Regularizing Optimizer Updates via Feasible-Set Projection

Insung Yun DLSTJD0209@HANYANG.AC.KR
Department of Artificial Intelligence, Hanyang University, Seoul, Korea

Sungyoon Lee SUNGYOONLEE@HANYANG.AC.KR
Department of Computer Science, Hanyang University, Seoul, Korea

Abstract

Modern optimizers can produce parameter updates with radial components that increase weight norms during training. Since excessive weight-norm growth is closely related to poor generalization, controlling the geometry of updates provides a natural form of regularization without redesigning the optimizer itself. We propose *Regularizing Optimizer Updates via Feasible-Set Projection*, a simple update constraint applied at the final stage of optimization. Given a raw update from a base optimizer, we project it onto the half-space of directions that are orthogonal to or inward-facing with respect to the current parameter vector, thereby suppressing norm growth while preserving the tangential component of the optimizer's update. The proposed constraint is compatible with general optimizer updates and introduces no additional hyperparameter tuning. In this work, we apply the proposed constraint to Adam and compare it against AdamW, showing that it provides one-step norm control, a bounded parameter-norm trajectory, and improved generalization in our experiments.

1. Introduction

Controlling the magnitude of model parameters has long been recognized as a key factor in generalization. In particular, excessive growth of weight norms is often associated with increased capacity or complexity, which can adversely affect generalization bounds and empirical generalization [1, 12, 14]. To mitigate this issue, modern optimization methods commonly incorporate explicit regularization mechanisms such as ℓ_2 regularization [9, 12] or decoupled weight decay [13]. These approaches penalize the magnitude of parameters and have become standard components in training deep neural networks [6].

However, conventional norm-based regularization mechanisms are agnostic to the geometric relationship between the parameter vector and the optimizer update. They shrink parameters independently of whether the update direction increases or decreases the parameter norm. In addition, standard weight decay applies a constant penalty coefficient uniformly across all parameters, even though different parameters or parameter matrices may benefit from different amounts of regularization [8]. This uniform treatment ignores the structure of the optimization dynamics and often requires careful hyperparameter tuning.

In this work, we propose a simple yet effective alternative that directly constrains the geometry of parameter updates. Our method enforces a constraint based on the relationship between the current parameter vector and the update direction, suppressing only the norm-increasing radial component while preserving the tangential component responsible for optimization. The proposed constraint is applied at the final stage of the update and is therefore compatible, in principle, with a

© I. Yun & S. Lee.

wide range of optimizers, including Adam [11], Lion [5], and Muon [10], without introducing additional hyperparameters. In this work, we instantiate the proposed constraint on top of Adam and compare the resulting method against AdamW, focusing on whether geometric update constraints can provide norm control and generalization benefits without an explicit weight-decay coefficient. This yields a unified, optimizer-agnostic regularization mechanism that controls parameter norm growth through update geometry rather than explicit penalization.

Together, our formulation provides a closed-form projection rule, norm-control guarantees, and ImageNet-1K evidence against a tuned AdamW baseline, positioning feasible-set projection as a simple geometric regularization mechanism for optimizer updates.

1.1. Background

Weight decay is a standard mechanism for controlling parameter norms during training. For stochastic gradient descent, ℓ_2 regularization and weight decay lead to equivalent updates, but this equivalence breaks down for adaptive optimizers because the ℓ_2 term is transformed by adaptive preconditioning. Decoupled weight decay addresses this issue by applying parameter shrinkage separately from the optimizer update [13]. This separation avoids mixing objective and regularization gradients in Adam's moment buffers [2].

Recent methods further make norm control update-dependent or geometry-aware. Cautious Weight Decay (CWD) applies decay only to coordinates where the parameter and optimizer update have aligned signs, yielding a coordinatewise masking rule [4]. AdamO instead uses a radial–tangential decomposition of optimizer dynamics, where radial updates handle norm control while Adam-style adaptive preconditioning is confined to the tangential subspace [3]. Our method follows the same broad goal of geometry-aware norm control, but it does not apply coordinatewise masking or redesign the optimizer dynamics. Instead, it leaves the base optimizer update intact and applies a final-stage feasible-set projection to the full update.

2. Method

We formulate our method as a final-stage modification of the update direction produced by a base optimizer. Rather than changing the loss function, adding an explicit penalty term, or modifying the internal states of the optimizer, our method takes the raw optimizer update as given and replaces it with a feasible update direction. The goal is to preserve the original optimizer update as much as possible while preventing directions that increase the radial component of the parameter vector. We first define this feasible-set projection and derive its closed-form update rule, and then show that the resulting update yields a bounded parameter-norm trajectory.

2.1. Feasible-Set Projection

Let $w_t \in \mathbb{R}^n$ denote the current parameter vector, and let $\Delta_t \in \mathbb{R}^n$ denote the raw update direction produced by a base optimizer. We define the constrained update direction d_t as the closest feasible direction to Δ_t, and update the parameter by

$$w_{t+1} = w_t - \eta_t d_t, \tag{1}$$

where η_t denotes the step-size at iteration t. The constrained direction d_t is obtained by solving

$$d_t = \arg\min_{d \in C(w_t)} \frac{1}{2}\|d - \Delta_t\|^2, \tag{2}$$

where the feasible update set is defined as

$$C(w_t) \triangleq \{d \in \mathbb{R}^n : \langle w_t, d\rangle \geq 0\}. \tag{3}$$

Thus, the base optimizer proposes Δ_t, while our method applies the feasible direction d_t.

The constraint in Eq. (3) excludes update directions whose application would increase the radial component of the parameter vector. Since the actual parameter displacement is $-\eta_t d_t$, the condition $\langle w_t, d_t\rangle \geq 0$ ensures that the displacement is orthogonal to, or inward-facing with respect to, the current parameter vector.

When $\langle w_t, \Delta_t\rangle < 0$, the raw optimizer update is infeasible and is projected onto the boundary of the feasible set,

$$\partial C(w_t) = \{d \in \mathbb{R}^n : \langle w_t, d\rangle = 0\}.$$

The corresponding boundary projection is

$$\Pi_{\partial C(w_t)}(\Delta_t) = \Delta_t - \frac{\langle w_t, \Delta_t\rangle}{\langle w_t, w_t\rangle} w_t. \tag{4}$$

Therefore, the constrained update admits the closed form

$$d_t = \begin{cases} \Delta_t, & \langle w_t, \Delta_t\rangle \geq 0, \\ \Pi_{\partial C(w_t)}(\Delta_t), & \langle w_t, \Delta_t\rangle < 0. \end{cases} \tag{5}$$

Thus, if the raw optimizer update is already feasible, it is left unchanged. If it is infeasible, only the radial component responsible for increasing the parameter norm is removed, while the tangential component is preserved. Algorithm 1 summarizes the resulting final-stage projection applied to a base optimizer. The derivation of Eq. (4)–(5) from Eq. (2) is provided in Appendix A.

2.2. Bounded Parameter-Norm Trajectory

We next show that the proposed feasible-set projection prevents unbounded growth of the parameter norm under bounded-update and square-summable step-size assumptions. Assume that

$$\|\Delta_t\| \leq G \quad \text{for all } t, \qquad \sum_{t=0}^{\infty} \eta_t^2 < \infty. \tag{6}$$

Under these assumptions, the proposed update yields a bounded parameter-norm trajectory:

$$\sup_{T \geq 0} \|w_T\| < \infty. \tag{7}$$

This boundedness result shows that, under the stated assumptions, the proposed constraint prevents the parameter norm from diverging. Since controlling weight norms is closely related to capacity control and generalization in neural networks [1, 12, 14], such norm stabilization provides a theoretical basis for expecting improved generalization. The proof of Eq. (7) is provided in Appendix B.

Algorithm 1: Blockwise Feasible-Set Projection for a Base Optimizer

given: learning rates $\{\eta_t\}_{t\in\mathbb{N}} \subset \mathbb{R}_{>0}$, initial parameters $x_1 = \{x_1^{(\ell)}\}_{\ell=1}^{L}$, base optimizer $\mathcal{O}$
initialize time step $t \leftarrow 1$;
while *not converged* **do**
  $g_t \leftarrow \text{STOCHASTICGRADIENT}(x_t)$;
  $\Delta_t \leftarrow \mathcal{O}(g_t)$; `// raw update direction from the base optimizer`
  for $\ell = 1, \ldots, L$ **do**
    $d_t^{(\ell)} \leftarrow \Delta_t^{(\ell)}$;
    $s_t^{(\ell)} \leftarrow \langle x_t^{(\ell)}, \Delta_t^{(\ell)} \rangle$;
    $r_t^{(\ell)} \leftarrow \langle x_t^{(\ell)}, x_t^{(\ell)} \rangle$;
    if $s_t^{(\ell)} < 0$ *and* $r_t^{(\ell)} > 0$ **then**
      $d_t^{(\ell)} \leftarrow \Delta_t^{(\ell)} - \dfrac{s_t^{(\ell)}}{r_t^{(\ell)}} x_t^{(\ell)}$;
    end
    $x_{t+1}^{(\ell)} \leftarrow x_t^{(\ell)} - \eta_t d_t^{(\ell)}$;
  end
  $t \leftarrow t + 1$;
end
return *optimized parameters* x_t;

3. Experiments

3.1. Experimental Setup

We evaluate the proposed feasible-set projection on ImageNet-1K using a ViT-B/16 backbone [7]. We compare Adam equipped with the proposed feasible-set projection against AdamW. Detailed training settings, including data augmentation, learning-rate schedule, and batch size, are provided in Appendix C.1.

Baseline and proposed method. The AdamW baseline uses learning rate 3×10^{-4} and weight decay 0.1, selected by the hyperparameter sweeps in Appendix C.2. For the proposed method, we apply feasible-set projection to the update produced by Adam at the final update stage, using the same learning rate 3×10^{-4}. No explicit weight decay is used for the proposed method.

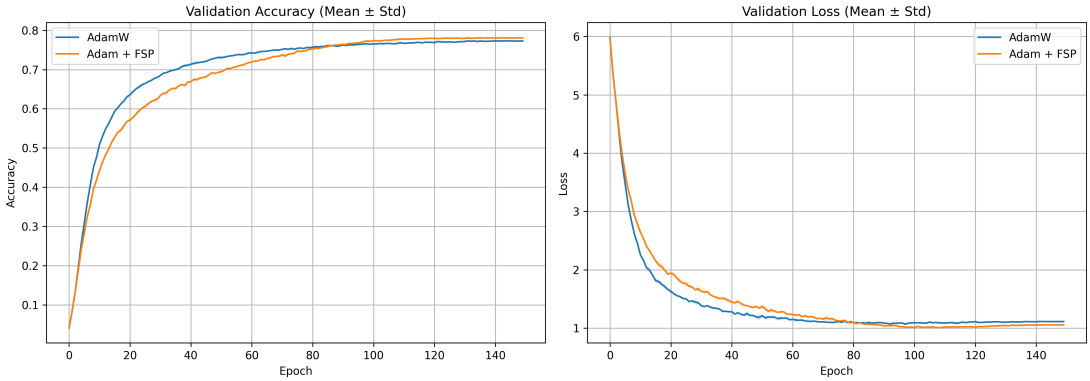
3.2. Results

Table 1 summarizes the comparison between AdamW and Adam with the proposed feasible-set projection, denoted as Adam + FSP. The top table reports the best validation accuracy, final validation accuracy, and mean epoch time, while the bottom panels show the validation accuracy and validation loss curves averaged over three seeds.

Adam + FSP achieves higher validation accuracy than AdamW in both best and final metrics. AdamW improves faster during the early stage of training, whereas Adam + FSP continues to improve in the later stage and eventually reaches higher validation accuracy. The validation loss curve shows a similar late-stage behavior: Adam + FSP achieves lower validation loss after the mid-to-late

Table 1: ImageNet-1K results with ViT-B/16.

Method	Best Acc.	Final Acc.	Epoch Time (s)
AdamW	0.7738 ± 0.0015	0.7729 ± 0.0014	$\mathbf{595.52} \pm \mathbf{0.84}$
Adam + FSP	$\mathbf{0.7816} \pm \mathbf{0.0012}$	$\mathbf{0.7807} \pm \mathbf{0.0012}$	607.96 ± 1.48



training phase. This supports the view that feasible-set projection can act as an effective regularization mechanism by constraining norm-increasing radial components while preserving the optimizer's update structure. Additional norm-trajectory analyses in Appendix C.3 further support this interpretation by showing that Adam + FSP maintains substantially smaller matrix-weight norms than AdamW under the same training setting.

4. Conclusion and Discussion

We proposed *Regularizing Optimizer Updates via Feasible-Set Projection*, a simple final-stage update constraint for controlling parameter-norm growth. Instead of adding an explicit penalty term or modifying the internal dynamics of the base optimizer, the proposed method projects the raw optimizer update onto a feasible set determined by the current parameter vector. This removes norm-increasing radial components while preserving the tangential component of the update. We showed that the resulting update provides one-step norm control and yields a bounded parameter-norm trajectory under the stated assumptions.

Empirically, we instantiated the method on top of Adam and compared it against AdamW on ImageNet-1K with ViT-B/16. The results show that Adam with feasible-set projection achieves higher validation accuracy and lower late-stage validation loss than AdamW in our setting, suggesting that update-geometry-based regularization provides a promising additional mechanism for controlling parameter-norm growth.

There are several directions for future work. First, although the proposed constraint is optimizer-agnostic in formulation, our experiments in this work focus on an Adam-based instantiation. Applying the same feasible-set projection to other optimizers such as Lion and Muon is an important next step. Second, our current method removes the need for an explicit weight-decay coefficient, but it may also be useful in combination with a small amount of weight decay. Future experiments will investigate whether weak additional weight decay further improves generalization when combined with feasible-set projection.

References

[1] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[2] Johan Bjorck, Kilian Q. Weinberger, and Carla P. Gomes. Understanding decoupled and early weight decay. *CoRR*, abs/2012.13841, 2020. URL https://arxiv.org/abs/2012.13841.

[3] Hao Chen, Jh Yuan, and Hanmin Zhang. Decoupled orthogonal dynamics: Regularization for deep network optimizers. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026. URL https://openreview.net/forum?id=gQCstM1Cjj.

[4] Lizhang Chen, Jonathan Li, Kaizhao Liang, Baiyu Su, Cong Xie, Chen Liang, Ni Lao, and qiang liu. Cautious weight decay. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Gwe6gbGng5.

[5] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

[6] Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=YrAxxscKM2.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[8] Jörg K Franke, Michael Hefenbrock, Gregor Koehler, and Frank Hutter. Improving deep learning optimization through constrained parameter regularization. *Advances in Neural Information Processing Systems*, 37:8984–9025, 2024.

[9] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[10] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[14] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1376–1401, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Neyshabur15.html.

Appendix A. Projection Derivation

We derive the closed-form solution of the projection problem for $w_t \neq 0$:

$$d_t = \arg\min_{d \in C(w_t)} \frac{1}{2}\|d - \Delta_t\|^2, \qquad C(w_t) = \left\{ d \in \mathbb{R}^d : \langle w_t, d \rangle \geq 0 \right\}. \tag{8}$$

Define

$$f(d) \triangleq \frac{1}{2}\|d - \Delta_t\|^2, \qquad g(d) \triangleq -\langle w_t, d \rangle. \tag{9}$$

Then the problem can be written as

$$d_t = \arg\min_d f(d) \quad \text{s.t.} \quad g(d) \leq 0. \tag{10}$$

Since f is strongly convex and the feasible set is convex, the minimizer is unique. Moreover, for this convex problem, any point satisfying the KKT conditions is a global minimizer.

The KKT conditions are

$$\nabla f(d) + \lambda \nabla g(d) = d - \Delta_t - \lambda w_t = 0, \tag{i}$$

$$\lambda \geq 0, \tag{ii}$$

$$\lambda g(d) = -\lambda \langle w_t, d \rangle = 0, \tag{iii}$$

$$g(d) = -\langle w_t, d \rangle \leq 0. \tag{iv}$$

From (i), we obtain

$$d = \Delta_t + \lambda w_t. \tag{11}$$

Hence,

$$\begin{aligned} \langle w_t, d \rangle &= \langle w_t, \Delta_t + \lambda w_t \rangle \\ &= \langle w_t, \Delta_t \rangle + \lambda \|w_t\|^2. \end{aligned} \tag{12}$$

Case 1: $\langle w_t, \Delta_t \rangle \geq 0$**.** We have

$$\langle w_t, d \rangle = \langle w_t, \Delta_t \rangle + \lambda \|w_t\|^2 \geq \lambda \|w_t\|^2 \geq 0. \tag{13}$$

By complementary slackness,

$$\lambda \langle w_t, d \rangle = 0. \tag{14}$$

If $\langle w_t, d \rangle > 0$, then complementary slackness immediately implies

$$\lambda = 0. \tag{15}$$

If $\langle w_t, d \rangle = 0$, then

$$0 = \langle w_t, d \rangle \geq \lambda \|w_t\|^2 \geq 0. \tag{16}$$

Hence,

$$\lambda \|w_t\|^2 = 0. \tag{17}$$

For $w_t \neq 0$, this implies $\lambda = 0$. Therefore, in either case,

$$\lambda = 0. \tag{18}$$

Substituting $\lambda = 0$ into $d = \Delta_t + \lambda w_t$, we obtain

$$d_t = \Delta_t. \tag{19}$$

Case 2: $\langle w_t, \Delta_t \rangle < 0$**.** Since

$$\langle w_t, d \rangle = \langle w_t, \Delta_t \rangle + \lambda \|w_t\|^2 \geq 0, \tag{20}$$

it follows that

$$\lambda \geq -\frac{\langle w_t, \Delta_t \rangle}{\|w_t\|^2}. \tag{21}$$

Since $\langle w_t, \Delta_t \rangle < 0$, we have

$$\lambda > 0. \tag{22}$$

Therefore, from $\lambda \langle w_t, d \rangle = 0$, it follows that

$$\langle w_t, d \rangle = 0, \qquad \lambda = -\frac{\langle w_t, \Delta_t \rangle}{\|w_t\|^2}. \tag{23}$$

Substituting this into $d = \Delta_t + \lambda w_t$, we obtain

$$d_t = \Delta_t - \frac{\langle w_t, \Delta_t \rangle}{\|w_t\|^2} w_t = \Delta_t - \frac{\langle w_t, \Delta_t \rangle}{\langle w_t, w_t \rangle} w_t. \tag{24}$$

Thus,

$$\Pi_{\partial C(w_t)}(\Delta_t) = \Delta_t - \frac{\langle w_t, \Delta_t \rangle}{\langle w_t, w_t \rangle} w_t. \tag{25}$$

Combining the two cases, we obtain

$$d_t = \begin{cases} \Delta_t, & \langle w_t, \Delta_t \rangle \geq 0, \\ \Pi_{\partial C(w_t)}(\Delta_t), & \langle w_t, \Delta_t \rangle < 0. \end{cases} \tag{26}$$

Appendix B. Proof of Bounded Parameter-Norm Trajectory

We prove that the proposed feasible-set projection yields a bounded parameter-norm trajectory under the assumptions in Eq. (6). Recall the update rule

$$w_{t+1} = w_t - \eta_t d_t. \tag{27}$$

Taking squared norms gives

$$\begin{aligned} \|w_{t+1}\|^2 &= \|w_t - \eta_t d_t\|^2 \\ &= \|w_t\|^2 - 2\eta_t \langle w_t, d_t \rangle + \eta_t^2 \|d_t\|^2. \end{aligned} \tag{28}$$

Therefore,

$$\|w_{t+1}\|^2 - \|w_t\|^2 = -2\eta_t \langle w_t, d_t \rangle + \eta_t^2 \|d_t\|^2. \tag{29}$$

Since $d_t \in C(w_t)$ by construction, we have

$$\langle w_t, d_t \rangle \geq 0. \tag{30}$$

Also, since $\eta_t > 0$, it follows that

$$-2\eta_t \langle w_t, d_t \rangle \leq 0. \tag{31}$$

Hence,

$$\|w_{t+1}\|^2 - \|w_t\|^2 \leq \eta_t^2 \|d_t\|^2. \tag{32}$$

Moreover, from the closed-form update in Eq. (26), the projected update satisfies

$$\|d_t\| \leq \|\Delta_t\|. \tag{33}$$

Indeed, if $\langle w_t, \Delta_t \rangle \geq 0$, then $d_t = \Delta_t$. Otherwise, $d_t = \Pi_{\partial C(w_t)}(\Delta_t)$ is obtained by removing the component of Δ_t parallel to w_t, and hence its norm cannot exceed that of Δ_t. Thus,

$$\|w_{t+1}\|^2 - \|w_t\|^2 \leq \eta_t^2 \|\Delta_t\|^2. \tag{34}$$

Equivalently,

$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + \eta_t^2 \|\Delta_t\|^2. \tag{35}$$

Assume that $\|\Delta_t\| \leq G$ for all t. Then

$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + G^2 \eta_t^2. \tag{36}$$

Iterating this inequality from $t = 0$ to $T - 1$ gives

$$\begin{aligned} \|w_T\|^2 &\leq \|w_0\|^2 + G^2 \sum_{t=0}^{T-1} \eta_t^2 \\ &\leq \|w_0\|^2 + G^2 \sum_{t=0}^{\infty} \eta_t^2. \end{aligned} \tag{37}$$

By the square-summability assumption,

$$\sum_{t=0}^{\infty} \eta_t^2 < \infty. \tag{38}$$

Therefore,

$$\|w_T\|^2 \leq \|w_0\|^2 + G^2 \sum_{t=0}^{\infty} \eta_t^2 < \infty \qquad \text{for all } T. \tag{39}$$

Consequently,

$$\sup_{T \geq 0} \|w_T\| < \infty. \tag{40}$$

This proves the bounded parameter-norm trajectory.

Appendix C. Experimental Details and Hyperparameter Selection

C.1. Training Details

Training protocol. All models are trained for 150 epochs with a global batch size of 1024. We use a cosine learning-rate schedule with 7 warmup epochs. The learning rate is set to 3×10^{-4} for both methods, with Adam hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-9}$. All results are reported over three random seeds, $\{0, 1, 2\}$.

Data augmentation and regularization. For training, we use random resized cropping, random horizontal flipping, and ImageNet normalization. For validation, images are resized to 256, center-cropped to 224×224, and normalized using the same ImageNet statistics. We additionally use Mixup with $\alpha = 0.8$, CutMix with $\alpha = 1.0$, label smoothing with coefficient 0.1, and stochastic depth with drop-path rate 0.1. Dropout is set to 0.0.

C.2. Hyperparameter Selection

Before the final comparison, we perform a learning-rate sweep and a weight-decay sweep for the AdamW baseline. We first sweep the learning rate over $\{3 \times 10^{-4}, 5 \times 10^{-4}, 7 \times 10^{-4}\}$ without weight decay and select 3×10^{-4} according to validation accuracy. Using this learning rate, we then sweep the AdamW weight decay and select 0.1 as the final baseline value.

Table 2: Learning-rate sweep for the AdamW baseline. The best learning rate is selected according to validation accuracy.

Learning rate	Final val. loss	Final val. acc.	Best val. loss	Best val. acc.
3.0×10^{-4}	**1.0806**	**0.7648**	**1.0792**	**0.7650**
5.0×10^{-4}	1.0971	0.7628	1.0956	0.7630
7.0×10^{-4}	1.0988	0.7591	1.0963	0.7593

Table 3: First weight-decay sweep for AdamW with learning rate 3×10^{-4}.

Weight decay	Final val. loss	Final val. acc.	Best val. loss	Best val. acc.
1.0×10^{-2}	1.0760	0.7694	1.0722	0.7697
3.0×10^{-2}	1.0855	0.7656	1.0806	0.7667
5.0×10^{-2}	1.0742	0.7698	1.0615	0.7702
7.0×10^{-2}	1.0642	0.7702	**1.0489**	0.7704
1.0×10^{-1}	**1.0548**	**0.7724**	1.0493	**0.7729**

Table 4: Second weight-decay sweep for AdamW with learning rate 3×10^{-4}.

Weight decay	Final val. loss	Final val. acc.	Best val. loss	Best val. acc.
1.0×10^{-1}	1.0455	**0.7748**	1.0407	**0.7748**
1.2×10^{-1}	1.0492	0.7733	1.0432	0.7733
1.5×10^{-1}	**1.0282**	0.7730	**1.0256**	0.7733

Across the two weight-decay sweeps, weight decay 0.1 achieves the highest validation accuracy and is therefore used for the AdamW baseline in the final comparison.

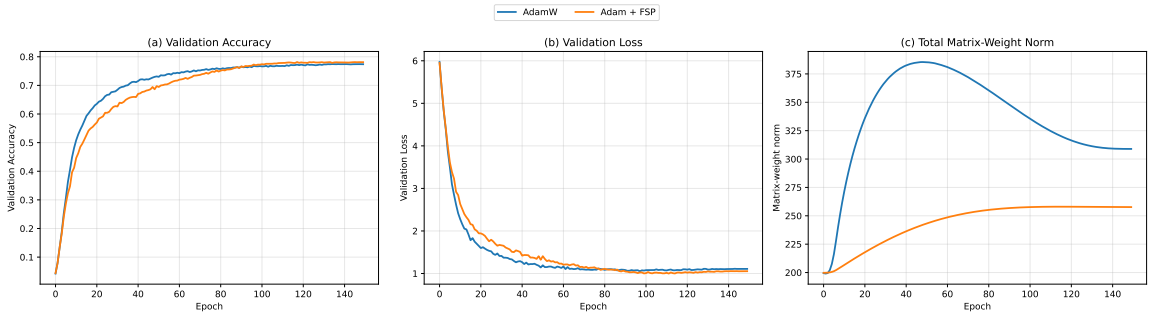


Figure 1: Single-seed training dynamics and total matrix-weight norm trajectory under the same ImageNet-1K ViT-B/16 setting as the main experiment. Adam + FSP maintains a substantially smaller matrix-weight norm than AdamW after the early training phase, despite using no explicit weight decay, while achieving competitive late-stage validation performance.

C.3. Norm-Control Behavior

To further examine the norm-control effect of feasible-set projection, we track the Frobenius norm of trainable matrix-valued weights under the same ImageNet-1K ViT-B/16 setting as the main experiment. Figure 1 reports the single-seed training dynamics and total matrix-weight norm trajectory. Adam + FSP maintains a substantially smaller matrix-weight norm than AdamW after the early training phase, despite using no explicit weight decay. This supports the interpretation that feasible-set projection suppresses norm-increasing radial components in practice.

Figure 2 further reports the block-wise norm ratio between Adam + FSP and AdamW. The ratios are mostly below one after the early training phase, indicating that the norm-control effect is distributed across Transformer blocks rather than being driven by a single layer.

LLM Usage

Large language models were used only as writing aids to improve clarity, grammar, and organization. The theoretical ideas, proof strategy, experimental design, and interpretation of results were developed and directed by the authors. All technical content and final manuscript decisions were reviewed and approved by the authors.

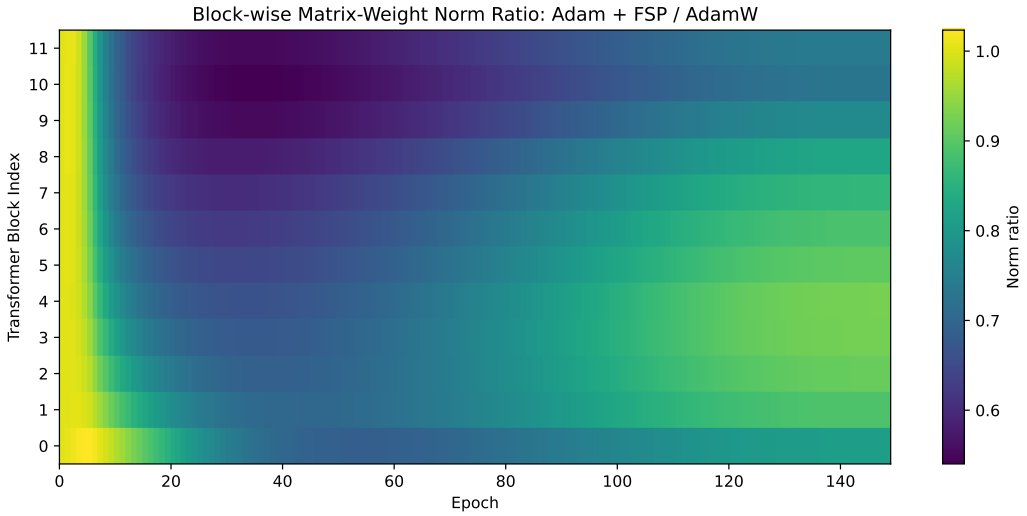


Figure 2: Block-wise matrix-weight norm ratio between Adam + FSP and AdamW. Each entry shows the ratio $\|W_{i,t}^{\mathrm{FSP}}\|_F / \|W_{i,t}^{\mathrm{AdamW}}\|_F$ for Transformer block i at epoch t. Values below one indicate smaller block-wise norms under Adam + FSP. The ratios are mostly below one after the early training phase, suggesting that the norm-control effect is broadly distributed across Transformer blocks.