ON FAST HYPERPARAMETER TRANSFER ACROSS WIDTH AND TRAINING HORIZON

Anonymous authors
Paper under double-blind review

ABSTRACT

Hyperparameter (HP) transfer, which avoids expensive tuning on large neural networks by reusing values tuned on small models, can substantially reduce large-scale pretraining costs. Yet existing evidence is largely qualitative and does not precisely characterize the computational gain. We study HP transfer more rigorously through the framework of *fast transfer* [GWB25], which relates computational benefits to the convergence rates of the optimal loss and HPs with model scale n. We first prove a second-order Tensor Program master theorem for general neural computations at a fixed training horizon T. For fully connected polynomial networks trained by gradient descent under the Maximal Update Parameterization (μP), this yields a sharp $n^{-1/2}$ fluctuation of the loss and optimal HPs. While this result quantifies loss and HP convergence, we argue that fixed-T asymptotics do not explain the computational benefit of μ-transfer, as the leading centered loss fluctuations conflict with empirically observed scaling laws and loss monotonicity. This motivates us to investigate jointly diverging model width n and training horizon T in a minimal tractable setting: tuning the gradient descent learning rate η in a power-law random features model. Using a resolvent expansion, we derive sharp loss asymptotics and HP convergence, and establish necessary and sufficient spectral conditions for fast transfer at width-dependent training horizons. We further discuss the scaling behaviors at the compute-optimal frontier and the effect of stochastic gradients.

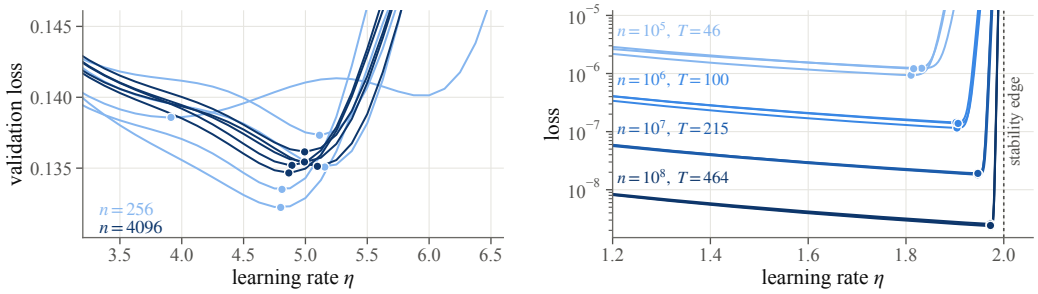
1 INTRODUCTION

Neural scaling laws suggest that the performance of large neural networks improves predictably as model size, data, and compute increase [HNA+17; KMH+20; HBM+22]. Due to the prohibitive computational cost of training models at increasing scales, small-scale experiments have become an essential tool for forecasting and designing much larger training runs. One prominent example of such small-proxy forecasting is the *Tensor Program* framework [Yan20; YH21; YL23], which argues that, under an appropriate width-dependent parameterization, neural-network training admits a nontrivial feature-learning limit, while hyperparameters (HPs) – most notably the learning rate – remain of constant order as width grows. This motivates the μ-transfer strategy [YHB+22]: tune HPs on a small proxy model and reuse the selected values at a much larger scale. Empirically, this strategy can substantially reduce tuning costs while retaining near-optimal large-scale performance [YHB+22; Lin24; NMHO24; EXW+24].

Evidence for HP transfer is nevertheless largely qualitative. A typical analysis (e.g., [YHB+22; BNL+24; YYZH24; DZN+25; MAB+26]) overlays loss-versus-HP curves at several widths and visually inspects if the valleys align. Such alignment indicates that the optimal HP $\eta_n^\star$ stabilizes as the model scale (width) n grows, and in some toy models $\eta_n^\star$ provably converges [Hay26]. But convergence without a rate says neither how fast $\eta_n^\star$ stabilizes nor whether tuning a small-scale proxy and transfer is cheaper than tuning the target model directly. [GWB25] introduced a quantitative framework to understand the computational gain of μ-transfer. Let $\phi_n(\eta)$ be the (training or validation) loss at scale n, and consider the *loss gap* $a_n := |\phi_n(\eta_n^\star) - \phi_\infty(\eta_\infty^\star)|$, the *HP gap* $b_n := |\eta_n^\star - \eta_\infty^\star|$, and the local sensitivity μ_n defined by $\phi_n(\eta) - \phi_n(\eta_n^\star) \asymp \mu_n|\eta - \eta_n^\star|^2$. Their *fast* transfer criterion is when $\mu_n b_n^2 = o(a_n)$: the loss incurred by the proxy's HP error is negligible compared with the gain from scale — see also [KB26]. Under a grid-search cost model, this condition makes μ-transfer asymptotically more compute-efficient than direct tuning at the target scale[1]. Provided that larger models yield better performance (a premise we revisit), the benefit of transfer thus reduces to the

[1] In this work, we do not separate the definitions of fast vs. useful transfer in [GWB25], and use "fast transfer" to describe scenarios where μ-transfer offers computational gain over direct grid search (Definition 1).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Fixed horizon ($T = 4$): wider is not better. **(b)** Growing horizon ($T = n^{1/3}$): monotone gain.

Figure 1: Two regimes of learning-rate transfer. **(a)** μP MLPs of widths 256 and 4096 trained for $T = 4$ GD steps, five initializations each. Wider networks are less noisy but not better: their tuned losses fall on both sides of those of narrower networks (Section 3). **(b)** Sketched linear regression trained for $T = n^{1/3}$ steps ($\alpha = 2$, $\rho = 2.5$, three sketches per width). The tuned loss decreases monotonically as width and horizon grow together, and the optimal learning rate moves slowly toward the stability edge $\eta = 2$ (Section 4).

comparing convergence rates of the optimal loss and HP. [GWB25] gave the first provable example of fast transfer for ℓ_2 regularization in random-features regression. Two questions remain open.

1. Does fast transfer hold for the *learning rate*, the HP most commonly transferred in practice?
2. What is the relevant asymptotic regime for studying the benefit of μ-transfer?

We address both questions in two complementary scaling regimes (Figure 1).

Fixed horizon: second-order Tensor Programs (Section 3). Tensor Programs (TPs) [Yan20; YH21] provide a mathematical description of the infinite-width limit of bounded neural computations. A TP is a fixed, finite sequence of matrix multiplications and coordinatewise nonlinearities; it can express the forward and backward passes of standard architectures, and hence training by a fixed number T of gradient descent (GD) steps. The TP master theorem is a first-order law of large numbers: as the width $n \to \infty$, every scalar computed by the program, such as the training or validation loss, converges to a deterministic limit, which under μP describes a feature-learning infinite-width network.

- **A central limit theorem for TPs.** We prove the next-order correction to the master theorem: the scalars of any finite NETSOR$^\top$ program fluctuate jointly as $n^{-1/2}$ times a Gaussian vector, whose covariance is given by a recursion over the program (Theorem 1). GD training of μP MLPs with polynomial activations falls in this class. For such networks, the entire loss vs. learning rate curve fluctuates at scale $n^{-1/2}$; hence $a_n \asymp n^{-1/2}$ and $b_n = O(n^{-1/2})$ at a nondegenerate optimum, and the fast-transfer criterion in [GWB25] $\mu_n b_n^2 = o(a_n)$ holds (Theorem 5). This turns the qualitative stabilization of the optimal learning rate under μP into explicit rates, and the argument extends to other HPs that enter the program polynomially, such as momentum or weight decay.
- **Fixed horizons cannot explain the benefit of transfer.** It is tempting to conclude that learning-rate transfer is fast, but we argue that the fixed-T asymptotic is the wrong regime for this question. The $n^{-1/2}$ correction to the tuned loss is *centered*, so of two independently initialized networks, the wider one has lower tuned loss with probability tending to $1/2$ (Figure 1(a)). This contradicts the monotone improvement with scale that makes transfer worthwhile (assumed in [GWB25]) and exhibited by scaling laws. Averaging over many seeds does reveal a monotone gain, but this uncentered term is of order n^{-1} (cf. the perturbative expansions in [DGA20; RYH22]), much faster than empirical scaling exponents, and the number of seeds must be chosen in an n-dependent way. More importantly, in practice the number of training tokens is at least comparable to the width of the largest models. This fluctuation-dominated regime is therefore rarely observed, as it is far from compute-optimal recipes which grow the amount of training data with model size [HBM+22].

Diverging horizon: sketched linear regression (Section 4). This negative result motivates us to treat the training horizon T as a scaling variable. Since no general TP-like theory describes the limit of neural networks at a diverging horizon, we study a simplified model in which the joint width–horizon limit is tractable: width-n sketched linear regression on Gaussian inputs with covariance eigenvalues $\lambda_j = j^{-\alpha}$ and target coefficients $b_j^2 = j^{-\beta}$, a model extensively studied in the theory of scaling laws [BAP24; PPXP24; LWK+24]. The parameters are trained by T steps of GD, with the learning rate η tuned to minimize the population squared loss. The difficulty of the problem is governed by $\rho = (\beta - 1)/\alpha$, with larger ρ corresponding to an easier target, and we define $s = 1 - 1/\alpha$.

- **Growing horizons create a monotone gain from scale.** At infinite width, the optimal learning rate drifts to the stability edge as $2 - \Theta(\log T/T)$, because the slowly learned tail of the spectrum rewards the largest stable step (Proposition 1). At finite width, the correction to the tuned loss splits into a centered CLT fluctuation and a positive *capacity bias* of order $n^{-1}T^{-s}$ (Theorem 3). The fluctuation is a cross term with the infinite-width residual, which GD drives to zero, whereas the bias arises from the sketch's distortion of the target and decays only as T^{-s}. For targets of finite norm ($\rho > 1$), the bias therefore dominates once $T^{\rho+1-2s} \gg n$, and the wider model achieves lower loss with probability tending to one. The joint scaling of width and horizon thus produces the monotone gain from scale on which transfer relies.
- **Necessary & sufficient conditions for fast transfer with rate separation.** We show that the learning rate tuned at width n incurs only a lower-order loss at infinite width when $T \asymp n^{\gamma}$ with $\gamma < 1/2$, and that this window is sharp (Theorem 3 and Proposition 12). Since the curvature at the optimum scales as $T^{1-\rho}$, which grows or vanishes with scale depending on the problem difficulty, the bounded-curvature condition of [GWB25] no longer applies. We therefore compare μ-transfer (a grid search on a width-n proxy, followed by one width-m target run) directly with every direct grid search under the same compute budget, for two horizon scalings with $\gamma < 1/2$:
  - *Proportional horizon:* each width-k model is trained for $T_k \asymp k^{\gamma}$ steps so that larger models are trained longer. Under a grid-proximity condition, transfer is fast with rate separation (Definition 1) iff $\rho_-(\gamma) < \rho < 1/\gamma + s$ for some $\rho_-(\gamma)$. Above this range (easy problems), even the coarsest direct grid attains the optimal rate, so tuning brings little benefit. Below it , the components of the target learned by the small and large models differ enough that learning rate does not transfer, in line with empirical observations [GWB25] that transfer can fail on "hard samples".
  - *Common horizon:* all runs use the same horizon $T = \lfloor m^{\gamma} \rfloor$. We show that transfer is again fast with rate separation on an explicit, nonempty range of ρ, where the target model also achieves lower tuned loss than the proxy with probability tending to one.
- **Compute optimality and stochastic gradients.** A practically relevant regime is *compute-optimal scaling* [HBM+22; PPXP24], which balances optimization error against model capacity under a fixed compute budget. In this regime, we have fast transfer if and only if $\beta \geq 3\alpha$, which is exactly when the compute-optimal horizon stays inside the window $T \lesssim \sqrt{n}$ (Theorem 4). Finally, we outline how stochastic gradients reshape the loss vs. learning-rate curve and discuss the fast transfer implications (Section 4.3).

Related work. Our CLT adds the next order to the laws of large numbers of TPs [Yan20; YH21; YL23]. Related finite-width fluctuations have been characterized for mean-field networks [SS20; RVE22; CRBVE20; PN21], heuristically via dynamical mean-field theory [BP23], and through perturbative $1/n$ expansions near initialization [DGA20; HN20; RYH22]. From the perspective of scaling laws, [BDK+24] distinguished *variance-limited* scaling, an $O(n^{-1})$ correction to the averaged loss around a well-defined infinite-width limit, from *resolution-limited* scaling with spectrum-dependent exponents; see also [ABSP23; VAB+23]. Our fixed-horizon regime is variance-limited, with the horizon playing the role of the sample size, yet a single run is dominated by a centered $n^{-1/2}$ fluctuation; growing the horizon with the width instead brings out a one-sided capacity bias. Linear models with power-law spectra are the standard solvable models of scaling laws [MRS22; BDK+24; BAP24; PPXP24; LWK+24; DLM24], but these works do not study how the optimal learning rate transfers across scale. On the theory of HP transfer, [NMHO24] related learning-rate transfer to the width-consistency of sharpness, and [Hay26] proved that the optimal learning rate of linear networks converges with width. Our quantitative framework builds on [GWB25; KB26], which relate the benefit of μ-transfer to the convergence rates of the optimal loss and HP.

2 Background: Fast Transfer and Grid Search

Let $\phi_n(\eta)$ be the loss after training a model of width n with HP η; lower is better. For the learning rate in μP, η is the width-independent base learning rate, and each layer uses η times a fixed power of n. Let $\eta_n^\star \in \arg\min_{\eta \in \mathcal{E}} \phi_n(\eta)$ over a compact interval $\mathcal{E} = [\eta_-, \eta_+]$, and $\phi_n^\star := \phi_n(\eta_n^\star)$. HP transfer presumes a limit $\phi_n \to \phi_\infty$ whose minimizer $\eta_\infty^\star$ is unique, so that $\eta_n^\star \to \eta_\infty^\star$; this alone does not say whether estimating $\eta_\infty^\star$ on a smaller model saves compute.

Fast transfer. Following [GWB25], define the loss gap and the HP gap

$$a_n := |\phi_n^\star - \phi_\infty^\star|, \qquad b_n := |\eta_n^\star - \eta_\infty^\star|. \tag{1}$$

The loss gap measures the benefit of scale, and the HP gap measures how accurately a proxy of width n locates the optimal infinite-width HP. If ϕ_∞ has curvature $\mu := \phi''_\infty(\eta^\star_\infty) > 0$, training a large model with $\eta^\star_n$ instead of $\eta^\star_\infty$ costs about $\frac{1}{2}\mu b_n^2$ in loss. [GWB25] call transfer *fast* if $\mu b_n^2 = o(a_n)$: the error of the proxy's HP is negligible compared with the gain from scale. To relate this condition to computational cost, they compare direct grid search at the target scale with grid search on a smaller proxy followed by a single target-scale run. Under their regularity assumptions, fast transfer is equivalent to *useful* transfer: achieving the same loss with asymptotically less compute than direct tuning. Their analysis assumes a constant curvature which does not hold in the setting of Section 4. We therefore define *fast transfer* directly in terms of computational savings, as formalized below.

Grid search and its cost. Let C_k be the cost of one training run at width k; in Section 4, $C_k = kT$ for T steps. We select HPs by searching over uniform grids,

$$\mathcal{G}_G := \{\eta_- + j(\eta_+ - \eta_-)/G : 0 \le j \le G\}, \qquad \bar{\eta}_k \in \arg\min_{\eta \in \mathcal{G}_{G_k}} \phi_k(\eta),$$

so a grid search at width k costs $(G_k + 1)C_k$. Given a budget $\mathcal{F}$, we compare two protocols. *Direct tuning* runs a grid search at a single width k, with $(G_k + 1)C_k \le \mathcal{F}$, and returns $\phi_k(\bar{\eta}_k)$. *Transfer* runs a grid search on a proxy of width n, then trains one independent target model of width m with the selected HP, with $(G_n + 1)C_n + C_m \le \mathcal{F}$, and returns $\phi_m(\bar{\eta}_n)$.

Definition 1 (Fast transfer)**.** *Let all widths and grid sizes be fixed powers of $\mathcal{F}$. For a loss metric ϕ, transfer is* fast *if there exists a transfer allocation (n, G_n, m) such that for every direct allocation (k, G_k): $\mathbb{P}\{\phi_m(\bar{\eta}_n) < \phi_k(\bar{\eta}_k)\} \to 1$, as $\mathcal{F} \to \infty$. In addition, if there exist $\epsilon > 0$ such that*

$$\frac{|\phi_m(\bar{\eta}_n) - \phi^\star_\infty|}{|\phi_k(\bar{\eta}_k) - \phi^\star_\infty|} = o_{\mathbb{P}}(\mathcal{F}^{-\epsilon}),$$

we call it fast transfer with rate separation. *When the horizon varies between runs, each excess loss is measured against the infinite-width optimum at the horizon of that run.*

Roughly speaking, fast transfer is defined by comparing the achieved loss (direct tuning vs. transfer) at the same compute $\mathcal{F}$, whereas *fast transfer with rate separation* is a stronger condition that requires the proxy-tuned (transferred) loss to converge at a faster rate to the infinite-scale limit.

3 Second-order Tensor Programs at a fixed horizon

3.1 A central limit theorem for Tensor Programs

A Tensor Program describes a general neural computation at a fixed optimization horizon, via a finite sequence of operations on vectors and scalars [Yan20]. We work with the class of $\mathrm{NETSOR}^{\top+}$ programs (see Appendix A.1 for formal definitions). Starting from Gaussian matrices and vectors, these programs use three commands: *MatMul* multiplies a previously computed vector by an initial matrix W or $W^\top$; *Nonlin* applies a function coordinatewise to previously computed vectors, possibly using previously computed scalar observables as parameters; and *Moment* computes a scalar observable $\frac{1}{n}\sum_{\alpha=1}^n \varphi(x^1_\alpha, \dots, x^k_\alpha)$, where $x^1, \dots, x^k$ are previously computed vectors. In particular, these operations can represent MLP forward and backward passes and any fixed number of gradient steps on a fixed dataset. The master theorem of [Yan20] establishes a law of large numbers for these computations. As $n \to \infty$ with the program fixed, the empirical distribution of the vector coordinates converges to an explicitly characterized limiting law. Each scalar observables computed by Moment converges to the expectation of its defining function under this law. Under μP, these limits describe the infinite-width network and its feature-learning dynamics [YH21].

We introduce the *split* $\mathrm{NETSOR}^{\top+}$ program to characterize second-order fluctuations of any $\mathrm{NETSOR}^{\top}$ program — this covers finite-horizon GD training of multi-layer perceptrons with pseudo-Lipscthiz activation functions. More specifically, in a split program, all matrix-vector products are computed in an initial $\mathrm{NETSOR}^{\top}$ phase that neither computes nor uses scalar observables; scalar computations follow in a second phase, with no further matrix-vector products.

Theorem 1 (Second-order master theorem)**.** *Let Π_n be a finite split $\mathrm{NETSOR}^{\top+}$ program over the Gaussian setup of Definition 9, whose nonlinearities satisfy Assumption 2. Let $Y_n = (Y_n^1, \dots, Y_n^d)$ be a fixed list of its scalar observables, with deterministic first-order limits $\mathring{y}$. Then*

$$\sqrt{n}\,(Y_n - \mathring{y}) \Rightarrow \mathcal{N}(0, \Sigma_Y), \tag{2}$$

where Σ_Y is computed by the finite recursion (109) over the commands of Π_n, initialized by the joint fluctuations of the initial scalar observables.

The limiting covariance captures both fluctuations of coordinate averages and correlations introduced by reusing weights. After conditioning on earlier products $Wx^1, \dots, Wx^k$, the unexplained part of a new product Wx is Gaussian noise with a finite-rank projection removed [Bol14; BM11]. This projection does not affect the law of large numbers, but can contribute at order $n^{-1/2}$ to the scalar observables and must therefore be included in the CLT. We account for these contributions by revealing Gaussian vectors in command order and updating the fluctuation state through the linear recursion $Z^{(\kappa)} = A_\kappa Z^{(\kappa-1)} + B_\kappa \zeta_\kappa$, where ζ_κ is an independent Gaussian innovation and the deterministic matrices A_κ, B_κ are computed from the first-order limit (Appendix A.1).

3.2 Learning-rate transfer for μP MLPs

Consider an MLP with $L-1$ hidden layers of width n and polynomial activation σ, trained on fixed data $X = (x_1, \dots, x_m) \in \mathbb{R}^{d\times m}$ with labels $y \in \mathbb{R}^m$. At step t, its weights are $W_1^t \in \mathbb{R}^{n\times d}$, $W_i^t \in \mathbb{R}^{n\times n}$ for $2 \le i \le L-1$, and $V^t \in \mathbb{R}^n$. Their initial entries are mutually independent centered Gaussians with fixed layerwise variances. The hidden-layer activations and predictions are

$$H_1^t = \sigma(d^{-1/2} W_1^t X), \qquad H_i^t = \sigma(n^{-1/2} W_i^t H_{i-1}^t), \qquad F_n^t = n^{-1} V^t H_{L-1}^t,$$

where $2 \le i \le L-1$, σ acts entrywise, $H_i^t \in \mathbb{R}^{n\times m}$ contains the i-th layer output on the m training inputs, and $F_n^t \in \mathbb{R}^m$ contains their predictions. The weights are initialized independently with centered Gaussian entries of fixed variances $\sigma_{W_\ell}^2$ and σ_V^2.

For a polynomial loss $\ell(F, y)$, we train for T full-batch GD steps using the μP updates $\theta^{t+1} = \theta^t - \eta n \nabla_\theta \ell(F_n^t, y)$ for each weight matrix θ (Appendix A.2). We define $\mathcal{L}_n(\eta) := \ell(F_n^T(\eta), y)$ as the training loss after T steps with learning rate η, and write $\mathcal{L}_\infty(\eta)$ for its infinite-width limit. Thus $\phi_n = \mathcal{L}_n$ in Section 2. Below, d, m, L, T remain fixed as $n \to \infty$.

Theorem 2 (Fluctuations of the learning-rate curve)**.** *Under the setup above, there exists a degree D such that $\mathcal{L}_n(\eta)$ is a degree-D polynomial in the learning rate where the coefficients $\mathcal{L}_n^{(k)}$ are scalar observables of a split* $\mathrm{NETSOR}^{\top+}$ *program independent of η. Consequently,*

$$\mathcal{L}_n(\eta) = \textstyle\sum_{k=0}^D \eta^k \mathcal{L}_n^{(k)}, \qquad \sqrt{n}(\mathcal{L}_n - \mathcal{L}_\infty) \Rightarrow \mathbb{G}(\eta) = \textstyle\sum_{k\le D} \eta^k \mathbb{G}_k,$$

where $\{\mathbb{G}_k\}$ is the joint Gaussian limit of the rescaled coefficients $\{\mathcal{L}_n^{(k)}\}$ (Theorem 1). The convergence holds jointly with its derivatives and uniformly on compact sets.

Furthermore, if (i) $\lim_n \frac{1}{n}(H_{L-1}^0)^\top H_{L-1}^0 \succ 0$, (ii) $\sigma_V^2 > 0$, and (iii) $\nabla_F \ell(0, y) \neq 0$, then $s^2(\eta) := \operatorname{Var} \mathbb{G}(\eta) > 0$ for all but at most $2D$ values of η.

The above conditions only need to hold at initialization $\eta = 0$: condition (ii) requires V to have non-zero initialization, while conditions (i) (iii) hold almost surely for non-degenerate data. The same results hold for the loss on a fixed finite set of held-out inputs. Uniform convergence of the whole curve at rate $n^{-1/2}$ then gives the gaps (1):

Corollary 1 (Rates at a fixed horizon)**.** *Suppose $\mathcal{L}_\infty$ has a unique minimizer $\eta_\infty^\star \in \operatorname{int} \mathcal{E}$ with $\mu = \mathcal{L}_\infty''(\eta_\infty^\star) > 0$ and $s(\eta_\infty^\star) > 0$. Then*

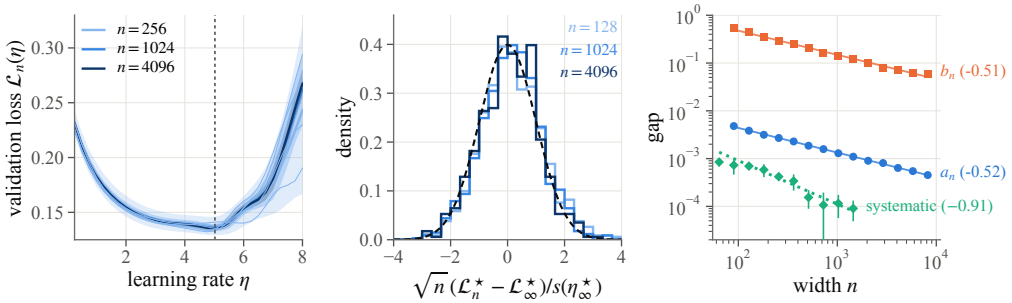
$$\sqrt{n}\,(\mathcal{L}_n^\star - \mathcal{L}_\infty^\star) \Rightarrow \mathcal{N}\big(0, s^2(\eta_\infty^\star)\big), \qquad \sqrt{n}\,(\eta_n^\star - \eta_\infty^\star) \Rightarrow -\tfrac{\mathbb{G}'(\eta_\infty^\star)}{\mathcal{L}_\infty''(\eta_\infty^\star)}.$$

In particular $a_n \asymp_{\mathbb{P}} n^{-1/2}$ and $b_n = O_{\mathbb{P}}(n^{-1/2})$, so $\mu b_n^2 = O_{\mathbb{P}}(n^{-1}) = o_{\mathbb{P}}(a_n)$. Moreover, for independent initializations at widths $n \le m$ with $n \to \infty$, $\mathbb{P}\{\mathcal{L}_m^\star < \mathcal{L}_n^\star\} \to \frac{1}{2}$.

Figure 2 confirms these predictions on a three-layer μP network.

3.3 What a fixed horizon can and cannot capture

The mechanism behind Corollary 1 is generic. A random perturbation of size ϵ of a function with a nondegenerate minimum moves the minimizer and minimum value by $O(\epsilon)$. The unperturbed function, evaluated at the new minimizer, changes only by $O(\epsilon^2)$, since it is stationary at its own minimum. Taking $\epsilon = n^{-1/2}$ therefore yields $a_n \asymp n^{-1/2}$ and $\mu b_n^2 \asymp n^{-1}$. The argument applies verbatim to other HPs that enter the program polynomially, such as momentum or weight decay. At fixed horizon, the criterion $\mu b_n^2 = o(a_n)$ thus follows automatically whenever a nondegenerate infinite-width limit exists. Hence it does not distinguish HPs that transfer well from those that do not.



(a) Loss curves concentrate **(b)** Tuned loss: centered Gaussian **(c)** Gaps decay as $n^{-1/2}$

Figure 2: Fixed-horizon fluctuations of a 3-layer μP MLP (tanh, $T = 4$ GD steps, validation loss; widths 64 to 8192; see Appendix A.6 for details). **(a)** Mean ± 2 s.d. over 128 initializations, and five single runs at $n = 256$; the dotted line marks $\eta_\infty^\star$. **(b)** The rescaled tuned loss matches the standard normal density (dashed) at every width. A width-8192 network has lower tuned loss than an independent width-128 network with probability only 0.55. **(c)** The loss gap a_n and the HP gap b_n decay as $n^{-1/2}$ (fitted exponents in parentheses). The systematic part $\mathbb{E}\mathcal{L}_n^\star - \mathcal{L}_\infty^\star$ (± 2 s.e.) is consistent with n^{-1} (fitted exponent -0.91 ± 0.19, dotted) and lies well below a_n.

More importantly, the premise of loss monotonicity in Section 2 breaks down in the fixed-T asymptotic. The leading finite-width correction is a centered Gaussian fluctuation of order $n^{-1/2}$: increasing width reduces variance, but does not systematically improve performance. In Figure 2, for instance, a network of width 8192 outperforms one of width 128 with probability only 0.55. This violates the monotonicity in scale assumed by [GWB25] and observed in scaling laws [KMH+20; HBM+22]. The monotone improvement appears only at the next order, approximately n^{-1} in Figure 2(c), consistent with the variance-limited $1/n$ correction to averaged loss in [BDK+24]. Averaging over initializations suppresses the leading fluctuation and exposes systematic finite-width corrections [VAB+23], but resolving the n^{-1} effect in Figure 2(c) requires a number of seeds that grows with n. Moreover, the resulting gain is of the same order as the transfer cost, $\mu b_n^2 \asymp n^{-1}$. Hence the condition $\mu b_n^2 = o(a_n)$ is satisfied only because a_n is dominated by zero-mean finite-width noise, rather than by a genuine systematic benefit from scale.

The relevant regime for transfer is therefore one in which training time is not fixed as the model size grows. This aligns with the compute-optimal scaling in which the number of training tokens increases with the number of parameters [HBM+22; DZN+25], so a theory of HP transfer should allow $T \to \infty$ jointly with $n \to \infty$. No general TP theory currently covers such diverging horizons. We therefore turn to the simplest model in which this joint limit can be analyzed explicitly.

4 Sketched Linear Regression under Growing Horizons

Consider noiseless linear regression with centered inputs $x \in \mathbb{R}^d$, covariance $\Sigma = \mathbb{E}[xx^\top]$ normalized so that $\|\Sigma\|_{\mathrm{op}} = 1$, and labels $y = \langle x, w^\star \rangle$. The dimension d may be infinite, with $\operatorname{tr}\Sigma < \infty$ (Appendix A.3). A width-n sketched linear model predicts through a Gaussian sketch $W_n \in \mathbb{R}^{n \times d}$,

$$f_n(x;\theta) = \langle W_n x, \theta \rangle, \qquad \theta \in \mathbb{R}^n, \qquad (W_n)_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/n),$$

where n is both the number of random features and the number of trainable parameters. We train θ by population GD with a constant learning rate η,

$$L_n(\theta) = \tfrac{1}{2}\mathbb{E}\big[(f_n(x;\theta) - y)^2 \mid W_n\big], \qquad \theta_{t+1} = \theta_t - \eta \nabla L_n(\theta_t), \qquad \theta_0 = 0,$$

and write $L_{n,T}(\eta) := L_n(\theta_T)$ for the loss after T steps. Its infinite-width limit $L_{\infty,T}(\eta)$ is the loss of the same GD on the unsketched model $\langle x, w \rangle$ from $w_0 = 0$. Let $b := \Sigma^{1/2} w^\star$ and $A_n := \Sigma^{1/2} W_n^\top W_n \Sigma^{1/2}$, the sample covariance of n independent $\mathcal{N}(0, \Sigma)$ vectors. Each GD step multiplies the residual $\Sigma^{1/2}(W_n^\top \theta_t - w^\star)$ by $I - \eta A_n$, so (Appendix A.3)

$$L_{n,T}(\eta) = L_T(A_n, \eta), \qquad L_{\infty,T}(\eta) = L_T(\Sigma, \eta), \qquad L_T(A, \eta) := \tfrac{1}{2}\big\langle b, (I - \eta A)^{2T} b \big\rangle. \tag{3}$$

Throughout this section, we optimize over $\eta \in [1, 2]$, where $2 = 2/\lambda_{\max}(\Sigma)$ is the stability edge of GD at infinite width, and denote the width-n and infinite-width optimal learning rates by $\eta_{n,T}$ and $\eta_{\infty,T}$. Since $\eta \mapsto L_T(A, \eta)$ is convex, either learning rate also minimizes the corresponding loss over all $\eta \geq 0$ whenever it lies in $(1, 2)$. This holds for $\eta_{\infty,T}$ when T is large and, with probability tending to one, for $\eta_{n,T}$ (Theorem 3).

Assumption 1 (Source-capacity condition). *In an eigenbasis of Σ, the eigenvalues and coordinates $b_j = \sqrt{\lambda_j}\, w_j^\star$ satisfy $\lambda_j = j^{-\alpha}$ and $b_j^2 = j^{-\beta}$, $\alpha, \beta > 1$. We set $\rho := \frac{\beta-1}{\alpha}$, $s := 1 - \frac{1}{\alpha} \in (0,1)$.*

The capacity exponent α sets how fast the spectrum decays, and the source exponent β controls how complex the labels y are [CDV07]. Larger ρ means an easier learning problem, and $\rho > 1$ implies that $w^\star$ has finite norm $\|w^\star\|^2 < \infty$; for $\rho \le 1$, y is defined as an L^2 limit (Appendix A.3).

4.1 Optimal learning rates and finite-width losses

Proposition 1. *As $T \to \infty$,*

$$\eta_{\infty,T} = 2 - \tfrac{(\rho+1)\log T + O(1)}{2T}, \qquad L_{\infty,T}(\eta_{\infty,T}) \asymp T^{-\rho}, \qquad L''_{\infty,T}(\eta_{\infty,T}) \asymp T^{1-\rho}. \tag{4}$$

Two parts of the loss compete: the leading eigendirection of Σ contributes $\frac{1}{2}(1-\eta)^{2T}$, which favours smaller learning rate; the contribution from all remaining eigendirections is of order $(\eta T)^{-\rho}$ and decreases as the learning rate increases. Balancing their derivatives, $T|1-\eta_{\infty,T}|^{2T} \asymp T^{-\rho}$, gives $2 - \eta_{\infty,T} = \{(\rho+1)\log T + O(1)\}/(2T)$ (Figure 6(a)), consistent with the observation that well-tuned GD operates near its stability edge. The curvature $T^{1-\rho}$ diverges for hard problems ($\rho < 1$) and vanishes for easy ones ($\rho > 1$) (Figure 6(b)).

Let $Z_{n,T} := D_A L_T(\Sigma, \eta_{\infty,T})[A_n - \Sigma]$, where D_A is the Fréchet derivative in A, be the first-order effect of the finite sketch, with variance $\sigma^2_{n,T}$. Let $B_T(\eta) \equiv 0$ when $\rho \le 1$, and for $\rho > 1$,

$$B_T(\eta) := \tfrac{\|w^\star\|^2}{2} \textstyle\sum_{j\ge 2} \lambda_j (1-\eta\lambda_j)^{2T}, \tag{5}$$

Theorem 3 (Growing horizon). *Let Assumption 1 hold, $T = T_n \to \infty$, and $T^2/n \to 0$. Then $\mathbb{P}\{\eta_{n,T} \in (1,2)\} \to 1$, $Z_{n,T}/\sigma_{n,T} \Rightarrow \mathcal{N}(0,1)$, and*

$$\sigma_{n,T} \asymp n^{-1/2} \times \begin{cases} T^{-\rho}, & 0 < \rho < 1, \\ T^{-(\rho+1)/2}, & \rho > 1, \end{cases} \tag{6}$$

$$\tfrac{B_T(\eta_{\infty,T})}{n} \sim \tfrac{\|w^\star\|^2 \Gamma(s)}{2\alpha 4^s} \tfrac{T^{-s}}{n} \qquad (\rho > 1). \tag{7}$$

With $\varepsilon_{n,T} := \sigma_{n,T} + n^{-1} B_T(\eta_{\infty,T})$, we have

$$L_{n,T}(\eta_{n,T}) - L_{\infty,T}(\eta_{\infty,T}) = Z_{n,T} + n^{-1} B_T(\eta_{\infty,T}) + o_{\mathbb{P}}(\varepsilon_{n,T}). \tag{8}$$

$$|L_{n,T}(\eta_{n,T}) - L_{\infty,T}(\eta_{\infty,T})| = \Theta_{\mathbb{P}}(\varepsilon_{n,T}), \quad L_{\infty,T}(\eta_{n,T}) - L_{\infty,T}(\eta_{\infty,T}) = o_{\mathbb{P}}(\varepsilon_{n,T}). \tag{9}$$

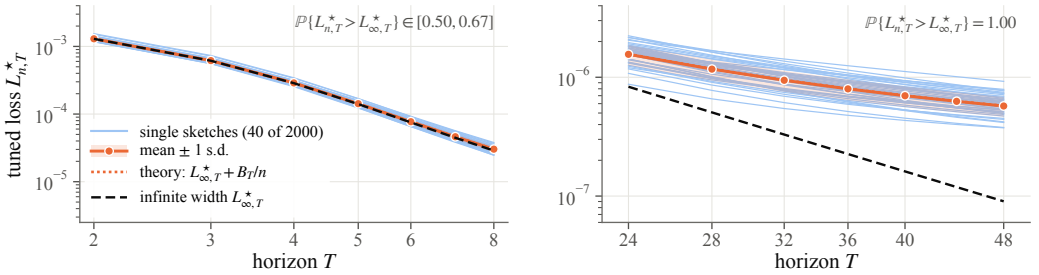
The proof is given in Appendix I. When $\rho \le 1$, $B_T(\eta) \equiv 0$, and the leading finite-width correction is a two-sided fluctuation, as at a fixed training horizon. When $\rho > 1$, the correction also contains a systematic positive bias. To see where these contributions come from, write the loss difference as

$$L_T(A_n,\eta) - L_T(\Sigma,\eta) = \underbrace{\langle (I-\eta\Sigma)^T b,\, \Delta_T b\rangle}_{\text{fluctuation}} + \underbrace{\tfrac{1}{2}\|\Delta_T b\|^2}_{\text{nonnegative}}, \quad \Delta_T := (I-\eta A_n)^T - (I-\eta\Sigma)^T.$$

The cross term is linear in $A_n - \Sigma$ to leading order and contributes to the fluctuation $Z_{n,T}$. The leading expectation of the quadratic term gives the bias $n^{-1}B_T(\eta_{\infty,T})$. As training proceeds, GD reduces the infinite-width residual and hence the cross term. Sketching errors along directions with small eigenvalues decay more slowly, allowing the quadratic term to dominate. For $\rho > 1$, this transition occurs when $T^{\rho+1-2s} \gg n$. In this regime, the larger model achieves a lower loss with probability tending to one, and the optimally tuned loss follows $L_{n,T}(\eta_{n,T}) \approx c_0 T^{-\rho} + c_1 n^{-1} T^{-s}$ consistent with the scaling law in [PPXP24]. Figure 3 illustrates the fluctuation-dominated and bias-dominated regimes. Figure 6(c) shows the probability that the larger model achieves a lower loss as a function of the ratio of bias to fluctuation scale, $B_T(\eta_{\infty,T})/(n\sigma_{n,T})$.

4.2 Transfer under a compute budget

Theorem 3 shows that the loss penalty from transferring the width-n optimal learning rate to the infinite-width model is negligible compared with the finite-width loss difference (Equations (9)), but does not yet establish whether transfer saves compute. We therefore compare transfer tuning with *direct grid search* at the target width (Section 2). We assign a cost $C_k = kT$ to training a width-k model for T steps, counting scalar parameter updates, and select learning rates from the uniform grids. We consider two choices of training horizon. Under *proportional horizons*, each width-k model is trained for $T_k \asymp k^\gamma$ steps. Under a *common horizon*, all grid evaluations and training runs use $T = \lfloor m^\gamma \rfloor$ steps, where m is the target width. See Appendix A.4 for full statement.



(a) Short horizons ($T \leq 8$): centered fluctuation **(b)** Long horizons ($T \geq 24$): positive capacity bias

Figure 3: The two finite-width regimes at a single width $n = 2^{16}$ ($\rho = 3$, $\alpha = 2$). The loss at the width-n optimal learning rate decreases monotonically in T. **(a)** At short horizons, single sketches fall on both sides of the infinite-width optimal loss $L_{\infty,T}(\eta_{\infty,T})$: the centered term $Z_{n,T}$ dominates. **(b)** At long horizons, every sketch is worse than the infinite-width model, and the mean matches $L_{\infty,T}(\eta_{\infty,T}) + n^{-1}B_T(\eta_{\infty,T})$.

Proposition 2 (Fast transfer with rate separation; informal)**.** *Let Assumption 1 hold and* $0 < \gamma < 1/2$.

(i) At proportional horizons, and under a grid-proximity condition, transfer is fast with rate separation (Definition 1) if and only if $\rho_-(\gamma) < \rho < 1/\gamma + s$*, where the threshold* $\rho_-(\gamma)$ *is explicit and does not depend on* α *(Proposition 4).*

(ii) At a common horizon, transfer is fast with rate separation whenever $\max\{1/\gamma - 2 + 3s,\ 4 + s - 1/\gamma\} < \rho < 1/\gamma + s$*, a nonempty range. There, the larger model also has lower optimal loss than the smaller model with probability tending to one (Theorem 7).*

For $\rho > 1/\gamma + s$, the finite-width bias $n^{-1}T^{-s}$ dominates the infinite-width loss $T^{-\rho}$, and taking any $\eta \in [1, 2 - \widetilde{O}(1/T)]$ already achieves the optimal polynomial rate. Under proportional horizons, rate separation fails when $\rho < \rho_-(\gamma)$: the smaller model is trained for fewer steps and selects a learning rate that is too small for the larger model's longer horizon, slowing its convergence. A common horizon eliminates this mismatch. For sufficiently small ρ, however, the loss reduction from increasing width is smaller than fluctuations across independent sketches, so the smaller model's selected learning rate may not perform well on the larger model.

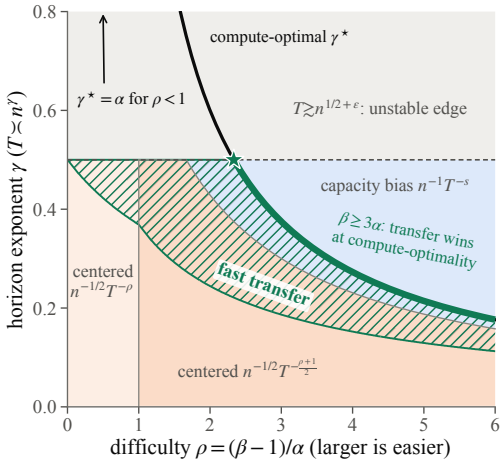
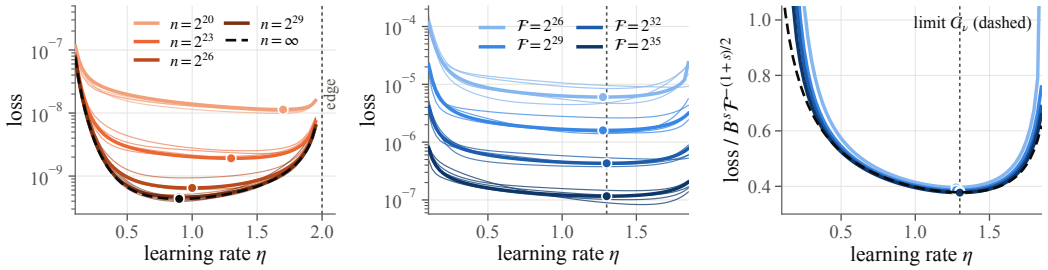


Figure 4: Phase diagram for proportional horizons $T_k \asymp k^\gamma$ ($\alpha = 1.5$, cost nT). Shading: the leading finite-width correction of Theorem 3, which covers $\gamma < 1/2$; above $\gamma = 1/2$, the stability edge fluctuates more than its distance to the optimum. Hatched: fast transfer with rate separation (Definition 1). Black: the compute-optimal exponent $\gamma^\star = 1/(\rho - s)$. Its green part, $\beta \geq 3\alpha$, is where transfer beats every direct grid search by a constant factor (Theorem 4).

Compute-optimal scaling. If the learning rate could be tuned at no cost, the best allocation under $nT \leq \mathcal{F}$ would attain an asymptotically sharp frontier $\mathcal{L}(\mathcal{F})$ with two phases (Proposition 5 and Corollary 2). For $\rho < 1$, finite width is negligible until the horizon reaches $T \asymp n^\alpha$, this gives $\mathcal{L}(\mathcal{F}) \asymp \mathcal{F}^{-\alpha\rho/(1+\alpha)}$. For $\rho > 1$, the optimal allocation balances the two terms of the scaling law, $T^{-\rho} \asymp n^{-1}T^{-s}$. This gives $T \asymp n^{1/(\rho-s)} = n^{\alpha/(\beta-\alpha)}$ and $\mathcal{L}(\mathcal{F}) \asymp \mathcal{F}^{-(\beta-1)/\beta}$. Both exponents can also be recovered from the population gradient descent terms in the loss decomposition of [PPXP24]; the following theorem characterizes when HP transfer attains this compute-optimal exponent and outperforms direct tuning.

Theorem 4 (Fast transfer under compute-optimal scaling)**.** *Let Assumption 1 hold, and compare transfer with direct tuning under the same budget* $\mathcal{F}$.

(i) If $\beta \geq 3\alpha$*, the learning rate can be selected at width* $n = o(m)$ *and used at width* m*. At proportional or common horizons, the resulting loss reaches* $\mathcal{L}(\mathcal{F})$ *to leading order, and we have fast transfer with probability tending to one.*



(a) Fixed $T = 512$, $B = 16384$ **(b)** Compute-optimal: stable optimum **(c)** Rescaled: the curves collapse

Figure 5: Online SGD in sketched linear regression ($\alpha = 2$, $\beta = 8$). Thin lines are single SGD runs, each on its own sketch, thick lines expected losses averaged over sketches. **(a)** At a fixed horizon and batch ($T = 512$, $B = 16384$), as the width grows, the curves converge to the infinite-width curve (dashed), and the optimal learning rate moves from near the edge to its interior minimizer ($n/(BT)$ increases from $1/8$ to 64). **(b)** At compute-optimality ($n = \sqrt{\mathcal{F}}$, $B \approx 1.4\,\mathcal{F}^{1/4}$, $T = \mathcal{F}/(nB)$), the optimal learning rate stays at the minimizer η_ν of the limit (dotted). **(c)** Divided by $B^s\mathcal{F}^{-(1+s)/2}$, the curves of (b) collapse onto the limit G_ν (dashed).

(ii) If $\beta < 3\alpha$, then for every horizon schedule attaining the compute-optimal rate and every choice of $1 \le n \le m$ and $G_n \ge 1$, there exists an direct tuning procedure that beats transfer by an arbitrarily large factor with non-zero probability (Proposition 6).

The compute-optimal horizon exponent $\alpha/(\beta - \alpha)$ is at most $1/2$ exactly when $\beta \ge 3\alpha$, and for $\rho \le 1$ the exponent is $\alpha > 1/2$. Transfer reduces computational cost under compute-optimal scaling when training stays within the regime $T = O(\sqrt{n})$ (green curve in Figure 4). Outside this regime, the learning rate selected by the smaller model may not lie below the larger model's stability edge, and thus convergence of the optimal learning rate thus does not guarantee that transfer saves compute.

4.3 Online stochastic gradient descent

With population GD, the optimal learning rate converges to the stability edge $\eta = 2$ (Proposition 1). This is an artifact of noiseless training: for large T, population GD penalizes a step only when it is close to unstable, whereas tuned learning rates of stochastic training lie well inside the stable range. We illustrate this difference in the same sketched linear regression model with $\beta > 2\alpha$, where each step averages the gradients of B fresh samples. At compute-optimal scaling, $n = \nu\sqrt{\mathcal{F}}$ and $T = \mathcal{F}/(nB)$ for a fixed $\nu > 0$, with batch size $B = \mathcal{F}^{\xi+o(1)} \to \infty$, where $0 \le \xi < (\rho - 1 - s)/(2(\rho - s))$. Using the deterministic equivalent of [PPXP24], the loss curve of a single run, divided by its order $B^s\mathcal{F}^{-(1+s)/2}$, converges in probability to

$$G_\nu(\eta) = \frac{\Gamma(s)}{2\alpha 2^s}\,\eta^{-s}\Big(\underbrace{\nu^{s-1}\|w^\star\|^2}_{\text{capacity bias}} + \underbrace{\tfrac{s}{4}\,\nu^{s+1}\textstyle\sum_{j\ge 1}\frac{b_j^2}{\lambda_j(1-\eta\lambda_j/2)}}_{\text{gradient noise}}\Big), \qquad 0 < \eta < 2.$$

The capacity bias, of order $n^{-1}T^{-s}$, decreases all the way to the stability edge. The noise injected early in training, of order $B^{-1}T^{-1-s}$, diverges as $\eta \to 2$, where the top direction no longer damps it. Balancing the two, G_ν is minimized at some η_ν strictly inside $(0, 2)$, away from the stability edge (Figure 5). More importantly, SGD enables fast transfer through a different mechanism: at compute-optimal scaling, the loss curves collapse after rescaling [QXW+25; BZD+26]. Hence the optimal learning rate converges to η_ν across scale, and $G''_\nu(\eta_\nu) > 0$, so the rescaled curve has positive curvature at its optimum (Figure 5(b,c)). Away from compute-optimal scaling, by contrast, the optimal learning rate drifts with scale (Figure 5(a), where we fixed the time horizon T and scale n), which may hinder fast transfer. We leave a full characterization of this phenomenon to future work.

5 Conclusion

Hyperparameter transfer is a crucial component of training large models, yet it lacks proper mathematical foundations. We question the common wisdom based on the tensor programs line of work, by showing that the fixed horizon setting fails to capture key aspects of practical learning rate transfer, such as scaling laws and monotonicity of the loss. In the more realistic scenario where width and horizon both grow, we show that transfer can be beneficial in many regimes of interest, depending on spectral properties of the data. Understanding this beyond linear models is an important next step.

AI USE STATEMENT

Generative AI tools were used to revise the writing, assist with routine coding, clarify and fill in technical details after the proof strategies and core arguments had been established, check mathematical arguments, and identify relevant prior work. They were not used to formulate the paper's central ideas, design the problem setting, or derive its principal theoretical results. The authors take full responsibility for the manuscript and accompanying materials, including all AI-assisted text, mathematical claims, and code.

REPRODUCIBILITY STATEMENT

The appendices provide detailed assumptions, full theorem statements, and proofs for our Tensor Program and sketched linear regression results. Appendix A.6 describes the simulations, including data generation, hyperparameters, and the number of independent runs. All experiments use synthetic data and can be completed in a few hours on a laptop.

REFERENCES

[ABSP23] Alexander Atanasov, Blake Bordelon, Sabarish Sainathan, and Cengiz Pehlevan, *The onset of variance-limited behavior for networks in the lazy and rich regimes*, International Conference on Learning Representations, 2023.

[BAP24] Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan, *A dynamical model of neural scaling laws*, Proceedings of the 41st International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 235, PMLR, 2024, pp. 4345–4382.

[BDK+24] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma, *Explaining neural scaling laws*, Proceedings of the National Academy of Sciences **121** (2024), no. 27, e2311878121.

[BM11] Mohsen Bayati and Andrea Montanari, *The dynamics of message passing on dense graphs, with applications to compressed sensing*, IEEE Transactions on Information Theory **57** (2011), no. 2, 764–785.

[BNL+24] Blake Bordelon, Lorenzo Noci, Mufan Li, Boris Hanin, and Cengiz Pehlevan, *Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit*, International Conference on Learning Representations, 2024.

[Bol14] Erwin Bolthausen, *An iterative construction of solutions of the TAP equations for the Sherrington–Kirkpatrick model*, Communications in Mathematical Physics **325** (2014), no. 1, 333–366.

[BP23] Blake Bordelon and Cengiz Pehlevan, *Dynamics of finite width kernel and prediction fluctuations in mean field neural networks*, Advances in Neural Information Processing Systems, vol. 36, 2023.

[BZD+26] Shane Bergsma, Bin Zhang, Nolan Dey, Shaheer Muhammad, Gurpreet Gosal, and Joel Hestness, *Scaling with collapse: Efficient and predictable training of llm families*, International Conference on Learning Representations, vol. 2026, 2026, pp. 110701–110737.

[CDV07] Andrea Caponnetto and Ernesto De Vito, *Optimal rates for the regularized least-squares algorithm*, Foundations of Computational Mathematics **7** (2007), no. 3, 331–368.

[CRBVE20] Zhengdao Chen, Grant Rotskoff, Joan Bruna, and Eric Vanden-Eijnden, *A dynamical central limit theorem for shallow neural networks*, Advances in Neural Information Processing Systems, vol. 33, 2020.

[DGA20] Ethan Dyer and Guy Gur-Ari, *Asymptotics of wide networks from Feynman diagrams*, International Conference on Learning Representations, 2020.

[DLM24] Leonardo Defilippis, Bruno Loureiro, and Theodor Misiakiewicz, *Dimension-free deterministic equivalents and scaling laws for random feature regression*, Advances in Neural Information Processing Systems, vol. 37, 2024, pp. 104630–104693.

[DZN+25] Nolan Dey, Bin Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness, *Don't be lazy: CompleteP enables compute-efficient deep transformers*, Advances in Neural Information Processing Systems, vol. 38, 2025, pp. 152786–152818.

[EXW+24] Katie E. Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington, *Scaling exponents across parameterizations and optimizers*, Proceedings of the 41st International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 235, PMLR, 2024, pp. 12666–12700.

[GWB25] Nikhil Ghosh, Denny Wu, and Alberto Bietti, *Understanding the mechanisms of fast hyperparameter transfer*, arXiv preprint arXiv:2512.22768 (2025).

[Hay26] Soufiane Hayou, *A proof of learning rate transfer under μP*, Proceedings of the 29th International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 300, PMLR, 2026, pp. 3322–3330.

[HBM+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre, *Training compute-optimal large language models*, arXiv preprint arXiv:2203.15556 (2022).

[HL15] Erich Häusler and Harald Luschgy, *Stable convergence and stable limit theorems*, Probability Theory and Stochastic Modelling, vol. 74, Springer, 2015.

[HN20] Boris Hanin and Mihai Nica, *Finite depth and width corrections to the neural tangent kernel*, International Conference on Learning Representations, 2020.

[HNA+17] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou, *Deep learning scaling is predictable, empirically*, arXiv preprint arXiv:1712.00409 (2017).

[KB26] Dayal Singh Kalra and Maissam Barkeshli, *Quantifying hyperparameter transfer and the importance of embedding layer learning rate*, arXiv preprint arXiv:2605.21486 (2026).

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, *Scaling laws for neural language models*, arXiv preprint arXiv:2001.08361 (2020).

[Lin24] Lucas Lingle, *An empirical study of μP learning rate transfer*, arXiv preprint arXiv:2404.05728 (2024).

[LWK+24] Licong Lin, Jingfeng Wu, Sham M. Kakade, Peter L. Bartlett, and Jason D. Lee, *Scaling laws in linear regression: Compute, parameters, and data*, Advances in Neural Information Processing Systems, vol. 37, 2024, pp. 60556–60606.

[MAB+26] Bruno Mlodozeniec, Pierre Ablin, Louis Béthune, Dan Busbridge, Michal Klein, Jason Ramapuram, and Marco Cuturi, *Completed hyperparameter transfer across modules, width, depth, batch and duration*, International Conference on Learning Representations, 2026.

[MRS22] Alexander Maloney, Daniel A. Roberts, and James Sully, *A solvable model of neural scaling laws*, arXiv preprint arXiv:2210.16859 (2022).

[NMHO24] Lorenzo Noci, Alexandru Meterez, Thomas Hofmann, and Antonio Orvieto, *Super consistency of neural network landscapes and learning rate transfer*, Advances in Neural Information Processing Systems, vol. 37, 2024, pp. 102696–102743.

[Pie16] Rafał Pierzchała, *Markov's inequality and polynomial mappings*, Mathematische Annalen **366** (2016), 57–82.

[PN21] Huy Tuan Pham and Phan-Minh Nguyen, *Limiting fluctuation and trajectorial stability of multilayer neural networks with mean field training*, Advances in Neural Information Processing Systems, vol. 34, 2021.

[PPXP24] Elliot Paquette, Courtney Paquette, Lechao Xiao, and Jeffrey Pennington, *4+3 phases of compute-optimal neural scaling laws*, Advances in Neural Information Processing Systems, vol. 37, 2024, pp. 16459–16537.

[QXW+25] Shikai Qiu, Lechao Xiao, Andrew Gordon Wilson, Jeffrey Pennington, and Atish Agarwala, *Scaling collapse reveals universal dynamics in compute-optimally trained neural networks*, Proceedings of the 42nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 267, PMLR, 2025, pp. 50697–50720.

[RVE22] Grant Rotskoff and Eric Vanden-Eijnden, *Trainability and accuracy of artificial neural networks: An interacting particle system approach*, Communications on Pure and Applied Mathematics **75** (2022), no. 9, 1889–1935.

[RYH22] Daniel A. Roberts, Sho Yaida, and Boris Hanin, *The principles of deep learning theory*, Cambridge University Press, 2022.

[SS20] Justin Sirignano and Konstantinos Spiliopoulos, *Mean field analysis of neural networks: A central limit theorem*, Stochastic Processes and their Applications **130** (2020), no. 3, 1820–1852.

[Tao12] Terence Tao, *Topics in random matrix theory*, Graduate Studies in Mathematics, vol. 132, American Mathematical Society, Providence, RI, 2012.

[VAB+23] Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan, *Feature-learning networks are consistent across widths at realistic scales*, Advances in Neural Information Processing Systems, vol. 36, 2023.

[Ver26] Roman Vershynin, *High-dimensional probability: An introduction with applications in data science*, 2 ed., Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2026.

[Yan20] Greg Yang, *Tensor programs III: Neural matrix laws*, arXiv preprint arXiv:2009.10685 (2020).

[YH21] Greg Yang and Edward J. Hu, *Tensor programs IV: Feature learning in infinite-width neural networks*, Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 11727–11737.

[YHB+22] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao, *Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer*, arXiv preprint arXiv:2203.03466 (2022).

[YL23] Greg Yang and Etai Littwin, *Tensor programs IVb: Adaptive optimization in the infinite-width limit*, arXiv preprint arXiv:2308.01814 (2023).

[YYZH24] Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou, *Tensor programs VI: Feature learning in infinite depth neural networks*, International Conference on Learning Representations, 2024.