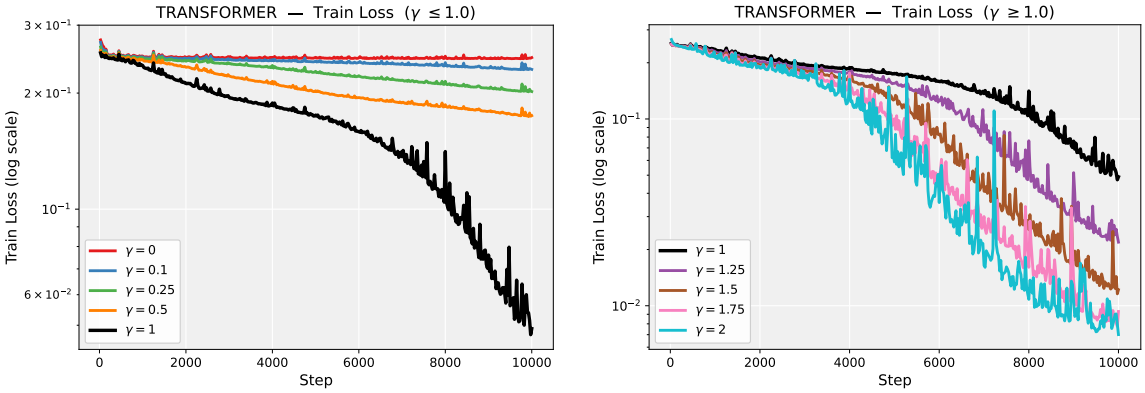


Figure 18: Transformer model trained on SST2-5k. We fix $\alpha = 1.0$ and vary γ.