Low-Rank Training and Low-Rank Adapters

Timon Klein and Piotr Minakowski
Otto von Guericke University Magdeburg

March 10th, 2026
MLaftermath

Low-Rank Training

A geometric framework for momentum-based optimizers for low-rank training

Steffen Schotthöfer* **Timon Klein†** **and** **Jonas Kusch‡**

*Computer Science and Mathematics Division; Oak Ridge National Laboratory; Oak Ridge, TN 37831 USA; Mail correspondence: schotthofers@ornl.gov

†Department of Mathematics; Otto von Guericke University Magdeburg; 39106 Magdeburg; Germany

‡Scientific Computing; Norwegian University of Life Sciences; Drøbakveien 31, 1433 Ås; Norway

Authors

Steffen Schotthöfer

Householder Fellow in the Mathematics in Computation Section at Oak Ridge National Laboratory

schotthofers@ornl.gov

Timon Klein

PhD Student in Mathematical Optimization at Otto von Guericke University Magdeburg

timon.klein@ovgu.de

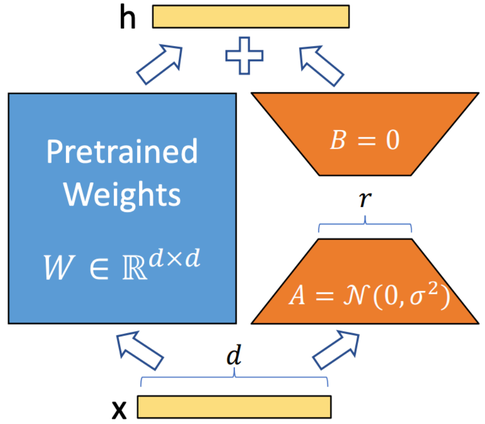
Jonas Kusch

Associate Professor in Scientific Computing at Norwegian University of Life Sciences

jonas.kusch@nmbu.no

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

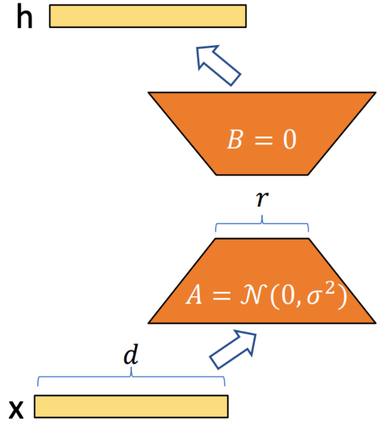
Low Rank Compression and Fine-Tuning of Neural Networks



Low Rank Finetuning

- Hu et al., *LoRA: Low-Rank Adaptation of Large Language Models. 2021*
- Zhang et al., *AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning.* 2023
- Liu et al., *DoRA: Weight-Decomposed Low-Rank Adaptation. 2024*

$$z(x) = \sigma\left(Wx + AB^\top x\right)$$



Low Rank Compression

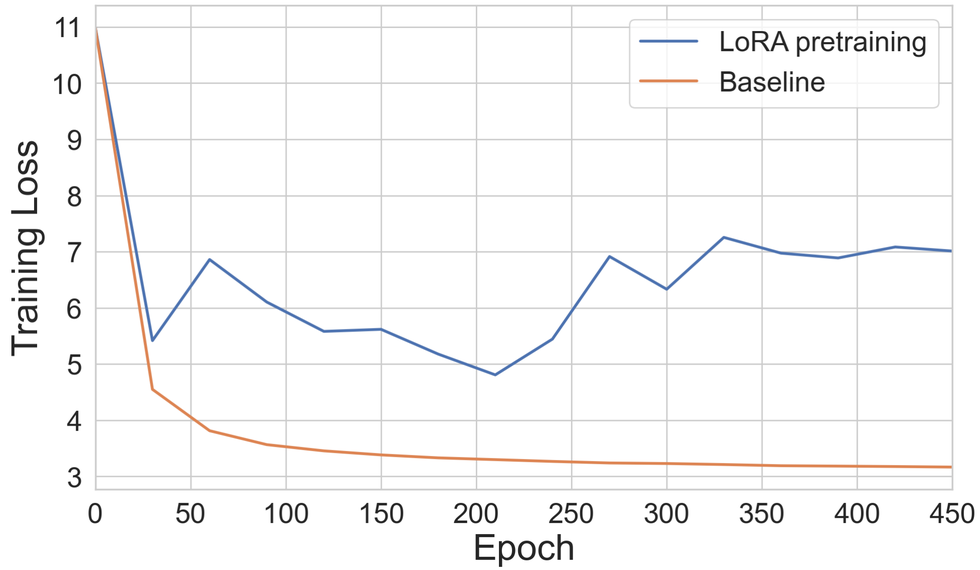
- Denton et al., *Exploiting Linear Structure Within Convolutional Networks.* 2014
- A. Novikov, et al., *Tensorizing neural networks.* 2015
- A. Tjandra, S. Sakti, and S. Nakamura. *Compressing recurrent neural network with tensor train.* 2017

$$z(x) = \sigma\left(AB^\top x\right)$$

Low Rank Attention

- DeepSeek-AI, "*Multihhead Latent Attention*". 2024
- Ainslie, et al., *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. '23*

$$z(x) = \sigma\left(xAB^\top x^\top\right)$$



GPT-2 on OWT, low-rank MLP: Low rank training stalls. Why?

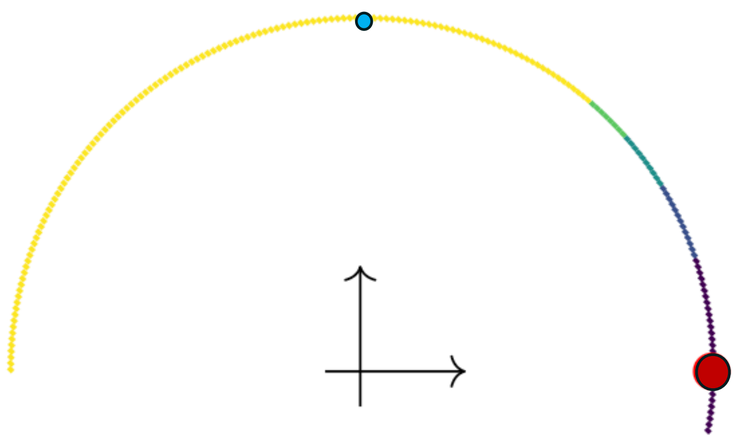
LoRA

Memory savings

vs

Training stability

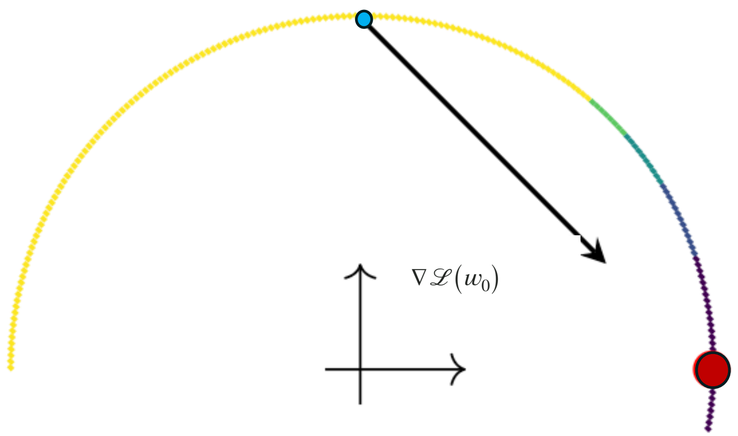
Thought Experiment: Manifold Constrained Optimization



$$\min_{w \in \mathcal{M}} \mathcal{L}(w) = \frac{1}{2} \left\| [1,0] - w \right\|_2^2$$

- Manifold: Unit circle $\mathcal{M} = \left\{ w \in \mathbb{R}^2 : \|w\| = 1 \right\}$
- Initialization at $w_0 = [0,1]$ solution at $w_* = [1,0]$

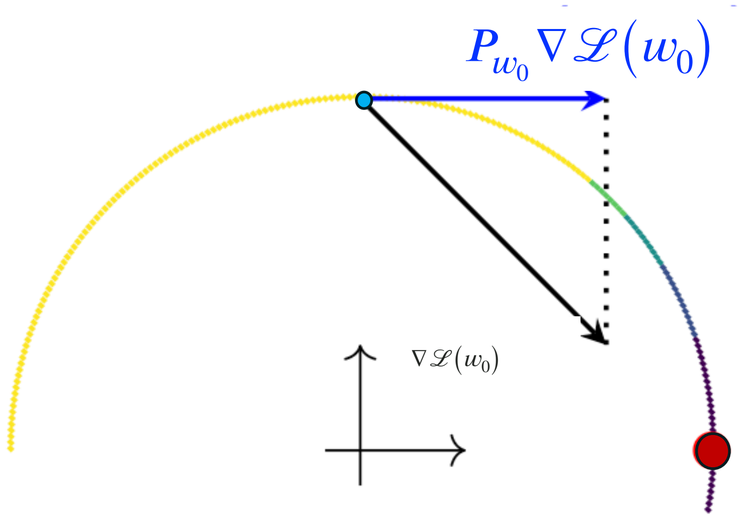
Thought experiment: Manifold Constrained Optimization



$$\min_{w \in \mathscr{M}} \mathscr{L}(w) = \frac{1}{2} \left\| [1,0] - w \right\|_2^2$$

- Manifold: Unit circle $\mathscr{M} = \left\{ w \in \mathbb{R}^2 : \|w\| = 1 \right\}$
- Initialization at $w_0 = [0,1]$ solution at $w_* = [1,0]$
- Gradient $\nabla \mathscr{L}(w_0) = [-1,1]$

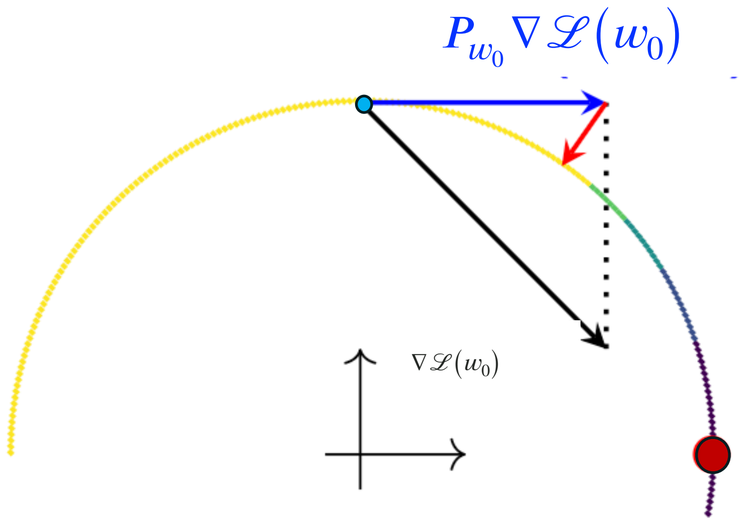
Thought experiment: Manifold Constrained Optimization



$$\min_{w \in \mathscr{M}} \mathscr{L}(w) = \frac{1}{2} \left\| [1,0] - w \right\|_2^2$$

- Manifold: Unit circle $\mathscr{M} = \{ w \in \mathbb{R}^2 : \|w\| = 1 \}$
- Initialization at $w_0 = [0,1]$ solution at $w_* = [1,0]$
- Gradient $\nabla \mathscr{L}(w_0) = [-1,1]$
- Riemannian gradient $P_{w_0} \nabla \mathscr{L}(w_0) = [0,1]$
  ➔ orthogonal projection P_{w_0}

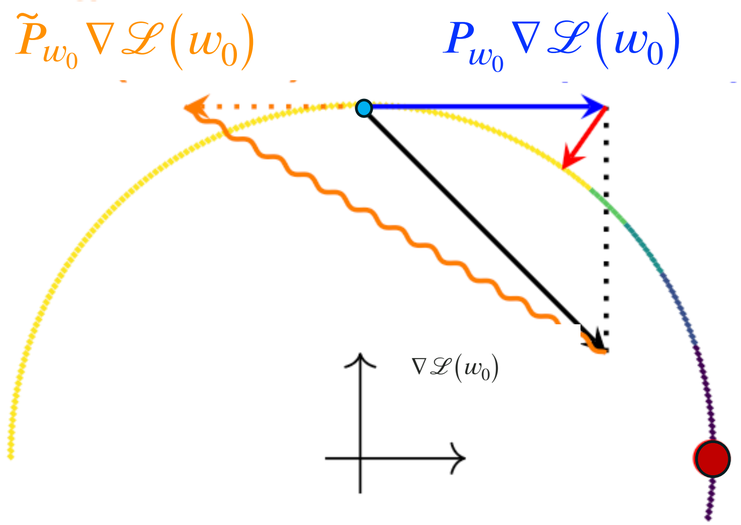
Thought experiment: Manifold Constrained Optimization



$$\min_{w \in \mathscr{M}} \mathscr{L}(w) = \frac{1}{2} \left\| [1,0] - w \right\|_2^2$$

- Manifold: Unit circle $\mathscr{M} = \{ w \in \mathbb{R}^2 : \|w\| = 1 \}$
- Initialization at $w_0 = [0,1]$ solution at $w_* = [1,0]$
- Gradient $\nabla \mathscr{L}(w_0) = [-1,1]$
- Riemannian gradient $P_{w_0} \nabla \mathscr{L}(w_0) = [0,1]$
  ➔ orthogonal projection P_{w_0}
- Retraction onto unit circle $\mathscr{M}$

Thought experiment: Manifold Constrained Optimization



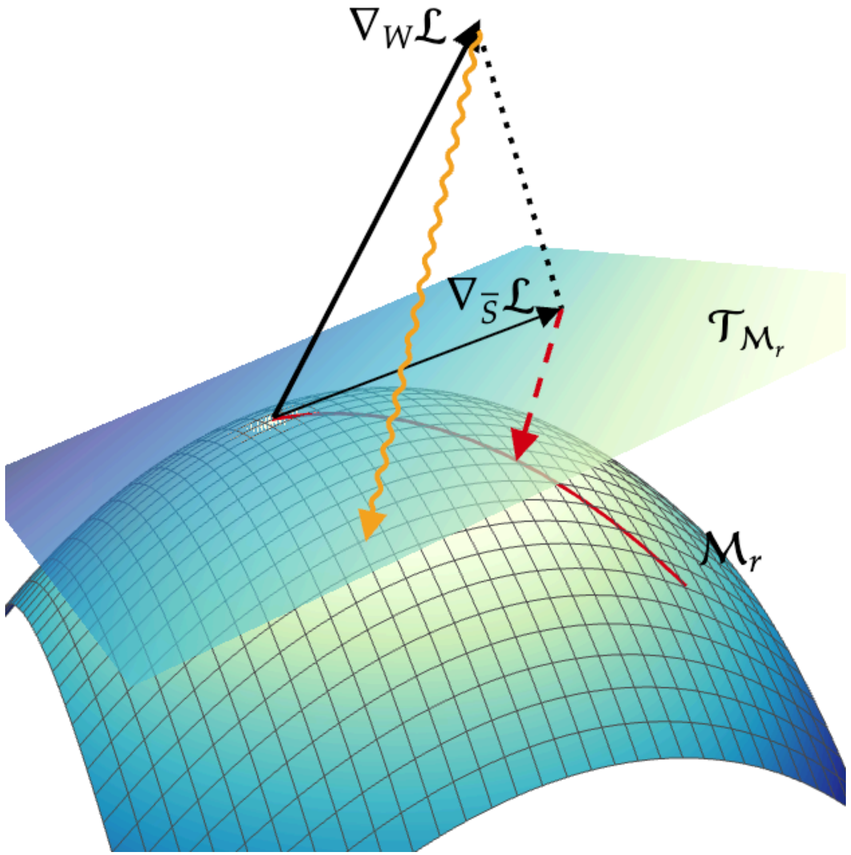
$$\min_{w \in \mathscr{M}} \mathscr{L}(w) = \frac{1}{2} \left\| [1,0] - w \right\|_2^2$$

- Manifold: Unit circle $\mathscr{M} = \{ w \in \mathbb{R}^2 : \|w\| = 1 \}$
- Initialization at $w_0 = [0,1]$ solution at $w_* = [1,0]$
- Gradient $\nabla \mathscr{L}(w_0) = [-1,1]$
- Riemannian gradient $P_{w_0} \nabla \mathscr{L}(w_0) = [0,1]$
  ➔ orthogonal projection P_{w_0}
- Retraction onto unit circle $\mathscr{M}$
- Non orthogonal $\tilde{P}_{w_0}$ breaks the structure

$\dot{w} = -\nabla \mathscr{L}$ vs $\dot{w} = -P_w \nabla \mathscr{L}$ vs $\dot{w} = -\tilde{P}_w \nabla \mathscr{L}$

gradient flow | Riemannian gradient flow

Low Rank training is manifold constrained optimization



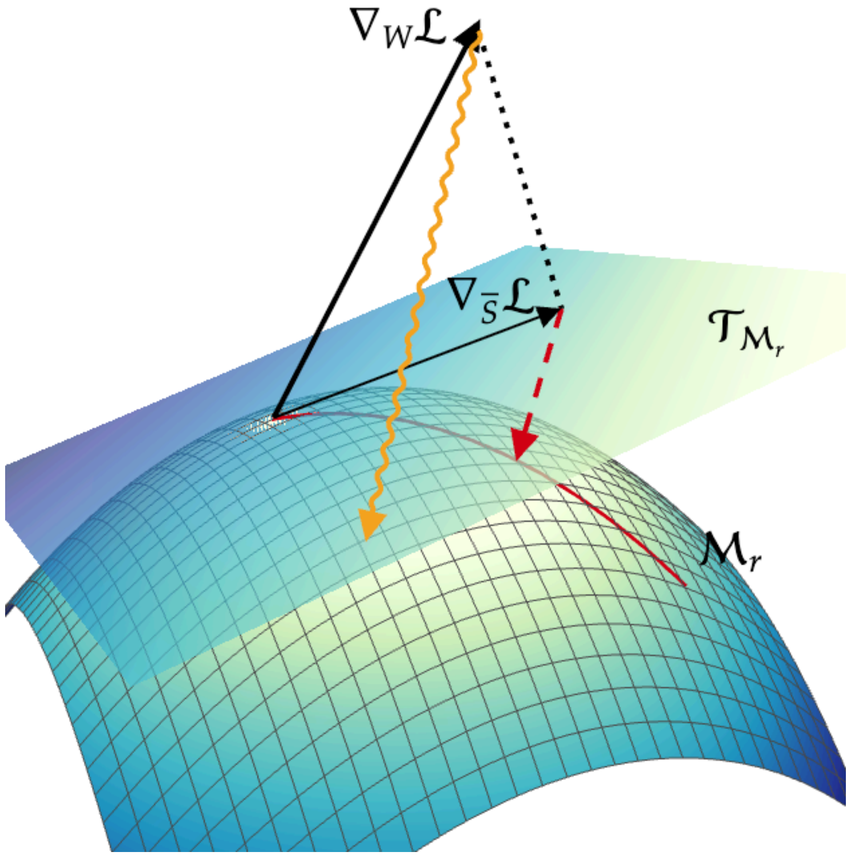
$$\min_{W \in \mathcal{M}} \mathcal{L}(X, Y; W)$$

- Manifold $\mathcal{M} = \left\{ W \in \mathbb{R}^{n \times n} : \operatorname{rank}(W) = r \right\}$
- LoRA ansatz: $W = AB^\top$ with $A, B \in \mathbb{R}^{n \times r}$
- Gradient flow: $\dot{W} = \dot{A}B^\top + A\dot{B}^\top = \tilde{P}_W \nabla_W \mathcal{L}$ (chain rule)
- $\tilde{P}_W$ is defined by $\left[A, \dot{A}\right]$, and $\left[B, \dot{B}\right]$

➔ not orthogonal

➔ no steepest descent on $\mathcal{M}$

Low Rank training is manifold constrained optimization



$$\min_{W \in \mathcal{M}} \mathcal{L}(X, Y; W)$$

- Manifold $\mathcal{M} = \left\{ W \in \mathbb{R}^{n \times n} : \operatorname{rank}(W) = r \right\}$
- AdaLoRA ansatz: $W = USV^\top$ with $U, V \in \mathbb{R}^{n \times r}, S \in \mathbb{R}^{r \times r}$
- Gradient flow: $\dot{W} = \dot{U}SV^\top + U\dot{S}V^\top + USV^\top = \tilde{P}_W \nabla_W \mathcal{L}$ (chain rule)
- $\tilde{P}_W$ is defined by $[U, \dot{U}]$, and $[V, \dot{V}]$

  ➔ not orthogonal

  ➔ no steepest descent on $\mathcal{M}$

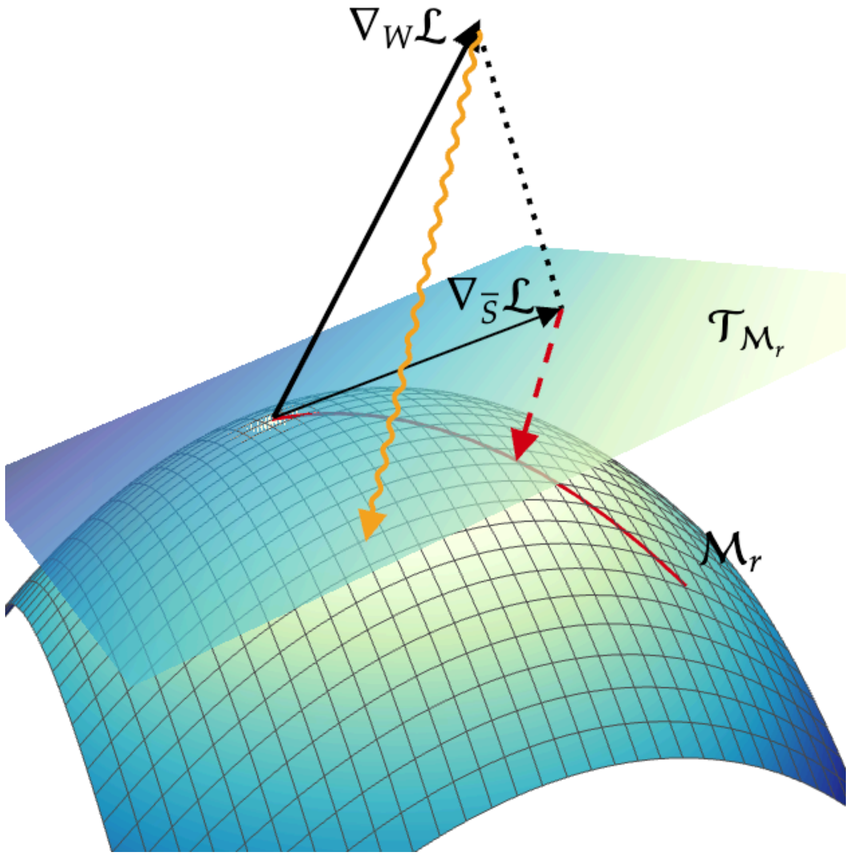
DLRT: Dynamical Low Rank Training

Efficient evolution of projected gradient flow

S.S., Zangrando, Kusch, Ceruti, Tudisco; *Low-Rank Lottery Tickets ...* ; NeurIPS 2022

$$\dot{W} = P_W \nabla \mathcal{L}$$

$$\min_{W \in \mathcal{M}} \mathcal{L}(X, Y; W)$$



- Manifold $\mathcal{M} = \left\{ W \in \mathbb{R}^{n \times n} : \operatorname{rank}(W) = r \right\}$
- DLRT ansatz: $W = USV^\top$ with $U, V \in \mathbb{R}^{n \times r}$, $S \in \mathbb{R}^{r \times r}$
- Gradient flow: $\dot{W} = \dot{U}SV^\top + U\dot{S}V^\top + US\dot{V}^\top = P_W \nabla_W \mathcal{L}$
- Construct P_W with orthogonal bases

  $\bar{U} = \operatorname{ortho}\left\{ [U, \dot{U}] \right\}$, and $\bar{V} = \operatorname{ortho}\left\{ [V, \dot{V}] \right\}$

  ➔ Basis for tangent space $\mathcal{T}_{\mathcal{M}}$

  ➔ enables steepest descent on $\mathcal{M}$

Construct $\bar{U}$

Construct $\bar{V}$

Optimize $\bar{S}$

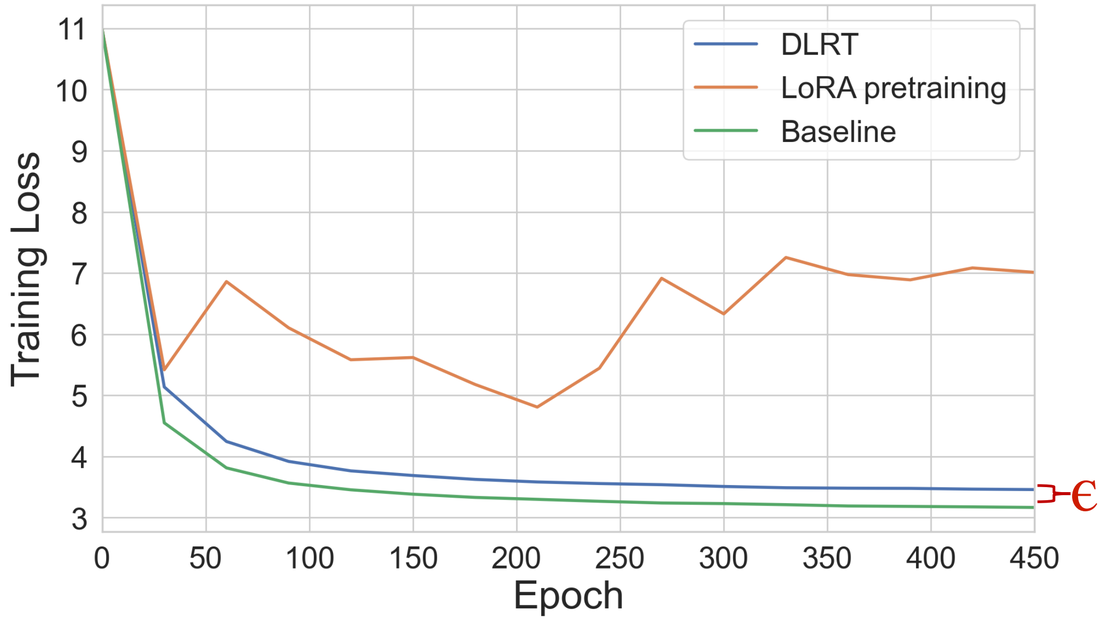
Retract onto $\mathcal{M}$

Memory cost – slightly better than LoRA

- Weights: $\mathcal{O}(2nr + r^2)$
- Gradients: $\mathcal{O}(2nr)$ for basis update $\mathcal{O}(r^2)$ for optimization
- Optimizer states: $\mathcal{O}(r^2)$

DLRT: Dynamical Low Rank Training

Schotthöfer., Zangrando, Kusch, Ceruti, Tudisco; *Low-Rank Lottery Tickets ...* ; NeurIPS 2022
Zangrando , S.S., Ceruti, Kusch, Tudisco; *Geometry-aware training […] in tensor Tucker format* ; NeurIPS ’24
S.S., Zangrando Ceruti, Kusch, Tudisco; *GeoLoRA […]*; ICLR ’25

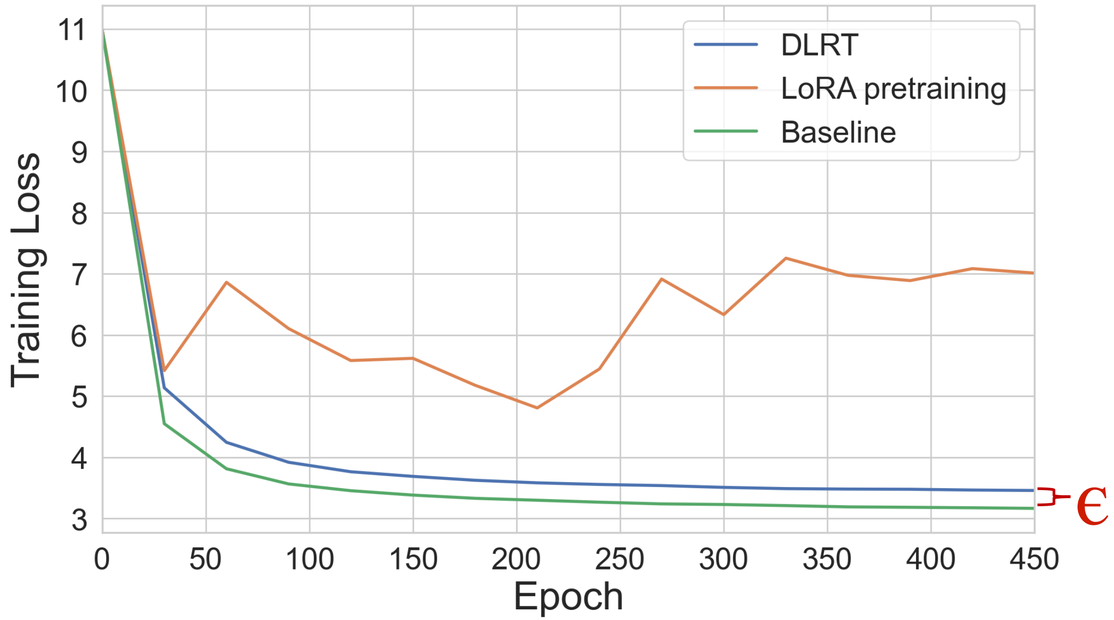


GPT-2 on OWT, low-rank MLP: DLRT beats LoRA

- DLRT inherits training robustness from full training
  - Provable: Optimality, loss descent, convergence
  - Hyperparameter can be transferred from full training
- Provable error bound to full rank training
  $\left\| W_{\text{full rank}}(t) - W_{\text{DLRT}}(t) \right\| < \epsilon(\lambda, \vartheta)$
- Automatic rank selection (like AdaLoRA)

DLRT: Dynamical Low Rank Training

Schotthöfer, Zangrando, Kusch, Ceruti, Tudisco; *Low-Rank Lottery Tickets ...* ; NeurIPS 2022
Zangrando , Schotthöfer, Ceruti, Kusch, Tudisco; *Geometry-aware training […] in tensor Tucker format* ; NeurIPS ’24
Schotthöfer, Zangrando Ceruti,Tudisco, Kusch; *GeoLoRA […]*; ICLR ’25



GPT-2 on OWT, low-rank MLP: DLRT beats LoRA

- DLRT inherits training robustness from full training
  - Provable: Optimality, loss descent, convergence
  - Hyperparameter can be transferred from full training
- Provable error bound to full rank training

$$\left\| W_{\text{full rank}}(t) - W_{\text{DLRT}}(t) \right\| < \epsilon(\lambda, \vartheta)$$

- Automatic rank selection (like AdaLoRA)

What about momentum methods/Adam?

Schotthöfer, Klein, Kusch; *A geometric framework for momentum-based optimizers for low-rank training*; NeurIPS ‘25

Momentum gradient flow

$$\dot{W} = \mathscr{V}$$
$$\dot{\mathscr{V}} + \gamma \mathscr{V} = -\nabla_W \mathscr{L}$$

DLRT Momentum gradient flow

$$\dot{W} = P_W \mathscr{V}$$
$$\dot{\mathscr{V}} + \gamma \mathscr{V} = -P_W \nabla_W \mathscr{L}$$

➔ Bases U, V for W can be re-used for momentum terms
- Extendable for Adam, AdamW

Low-Rank Adapters

Mitigating Subject Dependency in EEG Decoding with Subject-Specific Low-Rank Adapters

Timon Klein [1] **Piotr Minakowski** [1] **Sebastian Sager** [1 2] **Steffen Schotthöfer** [3]

[1]Department of Mathematics; Otto von Guericke University Magdeburg; 39106 Magdeburg; Germany. [2]Max Planck Institute for Dynamics of Complex Technical Systems; 39106 Magdeburg; Germany. [3]Computer Science and Mathematics Division; Oak Ridge National Laboratory; Oak Ridge, TN 37831; USA. Correspondence to: Timon Klein <timon.klein@ovgu.de>.

Authors

Timon Klein

PhD Student in Mathematical Optimization at Otto von Guericke University Magdeburg

timon.klein@ovgu.de

Piotr Minakowski

AI Expert for Energy Load Forecasting and Balancing

pminak@gmail.com

Sebastian Sager

Professor in Mathematical Optimization at Otto von Guericke University Magdeburg

Max Planck Fellow at MPI Magdeburg

sager@ovgu.de

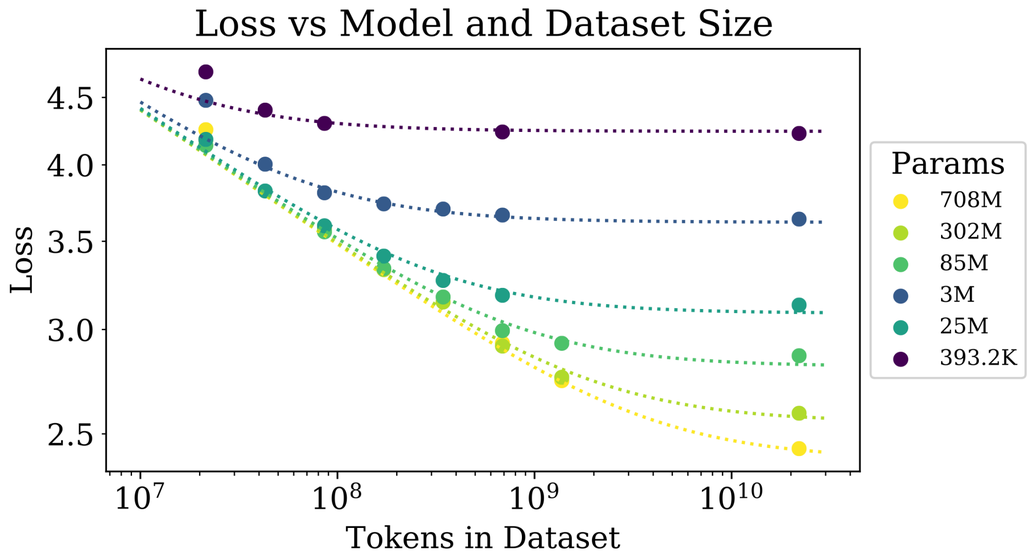
Steffen Schotthöfer

Householder Fellow in the Mathematics in Computation Section at Oak Ridge National Laboratory

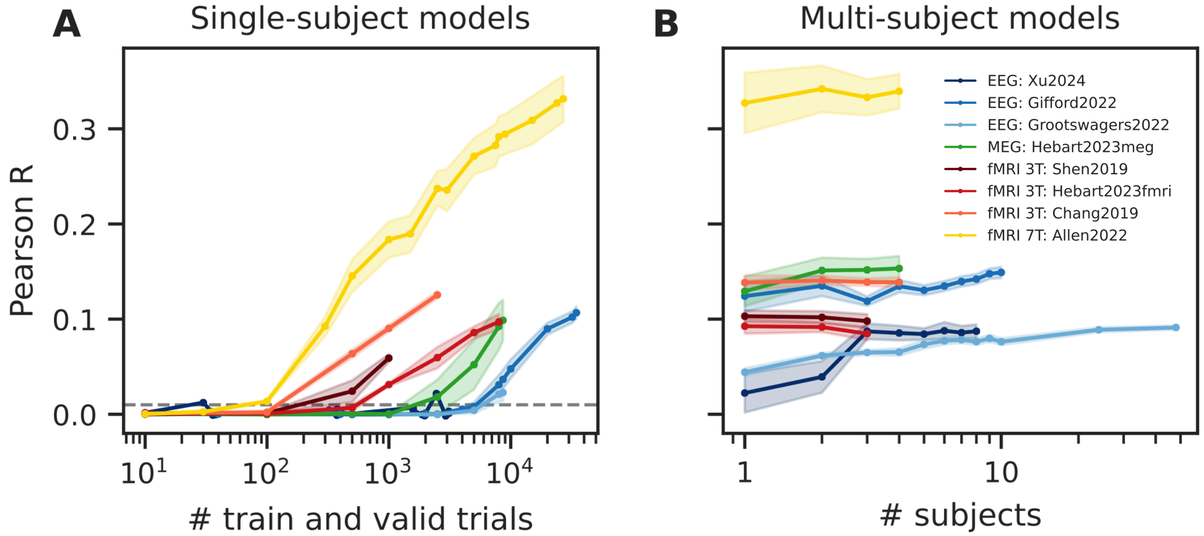
schotthofers@ornl.gov

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Motivation: Scaling Laws

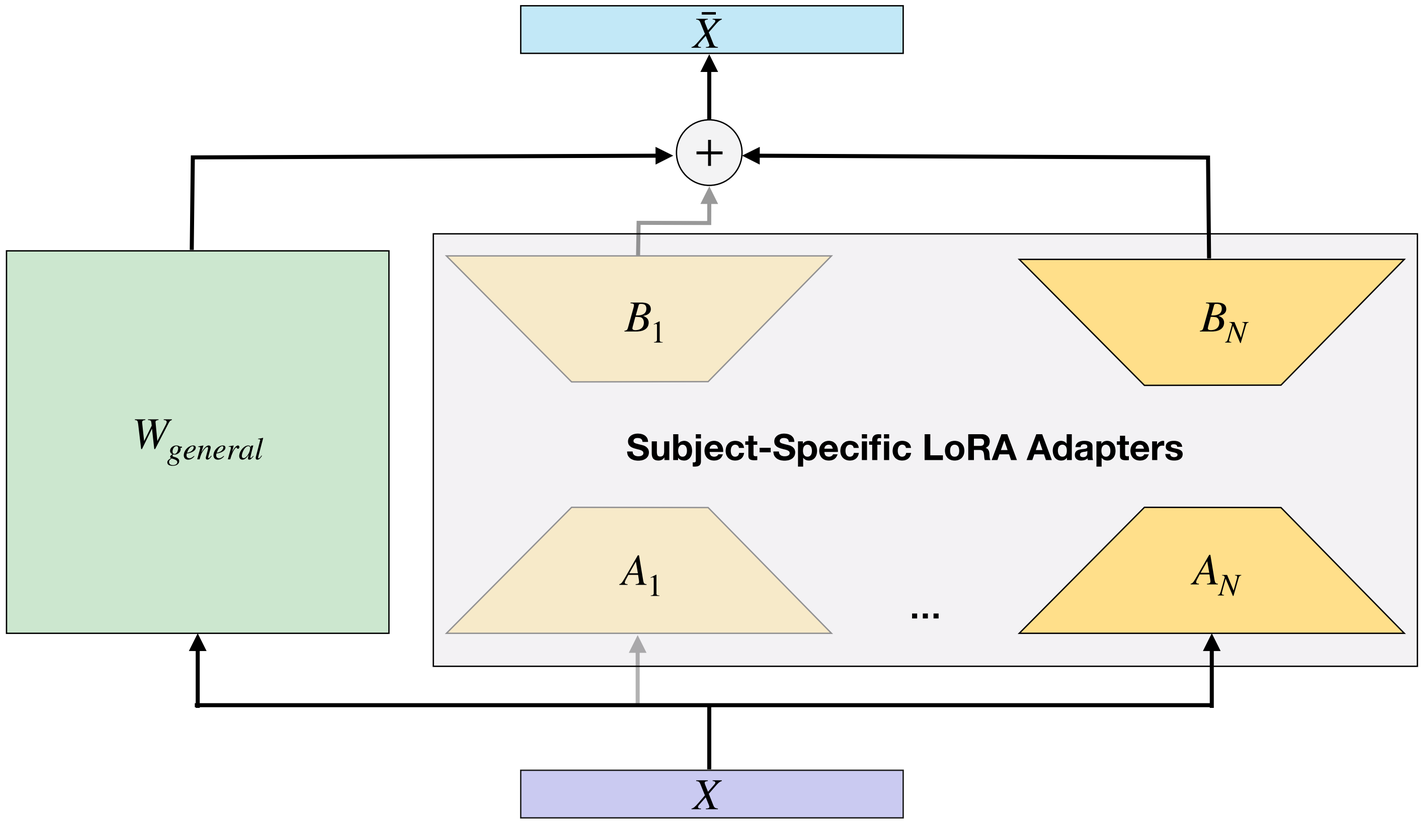


Scaling Laws for Neural Language Models
J. Kaplan et al.



Scaling laws for decoding images from brain activity
H. Banville et al.

Subject-Conditioned Layer



$$\bar{X} = \sigma\left(XW_{general}^{\top} + \sum_{s=1}^{N}(M_s \cdot X)A_sB_s\right)$$

$W_{general}$ shared Weights

A_sB_s subject-specific Weights

$A_s \in \mathbb{R}^{n\times r}$, $B_s \in \mathbb{R}^{r\times m}$

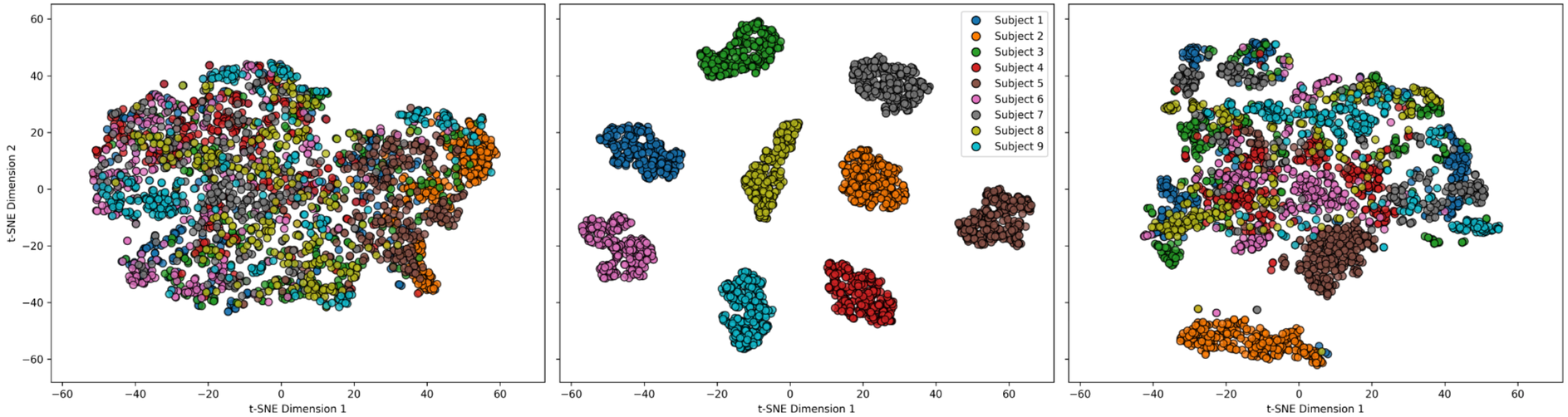
M_s := binary Mask

Subject $i \in \{1,\ldots,N\}$

Rank of Adapter: r

Initialization: $A_s = \mathcal{N}(0,\sigma^2)$, $B_s = 0$

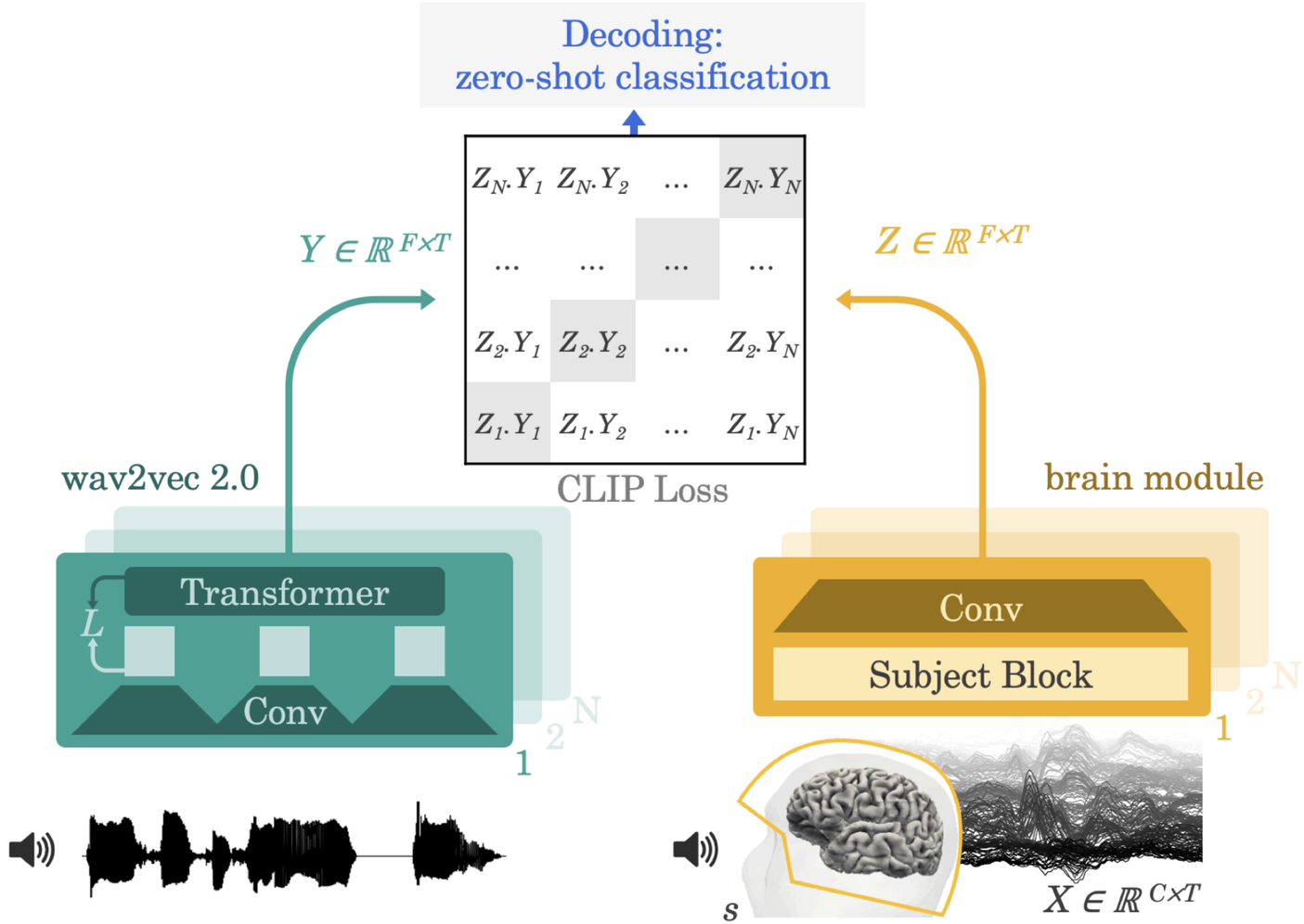
Numerical Results



Method	Total # Params	Active # Params	BCI Comp. IV2a (4 Classes)	BCI Comp. IV2b (2 Classes)
EEGNeX Subject-Agnostic [1]	55 972	55 972	55.00% ± 0.66%	74.28% ± 0.39%
EEGNeX Subject-Specific	55 972	55 972	59.36% ± 14.42%	75.92% ± 14.32%
EEGNex Subject-Specific LoRA	81 124	10 980	53.60% ± 2.91%	69.64% ± 0.27%
EEGNeX Subject-Conditioned Layer	134 884	64 740	**64.72% ± 0.56%**	**76.48% ± 0.25%**
PBT Subject-Agnostic	867 460	867 460	50.82% ± 0.85%	75.49% ± 1.10%
PBT Subject-Specific	867 460	867 460	44.16% ± 12.19%	75.63% ± 14.23%
PBT Subject-Specific LoRA	680 736	135 712	25.81% ± 0.52%	53.70% ± 0.80%
PBT Subject-Conditioned Layer	1 480 612	935 588	**54.51% ± 0.30%**	**76.36% ± 0.23%**

[1] Toward reliable signals decoding for electroencephalogram:
A benchmark study to EEGNeX
X. Chen et al.

BrainMagick



Decoding speech perception from non-invasive brain recordings
Defossez, A. et al.

BrainMagick

	Method	subject-specific	Accuracy [%] ± Std. [%]		
		#Params	Top 1	Top 5	Top 10
SuLoRA	r=4, α=4	131k	37.56 ± 0.67	58.72 ± 0.52	67.06 ± 0.54
	r=4, α=8	131k	38.85 ± 0.33	60.17 ± 0.45	68.52 ± 0.39
	r=4, α=12	131k	39.04 ± 0.20	60.36 ± 0.21	68.77 ± 0.36
	r=8, α=8	190k	39.82 ± 0.72	61.03 ± 0.68	69.39 ± 0.62
	r=8, α=16	190k	39.61 ± 0.48	61.02 ± 0.27	69.40 ± 0.37
	r=8, α=24	190k	40.32 ± 0.50	61.5 ± 0.23	69.79 ± 0.29
	r=16, α=16	306k	40.50 ± 0.27	61.77 ± 0.19	69.97 ± 0.22
	r=16, α=32	306k	40.50 ± 0.27	61.86 ± 0.32	70.09 ± 0.29
	r=16, α=48	306k	40.68 ± 0.51	61.94 ± 0.39	70.20 ± 0.36
	r=32, α=32	540k	40.95± 0.54	62.22 ± 0.64	70.34 ± 0.47
	r=32, α=64	540k	41.65 ± 0.76	62.63 ± 0.84	70.76 ± 0.76
	r=32, α=96	540k	41.44 ± 0.32	62.67 ± 0.31	70.60 ± 0.18
	r=64, α=64	1,006k	40.97 ± 0.44	62.15 ± 0.49	70.30 ± 0.49
	r=64, α=128	1,006k	41.74 ± 0.49	62.90 ± 0.43	70.98 ± 0.45
	r=64, α=192	1,006k	**41.94 ± 0.92**	**63.20 ± 0.65**	**71.18 ± 0.61**
	r=64, α=256	1,006k	41.81 ± 0.21	63.00 ± 0.11	70.95 ± 0.21
	r=96, α=192	1,473k	41.62 ± 0.13	62.73 ± 0.17	70.85 ± 0.16
	Baseline	1,968k	40.60 ± 0.18	61.84 ± 0.21	70.00 ± 0.21
	BL w/o Sub.-Layer	73k	19.23 ± 0.41	36.69 ± 0.54	45.48 ± 0.69

Institute of Mathematical Optimization

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG