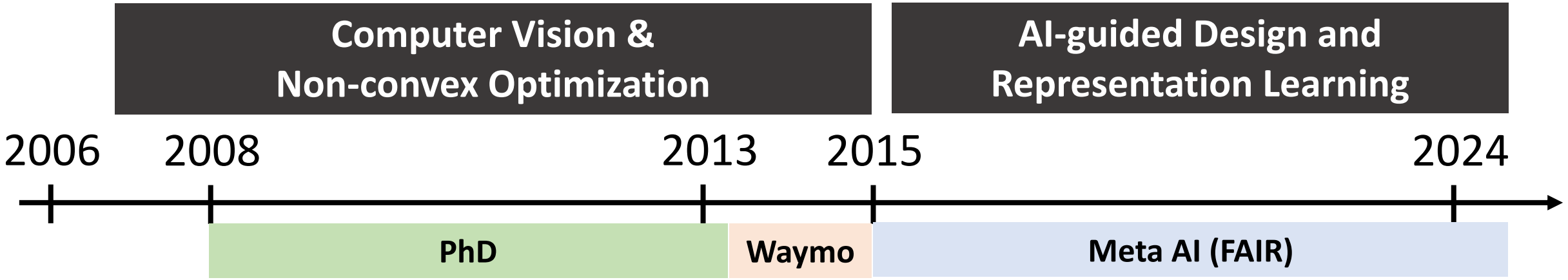
Towards Explainable, Efficient and Effective AI-guided Design

Yuandong Tian
Research Scientist

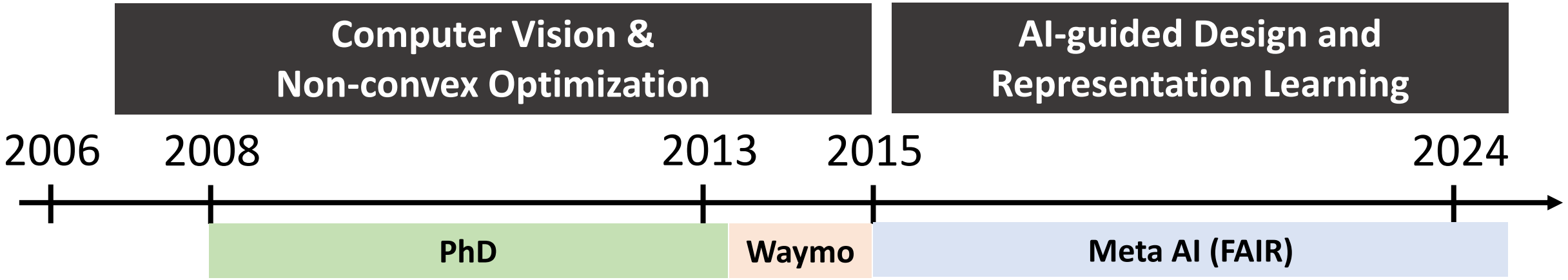
Meta AI (FAIR)

facebook Artificial Intelligence

Career Path


Computer Vision &
Non-convex Optimization
AI-guided Design and
Representation Learning
2006
2008
2013
2015
2024
PhD
Waymo
Meta AI (FAIR)

Career Path



Novelist in my spare time

AI-guided Design in Games (2015-2020)

Games

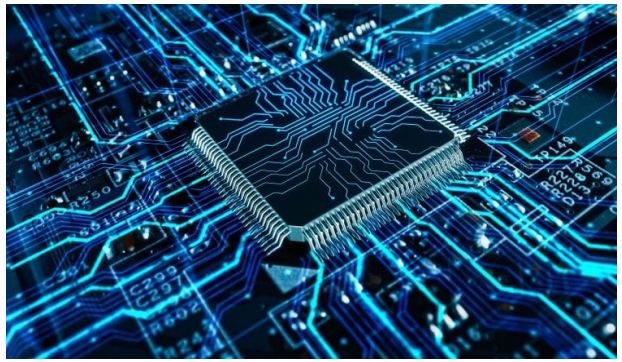
[DarkForest, Y. Tian et al, ICLR'16]
[F1, Y. Wu and Y. Tian, ICLR'17]
[ELF, Y. Tian et al, NeurIPS'17]
[OpenGo, Y. Tian et al, ICML'19]
[JPS, Y. Tian et al, NeurIPS'20]
…

Agent

[MiniRTSv2, H. Hu et al, NeurIPS'19]
[M^{3}RL, T. Shu and Y. Tian, ICLR'19]
[CollaQ, T. Zhang et al]
[NovelD, T. Zhang et al, NeurIPS'21]
…

AI-guided Design in Real-world Scenarios (2021-)

Games



Industrial-scale Engineering

[NeuroPlan, H. Zhu et al, SIGCOMM'21]
[AutoCAT, M. Luo et al, HPCA'23]
[MACTA, J. Cui et al, ICLR'23]
[SurCo, A. Ferber, ICML'23]
[LANCER, A. Zharmagambetov, NeurIPS'23]
[NeuroShard, D. Zha et al, MLSys'23]
[CZP, A. Cohen, AI4Science Caltech workshop'23]
…

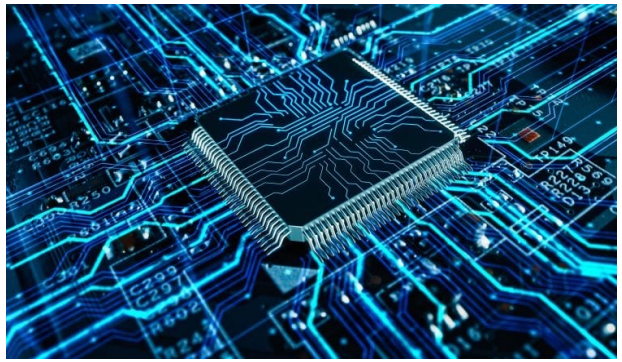
Agent

Creativity

[Re3, K. Yang et al, EMNLP'23]
[DOC, K. Yang et al, ACL'23]
[E2EStoryGenerator, H. Zhu, arXiv'23]
[PerSE, D. Wang, arXiv'23]
…

AI-guided Design in Real-world Scenarios

Games



Industrial-scale Engineering

Agent

Creativity

My Principle for AI-guided Design

Open the black-box of networks

Small memory, less compute

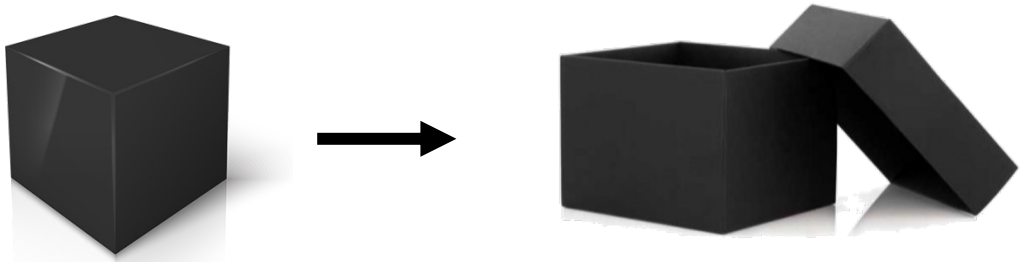
Strong performance in real-world cases

Foundational Understanding of Deep Models

Explainable Models

Open the black-box of networks

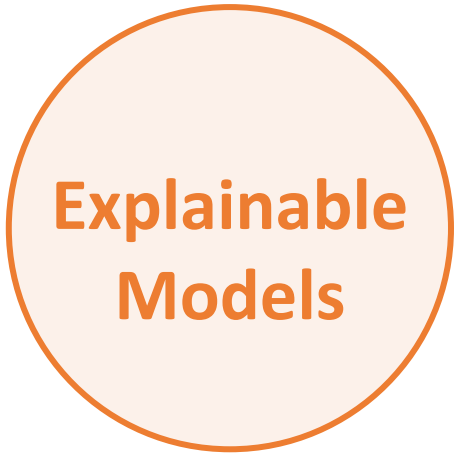
Understand the models!



Blackbox

Open the Blackbox

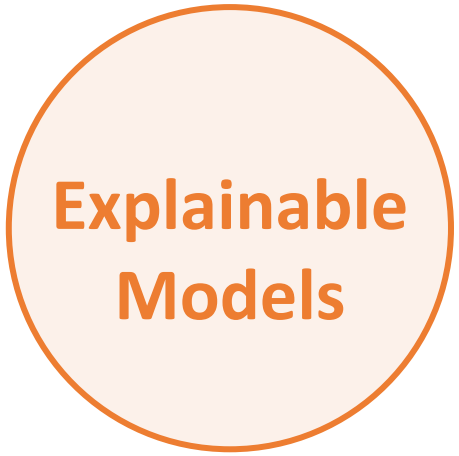
… Leads to More efficient training/inference



Open the black-box of networks

Small memory, less compute

… Leads to Breakthrough Design Applications



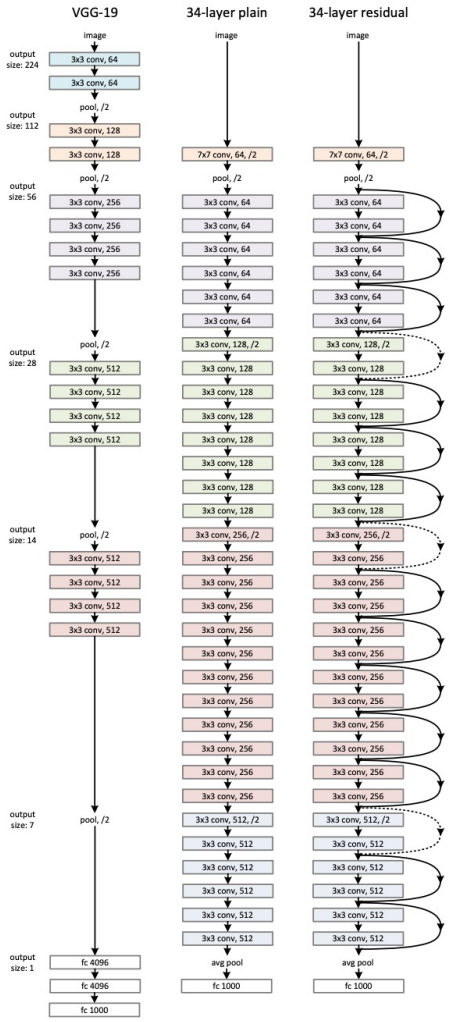
Open the black-box of networks

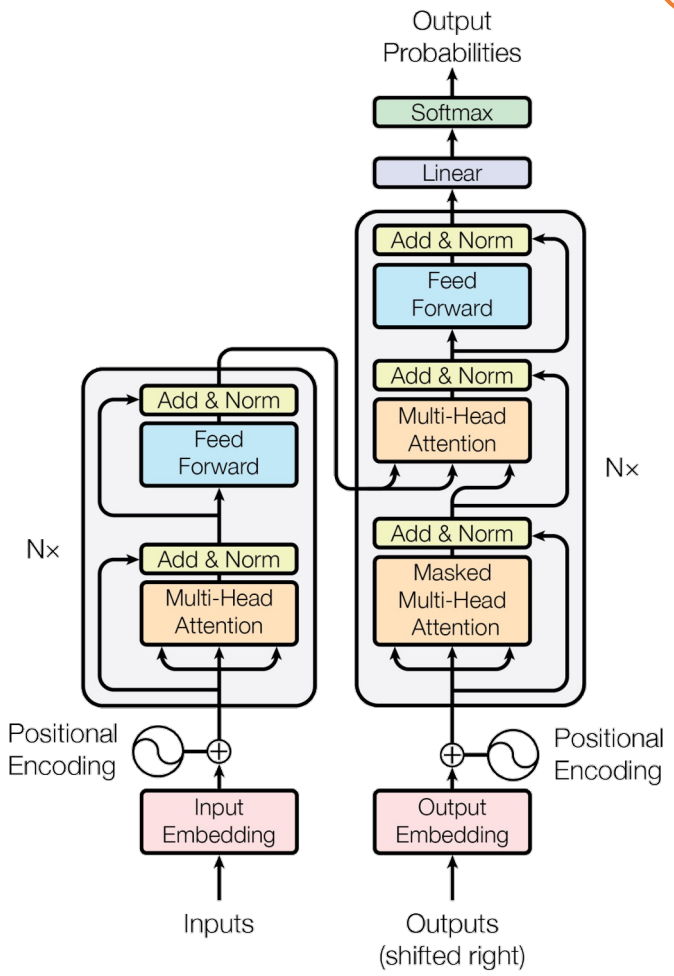
Small memory, less compute

Strong performance in real-world cases

Explainable Models

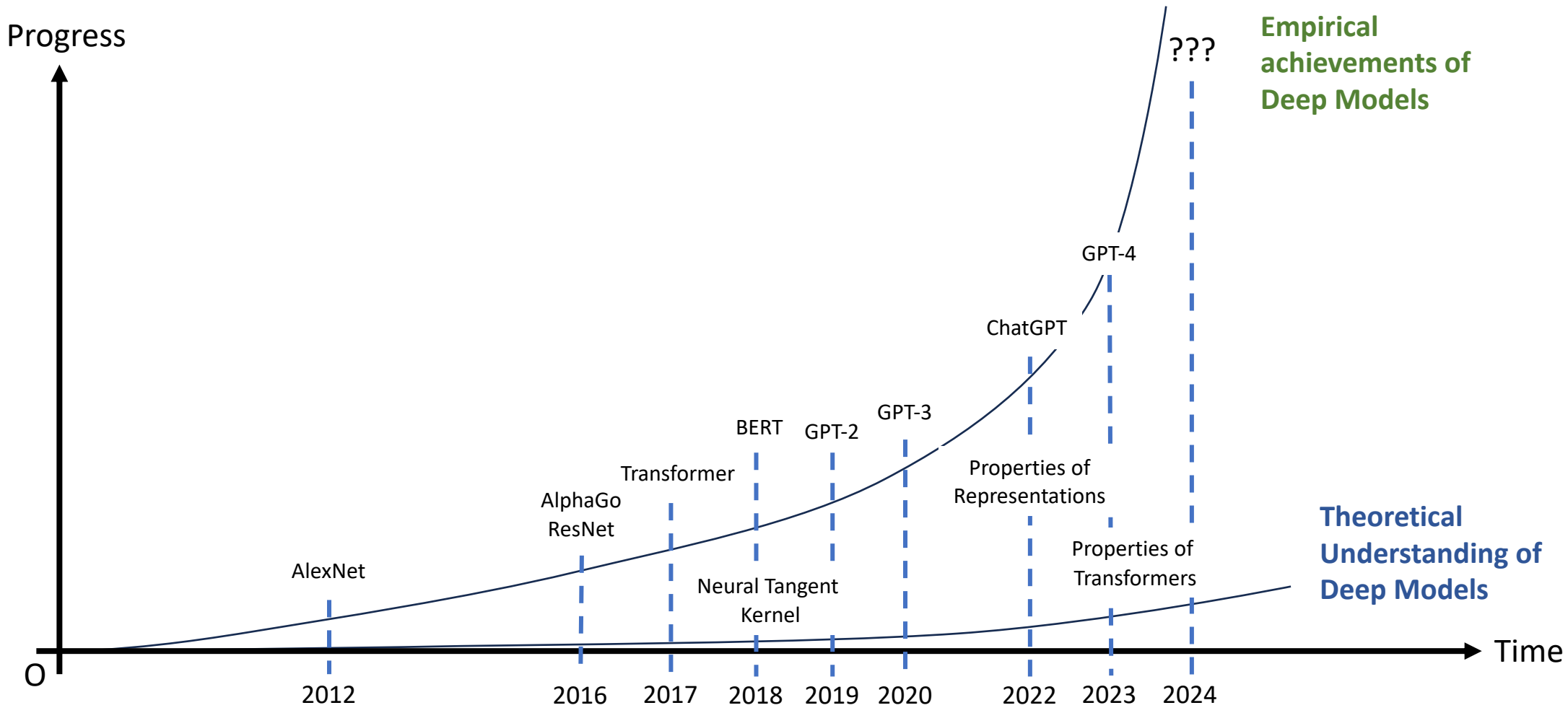
Open the black-box of networks



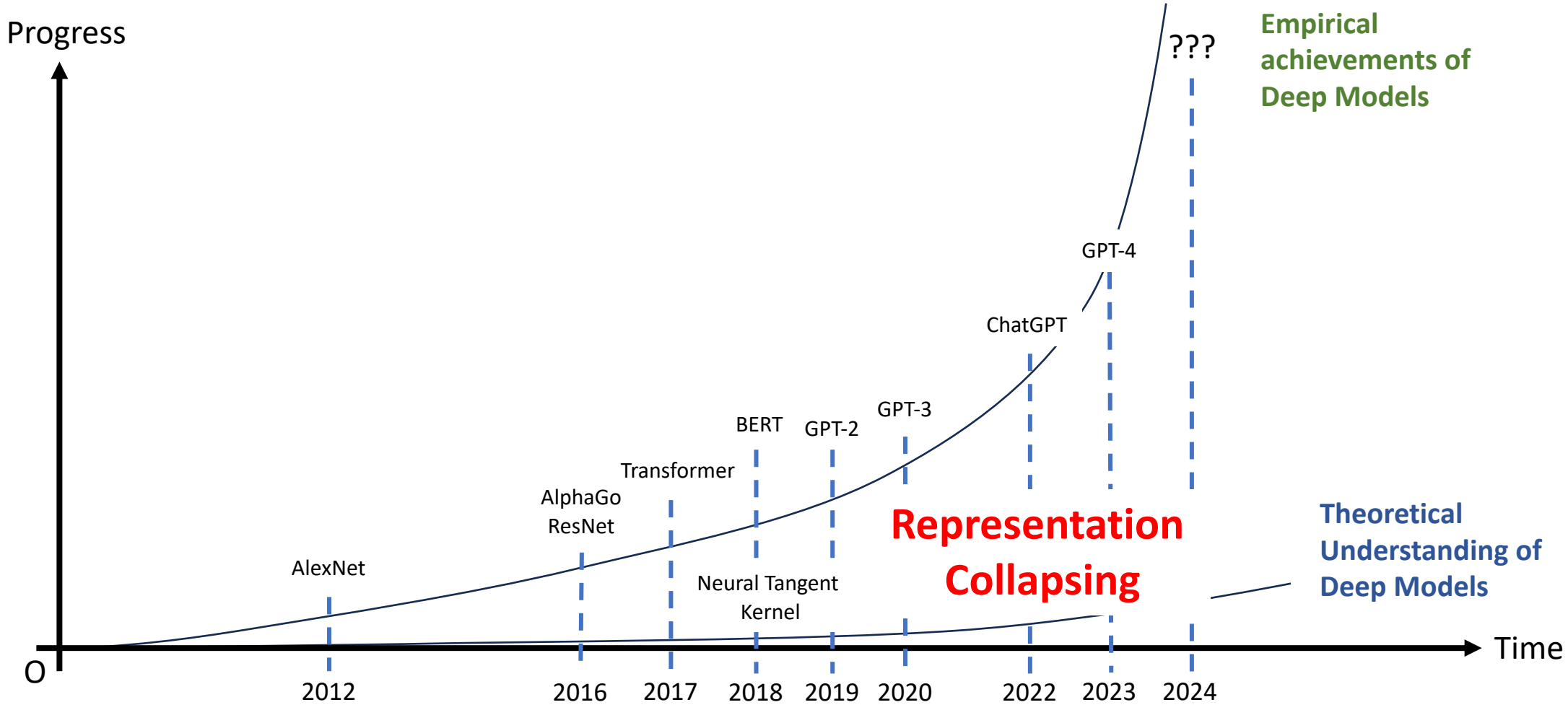


Part I: Understand How Neural Networks work

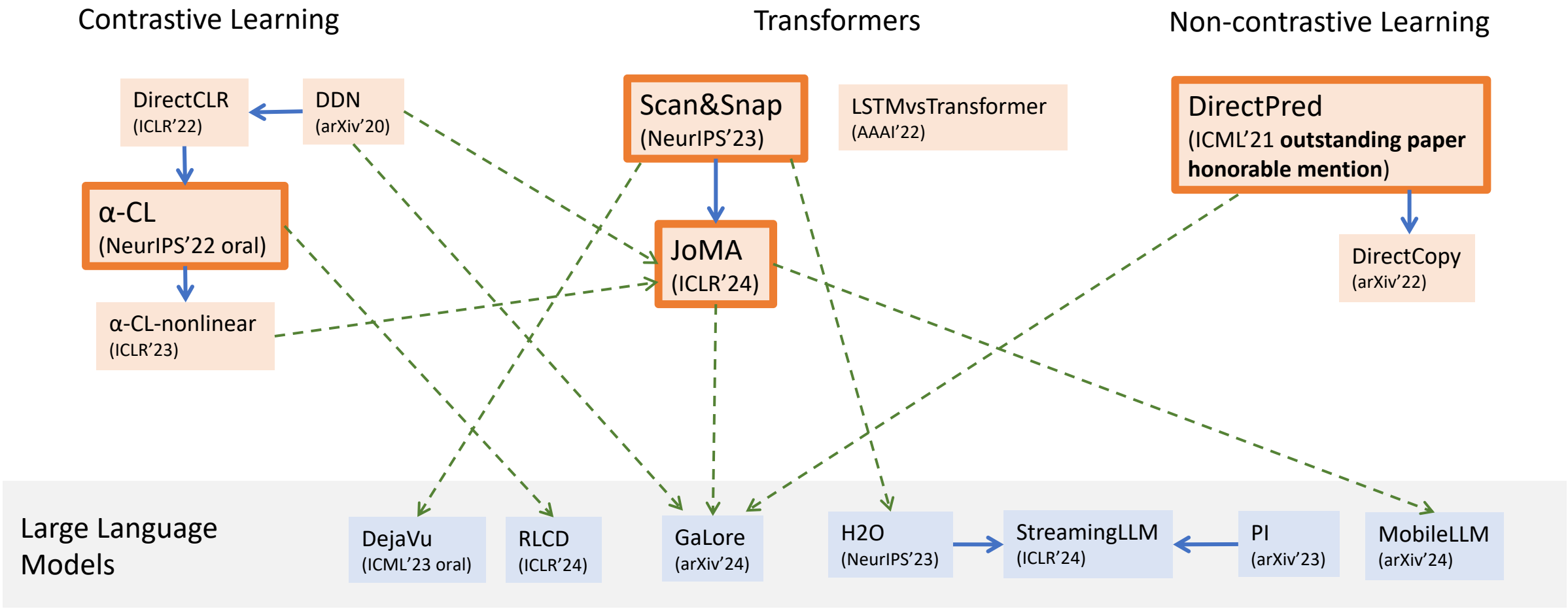
A sharp difference between theory and practice



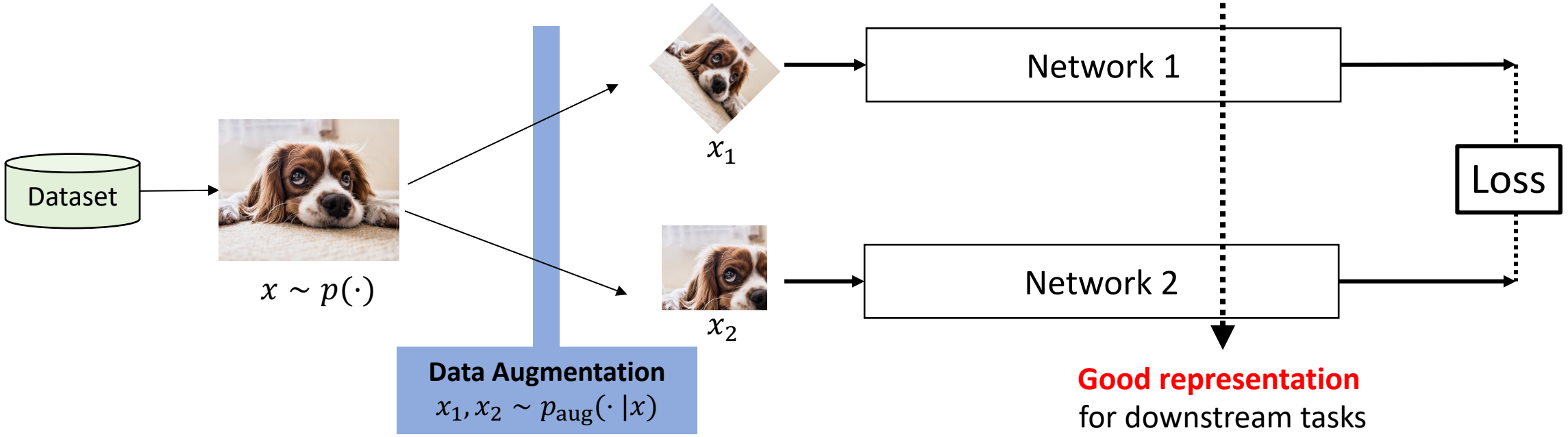
A sharp difference between theory and practice



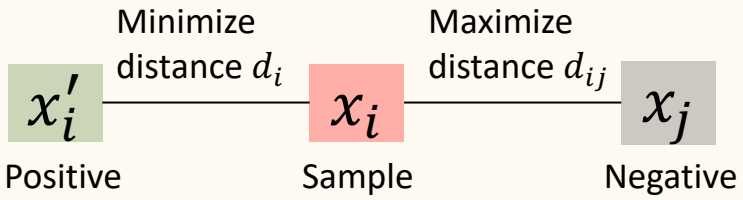
Understanding Representations Collapsing

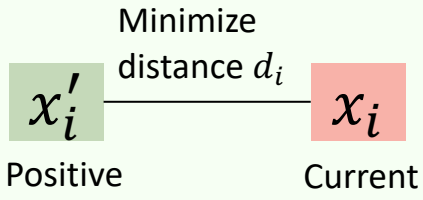


Contrastive versus Non-contrastive Learning



Contrastive Learning

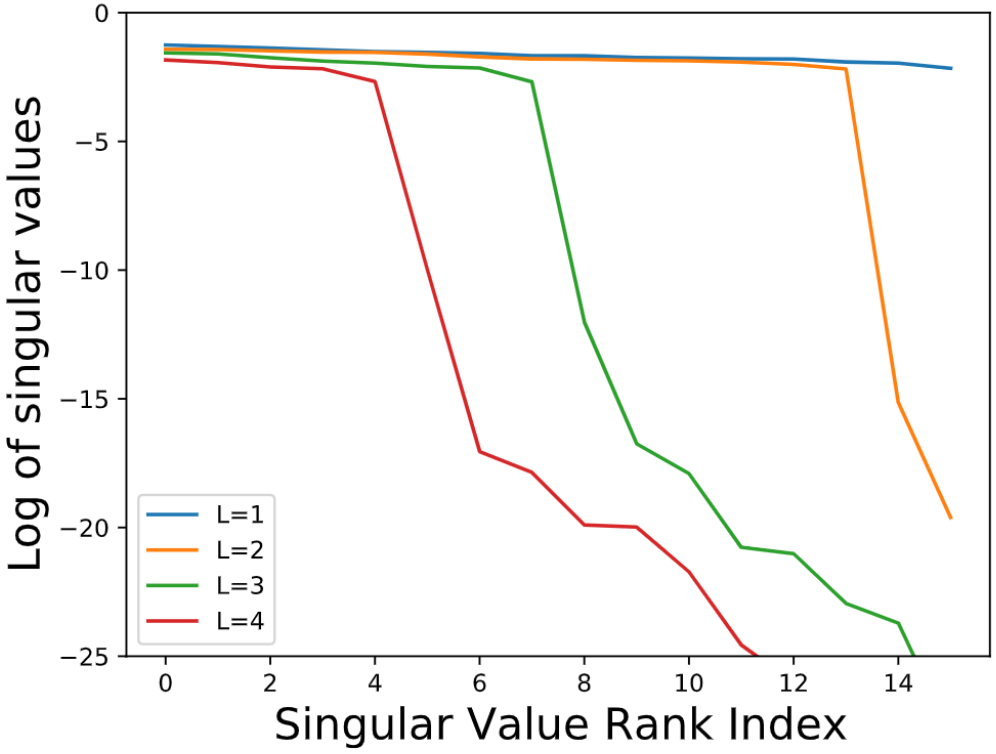


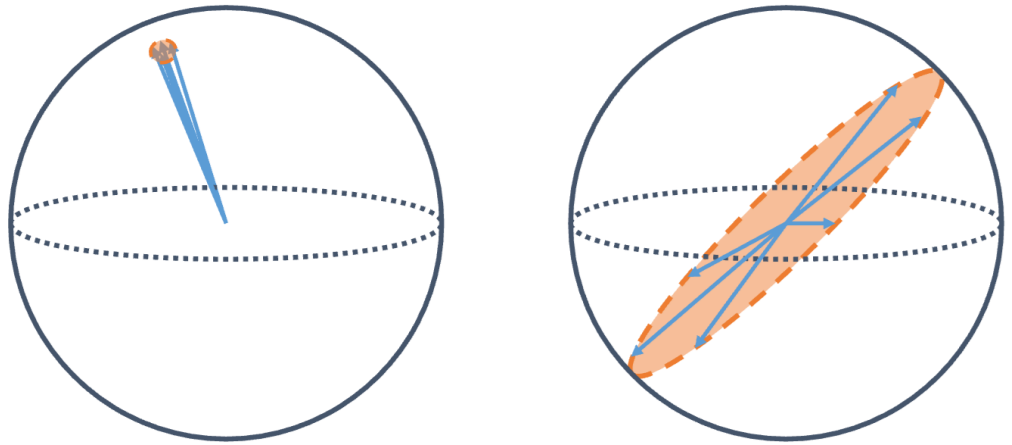


Non-Contrastive Learning

Representation Collapses in Contrastive Learning

Shouldn't contrastive learning make full use of all dimensions? The answer is **No…**



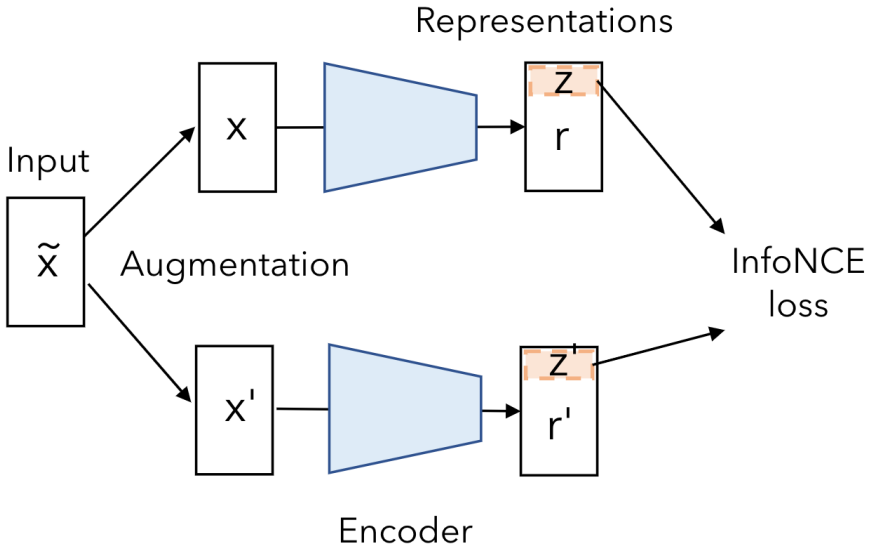


complete collapse dimensional collapse

DirectCLR *[L. Jing, P. Vincent, Y. LeCun, **Y. Tian**, Understanding Dimensional Collapse in Contrastive Self-supervised Learning]*

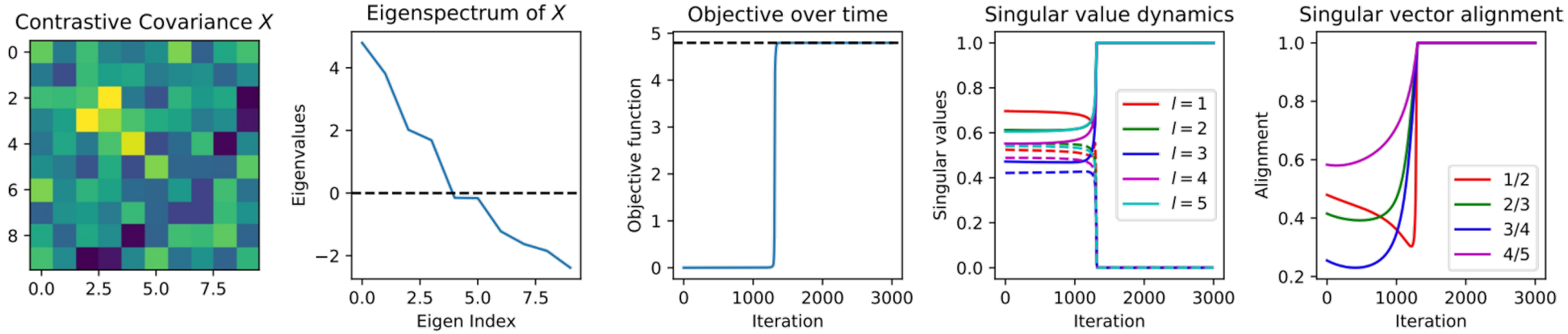
Representation Collapses in Contrastive Learning

If things are collapsed during training, why not just pick a subset of the dimensions directly?



Loss function	Projector	Top-1 Accuracy
SimCLR	2-layer nonlinear projector	66.5
SimCLR	1-layer linear projector	61.1
SimCLR	no projector	51.5
DirectCLR	no projector	62.7

Representation Collapses in Contrastive Learning (Theoretical Study, Linear case)



Theorem 3 (Representation Learning with `DeepLin` is PCA). *If $\lambda_{\max}(X_\alpha) > 0$, then for any local maximum $\theta \in \Theta$ of Eqn. 11 whose $W_{>1}^\top W_{>1}$ has distinct maximal eigenvalue:*

- *there exists a set of unit vectors $\{v_l\}_{l=0}^L$ so that $W_l = v_l v_{l-1}^\top$ for $1 \le l \le L$, in part*
- *θ is global optimal with objective $\mathcal{E}^* = \lambda_{\max}(X_\alpha)$.*

Corollary 3. *If we additionally use per-filter normalization (i.e., $\|w_{lk}\|_2 = 1/\sqrt{n_l}$), then Thm. 3 holds and v_l is more constrained: $[v_l]_k = \pm 1/\sqrt{n_l}$ for $1 \le l \le L-1$.*

Linear deep network → Full collapse 😱

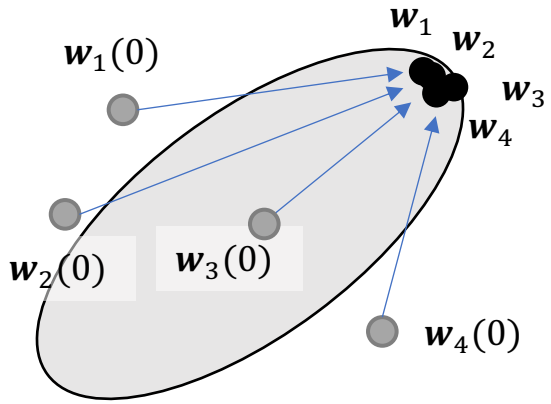
2. All W_l has rank-1 structure

	CIFAR-10			*STL-10*		
	100 epochs	300 epochs	500 epochs	100 epochs	300 epochs	500 epochs
$\mathcal{L}_{quadratic}$	63.59 ± 2.53	73.02 ± 0.80	73.58 ± 0.82	55.59 ± 4.00	64.97 ± 1.45	67.28 ± 1.21
$\mathcal{L}_{nce}$	84.06 ± 0.30	87.63 ± 0.13	87.86 ± 0.12	78.46 ± 0.24	82.49 ± 0.26	83.70 ± 0.12
backprop $\alpha(\boldsymbol{\theta})$	83.42 ± 0.25	87.18 ± 0.19	87.48 ± 0.21	77.88 ± 0.17	81.86 ± 0.30	83.19 ± 0.16
α-CL-r_H	84.27 ± 0.24	87.75 ± 0.25	87.92 ± 0.24	78.53 ± 0.35	82.62 ± 0.15	83.74 ± 0.18
α-CL-r_γ	83.72 ± 0.19	87.51 ± 0.11	87.69 ± 0.09	78.22 ± 0.28	82.19 ± 0.52	83.47 ± 0.34
α-CL-r_s	84.72 ± 0.10	86.62 ± 0.17	86.74 ± 0.15	76.95 ± 1.06	80.64 ± 0.77	81.65 ± 0.59
α-CL-direct	$\mathbf{85.09 \pm 0.13}$	$\mathbf{88.00 \pm 0.12}$	$\mathbf{88.16 \pm 0.12}$	$\mathbf{79.38 \pm 0.16}$	$\mathbf{82.99 \pm 0.15}$	$\mathbf{84.06 \pm 0.24}$

[**Y. Tian**, *Understanding Deep Contrastive Learning via Coordinate-wise Optimization*, NeurIPS'22]

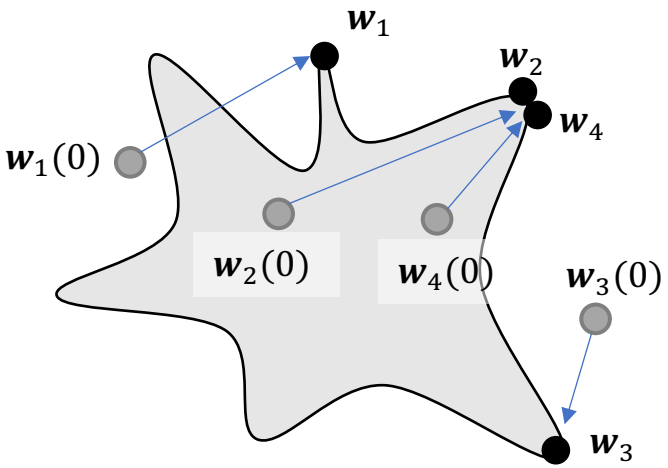
Representation Collapses in Contrastive Learning (Theoretical Study, Nonlinear case)

How to prevent complete collapsing?? Using Nonlinearity!!



Linear model

1. Every w_k converges to the global maximal eigenvector
2. More nodes do NOT help.

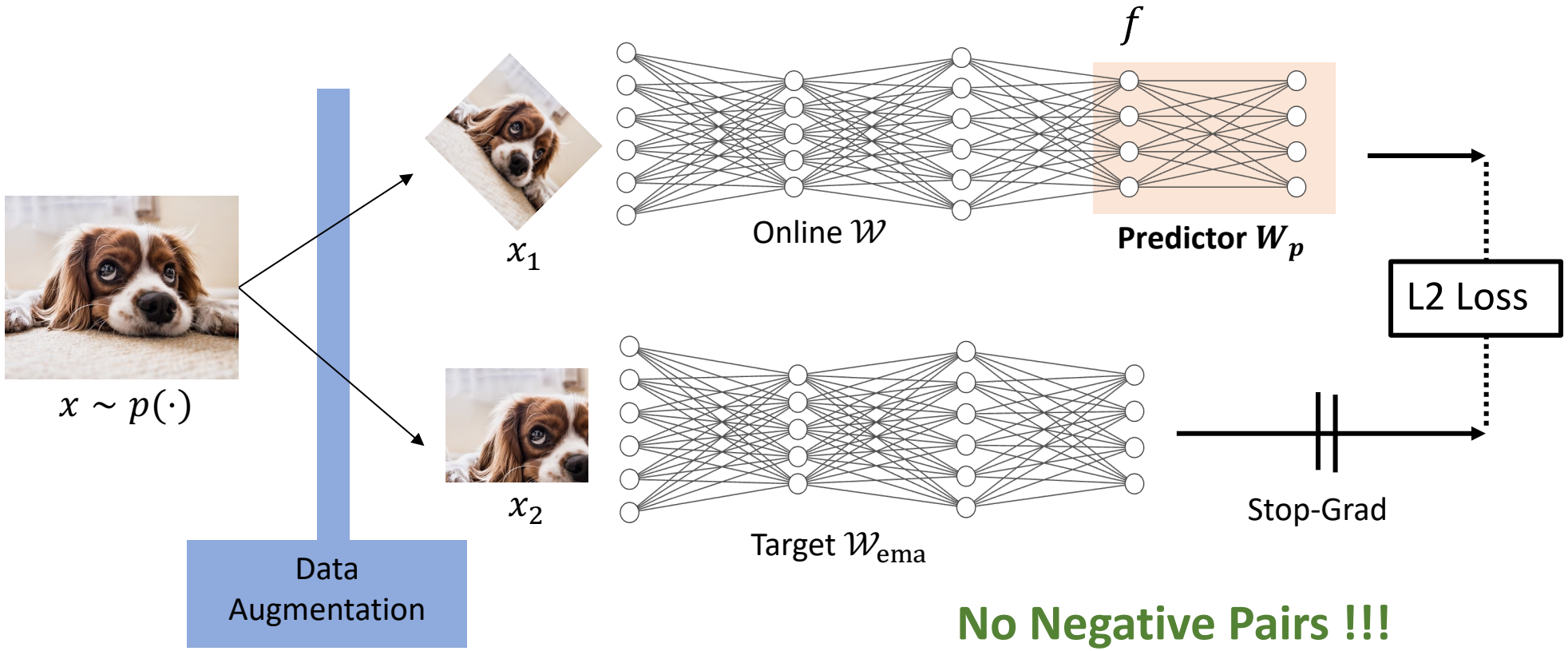


Nonlinear model

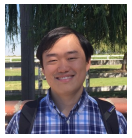
1. Each w_k can converge to different patterns
2. More nodes with diverse initialization learn more patterns!

[**Y. Tian**, *Understanding the Role of Nonlinearity in Training Dynamics of Contrastive Learning*, ICLR'23]

Why Non-contrastive Learning doesn't collapse?

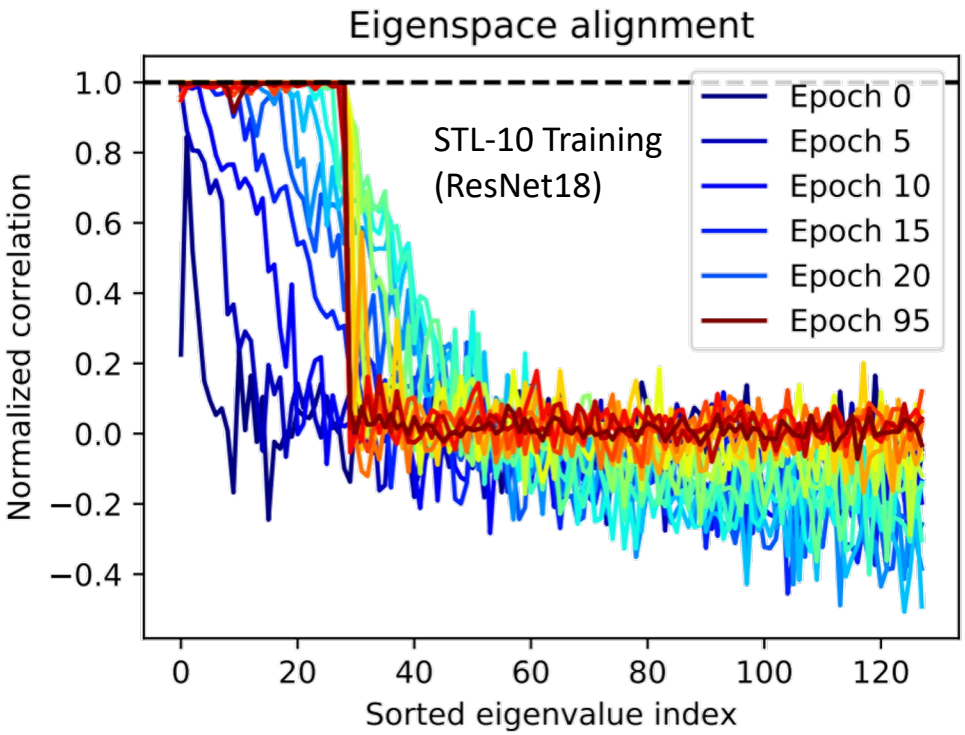


No Negative Pairs !!!



DirectPred [**Y. Tian** et al, Understanding Self-Supervised Learning Dynamics without Contrastive Pairs, ***ICML'21 Outstanding Paper Honorable Mentions***]

Why Non-contrastive Learning doesn't collapse?

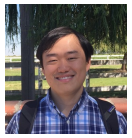


Theorem 3: Under certain conditions,

$$FW_p - W_pF \to 0 \text{ when } t \to +\infty$$

and the eigenspace of W_p and F gradually **aligns**.

$F \coloneqq \mathbb{E}[\boldsymbol{f}\boldsymbol{f}^T]$ is the statistics of the input before W_p



DirectPred [**Y. Tian** et al, Understanding Self-Supervised Learning Dynamics without Contrastive Pairs, ***ICML'21 Outstanding Paper Honorable Mentions***]

Why Non-contrastive Learning doesn't collapse?

1D dynamics of the eigenvalue p_j of W_p:

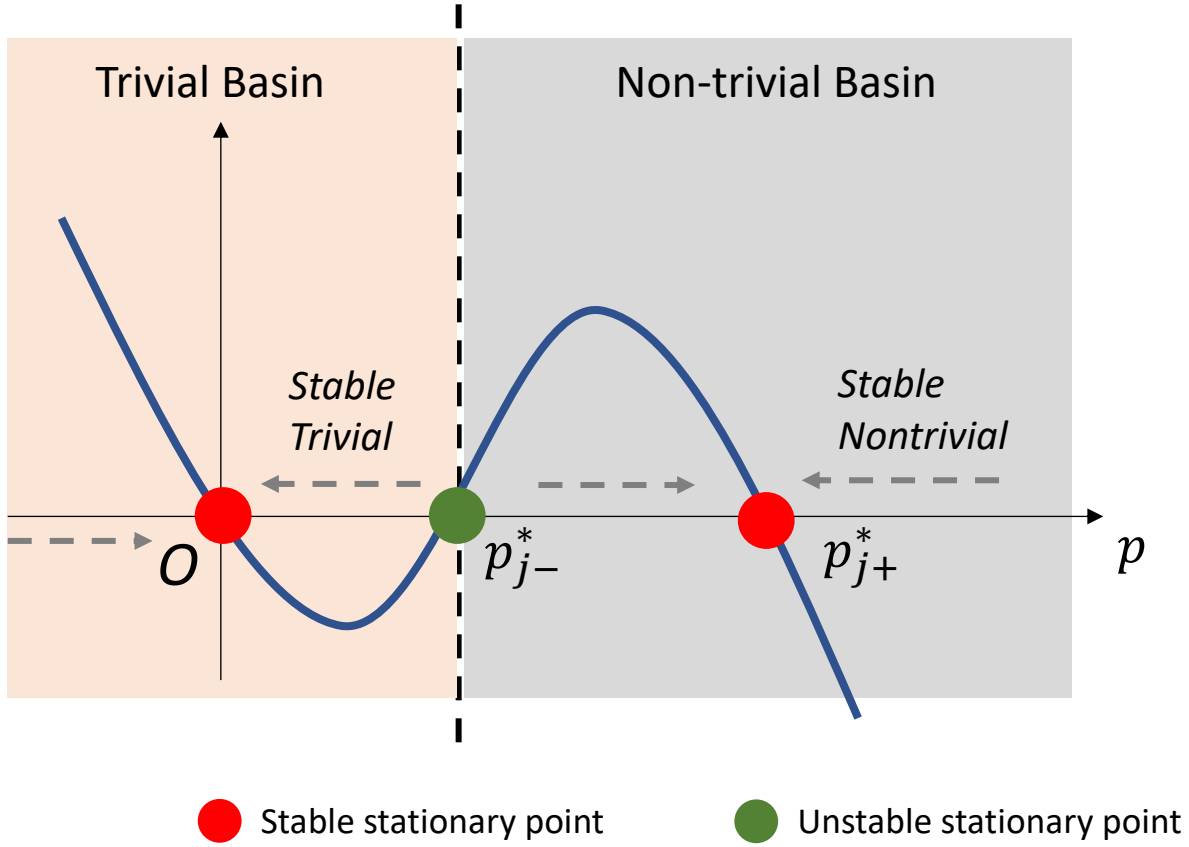
$$\dot{p}_j = p_j^2 \left[\tau(t) - (1+\sigma^2)p_j\right] - \eta p_j$$

EMA

Variance due to data augmentation

Weight Decay

$$p_{j-}^* = \frac{\tau - \sqrt{\tau^2 - 4\eta(1+\sigma^2)}}{2(1+\sigma^2)} \sim \frac{\eta}{\tau}$$



DirectPred: Practical Algorithm over SGD

Setting W_p directly rather than using gradient update.

1. Estimate $\hat{F} = \rho\hat{F} + (1-\rho)E[\boldsymbol{f}\boldsymbol{f}^T]$
2. Eigen-decompose $\hat{F} = \hat{U}\Lambda_F\hat{U}^T$
3. Set W_p accordingly

Guaranteed Eigenspace Alignment ☺

Downstream Classification Top-1	Number of epochs 100	300	500
STL-10			
DirectPred	$\mathbf{77.86 \pm 0.16}$	78.77 ± 0.97	78.86 ± 1.15
DirectPred (freq=5)	77.54 ± 0.11	$\mathbf{79.90 \pm 0.66}$	$\mathbf{80.28 \pm 0.62}$
SGD baseline	75.06 ± 0.52	75.25 ± 0.74	75.25 ± 0.74
CIFAR-10			
DirectPred	$\mathbf{85.21 \pm 0.23}$	$\mathbf{88.88 \pm 0.15}$	89.52 ± 0.04
DirectPred (freq=5)	84.93 ± 0.29	88.83 ± 0.10	$\mathbf{89.56 \pm 0.13}$
SGD baseline	84.49 ± 0.20	88.57 ± 0.15	89.33 ± 0.27

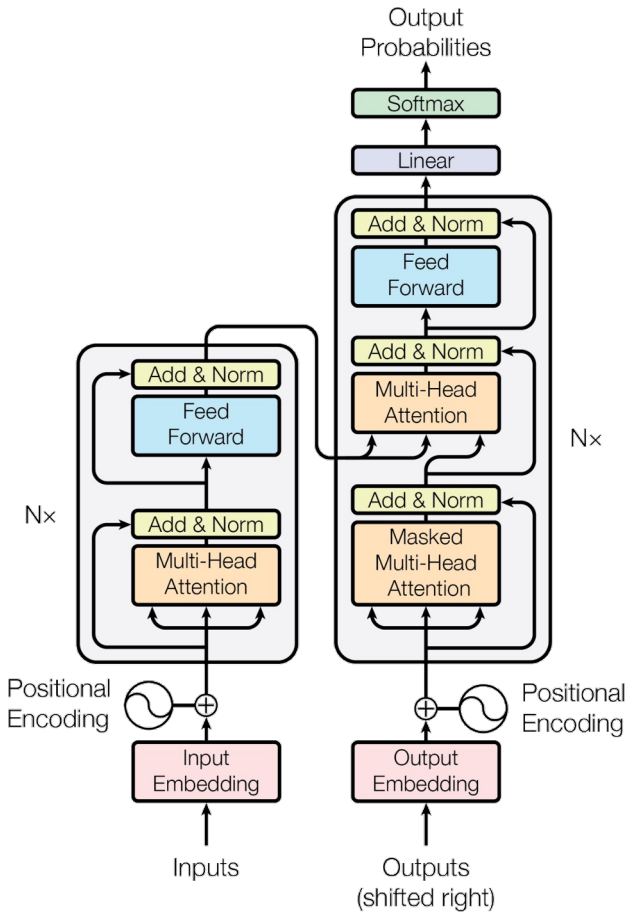
Downstream classification (ImageNet):

BYOL variants	*Accuracy (60 ep)* Top-1	Top-5	*Accuracy (300 ep)* Top-1	Top-5
2-layer predictor*	**64.7**	**85.8**	**72.5**	90.8
linear predictor	59.4	82.3	69.9	89.6
DirectPred	64.4	**85.8**	72.4	**91.0**

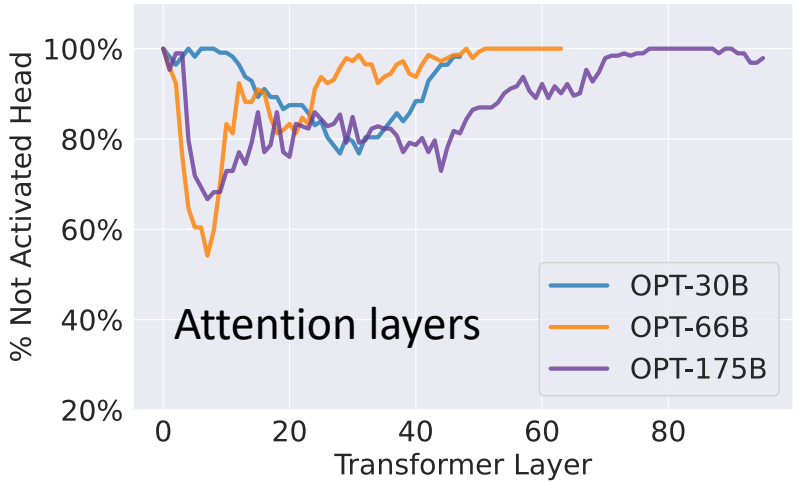
* 2-layer predictor is BYOL default setting.

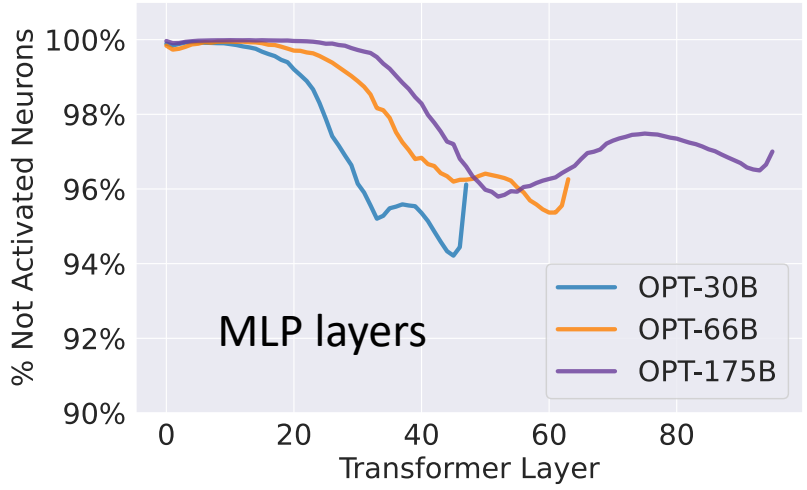
DirectPred using linear predictor is better than SGD with linear predictor, and is comparable with 2-layer predictor.

Let's check Collapsing ("sparsity") in Transformers!



[A. Vaswani et al, Attention is all you need, NeurIPS'17]

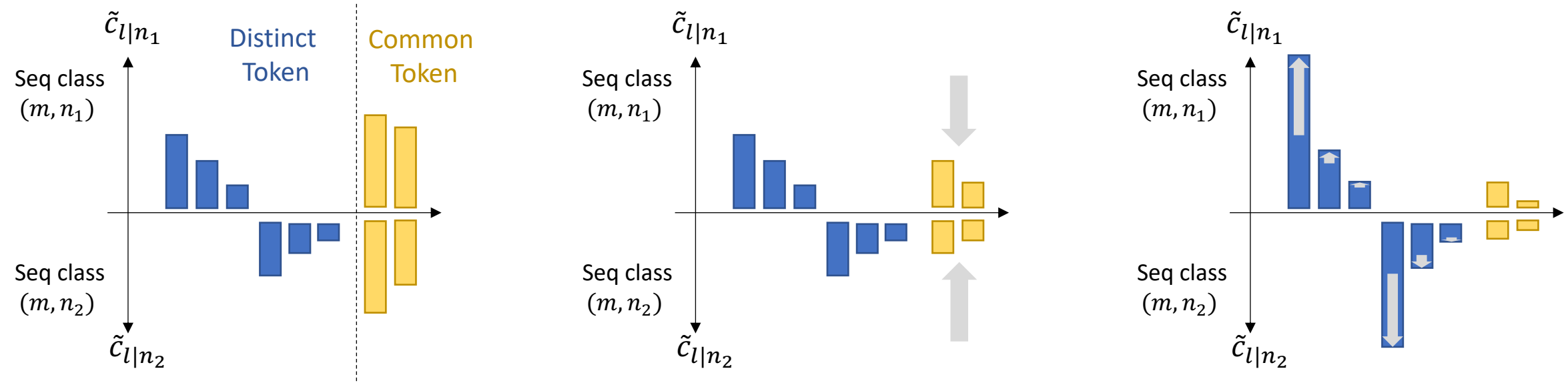




[Z. Liu et al, *Deja vu: Contextual sparsity for efficient LLMs at inference time*, ICML'23 (oral)]

Representation Collapses ("sparsity") in Self-Attention

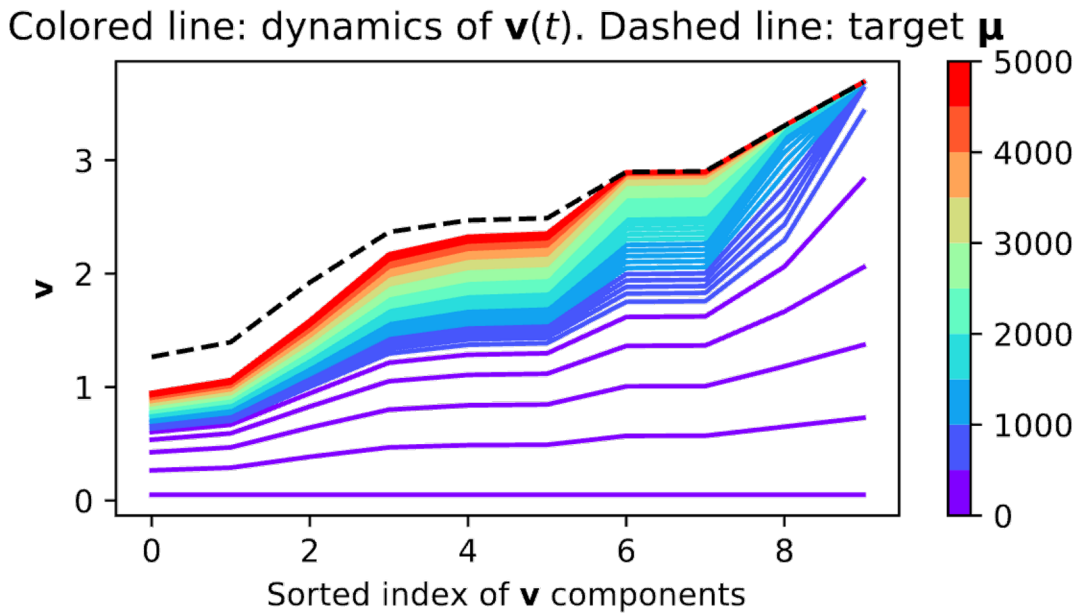
One layer Transformer, linear MLP

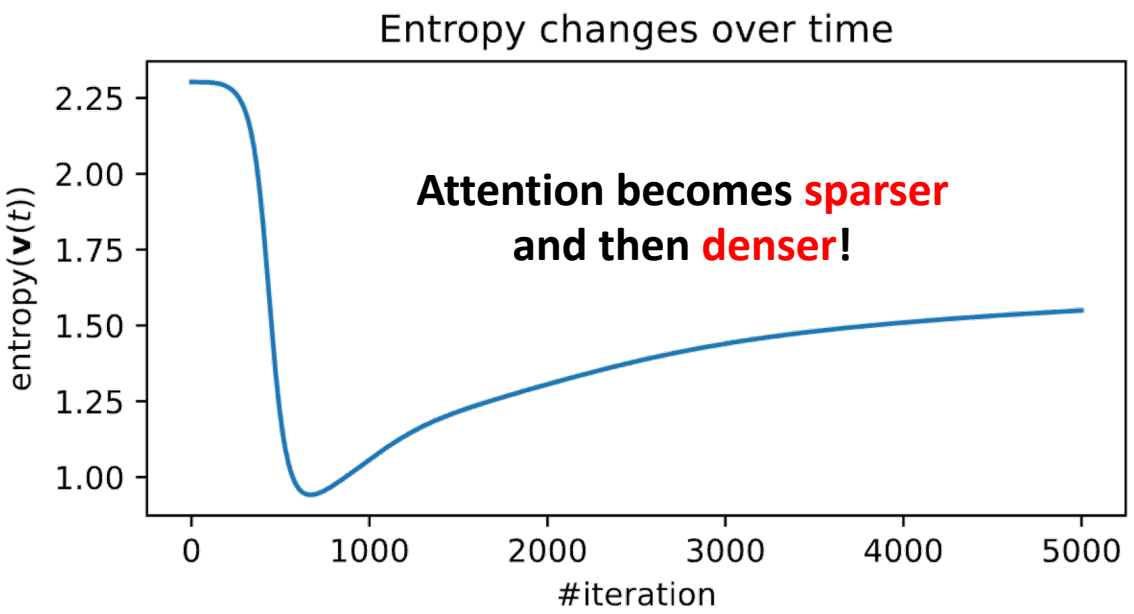


[**Y. Tian** et al, *Scan and Snap: Understanding Training Dynamics and Token Composition in 1-layer Transformer,* NeurIPS'23]

Attn Collapses and Resumes(!) for nonlinear MLP

Multilayer Transformer, non-linear MLP





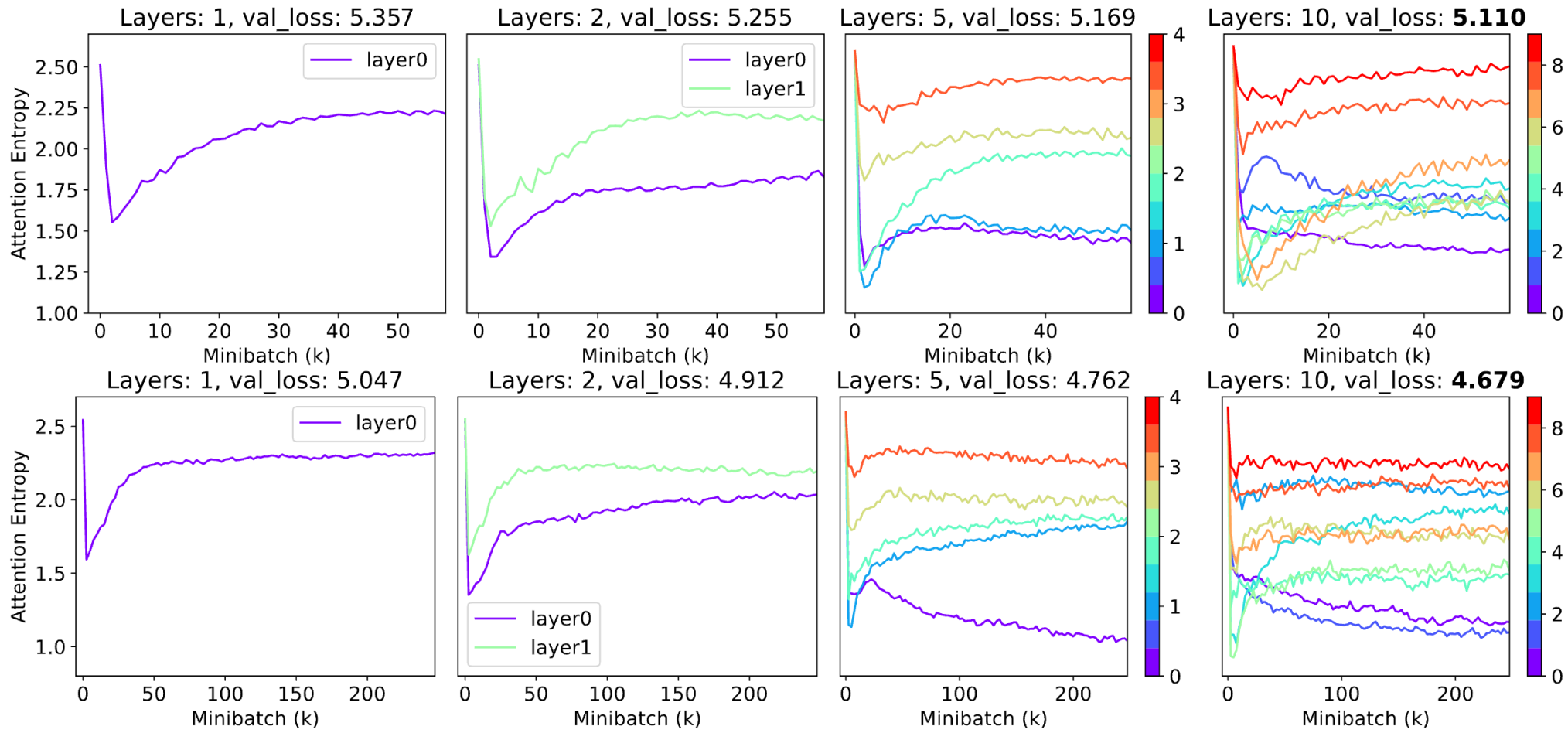
Speed of salient feature learning >>> speed of non-salient feature learning

[**Y. Tian** et al, *JoMA: Demystifying Multilayer Transformers via JOint Dynamics of MLP and Attention,* ICLR'24]

Attn Collapses and Resumes(!) for nonlinear MLP

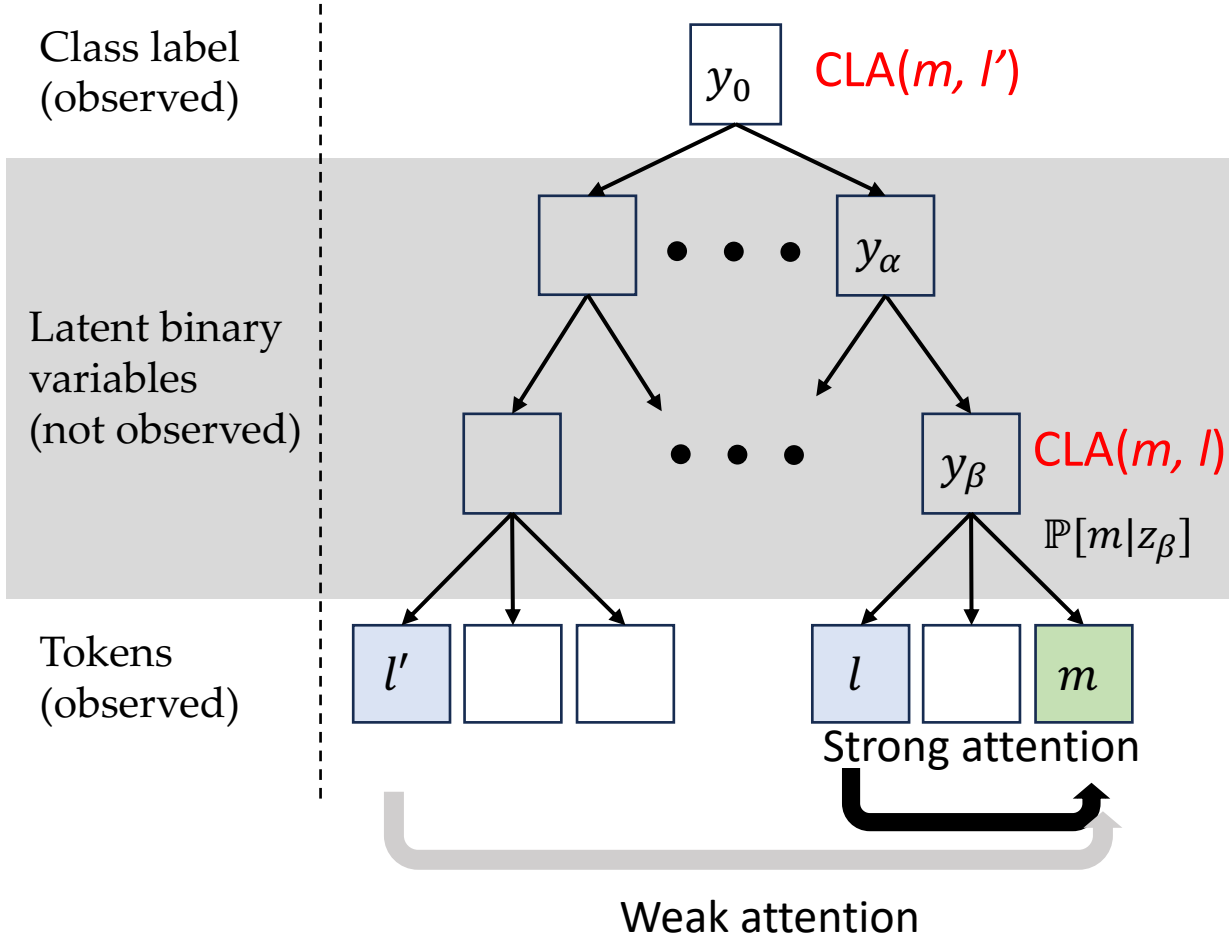
Wikitext2

Wikitext103



[**Y. Tian** et al, *JoMA: Demystifying Multilayer Transformers via JOint Dynamics of MLP and Attention,* ICLR'24]

Data Hierarchy & Multilayer Transformer



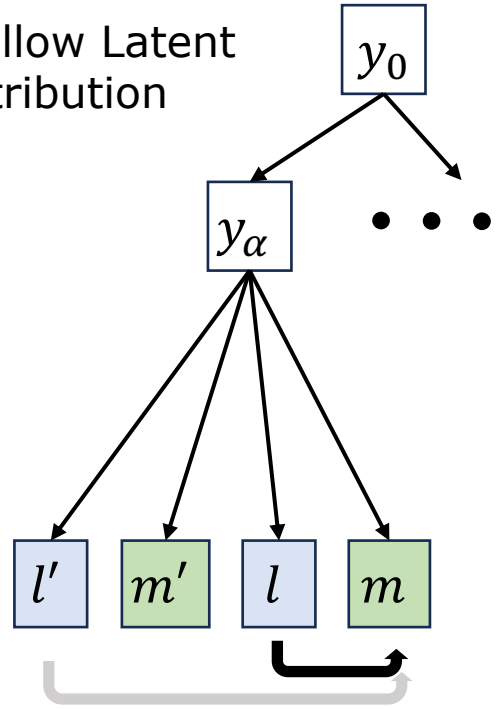
$$\mathbb{P}[l|m] \approx 1 - \frac{H}{L}$$

H: height of the common latent ancestor (CLA) of l & m

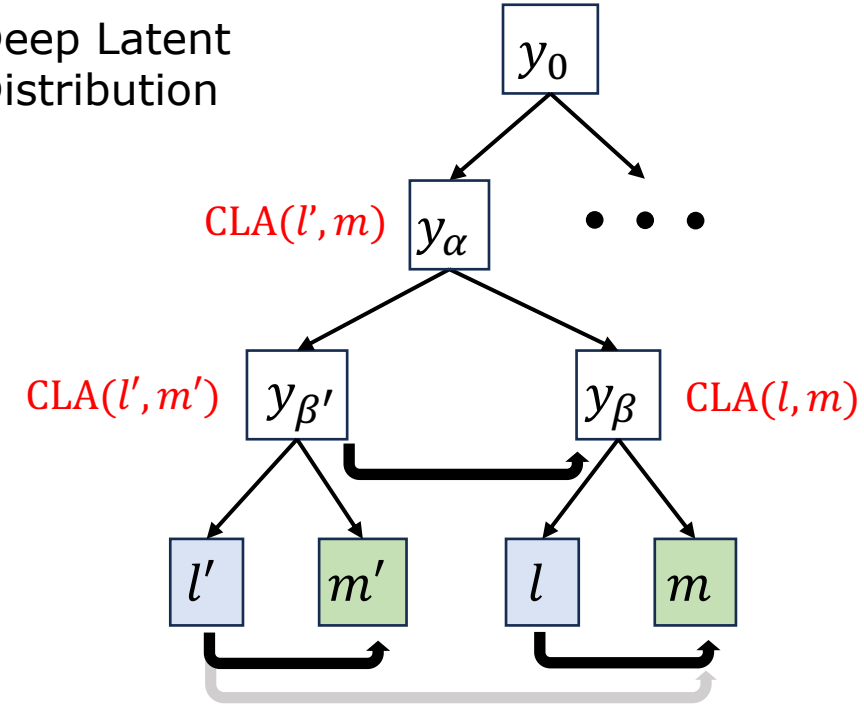
L: total height of the hierarchy

Collapsing enables Hierarchy-agnostic Learning!

Shallow Latent Distribution



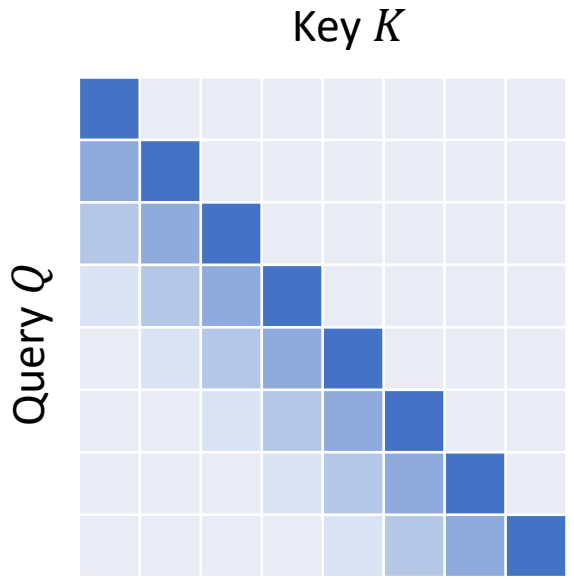
Deep Latent Distribution

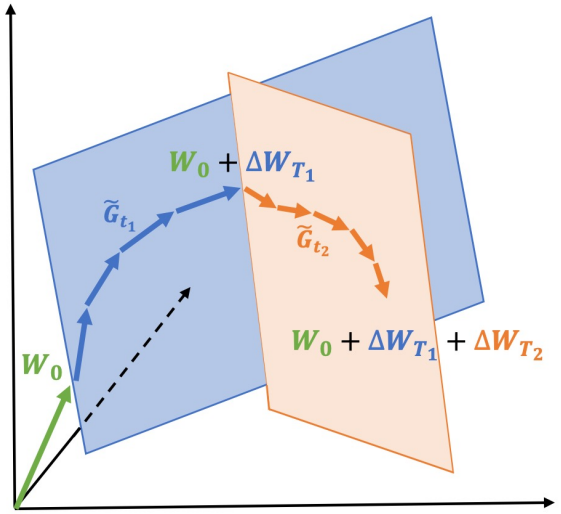


Learning the current hierarchical structure by
slowing down the association of tokens that are not directly correlated

Efficient Learning/ Inference

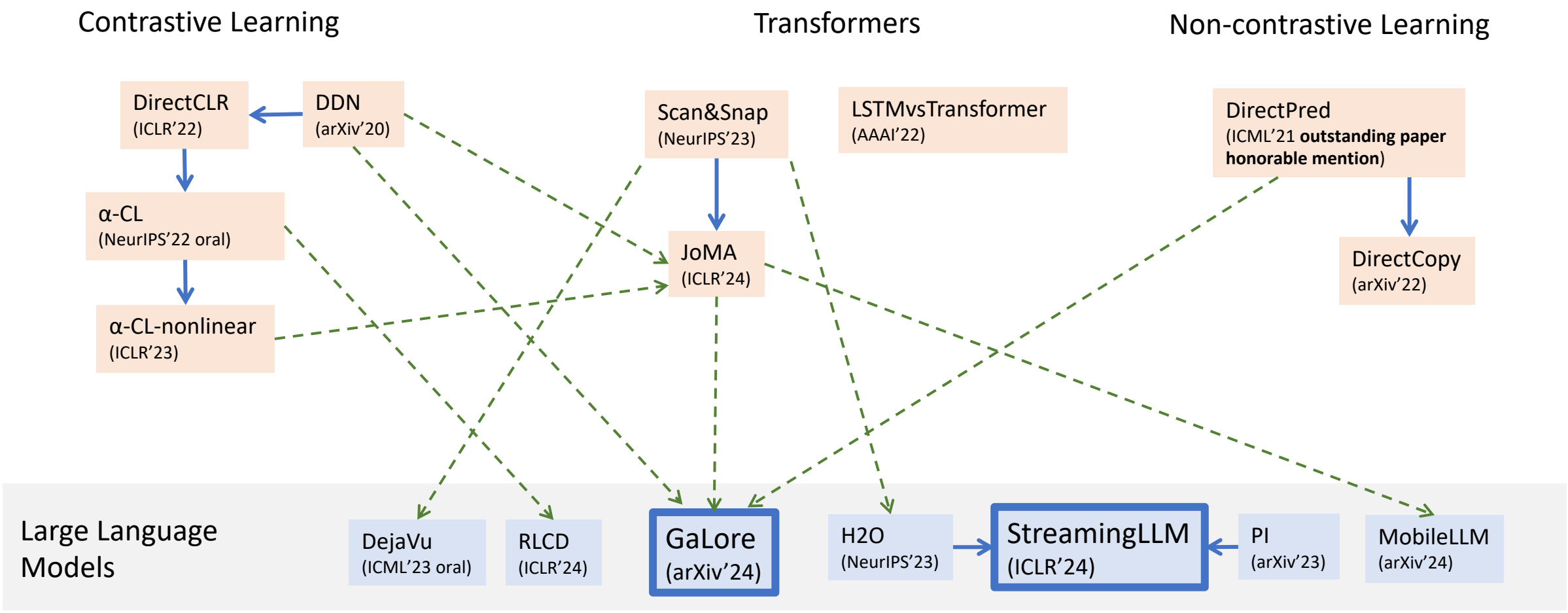
Small memory, less compute



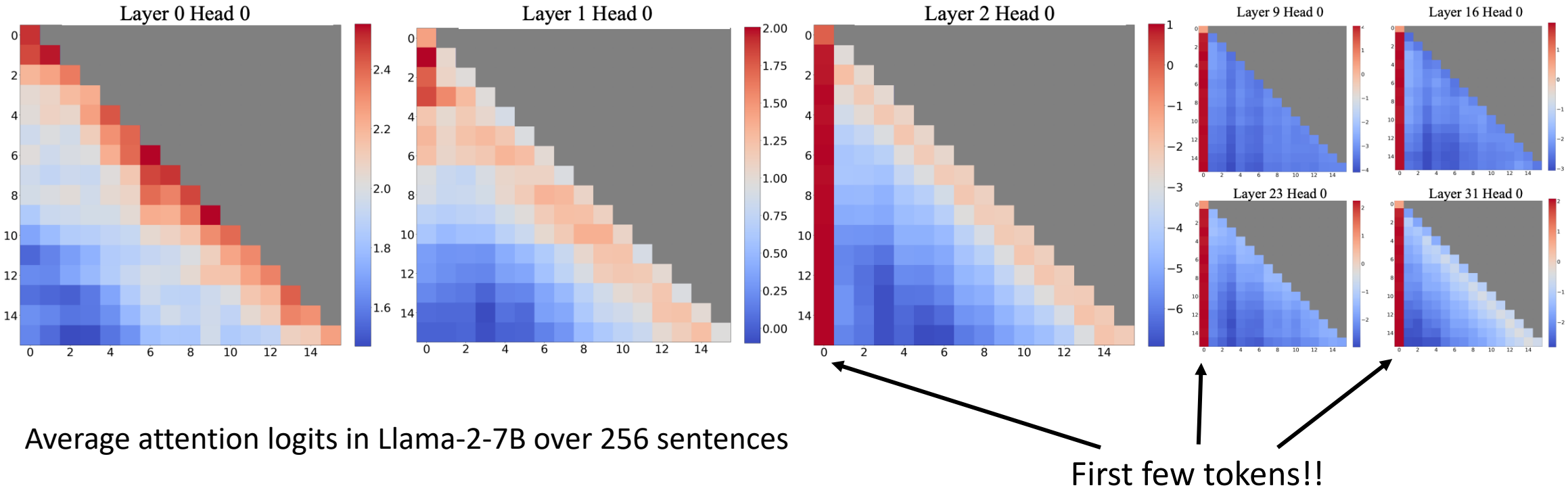


Part II: Design Efficient Training/Inference Approach

Understanding Collapsing of Representations



Attention Sinks: Initial tokens draw a lot of attentions

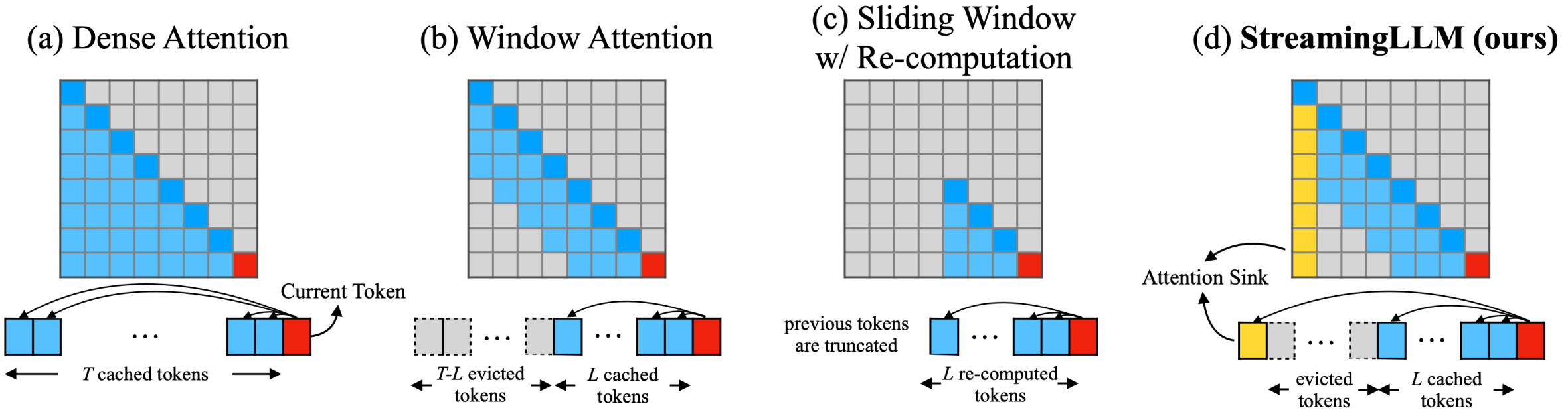


Average attention logits in Llama-2-7B over 256 sentences

- Observation: Initial tokens have large attention scores, even if they're not semantically significant.
- Attention Sink: Tokens that disproportionately attract attention irrespective of their relevance.

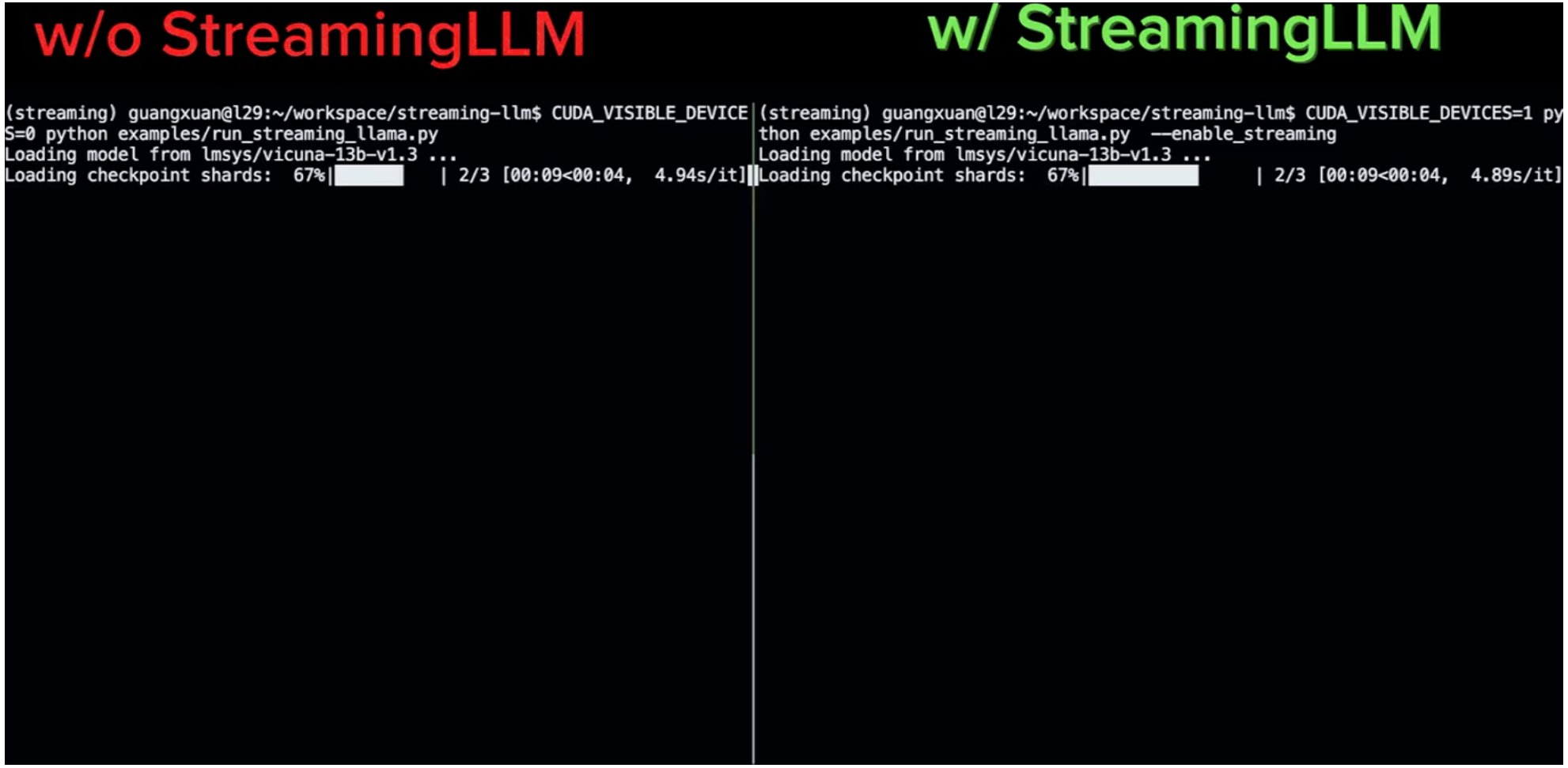
[G. Xiao, **Y. Tian**, et al, *Efficient Streaming Language Models with Attention Sinks,* ICLR'24]

StreamingLLM



[G. Xiao, **Y. Tian**, et al, *Efficient Streaming Language Models with Attention Sinks*, ICLR'24]

StreamingLLM



Impact of StreamingLLM

Massachusetts Institute of Technology

Education Research Innovation Admissions + Aid Campus Life News

MIT News
ON CAMPUS AND AROUND THE WORLD

SUBSCRIB

A new way to let AI chatbots converse all day without crashing

Researchers developed a simple yet effective solution for a puzzling problem that can worsen the performance of large language models such as ChatGPT.

Adam Zewe | MIT News
February 13, 2024

Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | D

← Back to blog

Attention Sinks in endless fluency

Community blog post Published October 9, 2023

tomaarsen
Tom Aarsen

Events Video Special Issues Jobs **VentureBeat**

Artificial Intelligence | Security | Data Infrastructure | Automation | Enterpris

StreamingLLM shows how one token can keep AI models running smoothly indefinitely

EFFICIEN
WITH AT

Guangxuan Xiao[1]* Yuandong Tian[2] Beidi Chen[3] Song Han[1] Mike Lew

[1] Massachusetts Institute of Technology
[2] Meta AI
[3] Carnegie Mellon University

https://github.com/mit-han-lab/streaming-llm

ABSTRACT

Deploying Large Language Models (LLMs) in streaming multi-round dialogue, where long interactions are expected, challenges. Firstly, during the deco

Efficient Streaming Language Models with Attention Sinks (Paper Explained)

Yannic Kilcher
247K subscribers

Subscr

huggingface / transformers Public

Code | Issues 799 | Pull requests 252 | Actions | Projects 25 | Security | Insights

Generate: New `Cache` abstraction and Attention Sinks su

Merged ydshieh merged 35 commits into huggingface:main from tomaarsen:feat/kv_cache_class on Dec

Conversation 135 | Commits 35 | Checks 3 | Files changed 14

tomaarsen commented on Oct 9, 2023 • edited

Closes #26553

Hello!

What does this PR do?

Haihao Shen
@HaihaoShen

📢StreamingLLM landed in Intel Extension for Transformers LLM inference infinity on CPU, up to 4M tokens!
🎯Check out the code: github.com/intel/intel-ex..., search "StreamingLLM" and have a try!
#oneapi @intel @huggingface @Guangxuan_Xiao @_akhaliq

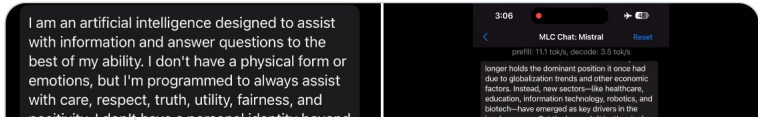
intel/**intel-extension-for-transformers**

intel.

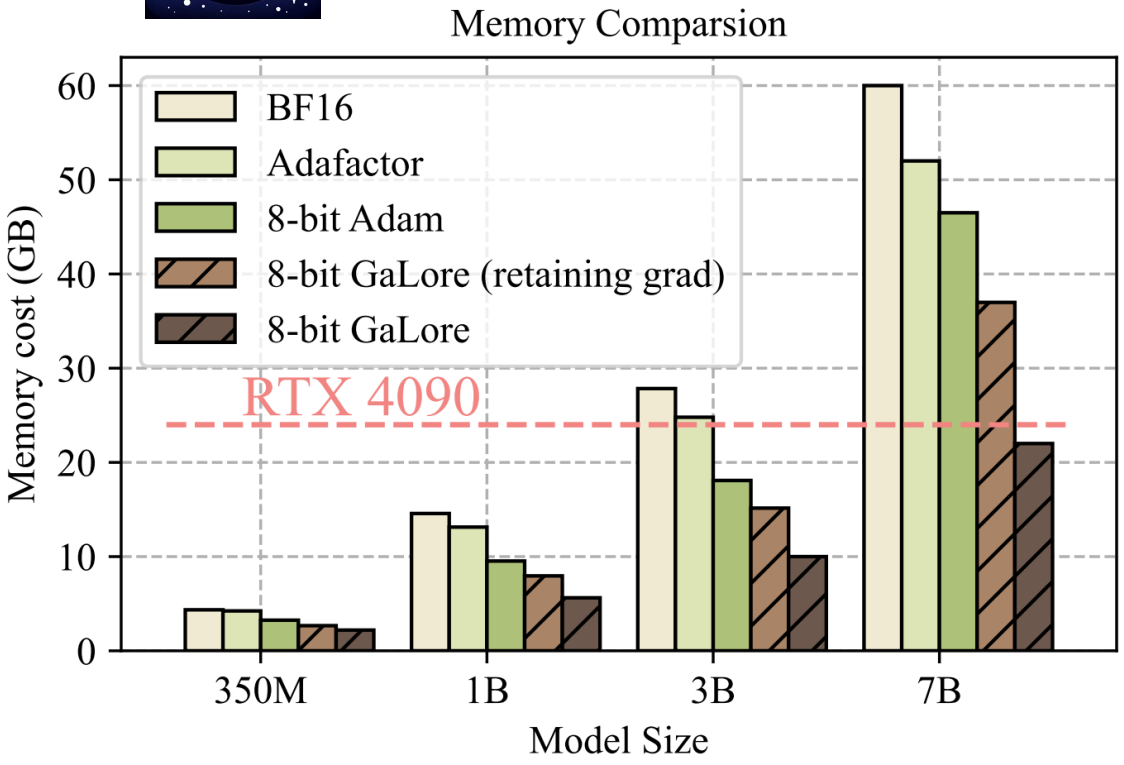
David Pissarra
@davidpissarra

Run the Mistral-7B-Instruct-v0.2 model on iPhone! Supports now StreamingLLM for endless generation. Try the MLC Chat App via TestFlight llm.mlc.ai

For native LLM deployment, attention sinks are particularly helpful for longer generation with less memory requirement.



GaLore: Pre-training 7B model on RTX 4090 (24G)



	Rank	Retain grad	Memory	Token/s
8-bit AdamW		Yes	40GB	1434
8-bit GaLore	16	Yes	28GB	1532
8-bit GaLore	128	Yes	29GB	1532
16-bit GaLore	128	Yes	30GB	**1615**
16-bit GaLore	128	No	**18GB**	1587
8-bit GaLore	1024	Yes	36GB	1238

* SVD takes around 10min for 7B model, but runs every T=500-1000 steps.

Third-party evaluation by @llamafactory_ai

[J. Zhao, …, **Y. Tian**, *GaLore: Memory-Efficient LLM Training by Gradient Low-Rank Projection*, arXiv'24]

Full-rank Training

Regular full-rank training. At time step t, $G_t = -\nabla_W \varphi_t(W_t) \in \mathbb{R}^{m \times n}$ is the backpropagated (negative) gradient matrix. Then the regular pre-training weight update can be written down as follows (η is the learning rate):

$$W_T = W_0 + \eta \sum_{t=0}^{T-1} \tilde{G}_t = W_0 + \eta \sum_{t=0}^{T-1} \rho_t(G_t) \qquad (1)$$

Adam (needs running momentum M_t and variance V_t as optimizer states)

$$M_t = \beta_1 M_{t-1} + (1-\beta_1) G_t$$
$$V_t = \beta_2 V_{t-1} + (1-\beta_2) G_t^2$$
$$\tilde{G}_t = M_t / \sqrt{V_t + \epsilon}$$

Memory Usage	Weight (W)	Optim States (M_t, V_t)	Projection (P)	Total
Full-rank	mn	$2mn$	0	$3mn$

Low-rank Adaptor (LoRA)

Low-rank updates.. For a linear layer $W \in \mathbb{R}^{m \times n}$, LoRA and its variants utilize the low-rank structure of the update matrix by introducing a low-rank adaptor AB:

$$W_T = W_0 + B_T A_T, \tag{5}$$

And we optimize B_T and A_T using Adam

Adam (needs running momentum M_t and variance V_t as optimizer states)

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) G_t$$
$$V_t = \beta_2 V_{t-1} + (1 - \beta_2) G_t^2$$
$$\tilde{G}_t = M_t / \sqrt{V_t + \epsilon}$$

Memory Usage	Weight (W)	Optim States (M_t, V_t)	Projection (P)	Total
Full-rank	mn	$2mn$	0	$3mn$
Low-rank adaptor	$mn + mr + nr$	$2(mr + nr)$	0	$mn + 3(mr + nr)$

W_0 B_T A_T B_T A_T

Memory Saving with GaLore

Algorithm 1: GaLore, PyTorch-like

```
for weight in model.parameters():
    grad = weight.grad
    # original space -> compact space
    lor_grad = project(grad)
    # update by Adam, Adafactor, etc.
    lor_update = update(lor_grad)
    # compact space -> original space
    update = project_back(lor_update)
    weight.data += update
```

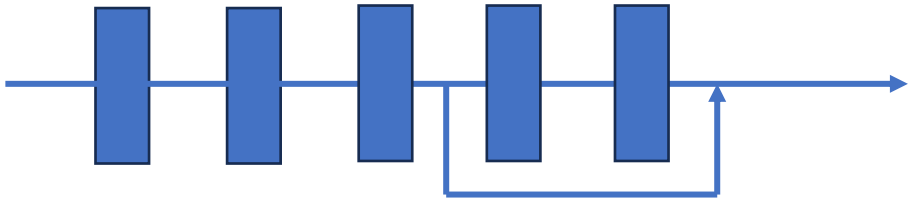
GaLore

$G_t \leftarrow -\nabla_W \phi(W_t)$
If t % T == 0:
    Compute $P_t = \text{SVD}(G_t) \in \mathbb{R}^{m \times r}$
$R_t \leftarrow P_t^T G_t$ *{project}*
$\tilde{R}_t \leftarrow \rho(R_t)$ *{Adam in low-rank}*
$\tilde{G}_t \leftarrow P_t \tilde{R}_t$ *{project-back}*
$W_{t+1} \leftarrow W_t + \eta \tilde{G}_t$

Memory Usage	Weight (W)	Optim States (M_t, V_t)	Projection (P)	Total
Full-rank	mn	$2mn$	0	$3mn$
Low-rank adaptor	$mn + mr + nr$	$2(mr + nr)$	0	$mn + 3(mr + nr)$
GaLore	mn	$2nr$	mr	$mn + mr + 2nr$
	W_t	R_t	P_t	

Why gradient is low-rank?

Reversible models [**Y. Tian**. DDN, arXiv'20]



There exists $K(\boldsymbol{x}; W)$ so that

1. [Forward] $\boldsymbol{y} = K(\boldsymbol{x}; W)\boldsymbol{x}$
2. [Backward] $\boldsymbol{g_x} = K^\top(\boldsymbol{x}; W)\boldsymbol{g_y}$

Here $K(\boldsymbol{x}; W)$ depends on the input x and weight W in the network $\mathcal{N}$.

Example: Linear, ReLU / LeakyReLU

Property of Reversible models

For reversible models trained with ℓ_2 loss or softmax

$$G_t = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{a}_i - B_i W_t \boldsymbol{f}_i)\boldsymbol{f}_i^\top$$

Here B_i are PSD matrices

Gradient becomes low-rank:

$$\mathrm{sr}(G_t) \le \mathrm{sr}\left(G_{t_0}^{\#}\right) + O\left[\left(\frac{1-\eta\lambda_2}{1-\eta\lambda_1}\right)^{2(t-t_0)}\right]$$

$\lambda_1 < \lambda_2$ are two smallest distinct eigenvectors of $S \coloneqq \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{f}_i\boldsymbol{f}_i^\top \otimes B_i$

Convergence Analysis

For gradient in the following form

$$G = \sum_i A_i - \sum_i B_i W C_i$$

Let $R = P^\top G Q$ be projected gradient, then

$$\|R_t\|_F \leq (1 - \eta M)\|R_{t-1}\|_F \to 0$$

Where $M := \frac{1}{N}\sum_i \min_t \lambda_{\min}(\hat{B}_{it})\lambda_{\min}(\hat{C}_{it}) - L_A - L_B L_C D^2$

Does that mean it works?

No… $R_t \to 0$ just means the gradient within the subspace vanishes.

How to continue optimization?

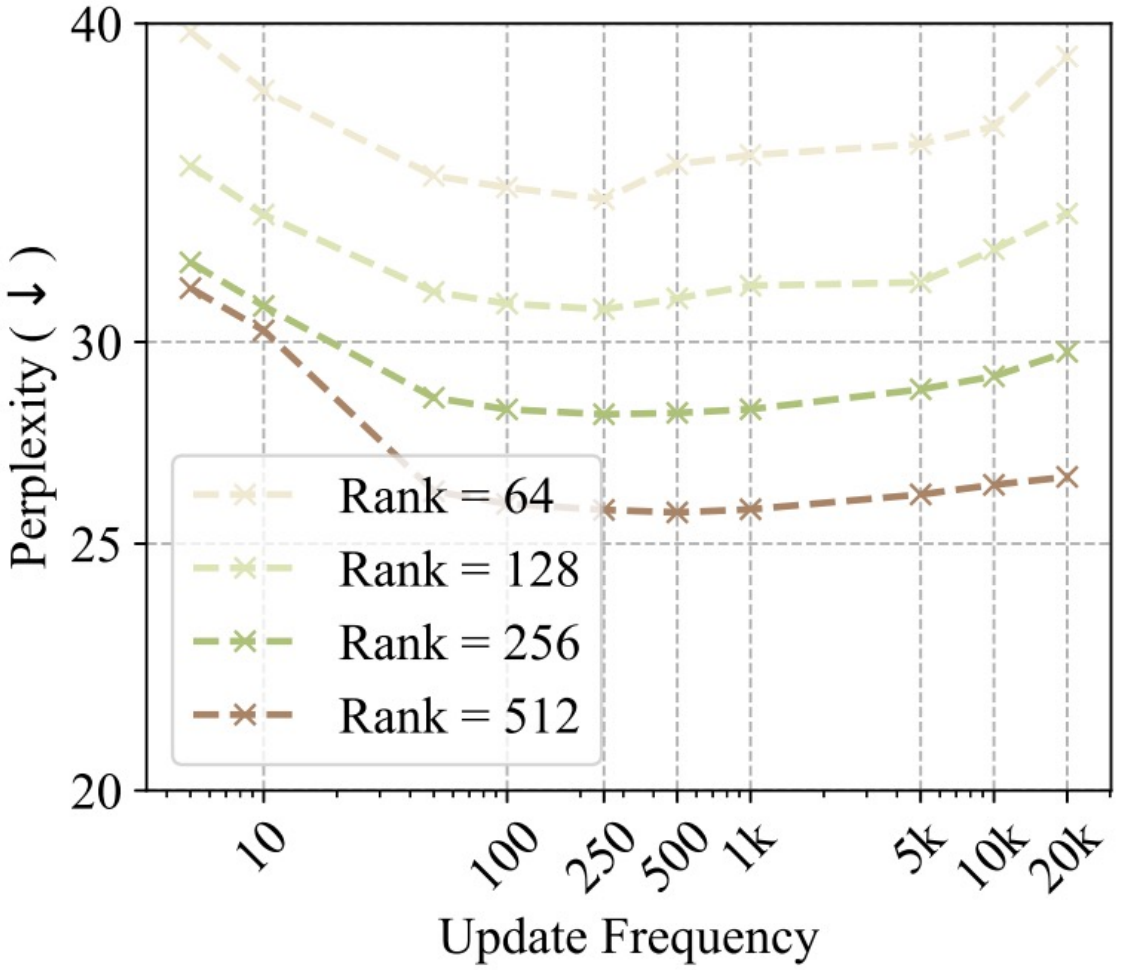
Change the projection from time to time!

If t % T == 0:

$$P_t = \text{SVD}(G_t) \in \mathbb{R}^{m \times r}$$

$$W_t = W_0 + \sum_i \Delta W_{T_i}$$

$$G = \sum_i A_i - \sum_i B_i W C_i$$



Pre-training Results (LLaMA 7B)

Params	Hidden	Intermediate	Heads	Layers	Steps	Data amount
60M	512	1376	8	8	10K	1.3 B
130M	768	2048	12	12	20K	2.6 B
350M	1024	2736	16	24	60K	7.8 B
1 B	2048	5461	24	32	100K	13.1 B
7 B	4096	11008	32	32	150K	19.7 B

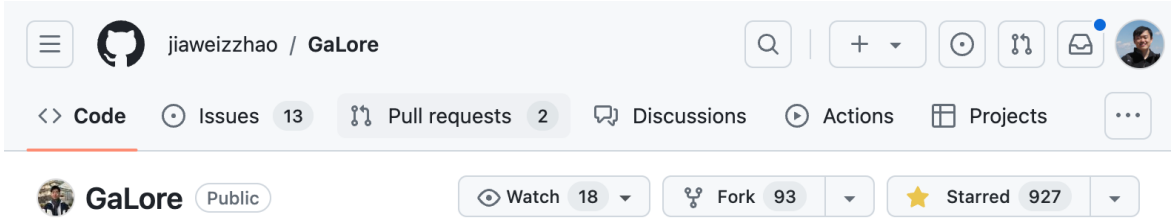
	Mem	40K	80K	120K	150K
8-bit GaLore	18G	17.94	15.39	14.95	14.65
8-bit Adam	26G	18.09	15.47	14.83	14.61
Tokens (B)		5.2	10.5	15.7	19.7

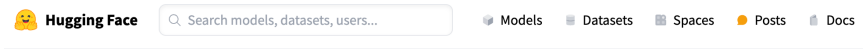
* Experiments are conducted on 8 x 8 A100

	60M	130M	350M	1B
Full-Rank	34.06 (0.36G)	25.08 (0.76G)	18.80 (2.06G)	15.56 (7.80G)
GaLore	**34.88** (0.24G)	**25.36** (0.52G)	**18.95** (1.22G)	**15.64** (4.38G)
Low-Rank	78.18 (0.26G)	45.51 (0.54G)	37.41 (1.08G)	142.53 (3.57G)
LoRA	34.99 (0.36G)	33.92 (0.80G)	25.58 (1.76G)	19.21 (6.17G)
ReLoRA	37.04 (0.36G)	29.37 (0.80G)	29.08 (1.76G)	18.33 (6.17G)
r/d_{model}	128 / 256	256 / 768	256 / 1024	512 / 2048
Training Tokens	1.1B	2.2B	6.4B	13.1B

* On LLaMA 1B, ppl is better (~14.97) with ½ rank (1024/2048)

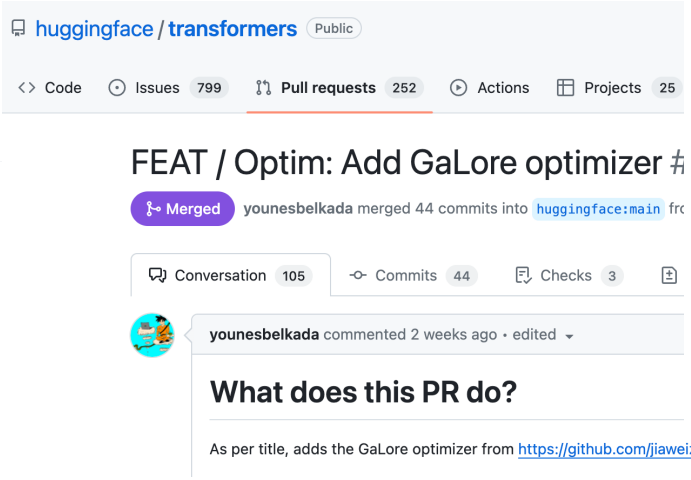
Impact of GaLore





Back to blog

GaLore: Advancing Large Model Training on Consumer-grade Hardware



FEDML AI @FEDML_AI

LinkedIn / Twitter post:

🚀 Exciting News! 🚀 #pretraining #finetuning #llm #GaLore #FEDML

🌟 FEDML Nexus AI platform now unlocks the pre-training and fine-tuning of LLaMA-7B on geo-distributed RTX4090s!

📈 By supporting the newly developed GaLore as a ready-to-launch job in FEDML Nexus AI, we have enabled the pre-training and fine-tuning of models like LLaMA 7B with a token batch size of 256 on a single RTX 4090, without additional memory optimization.

Memory-efficient LLM Training with GaLore

A novel approach for memory-efficient LLM finetuning, how to use it and what to expect

Geronimo · Follow
6 min read · 1 day ago

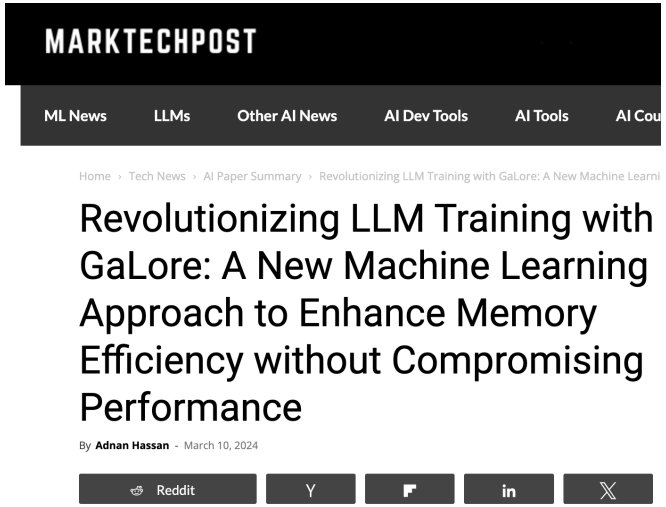
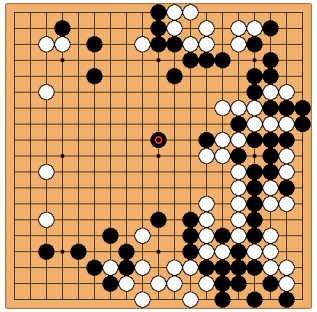


Table 1: Efficient fine-tuning techniques featured in LLAMAFACTORY. Techniques that are compatible with each other are marked with ✓, while those that are not compatible are marked with ✗.

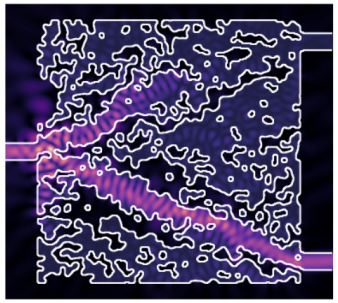
	Freeze-tuning	GaLore	LoRA	DoRA
Mixed precision	✓	✓	✓	✓
Checkpointing	✓	✓	✓	✓
Flash attention	✓	✓	✓	✓
S^2 attention	✓	✓	✓	✓
Quantization	✗	✗	✓	✓
Unsloth	✗	✗	✓	✗

Effective Design Application

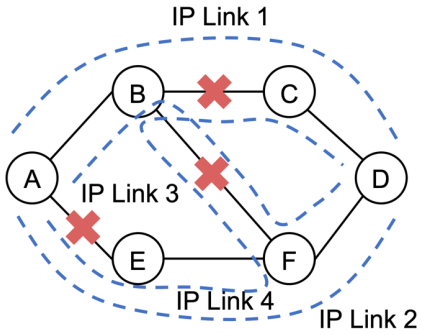
Strong performance in real-world cases



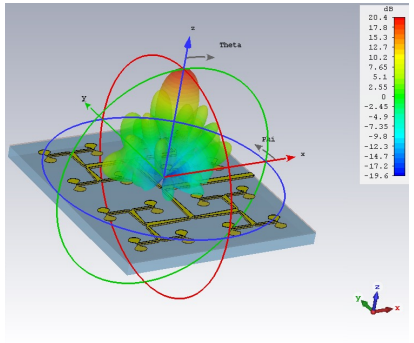
Go Game



Inverse Photonics Design



Network Planning



Antenna Design

Part III: Real-world Design Application

DarkForestGo (2015)

DarkForest versus Koichi Kobayashi (9p)

[**Y. Tian** and Y. Zhu, *Better Computer Go Player with Neural Network and Long-term Prediction,* ICLR'16]

MIT Technology Review

Intelligent Machines

How Facebook's AI Researchers Built a Game-Changing Go Engine

The best human players easily beat the best computer-based Go engines. That looks set to change thanks to a new approach pioneered by Facebook's artificial intelligence researchers.

by **Emerging Technology from the arXiv** Dec 4, 2015

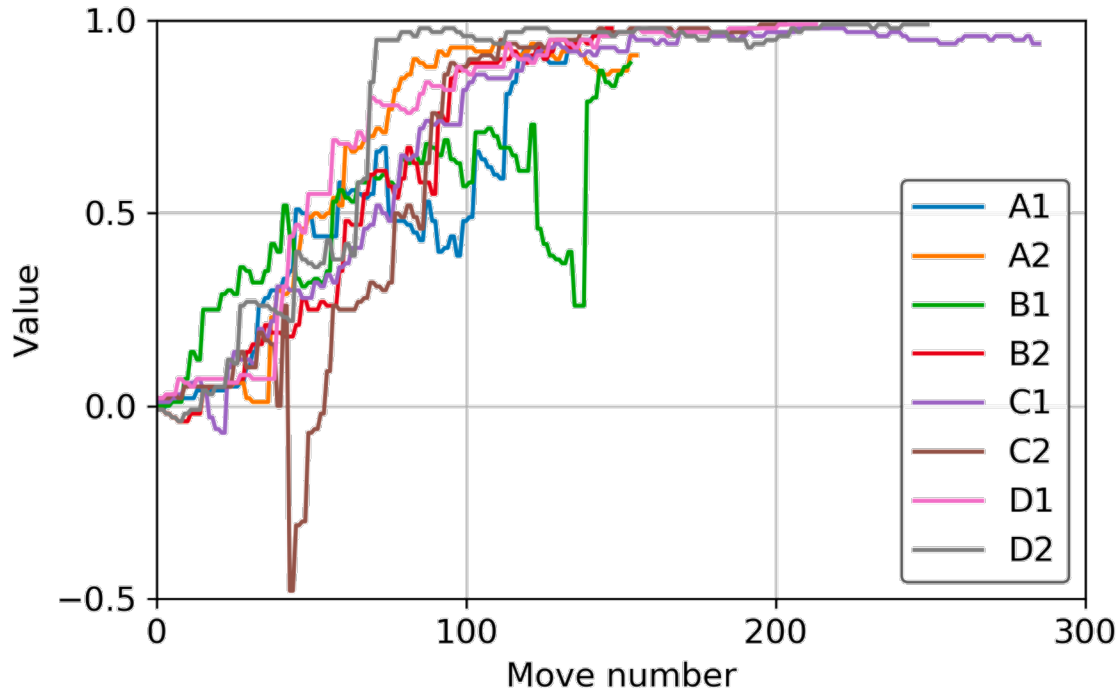
One of the last bastions of human mastery over computers is the game of Go —the best human players beat the best Go engines with ease.

ELF OpenGo

Vs top professional players

Name (rank)	ELO (world rank)	Result
Kim Ji-seok	3590 (#3)	5-0
Shin Jin-seo	3570 (#5)	5-0
Park Yeonghun	3481 (#23)	5-0
Choi Cheolhan	3466 (#30)	5-0

Single V100 GPU, 80k rollouts, 50 seconds
Offer unlimited thinking time for the players

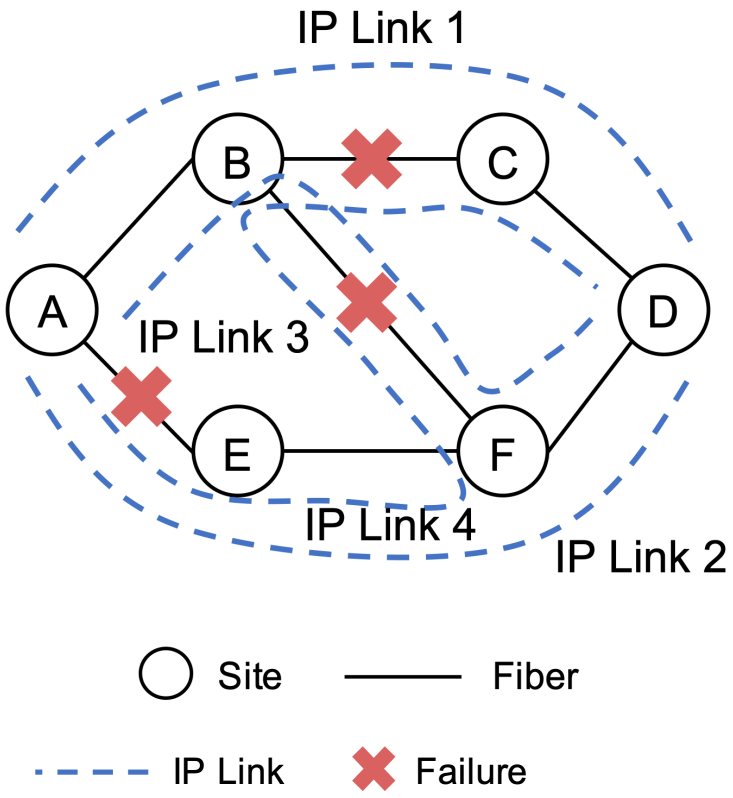


Github 3k+ stars,
Open source data/model/code

[**Y. Tian** et al, *ELF OpenGo: An Analysis and Open Reimplementation of AlphaZero,* ICML 2019]

Network planning

A->D: 100Gbps, under several single-fiber failures



Integer Linear Programming problem

$$\min \sum_{l \in L} (C_l \times cost_{IP} + \sum_{f \in \Psi_l} cost_f) \quad (1)$$

$$\text{s.t.} \sum_{l:l_{src}=n} Y(l,\omega,\lambda) - \sum_{l:l_{dst}=n} Y(l,\omega,\lambda) = Traffic(\omega,n)$$

$$\forall \omega \in \Omega, \lambda \in \Lambda \quad (2)$$

$$C_l \geq \sum_{\omega} Y(l,\omega,\lambda), \ \forall \lambda \in \Lambda \quad (3)$$

$$\sum_{l \in \Delta_f} C_l \times \phi_{lf} \leq S_f \quad (4)$$

$$C_l \geq C_l^{min} \quad (5)$$

Human-designed heuristics to trade optimality for tractability

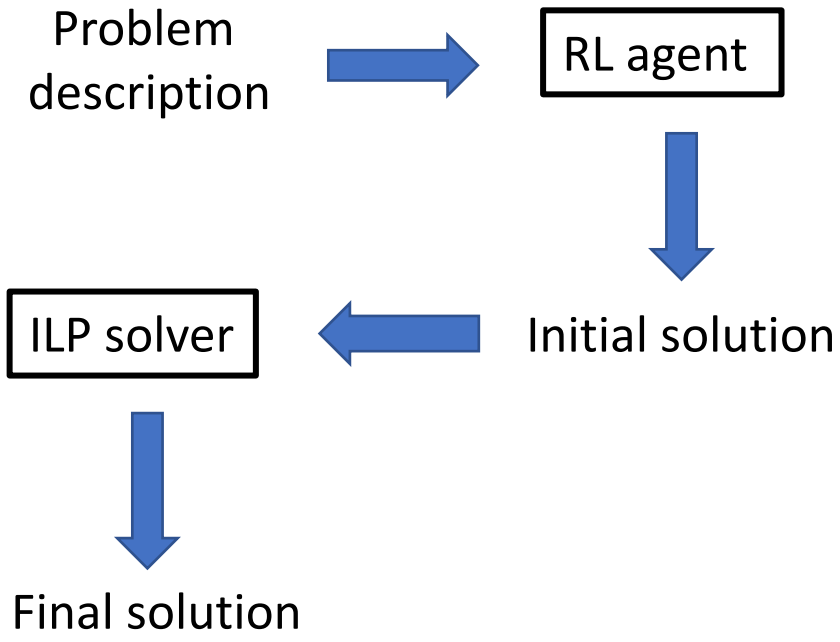
GUROBI OPTIMIZATION

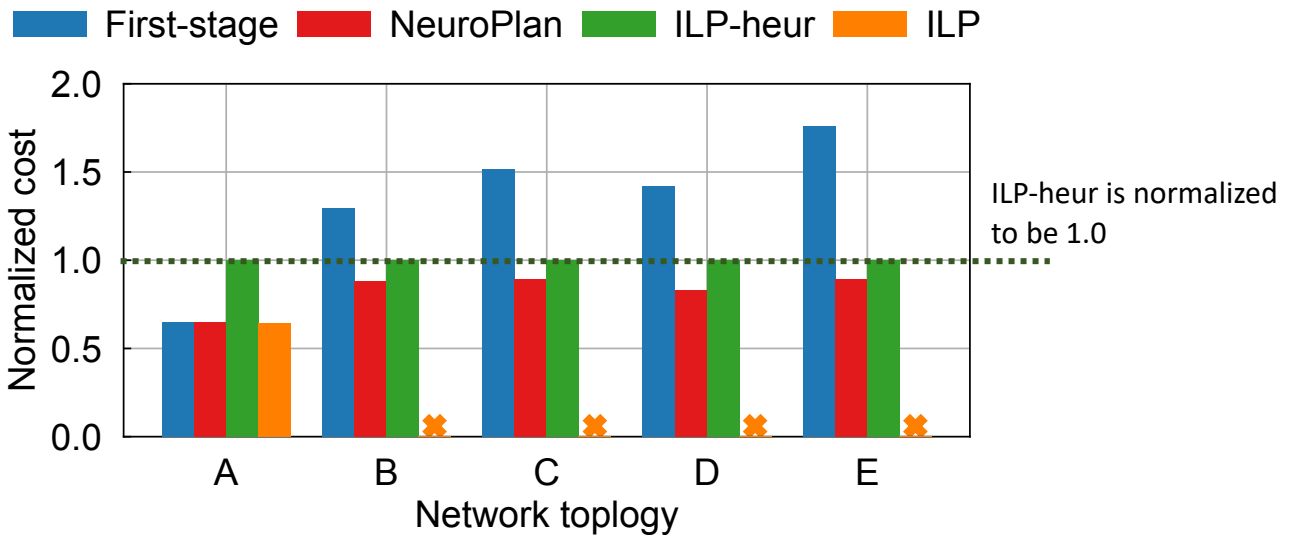
IBM CPLEX

[H. Zhu et al, *Network planning with deep reinforcement learning*, ACM SIGCOMM'21]

Network planning

Two stage approach





NeuroPlan automatically learns good heuristics.

https://github.com/netx-repo/neuroplan

[H. Zhu et al, *Network planning with deep reinforcement learning*, ACM SIGCOMM'21]

SurCo: Optimizing Nonlinearity with a Linear Surrogate

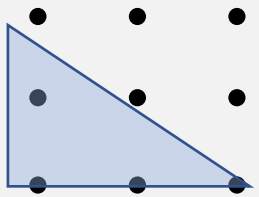
Idea: Learn a MILP objective whose optimal x^* solves the nonlinear problem

Originally

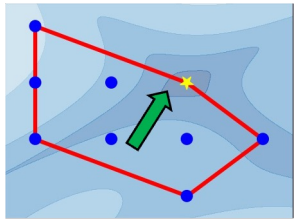
Nonlinear optimization with combinatorial constraints

$$\min_{x} f(x; y)$$

$$\text{s.t } x \in \Omega =$$



combinatorial constraints



Predict surrogate cost $c = c(y)$

Now

Surrogate optimization

$$x^*(y) = \operatorname{argmin}_{x} c(y)^T x$$

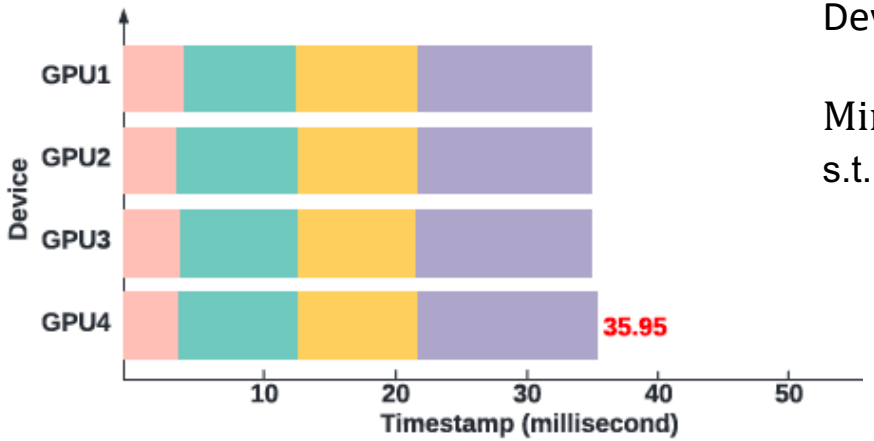
$$\text{s.t } x \in \Omega$$

solved by existing combinatorial solvers

$x^*(y)$ optimizes $f(x; y)$ as much as possible

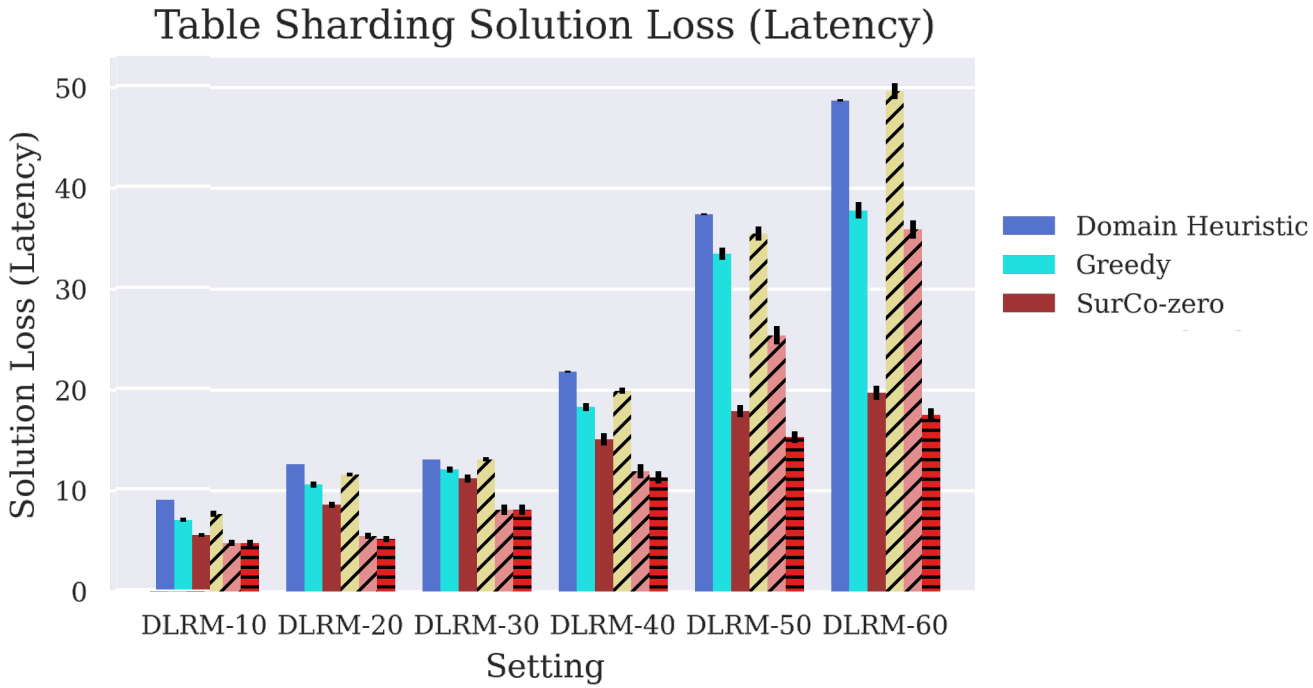
[A. Ferber, …, **Y. Tian**, *SurCo: Learning Linear Surrogates For Combinatorial Nonlinear Optimization Problems*, ICML'23 and and **outstanding paper** in SODS workshop]

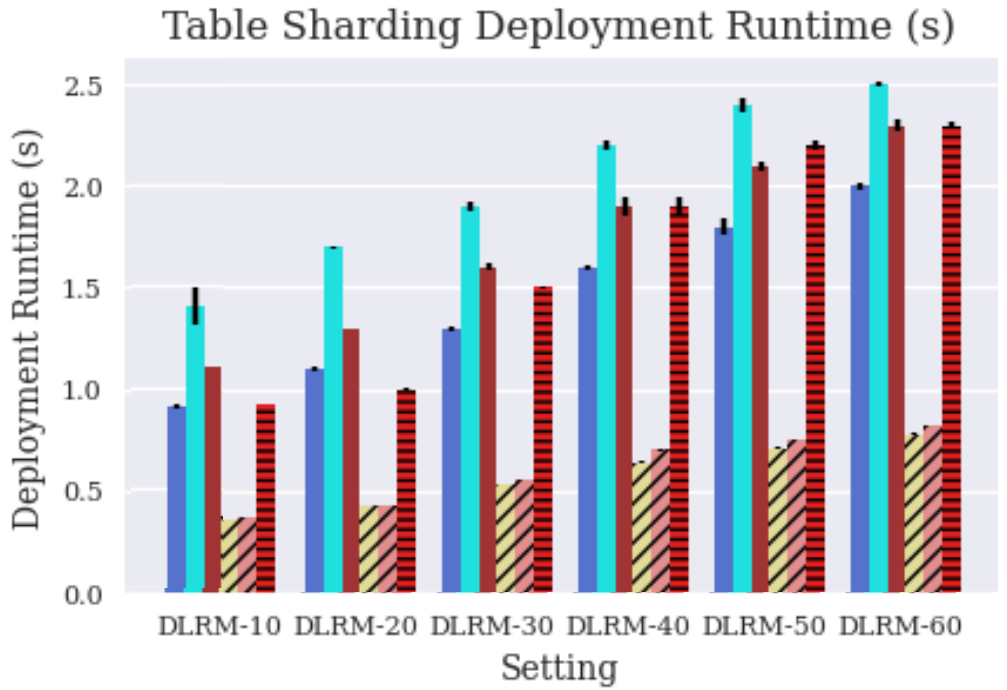
Table Sharding



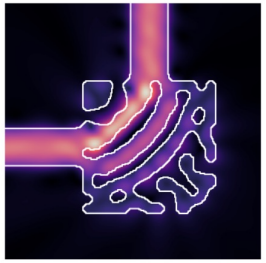
k tables, n identical devices
Table i has memory requirement m_i
Device j has memory capacity M_j

$\mathrm{Min}_x \; \boldsymbol{L}(\{x_{ij}\})$
s.t. $\sum_i x_{ij} m_i \le M_j,\ \sum_j x_{ij} = 1,\ x_{ij} \in \{0,1\}$

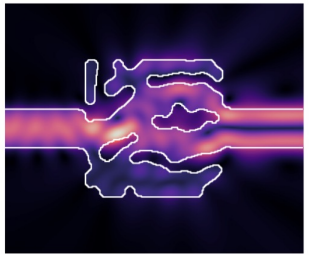




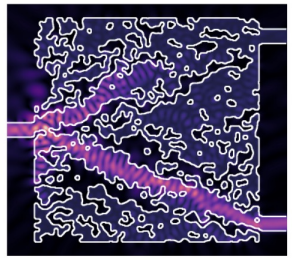
Inverse Photonics Design



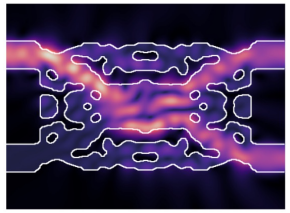
Waveguide bend



Mode converter



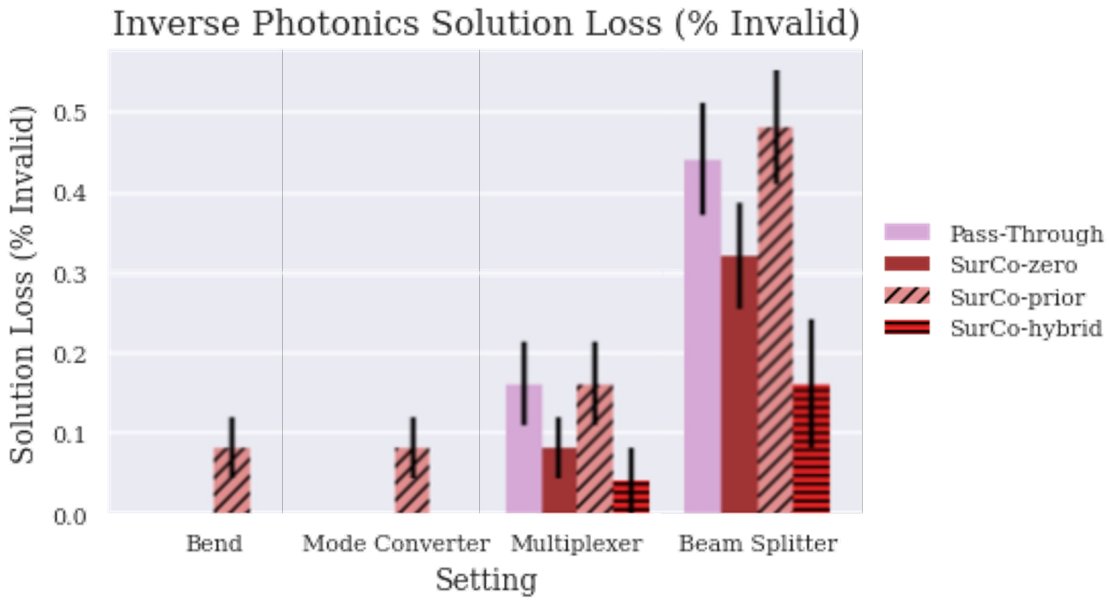
Wavelength division multiplexer

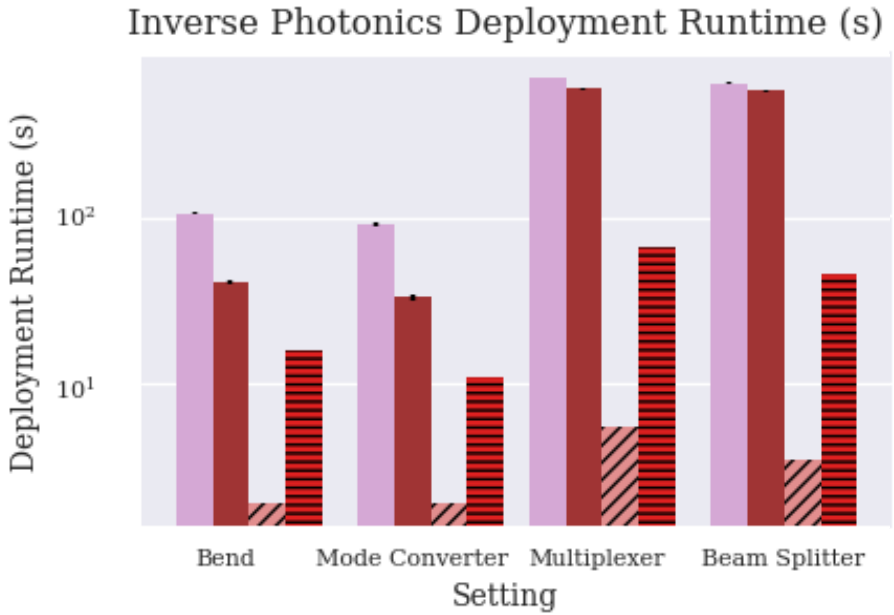


Beam splitter

Nonlinear Loss function:

$$\mathcal{L}(S) = \left(\left|\left| \text{softplus}\left(g \frac{|S|^2 - |S_{\text{cutoff}}|^2}{\min(w_{\text{valid}})} \right) \right|\right|_2 \right)^2$$

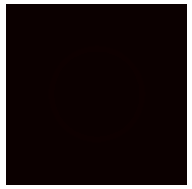




Surrogate Models for linear PDE

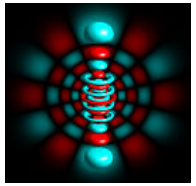
$$\frac{\partial^n \psi}{\partial t^n} = F(\psi, \nabla_{\boldsymbol{x}}\psi, \dots; \boldsymbol{h})$$

- $\psi(x,t)$ is the spatial-temporal signal under time evolutions.
- F is a linear function with respect to ψ and its derivatives
- $\boldsymbol{h}$ is design choice.



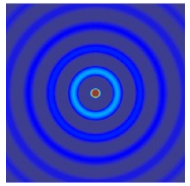
$$\frac{\partial \psi}{\partial t} = \nabla^2 \psi$$

Heat equation



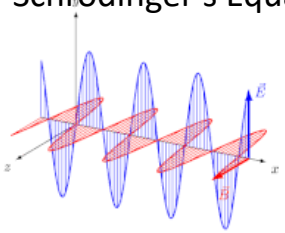
$$i\hbar \frac{\partial \psi}{\partial t} = \left[-\frac{\hbar^2}{2m}\nabla^2 + V\right]\psi$$

Schrodinger's Equation



$$\frac{\partial^2 \psi}{\partial t^2} = c^2 \nabla^2 \psi$$

Wave equation



$$\nabla \cdot E = \frac{\rho}{\epsilon_0}, \nabla \cdot B = 0$$

$$\nabla \times E = -\frac{\partial B}{\partial t}, \nabla \times B = \mu_0 j + \frac{1}{c^2}\frac{\partial E}{\partial t}$$

Maxwell's equation

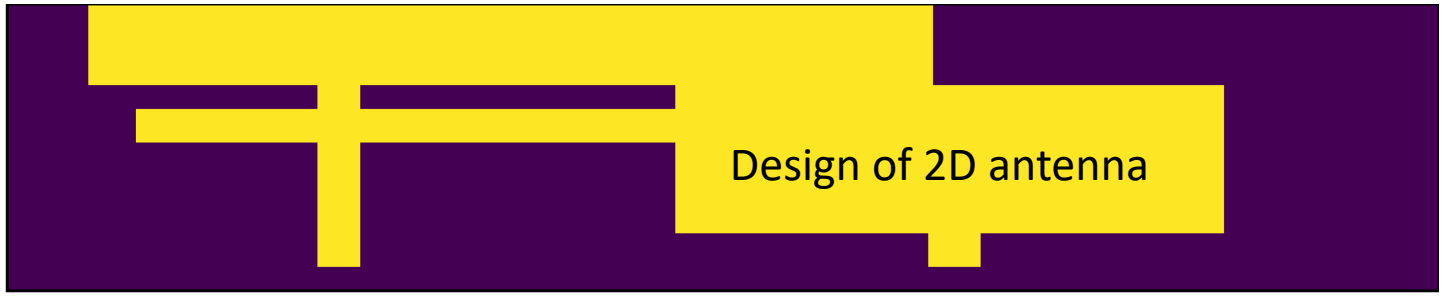
Tricky to simulate accurately and efficiently → Can we do better?

[A. Cohen*, …, **Y. Tian***, *Modeling Scattering Coefficients using Self-Attentive Complex Polynomials with Image-based Representation, AI4Science Caltech Workshop 2023]*

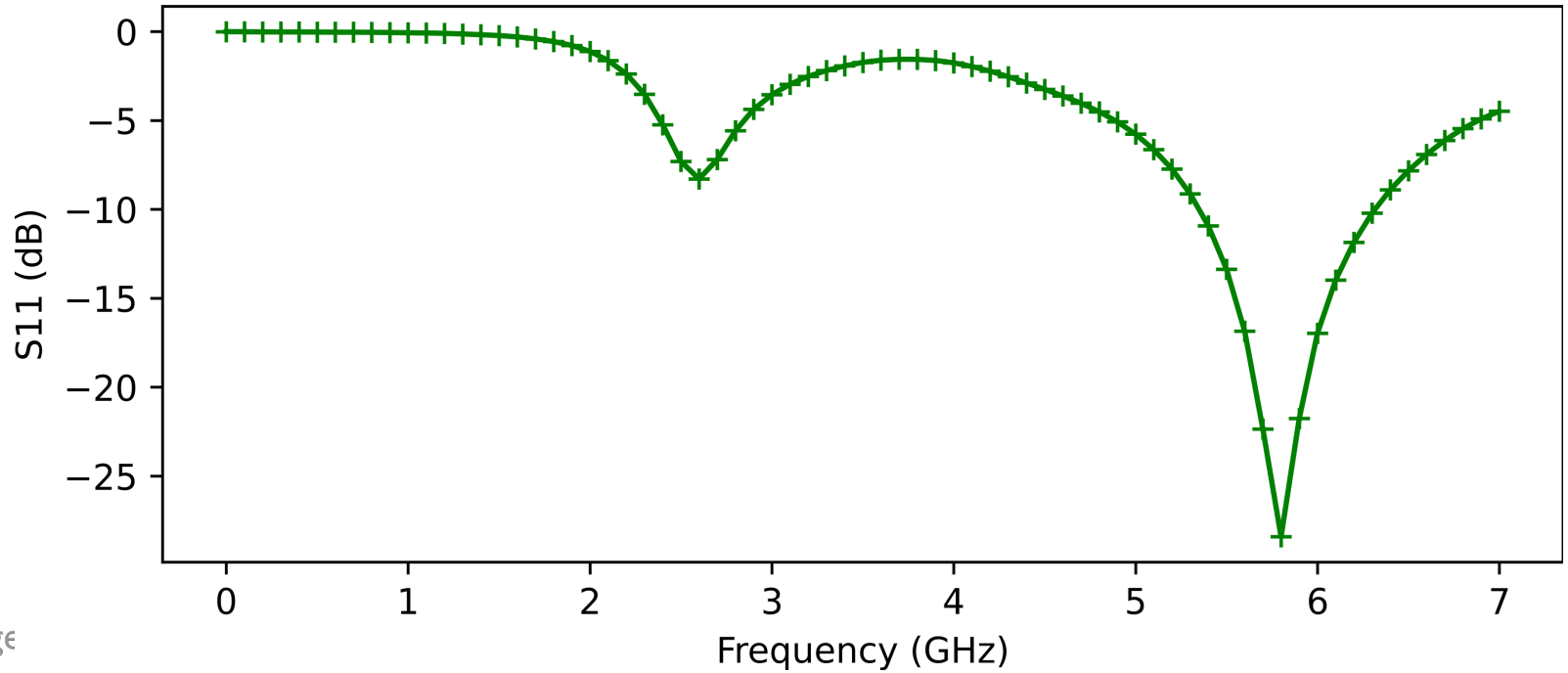
Antenna Design problem

Goal:
find the right design to achieve the right frequency response

$\boldsymbol{h} =$



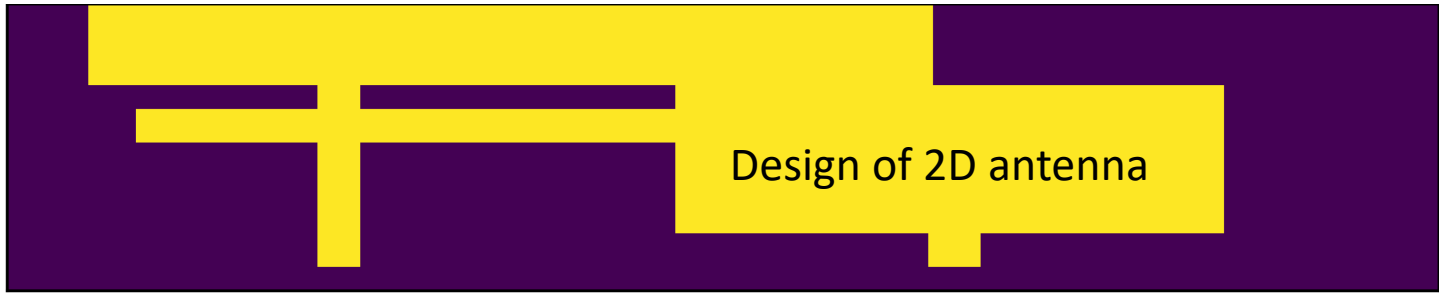
$S_{11}(\psi) =$



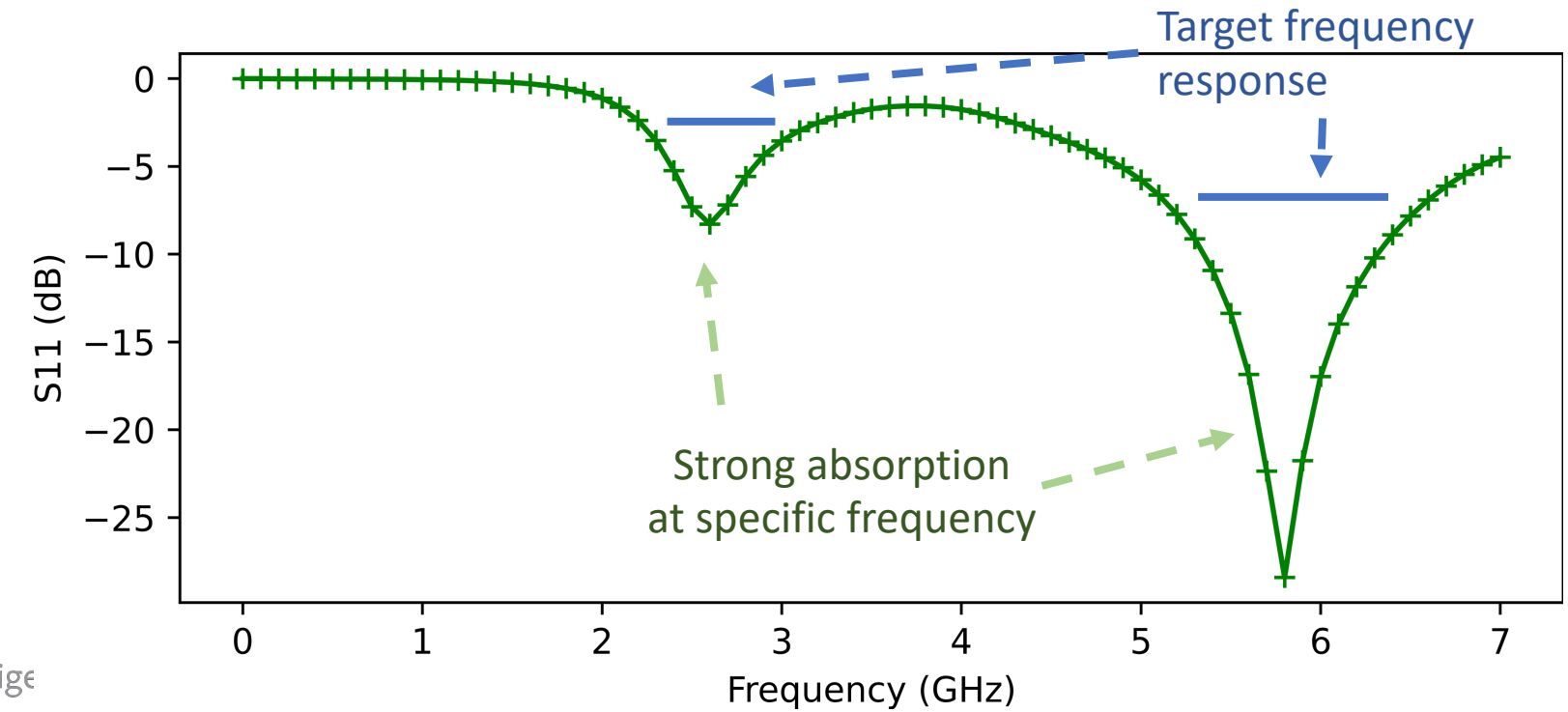
Antenna Design problem

Goal:
find the right design to achieve the right frequency response

$\boldsymbol{h} =$



$S_{11}(\psi) =$



Parametric formula for Linear PDEs

Theorem: For any linear coefficients $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$:

$$\frac{\boldsymbol{b}_1^T \widehat{\boldsymbol{\phi}}(\omega)}{\boldsymbol{b}_2^T \widehat{\boldsymbol{\phi}}(\omega)} = c_0(\boldsymbol{h}) \prod_{k=1}^{K_1} (\omega - \mathbf{z}_k(\boldsymbol{h})) \prod_{k=1}^{K_2} (\omega - \mathbf{p}_k(\boldsymbol{h}))^{-1}$$

where the constant $c_0(\boldsymbol{h})$, zeros $z_k(\boldsymbol{h})$ and poles $p_k(\boldsymbol{h})$ are complex functions of the design choice $\boldsymbol{h}$

For Antenna Design:

The *Scattering Coefficients* $S_{11}(\omega)$:

$$S_{11}(\omega) = \frac{Z_{\text{in}}(\omega) - Z_0}{Z_{\text{in}}(\omega) + Z_0}$$

Impedance $\mathrm{Z}(\omega) \coloneqq V(\omega)/I(\omega)$

Voltage (in Fourier domain)

Current (in Fourier domain)

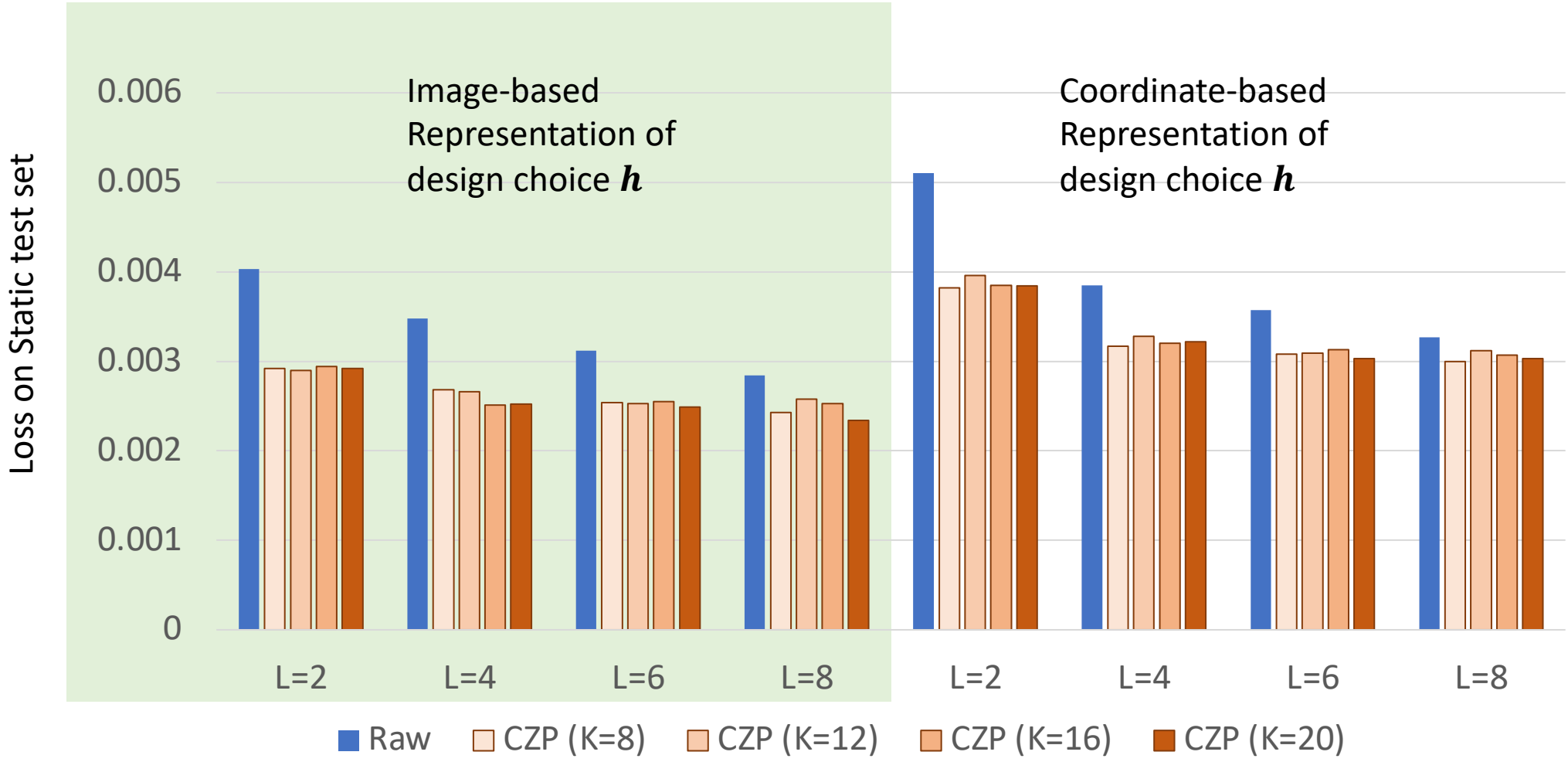
Both are linear function w.r.t. signal $\widehat{\boldsymbol{\phi}}(\omega)$

Final Parametric Form of the scattering coefficients $S_{11}(\omega)$:

$$\log|S_{11}(\omega)| = \log|\mathbf{c}_o(\boldsymbol{h})| + \sum_{k=1}^{K} \log \frac{|\omega - \boldsymbol{z}_k(\boldsymbol{h})|}{|\omega - \boldsymbol{p}_k(\boldsymbol{h})|}$$

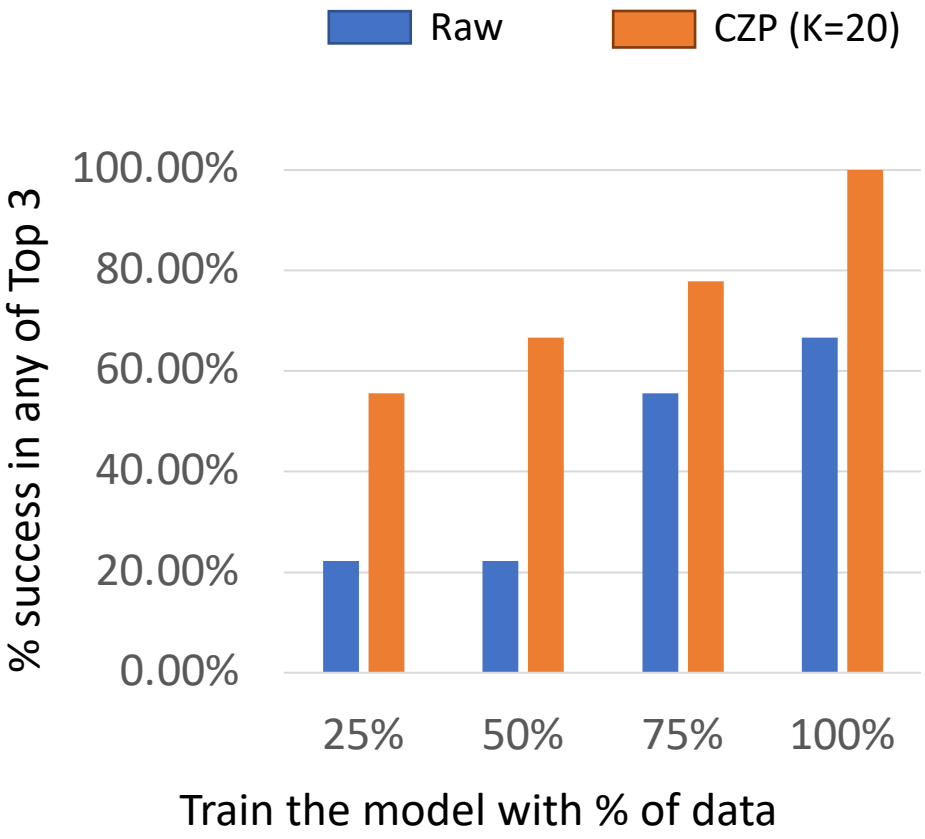
[A. Cohen*, …, **Y. Tian***, *Modeling Scattering Coefficients using Self-Attentive Complex Polynomials with Image-based Representation, AI4Science Caltech Workshop 2023]*

Off-policy Evaluation: Surrogate Model Test Loss

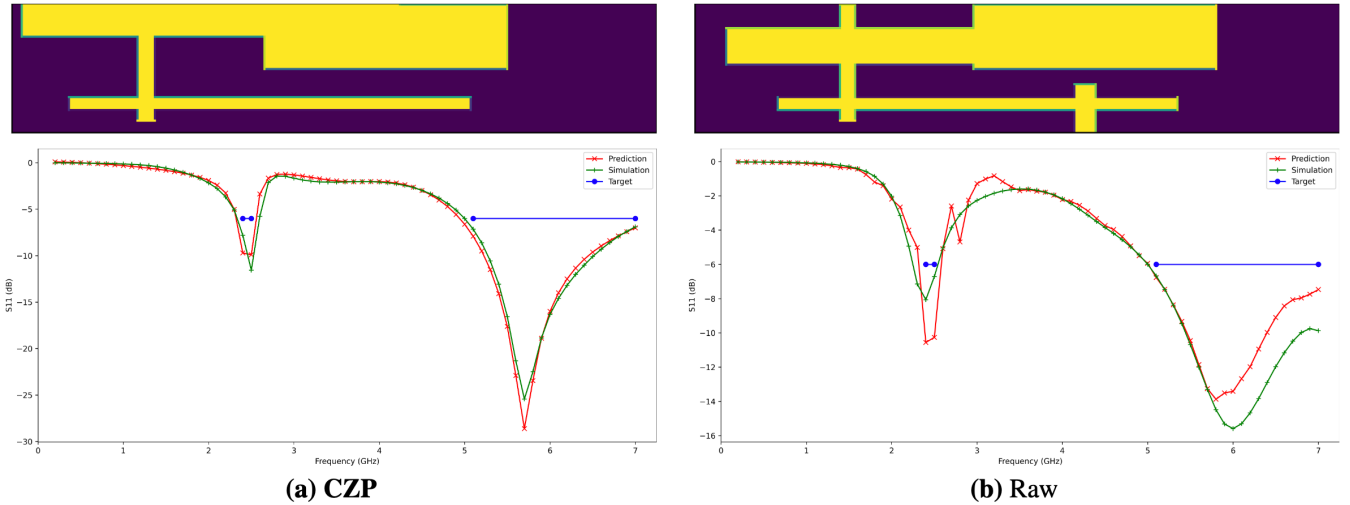


Online Evaluation: CZP model with Search

Goal: to find a solution to satisfy the frequency constraints (verified with CST)

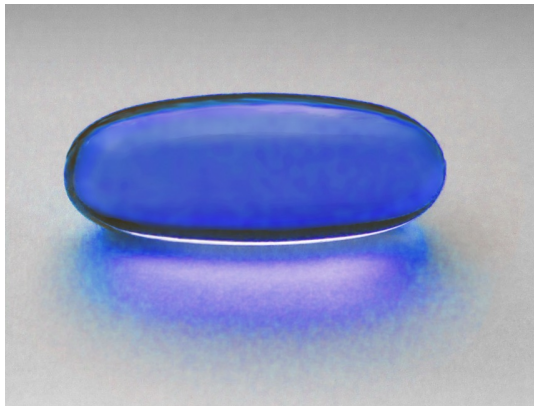


Our CZP model captures the smooth structure of scattering coefficients $S_{11}(\omega)$



(a) CZP (b) Raw

As a sci-fi novelist, …



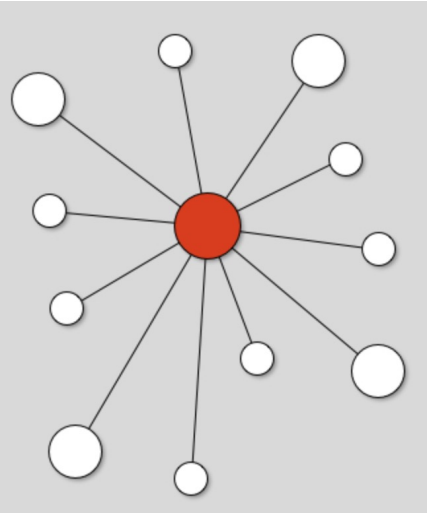
Centralized future

Decentralized future

Why centralization does not work?

- Infra barriers
  - Safety
  - Cost
  - Bandwidth
  - Latency (upper limit: speed of light!)
  - Privacy

- Other factors
  - Monopoly suppresses competition.
  - Bureaucracy



What happens to a Decentralized future?

- Data
  - Data become precious and private.
  - Individual demonstrates their values via unique experience.
    - E.g., Unique story to share, special skills to help others, etc.

- Model
  - Localized, private, specialized and personalized models for each person.
  - Trained with personalized data / experience.

- How to properly assign credits for human uniqueness?

AI-guided Design for Assistive Agents

- Dream Goals:
  - “Talk with the publisher, and make a deal for my novel”
  - “Book a 3-day round trip to Princeton. Make sure I have time to explore NYC”
  - “Read a handbook of the car, and fix the engine light”

- Milestones
  - Make Transformer better at reasoning / planning [Searchformer, Lehert et al.]
  - Personalized model to evaluate user preference [PerSE, D. Wang et al]
  - Leveraging LLMs as a tool
    - Travel planning [J. Xie et al, Travelplanner, arXiv’24]
    - Story Generation [Re3, K. Yang et al, EMNLP’23] [DOC, K. Yang et al, ACL’23][E2EStoryGenerator, H. Zhu]

AI-guided Design for Efficient training/inference

- Dream Goals:
  - “My models get updated locally everyday, and give me good suggestions”

- Milestones
  - Long-context fast inference [PI, S. Chen et al][H2O, Z. Zhang et al, NeurIPS’23][StreamingLLM, G. Xiao et al, ICLR’24]
  - Possible to train a 7B model on consumer-grade GPUs [GaLore, J. Zhao et al]
  - Sub-billion (350m) models trained from scratch [MobileLLM, Z. Liu et al]
  - Understand how Transformers learn features from the data [LSTMvsTransformer, H. Shi et al, AAAI’22] [JoMA, Tian et al, ICLR’24]

AI-guided Design for Next Generation Hardware

- Lack of chips!
- Dream Goal:
  - “Self-assembled, personalized chips and computers”
  - “Human beats aliens, not by brave pilots, but by self-replicating nanobots”

- I don’t have milestones... Well, not quite...
  - Inverse photonic design [SurCo, GenCo, A. Ferber et al]
  - Antenna Design [CZP, A. Cohen et al, AI4Science Caltech workshop]

Thanks!