Greedy Alignment Principle for Optimizer Hyperparameter Selection

Jaerin Lee & Kyoung Mu Lee
Computer Vision Lab, ASRI
Seoul National University
Seoul 08826, Korea
{ironjr,kyoungmu}@snu.ac.kr

Abstract

Recent works have shown that gradient–update alignment is a powerful signal for modulating optimizer updates, often leading to faster training. We promote this update-wise heuristic as a mathematically grounded principle for selecting and tuning optimizer hyperparameters. By treating gradients and updates as signals and an optimizer as a causal filter that maps between them, we formulate optimizer selection as maximizing the expected drop rate in loss over a prescribed family of optimizers. We show that this objective is exactly the inner product between the optimizer filter and the gradient autocorrelation, and prove that a greedy optimum exists and has a stability bound under perturbations of the estimated gradient statistics. Specializing in momentum-based optimizers, the theory yields simple dynamic momentum selection rules for both SGD+Momentum and Adam/AdamW. Experiments across image classification, language model fine-tuning, and vision transformer fine-tuning show that the resulting dynamic momentum rules match or improve upon the best fixed hyperparameters found via manual sweeps, reducing the need for exhaustive momentum sweeps.

1 Introduction

Gradient-based learning has been the *de facto* standard in machine learning practice. Over the past decades, first-order methods have matured into a rich collection of optimizers. Dozens of gradient-based optimizers have been proposed in the literature [52, 36, 44, 11, 68, 21, 41, 10, 42, 49, 60] and are widely used in practice, improving upon the baseline SGD [52] in their specialized tasks. Although their philosophies and design choices are distinct, many of them share the same principle of momentum-based filtering of incoming gradient streams [39] to stabilize learning dynamics. The momentum hyperparameter β is a crucial design choice that governs the trade-off between stability and learning speed. However, the selection of this hyperparameter is highly dependent on the underlying task and the type of optimizers used. Still, the choice is largely left to the practitioner's intuition and trial-and-error.

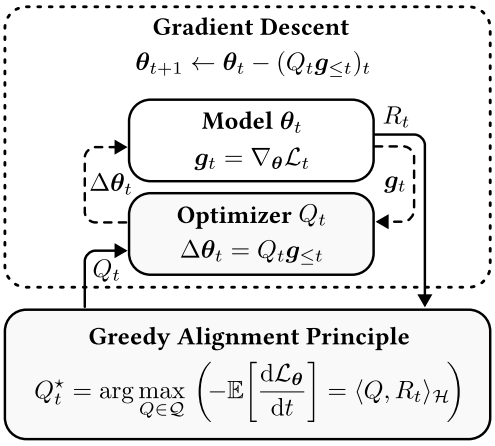


Figure 1: Just as optimizers train models by parameter updates $\boldsymbol{u}$, models can also select an optimizer from a candidate set $\mathcal{Q}$ using the gradient statistics R.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To address this problem, we consider a different line of work that post-processes the parameter updates produced by optimizers. Notably, Cautious optimizer [41] selectively applies the parameter update components based on the sign of coordinate-wise gradient-update products. MGUP [10] further uses momentum-gradient alignment to reweight the update components. These works empirically demonstrate that the *instantaneous alignment* $\boldsymbol{g}^\top \boldsymbol{u}$ between gradients $\boldsymbol{g}$ and parameter updates $\boldsymbol{u}$ is an informative signal for assessing, controlling, and improving the learning dynamics of existing gradient-based optimizers [52, 36, 44, 11, 34].

This update-wise greedy alignment strategy can be formulated as follows. In gradient-based learning, we minimize a loss $\mathcal{L}(\boldsymbol{\theta})$ with respect to the parameter $\boldsymbol{\theta}$ using its gradient $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}$. At each training step t, the parameter $\boldsymbol{\theta}$ is updated by $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \boldsymbol{u}_t$. The heuristics of greedy alignment can then be formulated into an optimization problem:

$$\boldsymbol{u}_t^\star \;=\; \arg\max_{\boldsymbol{u}_t \in \mathcal{U}_t} \boldsymbol{g}_t^\top \boldsymbol{u}_t, \tag{1}$$

where $\mathcal{U}_t$ is the set of parameter updates proposed in step t. Interestingly, by the chain rule, this problem reduces to maximizing the *instantaneous loss drop rate*:

$$\boldsymbol{g}_t^\top \boldsymbol{u}_t \;=\; -\nabla_{\boldsymbol{\theta}} \mathcal{L}^\top \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} \;=\; -\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}t}. \tag{2}$$

This partially justifies the community wisdom [41, 10] built on greedy alignment of updates.

Despite their elegance, the idea of greedily maximizing alignment has been applied only by modifying the outputs of fixed optimizers. We promote this idea into a mathematically grounded principle for selecting and tuning the optimizers themselves. Adopting the signal processing analogy [39], we treat gradients $\boldsymbol{g}_t$ as a time-varying signal. Similarly, the parameter update $\boldsymbol{u}_t$ is a function of the causal gradient stream $\boldsymbol{g}_{\leq t}$. Then, the optimizer Q can be regarded as a *signal processing filter* $Q : \boldsymbol{g} \mapsto \boldsymbol{u}$, rather than an algorithmic procedure, mapping the gradient stream $\boldsymbol{g}$ to the update stream $\boldsymbol{u}$:

$$\boldsymbol{u}_t \;=\; (Q\boldsymbol{g})_t. \tag{3}$$

This filter view of the optimizer Q enables us to apply the greedy alignment principle at the operator-level. We further specialize to stable causal linear filters Q, which includes SGD [52] with momentum as a canonical special case. Then, the expected alignment becomes a Hilbert inner product between the optimizer filter Q and the gradient autocorrelation $R_{t,k} = \mathbb{E}[\boldsymbol{g}_t \boldsymbol{g}_{t-k}^\top]$, as elaborated in Section 2.1,

$$\mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{u}_t] \;=\; \langle Q, R \rangle_{\mathcal{H}}. \tag{4}$$

In other words, update-wise *expected* alignment is equivalent to *alignment between the optimizer filter Q and the gradient statistics R*. Allowing Q to be time-varying and input-adaptive, this framework also handles preconditioned methods such as AdamW [44].

Lifting the greedy alignment principle to this filter viewpoint unlocks a theoretical toolbox, especially useful in regimes where the *global optimality criterion is nearly impossible to achieve*. The heuristic practice of selecting optimizer hyperparameters ϕ is now mathematically grounded as an optimization problem within a prescribed family of optimizers $Q \in \mathcal{Q}_\phi$, which is also equivalent to maximizing the instantaneous expected loss drop rate. As illustrated in Figure 1, this dual problem of rate maximization coexists with the primary problem of loss minimization. By merging equations 1 and 4, we reveal two design factors that determine the best optimizer according to the greedy alignment principle: (1) the *hyperparameterized family of optimizers* $Q \in \mathcal{Q}_\phi$, and (2) the *gradient autocorrelation* R. Therefore, the problem reduces further to selecting the hyperparameter ϕ based on the gradient statistics R. In Section 2.2, we prove that this greedy optimum is well-defined, attained under compactness assumptions, and characterized by the support function of the candidate family. Section 3 focuses on momentum-based optimizers, a large family of optimizers of our special interest [52, 36, 44]. Section 4 discusses practical concerns in applying the theory, and experiments in Section 5 support the resulting dynamic selection rules in this focused case.

In summary, our contributions are as follows:

- We lift update-wise greedy alignment rules to an *optimizer-level principle* by modeling a first-order optimizer as a causal filter and formulate optimizer selection as maximizing the expected first-order loss drop rate $-\mathrm{d}\mathcal{L}/\mathrm{d}t$ against the gradient autocorrelation R.
- We prove that, for causal linear filters and compact candidate families, this problem is well-defined, attains an optimum, and is stable under perturbations of the estimated gradient statistics.

- We show that optimizer hyperparameterization reduces this problem into scoring hyperparameters, yielding a principled basis for dynamic hyperparameter selection.
- Applying this framework to momentum-based optimizers [52, 36, 44], we derive dynamic selection rules that exhibit matched or improved results compared with manually tuned fixed-momentum baselines, reducing the need for expensive hyperparameter sweeps.

2 Gradient-update alignment to filter-statistics alignment

2.1 Problem statement

As mentioned in the previous section, we formalize the *greedy alignment principle* for gradient-based optimizers, starting from the update-wise principle in equation 1. Let an optimizer Q be a *causal filter* that translates a gradient stream $\boldsymbol{g}_{\leq t}$ into an update $\boldsymbol{u}_t = -\mathrm{d}\boldsymbol{\theta}/\mathrm{d}t$. We then choose the optimizer according to the following maximization problem

$$Q^\star \;=\; \arg\max_{Q\in\mathcal{Q}} \mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{u}_t], \qquad \mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{u}_t] \;=\; \mathbb{E}[\boldsymbol{g}_t^\top\, (Q\boldsymbol{g})_t] \;=\; -\mathbb{E}\left[\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}t}\right] \;=:\; P, \tag{5}$$

where $\mathcal{Q}$ is the candidate set of optimizers. This lifts the update rule of equation 1 to an optimizer-level selection criterion within the prescribed family of optimizers $\mathcal{Q}$. For simplicity of notation, define the *learning power* $P := -\mathbb{E}[\mathrm{d}\mathcal{L}/\mathrm{d}t]$ as the expected speed of loss drop. Note that we are deriving the theory in the continuous-time training step t for compactness. The validity of the results in discrete-time is straightforward and is provided in Section 3.2.

Consider a linear dynamic optimizer Q operating as a causal convolution

$$\boldsymbol{u}_t \;=\; (Q\boldsymbol{g})_t \;=\; (Q_t * \boldsymbol{g}_{\leq t})_t \;=\; \sum_{k=0}^{\infty} Q_{t,k}\, \boldsymbol{g}_{t-k}. \tag{6}$$

This formulation generalizes first-order optimizers, especially those with momentum terms. Define the *gradient autocorrelation* as $R_{t,k} := \mathbb{E}[\boldsymbol{g}_t\, \boldsymbol{g}_{t-k}^\top]$, for $k \geq 0$. Then Q and R are both elements of an admissible Hilbert space $\mathcal{H}$ of square-summable causal sequences

$$\mathcal{H} \;=\; \Big\{A = (A_k)_{k\geq 0} \;:\; \sum_{k\geq 0} \mathrm{Tr}(A_k^\top A_k) < \infty\Big\}, \qquad \langle A, B\rangle_{\mathcal{H}} \;=\; \sum_{k\geq 0} \mathrm{Tr}(A_k^\top B_k), \tag{7}$$

where we can define an inner product between optimizer transfer function Q and gradient autocorrelation R. This is exactly the learning power P in equation 5.

Proposition 2.1 (Learning power is inner product)**.** [proof] *Assume that (A1) the gradient stream $\boldsymbol{g}_t$ has finite and uniformly bounded second moments for all t: $\sup_{t\in\mathbb{Z}} \mathbb{E}\|\boldsymbol{g}_t\|^2 < \infty$; (A2) the optimizer filter Q_t is absolutely summable in operator norm: $\sum_{k=0}^{\infty} \|Q_{t,k}\|_{op} < \infty$; and (A3) the autocorrelation sequence $R_t = \{R_{t,k}\}_{k\geq 0}$ belongs to $\mathcal{H}$. Then for every t,*

$$P_t(Q) \;=\; \mathbb{E}\big[\boldsymbol{g}_t^\top \boldsymbol{u}_t\big] \;=\; \sum_{k=0}^{\infty} \mathrm{Tr}\big(Q_{t,k}^\top R_{t,k}\big) \;=\; \langle Q_t, R_t\rangle_{\mathcal{H}}. \tag{8}$$

Note that the assumptions (A1-3) generally hold for any stable machine learning system. The filter-level principle in equation 5 can be rewritten into the following maximization problem:

$$Q_t^\star \;=\; \arg\max_{Q\in\mathcal{Q}} \langle Q, R_t\rangle_{\mathcal{H}}. \tag{♠}$$

As long as the set of candidate optimizers $\mathcal{Q}$ is convex and bounded, selecting the best optimizer under the greedy alignment principle (♠) is a *convex optimization problem with a linear objective*, of which we have a rich understanding [8]. Note that $\mathcal{Q}$ should be bounded to avoid arbitrarily large learning rates. Furthermore, the optimal optimizer $Q_t^\star$ is a function of the (1) candidate set $\mathcal{Q}$ and the (2) gradient statistics R_t. Note the subscript t. This formulation allows us to dynamically adapt the optimizer Q with respect to time-varying R_t.

2.2 General solutions

We start by showing that the solution of problem (♠) is indeed attainable. The following definitions are required for the theory's formalism. Let a *Minkowski gauge* $\gamma_{\mathcal{Q}}(Q)$ and a *polar set* $\mathcal{Q}^\circ$ be

$$\gamma_{\mathcal{Q}}(Q) \;\coloneqq\; \inf\{c > 0 : Q \in c\mathcal{Q}\}, \qquad \mathcal{Q}^\circ \;\coloneqq\; \{R \in \mathcal{H} \mid \sup_{Q\in\mathcal{Q}} \langle Q, R\rangle_{\mathcal{H}} \le 1\}, \tag{9}$$

In simpler terms, $\gamma_{\mathcal{Q}}(Q)$ measures the minimum scaling of a convex set $\mathcal{Q}$ to contain Q, and $\mathcal{Q}^\circ$ is a set of gradient statistics R that upper-bounds the inner product by one. Then we can define a *symmetrized polar gauge* $\|\cdot\|^{\mathrm{sym}}_{\mathcal{Q}^\circ}$ as

$$\|R\|^{\mathrm{sym}}_{\mathcal{Q}^\circ} \;\coloneqq\; \max\{\gamma_{\mathcal{Q}^\circ}(R),\, \gamma_{\mathcal{Q}^\circ}(-R)\}. \tag{10}$$

We are now ready to introduce our first theorem. The common subscript t is omitted for brevity.

Theorem 2.2 (Greedy optimal first-order optimizers)**.** [proof] *Let $\mathcal{H}$ be a Hilbert space. Let $\mathcal{Q} \subset \mathcal{H}$ be a compact nonempty convex set of causal optimizer filters with $0 \in \mathcal{Q}$. Then for every $R \in \mathcal{H}$,*

(i) (Existence): *The maximum $P^\star(R)$ of problem (♠) is attained, sublinear, and finite everywhere.*

(ii) (Construction): *Any optimal optimizer $Q^\star$ is a subgradient of $P^\star = \gamma_{\mathcal{Q}^\circ}$ at R: $Q^\star \in \partial_R P^\star(R)$. If the maximizer is unique, $P^\star$ is Gâteaux differentiable at R and $Q^\star = \nabla_R P^\star(R)$.*

(iii) (Lipschitz continuity): *The estimated optimal learning power under surrogate gradient statistics $\hat{R}$ is bounded by $|P^\star(R) - P^\star(\hat{R})| \le \|R - \hat{R}\|^{\mathrm{sym}}_{\mathcal{Q}^\circ}, \forall R, \hat{R} \in \mathcal{H}$.*

Its proof is in Appendix E. From Theorem 2.2, we guarantee that: (1) the optimizer selection problem under the principle of greedy alignment is well-defined, and (2) there exists the optimal optimizer $Q^\star_t$, and (3) Lipschitz continuity gives us a stability bound of the optimal learning power $P^\star$ when we use the estimated gradient statistics $\hat{R}$ (for example, autocorrelations from a mini-batch stream) instead of the true statistics R. It should be noted that the optimal solution to the problem (♠) is constructed from the candidate family $\mathcal{Q}$ and the gradient statistics R in the form $Q^\star = \partial_R \gamma_{\mathcal{Q}^\circ}(R)$. The exact optimality is constructible for various families of optimizers, as we elaborate in Appendix C.

Specifically, the candidate family $\mathcal{Q}$ can be characterized by a set of hyperparameters ϕ, (e.g., EMA momentum β). These hyperparameters later describe the greedy optimal optimizer $Q^\star$ as well. In other words, the optimizer-selection principle in equation 5 is reducible to a mathematical framework to choose *optimizer hyperparameters*. We make this formal. Define a closed convex hull $\mathrm{cl\,co}(S)$ of a set $S \subset \mathcal{H}$ as the closure of all finite convex combinations

$$\mathrm{cl\,co}(S) \;\coloneqq\; \overline{\mathrm{co}(S)}, \quad \text{where} \quad \mathrm{co}(S) \;\coloneqq\; \Big\{ \sum_{i=1}^{m} \lambda_i s_i : m < \infty, s_i \in S, \lambda_i \ge 0, \sum_{i=1}^{m} \lambda_i = 1 \Big\}. \tag{11}$$

This allows us to study hyperparameterized settings as a convex optimization framework.

Corollary 2.3 (Family-hull reduction)**.** [proof] *Let Φ be a nonempty set of hyperparameters and $\mathcal{F}_\Phi = \{Q_\phi : \phi \in \Phi\} \subset \mathcal{H}$ a bounded family of optimizer filters. Define $\mathcal{Q}_\Phi \coloneqq \mathrm{cl\,co}(\mathcal{F}_\Phi \cup \{0\})$. Then for every $R \in \mathcal{H}$,*

$$\sup_{Q\in\mathcal{Q}_\Phi} \langle Q, R\rangle_{\mathcal{H}} \;=\; \Big(\sup_{\phi\in\Phi} \mathbb{E}[\boldsymbol{g}_t^\top (Q_\phi \boldsymbol{g})_t] \Big)_+, \tag{12}$$

where $(x)_+ \coloneqq \max\{x, 0\}$. If Φ is finite, the supremum is a maximum.

This maximization of the expected alignment $\mathbb{E}[\boldsymbol{g}_t^\top (Q_\phi \boldsymbol{g})_t]$ over the governing hyperparameter ϕ is an *exact reduction* of the original problem (♠), not a heuristic approximation. The learning power P is linear in Q. Provided that the set of hyperparameters Φ is compact and the mapping $\phi \mapsto Q_\phi$ is continuous, this convex program on a convex set $\mathcal{Q}_\Phi$ therefore attains its maximum at its extreme points, namely in $\mathcal{F}_\Phi \cup \{0\}$. We can therefore apply Theorem 2.2 to a hyperparameterized family of optimizers. This allows us to treat various existing families of optimizers as mathematically grounded objects. In the main text, we focus on linear causal optimizers for simplicity, but the theory can be generalized to nonlinear causal optimizers, as we elaborate in Appendix B.

3 Momentum families and local validity

3.1 Hyperparameter selection for momentum-based optimizers

We instantiate the general theory on the most commonly used nontrivial family of optimizers: momentum-based ones. Many practical optimizers, including SGD+Momentum [52] and Adam/AdamW [36, 44], are special cases of this class. This class of optimizers is particularly interesting because a one-pole IIR filter can represent the EMA momentum:

$$\mathcal{Q}_{\mathrm{1p}} \coloneqq \{Q_{\beta,k} \;=\; \eta\beta^k I : \eta \geq 0, \beta \in \mathcal{B} \subset [0,1)\}, \qquad (Q_\beta \boldsymbol{g})_t \;=\; \eta \sum_{k=0}^{\infty} \beta^k \boldsymbol{g}_{t-k}, \tag{13}$$

where $\mathcal{B}$ is compact, e.g., $\mathcal{B} = [0, \bar{\beta}]$ or a finite set of candidates. The family of 1-pole optimizer filters $\mathcal{Q}_{\mathrm{1p}}$ is a mathematical description of the family of EMA momentum-based first-order optimizers. Define the unnormalized momentum $\boldsymbol{m}_{\beta,t} \coloneqq \beta \boldsymbol{m}_{\beta,t-1} + \boldsymbol{g}_t$ at time t with the momentum hyperparameter β. If we constrain the family to be norm-bounded, i.e., $\mathcal{Q}_{\mathrm{H}}(B) \coloneqq \{Q : \|Q\|_{\mathcal{H}} \leq \sqrt{B}\}$ and $\mathcal{Q}_{\eta,\beta} \coloneqq \mathcal{Q}_{\mathrm{1p}} \cap \mathcal{Q}_{\mathrm{H}}(B)$, then we get the SGD+Momentum optimizer with optimal hyperparameters.

Theorem 3.1 (Greedy optimal SGD+Momentum)**.** [proof] *Consider the norm-bounded family $\mathcal{Q}_{\eta,\beta} = \mathcal{Q}_{lp} \cap \mathcal{Q}_{\mathrm{H}}(B)$ of optimizer filters. Then problem (♠) under the prescribed family $\mathcal{Q}_{\eta,\beta}$ reduces to the* SGD *optimizer with* greedy optimal momentum hyperparameter*:*

$$\beta_t^\star \;=\; \arg\max_{\beta \in \mathcal{B}} \sqrt{1-\beta^2}\; \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{m}_{\beta,t}], \tag{14}$$

The score in Theorem 3.1 acts as a normalized probe of the gradient autocorrelation. Note that the norm budget B disappears. The factor $\sqrt{1-\beta^2}$ comes from the norm of the Hilbert space $\mathcal{H}$ that defines the inner product between the optimizer Q_t and the statistics R_t. This factor additionally removes trivial preferences for filters with either no memory ($\beta = 0$) or infinite memory ($\beta = 1$). The following corollary is a neat theoretical result that uncovers the connection between the momentum hyperparameter β and the gradient statistics, enriching the meaning of *filter–statistics alignment*.

Corollary 3.2 (One-pole gradients recover one-pole momentum)**.** [proof] *Assume that the autocorrelation of the gradient stream $\boldsymbol{g}$ has the one-pole scalar trace autocorrelation function:*

$$r_k \;=\; \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{g}_{t-k}] \;=\; r_0 \rho^k, \qquad r_0 > 0, \quad 0 \leq \rho < 1. \tag{15}$$

Then the SGD+Momentum score in Theorem 3.1 becomes

$$\sqrt{1-\beta^2}\, \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{m}_{\beta,t}] \;=\; r_0 \frac{\sqrt{1-\beta^2}}{1-\beta\rho}. \tag{16}$$

The maximum is attained where the gradient pole ρ is aligned with the momentum pole β: $\beta^\star = \rho$. Moreover, if the range is restricted to $\beta \in [0, \bar{\beta}]$ with $0 < \bar{\beta} < 1$, then $\beta^\star = \max\{0, \min\{\bar{\beta}, \rho\}\}$.

In other words, the momentum hyperparameter β is more than an arbitrary smoothing coefficient. In this one-pole case, its greedy optimum $\beta^\star$ exactly recovers the temporal coherence of the gradients $\boldsymbol{g}$.

Furthermore, we can easily extend Theorem 3.1 to widely used preconditioned optimizers such as Adam [36]. Let normalized first and second moments $\boldsymbol{\mu}_{\beta_1,t}$ and $\boldsymbol{v}_{\beta_2,t}$ as:

$$\boldsymbol{\mu}_{\beta_1,t} \;=\; \beta_1 \boldsymbol{\mu}_{\beta_1,t-1} + (1-\beta_1)\boldsymbol{g}_t, \qquad \boldsymbol{v}_{\beta_2,t} \;=\; \beta_2 \boldsymbol{v}_{\beta_2,t-1} + (1-\beta_2)\boldsymbol{g}_t^2. \tag{17}$$

We treat the Adam denominator as a time-varying cost vector $c_j \coloneqq (v_{\beta_2,j} + \epsilon)^{1/2}$, where j is the parameter coordinate index and $\epsilon > 0$ is a small regularization term. Define an elementwise weighted norm-bounded family $\mathcal{Q}_{\mathrm{D}}(B, \boldsymbol{c})$ and an Adam 1-pole family $\mathcal{Q}_{\mathrm{1p}}(\boldsymbol{c})$ of optimizer filters.

$$\mathcal{Q}_{\mathrm{D}} \coloneqq \Big\{ \operatorname{diag}(q_j) : \sum_j c_j \sum_{k \geq 0} |q_{j,k}|^2 \leq B \Big\}, \quad \mathcal{Q}_{\mathrm{1p}}(\boldsymbol{c}) \coloneqq \big\{ q_{j,k} = \eta(1-\beta_1)\beta_1^k c_j^{-1} : \eta \geq 0, 0 < \beta_1 < 1 \big\}. \tag{18}$$

Corollary 3.3 then extends Theorem 3.1 to Adam [36]. In this corollary, we condition on the realized second-moment state $\boldsymbol{v}_{\beta_2,t}$ for each candidate β_2 and view Adam as a local time-varying filter.

Algorithm 1 K-Switch SGD+Momentum

Require: η, $\{\beta_k\}_{k=1}^K$

$\boldsymbol{\mu}_k \leftarrow \mathbf{0}$ for all k
for each step n **do**
  $\boldsymbol{g} \leftarrow \nabla_\theta \mathcal{L}(\boldsymbol{\theta})$
  for $k = 1, \ldots, K$ **do**
    $\boldsymbol{\mu}_k \leftarrow \beta_k \boldsymbol{\mu}_k + \boldsymbol{g}$
    $A_k \leftarrow \sqrt{1-\beta_k^2}\, \boldsymbol{g}^\top \boldsymbol{\mu}_k$
  end for
  $k^\star \leftarrow \arg\max_k A_k$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \boldsymbol{\mu}_{k^\star}$
end for

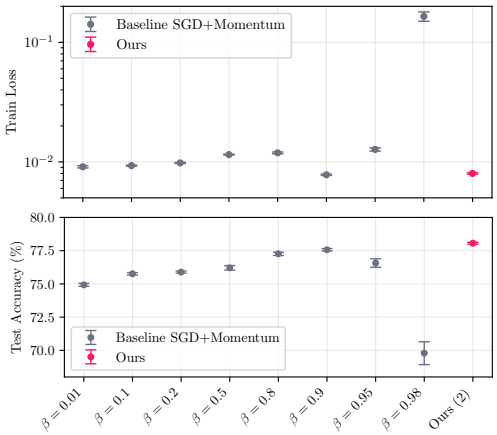


Figure 2: Empirical instantiation of Theorem 3.1 for SGD+Momentum. The greedy optimal SGD+Momentum of Algorithm 1 is compared with fixed-hyperparameter baselines on the CIFAR-100 dataset [38] with ResNet-18 [30]. The error bars indicate the mean and standard deviation over 10 runs.

Table 1: Demonstration of Theorem 3.1 for SGD with momentum. Best baseline at $\beta = 0.9$. mean $\pm$ std.

Method	Test acc. %	Train loss
Best baseline	77.57 ± 0.09	0.0078 ± 0.0001
Ours	78.06 ± 0.07	0.0080 ± 0.0001

Table 2: Demonstration of Corollary 3.3 for Adam. Best baseline at $(\beta_1, \beta_2) = (0.8, 0.999)$. mean $\pm$ std.

Method	Test acc. %	Train loss
Best baseline	73.20 ± 0.21	0.0324 ± 0.0042
Ours	73.26 ± 0.31	0.0115 ± 0.0010

Corollary 3.3 (Greedy optimal Adam/AdamW)**.** [proof] *Consider the weighted norm-bounded family $\mathcal{Q}_{\eta,\beta_1,\beta_2} = \mathcal{Q}_{\mathrm{D}}(B, \boldsymbol{c}) \cap \mathcal{Q}_{1\mathrm{p}}(\boldsymbol{c})$ of optimizer filters. Then problem (♠) under the prescribed family $\mathcal{Q}_{\eta,\beta_1,\beta_2}$ reduces to the* Adam/AdamW *optimizer with* greedy optimal hyperparameters*:*

$$(\beta_1^\star, \beta_2^\star)_t \;=\; \arg \max_{\beta_1,\beta_2 \in (0,1)^2} \sqrt{\tfrac{1+\beta_1}{1-\beta_1}}\, \frac{1}{\sqrt{\sum_j c_{\beta_2,t,j}^{-1}}}\, \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{u}_{\beta_1,\beta_2,t}], \tag{19}$$

where $c_{\beta_2,t,j} = \sqrt{v_{\beta_2,t,j} + \epsilon}$ and $\boldsymbol{u}_{\beta_1,\beta_2,t} := \boldsymbol{\mu}_{\beta_1,t} \odot \boldsymbol{c}_{\beta_2,t}^{-1}$ is the Adam update direction without bias correction and decoupled weight decay, which is applied separately in AdamW. The expectation is understood conditionally on the realized second-moment state $\boldsymbol{c}_{\beta_2,t}$.

All the proofs are in Appendix E. This derivation shows that the greedy optimal hyperparameters are not universal constants but functions of the gradient autocorrelation structure. In practice, training algorithms are online and therefore approximate the expectation using either the instantaneous score $\boldsymbol{g}_t^\top \boldsymbol{u}_t$ or the average of recent batch scores. Theorem 3.1 and Corollary 3.3 show that the greedy optimal hyperparameters that maximize the expected loss drop are attained by maximizing the alignment score between the gradients and the optimizer responses for both SGD+Momentum and Adam/AdamW. This opens a new interpretation of these algorithms from the filter viewpoint: SGD+Momentum can be interpreted as an *optimal 1-pole approximation of the norm-bounded optimizer*, and Adam/AdamW can be interpreted as an *optimal 1-pole approximation of the dynamic diagonal optimizer with a preconditioner $\boldsymbol{c}_{\beta_2,t}^{-1}$*.

3.2 Finite-step descent bridge

So far, we have derived the theory based on equation 5, which assumes continuous-time gradient and update streams. In reality, training is performed step by step, leaving a gap between the continuous-time theory and the discrete-time algorithm. The following theorem closes this gap.

Theorem 3.4 (Finite-step descent)**.** [proof] *Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be differentiable with the L_s-Lipschitz gradient. Fix $\boldsymbol{\theta} \in \mathbb{R}^d$ and let $\boldsymbol{g} := \nabla_{\boldsymbol{\theta}} \mathcal{L}$. For any update $\boldsymbol{u} \in \mathbb{R}^d$ and learning rate $\eta > 0$, define the alignment score $A(\boldsymbol{u}) := \boldsymbol{g}^\top \boldsymbol{u}$ and the one-step loss drop $D_\eta(\boldsymbol{u}) := \mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta} - \eta \boldsymbol{u})$. Then*

$$\left| D_\eta(\boldsymbol{u}) - \eta A(\boldsymbol{u}) \right| \;\le\; \frac{L_s \eta^2}{2} \|\boldsymbol{u}\|^2. \tag{20}$$

Algorithm 2 K-Switch Adam

Require: η, $\{(\beta_{1,k}, \beta_{2,k})\}_{k=1}^{K}$

$\boldsymbol{\mu}_k, \boldsymbol{v}_k \leftarrow \mathbf{0}$ for all k
for each step t **do**
  $\boldsymbol{g} \leftarrow \nabla_\theta \mathcal{L}(\boldsymbol{\theta})$
  for $k = 1, \ldots, K$ **do**
    $\boldsymbol{v}_k \leftarrow \beta_{2,k}\boldsymbol{v}_k + (1-\beta_{2,k})\boldsymbol{g}^2$
    $\boldsymbol{\mu}_k \leftarrow \beta_{1,k}\boldsymbol{\mu}_k + (1-\beta_{1,k})\boldsymbol{g}$
    $\boldsymbol{u}_k \leftarrow \boldsymbol{\mu}_k / (\sqrt{\boldsymbol{v}_k + \epsilon})$
    $a_k \leftarrow \sqrt{\frac{1+\beta_{1,k}}{(1-\beta_{1,k})(\sum_j (v_{k,j}+\epsilon)^{-1/2})}}$
    $A_k \leftarrow a_k \boldsymbol{g}^\top \boldsymbol{u}_k$
  end for
  $k^\star \leftarrow \arg\max_k A_k$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \boldsymbol{u}_{k^\star}$
end for

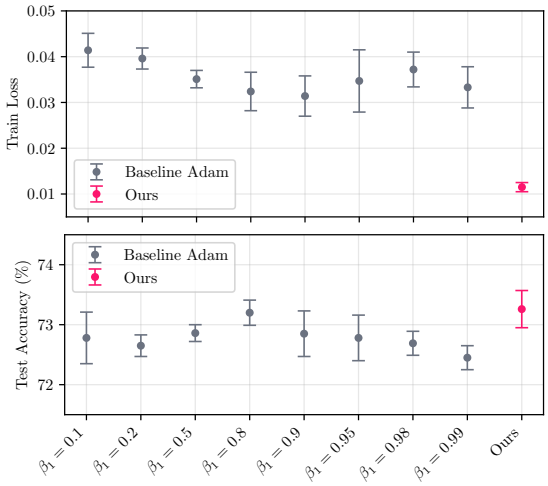


Figure 3: Empirical instantiation of Corollary 3.3 for Adam [36]. The greedy optimal Adam of Algorithm 2 is compared with fixed-hyperparameter baselines on the CIFAR-100 dataset [38] with ResNet-18 [30]. The error bars indicate the mean and standard deviation over 10 runs.

Table 3: AdamW on Gemma-2B with MetaMathQA-395K validated on GSM8K. mean ± std.

Method	Test acc. %	Train loss
Best baseline	52.57 ± 1.10	0.2080 ± 0.0004
Ours	52.77 ± 0.93	0.2084 ± 0.0003

Table 4: AdamW on Llama-3-8B with MetaMathQA-395K dataset. mean ± std.

Method	Test acc. %	Train loss
Best baseline	76.20 ± 0.33	0.1927 ± 0.0005
Ours	76.30 ± 0.31	0.1925 ± 0.0005

Therefore, $\boldsymbol{g}^\top \boldsymbol{u}$ is the correct first-order term of the actual one-step loss decrease. The approximation is reliable as long as the local quantity $\eta\|\boldsymbol{u}\|$ is small enough. Outside this regime, the alignment score remains a first-order approximation but should be paired with a norm control or other stabilization. The following corollary further solidifies the greedy alignment as a reliable local decision rule.

Corollary 3.5 (Near-optimality of greedy alignment)**.** [proof] *Let $\mathcal{U}$ be a candidate set of updates with $\|\boldsymbol{u}\| \le \rho$ for all $\boldsymbol{u} \in \mathcal{U}$. Let $\boldsymbol{u}^\star \in \arg\max_{\boldsymbol{u}\in\mathcal{U}} A(\boldsymbol{u})$. Then*

$$D_\eta(\boldsymbol{u}^\star) \;\ge\; \max_{\boldsymbol{v}\in\mathcal{U}} D_\eta(\boldsymbol{v}) - L_s\eta^2\rho^2. \tag{21}$$

In particular, if $L_s\eta^2\rho^2$ is small, then maximizing $A(\boldsymbol{u}) = \boldsymbol{g}^\top\boldsymbol{u}$ is an $O(\eta^2)$-accurate surrogate for maximizing the one-step decrease.

Corollary 3.5 says that over norm-bounded candidate updates, greedy maximum gives a near-optimal choice for the true one-step loss decrease. In other words, as long as the bound of the local regime $\eta\|\boldsymbol{u}\|$ remains small enough, greedy alignment is a reliable local decision rule. The operator-level greedy objective (♠) is an expected inner product $P(Q) = \mathbb{E}[\boldsymbol{g}_t^\top (Q\boldsymbol{g})_t]$. The online implementation replaces this expectation by an (averaged) instantaneous score $A_t(\boldsymbol{u}) \coloneqq \boldsymbol{g}_t^\top \boldsymbol{u}_t$. Theorem 3.4 and Corollary 3.5 justify the use of $\boldsymbol{g}^\top\boldsymbol{u}$ as a local surrogate by showing that since the second-order error is controlled in the local regime, greedy alignment based on $\boldsymbol{g}^\top\boldsymbol{u}$ is a reliable first-order approximation of the actual loss decrease.

4 Realization and practical concerns

4.1 A finite-candidate realization of greedy optimizer selection

This section provides a simple realization of the greedy optimal algorithm for optimizer selection, dubbed K-Switch. Rather than attempting a full hyperparameter sweep over the entire training dynamics, K-Switch is an online algorithm for the finite-candidate greedy alignment objective. The argmax operator in the theory can be implemented in multiple ways; we explore the simplest possible option by selecting from a fixed set of K candidate optimizers. For each candidate $k = 1, \ldots, K$,

we maintain an optimizer state $Q_{k,t}$ and compute the candidate update $\boldsymbol{u}_{k,t} = (Q_{k,t}\boldsymbol{g}_{\le t})_t$. Define normalized candidates $\tilde{Q}_{k,t}$ by $(\tilde{Q}_{k,t}\boldsymbol{g}_{\le t})_t = a_{k,t}\boldsymbol{u}_{k,t}$, with normalization factors $a_{k,t} > 0$ specific to the prescribed family of optimizers, e.g., $a_{k,t} = \sqrt{1-\beta_k^2}$ for SGD+Momentum [52]. Generally, we find the candidate that maximizes the normalized alignment score $\bar{A}_{k,t} \coloneqq \mathbb{E}[a_k \boldsymbol{g}_t^\top \boldsymbol{u}_{k,t}]$:

$$k_t^\star = \arg\max_{k\in\{1,\dots,K\}} \bar{A}_{k,t}. \tag{22}$$

By the family-hull reduction in Corollary 2.3, optimizing over the convex hull of these candidates strictly reduces to selecting the best single candidate over the K options:

Corollary 4.1 (Exact finite-candidate K-Switch)**.** [proof] *For* $P_t(Q) \coloneqq \mathbb{E}[\boldsymbol{g}_t^\top (Q\boldsymbol{g})_t]$,

$$\sup_{Q\in\mathrm{co}\{\tilde{Q}_{k,t}\}_{k=1}^K} P_t(Q) \;=\; \max_k P_t(\tilde{Q}_{k,t}) \;=\; \max_k \mathbb{E}[a_{k,t}\boldsymbol{g}_t^\top \boldsymbol{u}_{k,t}]. \tag{23}$$

The implication is straightforward: K-Switch is not a heuristic approximation, but the exact finite-candidate solution of Theorem 2.2 over the convexified normalized family.

4.2 Practical concerns

Online algorithm and stability. In practice, the exact expectation $\bar{A}_{k,t}$ is inaccessible, and we must rely on the mini-batch estimate $\hat{A}_{k,t} \coloneqq \mathbb{E}_{\text{batch}}[a_k \boldsymbol{g}_t^\top \boldsymbol{u}_{k,t}]$. The following proposition ensures that this statistical approximation remains stable.

Proposition 4.2 (Online stability)**.** [proof] *Let* $\hat{A}_k$ *be an online estimate of* A_k*, and assume* $\max_k |\hat{A}_k - A_k| \le \varepsilon$*. If* $\hat{k} \in \arg\max_k \hat{A}_k$ *and* $k^\star \in \arg\max_k A_k$*, then* $A_{k^\star} - A_{\hat{k}} \le 2\varepsilon$*. If the best candidate is unique with a gap* $> 2\varepsilon$*, then* $\hat{k} = k^\star$*.*

Proposition 4.2 translates the Lipschitz stability of Theorem 2.2(iii) from $P^\star(R)$ to a finite argmax over the unit-energy family $\mathcal{F} = \{Q/\|Q\| : Q \text{ an optimizer}\}$. Thus, the resulting K-Switch algorithms in Algorithms 1 and 2 are robust streaming estimators of the exact finite-candidate greedy alignment maximizer. In practice, we can further stabilize the K-Switch algorithms by the following modifications. This leads to Algorithms 1 and 2. For clarity, Algorithm 2 shows the gradient-driven Adam update and omits bias correction and decoupled weight decay. In AdamW experiments, the alignment score is computed only on the gradient-driven update, while the decoupled weight decay is applied identically across all candidates after the selected update. Lines highlighted in pink are the actual augmentations to the baseline algorithms; all other lines are unchanged.

Separating the roles of the optimizer and the learning rate scheduler. In practical scenarios, the parameter update $\boldsymbol{u}_t$ is scaled by a learning rate scheduler η_t which is usually implemented externally to the optimizers. Respecting this common design choice, we decouple hyperparameter selection from parameter innovation. The Hilbert-norm factor $a_{k,t}$ is used exclusively in the hyperparameter selection stage, comparing different optimizer hyperparameters on an equal-energy trust region to identify the best-aligned filter pole (e.g., β) without introducing scale bias (Corollary 3.2). Once $k_\star$ is selected, we can update the parameters via $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t \boldsymbol{u}_{k^\star,t}$, with $a_{k,t}$ absorbed in the externally set learning rate η_t. This ensures compatibility with existing frameworks using learning rate scheduling and prevents this additional step-size modulation from interfering with the directional alignment of the gradient $\boldsymbol{g}_t$ and parameter update $\boldsymbol{u}_t$. This is shown in Algorithms 1 and 2.

5 Experiments

Implementation details. One of the practical advantages of our framework is the *automatic* selection of momentum hyperparameters in a greedy sense, as shown in Theorem 3.1 and Corollary 3.3. We provide empirical justification of the theory in previous sections by the simplest instantiations of these theoretical frameworks elaborated in Section 4: K-Switch dynamic optimizers with $K = 2$. The candidate endpoints are chosen a priori to cover the conventional working range of the corresponding optimizer family, rather than by sweeping over all pairs. In contrast, fixed baselines are selected by an oracle sweep over the reported grid. Complete details and results are provided in Appendix D.

Table 5: AdamW on Gemma-2B with Commonsense-170K dataset. mean ± std.

Gemma-2B (LoRA)	BoolQ	PIQA	Social IQA	HellaSwag	Winogrande	OBQA	**Avg**
Best baseline	65.69 ± 0.29	78.93 ± 0.49	73.61 ± 0.16	74.07 ± 0.28	71.61 ± 0.44	72.67 ± 0.68	72.76 ± 0.17
Ours	65.31 ± 0.04	79.00 ± 0.36	73.58 ± 0.06	75.09 ± 1.02	71.80 ± 0.39	73.27 ± 1.15	73.01 ± 0.27
Gemma-2B (Full FT)	BoolQ	PIQA	Social IQA	HellaSwag	Winogrande	OBQA	**Avg**
Best baseline	62.79 ± 0.27	74.12 ± 0.26	66.63 ± 0.33	40.50 ± 1.15	61.48 ± 0.32	62.60 ± 1.02	61.35 ± 0.27
Ours	63.29 ± 0.78	75.70 ± 0.22	68.41 ± 0.69	42.47 ± 1.06	62.46 ± 4.64	64.40 ± 0.86	62.79 ± 0.83

Table 6: AdamW on Vision Transformer fine-tuning tasks with LoRA. mean ± std.

ViT-B (rank = 32)	Cars	CIFAR-100	CUB-200	DTD	Food-101	RESISC45	SUN397	**Avg**
Best baseline	77.56 ± 0.09	91.74 ± 0.07	84.67 ± 0.06	78.32 ± 0.38	88.13 ± 0.03	94.54 ± 0.01	72.72 ± 0.08	83.95 ± 0.06
Ours	77.95 ± 0.38	91.87 ± 0.02	84.56 ± 0.13	78.23 ± 0.48	88.16 ± 0.09	94.24 ± 0.09	72.71 ± 0.21	83.96 ± 0.10
ViT-L (rank = 8)	Cars	CIFAR-100	CUB-200	DTD	Food-101	RESISC45	SUN397	**Avg**
Best baseline	84.89 ± 0.12	93.20 ± 0.08	87.08 ± 0.21	80.04 ± 0.18	89.98 ± 0.07	95.13 ± 0.08	75.18 ± 0.10	86.50 ± 0.05
Ours	85.40 ± 0.11	93.05 ± 0.01	86.80 ± 0.12	80.74 ± 0.44	90.04 ± 0.15	95.07 ± 0.02	74.87 ± 0.08	86.57 ± 0.07

ResNet-18 on CIFAR-100. Using this simple instantiation, we empirically demonstrate the theory by training a ResNet-18 [30] model on the CIFAR-100 dataset [38]. Baseline optimizers are trained with fixed hyperparameters, in line with typical machine learning practices. For the momentum of SGD+Momentum, we tried $\beta \in [0.01, 0.999]$ and reported the best. For Adam, we tried $\beta_1 \in [0.1, 0.99]$ while keeping $\beta_2 = 0.999$ fixed and reported the best. Figures 2 and 3 and Tables 1 and 2 summarize the results. Our K-Switch dynamic optimizers achieve comparable and often better performance than the baseline optimizers with fixed hyperparameters in both final accuracy and convergence speed. This largely reduces the need for a heavy hyperparameter sweep.

Large language models and vision transformers. We also demonstrate our framework in more practical scenarios: training large language models (LLMs) and vision transformers (ViTs). We train a Gemma-2B [26] and Llama-3-8B [28] using low rank adaptation (LoRA) [31] with standard settings on the MetaMathQA-395K dataset [67] and compare the results with baseline optimizers of fixed hyperparameters on GSM8K [17]. Table 3 and 4 summarize the results of ten runs for each model. Furthermore, we also train a Gemma-2B with both LoRA and full fine-tuning on the Commonsense-170K dataset [32] using our and baseline optimizers of fixed hyperparameters. We then compare the results on various reasoning tasks such as BoolQ [16], PIQA [6], Social IQA [54], HellaSwag [69], Winogrande [53], and OBQA [32], and summarize the results in Table 5. Finally, we train ViT-B and ViT-L models [20] by LoRA on various image classification tasks including Cars [37], CIFAR-100 [38], CUB-200 [61], DTD [15], Food-101 [7], RESISC45 [13], and SUN397 [64]. Table 6 summarizes the results. Detailed settings and extended results are provided in Appendix D.

The strongest large-model evidence appears on Commonsense-170K, where K-Switch AdamW improves the unweighted average over the best reported fixed-β_1 baseline by $+0.25$ points for LoRA and $+1.44$ points for full fine-tuning. In the MetaMathQA and ViT settings, our K-Switch remains competitive with oracle-selected fixed-momentum baselines.

Computational overhead. Table 7 compares total hyperparameter search cost of our K-Switch against fixed-β sweeps. We report per-step wall-clock time measured with the same fused PyTorch/Triton implementation of the optimizers. We can reduce the additional memory requirements of $K = 2$ switch to below $+100\%$ with this implementation. In LoRA-based fine-tuning, the additional memory overhead can be lower than $+10\%$. For $K = 2$, the practical default, per-step runtime overhead is small in LoRA regimes and moderate in full-model runs. Nevertheless, full sweep requires multiple runs of the same task to get the best hyperparameter settings, whereas our K-switch adapts the hyperparameters in real-time, requiring only a single run for each task. Regarding the final performance, our K-Switch consistently outperforms the best fixed-hyperparameter settings found by dense manual sweep. Our theory provides a practical alternative to manual hyperparameter sweeps, reducing the estimated hyperparameter tuning cost by at least $\times 4.62$ across various tasks.

6 Limitations and scope

Greedy paradigm. Our objective $\max_{Q \in \mathcal{Q}} \langle Q, R_t \rangle_{\mathcal{H}}$ is a greedy training-statistics objective that maximizes the expected first-order loss drop rate. Thus, our theory should not be interpreted as

Table 7: Hyperparameter tuning cost comparison between K-Switch and full hyperparameter sweep.

	ResNet-18 (full model)	Gemma-2B ($r = 32$ LoRA)	Gemma-2B (Full FT)	Llama-3-8B ($r = 32$ LoRA)	ViT-Base ($r = 32$ LoRA)	ViT-Large ($r = 8$ LoRA)
Original VRAM	0.961 GiB	16.303 GiB	16.943 GiB	19.910 GiB	1.078 GiB	2.049 GiB
K-switch VRAM	1.344 GiB	16.303 GiB	27.920 GiB	21.381 GiB	1.080 GiB	2.049 GiB
Memory overhead	$+39.84\%$	$+0.00\%$	$+64.78\%$	$+7.39\%$	$+0.18\%$	$+0.00\%$
# β candidates	8	6	6	6	6	6
Per-step runtime	$+14.0\%$	-0.50%	$+29.9\%$	$+0.33\%$	$+2.21\%$	$+1.01\%$
Full sweep speedup	$\times\mathbf{7.02}$	$\times\mathbf{6.03}$	$\times\mathbf{4.62}$	$\times\mathbf{5.98}$	$\times\mathbf{5.87}$	$\times\mathbf{5.94}$

a global convergence guarantee, a validation-loss guarantee, or a population-optimality result in arbitrary nonlinear nonconvex landscapes. Long-horizon guarantees should be incorporated with optimizer-specific analyses of the underlying optimizer family.

Dependence on optimizer family. The framework optimizes within a user-specified feasible family $\mathcal{Q}$ and, by itself, does not determine which class of optimizers is best for a given task. In practice, this means that our method reduces the inner search over hyperparameters while still relying on a reasonable candidate family chosen by the practitioner.

Adaptive optimizers and overhead. Adaptive optimizers such as Adam/AdamW involve nonlinear state-dependent operations, so we treat them through realized update streams or local time-varying filter approximations rather than as globally linear time-invariant filters. Our K-Switch implementation is a simple reference instantiation of the theory; it requires maintaining multiple candidate optimizer states, so memory and runtime costs depend on K, model size, etc.

7 Conclusion

By interpreting optimizers as statistical filters, we recast optimizer tuning into a principled greedy optimization problem: selecting the causal filter that maximizes the expected rate of loss decrease. This dual problem, running alongside the primary problem of loss minimization, allows us to choose a greedy-optimal optimizer based on the current gradient statistics while training the model with the selected optimizer. Specializing in momentum-based optimizers, this perspective transforms optimizer hyperparameter selection into an online alignment problem. The resulting K-Switch algorithm is lightweight and empirically reduces the need for exhaustive fixed-momentum sweeps.

Acknowledgments and Disclosure of Funding

This work was supported in part by the IITP grants [No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University), No. RS-2025-02303870, No.2022-0-00156] funded by the Korean government (MSIT).

Broader impacts

The main goal of this paper is two-fold: (1) to reduce the wasteful time and budget currently spent on hyperparameter tuning, and (2) to broaden the accessibility of machine learning techniques by alleviating the need for such efforts. By advancing this direction, we expect to lower the computational overhead and operational costs of large-scale machine learning systems. Ultimately, we hope that this contributes to a more democratic and sustainable AI ecosystem, benefiting both society and the environment.

References

[1] S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[2] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, 2016.

[3] A. G. Baydin, R. Cornish, D. M. Rubio, M. Schmidt, and F. Wood. Online learning rate adaptation with hypergradient descent. In *ICLR*, 2018.

[4] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le. Neural optimizer search with reinforcement learning. In *ICML*, 2017.

[5] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *NeurIPS*, 2011.

[6] Y. Bisk, R. Zellers, R. Le Bras, J. Gao, and Y. Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

[7] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[9] K. Chandra, A. Xie, J. Ragan-Kelley, and E. Meijer. Gradient descent: The ultimate optimizer. In *NeurIPS*, 2022.

[10] D. Chang and G. Yuan. MGUP: A momentum-gradient alignment update policy for stochastic optimization. In *NeurIPS*, 2025.

[11] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le. Symbolic discovery of optimization algorithms. In *NeurIPS*, 2023.

[12] Y. Chen, X. Song, C. Lee, Z. Wang, Q. Zhang, D. Dohan, K. Kawakami, G. Kochanski, A. Doucet, M. Ranzato, S. Perel, and N. de Freitas. Towards learning universal hyperparameter optimizers with transformers. In *NeurIPS*, 2022.

[13] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[14] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl. On empirical comparisons of optimizers for deep learning, 2019.

[15] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[16] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

[17] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[18] A. Defazio and K. Mishchenko. Learning-rate-free learning by D-Adaptation. In *ICML*, 2023.

[19] A. Defazio, X. A. Yang, A. Khaled, K. Mishchenko, H. Mehta, and A. Cutkosky. The road less scheduled. In *NeurIPS*, 2024.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[21] T. Dozat. Incorporating Nesterov momentum into Adam. In *ICLR Workshop*, 2016.

[22] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

[23] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[24] S. Falkner, A. Klein, and F. Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *ICML*, 2018.

[25] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil. Forward and reverse gradient-based hyperparameter optimization. In *ICML*, 2017.

[26] Gemma Team et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2023.

[27] B. Goujaud, C. Moucer, F. Glineur, J. M. Hendrickx, A. B. Taylor, and A. Dieuleveut. PEPit: Computer-assisted worst-case analyses of first-order optimization methods in Python. *Mathematical Programming Computation*, 16:337–367, 2024.

[28] A. Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[29] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *ICML*, 2018.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[31] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[32] Z. Hu, L. Wang, Y. Lan, W. Xu, E.-P. Lim, L. Bing, X. Xu, S. Poria, and R. Lee. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *EMNLP*. ACL, 2023.

[33] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[34] K. Jordan. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

[35] D. Kim and J. A. Fessler. On the convergence analysis of the optimized gradient method. *Journal of Optimization Theory and Applications*, 172(1):187–205, 2017.

[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[37] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *ICCVW*, 2013.

[38] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Canada, 2009.

[39] J. Lee, B. G. Kang, K. Kim, and K. M. Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2024.

[40] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(6765–6816), 2018.

[41] K. Liang, L. Chen, B. Liu, and Q. Liu. Cautious optimizers: Improving training with one line of code. In *ICLR*, 2026.

[42] H. Liu, Z. Li, D. L. W. Hall, P. Liang, and T. Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *ICLR*, 2024.

[43] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020.

[44] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[45] L. Luo, Y. Xiong, Y. Liu, and X. Sun. Adaptive gradient methods with dynamic bound of learning rate. In *ICLR*, 2019.

[46] D. Maclaurin, D. Duvenaud, and R. P. Adams. Gradient-based hyperparameter optimization through reversible learning. In *ICML*, 2015.

[47] J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*, 2015.

[48] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269(3):543–547, 1983. English translation: Soviet Mathematics Doklady, 27(2):372–376, 1983.

[49] T. Pethick, W. Xie, K. Antonakopoulos, Z. Zhu, A. Silveti-Falls, and V. Cevher. Training deep learning models with norm-constrained lmos. In *ICML*, 2025.

[50] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[51] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *ICLR*, 2018.

[52] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[53] K. Sakaguchi, R. Le Bras, C. Bhagavatula, and Y. Choi. WinoGrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021. doi: 10.1145/3474381.

[54] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi. Social IQA: Commonsense reasoning about social interactions. In *EMNLP*, 2019.

[55] R. M. Schmidt, F. Schneider, and P. Hennig. Descending through a crowded valley — benchmarking deep learning optimizers. In *ICML*, 2021.

[56] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots. Truncated back-propagation for bilevel optimization. In *AISTATS*, 2019.

[57] N. Shazeer and M. Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICML*, 2018.

[58] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[59] T. Tieleman and G. Hinton. Lecture 6.5—-RMSProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[60] N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. M. Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *ICLR*, 2025.

[61] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[62] R. Wang, Y. Xiong, M. Cheng, and C.-J. Hsieh. Efficient non-parametric optimizer search for diverse tasks. In *NeurIPS*, 2022.

[63] O. Wichrowska, N. Maheswaranathan, M. W. Hoffman, S. G. Colmenarejo, M. Denil, N. de Freitas, and J. Sohl-Dickstein. Learned optimizers that scale and generalize. In *ICML*, 2017.

[64] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN Database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[65] X. Xie, P. Zhou, H. Li, Z. Lin, and S. Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9508–9520, 2024.

[66] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *ICLR*, 2020.

[67] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *ICLR*, 2024.

[68] H. Yuan, Y. Liu, S. Wu, X. Zhou, and Q. Gu. MARS: Unleashing the power of variance reduction for training large models. In *ICML*, 2025.

[69] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, 2019.

[70] M. R. Zhang, J. Lucas, G. E. Hinton, and J. Ba. Lookahead optimizer: k steps forward, 1 step back. In *NeurIPS*, 2019.

[71] W. Zheng, T. Chen, T.-K. Hu, and Z. Wang. Symbolic learning to optimize: Towards interpretability and scalability. *arXiv preprint arXiv:2203.06578*, 2022.

[72] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, and J. S. Duncan. AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients. In *NeurIPS*, 2020.

A Related work

Momentum, adaptive optimizers, and the tuning bottleneck. Momentum [50, 48] has been a central mechanism in first-order optimization for decades [58], and SGD [52] with momentum remains a canonical optimizer in deep learning. In order to accelerate, stabilize, and better generalize in training large models, dozens of adaptive optimizers have been proposed to modify momentum, by preconditioning or scaling the parameter updates. To name a few, this includes AdaGrad [23], RMSProp [59], Adam [36], AdamW [44], Nadam [21], AMSGrad [51], Adafactor [57], AdaBound [45], RAdam [43], AdaBelief [72], LAMB [66], Adan [65], Sophia [42], Lion [11], and MARS [68]. Most of them introduce their own set of hyperparameters, each with different recommended ranges and tuning strategies. Optimizer-level mechanisms such as Lookahead [70], Schedule-Free learning [19], and D-Adaptation [18] reduce reliance on manually chosen schedules or step sizes. Despite this progress, optimizer performance is highly sensitive to the tuning procedure. Large-scale studies [14, 55] have shown that optimizer rankings vary under different hyperparameter search spaces and that no single optimizer dominates across tasks. We address this bottleneck from a different approach: rather than proposing a new fixed optimizer, we formulate the online selection of optimizer hyperparameters as a greedy alignment problem between an optimizer filter and the gradient autocorrelation.

Gradient–update alignment. Several works have studied the alignment between gradients and generated parameter updates to accelerate and stabilize training. Lion [11] has discovered sign-based momentum updates through symbolic search. Cautious Optimizer framework [41] proposes a simple hard masking rule for momentum-based optimizers, which updates only on coordinates where the proposed optimizer update and the current gradient are positively aligned. MGUP [10], in contrast, relaxes hard rejection with step size modulation to weight the updates on high-alignment coordinates while preserving small but nonzero updates on the remaining coordinates. These methods operate primarily at the *actuation* level: given an update, they decide how much of it should be applied to each parameter coordinate. Our framework operates one level above this: we apply alignment to select the optimizer itself. In this sense, Cautious-style masking and MGUP-style prioritization are complementary safeguards for applying a selected update, whereas our main object is the filter-selection objective.

Hyperparameter optimization. Classical hyperparameter optimization methods, including Bayesian optimization, random search [5], Hyperband [40], BOHB [24], and population-based training [33], tune hyperparameters by repeatedly or partially training models under different configurations. Instead, gradient-based hyperparameter optimization methods make the training procedure differentiable or approximate hypergradients, enabling optimization of learning rates, momentum schedules, regularization coefficients, and other training hyperparameters [46, 25, 3, 56, 9, 12]. These methods typically optimize validation performance or long-horizon objectives, often with nontrivial unrolling, meta-training, or repeated evaluations. Our method is more lightweight, operating in local, online regimes, rather than global, offline regimes. We use the current training gradient and candidate updates to estimate the greedy training-power objective, avoiding validation unrolling or HPO outer loops.

Performance-guided or learned optimizer design. The Performance Estimation Problem (PEP) framework analyzes or designs first-order methods through worst-case convex optimization objectives [22, 35, 27]. On the other hand, optimizer discovery methods search over update-rule programs or symbolic expressions, as in neural optimizer search [4], symbolic learning [71], and non-parametric optimizer search [62]. Learning-to-optimize approaches train optimizer networks, either RNN-based or hierarchical [2, 63], to return update rules from gradient information. These approaches are oriented to discovering new optimizers, whereas our framework assumes a prescribed optimizer family and gives a closed-form greedy score for selecting among its members. Thus, our work is closer to online optimizer selection than to optimizer discovery, learning, or synthesis.

B Generalization beyond linear optimizers and finite-candidate exactness

This section provides more detailed mathematical foundations and generalization of the main text, which were omitted for brevity and simplicity. In the main text, we have developed the theory mainly for *causal linear filters* Q. Linearity gives the inner product structure of the learning power functional

$P_t(Q) = \langle Q, R \rangle_{\mathcal{H}}$, leading to elegant theory uncovering interpretable relationships between the optimizer Q and the gradient autocorrelation R. We showed that this simplest case already captures existing practical optimizers such as SGD+Momentum [52]. However, the resulting framework relies on restrictive linearity assumptions, and extension to adaptive optimizers such as Adam [36] and AdamW [44] required additional assumptions. Here, we generalize beyond linearity and extend the theory to general causal optimizers Q. The advantage is clear: we can apply the same *greedy alignment principle* to nonlinear causal optimizers as well.

We begin with the general form of the learning power functional:

$$P_t(Q) \;=\; \mathbb{E}[\boldsymbol{g}_t^\top (Q\boldsymbol{g})_t] \tag{24}$$

where $\boldsymbol{g}_t$ is a gradient stream and Q is a general causal optimizer. Even though Q is not a linear functional over the gradient stream $\boldsymbol{g}_t$, the learning power functional is still linear in Q. The key insight is that the additivity and homogeneity of an optimizer filter Q comes from the additivity and homogeneity of the generated parameter updates $\boldsymbol{u}_t = (Q\boldsymbol{g})_t$, not from the relationship between Q and $\boldsymbol{g}_t$.

Lemma B.1 (Learning power functional is linear in Q)**.** *For any two causal optimizers Q_1 and Q_2, and any scalars α_1 and α_2,*

$$P_t(\alpha_1 Q_1 + \alpha_2 Q_2) \;=\; \alpha_1 P_t(Q_1) + \alpha_2 P_t(Q_2) \tag{25}$$

Proof. Let Q_1 and Q_2 be arbitrary causal optimizers (not necessarily linear in $\boldsymbol{g}$). Define the combined optimizer $Q = \alpha_1 Q_1 + \alpha_2 Q_2$ by its action on the gradient stream: $(Q\boldsymbol{g})_t \coloneqq \alpha_1 (Q_1\boldsymbol{g})_t + \alpha_2 (Q_2\boldsymbol{g})_t$. Then:

$$\begin{aligned}
P_t(\alpha_1 Q_1 + \alpha_2 Q_2) \;&=\; \mathbb{E}[\boldsymbol{g}_t^\top (Q\boldsymbol{g})_t] \\
&=\; \mathbb{E}[\boldsymbol{g}_t^\top (\alpha_1 (Q_1\boldsymbol{g})_t + \alpha_2 (Q_2\boldsymbol{g})_t)] \\
&=\; \mathbb{E}[\alpha_1 \boldsymbol{g}_t^\top (Q_1\boldsymbol{g})_t + \alpha_2 \boldsymbol{g}_t^\top (Q_2\boldsymbol{g})_t] \\
&=\; \alpha_1 \mathbb{E}[\boldsymbol{g}_t^\top (Q_1\boldsymbol{g})_t] + \alpha_2 \mathbb{E}[\boldsymbol{g}_t^\top (Q_2\boldsymbol{g})_t] \\
&=\; \alpha_1 P_t(Q_1) + \alpha_2 P_t(Q_2)
\end{aligned}$$

The key observation is that linearity of P_t in Q follows from (i) the definition of linear combination of optimizers acting pointwise on the output parameter updates $\boldsymbol{u}_t = (Q\boldsymbol{g})_t$, and (ii) linearity of expectation. This holds regardless of whether Q_1 or Q_2 are themselves linear operators on $\boldsymbol{g}$. □

Lemma B.1 applies to general nonlinear adaptive optimizers. For example, consider the Adam optimizer family: The Adam update $Q^{\text{Adam}}_{\beta_1,\beta_2} : \boldsymbol{g}_{\le t} \mapsto \boldsymbol{u}_t$ is indeed a complex nonlinear functional of the gradient stream $\boldsymbol{g}$. The output $(Q^{\text{Adam}}_{\beta_1,\beta_2}\boldsymbol{g})_t = \boldsymbol{m}_{\beta_1,t}/(\sqrt{\boldsymbol{v}_{\beta_2,t}} + \epsilon)$ depends nonlinearly on the entire gradient history through the exponential moving averages $\boldsymbol{m}_{\beta_1,t}$ and $\boldsymbol{v}_{\beta_2,t}$. The lemma, however, concerns linearity in Q, not linearity of Q in $\boldsymbol{g}$. That is, we can still apply the greedy alignment principle for the compact candidate family $\mathcal{Q}^{\text{Adam}}_{\mathcal{B}} = \{Q^{\text{Adam}}_{\beta_1,\beta_2} : (\beta_1,\beta_2) \in \mathcal{B}\}$ of Adam optimizers to select the optimal hyperparameters $(\beta_1^\star, \beta_2^\star)$ that maximize the learning power $P_t(Q^{\text{Adam}}_{\beta_1,\beta_2})$ under the constraint $\mathcal{Q}^{\text{Adam}}_{\mathcal{B}}$. Other nonlinear optimizers such as AdamW [44], Lion [11], Muon [34], Shampoo [29], and clipping-based optimizers can be treated as black-box causal maps and selected by the same criterion.

The following theorems justify the generalization by extending Section 2 to general nonlinear adaptive optimizers.

Theorem B.2 (Greedy optimal causal maps)**.** *Let $\mathcal{X}$ be a real locally convex vector space of causal maps $Q : \boldsymbol{g}_{\le t} \mapsto \boldsymbol{u}_t$, closed under pointwise linear combinations. Fix the gradient stream $\boldsymbol{g}_t$ at time t and define*

$$P_t(Q) \;\coloneqq\; \mathbb{E}[\boldsymbol{g}_t^\top (Q\boldsymbol{g})_t]. \tag{26}$$

Assume $P_t \in \mathcal{X}^*$ is a continuous linear functional in the dual space $\mathcal{X}^*$. Let $\mathcal{Q} \subset \mathcal{X}$ be nonempty and compact. Define the support value

$$\sigma_{\mathcal{Q}}(P_t) \;\coloneqq\; \max_{Q \in \mathcal{Q}} P_t(Q). \tag{27}$$

Then:

(i) (Existence): *The maximum is attained and finite.*

(ii) (Subgradient characterization): *If $\mathcal{Q}$ is closed and convex, then*

$$\partial\sigma_{\mathcal{Q}}(P_t) \;=\; \arg\max_{Q\in\mathcal{Q}} P_t(Q), \tag{28}$$

where the subgradient is taken with respect to the dual pairing between $\mathcal{X}^*$ and $\mathcal{X}$. In particular, if the maximizer is unique, the support function has a unique subgradient at P_t.

(iii) (Lipschitz continuity): *For any $P_t, \hat{P}_t \in \mathcal{X}^*$,*

$$|\sigma_{\mathcal{Q}}(P_t) - \sigma_{\mathcal{Q}}(\hat{P}_t)| \;\le\; \max\{\sigma_{\mathcal{Q}}(P_t - \hat{P}_t), \sigma_{\mathcal{Q}}(\hat{P}_t - P_t)\}. \tag{29}$$

Proof. (i) Existence: Since $\mathcal{Q}$ is compact and P_t is continuous, the map $Q \mapsto P_t(Q)$ attains its maximum on $\mathcal{Q}$ by the Weierstrass theorem. Finiteness follows from compactness and continuity.

(ii) Subgradient characterization: First, we show that $\sigma_{\mathcal{Q}}$ is sublinear. For $a \ge 0$,

$$\sigma_{\mathcal{Q}}(aP_t) \;=\; \max_{Q\in\mathcal{Q}} aP_t(Q) \;=\; a\max_{Q\in\mathcal{Q}} P_t(Q) \;=\; a\sigma_{\mathcal{Q}}(P_t). \tag{30}$$

For $P_t, \hat{P}_t \in \mathcal{X}^*$,

$$\sigma_{\mathcal{Q}}(P_t + \hat{P}_t) \;=\; \max_{Q\in\mathcal{Q}}(P_t(Q) + \hat{P}_t(Q)) \;\le\; \max_{Q\in\mathcal{Q}} P_t(Q) + \max_{Q\in\mathcal{Q}} \hat{P}_t(Q). \tag{31}$$

Thus $\sigma_{\mathcal{Q}}$ is positively homogeneous and subadditive, hence sublinear.

Now let $Q_t^\star \in \arg\max_{Q\in\mathcal{Q}} P_t(Q)$. Then $\sigma_{\mathcal{Q}}(P_t) = P_t(Q_t^\star)$. For any $h \in \mathcal{X}^*$,

$$\sigma_{\mathcal{Q}}(P_t + h) \;=\; \max_{Q\in\mathcal{Q}}(P_t(Q) + h(Q)) \;\ge\; P_t(Q_t^\star) + h(Q_t^\star) \;=\; \sigma_{\mathcal{Q}}(P_t) + h(Q_t^\star). \tag{32}$$

This is exactly the subgradient inequality, thus $Q_t^\star \in \partial\sigma_{\mathcal{Q}}(P_t)$.

Conversely, suppose $\bar{Q} \in \partial\sigma_{\mathcal{Q}}(P_t)$, i.e. for all $h \in \mathcal{X}^*$,

$$\sigma_{\mathcal{Q}}(P_t + h) \;\ge\; \sigma_{\mathcal{Q}}(P_t) + h(\bar{Q}). \tag{33}$$

We first show $\bar{Q} \in \mathcal{Q}$. For any $\tilde{h} \in \mathcal{X}^*$ and any $\tau > 0$, applying the inequality with $h = \tau\tilde{h}$ and combining with subadditivity and positive homogeneity of $\sigma_{\mathcal{Q}}$,

$$\sigma_{\mathcal{Q}}(P_t) + \tau\sigma_{\mathcal{Q}}(\tilde{h}) \;\ge\; \sigma_{\mathcal{Q}}(P_t + \tau\tilde{h}) \;\ge\; \sigma_{\mathcal{Q}}(P_t) + \tau\tilde{h}(\bar{Q}). \tag{34}$$

Dividing by τ gives $\sigma_{\mathcal{Q}}(\tilde{h}) \ge \tilde{h}(\bar{Q})$ for all $\tilde{h} \in \mathcal{X}^*$. Since $\mathcal{Q}$ is closed and convex, the support-function characterization implies $\bar{Q} \in \mathcal{Q}$.

Next, applying the subgradient inequality with $h = -P_t$,

$$0 \;=\; \sigma_{\mathcal{Q}}(0) \;\ge\; \sigma_{\mathcal{Q}}(P_t) - P_t(\bar{Q}), \tag{35}$$

so $P_t(\bar{Q}) \ge \sigma_{\mathcal{Q}}(P_t)$. Combined with $\bar{Q} \in \mathcal{Q}$, which gives $P_t(\bar{Q}) \le \sigma_{\mathcal{Q}}(P_t)$, we conclude $P_t(\bar{Q}) = \sigma_{\mathcal{Q}}(P_t)$, i.e., $\bar{Q} \in \arg\max_{Q\in\mathcal{Q}} P_t(Q)$. Thus $\partial\sigma_{\mathcal{Q}}(P_t) = \arg\max_{Q\in\mathcal{Q}} P_t(Q)$.

(iii) Lipschitz continuity: Let $\delta = P_t - \hat{P}_t$. Then

$$\sigma_{\mathcal{Q}}(P_t) - \sigma_{\mathcal{Q}}(\hat{P}_t) \;=\; \sup_{Q\in\mathcal{Q}} P_t(Q) - \sup_{Q\in\mathcal{Q}} \hat{P}_t(Q) \;\le\; \sup_{Q\in\mathcal{Q}}(P_t(Q) - \hat{P}_t(Q)) \;=\; \sup_{Q\in\mathcal{Q}} \delta(Q). \tag{36}$$

Swapping P_t and $\hat{P}_t$,

$$\sigma_{\mathcal{Q}}(\hat{P}_t) - \sigma_{\mathcal{Q}}(P_t) \;\le\; \sup_{Q\in\mathcal{Q}}(-\delta(Q)). \tag{37}$$

Combining both inequalities gives the stated bound. □

We get stronger uniqueness result if $\mathcal{X}$ is finite-dimensional (so that the number of hyperparameters and the number of delays are finite), or more generally if the standard regularity conditions for support functions $\sigma_{\mathcal{Q}} : \mathcal{X}^* \to \mathbb{R}$ on locally convex spaces holds. In these cases, the support function is Gâteaux differentiable at P_t:

$$D\sigma_{\mathcal{Q}}(P_t)(h) \;=\; \lim_{\tau\to 0} \frac{\sigma_{\mathcal{Q}}(P_t + \tau h) - \sigma_{\mathcal{Q}}(P_t)}{\tau} \;=\; h(Q_t^\star). \tag{38}$$

This is linear in h, so the maximizer $Q_t^\star$ is the unique subgradient of $\sigma_{\mathcal{Q}}$ at P_t. We can then simplify the construction of the greedy optimal causal map to:

$$Q_t^\star = \nabla \sigma_{\mathcal{Q}}(P_t). \tag{39}$$

This is coherent with Theorem 2.2 in Section 2 for linear optimizers.

The next corollary extends the family-hull reduction theorem of Section 2 to general causal maps.

Corollary B.3 (Family-hull reduction for arbitrary causal maps)**.** *Let $\mathcal{F} \subset \mathcal{X}$ be any nonempty set of causal maps, and let $P_t \in \mathcal{X}^*$ be the learning-power functional. Then*

$$\sup_{Q \in \mathrm{co}(\mathcal{F})} P_t(Q) \;=\; \sup_{Q \in \mathcal{F}} P_t(Q). \tag{40}$$

If $\mathcal{F}$ is compact and P_t is continuous, the supremum on the right-hand side is attained. Moreover, if a convex combination of elements of $\mathcal{F}$

$$Q^\star \;=\; \sum_{i=1}^{m} \alpha_i Q_i, \qquad \textit{where} \quad Q_i \in \mathcal{F},\ \alpha_i > 0,\ \sum_i \alpha_i = 1, \tag{41}$$

attains the supremum over $\mathrm{co}(\mathcal{F})$, then every active component Q_i with $\alpha_i > 0$ also attains the same maximal score:

$$P_t(Q_i) \;=\; \sup_{Q \in \mathcal{F}} P_t(Q). \tag{42}$$

Proof. Since $\mathcal{F} \subseteq \mathrm{co}(\mathcal{F})$,

$$\sup_{Q \in \mathrm{co}(\mathcal{F})} P_t(Q) \;\geq\; \sup_{Q \in \mathcal{F}} P_t(Q). \tag{43}$$

For the reverse inequality, let $Q \in \mathrm{co}(\mathcal{F})$. Then $Q = \sum_{i=1}^m \alpha_i Q_i$ for some $Q_i \in \mathcal{F}$, $\alpha_i \geq 0$, and $\sum_i \alpha_i = 1$. By linearity of P_t (Lemma B.1),

$$P_t(Q) \;=\; \sum_i \alpha_i P_t(Q_i) \;\leq\; \sum_i \alpha_i \sup_{Q' \in \mathcal{F}} P_t(Q') \;=\; \sup_{Q' \in \mathcal{F}} P_t(Q'). \tag{44}$$

Taking the supremum over $Q \in \mathrm{co}(\mathcal{F})$ gives

$$\sup_{Q \in \mathrm{co}(\mathcal{F})} P_t(Q) \;\leq\; \sup_{Q \in \mathcal{F}} P_t(Q). \tag{45}$$

Thus equality holds.

If $\mathcal{F}$ is compact and P_t is continuous, then P_t attains its maximum on $\mathcal{F}$.

Finally, suppose a convex combination of active components $Q^\star = \sum_{i=1}^m \alpha_i Q_i$ with $\alpha_i > 0$ and $\sum_i \alpha_i = 1$ attains the supremum. Let $P^\star = \sup_{Q \in \mathcal{F}} P_t(Q)$. Then $P_t(Q_i) \leq P^\star$ for all i, and by linearity of P_t (Lemma B.1),

$$P^\star \;=\; P_t(Q^\star) \;=\; \sum_i \alpha_i P_t(Q_i) \;\leq\; \sum_i \alpha_i P^\star \;=\; P^\star. \tag{46}$$

Since all $\alpha_i > 0$, the only way the weighted average equals $P^\star$ is if $P_t(Q_i) = P^\star$ for every active i. Thus every Q_i also attains the same maximal score $P^\star$. □

The corollary implies that the greedy alignment principle can be applied to optimize over families containing general nonlinear adaptive optimizers. If the set of causal maps $\mathcal{X}$ is convex, greedy optimizer selection is a convex optimization problem even though each individual optimizer $Q \in \mathcal{X}$ may be highly nonlinear in the gradient stream $\boldsymbol{g}$. For a nonconvex hyperparameterized family, the family-hull reduction in Corollary B.3 shows that optimizing over its convex hull gives the same value as selecting the best original candidate. Therefore, our finite K-Switch selection algorithm is exact for the convexified family.

C Closed-form solutions for five canonical families of optimizers

This section solves the optimization problem (♠) using Theorem 2.2. Exact closed form solutions for the greedy-optimal optimizer $Q^\star$ and the corresponding optimal learning power $P^\star(R)$ are obtained for the prescribed family $\mathcal{Q}$ and gradient autocorrelation sequence R. This theoretical study reveals how widely-used families of optimizers emerge as special cases of the general optimization problem. Specifically, we classify the greedy-optimal optimizers $Q^\star$ into five canonical families of causal filters $\mathcal{Q}$. As we see in the following, the prescribed family $\mathcal{Q}$ characterizes the solution optimizer $Q^\star$ and its associated hyperparameters.

We begin by observing that most common practical optimizers are symmetric. For example, every component-wise updater has a diagonal filter response Q, hence is symmetric. For time lag $k > 0$, consider the gradient autocorrelation

$$R_{t,k} \;=\; \mathbb{E}[\boldsymbol{g}_t\, \boldsymbol{g}_{t-k}^\top]. \tag{47}$$

This quantity is not generally symmetric nor PSD. However, for symmetric or PSD filter families, only its symmetric part contributes to the actual learning power. Therefore, we focus on symmetric filters by first defining the symmetric part of the autocorrelation R:

$$S_{t,k} \;\coloneqq\; \frac{R_{t,k} + R_{t,k}^\top}{2}. \tag{48}$$

$S_{t,k}$ is a symmetric matrix for each time t and delay k; it is PSD at $k = 0$ (since $S_{t,0} = R_{t,0} = \mathbb{E}[\boldsymbol{g}_t \boldsymbol{g}_t^\top]$), but for $k > 0$ it may be indefinite, motivating the use of its positive part $S_{t,k}^+$ defined below. Whenever the filter $Q_{t,k}$ is symmetric, i.e., $Q_{t,k} = Q_{t,k}^\top$, the trace of the inner products are equal, i.e.,

$$\mathrm{Tr}(Q_{t,k}^\top R_{t,k}) \;=\; \mathrm{Tr}(Q_{t,k} S_{t,k}). \tag{49}$$

Let

$$S_{t,k} = U_{t,k}\, \mathrm{diag}(\lambda_{1,t,k} \geq \cdots \geq \lambda_{d,t,k})\, U_{t,k}^\top \tag{50}$$

and denote the positive part by

$$S_{t,k}^+ = U_{t,k}\, \mathrm{diag}([\lambda_{i,t,k}]_+)\, U_{t,k}^\top. \tag{51}$$

Consider the following five types of causal filter families:

- *Hilbert ball type* $\mathcal{Q}_{\mathrm{H}}(B) = \{Q : \|Q\|_{\mathcal{H}}^2 = \sum_{k=0}^\infty \mathrm{Tr}(Q_{t,k}^\top Q_{t,k}) \leq B\}$ is the simplest and the largest family that does not favor any particular direction in the parameter space, but requires larger memory to store its hyperparameters.
- *Spectral type* $\mathcal{Q}_{\mathrm{S}}(\tau, \Lambda) = \{Q \succeq 0 : \mathrm{Tr}(Q_{t,k}) \leq \tau_k,\, 0 \preceq Q_{t,k} \preceq \Lambda_k I\, \forall k\}$ is a trust region that upper limits the (1) per-direction spectrum for safety and the (2) trace for total update budget, simultaneously at each time delay k.
- *Lyapunov type* $\mathcal{Q}_{\mathrm{L}}(B) = \{Q \succeq 0 : Q_{t,k} = \Pi_{t,k}^+ Q_{t,k} \Pi_{t,k}^+,\, \mathrm{Tr}(Q_{t,k}^\top S_{t,k}^+ Q_{t,k}) \leq B_k\, \forall k\}$, where $\Pi_{t,k}^+$ is the projection onto $\mathrm{range}(S_{t,k}^+)$, utilizes the positive symmetric part of the lag-correlation sequence itself as the metric, leading to a natural dynamic Lyapunov-like stability condition.
- *Metric type* $\mathcal{Q}_{\mathrm{M}}(B, M) = \{Q \succeq 0 : \mathrm{Tr}(Q_{t,k}^\top M_{t,k} Q_{t,k}) \leq B_k, M_k \succ 0\, \forall k\}$ is a general preconditioning optimizer that utilizes a general positive-definite metric M_k to control the learning power.
- *Diagonal type* $\mathcal{Q}_{\mathrm{D}}(B, c) = \{Q \succeq 0 : Q_{t,k} = \mathrm{diag}(q_{j,k}) \succeq 0,\, \sum_j c_{j,k} q_{j,k}^2 \leq B_k\, \forall k\}$ represents element-wise optimizers, a memory-efficient family that are commonly used in large-scale machine learning.

Instantiating the construction from Theorem 2.2 on each of these families, we obtain the closed-form greedy-optimal optimizer $Q^\star$ and the corresponding optimal learning power $P^\star(S)$.

Theorem C.1 (Closed-form solutions for causal filter families)**.** *Omit the time index t for brevity. Assume throughout that all displayed series converge absolutely. Note that the Hilbert family uses the full lag moment R. The remaining four families depend only on the symmetric autocorrelation $S_k = \mathrm{Sym}(R_k)$ and their positive parts S_k^+. The closed-form optimal solutions corresponding to each of the five causal filter families are as follows:*

(i) (Hilbert ball): *The optimal $Q^\star_{\mathrm{H}}$ is proportional to the autocorrelation R, i.e.,*

$$Q^\star_{\mathrm{H}} \;=\; \frac{\sqrt{B}}{\|R\|_{\mathcal{H}}} R, \tag{52}$$

if $R \neq 0$; otherwise any feasible Q is optimal. This gives the optimal learning power

$$P^\star_{\mathrm{H}}(R) \;=\; \sqrt{B}\,\|R\|_{\mathcal{H}}. \tag{53}$$

(ii) (Spectral): *The optimal $Q^\star_{\mathrm{S},k}$ shares the same eigenstructure as S_k but with eigenvalues $q^\star_{i,k}$ instead of $\lambda_{i,k}$:*

$$Q^\star_{\mathrm{S},k} \;=\; U_k \,\mathrm{diag}(q^\star_{i,k})\, U_k^\top. \tag{54}$$

The eigenvalues $q^\star_{i,k}$ are determined by water-filling*: Let d_k^+ be the number of positive eigenvalues of S_k and $m_k = \min\{d_k^+, \lfloor \tau_k/\Lambda_k \rfloor\}$. The eigenvalues $q^\star_{i,k}$ are then given by:*

$$q^\star_{i,k} \;=\; \begin{cases} \Lambda_k, & \text{if } i \le m_k, \\ \tau_k - m_k\Lambda_k, & \text{if } i = m_k + 1 \le d_k^+ \text{ and } \tau_k < d_k^+\Lambda_k, \\ 0, & \text{otherwise}. \end{cases} \tag{55}$$

This gives the optimal learning power

$$P^\star_{\mathrm{S}}(S) \;=\; \sum_{k=0}^{\infty} \left[\Lambda_k \sum_{i=1}^{m_k} \lambda_{i,k} + \mathbf{1}_{m_k < d_k^+} (\tau_k - m_k\Lambda_k)\, \lambda_{m_k+1,k} \right], \tag{56}$$

with the convention $\lambda_{d_k^+ + 1,k} = 0$, so the second term vanishes when $\tau_k \ge d_k^+\Lambda_k$ (i.e., $m_k = d_k^+$).

(iii) (Lyapunov): *For each delay k, define Π_k^+ as the orthogonal projection onto* $\mathrm{range}(S_k^+)$*. Then, the optimal $Q^\star_{\mathrm{L},k}$ is given by*

$$Q^\star_{\mathrm{L},k} \;=\; \alpha_k \Pi_k^+, \qquad \alpha_k \;=\; \sqrt{\frac{B_k}{\mathrm{Tr}(S_k^+)}}, \tag{57}$$

if $S_k^+ \neq 0$; otherwise $Q^\star_{\mathrm{L},k} = 0$. This gives the optimal learning power

$$P^\star_{\mathrm{L}}(S) \;=\; \sum_{k=0}^{\infty} \sqrt{B_k\, \mathrm{Tr}(S_k^+)}. \tag{58}$$

(iv) (Metric, commuting case): *For each delay k, assume that $M_k \succ 0$ and S_k commute, so that they are simultaneously diagonalizable with eigenvalues $m_{i,k} > 0$ and $\lambda_{i,k}$, respectively. Let U_k denote their common eigenbasis. Then the optimal $Q^\star_{\mathrm{M},k}$ is given by*

$$Q^\star_{\mathrm{M},k} \;=\; \alpha_k U_k\, \mathrm{diag}\left(\frac{[\lambda_{i,k}]_+}{m_{i,k}} \right) U_k^\top, \qquad \alpha_k \;=\; \sqrt{\frac{B_k}{\sum_i [\lambda_{i,k}]_+^2 / m_{i,k}}}, \tag{59}$$

if $S_k^+ \neq 0$; otherwise $Q^\star_{\mathrm{M},k} = 0$. This gives the optimal learning power

$$P^\star_{\mathrm{M}}(S) \;=\; \sum_{k=0}^{\infty} \sqrt{B_k \sum_i [\lambda_{i,k}]_+^2 / m_{i,k}}. \tag{60}$$

Equivalently, under this commutativity assumption,

$$Q^\star_{\mathrm{M},k} = \alpha_k M_k^{-1} S_k^+. \tag{61}$$

(v) (Diagonal): *For each delay k, let $s_{j,k} \coloneqq (S_k)_{jj}$ denote the diagonal entries of S_k. The optimal $Q^\star_{\mathrm{D},k}$ is a diagonal matrix with elements given by*

$$Q^\star_{\mathrm{D},k} \;=\; \mathrm{diag}(q^\star_{j,k}), \qquad q^\star_{j,k} \;=\; \alpha_k \frac{[s_{j,k}]_+}{c_{j,k}}, \qquad \alpha_k \;=\; \sqrt{\frac{B_k}{\sum_\ell [s_{\ell,k}]_+^2 / c_{\ell,k}}}, \tag{62}$$

if $\sum_\ell [s_{\ell,k}]_+ > 0$; otherwise $Q^\star_{\mathrm{D},k} = 0$. This gives the optimal learning power

$$P^\star_{\mathrm{D}}(S) \;=\; \sum_{k=0}^{\infty} \sqrt{B_k \sum_j [s_{j,k}]_+^2 / c_{j,k}}. \tag{63}$$

Proof. We apply Theorem 2.2 to each causal filter family.

(i) Hilbert ball $\mathcal{Q}_{\mathrm{H}}(B) = \{Q : \|Q\|_{\mathcal{H}} \le \sqrt{B}\}$. The Lagrangian is $L(Q, \lambda) = \langle Q, R\rangle_{\mathcal{H}} - \lambda(\|Q\|_{\mathcal{H}}^2 - B)$. Taking the gradient with respect to Q and setting to zero gives

$$\nabla_Q L \;=\; R - 2\lambda Q \;=\; 0 \quad \Rightarrow \quad Q \;=\; \frac{R}{2\lambda}. \tag{64}$$

The constraint $\|Q\|_{\mathcal{H}} = \sqrt{B}$ gives $\|R/(2\lambda)\|_{\mathcal{H}} = \sqrt{B}$, so $2\lambda = \|R\|_{\mathcal{H}}/\sqrt{B}$. Hence

$$Q_{\mathrm{H}}^\star \;=\; \sqrt{B}\,\frac{R}{\|R\|_{\mathcal{H}}}, \qquad P_{\mathrm{H}}^\star(R) \;=\; \langle Q_{\mathrm{H}}^\star, R\rangle_{\mathcal{H}} \;=\; \sqrt{B}\,\|R\|_{\mathcal{H}}. \tag{65}$$

(ii) Spectral $\mathcal{Q}_{\mathrm{S}}(\tau, \Lambda) = \{Q : \mathrm{Tr}(Q_{t,k}) \le \tau_k,\ 0 \preceq Q_{t,k} \preceq \Lambda_k I\ \forall k\}$. The problem decouples over delays k. For each k, by von Neumann's trace inequality, the maximizer shares the eigenstructure of S_k:

$$Q_k^\star = U_k\,\mathrm{diag}(q_{i,k}^\star)\,U_k^\top, \tag{66}$$

where the eigenvalues $q_{i,k}^\star$ solve the linear program

$$\max_{0 \le q_{i,k} \le \Lambda_k} \sum_i q_{i,k}\,\lambda_{i,k} \quad \text{s.t.} \quad \sum_i q_{i,k} \le \tau_k. \tag{67}$$

Let d_k^+ be the number of positive eigenvalues of S_k and $m_k = \min\{d_k^+, \lfloor \tau_k/\Lambda_k \rfloor\}$. The optimal solution allocates maximum weight Λ_k to the largest eigenvalues $\lambda_{i,k}$ until the trace budget τ_k is exhausted, giving the water-filling formula:

$$q_{i,k}^\star \;=\; \begin{cases} \Lambda_k, & \text{if } i \le m_k, \\ \tau_k - m_k\Lambda_k, & \text{if } i = m_k + 1 \le d_k^+ \text{ and } \tau_k < d_k^+\Lambda_k, \\ 0, & \text{otherwise.} \end{cases} \tag{68}$$

(iii) Lyapunov $\mathcal{Q}_{\mathrm{L}}(B) = \{Q : Q_{t,k} \succeq 0,\ Q_{t,k} = \Pi_{t,k}^+ Q_{t,k} \Pi_{t,k}^+,\ \mathrm{Tr}(Q_{t,k}^\top S_{t,k}^+ Q_{t,k}) \le B_k\ \forall k\}$. The problem decouples over delays k. For each k, the Lagrangian on the support of S_k^+ is

$$L_k(Q_k, \mu_k) \;=\; \mathrm{Tr}(Q_k^\top S_k^+) - \mu_k\big(\mathrm{Tr}(Q_k^\top S_k^+ Q_k) - B_k\big). \tag{69}$$

The first-order condition gives

$$S_k^+ - 2\mu_k S_k^+ Q_k \;=\; 0 \quad \Rightarrow \quad S_k^+(I - 2\mu_k Q_k) \;=\; 0. \tag{70}$$

This implies $Q_k = \frac{1}{2\mu_k} I$ on the support of S_k^+, i.e., $Q_k = \alpha_k \Pi_k^+$ where $\alpha_k = \frac{1}{2\mu_k}$ and Π_k^+ is the projection onto $\mathrm{range}(S_k^+)$. Since L_k is concave in Q_k for $\mu_k > 0$ (as $S_k^+ \succeq 0$), the first-order condition gives the global maximum. Using the constraint:

$$\mathrm{Tr}(Q_k^\top S_k^+ Q_k) \;=\; \alpha_k^2\,\mathrm{Tr}(S_k^+) \;=\; \alpha_k^2 \sum_i [\lambda_{i,k}]_+ \;=\; B_k. \tag{71}$$

Therefore, $\alpha_k = \sqrt{B_k / \sum_i [\lambda_{i,k}]_+}$ if $\sum_i [\lambda_{i,k}]_+ > 0$; otherwise $Q_{\mathrm{L},k}^\star = 0$.

(iv) Metric $\mathcal{Q}_{\mathrm{M}}(B, M) = \{Q : Q_{t,k} \succeq 0,\ \mathrm{Tr}(Q_{t,k}^\top M_{t,k} Q_{t,k}) \le B_k,\ M_{t,k} \succ 0\,\forall k\}$. The problem decouples over delays k. For each k, assume M_k and S_k commute with common eigenbasis U_k and eigenvalues $m_{i,k} > 0$ and $\lambda_{i,k}$ respectively. The Lagrangian is

$$L_k(Q_k, \mu_k) \;=\; \mathrm{Tr}(Q_k^\top S_k) - \mu_k\big(\mathrm{Tr}(Q_k^\top M_k Q_k) - B_k\big). \tag{72}$$

Since $Q_k \succeq 0$, only positive eigenvalues $[\lambda_{i,k}]_+$ contribute to the objective. The first-order condition in the common eigenbasis gives

$$[\lambda_{i,k}]_+ - 2\mu_k m_{i,k} q_{i,k} \;=\; 0 \quad \Rightarrow \quad q_{i,k} \;=\; \frac{[\lambda_{i,k}]_+}{2\mu_k m_{i,k}}. \tag{73}$$

Using the constraint $\sum_i m_{i,k} q_{i,k}^2 = B_k$:

$$\sum_i m_{i,k} \left(\frac{[\lambda_{i,k}]_+}{2\mu_k m_{i,k}}\right)^2 \;=\; \frac{1}{4\mu_k^2} \sum_i \frac{[\lambda_{i,k}]_+^2}{m_{i,k}} \;=\; B_k. \tag{74}$$

Therefore, $q^\star_{i,k} = \alpha_k[\lambda_{i,k}]_+/m_{i,k}$ where $\alpha_k = \sqrt{B_k/\sum_i [\lambda_{i,k}]_+^2/m_{i,k}}$ if $\sum_i[\lambda_{i,k}]_+ > 0$; otherwise $Q^\star_{\mathrm{M},k} = 0$.

(v) Diagonal $\mathcal{Q}_{\mathrm{D}}(B,c) = \{Q : Q_{t,k} = \mathrm{diag}(q_{j,k}),\ q_{j,k} \geq 0,\ \sum_j c_{j,k} q_{j,k}^2 \leq B_k\ \forall k\}$. The problem decouples over delays k. Since Q_k is diagonal, $\langle Q_k, S_k\rangle$ depends on S_k only through its diagonal entries; write $s_{j,k} := (S_k)_{jj}$. Since $q_{j,k} \geq 0$, only positive diagonal entries $[s_{j,k}]_+$ contribute to the objective. For each k, we solve:

$$\max_{q_{j,k}\geq 0} \sum_j [s_{j,k}]_+ q_{j,k} \quad \text{s.t.} \quad \sum_j c_{j,k} q_{j,k}^2 \leq B_k. \tag{75}$$

By Cauchy–Schwarz:

$$\sum_j [s_{j,k}]_+ q_{j,k} \;=\; \sum_j (\sqrt{c_{j,k}} q_{j,k}) \frac{[s_{j,k}]_+}{\sqrt{c_{j,k}}} \;\leq\; \left(\sum_j c_{j,k} q_{j,k}^2\right)^{1/2} \left(\sum_j \frac{[s_{j,k}]_+^2}{c_{j,k}}\right)^{1/2}. \tag{76}$$

Equality holds when $q^\star_{j,k} \propto [s_{j,k}]_+/c_{j,k}$. Normalizing by the constraint gives the stated result. □

These analytic solutions reveal how the characteristics of different types of optimal dynamic optimizers $Q^\star$ are induced by controlling the feasible set $\mathcal{Q}$.

Hilbert family ↔ Matched causal filter. Budget $\mathcal{Q}_{\mathrm{H}}(B)$ produces a *matched filter* that aligns learning power with lag-correlation evidence $Q^\star \propto R$. It enjoys implementation simplicity but potentially over-concentrates on dominant temporal modes. As in Theorem 3.1, we can project this general class of optimizers into special geometries to calculate for the optimal hyperparameters. Momentum-based family is one canonical example of constraining this family to a one-pole filter, i.e., momentum is a one-pole projection of the matched gradient-statistics filter. This idea aligns with Corollary 3.2 in the main manuscript.

Per-lag spectral family ↔ Water-filling over positive active-power modes. Budget $\mathcal{Q}_{\mathrm{S}}(\tau,\Lambda)$ produces a *water-filling* dynamic optimizer that allocates budget to positive active-power modes until hitting the safety cap Λ_k. The spectral cap Λ_k acts as a *per-mode safety mechanism* analogous to *gradient clipping*, while the trace constraint τ_k controls the *per-lag learning-power budget* analogous to *learning rate scheduling*. This suggests that clipping and learning rate scheduling can be viewed as consequences of spectral constraints in the greedy alignment problem. Furthermore, we also observe a spectral analogue of *cautious alignment* [41]: negative active-power modes receive no budget under PSD-constrained filter families.

Lyapunov family ↔ Equal-power dynamic optimizer. Budget $\mathcal{Q}_{\mathrm{L}}(B)$ produces an *equal-power* dynamic optimizer that equalizes learning power uniformly across informative (positive) lag-correlation directions, preventing over-concentration while maintaining temporal efficiency. This produces a natural dynamic preconditioning effect, with uniform power allocation across all informative temporal directions.

Metric family ↔ General preconditioning optimizer. Budget $\mathcal{Q}_{\mathrm{M}}(B,M)$ produces a *general preconditioning* optimizer that adapts the learning rate statically or dynamically based on the metric M_k. This provides an abstract template for many existing preconditioning methods, such as K-FAC [47], Shampoo [29], and natural gradient descent [1].

Diagonal family ↔ Coordinate-wise dynamic optimizer ∼ Adam. Budget $\mathcal{Q}_{\mathrm{D}}(B,c)$ produces a *coordinate-wise* dynamic optimizer that adapts per-coordinate learning power proportional to the positive diagonal evidence $[s_{j,k}]_+$ and inversely proportional to the costs c_j. When $c_j \propto \sqrt{v_{\beta_2,t,j} + \epsilon}$ ($v_{\beta_2,t,j}$ being the EMA of g_j^2 at time t), this recovers the core mechanism of *Adam*: coordinate update is proportional to evidence divided by cost. This is elaborated in Corollary 3.3 in the main manuscript.

These closed-form solutions show that the optimizer family $\mathcal{Q}$ is not an arbitrary constraint: it is the inductive bias that determines how learning power, i.e., the expected loss drop rate $P^\star(R) = -\mathbb{E}[\mathrm{d}\mathcal{L}/\mathrm{d}t]$, is allocated across temporal and parameter-space modes. Different feasible geometries

Table 8: Hyperparameters and settings for math finetuning experiments on Gemma-2B [26] with LoRA [31]. Values reflect the experimental script.

Parameter	Value(s) / Description
Dataset	MetaMathQA-395K
Training subset size	100,000
Models tested	Gemma-2B (`google/gemma-2b`)
Hardware	$1 \times$ A100-80GB
Precision	Bfloat16 (BF16)
Optimizer	AdamW [36, 44]
Optimal AdamW-type optimizer	*two-option switch* with β_1 endpoints of 0.8 and 0.99
Epochs	1
Batch size (bs)	32
Learning rate (lr)	2×10^{-4}
Weight decay	0
Warmup ratio	0
Adapter type	LoRA [31]
LoRA Rank (r)	32
LoRA Scaling (α)	4
LoRA Dropout	0.05
Cutoff length	256
Adapter target modules	`q_proj, k_proj, v_proj, o_proj` `down_proj, up_proj, gate_proj`

allocate the same learning power differently, reflecting the different inductive biases of each family. The Hilbert family yields a matched filter, the spectral family yields water-filling under safety caps, the Lyapunov family equalizes power on informative modes, and the diagonal family recovers Adam-like cost-weighted coordinate adaptation.

D Experimental details

This section provides implementation details and full results of the experiments conducted to validate the theory in Section 3. First, we provide the hyperparameters and settings for the experiments in Tables 8, 9, 10, 11, and 12 of Appendix D.1. Additional experimental results are provided in Tables 13, 14, 15, and 16 of Appendix D.2. Our automatic hyperparameter tuning shows comparable performance across all datasets and models, including conventional residual networks [30], vision transformers [20], and modern large language models [26, 28] with or without using parameter-efficient fine-tuning methods like low-rank adaptation (LoRA) [31]. This demonstrates the practical usefulness of our framework.

D.1 Implementation details

Increasing the effective averaging horizon with EMA. During optimization, the gradients $\boldsymbol{g}$ and the parameter updates $\boldsymbol{u}$ vary over time, and the exact expectation $\mathbb{E}$ in the argmax objective $A_{k,t}$ is unknown. In the main implementation, we use the instantaneous mini-batch score $\boldsymbol{g}^\top \boldsymbol{u}$ to approximate the expected alignment. The effective averaging horizon can be further increased without additional computational overhead by maintaining an EMA of the argmax objective $A_{k,t}$. We use an EMA with decay factor $\alpha = 0.9$ for Adam/AdamW experiments in the main manuscript. Algorithm 2 omits the EMA implementation for brevity.

State decay under persistent negative alignment. Section 3.2 shows that the greedy alignment score $A = \boldsymbol{g}^\top \boldsymbol{u}$ is a faithful surrogate for the actual one-step drop rate in loss within a local regime where $\eta\|\boldsymbol{u}\|$ is small. Outside this regime, negative alignments $A < 0$ can occur [41, 10] when component-wise opposition dominates component-wise alignment: $A = \sum_j [g_j u_j]_+ - \sum_j [-g_j u_j]_+$ and $|\sum_j [-g_j u_j]_+| > |\sum_j [g_j u_j]_+|$. We thus treat a negative alignment as an indicator of low confidence in the accumulated optimizer states, rather than as a proof that every coordinate is harmful, and we apply a soft attenuation by halving the optimizer states, e.g., the running momentum, whenever a negative alignment $A < 0$ persists for 5 consecutive steps.

Table 9: Hyperparameters and settings for math finetuning experiments on Llama-3-8B [28] with LoRA [31]. Values reflect the experimental script.

Parameter	**Value(s) / Description**
Dataset	MetaMathQA-395K
Training subset size	100,000
Models tested	Llama-3-8B (`meta-llama/llama-3-8b`)
Hardware	$1 \times$ A100-80GB
Precision	Bfloat16 (BF16)
Optimizer	AdamW [36, 44]
Optimal AdamW-type optimizer	*two-option switch* with β_1 endpoints of 0.8 and 0.99
Epochs	1
Batch size (bs)	32
Learning rate (lr)	1×10^{-4}
Weight decay	0
Warmup ratio	0
Adapter type	LoRA [31]
LoRA Rank (r)	32
LoRA Scaling (α)	4
LoRA Dropout	0.05
Cutoff length	256
Adapter target modules	`q_proj, k_proj, v_proj, o_proj` `down_proj, up_proj, gate_proj`

ResNet-18 on CIFAR-100. For ResNet-18 [30] on CIFAR-100 [38] experiments, we follow the standard settings of [30]: 300 epochs with a learning rate decay of 0.1 at epochs 60, 120, and 160. All hyperparameters other than momentum are held fixed. We use a weight decay of 5×10^{-4}, batch size of 128, and a base learning rate of 0.1 for SGD and 0.01 for Adam [36]. For our optimal Adam-type optimizers using the *two-option switch*, we use β_1 options of 0.8 and 0.999, to ensure that these endpoints enclose the typical range of β_1 values used in practice. For our optimal SGD+Momentum-type optimizers using the *two-option switch*, we use momentum options of 0.8 and 0.99, again to ensure that the endpoints enclose the typical range of momentum values in practice. For optimal SGD+Momentum-type optimizers using the *five-option switch*, we use momentum options of 0.9, 0.95, 0.98, 0.99, and 0.995, to demonstrate the effectiveness of fine-grained control in dynamic hyperparameter tuning. In the main manuscript, we show only the results of the *two-option switch* for SGD+Momentum and Adam, which incurs lower overhead than the five-option variant. Our *five-option switch* for SGD+Momentum demonstrates that we can achieve *significantly better performance* than the baseline optimizer with fixed hyperparameters ($77.57\% \rightarrow 78.33\%$ test accuracy). These extended results are summarized in Tables 13 and 14 in the next section. However, current implementation of multi-option switch larger than two requires a significant amount of computation time (around +100% compared to the baseline) and memory usage. Therefore, we did not include the results in the main manuscript, opening up a future direction for more efficient implementation.

Gemma-2B and Llama-3-8B on MetaMathQA-395K. We summarize the hyperparameters and training settings for these experiments that produced Table 3 and Table 4 in Section 3 of the main manuscript. We use low-rank adaptation (LoRA) [31] with a rank of 32 and a scaling factor of 4 for both Gemma-2B [26] and Llama-3-8B [28]. We truncate the MetaMathQA-395K [67] training dataset to 100,000 examples for both models. The reported test accuracy is based on a separate, validation-only dataset, GSM8K [17]. Additional experimental results are provided in Table 15, where we show results for baseline optimizers with different β_1 values we have tested. In the main manuscript, only the best baseline optimizer results are shown for brevity: $\beta_1 = 0.5$ for Gemma-2B and $\beta_1 = 0.9$ for Llama-3-8B. For our optimal AdamW-type optimizers using the *two-option switch*, we use β_1 options of 0.8 and 0.99, to ensure that these endpoints enclose the typical range of β_1 values used in practice.

Gemma-2B on Commonsense-170K. We summarize the hyperparameters and training settings for these experiments in Table 5 in Section 3 of the main manuscript. We use low-rank adaptation

Table 10: Hyperparameters and settings for the main commonsense finetuning experiments on Gemma-2B [26] with LoRA [31].

Parameter	Value(s) / Description
Dataset	Commonsense-170K [32]
Models tested	Gemma-2B [26]
Hardware	$1 \times$ A100-80GB
Precision	Bfloat16 (BF16)
Optimizer	AdamW [36, 44]
Optimal AdamW-type optimizer	*two-option switch* with β_1 endpoints of 0.8 and 0.95
Epochs	1
Batch size (bs)	32
Learning rate (lr)	2×10^{-4}
Weight decay	0
Warmup ratio	0
Adapter type	LoRA [31]
LoRA Rank (r)	32
LoRA Scaling (α)	4
LoRA Dropout	0.05
Cutoff length	256
Adapter target modules	`q_proj, k_proj, v_proj, o_proj down_proj, up_proj, gate_proj`

Table 11: Hyperparameters and settings for the main commonsense finetuning experiments on Gemma-2B [26] with full fine-tuning.

Parameter	Value(s) / Description
Dataset	Commonsense-170K [32]
Models tested	Gemma-2B [26]
Hardware	$1 \times$ A100-80GB
Precision	Bfloat16 (BF16)
Optimizer	AdamW [36, 44]
Optimal AdamW-type optimizer	*two-option switch* with β_1 endpoints of 0.1 and 0.99
Epochs	1
Batch size (bs)	32
Learning rate (lr)	1×10^{-5}
Weight decay	0
Warmup ratio	0
Adapter type	Full fine-tuning
Cutoff length	256

(LoRA) [31] with a rank of 32 and a scaling factor of 4 for Gemma-2B [26]. We also demonstrate our optimizer with full fine-tuning on the Commonsense-170K [32] dataset. Additional experimental results are provided in Table 16 in the next section, where we show results for baseline optimizers with different β_1 values we have tested. In the main manuscript, only the best baseline optimizer results are shown for brevity: $\beta_1 = 0.95$ for LoRA and $\beta_1 = 0.5$ for full fine-tuning. For our optimal AdamW-type optimizers using the *two-option switch*, we use β_1 options of 0.1 and 0.99 for full fine-tuning and β_1 options of 0.8 and 0.95 for LoRA, to ensure that these endpoints enclose the typical range of β_1 values used in each type of experiment in practice. After fitting the Commonsense-170K [32] dataset, we evaluate performance on various reasoning datasets that are commonly used in the LLM literature. These include BoolQ [16], PIQA [6], Social IQA [54], HellaSwag [69], Winogrande [53], and OBQA [32]. We also report the average performance across all datasets.

ViT-B and ViT-L on various classification datasets. We summarize the hyperparameters and training settings for these experiments in Table 6 in Section 3 of the main manuscript. We tested two different configurations: ViT-Base [20] with rank-32 LoRA and ViT-Large with rank-8 LoRA. The same pre-trained weights are fine-tuned for various task-specific datasets, including Stanford Cars [37], CIFAR-100 [38], CUB-200 [61], DTD [15], Food-101 [7], RESISC45 [13], and SUN397 [64]. For our optimal AdamW-type optimizers using the *two-option switch*, we use β_1 options of 0.5 and 0.99

Table 12: Common hyperparameters and settings for ViT-Base and ViT-Large [20] finetuning experiments across various classification datasets.

Parameter	**Value(s) / Description**
Models tested	ViT-B / ViT-L [20]
Datasets	Stanford Cars, CIFAR-100, CUB-200, DTD, Food-101, RESISC45, SUN397
Hardware	$1 \times$ RTX 5090-32GB
Precision	Bfloat16 (BF16)
Optimizer	AdamW [36, 44]
Optimal AdamW-type optimizer	*two-option switch* with β_1 endpoints of 0.5 and 0.99
Batch size (bs)	256
Epochs	Stanford Cars: 20
	CIFAR-100: 7
	CUB-200: 25
	DTD: 25
	Food-101: 10
	RESISC45: 10
	SUN397: 15
Learning rate (lr)	Stanford Cars: 5×10^{-3}
	CIFAR-100: 1×10^{-3}
	CUB-200: 2×10^{-3}
	DTD: 2×10^{-3}
	Food-101: 2×10^{-3}
	RESISC45: 2×10^{-3}
	SUN397: 1×10^{-3}
Weight decay	0
Warmup ratio	0
Adapter type	LoRA [31]
LoRA Rank (r)	8, 32
LoRA Scaling (α)	4
LoRA Dropout	0
Target modules	`query`, `value`

Table 13: Full test results of SGD+Momentum on CIFAR-100 with ResNet-18. mean $\pm$ std.

Method	Test acc. %	Train loss
$\beta = 0.01$	74.93 ± 0.11	0.0091 ± 0.0002
$\beta = 0.1$	75.76 ± 0.09	0.0093 ± 0.0001
$\beta = 0.2$	75.89 ± 0.08	0.0098 ± 0.0001
$\beta = 0.5$	76.21 ± 0.17	0.0115 ± 0.0001
$\beta = 0.8$	77.26 ± 0.12	0.0119 ± 0.0002
$\beta = 0.9$	77.57 ± 0.09 🥉	0.0078 ± 0.0001 🥈
$\beta = 0.95$	76.57 ± 0.32	0.0127 ± 0.0004
$\beta = 0.98$	69.79 ± 0.86	0.1648 ± 0.0148
$\beta = 0.99$	55.62 ± 5.32	1.3609 ± 0.2312
$\beta = 0.995$	62.54 ± 3.49	0.7271 ± 0.0892
$\beta = 0.999$	68.48 ± 3.25	0.1772 ± 0.0229
Ours (2 options)	78.06 ± 0.07 🥈	0.0080 ± 0.0001 🥉
Ours (5 options)	78.33 ± 0.49 🥇	0.0073 ± 0.0001 🥇

Table 14: Full test results of Adam on CIFAR-100 with ResNet-18. mean $\pm$ std.

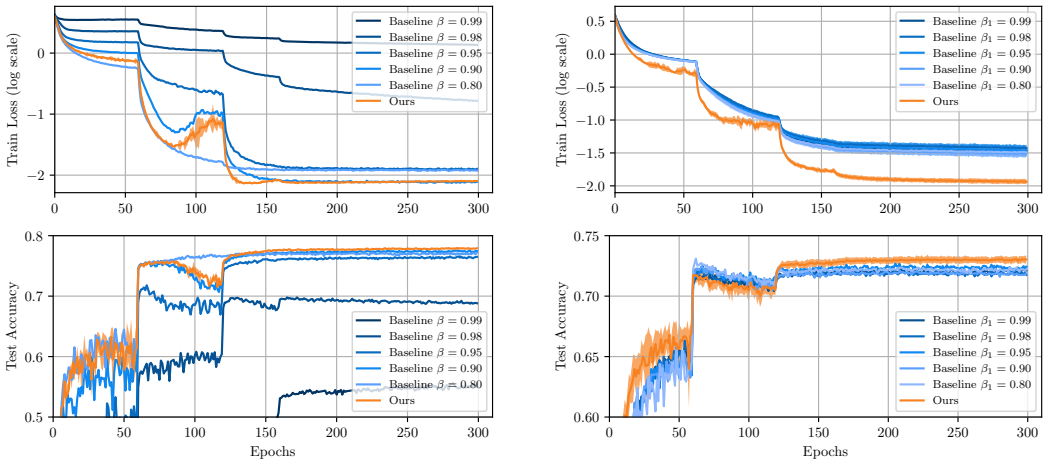
Method	Test acc. %	Train loss
$\beta_1 = 0.1$	72.78 ± 0.43	0.0414 ± 0.0037
$\beta_1 = 0.2$	72.65 ± 0.18	0.0396 ± 0.0023
$\beta_1 = 0.5$	72.86 ± 0.14 🥉	0.0351 ± 0.0019
$\beta_1 = 0.8$	73.20 ± 0.21 🥈	0.0324 ± 0.0042 🥉
$\beta_1 = 0.9$	72.85 ± 0.38	0.0314 ± 0.0044 🥈
$\beta_1 = 0.95$	72.78 ± 0.38	0.0347 ± 0.0068
$\beta_1 = 0.98$	72.69 ± 0.20	0.0372 ± 0.0038
$\beta_1 = 0.99$	72.45 ± 0.20	0.0333 ± 0.0045
Ours (2 options)	73.26 ± 0.31 🥇	0.0115 ± 0.0010 🥇

for all configurations. This covers the working range of β_1 values used in practice for each type of experiment. We also report the average performance across all datasets.

D.2 Extended experimental results

This section provides additional experimental results for the main manuscript. For visual clarity, we use gold, silver, and bronze medals to denote the best, second-best, and third-best results, respectively, in all tables hereafter.

ResNet-18 on CIFAR-100. Figures 2 and 3 in the main manuscript demonstrate how optimizer hyperparameters affect the final training loss and validation accuracy, and how our optimal optimizers



(a) Training curve of SGD+M optimizers.

(b) Training curve of Adam optimizers.

Figure 4: Demonstration of Theorem 3.1 and Corollary 3.3. Our instantiations of optimal optimizers are compared with baselines having fixed hyperparameters on the CIFAR-100 dataset [38] with ResNet-18 [30], following the standard settings of [30]. The line and shaded area indicate the mean and standard deviation over 10 runs. For clear visualization, each baseline plot shows only the best run. For SGD+Momentum, momentum below 0.8 showed suboptimal performance.

Table 15: Full test results of Gemma-2B trained with MetaMathQA-395K, validated on GSM8K. mean $\pm$ std.

Method	GSM8K acc. (%)	Train loss
$\beta_1 = 0.5$	52.57 ± 1.10	0.2080 ± 0.0004
$\beta_1 = 0.8$	52.31 ± 1.00	0.2080 ± 0.0004
$\beta_1 = 0.9$	51.76 ± 0.99	0.2081 ± 0.0004
$\beta_1 = 0.95$	51.12 ± 0.77	0.2085 ± 0.0004
$\beta_1 = 0.98$	51.25 ± 0.38	0.2093 ± 0.0005
$\beta_1 = 0.99$	50.97 ± 0.68	0.2103 ± 0.0004
Ours	52.77 ± 0.93	0.2084 ± 0.0003

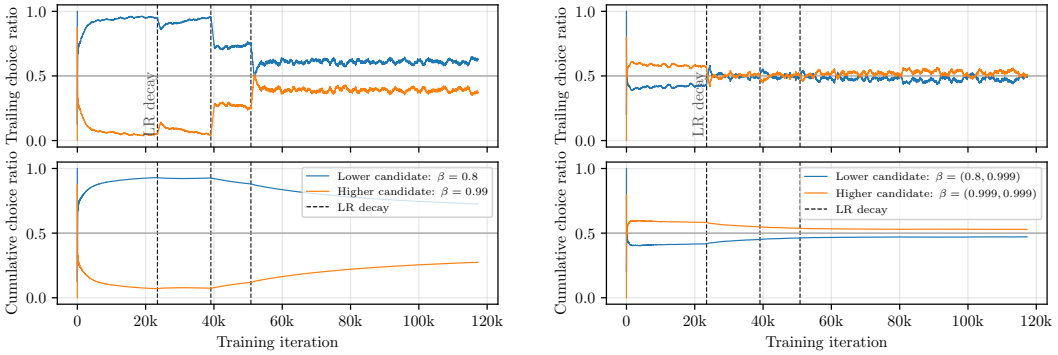
compare to baseline optimizers with fixed hyperparameters. Here, the complete results are provided in Tables 13 and 14. Figure 4 further compares the training curves of baseline optimizers and our optimal optimizers. For clarity, each baseline plot shows only the best run, and only baselines with robust momentum values are displayed. Specifically, we show results for $\beta \in [0.8, 0.99]$ for SGD+Momentum and $\beta_1 \in [0.8, 0.999]$ for Adam [36]. Although the abrupt learning rate decay of the scheduler introduces perturbations that are not considered in our theory, our implementation of the greedy optimal optimizer generally reduces the training loss rapidly and achieves better performance than the baselines. Moreover, the complete results in Tables 13 and 14 show that by increasing the number of selectable options from two to five, our instantiation of the greedy optimal optimizers achieves significantly better performance than baseline optimizers with any fixed hyperparameters. This opens up new opportunities for research into the *dynamic hyperparameter tuning* framework, which is first enabled by our theory.

Gemma-2B and Llama-3-8B on MetaMathQA-395K. We extend the experimental results in Table 3 of the main manuscript by providing the full baseline results in Table 15. The results show that our optimal optimizer yields a training loss comparable to the best baseline optimizer, while achieving better validation accuracy. Importantly, this achievement does not result from tedious manual hyperparameter tuning, but from our *dynamic hyperparameter tuning* framework enabled by our theory.

Gemma-2B on Commonsense-170K. An extended version of the results in Table 5 of the main manuscript is provided in Table 16. As in the previous experiments, our optimal optimizer achieves comparable and occasionally better performance than the best baseline optimizer with fixed hyperpa-

Table 16: Full test results of Gemma-2B trained with Commonsense-170K. mean ± std.

Gemma-2B (LoRA)	BoolQ	PIQA	Social IQA	HellaSwag	Winogrande	OBQA	**Avg**
$\beta_1 = 0.5$	65.25 ± 0.28	78.73 ± 0.27	73.97 ± 0.50	72.81 ± 1.54	70.96 ± 0.90	72.07 ± 0.34	72.30 ± 0.32
$\beta_1 = 0.8$	65.42 ± 0.20	78.80 ± 0.71	73.51 ± 0.23	73.46 ± 1.21	71.43 ± 0.28	72.20 ± 0.34	72.47 ± 0.25
$\beta_1 = 0.9$	65.31 ± 0.27	78.87 ± 0.67	73.66 ± 0.37	72.97 ± 1.47	71.40 ± 0.30	73.20 ± 0.65	72.57 ± 0.30
$\beta_1 = 0.95$	65.69 ± 0.29	78.93 ± 0.49	73.61 ± 0.16	74.07 ± 0.28	71.61 ± 0.44	72.67 ± 0.68	72.76 ± 0.17
$\beta_1 = 0.98$	65.65 ± 0.27	78.82 ± 0.40	73.52 ± 0.34	68.47 ± 1.83	71.45 ± 0.21	71.67 ± 1.23	71.60 ± 0.38
$\beta_1 = 0.99$	65.48 ± 0.43	78.69 ± 0.56	73.13 ± 0.15	68.66 ± 0.95	72.01 ± 0.52	73.00 ± 0.43	71.83 ± 0.23
Ours	65.31 ± 0.04	79.00 ± 0.36	73.58 ± 0.06	75.09 ± 1.02	71.80 ± 0.39	73.27 ± 1.15	73.01 ± 0.27
Gemma-2B (Full FT)	BoolQ	PIQA	Social IQA	HellaSwag	Winogrande	OBQA	**Avg**
$\beta_1 = 0.5$	62.79 ± 0.27	74.12 ± 0.26	66.63 ± 0.33	40.50 ± 1.15	61.48 ± 0.32	62.60 ± 1.02	61.35 ± 0.27
$\beta_1 = 0.8$	62.50 ± 0.22	72.62 ± 0.49	64.02 ± 0.19	40.11 ± 0.21	54.38 ± 0.62	57.20 ± 0.71	58.47 ± 0.19
$\beta_1 = 0.9$	62.42 ± 0.24	72.05 ± 0.38	62.88 ± 0.54	39.45 ± 0.33	52.28 ± 0.26	55.47 ± 0.52	57.43 ± 0.16
$\beta_1 = 0.95$	62.38 ± 0.20	71.60 ± 0.16	62.20 ± 0.36	39.14 ± 0.16	51.33 ± 0.48	54.13 ± 0.98	56.80 ± 0.20
$\beta_1 = 0.98$	62.42 ± 0.18	70.84 ± 0.65	61.19 ± 0.24	38.10 ± 0.24	51.09 ± 0.15	53.07 ± 0.34	56.12 ± 0.14
$\beta_1 = 0.99$	62.25 ± 0.01	70.82 ± 0.64	60.70 ± 0.25	37.41 ± 0.55	50.72 ± 0.43	52.87 ± 1.09	55.80 ± 0.24
Ours	63.29 ± 0.78	75.70 ± 0.22	68.41 ± 0.69	42.47 ± 1.06	62.46 ± 4.64	64.40 ± 0.86	62.79 ± 0.83



(a) SGDM: selection ratio over training.

(b) Adam: selection ratio over training.

Figure 5: Selection ratios of the candidate hyperparameters over training for SGDM and Adam. The experiments are the same ResNet-18 / CIFAR-100 runs in Figures 2 and 3.

rameters. In the main manuscript, we provided an abbreviated version of this table that only includes the best baseline optimizer. For reference, the best baseline is $\beta_1 = 0.95$ for LoRA training and $\beta_1 = 0.5$ for full fine-tuning.

Selected-β and score trajectories. We additionally provide visualizations of the selected hyperparameters and selection ratios over training for SGD+Momentum and Adam for the experiments in Figures 2 and 3. Our adaptive K-Switch framework rapidly switches between the optimizer hyperparameters, and the candidate preferences drift along the learning process. Figure 5 shows this drift of the cumulative selection ratios of each candidate over training. We observe gradual changes in the optimizers' hyperparameter preferences towards equal selection ratios (0.5) as training progresses, while learning rate decay abruptly changes the preferences of each candidate.

Table 17: Summary of the theorems presented in the main text. The proof of each theorem is linked next to its title with button: [proof]. All the proofs are located in Appendix E.

Result	What it says	In plain terms
Prop. 2.1	$P_t(Q) = \langle Q, R_t \rangle_{\mathcal{H}}$ under A1–A3	Alignment is a filter–statistics inner product.
Thm. 2.2	$\max_{Q \in \mathcal{Q}} P$ is the support function of $\mathcal{Q}$	A compact convex family has a greedy optimum, stable in R.
Cor. 2.3	$\sup_{\mathrm{cl\,co}(\mathcal{F} \cup \{0\})} P = (\sup_{\mathcal{F}} P)_+$	Scoring in the hyperparameterized set $\mathcal{F}$ is exact.
Thm. 3.1	$\beta^\star = \arg\max_\beta \sqrt{1-\beta^2}\, \mathbb{E}[\boldsymbol{g}^\top \boldsymbol{m}_\beta]$	Hilbert-energy-normalized SGDM selection.
Cor. 3.3	state-conditioned closed form for (β_1, β_2)	Hilbert-energy-normalized Adam/AdamW selection.
Thm. 3.4	$\lvert D_\eta - \eta A \rvert \le L_s \eta^2 \lVert \boldsymbol{u} \rVert^2/2$	Alignment is the first-order loss drop; η must stay local.
Cor. 3.5	D_η gap $L_s \eta^2 \rho^2$ on $\lVert \boldsymbol{u} \rVert \le \rho$	Greedy alignment is locally near-optimal for bounded updates.
Prop. 4.2	gap $> 2\varepsilon \Rightarrow$ correct $\arg\max$	Instantaneous scores work if the filter difference > noise.

E Proofs omitted from the main text

This section presents the proofs omitted from the main text. The proofs are organized in the same order as the theorems appear in the main manuscript.

Roadmap. Proposition 2.1 expands the causal convolution and uses Fubini–Tonelli to identify the filter–moment pairing. Theorem 2.2 recognizes the objective as the support function of a compact convex family. Corollary 2.3 is linearity over the closed convex hull. Theorem 3.1 and Corollary 3.3 score the unit-energy Hilbert or weighted-diagonal section of the one-pole family. Theorem 3.4 is the descent lemma. Corollary 3.5 establishes local near-optimality of greedy alignment over norm-bounded candidate updates. Proposition 4.2 transfers the Lipschitz bound to finite-sample scores. The basic strategies are also summarized in Table 17.

E.1 Proof of Proposition 2.1: Learning power is inner product

Proof of Proposition 2.1. Here we provide a detailed derivation of equation 8 that was abbreviated in the main text. We assume (A1) the gradient stream $\boldsymbol{g}_t$ has finite and uniformly bounded second moments: $M := \sup_{t \in \mathbb{Z}} \mathbb{E}\|\boldsymbol{g}_t\|^2 < \infty$, (A2) the optimizer filter Q_t is absolutely summable in operator norm: $\sum_{k=0}^{\infty} \|Q_{t,k}\|_{\mathrm{op}} < \infty$, and (A3) the lag-$k$ moment sequence $R_t = \{R_{t,k}\}_{k \ge 0}$ belongs to $\mathcal{H}$, so that the final pairing is a Hilbert inner product. The lag-k moment is defined as $R_{t,k} = \mathbb{E}[\boldsymbol{g}_t\, \boldsymbol{g}_{t-k}^\top]$. We start from the convolution definition of the dynamic optimizer:

$$\boldsymbol{u}_t = \sum_{k=0}^{\infty} Q_{t,k}\, \boldsymbol{g}_{t-k}. \tag{77}$$

Then we can calculate the instantaneous power as follows:

$$\begin{aligned}
P_t(Q) &= \mathbb{E}\big[\boldsymbol{g}_t^\top \boldsymbol{u}_t\big] && (78)\\
&= \mathbb{E}\Big[\boldsymbol{g}_t^\top \sum_{k=0}^{\infty} Q_{t,k}\, \boldsymbol{g}_{t-k}\Big] && (79)\\
&= \sum_{k=0}^{\infty} \mathbb{E}\big[\boldsymbol{g}_t^\top Q_{t,k}\, \boldsymbol{g}_{t-k}\big] \quad \text{(linearity of } \mathbb{E}) && (80)\\
&= \sum_{k=0}^{\infty} \mathrm{Tr}\big(Q_{t,k}\, \mathbb{E}[\boldsymbol{g}_{t-k}\, \boldsymbol{g}_t^\top]\big) && (81)\\
&= \sum_{k=0}^{\infty} \mathrm{Tr}\big(Q_{t,k}^\top\, \mathbb{E}[\boldsymbol{g}_t\, \boldsymbol{g}_{t-k}^\top]\big) \quad \text{(trace transpose)} && (82)\\
&= \sum_{k=0}^{\infty} \mathrm{Tr}\big(Q_{t,k}^\top\, R_{t,k}\big) && (83)\\
&= \langle Q_t, R_t \rangle_{\mathcal{H}}. && (84)
\end{aligned}$$

The interchange of summation and expectation is justified by Fubini–Tonelli as follows. By Cauchy–Schwarz,

$$\mathbb{E}\big|\boldsymbol{g}_t^\top Q_{t,k}\,\boldsymbol{g}_{t-k}\big| \le \|Q_{t,k}\|_{\mathrm{op}}\sqrt{\mathbb{E}\|\boldsymbol{g}_t\|^2}\sqrt{\mathbb{E}\|\boldsymbol{g}_{t-k}\|^2} \le M\|Q_{t,k}\|_{\mathrm{op}}. \tag{85}$$

Therefore,

$$\sum_{k=0}^{\infty}\mathbb{E}\big|\boldsymbol{g}_t^\top Q_{t,k}\,\boldsymbol{g}_{t-k}\big| \le M\sum_{k=0}^{\infty}\|Q_{t,k}\|_{\mathrm{op}} < \infty, \tag{86}$$

which establishes absolute convergence. The scalar-trace identity $\boldsymbol{a}^\top B\boldsymbol{c} = \mathrm{Tr}(B\boldsymbol{c}\boldsymbol{a}^\top)$ is used in the fourth line. This completes the derivation of equation 8. □

E.2 Proof of Theorem 2.2: Greedy optimal first-order optimizers

Proof of Theorem 2.2. We establish each claim in turn. Throughout, we assume $0 \in \mathcal{Q}$ as stated in the theorem.

(i) Existence: Since $\mathcal{Q}$ is compact and $Q \mapsto \langle Q, R\rangle_{\mathcal{H}}$ is continuous, the maximum is attained by the Weierstrass extreme value theorem. The optimal power $P^\star(R) = \sup_{Q\in\mathcal{Q}}\langle Q, R\rangle_{\mathcal{H}}$ is a supremum of linear functionals in R, hence sublinear (convex and positively homogeneous). Finiteness follows from compactness of $\mathcal{Q}$.

(ii) Construction: We establish the following conjugacy identities in the dynamic setting.

$$P^\star \;=\; \sigma_{\mathcal{Q}} \;=\; \delta_{\mathcal{Q}}^* \;=\; \gamma_{\mathcal{Q}^\circ}, \qquad (\gamma_{\mathcal{Q}})^* \;=\; \delta_{\mathcal{Q}^\circ}. \tag{87}$$

Here $\sigma_{\mathcal{Q}}(R) = \sup_{Q\in\mathcal{Q}}\langle Q, R\rangle_{\mathcal{H}}$ denotes the support function of $\mathcal{Q}$.

(1) Optimal power = support function = conjugate of indicator. By the definition of convex conjugate in $\mathcal{H}$,

$$(\delta_{\mathcal{Q}})^*(R) = \sup_{Q\in\mathcal{H}}\{\langle Q, R\rangle_{\mathcal{H}} - \delta_{\mathcal{Q}}(Q)\} = \sup_{Q\in\mathcal{Q}}\langle Q, R\rangle_{\mathcal{H}} = \sigma_{\mathcal{Q}}(R) = P^\star(R). \tag{88}$$

Thus $P^\star = \sigma_{\mathcal{Q}} = (\delta_{\mathcal{Q}})^*$.

(2) Optimal power = gauge of polar. By the definition of polar in $\mathcal{H}$, $R \in \mathcal{Q}^\circ$ if and only if $\sup_{Q\in\mathcal{Q}}\langle Q, R\rangle_{\mathcal{H}} \le 1$, i.e., $P^\star(R) \le 1$. Therefore

$$\gamma_{\mathcal{Q}^\circ}(R) = \inf\{\lambda > 0 : R \in \lambda\mathcal{Q}^\circ\} = \inf\{\lambda > 0 : P^\star(R) \le \lambda\} = P^\star(R). \tag{89}$$

Thus $P^\star = \gamma_{\mathcal{Q}^\circ}$.

(3) Conjugate of gauge = indicator of polar. We establish $(\gamma_{\mathcal{Q}})^* = \delta_{\mathcal{Q}^\circ}$. Consider two cases:

- If $R \in \mathcal{Q}^\circ$, then for all Q,

$$\langle Q, R\rangle_{\mathcal{H}} \le \gamma_{\mathcal{Q}}(Q)\cdot\sup_{Q'\in\mathcal{Q}}\langle Q', R\rangle_{\mathcal{H}} \le \gamma_{\mathcal{Q}}(Q), \tag{90}$$

  since $\sup_{Q'\in\mathcal{Q}}\langle Q', R\rangle_{\mathcal{H}} \le 1$ by definition of polar. Hence $\langle Q, R\rangle_{\mathcal{H}} - \gamma_{\mathcal{Q}}(Q) \le 0$ for all Q, with equality at $Q = 0$ (using $0 \in \mathcal{Q}$). Taking the supremum gives $(\gamma_{\mathcal{Q}})^*(R) = 0 = \delta_{\mathcal{Q}^\circ}(R)$.

- If $R \notin \mathcal{Q}^\circ$, there exists $Q_0 \in \mathcal{Q}$ with $\langle Q_0, R\rangle_{\mathcal{H}} > 1$. Since $Q_0 \in \mathcal{Q}$, we have $\gamma_{\mathcal{Q}}(Q_0) \le 1$. For any $\alpha > 0$, we have $\gamma_{\mathcal{Q}}(\alpha Q_0) = \alpha\gamma_{\mathcal{Q}}(Q_0)$, and thus

$$\begin{aligned}\langle \alpha Q_0, R\rangle_{\mathcal{H}} - \gamma_{\mathcal{Q}}(\alpha Q_0) &= \alpha\langle Q_0, R\rangle_{\mathcal{H}} - \alpha\gamma_{\mathcal{Q}}(Q_0)\\ &= \alpha(\langle Q_0, R\rangle_{\mathcal{H}} - \gamma_{\mathcal{Q}}(Q_0))\\ &\ge \alpha(\langle Q_0, R\rangle_{\mathcal{H}} - 1) \to +\infty\end{aligned}$$

  as $\alpha \to \infty$, since $\langle Q_0, R\rangle_{\mathcal{H}} > 1$. Hence $(\gamma_{\mathcal{Q}})^*(R) = +\infty = \delta_{\mathcal{Q}^\circ}(R)$.

Thus $(\gamma_{\mathcal{Q}})^* = \delta_{\mathcal{Q}^\circ}$.

This completes the proof of the conjugacy identities. We now prove the construction of the optimal optimizer using these identities.

Let $Q^\star \in \arg\max_{Q\in\mathcal{Q}}\langle Q, R\rangle_{\mathcal{H}}$. For any $M \in \mathcal{H}$,

$$P^\star(M) = \max_{Q\in\mathcal{Q}}\langle Q, M\rangle_{\mathcal{H}} \geq \langle Q^\star, M\rangle_{\mathcal{H}} = \langle Q^\star, R\rangle_{\mathcal{H}} + \langle Q^\star, M-R\rangle_{\mathcal{H}} = P^\star(R) + \langle Q^\star, M-R\rangle_{\mathcal{H}}, \tag{91}$$

which is the defining inequality for $Q^\star \in \partial_R P^\star(R)$. If the maximizer is unique, then $\partial_R P^\star(R) = \{Q^\star\}$. In finite-dimensional settings, or under standard support-function regularity conditions, this implies Gâteaux differentiability with $Q^\star = \nabla_R P^\star(R) = \nabla_R \gamma_{\mathcal{Q}^\circ}(R)$.

(iii) Lipschitz continuity in symmetrized polar gauge: We establish the one-sided bounds first. Since $P^\star = \gamma_{\mathcal{Q}^\circ}$ by the conjugacy identities, we have:

$$P^\star(R) - P^\star(\hat{R}) = \max_{Q\in\mathcal{Q}}\langle Q, R\rangle_{\mathcal{H}} - \max_{Q\in\mathcal{Q}}\langle Q, \hat{R}\rangle_{\mathcal{H}} \tag{92}$$

$$\leq \max_{Q\in\mathcal{Q}}\langle Q, R-\hat{R}\rangle_{\mathcal{H}} \tag{93}$$

$$= \gamma_{\mathcal{Q}^\circ}(R-\hat{R}). \tag{94}$$

Similarly, $P^\star(\hat{R}) - P^\star(R) \leq \gamma_{\mathcal{Q}^\circ}(\hat{R}-R)$. Therefore,

$$|P^\star(R) - P^\star(\hat{R})| \leq \max\{\gamma_{\mathcal{Q}^\circ}(R-\hat{R}), \gamma_{\mathcal{Q}^\circ}(\hat{R}-R)\} = \|R-\hat{R}\|_{\mathcal{Q}^\circ}^{\mathrm{sym}}. \tag{95}$$

Therefore, by using an estimated moment $\hat{R}$ instead of the true moment R, the error in optimal power can be bounded by $|P^\star(R) - P^\star(\hat{R})| \leq \|R-\hat{R}\|_{\mathcal{Q}^\circ}^{\mathrm{sym}}$. □

E.3 Proof of Corollary 2.3: Family-hull reduction

Proof of Corollary 2.3. Write any $Q \in \mathrm{co}(\mathcal{F}\cup\{0\})$ as $Q = \pi_0\cdot 0 + \sum_{j=1}^m \pi_j Q_{\phi_j}$ with $\pi_0, \pi_j \geq 0$ and $\pi_0 + \sum_j \pi_j = 1$. Let $M := \sup_\phi \langle Q_\phi, R\rangle_{\mathcal{H}}$. By linearity, $\langle Q, R\rangle_{\mathcal{H}} = \sum_j \pi_j \langle Q_{\phi_j}, R\rangle_{\mathcal{H}}$.

Case $M \geq 0$: Each $\langle Q_{\phi_j}, R\rangle_{\mathcal{H}} \leq M$ and $\sum_j \pi_j \leq 1$, so $\langle Q, R\rangle_{\mathcal{H}} \leq M$. If the supremum is attained by some $Q_{\phi^\star}$ with $\langle Q_{\phi^\star}, R\rangle_{\mathcal{H}} = M$, then this element attains the bound; otherwise the equality $\sup = M$ is understood at the level of suprema.

Case $M < 0$: Every $\langle Q_{\phi_j}, R\rangle_{\mathcal{H}} < 0$, so $\langle Q, R\rangle_{\mathcal{H}} \leq 0$ with equality at $Q = 0$.

In both cases, $\sup_{Q\in\mathrm{co}(\mathcal{F}\cup\{0\})}\langle Q, R\rangle_{\mathcal{H}} = M_+$. Passing to the closure preserves the supremum by continuity of $Q \mapsto \langle Q, R\rangle_{\mathcal{H}}$. Proposition 2.1 gives $\langle Q_\phi, R\rangle_{\mathcal{H}} = \mathbb{E}[\boldsymbol{g}_t^\top (Q_\phi \boldsymbol{g})_t]$. □

E.4 Proof of Theorem 3.1: Greedy optimal SGD+Momentum

Proof of Theorem 3.1. We work in the impulse-space as defined in Section 3, i.e., a Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ of causal LTI filters with matrix impulse response $\{q_k\}_{k\geq 0}$ with a Hilbert norm $\|Q\|_{\mathcal{H}}^2 = \sum_{k=0}^\infty \mathrm{Tr}(q_k^\top q_k)$, where k is the lag index.

By the main-text definitions, the SGD+Momentum family is $\mathcal{Q}_{\eta,\beta} = \mathcal{Q}_{1\mathrm{p}} \cap \mathcal{Q}_{\mathrm{H}}(B)$, where $\mathcal{Q}_{1\mathrm{p}} = \{Q_{\beta,k} = \eta\beta^k I : \eta \geq 0,\ \beta \in \mathcal{B} \subset [0,1)\}$ is the 1-pole family and $\mathcal{Q}_{\mathrm{H}}(B) = \{Q : \|Q\|_{\mathcal{H}} \leq \sqrt{B}\}$ is the Hilbert trust region. We solve $\max_{Q\in\mathcal{Q}_{\eta,\beta}}\langle Q, R\rangle_{\mathcal{H}}$ by parameterizing $Q = Q_{\text{SGD+M}}$ via the 1-pole impulse response and saturating the trust region.

Step 1 — Norm of the 1-pole optimizer. The impulse response is $Q_{\beta,k} = \eta\beta^k I$ in the family $\mathcal{Q}_{1\mathrm{p}}$. By definition,

$$\|Q_{\text{SGD+M}}\|_{\mathcal{H}}^2 = \sum_{k\geq 0} \mathrm{Tr}(Q_{\beta,k}^\top Q_{\beta,k}) = \eta^2 \sum_{k\geq 0} \beta^{2k}\,\mathrm{Tr}(I) = \frac{\eta^2 d}{1-\beta^2}, \tag{96}$$

where d is the parameter dimension. The trust region constraint $\|Q_{\text{SGD+M}}\|_{\mathcal{H}} \leq \sqrt{B}$ imposes

$$\eta \leq \sqrt{B(1-\beta^2)/d}. \tag{97}$$

Step 2 — Alignment with the moment operator. The inner product with R is

$$\langle Q_{\text{SGD+M}}, R\rangle_{\mathcal{H}} = \sum_{k\geq 0} \mathrm{Tr}(Q_{\beta,k}^\top R_{t,k}) = \eta \sum_{k\geq 0} \beta^k\ \mathrm{Tr}(R_{t,k}) = \eta \sum_{k\geq 0} \beta^k T_{t,k}, \tag{98}$$

where $T_{t,k} \coloneqq \mathrm{Tr}(R_{t,k})$.

Step 3 — Reduce to 1-D search; saturate trust region under positive score. Define the score function $S_t(\beta) \coloneqq \sum_{k\geq 0} \beta^k T_{t,k}$. For fixed $\beta \in \mathcal{B}$, if $S_t(\beta) > 0$, the objective is linear increasing in η and the maximizer saturates the trust region; if $S_t(\beta) \leq 0$, the best gain is $\eta = 0$. Since $\beta = 0$ gives $S_t(0) = T_{t,0} = \mathbb{E}\|\boldsymbol{g}_t\|^2 > 0$ whenever $0 \in \mathcal{B}$ and the gradient is nonzero, the global nontrivial maximizer has positive score.

The trust region-normalized gain is

$$A(\beta; t) \coloneqq \frac{\langle Q_{\text{SGD+M}}, R\rangle_{\mathcal{H}}}{\|Q_{\text{SGD+M}}\|_{\mathcal{H}}} = \frac{\sqrt{1-\beta^2}}{\sqrt{d}} \sum_{k\geq 0} \beta^k T_{t,k} = \frac{\sqrt{1-\beta^2}}{\sqrt{d}} S_t(\beta). \tag{99}$$

Hence the optimal momentum and learning rate are

$$\beta_t^\star = \arg\max_{\beta\in\mathcal{B}} \sqrt{1-\beta^2}\, S_t(\beta), \quad \eta_t^\star = \sqrt{B(1-(\beta_t^\star)^2)/d}. \tag{100}$$

Step 4 — Solving A for streaming gradients. Let $\boldsymbol{m}_{\beta,t} = \boldsymbol{g}_t + \beta \boldsymbol{g}_{t-1} + \beta^2 \boldsymbol{g}_{t-2} + \cdots$ be the unnormalized momentum at time t. This can be obtained from a sequential filtering process:

$$\boldsymbol{m}_{\beta,t} = \boldsymbol{g}_t + \beta\, \boldsymbol{m}_{\beta,t-1}, \quad \boldsymbol{m}_{\beta,0} = \boldsymbol{0}, \tag{101}$$

which is exactly the same as how typical autograd frameworks implement momentum. From $R_{t,k} = \mathbb{E}[\boldsymbol{g}_t\, \boldsymbol{g}_{t-k}^\top]$ and $T_{t,k} = \mathrm{Tr}(R_{t,k})$, we have

$$S_t(\beta) = \sum_{k=0}^{\infty} T_{t,k}\, \beta^k = \sum_{k=0}^{\infty} \mathrm{Tr}(\mathbb{E}[\boldsymbol{g}_t\, \boldsymbol{g}_{t-k}^\top])\, \beta^k = \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{g}_t \sum_{k=0}^{\infty} \beta^k \boldsymbol{g}_{t-k}^\top\right]\right) = \mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{m}_{\beta,t}], \tag{102}$$

where the last equality uses $\mathrm{Tr}(\boldsymbol{g}_t\, \boldsymbol{m}_{\beta,t}^\top) = \boldsymbol{g}_t^\top\, \boldsymbol{m}_{\beta,t}$. Substituting into $A(\beta; t) = \frac{\sqrt{1-\beta^2}}{\sqrt{d}} S_t(\beta)$, we can rewrite the optimal momentum as

$$\beta_t^\star = \arg\max_{\beta\in\mathcal{B}} A(\beta; t) = \arg\max_{\beta\in\mathcal{B}} \sqrt{1-\beta^2}\, \mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{m}_{\beta,t}], \tag{103}$$

where $\boldsymbol{m}_{\beta,t} = \sum_{k=0}^{\infty} \beta^k \boldsymbol{g}_{t-k}$ is the unnormalized momentum at time t with momentum parameter β. This completes the proof. Note that, in theory, the expectation is taken over the entire possible gradient sequence $\boldsymbol{g}_t$, which should be approximated in the real-world application. □

E.5 Proof of Corollary 3.2: One-pole gradients recover one-pole momentum

Proof of Corollary 3.2. By definition of $\boldsymbol{m}_{\beta,t}$,

$$A = \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{m}_{\beta,t}] = \sum_{k\geq 0} \beta^k \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{g}_{t-k}] = r_0 \sum_{k\geq 0} (\beta\rho)^k = \frac{r_0}{1-\beta\rho}. \tag{104}$$

The series converges since $|\beta\rho| < 1$. Hence

$$\sqrt{1-\beta^2}\, \mathbb{E}[\boldsymbol{g}_t^\top \boldsymbol{m}_{\beta,t}] = r_0 \frac{\sqrt{1-\beta^2}}{1-\beta\rho}. \tag{105}$$

Since $r_0 > 0$, it suffices to maximize $\log A$:

$$\frac{d}{d\beta} \log A(\beta) = -\frac{\beta}{1-\beta^2} + \frac{\rho}{1-\beta\rho} = \frac{\rho-\beta}{(1-\beta^2)(1-\beta\rho)}. \tag{106}$$

The denominator is positive for $\beta, \rho \in (-1, 1)$, so A increases for $\beta < \rho$ and decreases for $\beta > \rho$. Therefore the unique maximizer over $(-1, 1)$ is $\beta^\star = \rho$. Restricting to $\beta \in [0, \bar{\beta}]$ with $0 < \bar{\beta} < 1$ gives the projection $\max\{0, \min\{\bar{\beta}, \rho\}\}$. □

E.6 Proof of Corollary 3.3: Greedy optimal Adam/AdamW

Proof of Corollary 3.3. We work in the impulse-space as defined in Section 3. Throughout this proof, we condition on the realized second-moment state $\boldsymbol{c}_{\beta_2,t}$ for each candidate β_2 and treat it as a fixed time-varying diagonal preconditioner. All expectations below may therefore be read as conditional expectations given this state.

For the Adam family we use the weighted diagonal trust-region norm

$$\|Q\|_{\mathcal{D},\boldsymbol{c}}^2 = \sum_{j=1}^{d} c_j \sum_{k\geq 0} |q_{j,k}|^2, \tag{107}$$

where k is the lag index.

By the main-text definitions, the Adam family is $\mathcal{Q}_{\eta,\beta_1,\beta_2} = \mathcal{Q}_{\mathrm{D}}(B,\boldsymbol{c}) \cap \mathcal{Q}_{\mathrm{1p}}(\boldsymbol{c})$, where $\mathcal{Q}_{\mathrm{D}}(B,\boldsymbol{c}) = \{\mathrm{diag}(q_j) : \sum_j c_j \sum_{k\geq 0} |q_{j,k}|^2 \leq B\}$ is the elementwise weighted norm-bounded family and $\mathcal{Q}_{\mathrm{1p}}(\boldsymbol{c}) = \{q_{j,k} = \eta(1-\beta_1)\beta_1^k c_j^{-1} : \eta \geq 0,\ 0 < \beta_1 < 1\}$ is the Adam 1-pole family. We solve $\max_{Q\in\mathcal{Q}_{\eta,\beta_1,\beta_2}} \langle Q, R\rangle$ by parameterizing $Q = Q_{\mathrm{Adam}}$ via the diagonal 1-pole impulse response and saturating the trust region.

Step 1 — Norm of the diagonal optimizer. For each parameter coordinate j, we have the running second moment and the coordinate-wise cost:

$$v_{\beta_2,t,j} = (1-\beta_2)\sum_{k=0}^{\infty} \beta_2^k g_{t-k,j}^2, \quad c_{\beta_2,t,j} = (v_{\beta_2,t,j} + \epsilon)^{1/2}. \tag{108}$$

From the definition of $\mathcal{Q}_{\mathrm{1p}}(\boldsymbol{c})$, the per-coordinate impulse response at time t is:

$$q_{t,j,k} = \eta\,(1-\beta_1)\,\beta_1^k / c_{\beta_2,t,j}. \tag{109}$$

Therefore, the weighted norm of $\mathcal{Q}_{\mathrm{D}}(B,\boldsymbol{c})$ evaluates to:

$$\|Q_{\mathrm{Adam}}\|_{\mathcal{D},\boldsymbol{c}}^2 = \sum_{j=1}^{d} c_{\beta_2,t,j} \sum_{k\geq 0} |q_{t,j,k}|^2 = \eta^2\,\frac{(1-\beta_1)^2}{1-\beta_1^2} \sum_{j=1}^{d} \frac{1}{c_{\beta_2,t,j}} = \eta^2\,\frac{1-\beta_1}{1+\beta_1}\,W_{\beta_2,t}, \tag{110}$$

where $W_{\beta_2,t} \coloneqq \sum_j 1/c_{\beta_2,t,j}$ is the normalization factor.

Step 2 — Alignment with the moment operator. Since Q_{Adam} is diagonal, only the diagonal entries of $R_{t,k}$ enter the inner product; let $r_{t,k,j} \coloneqq R_{t,k,jj} = \mathbb{E}[g_{t,j}\, g_{t-k,j} \mid \boldsymbol{c}_{\beta_2,t}]$. Then

$$\langle Q_{\mathrm{Adam}}, R\rangle = \sum_{k\geq 0} \mathrm{Tr}(Q_k^\top R_{t,k}) = \sum_{k\geq 0}\sum_{j=1}^{d} q_{t,j,k}\, r_{t,k,j} \tag{111}$$

$$= \eta(1-\beta_1) \sum_{k\geq 0} \beta_1^k \sum_{j=1}^{d} \frac{r_{t,k,j}}{c_{\beta_2,t,j}} \tag{112}$$

$$= \eta(1-\beta_1) \sum_{k\geq 0} \beta_1^k\, T_{t,k}(\beta_2), \tag{113}$$

where $T_{t,k}(\beta_2) \coloneqq \sum_j r_{t,k,j}/c_{\beta_2,t,j}$ is the weighted trace of the moment operator.

Step 3 — Reduce to 1-D search; saturate trust region. For fixed (β_1,β_2), define the score

$$S_t(\beta_1,\beta_2) \coloneqq \sum_{k\geq 0} \beta_1^k\, T_{t,k}(\beta_2). \tag{114}$$

The inner product is linear in η while the constraint is quadratic. If $S_t(\beta_1,\beta_2) > 0$, the maximizer saturates the trust region; if $S_t(\beta_1,\beta_2) \leq 0$, the best gain is $\eta = 0$. In the closure of the admissible range, $\beta_1 = 0$ gives $S_t(0,\beta_2) = T_{t,0}(\beta_2) = \sum_j \mathbb{E}[g_{t,j}^2 \mid \boldsymbol{c}_{\beta_2,t}]/c_{\beta_2,t,j} > 0$ whenever the gradient is nonzero, so the global nontrivial maximizer has positive score.

The trust-region-normalized gain is

$$A(\beta_1, \beta_2; t) \coloneqq \frac{\langle Q_{\text{Adam}}, R \rangle}{\|Q_{\text{Adam}}\|_{\mathcal{D},\boldsymbol{c}}} = \frac{\sqrt{1-\beta_1^2}}{\sqrt{W_{\beta_2,t}}}\, S_t(\beta_1, \beta_2). \tag{115}$$

Therefore, we have the optimal Adam hyperparameters as

$$(\beta_1^\star, \beta_2^\star)_t = \arg \max_{0<(\beta_1,\beta_2)<1} A(\beta_1, \beta_2; t), \tag{116}$$

and the corresponding trust-region-saturating learning rate is

$$\eta_t^\star = \sqrt{B} \,/\, \|Q_{\text{Adam}}\|_{\mathcal{D},\boldsymbol{c}} = \sqrt{B}\, \sqrt{\frac{1+\beta_1^\star}{(1-\beta_1^\star)\, W_{\beta_2^\star,t}}}. \tag{117}$$

This follows from equation 110 above and the diagonal trust region definition:

$$\mathcal{Q}_{\text{D}}(B, \boldsymbol{c}) \coloneqq \{\text{diag}(q_j) : \textstyle\sum_j c_j \sum_{k\geq 0} |q_{j,k}|^2 \leq B\}. \tag{118}$$

Step 4 — Solving A for streaming gradients. Recall $r_{t,k,j} = R_{t,k,jj} = \mathbb{E}[g_{t,j}\, g_{t-k,j} \mid \boldsymbol{c}_{\beta_2,t}]$ for the diagonal entries at time t and lag k. Then

$$T_{t,k}(\beta_2) = \sum_{j=1}^{d} \frac{r_{t,k,j}}{c_{\beta_2,t,j}} = \sum_{j=1}^{d} \mathbb{E}\left[\frac{g_{t,j}\, g_{t-k,j}}{c_{\beta_2,t,j}} \,\middle|\, \boldsymbol{c}_{\beta_2,t}\right]. \tag{119}$$

$$S_t(\beta_1, \beta_2) = \sum_{k=0}^{\infty} \beta_1^k\, T_{t,k}(\beta_2) \tag{120}$$

$$= \sum_{k=0}^{\infty} \beta_1^k \sum_{j=1}^{d} \mathbb{E}\left[\frac{g_{t,j}\, g_{t-k,j}}{c_{\beta_2,t,j}} \,\middle|\, \boldsymbol{c}_{\beta_2,t}\right] \tag{121}$$

$$= \sum_{j=1}^{d} \mathbb{E}\left[\frac{g_{t,j}}{c_{\beta_2,t,j}} \sum_{k=0}^{\infty} \beta_1^k g_{t-k,j} \,\middle|\, \boldsymbol{c}_{\beta_2,t}\right] \tag{122}$$

$$= \frac{1}{1-\beta_1} \sum_{j=1}^{d} \mathbb{E}\left[g_{t,j}\, \frac{\mu_{\beta_1,t,j}}{c_{\beta_2,t,j}} \,\middle|\, \boldsymbol{c}_{\beta_2,t}\right] \tag{123}$$

$$= \frac{1}{1-\beta_1} \mathbb{E}\left[\sum_{j=1}^{d} g_{t,j}\, \frac{\mu_{\beta_1,t,j}}{(v_{\beta_2,t,j}+\epsilon)^{1/2}} \,\middle|\, \boldsymbol{c}_{\beta_2,t}\right] \tag{124}$$

$$= \frac{1}{1-\beta_1} \mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{u}_{\beta_1,\beta_2,t} \mid \boldsymbol{c}_{\beta_2,t}], \tag{125}$$

where

$$\mu_{\beta_1,t,j} = (1-\beta_1) \sum_{k=0}^{\infty} \beta_1^k g_{t-k,j} = \beta_1\, \mu_{\beta_1,t-1,j} + (1-\beta_1)\, g_{t,j}, \tag{126}$$

$$v_{\beta_2,t,j} = (1-\beta_2) \sum_{k=0}^{\infty} \beta_2^k g_{t-k,j}^2 = \beta_2\, v_{\beta_2,t-1,j} + (1-\beta_2)\, g_{t,j}^2, \tag{127}$$

are the normalized first and second moments for coordinate j at time t, and

$$\boldsymbol{u}_{\beta_1,\beta_2,t} = \boldsymbol{\mu}_{\beta_1,t} \odot \boldsymbol{c}_{\beta_2,t}^{-1}, \qquad u_{\beta_1,\beta_2,t,j} = \frac{\mu_{\beta_1,t,j}}{c_{\beta_2,t,j}} = \frac{\mu_{\beta_1,t,j}}{(v_{\beta_2,t,j}+\epsilon)^{1/2}}, \tag{128}$$

is the Adam update direction at time t, matching the main text. Substituting into the expression for A:

$$A(\beta_1, \beta_2; t) = \frac{\sqrt{1-\beta_1^2}}{\sqrt{W_{\beta_2,t}}}\, S_t(\beta_1, \beta_2) \tag{129}$$

$$= \frac{\sqrt{1-\beta_1^2}}{(1-\beta_1)\sqrt{W_{\beta_2,t}}}\, \mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{u}_{\beta_1,\beta_2,t} \mid \boldsymbol{c}_{\beta_2,t}] \tag{130}$$

$$= \sqrt{\frac{1+\beta_1}{1-\beta_1}}\, \frac{\mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{u}_{\beta_1,\beta_2,t} \mid \boldsymbol{c}_{\beta_2,t}]}{\sqrt{W_{\beta_2,t}}}. \tag{131}$$

Therefore, we can rewrite the optimal Adam hyperparameters as

$$(\beta_1^\star, \beta_2^\star)_t = \arg \max_{0<\beta_1<1,\, 0<\beta_2<1} \sqrt{\frac{1+\beta_1}{1-\beta_1}}\, \frac{\mathbb{E}[\boldsymbol{g}_t^\top\, \boldsymbol{u}_{\beta_1,\beta_2,t} \mid \boldsymbol{c}_{\beta_2,t}]}{\sqrt{W_{\beta_2,t}}}. \tag{132}$$

This completes the proof. Note that, in theory, the expectation is taken over the entire possible gradient sequence $\boldsymbol{g}_t$ conditioned on the second-moment state, which should be approximated in the real-world application. □

E.7 Proof of Theorem 3.4: Finite-step descent

Proof of Theorem 3.4. Since $\mathcal{L}$ has L_s-Lipschitz gradient, for any $\boldsymbol{x}, \boldsymbol{h} \in \mathbb{R}^d$,

$$\mathcal{L}(\boldsymbol{x}+\boldsymbol{h}) = \mathcal{L}(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})^\top \boldsymbol{h} + \int_0^1 \big(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}+s\boldsymbol{h}) - \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})\big)^\top \boldsymbol{h}\, ds. \tag{133}$$

Apply this with $\boldsymbol{x} = \boldsymbol{\theta}$ and $\boldsymbol{h} = -\eta\boldsymbol{u}$:

$$\mathcal{L}(\boldsymbol{\theta}-\eta\boldsymbol{u}) = \mathcal{L}(\boldsymbol{\theta}) - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})^\top \boldsymbol{u} - \int_0^1 \big(\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}-s\eta\boldsymbol{u}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\big)^\top \eta\boldsymbol{u}\, ds. \tag{134}$$

Since $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$,

$$D_\eta(\boldsymbol{u}) - \eta A(\boldsymbol{u}) = \int_0^1 \big(\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}-s\eta\boldsymbol{u}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\big)^\top \eta\boldsymbol{u}\, ds. \tag{135}$$

Taking absolute values and using Lipschitz continuity of $\nabla_{\boldsymbol{\theta}}\mathcal{L}$:

$$\begin{aligned}\big|D_\eta(\boldsymbol{u}) - \eta A(\boldsymbol{u})\big| &\le \int_0^1 \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}-s\eta\boldsymbol{u}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\| \cdot \eta\|\boldsymbol{u}\|\, ds \\ &\le \int_0^1 L_s s\eta\|\boldsymbol{u}\| \cdot \eta\|\boldsymbol{u}\|\, ds = L_s\eta^2\|\boldsymbol{u}\|^2 \int_0^1 s\, ds = \frac{L_s\eta^2}{2}\|\boldsymbol{u}\|^2. \end{aligned}$$ □

E.8 Proof of Corollary 3.5: Near-optimality of greedy alignment

Proof of Corollary 3.5. For any $\boldsymbol{v} \in \mathcal{U}$, by Theorem 3.4, $D_\eta(\boldsymbol{u}^\star) \ge \eta A(\boldsymbol{u}^\star) - \frac{L_s\eta^2}{2}\rho^2$. Since $\boldsymbol{u}^\star$ maximizes A, we have $A(\boldsymbol{u}^\star) \ge A(\boldsymbol{v})$, so $D_\eta(\boldsymbol{u}^\star) \ge \eta A(\boldsymbol{v}) - \frac{L_s\eta^2}{2}\rho^2$. Again by Theorem 3.4, $D_\eta(\boldsymbol{v}) \le \eta A(\boldsymbol{v}) + \frac{L_s\eta^2}{2}\rho^2$. Combining yields $D_\eta(\boldsymbol{u}^\star) \ge D_\eta(\boldsymbol{v}) - L_s\eta^2\rho^2$. Since $\boldsymbol{v}$ was arbitrary, the result follows. □

E.9 Proof of Corollary 4.1: Exact finite-candidate K-Switch

Proof of Corollary 4.1. The normalized maps $\tilde{Q}_{k,t}$ are causal because each $Q_{k,t}$ is causal and $a_{k,t}$ is a causal scalar normalization. Consider any element $Q \in \mathrm{co}(\tilde{\mathcal{F}}_t)$. By definition of the finite convex hull, there exist coefficients $\pi_k \ge 0$, $\sum_{k=1}^K \pi_k = 1$, such that

$$Q \;=\; \sum_{k=1}^K \pi_k \tilde{Q}_{k,t}. \tag{136}$$

The action of this convex combination on the gradient stream is the pointwise convex combination of the corresponding update streams:

$$(Q\boldsymbol{g})_t \;=\; \sum_{k=1}^K \pi_k (\tilde{Q}_{k,t}\boldsymbol{g})_t. \tag{137}$$

Therefore, by linearity of the inner product in the update and linearity of expectation,

$$\begin{aligned} P_t(Q) &= \mathbb{E}\big[\boldsymbol{g}_t^\top (Q\boldsymbol{g})_t\big] \\ &= \mathbb{E}\left[\boldsymbol{g}_t^\top \sum_{k=1}^{K} \pi_k (\tilde{Q}_{k,t}\boldsymbol{g})_t\right] \\ &= \sum_{k=1}^{K} \pi_k \mathbb{E}\Big[\boldsymbol{g}_t^\top (\tilde{Q}_{k,t}\boldsymbol{g})_t\Big] \\ &= \sum_{k=1}^{K} \pi_k P_t(\tilde{Q}_{k,t}) \\ &\le \max_{1\le k\le K} P_t(\tilde{Q}_{k,t}). \end{aligned} \tag{138}$$

Taking the supremum over all $Q \in \mathrm{co}(\tilde{\mathcal{F}}_t)$ gives

$$\sup_{Q\in\mathrm{co}(\tilde{\mathcal{F}}_t)} P_t(Q) \;\le\; \max_{1\le k\le K} P_t(\tilde{Q}_{k,t}). \tag{139}$$

The reverse inequality holds because every $\tilde{Q}_{k,t}$ belongs to $\mathrm{co}(\tilde{\mathcal{F}}_t)$. Hence

$$\sup_{Q\in\mathrm{co}(\tilde{\mathcal{F}}_t)} P_t(Q) \;=\; \max_{1\le k\le K} P_t(\tilde{Q}_{k,t}). \tag{140}$$

Finally, by definition of $\tilde{Q}_{k,t}$,

$$P_t(\tilde{Q}_{k,t}) \;=\; \mathbb{E}\Big[\boldsymbol{g}_t^\top (\tilde{Q}_{k,t}\boldsymbol{g})_t\Big] \;=\; \mathbb{E}\big[a_{k,t}\boldsymbol{g}_t^\top \boldsymbol{u}_{k,t}\big]\,. \tag{141}$$

This proves the displayed equality.

It remains to prove the active-support claim. Let

$$P^\star \;\coloneqq\; \max_{1\le k\le K} P_t(\tilde{Q}_{k,t}). \tag{142}$$

Suppose

$$Q^\star \;=\; \sum_{k=1}^{K} \pi_k \tilde{Q}_{k,t}, \qquad \pi_k \ge 0, \quad \sum_{k=1}^{K} \pi_k = 1, \tag{143}$$

attains the supremum over the convex hull. Then

$$P^\star \;=\; P_t(Q^\star) \;=\; \sum_{k=1}^{K} \pi_k P_t(\tilde{Q}_{k,t}) \;\le\; \sum_{k=1}^{K} \pi_k P^\star \;=\; P^\star. \tag{144}$$

Thus the weighted average equals its upper bound. If there existed an active index i with $\pi_i > 0$ and $P_t(\tilde{Q}_{i,t}) < P^\star$, then the weighted average would be strictly smaller than $P^\star$, a contradiction. Therefore every active component satisfies $P_t(\tilde{Q}_{k,t}) = P^\star$. The stated K-Switch selector is exactly the argmax over these finite candidate scores. This completes the proof. □

E.10 Proof of Proposition 4.2: Online stability

Proof. By the uniform error bound and optimality of $\hat{k}$ for the estimated scores,

$$A_{k^\star} \;\le\; \hat{A}_{k^\star} + \varepsilon \;\le\; \hat{A}_{\hat{k}} + \varepsilon \;\le\; A_{\hat{k}} + 2\varepsilon. \tag{145}$$

Therefore, $A_{k^\star} - A_{\hat{k}} \le 2\varepsilon$. If the maximizer is unique with a gap $\Delta \coloneqq A_{k^\star} - \max_{j\neq k^\star} A_j > 2\varepsilon$ and $\hat{k} \neq k^\star$, then $A_{k^\star} - A_{\hat{k}} \ge \Delta > 2\varepsilon$, contradicting the first claim. Hence $\hat{k} = k^\star$. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) state four contributions—(i) lifting greedy alignment to an operator-level principle, (ii) well-posedness and stability for compact families, (iii) reduction to hyperparameter scoring, and (iv) dynamic momentum rules—and each is supported by Sections 2, 3, 4, and 5, respectively, with proofs in Appendix E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 6 ("Limitations and scope").

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Assumptions (compactness of $\mathcal{Q}$, stability/causality of linear filters, ℓ^1-summability of R, etc.) are stated alongside each result in Sections 2–3. Complete proofs are provided in Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Algorithms are in Section 4, implementation details are in Section 5 and in Appendix D.1 All datasets used (CIFAR-100, MetaMathQA-395K, GSM8K, Commonsense-170K, Cars, CUB-200, DTD, Food-101, RESISC45, SUN397) are publicly available with citations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets we used are public and cited in Section 5 and Appendix D, and the algorithm and all hyperparameters are described in sufficient detail to reproduce results from scratch (Section 4 and Appendix D.1). The code is in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5 and Appendix D.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All result tables in Section 5 (Tables 3, 4, 5, 6) report mean $\pm$ standard deviation over multiple independent runs with different random seeds. Figures 2 and 3 show error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Per-experiment GPU type and count are listed in the hyperparameter tables of Appendix D.1, and compute time is reported in Section 5 and Appendix D.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have a "Broader impact" section in the main paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no new pre-trained models, image generators, or scraped datasets. Our contribution is an optimizer-selection algorithm.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Every dataset and every pre-trained model is cited in Section 5 and Appendix D, and all are used under their respective public licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets, model checkpoints, or other assets. This is a theoretical contribution.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or crowdsourcing, so IRB (or equivalent) approval is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.