A Proof of Learning Rate Transfer under μP

Soufiane Hayou

Department of Applied Mathematics and Statistics
Johns Hopkins University

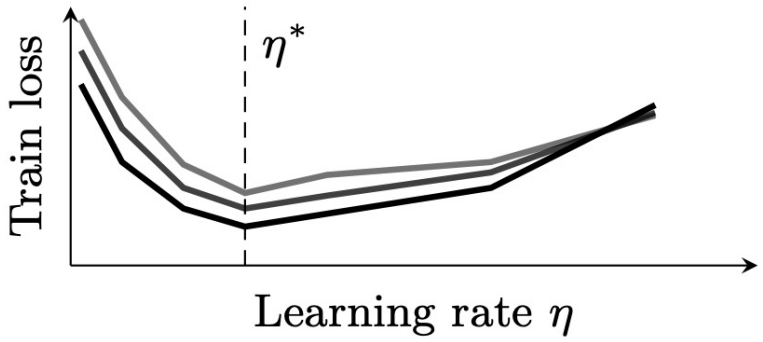
May 2nd, 2026

JOHNS HOPKINS
UNIVERSITY

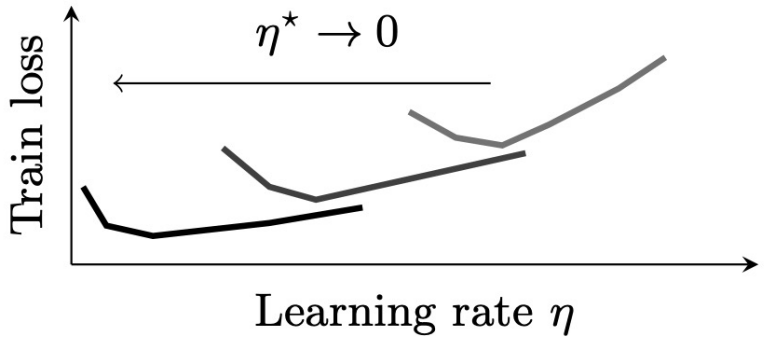
AISTATS

Learning Rate Transfer

✅ Good LR Transfer

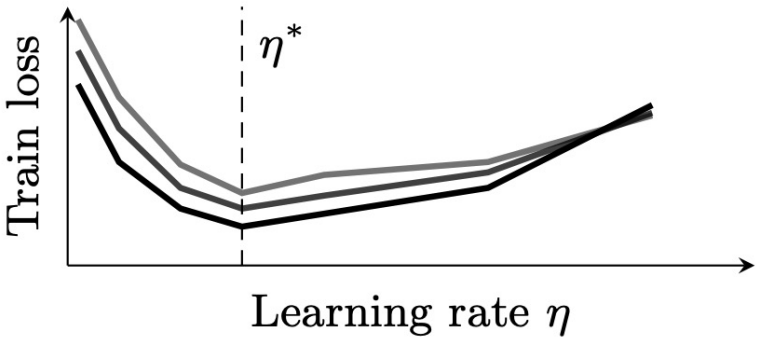


❌ No LR Transfer

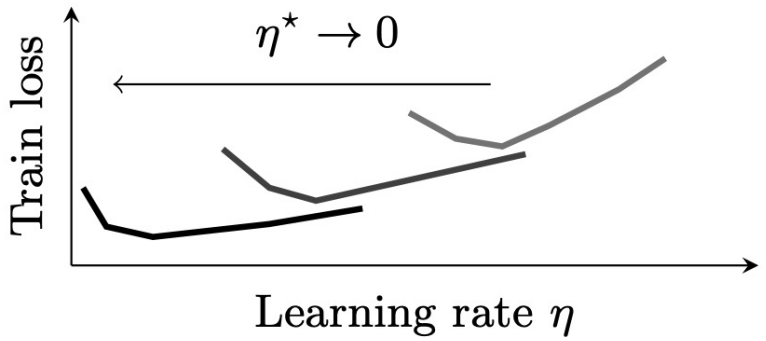


Learning Rate Transfer

✅ Good LR Transfer



❌ No LR Transfer



Tune LR on a small model ➔ use it for any model size

Lazy/Kernel Regime

$$\begin{cases} Y_0(x) = \frac{1}{\sqrt{d}} W_0 x \\ Y_\ell(x) = \frac{1}{\sqrt{n}} W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L]. \\ f(x) = \frac{1}{\sqrt{n}} W_{out} Y_L(x). \end{cases}$$

where $W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}$.

Neural Tangent Kernel

$$\begin{cases} Y_0(x) = \frac{1}{\sqrt{d}} W_0 x \\ Y_\ell(x) = \frac{1}{\sqrt{n}} W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L]. \\ f(x) = \frac{1}{\sqrt{n}} W_{out} Y_L(x). \end{cases}$$

where $W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}$.

- $Y_\ell^t(x)$: neural feature after t training steps (Gradient Descent)

Neural Tangent Kernel

$$\begin{cases} Y_0(x) = \frac{1}{\sqrt{d}} W_0 x \\ Y_\ell(x) = \frac{1}{\sqrt{n}} W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L]. \\ f(x) = \frac{1}{\sqrt{n}} W_{out} Y_L(x). \end{cases}$$

where $W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}$.

- $Y_\ell^t(x)$: neural feature after t training steps (Gradient Descent)

Then, as $n \to \infty$, $Y_l^t(x) - Y_l^0(x) = O_n(n^{-1/2})$ (coord-wise)

Neural Tangent Kernel

$$\begin{cases} Y_0(x) = \frac{1}{\sqrt{d}} W_0 x \\ Y_\ell(x) = \frac{1}{\sqrt{n}} W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L]. \\ f(x) = \frac{1}{\sqrt{n}} W_{out} Y_L(x). \end{cases}$$

where $W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}$.

- $Y_\ell^t(x)$: neural feature after t training steps (Gradient Descent)

Then, as $n \to \infty$, $Y_l^t(x) - Y_l^0(x) = O_n(n^{-1/2})$ (coord-wise)

No feature learning! (Kernel regime/Lazy Training)

Neural Tangent Kernel

$$\begin{cases} Y_0(x) = \frac{1}{\sqrt{d}} W_0 x \\ Y_\ell(x) = \frac{1}{\sqrt{n}} W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L]. \\ f(x) = \frac{1}{\sqrt{n}} W_{out} Y_L(x). \end{cases}$$

where $W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}$.

- $Y_\ell^t(x)$: neural feature after t training steps (Gradient Descent)

Then, as $n \to \infty$, $Y_l^t(x) - Y_l^0(x) = O_n(n^{-1/2})$ (coord-wise)

No feature learning! (Kernel regime/Lazy Training)

Jacot et al. (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks"
Chizat et al. (2019) "On Lazy Training in Differentiable Programming"
Bietti and Mairal (2019) "On the Inductive Bias of Neural Tangent Kernels"

Maximal Update Parametrization (μP)

$$\begin{cases} Y_0(x) = W_0 x \\ Y_\ell(x) = W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L] \\ f(x) = W_{out} Y_L(x). \end{cases}$$

$$W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}\ .$$

Definition (Neural Parametrization)

A set of exponents α_ℓ and c_ℓ that define

- *Initialization:* $W_\ell \sim \mathcal{N}(0, n^{-\alpha_\ell})$
- *Learning Rate:* $W_\ell \leftarrow W_\ell - \eta n^{-c_\ell} G_\ell$

Yang and Hu (2022). "Feature Learning in Infinite-Width Neural Networks"

Maximal Update Parametrization (μP)

$$\begin{cases} Y_0(x) = W_0 x \\ Y_\ell(x) = W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L] \\ f(x) = W_{out} Y_L(x). \end{cases}$$

$$W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}\ .$$

Definition (Neural Parametrization)

A set of exponents α_ℓ and c_ℓ that define

- *Initialization:* $W_\ell \sim \mathcal{N}(0, n^{-\alpha_\ell})$
- *Learning Rate:* $W_\ell \leftarrow W_\ell - \eta n^{-c_\ell} G_\ell$

1. Study the infinite-width ($n \to \infty$) limit for general NPs

Yang and Hu (2022). "Feature Learning in Infinite-Width Neural Networks"

Maximal Update Parametrization (μP)

$$\begin{cases} Y_0(x) = W_0 x \\ Y_\ell(x) = W_\ell \phi\big(Y_{\ell-1}(x)\big), & \ell \in [1:L] \\ f(x) = W_{out} Y_L(x). \end{cases}$$

$$W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, W_{out} \in \mathbb{R}^{k\times n}\ .$$

Definition (Neural Parametrization)

A set of exponents α_ℓ and c_ℓ that define

- *Initialization:* $W_\ell \sim \mathcal{N}(0, n^{-\alpha_\ell})$
- *Learning Rate:* $W_\ell \leftarrow W_\ell - \eta n^{-c_\ell} G_\ell$

1. Study the infinite-width ($n \to \infty$) limit for general NPs
2. Infer NPs that satisfies $\boxed{Y_l^t(x) - Y_l^0(x) = \Theta_n(1)}$ (Desideratum)

t fixed, $n \to \infty$

Yang and Hu (2022). "Feature Learning in Infinite-Width Neural Networks"

Maximal Update Parametrization (μP)

Standard Parametrization

	Input	Hidden	Output
Init Var	1	n^{-1}	n^{-1}
LR (SGD)	1	1	1
LR (Adam)	1	1	1

μP

	Input	Hidden	Output
Init Var	1	n^{-1}	n^{-2}
LR (SGD)	n	1	n^{-1}
LR (Adam)	1	n^{-1}	n^{-1}

Yang et al. (2022) "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer"

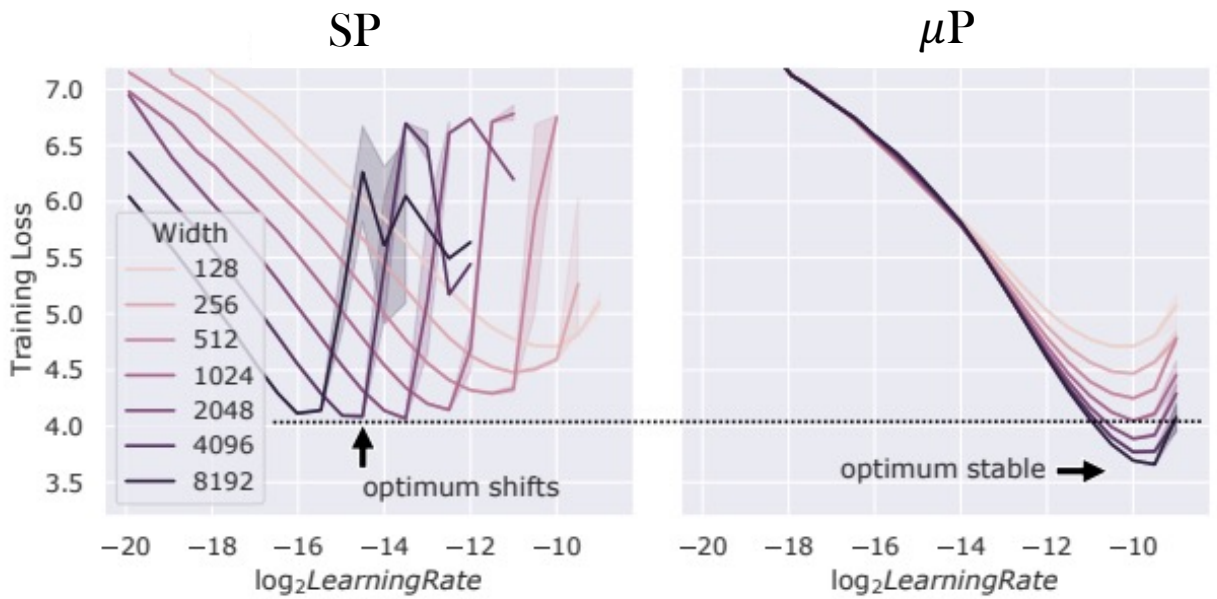
Maximal Update Parametrization (μP)

Standard Parametrization

	Input	Hidden	Output
Init Var	1	n^{-1}	n^{-1}
LR (SGD)	1	1	1
LR (Adam)	1	1	1

μP

	Input	Hidden	Output
Init Var	1	n^{-1}	n^{-2}
LR (SGD)	n	1	n^{-1}
LR (Adam)	1	n^{-1}	n^{-1}



Yang et al. (2022) "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer"

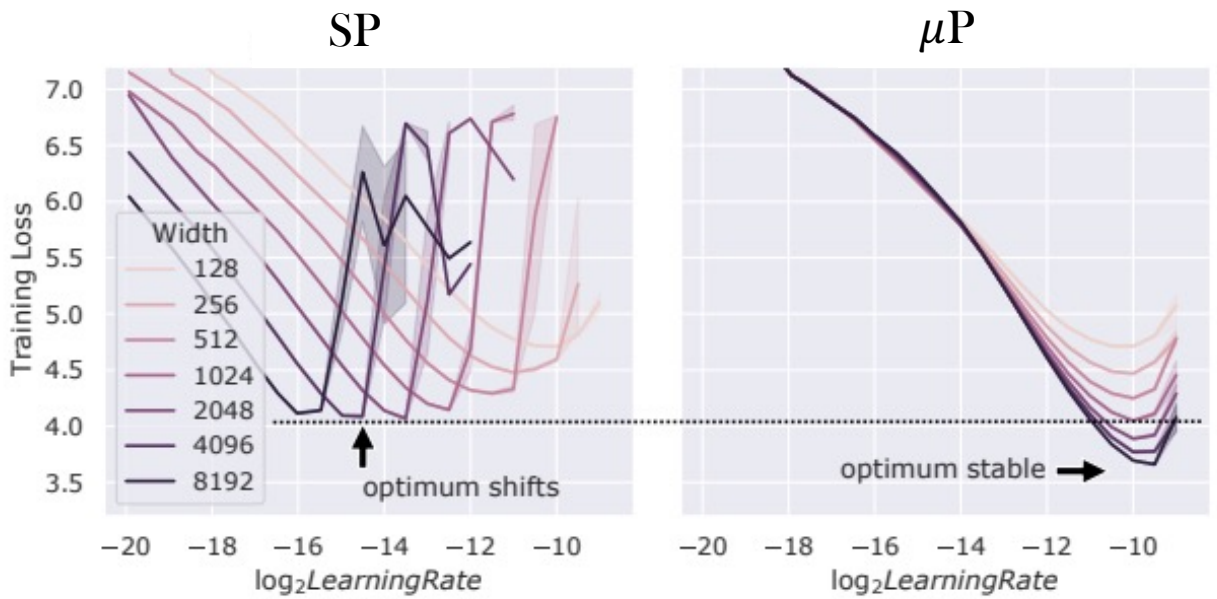
Maximal Update Parametrization (μP)

Standard Parametrization

	Input	Hidden	Output
Init Var	1	n^{-1}	n^{-1}
LR (SGD)	1	1	1
LR (Adam)	1	1	1

μP

	Input	Hidden	Output
Init Var	1	n^{-1}	n^{-2}
LR (SGD)	n	1	n^{-1}
LR (Adam)	1	n^{-1}	n^{-1}



Learning rate **transfers** across width!

Why?

Yang et al. (2022) "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer"

Learning Rate Transfer: The intuition

Fix t the number of training steps

View loss at step t as a function of $\eta > 0$

$$\mathcal{L}^{(t)}(\eta) = \frac{1}{m}\sum_{i=1}^{m} \ell oss(f_{\theta^{(t)}(\eta)}(x_i), z_i)$$

Intuition: Learning Rate Transfer (YH22)

Having $Y_l^t(x) - Y_l^0(x) = \Theta_n(1)$ makes the size of feature updates scale-invariant, so the optimal η (i.e. $argmin\ \mathcal{L}^{(t)}(\eta)$) should also be scale invariant...

Learning Rate Transfer: The intuition

Fix t the number of training steps

View loss at step t as a function of $\eta > 0$

$$\mathcal{L}^{(t)}(\eta) = \frac{1}{m}\sum_{i=1}^{m} \ell oss(f_{\theta^{(t)}(\eta)}(x_i), z_i)$$

Intuition: Learning Rate Transfer (YH22)

Having $Y_l^t(x) - Y_l^0(x) = \Theta_n(1)$ makes the size of feature updates scale-invariant, so the optimal η (i.e. $argmin\ \mathcal{L}^{(t)}(\eta)$) should also be scale invariant...

This work

- Formalizes Learning Rate Transfer
- Provides a proof for learning rate transfer in linear MLPs

Formalizing Learning Rate Transfer

t: number of training steps, n: network width, $\eta > 0$: learning rate

$$\mathcal{L}_n^{(t)}(\eta) = \frac{1}{m}\sum_{i=1}^{m} \ell oss(f_{\theta_n^{(t)}(\eta)}(x_i), z_i)$$

Optimal learning rate $\eta_n^{(t)} = \operatorname{argmin}_\eta \mathcal{L}^{(t)}(\eta)$

Formalizing Learning Rate Transfer

t: number of training steps, n: network width, $\eta > 0$: learning rate

$$\mathcal{L}_n^{(t)}(\eta) = \frac{1}{m}\sum_{i=1}^{m} \ell oss(f_{\theta_n^{(t)}(\eta)}(x_i), z_i)$$

Optimal learning rate $\eta_n^{(t)} = \text{argmin}_\eta \, \mathcal{L}^{(t)}(\eta)$

Definition (Learning Rate Transfer, Informal)

LR transfers with width n if the following holds:

For any $t > 0$, there exists $\boldsymbol{\eta_\infty^{(t)} > 0}$ such that $\boldsymbol{\eta_n^{(t)}}$ converges to $\boldsymbol{\eta_\infty^{(t)}}$ as $\boldsymbol{n \to \infty}$.

Linear MLP with μP

Model:

$$f(x) = V^\top W_L W_{L-1} \dots W_1 W_0 x$$

$$W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, V \in \mathbb{R}^n .$$
$$\theta = (W_\ell)_{1\le \ell \le L}$$

Only hidden layer weights W_ℓ are trainable!

μP for MLP trained with GD

Initialization: $W_0 \sim \mathcal{N}(0, d^{-1})$, $W_\ell \sim \mathcal{N}(0, n^{-1})$, $V \sim \mathcal{N}(0, n^{-2})$

Learning rate: $W_\ell \leftarrow W_\ell - \eta \nabla_{W_\ell} \mathcal{L}(\theta)$

Training data: $\mathcal{D} = \{(x_i, y_i),\ i = 1, \dots, m\}$

Linear MLP with μP

Model:

$$f(x) = V^\top W_L W_{L-1} \dots W_1 W_0 x$$

$$W_0 \in \mathbb{R}^{n\times d}, W_\ell \in \mathbb{R}^{n\times n}, V \in \mathbb{R}^n .$$
$$\theta = (W_\ell)_{1\le \ell \le L}$$

μP for MLP trained with GD

Initialization:	$W_0 \sim \mathcal{N}(0, d^{-1})$, $W_\ell \sim \mathcal{N}(0, n^{-1})$ $V \sim \mathcal{N}(0, n^{-2})$
Learning rate:	$W_\ell \leftarrow W_\ell - \eta \nabla_{W_\ell} \mathcal{L}(\theta)$

Training data: $\mathcal{D} = \{(x_i, y_i),\ i = 1, \dots, m\}$

Only hidden layer weights W_ℓ are trainable!

After 1 GD step:

$$f^{(1)}(x) = V^\top \prod_{\ell=1}^{L} \left(W_\ell^{(0)} - \eta \nabla_{W_\ell} \mathcal{L}\left(\theta^{(0)}\right) \right) W_0\, x$$
$$= f^{(0)}(x) + \sum_{\ell=1}^{L} \phi_\ell\, \eta^\ell$$

Learning Rate Transfer at $t = 1$

$$\boldsymbol{K} = \left(d^{-1}\, x_i^{\top} x_j\right)_{1 \le i,j \le m} \in \mathbb{R}^{m \times m} \qquad \boldsymbol{y} = (y_1, y_2, \ldots, y_m)^{\top} \in \mathbb{R}^m$$

$I \subset \mathbb{R}^+$: compact set

$$\eta_n^{(1)} = argmin_{\eta \in I}\, \mathcal{L}^{(t)}(\eta)$$

Learning Rate Transfer at $t = 1$

$$\boldsymbol{K} = \left(d^{-1}\, x_i^\top x_j\right)_{1\le i,j\le m} \in \mathbb{R}^{m\times m} \qquad \boldsymbol{y} = (y_1, y_2, \dots, y_m)^\top \in \mathbb{R}^m$$

$I \subset \mathbb{R}^+$: compact set

$$\eta_n^{(1)} = argmin_{\eta\in I}\, \mathcal{L}^{(t)}(\eta)$$

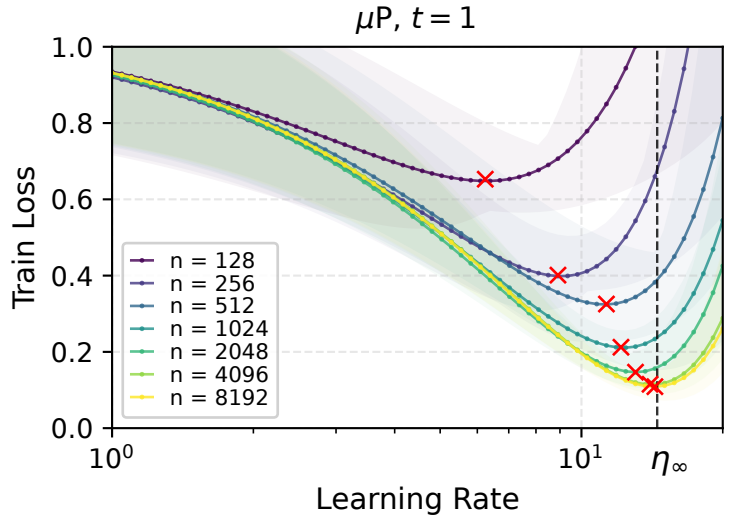
Theorem (H26, LR Transfer at $t = 1$)

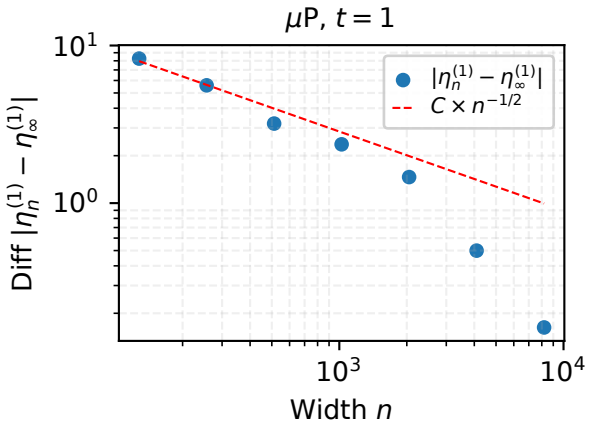
Assume that $\boldsymbol{K}\,\boldsymbol{y} \neq 0$. Then for any conpact set I

$$\eta_n^{(1)} - \eta_\infty^{(1)} = \mathcal{O}_{\mathbb{P}}(n^{-1/2})$$

where $\eta_\infty^{(1)} = \frac{m}{L}\frac{\boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{y}}{||\boldsymbol{K}\boldsymbol{y}||^2} > 0.$

Proof: Quadratic loss in the limit + Probabilistic bounds for $\partial_\eta^2 \mathcal{L}^{(t)}$





Standard Parametrization at $t = 1$

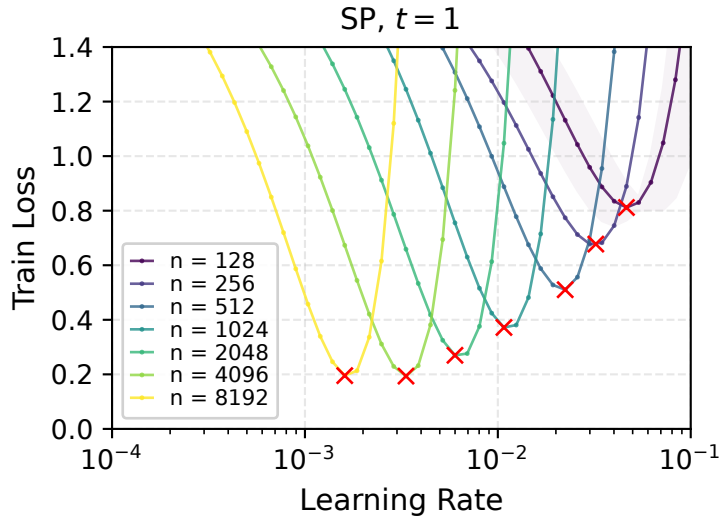
Initialization: $W_0 \sim \mathcal{N}(0, d^{-1})$,
$W_\ell \sim \mathcal{N}(0, n^{-1})$
$V \sim \mathcal{N}(0, n^{-1})$

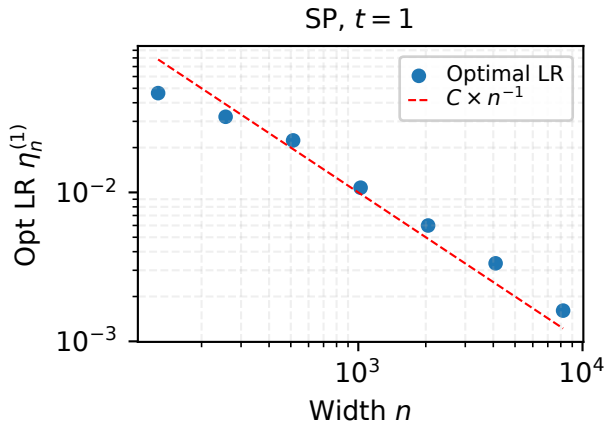
Learning rate: $W_\ell \leftarrow W_\ell - \eta \nabla_{W_\ell} \mathcal{L}(\theta)$

Theorem (H26, No LR Transfer with SP)

Assume that $K\,y \neq 0$. Then

$$\lim_{n\to\infty} \eta_n^{(1)} = 0 \ (in\ prob)$$





Learning Rate Transfer at any t

What about $t > 1$?

$$f^{(2)}(x) = V^\top \prod_{\ell=1}^{L} (W_\ell^{(1)} - \eta \nabla_{W_\ell} \mathcal{L}(\theta^{(1)}))\ x$$

Learning Rate Transfer at any t

What about $t > 1$?

$\theta^{(1)}$ depends on η!

$$f^{(2)}(x) = V^\top \prod_{\ell=1}^{L} (W_\ell^{(1)} - \eta \nabla_{W_\ell} \mathcal{L}(\theta^{(1)}))\ x$$

Each term $\nabla_{W_\ell} \mathcal{L}(\theta^{(1)})$ is a polynomial of degree $L - 1$ in η!

So $f^{(2)}(x)$ is a polynomial of degree $\propto L^2$ in η!

Learning Rate Transfer at any t

What about $t > 1$?

$\theta^{(1)}$ depends on η!

$$f^{(2)}(x) = V^\top \prod_{\ell=1}^{L} (W_\ell^{(1)} - \eta \nabla_{W_\ell} \mathcal{L}(\theta^{(1)}))\ x$$

Each term $\nabla_{W_\ell} \mathcal{L}(\theta^{(1)})$ is a polynomial of degree $L - 1$ in η!

So $f^{(2)}(x)$ is a polynomial of degree $\propto L^2$ in η!

Asymptotic behavior of the polynomial coefficients?

Non-trivial (non-linear limit in η!)

Proof machinery used for $t = 1$ completely fail for $t > 1$!

Learning Rate Transfer at any t

Number of steps: t,
Learning rate: $\eta > 0$

$$f^{(t)}(x) = V^\top \prod_{\ell=1}^{L} W_\ell^{(t)}\ x$$

$$= f^{(0)}(x) + \sum_{\ell=1}^{\mu(L,t)} \phi_\ell\ \eta^\ell$$

Lemma (H26, Deterministic Limits at $t > 0$)

For any ℓ, as model width $n \to \infty$, the coefficient ϕ_ℓ **converges** almost surely to a **deterministic** limit.

Proof machinery: Tensor Programs (Yang and Hu 2020)

Coefficient ϕ_ℓ can be expressed as a sequence of matrix-vector products involving the (random) matrices $W_0, W_1^{(0)}, W_2^{(0)}, \ldots, W_L^{(0)}, V$

Learning Rate Transfer at any t

Number of steps: t,
Learning rate: $\eta > 0$

$$\mathcal{L}_n^{(t)}(\eta) = \frac{1}{2m}\sum_{i=1}^{m}\left(f^{(t)}(x_i) - y_i\right)^2$$

$$\eta_n^{(t)} = \text{argmin}_{\eta\in I}\, \mathcal{L}^{(t)}(\eta)$$

Theorem (H26 , Learning Rate Transfer at any $t > 0$)

Assume that $\boldsymbol{Ky} \neq 0$. Then:

1. There exists a deterministic function $\mathcal{L}_\infty^{(t)}(.)$ s.t. for any compact set $I \subset \mathbb{R}^+$,

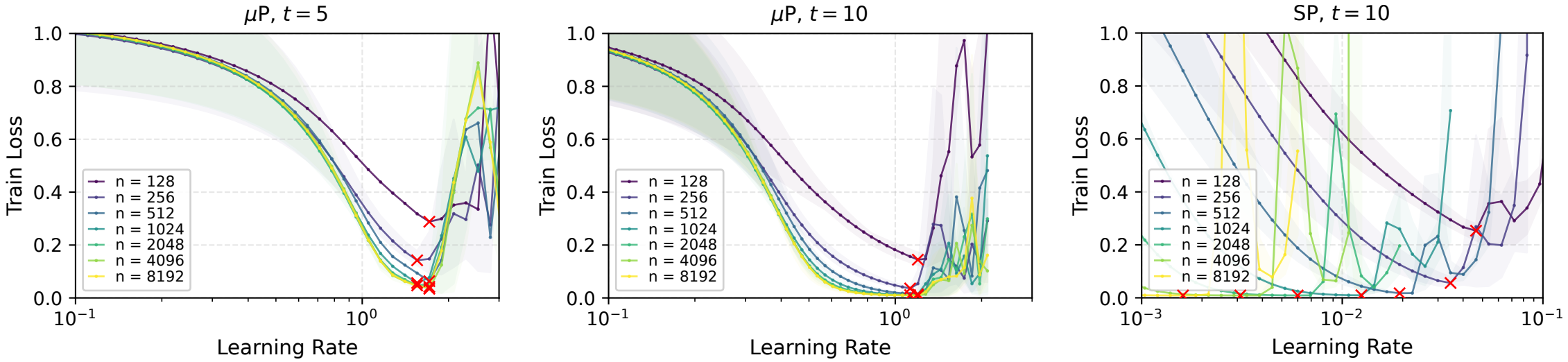
$$\text{Sup}_{\eta\in I}\left|\mathcal{L}_n^{(t)}(\eta) - \mathcal{L}_\infty^{(t)}(\eta)\right| \to 0 \quad a.s.$$

$$\text{argmin}_{\eta\in I}\, \mathcal{L}_\infty^{(t)}(\eta) \subset (0,\infty)$$

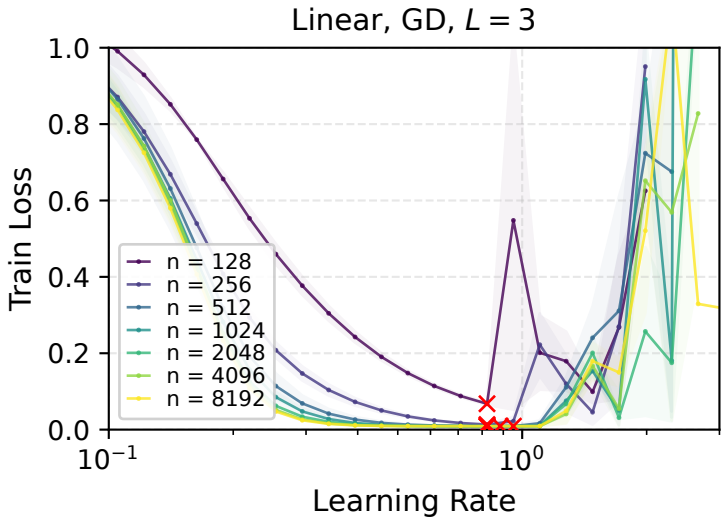
2. If $\mathcal{L}^{(\infty)}(\eta)$ has a unique minimizer $\eta_\infty^{(t)}$, then for any compact set $I \subset \mathbb{R}^+$ containing $\eta_\infty^{(t)}$

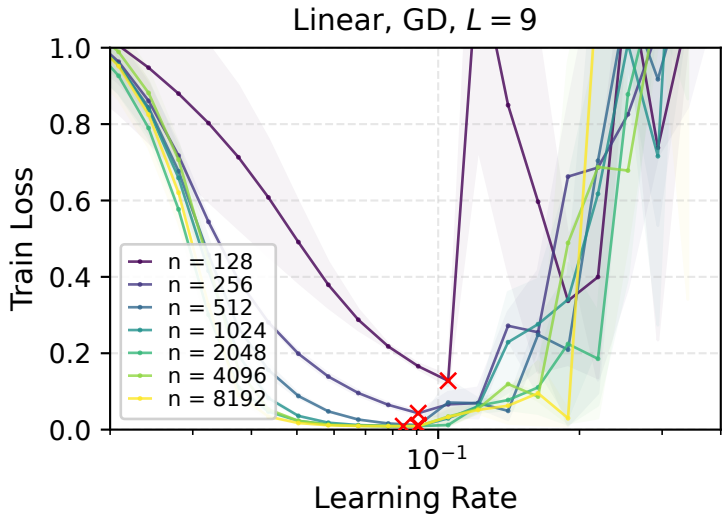
$$\boldsymbol{\eta_n^{(t)} \to \eta_\infty^{(t)}, \quad a.s.}$$

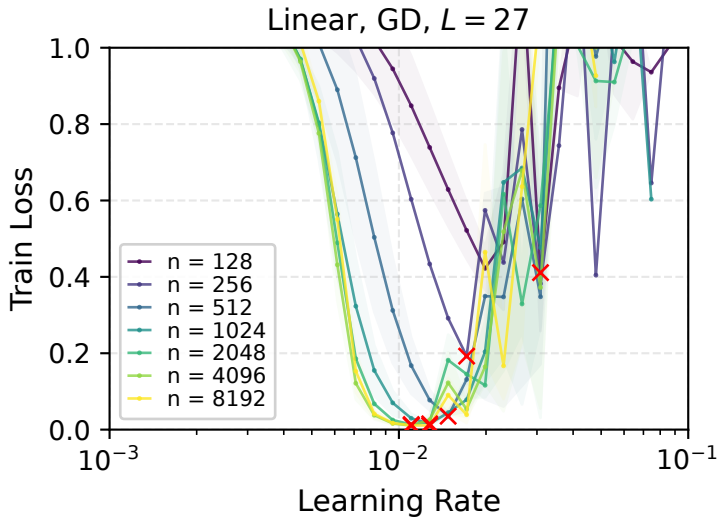
Empirical Results (L=3)

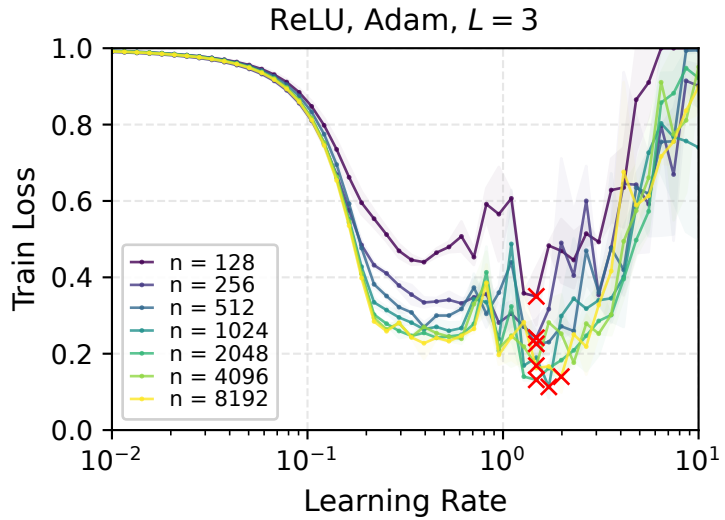


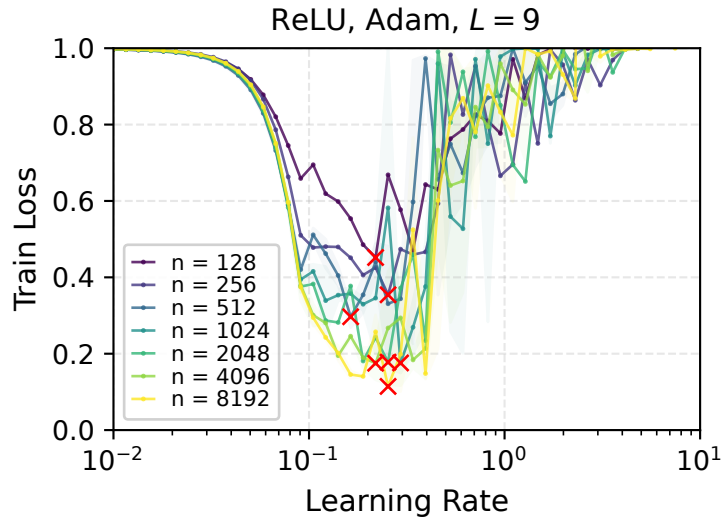
Empirical Results (t=20)

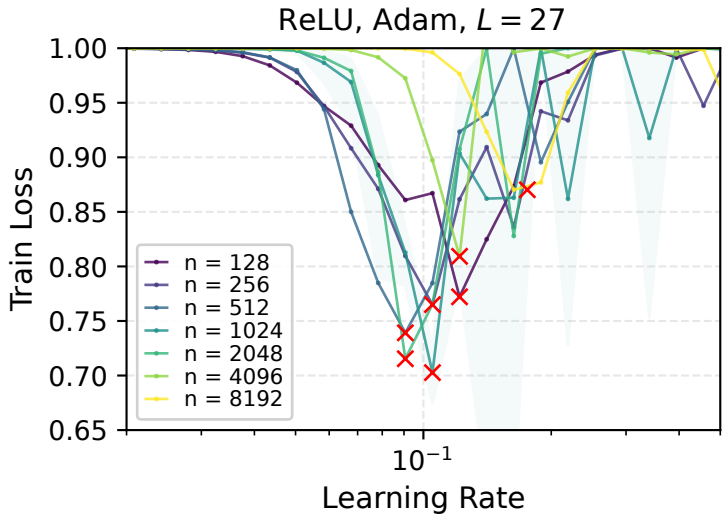




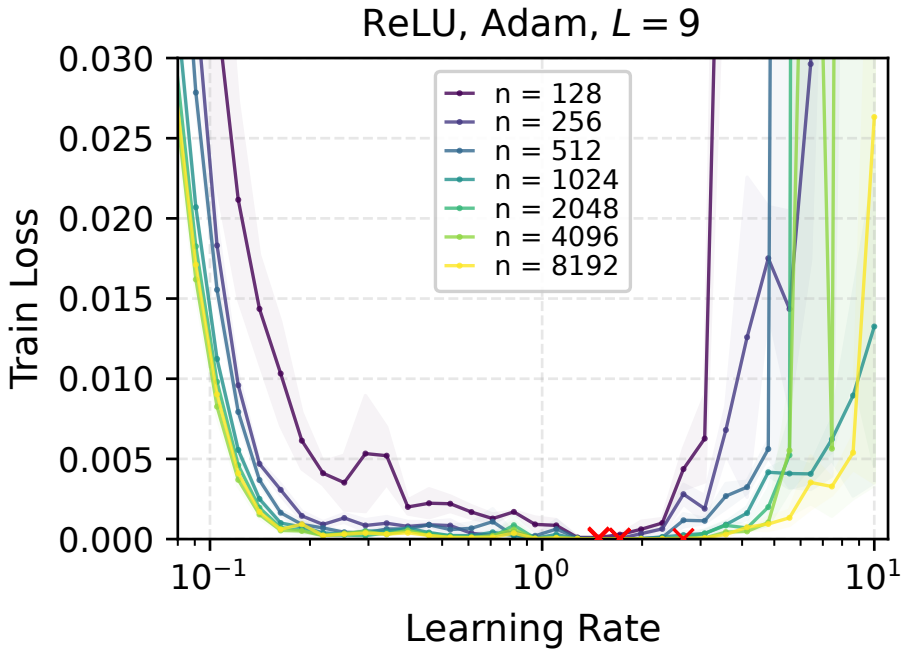








Empirical Results (t=100)



Limitations and Open Questions

- Linear MLP / Gradient Descent / Learning Rate

Limitations and Open Questions

- Linear MLP / Gradient Descent / Learning Rate
- Fixed number of steps t

Limitations and Open Questions

- Linear MLP / Gradient Descent / Learning Rate
- Fixed number of steps t
- Fast convergence rates (useful LR transfer)?

Ghosh et al. (2026), “Understanding the Mechanisms of Fast Hyperparameter Transfer”

Limitations and Open Questions

- Linear MLP / Gradient Descent / Learning Rate
- Fixed number of steps t
- Fast convergence rates (useful LR transfer)?

  Ghosh et al. (2026), "Understanding the Mechanisms of Fast Hyperparameter Transfer"

- Other scalable dimensions