

DeepThink Lab

Think deep, bold, & different



ECE ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY OF MICHIGAN

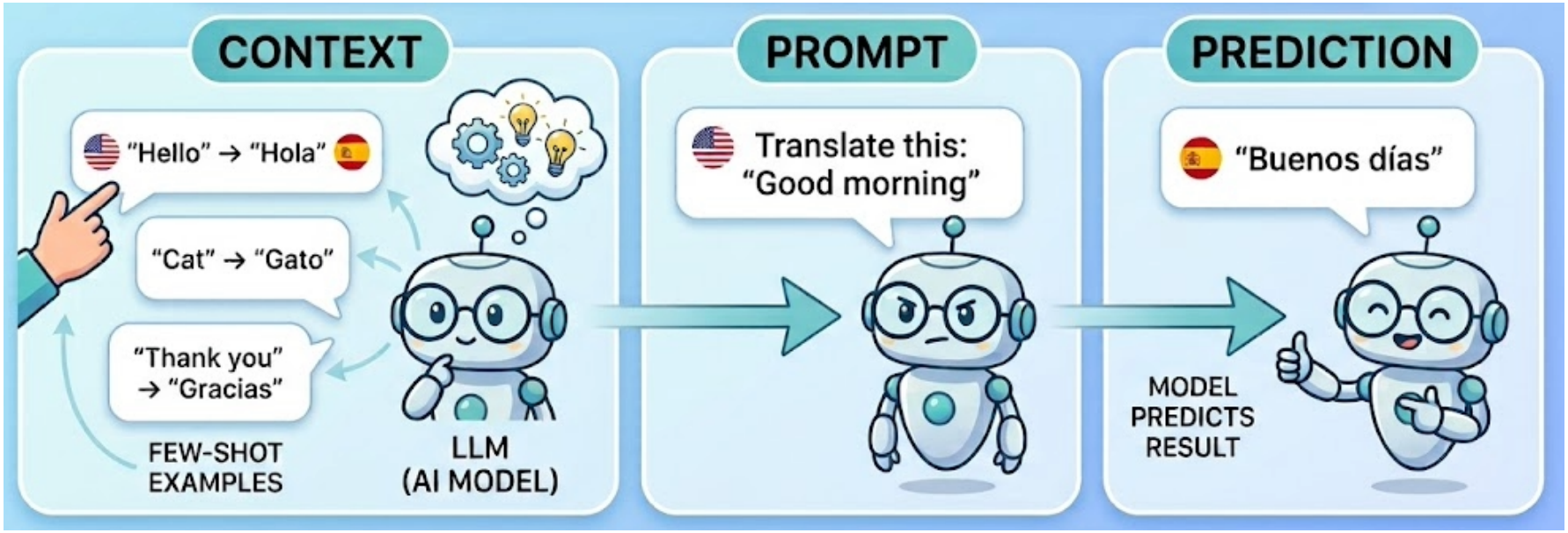
Training Task Diversity Shapes In-Context Learning via Low-Dimensional Subspaces

Qing Qu

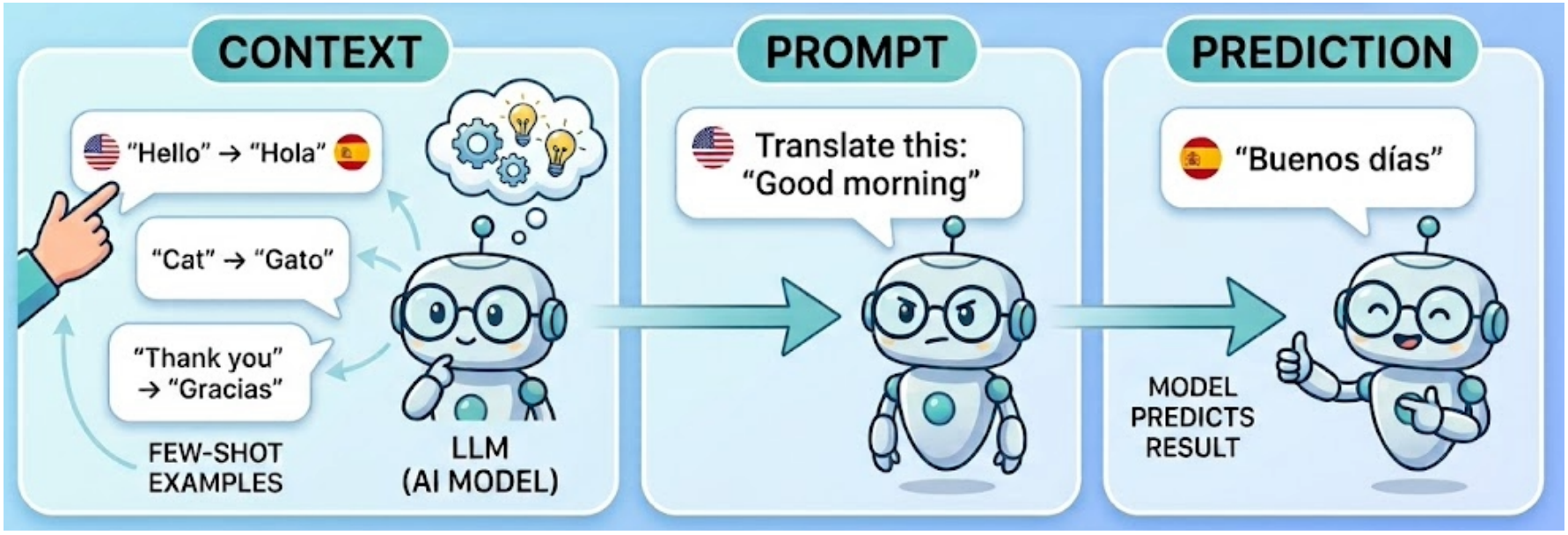
October 16, 2026

EECS, University of Michigan

In-Context Learning in LLMs



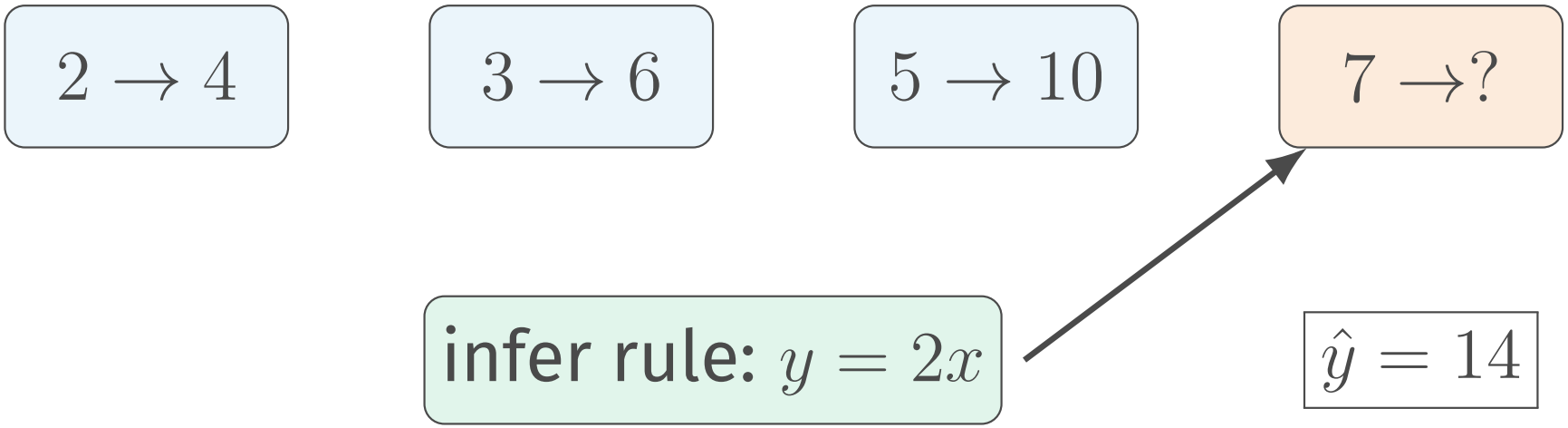
In-Context Learning in LLMs



An LLM can do zero-shot learning from **in-context examples**:

- **Context:** a few input-output examples $(x_1, y_1), \ldots, (x_n, y_n)$
- **Prompt:** a new query x_{query}
- **Prediction:** infer the pattern and provide the answer $\hat{y}$

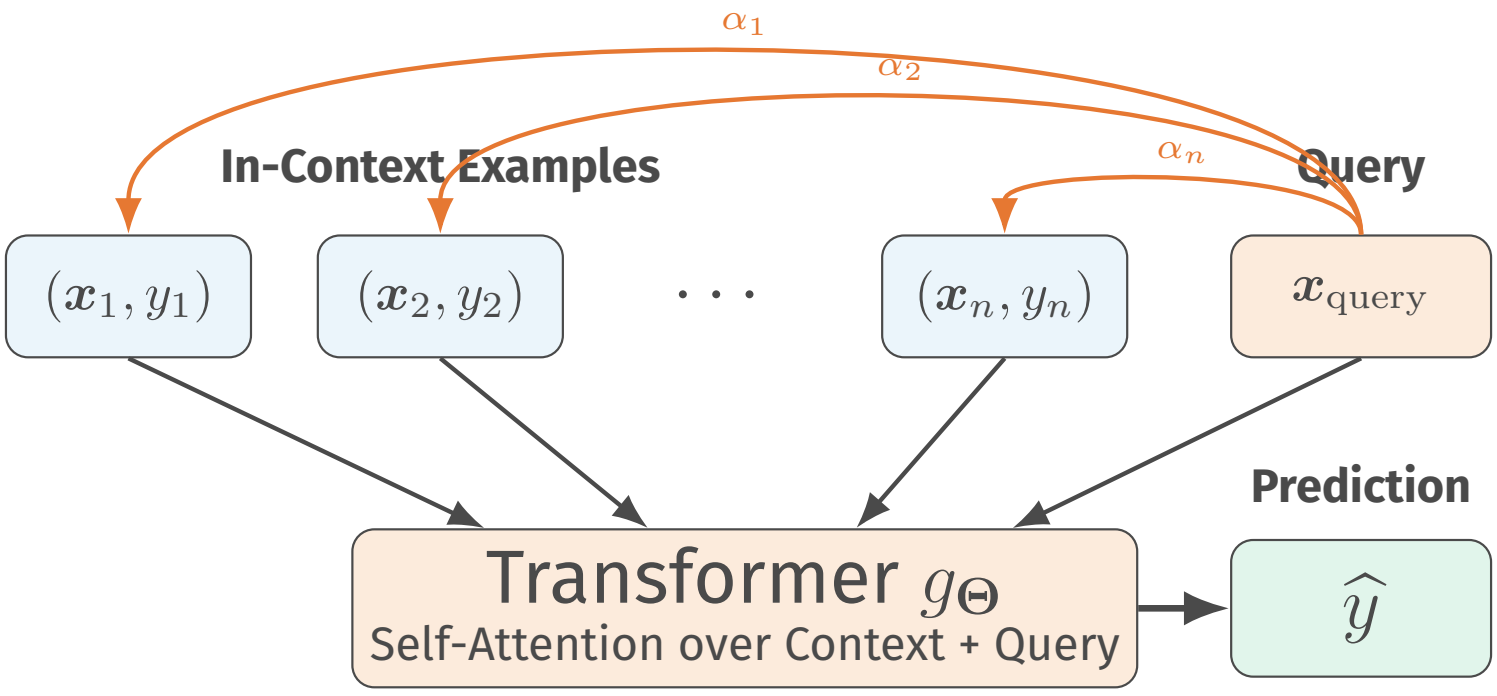
ICL: Infer Rules from In-Context Examples



$$\underbrace{(2,4),(3,6),(5,10)}_{\text{in-context examples}} \quad + \quad \underbrace{x = 7}_{\text{query}} \quad \Rightarrow \quad \underbrace{\widehat{y} = 14}_{\text{prediction}}$$

The model identifies **the rule** of *the task it is solving.*

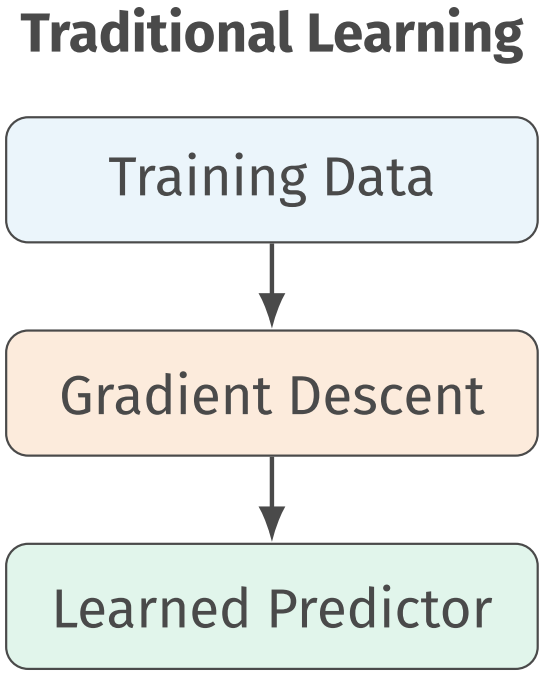
The Basic Setup of In-Context Learning



Self-attention lets the query retrieve and combine information from the in-context examples:

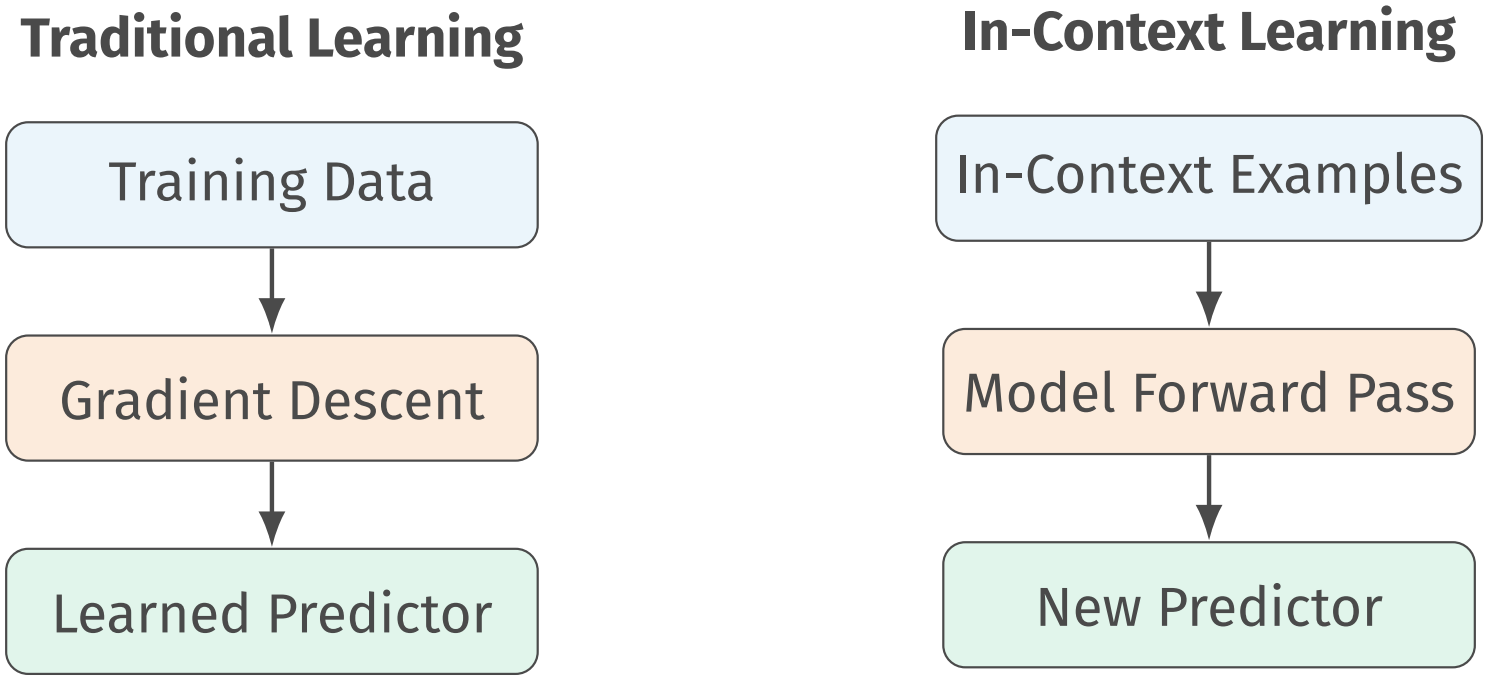
$$\hat{y} = g_\Theta\left((x_1, y_1), \ldots, (x_n, y_n), x_{\text{query}}\right)$$

Traditional Learning vs. In-Context Learning



$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t).$$

Traditional Learning vs. In-Context Learning

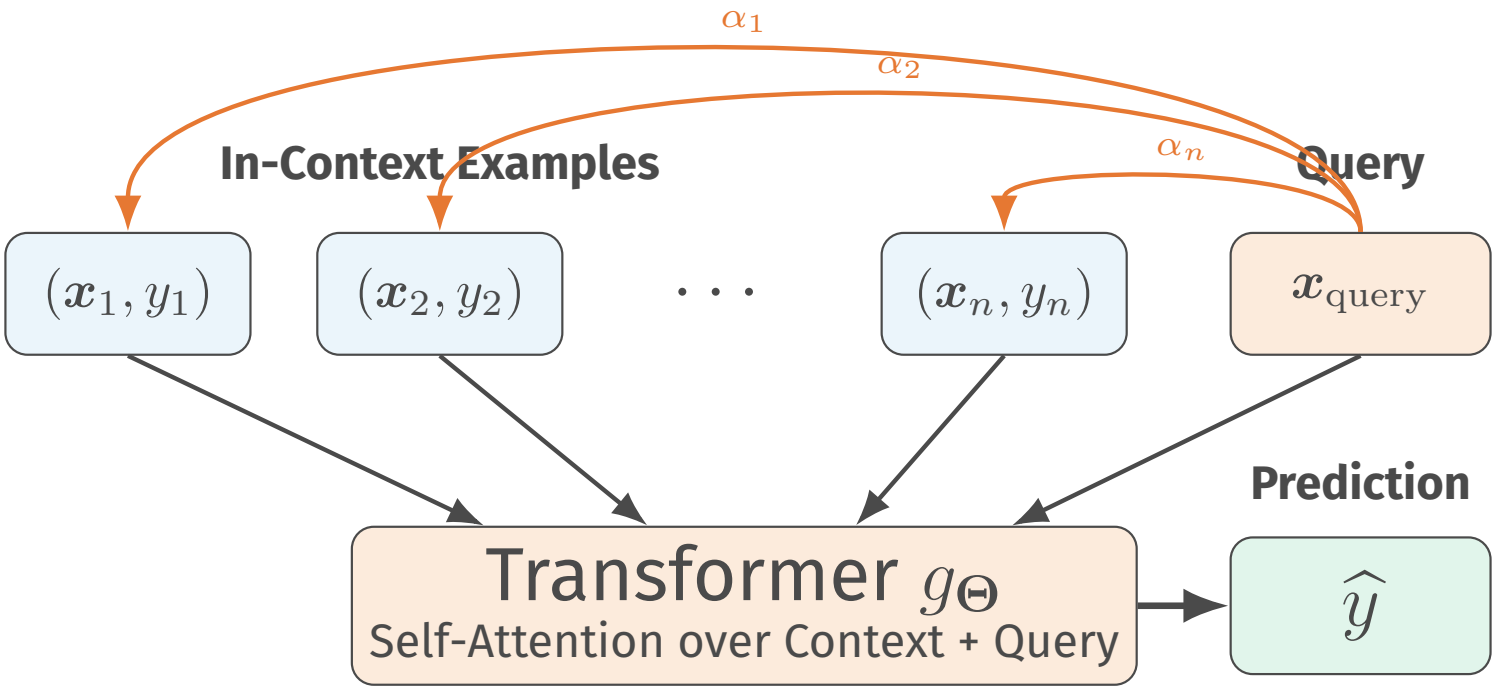


$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t).$$

$$\boxed{\boldsymbol{\theta}_{\text{before}} = \boldsymbol{\theta}_{\text{after}}}$$

The in-context examples act as the training set.

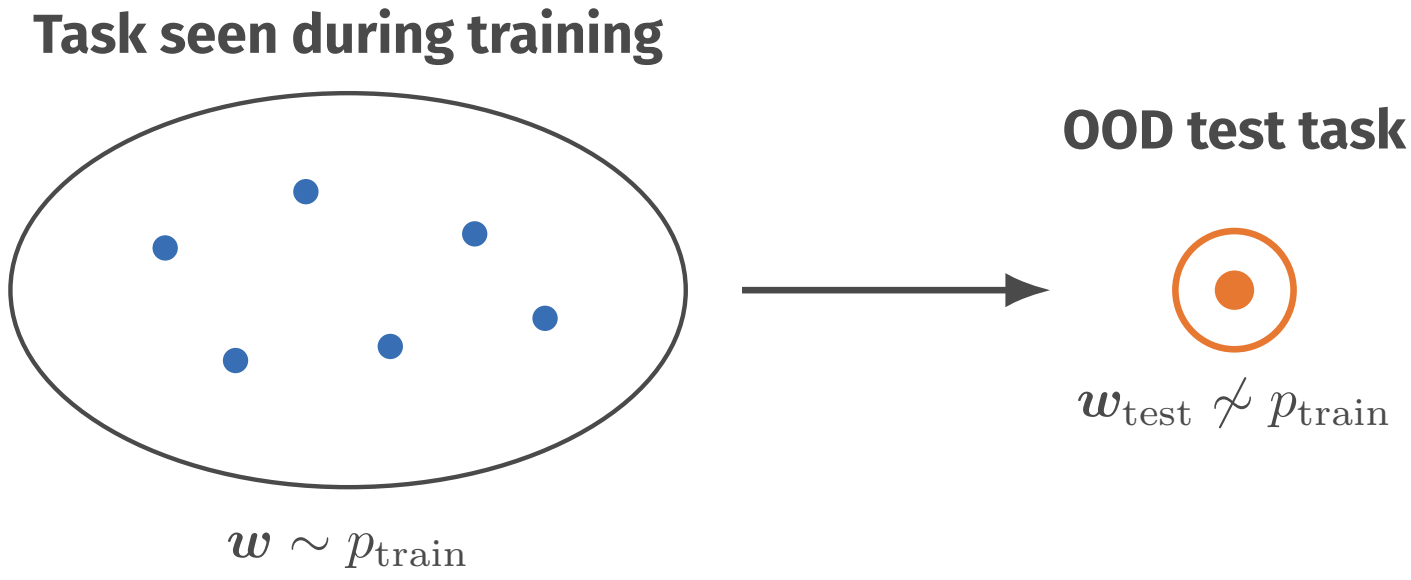
Studies of In-Context Learning in the Literature



Lots of empirical and theoretical studies on ICL:

- Function classes: (Garg et al. 2023; Guo et al. 2024)
- Working mechanism (Oswald et al. 2023; Ahn et al. 2023)
- The effect of task diversity: (Kim et al. 2024; Goddard et al. 2025)

The Out-of-Distribution (OOD) Generalization Ability of ICL

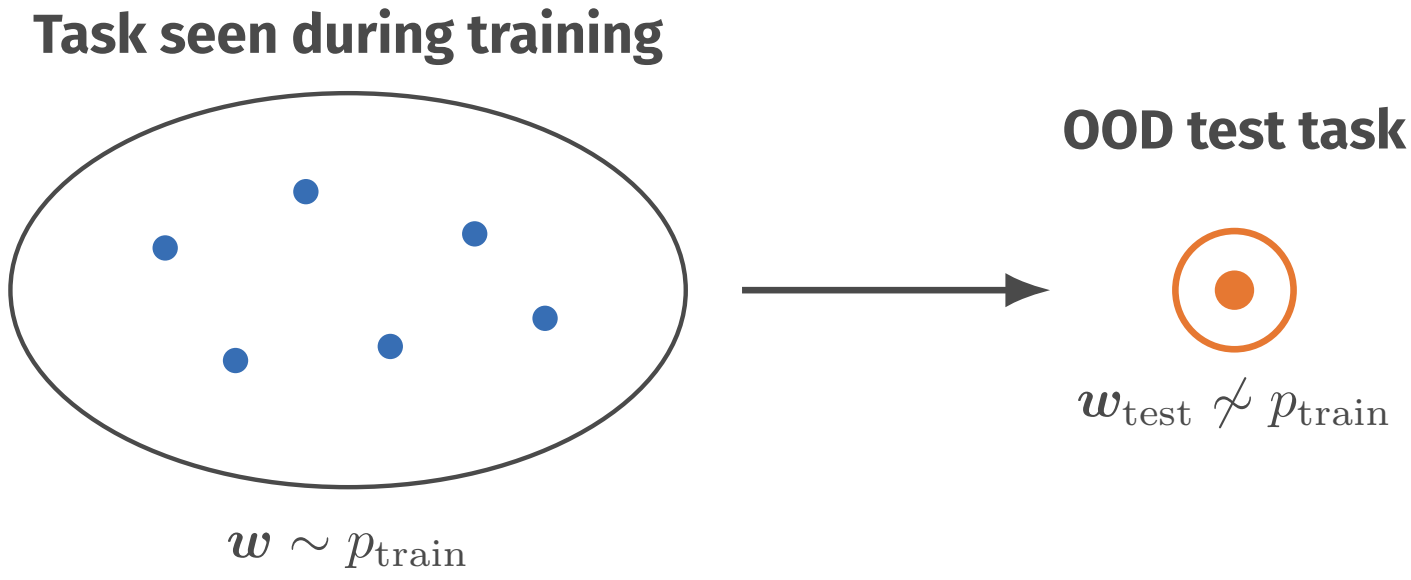


- **Training.** Consider a family of tasks parameterized by w:

$$y = f_w(x), \qquad w \sim p_{\text{train}}(w),$$

where the model sees $w_1, \cdots, w_N \sim p_{\text{train}}(w)$ during training.

The Out-of-Distribution (OOD) Generalization Ability of ICL



- **Training.** Consider a family of tasks parameterized by w:

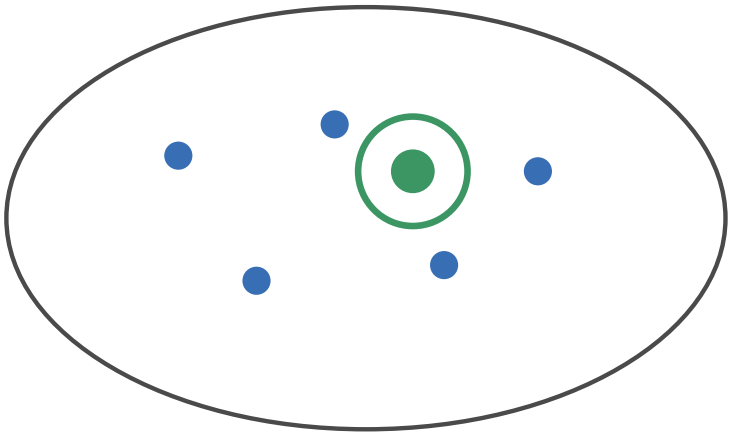
$$y = f_w(x), \qquad w \sim p_{\text{train}}(w),$$

  where the model sees $w_1, \cdots, w_N \sim p_{\text{train}}(w)$ during training.

- **Inference.** Given a few in-context examples, the model can make correct predictions on new tasks, $w_{\text{test}} \not\sim p_{\text{train}}$.

Two Different Notions of Generalization

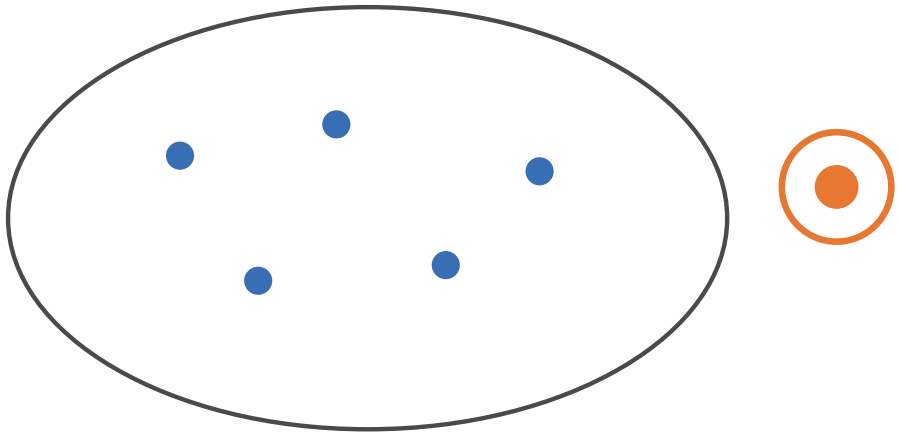
In-Distribution Generalization



$$\boldsymbol{w}_{\text{test}} \sim p_{\text{train}}$$

Test task resembles tasks encountered during training.

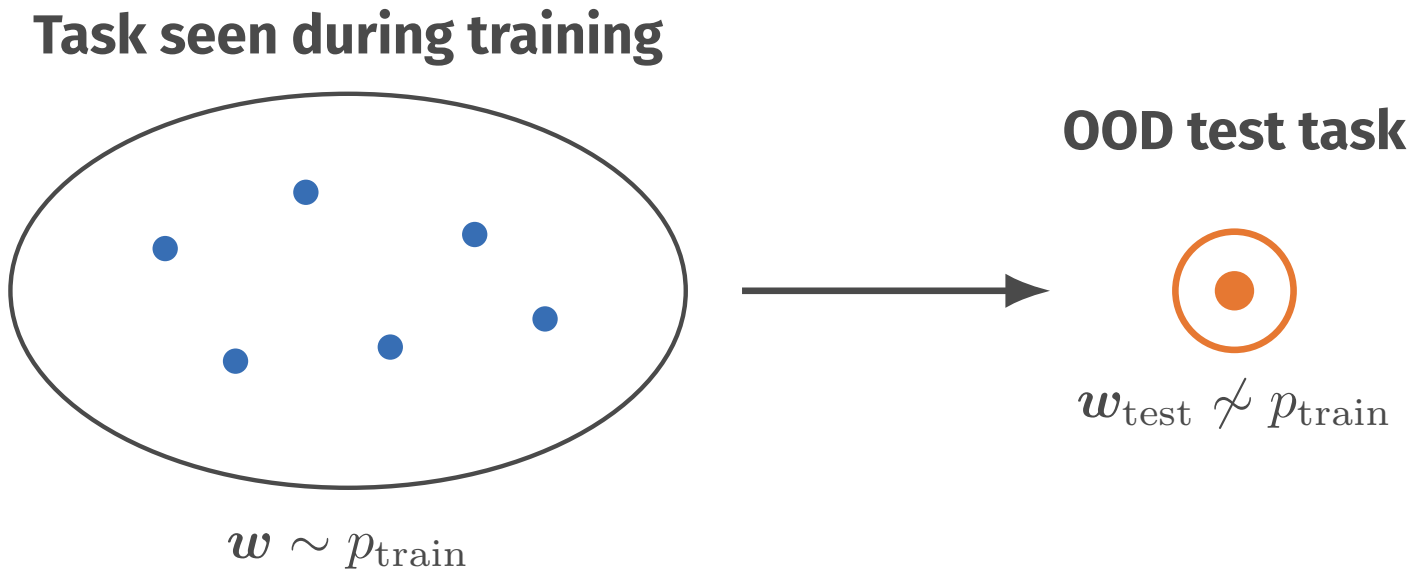
OOD Generalization



$$\boldsymbol{w}_{\text{test}} \not\sim p_{\text{train}}$$

Test task lies outside the training distribution.

In-Context Learning (ICL) & OOD Generalization

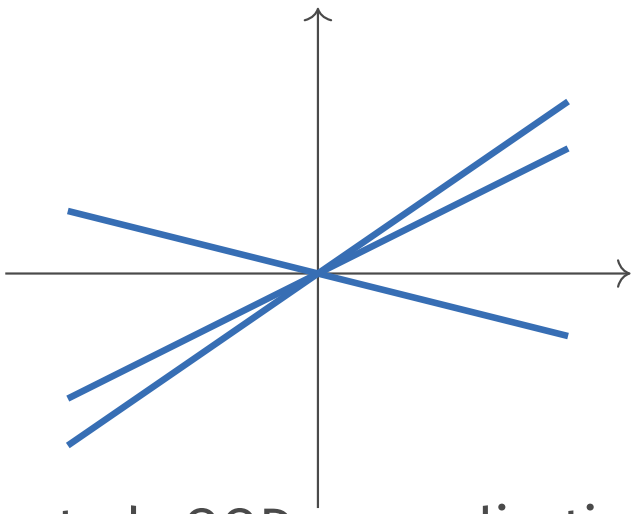


- LLMs can solve OOD tasks by leveraging **in-context examples.**
- Zero-shot learning with **No** weight updates needed!

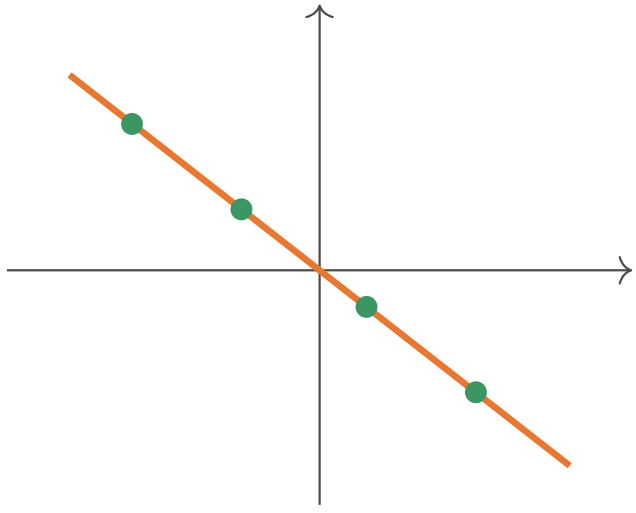
When and why do we achieve OOD generalization via ICL?

Linear Regression: A Simple Testbed[1]

Training: $w \in p_{\text{train}}$



Inference: $w_{\text{test}} \notin p_{\text{train}}$



We study OOD generalization on **linear regression** tasks:

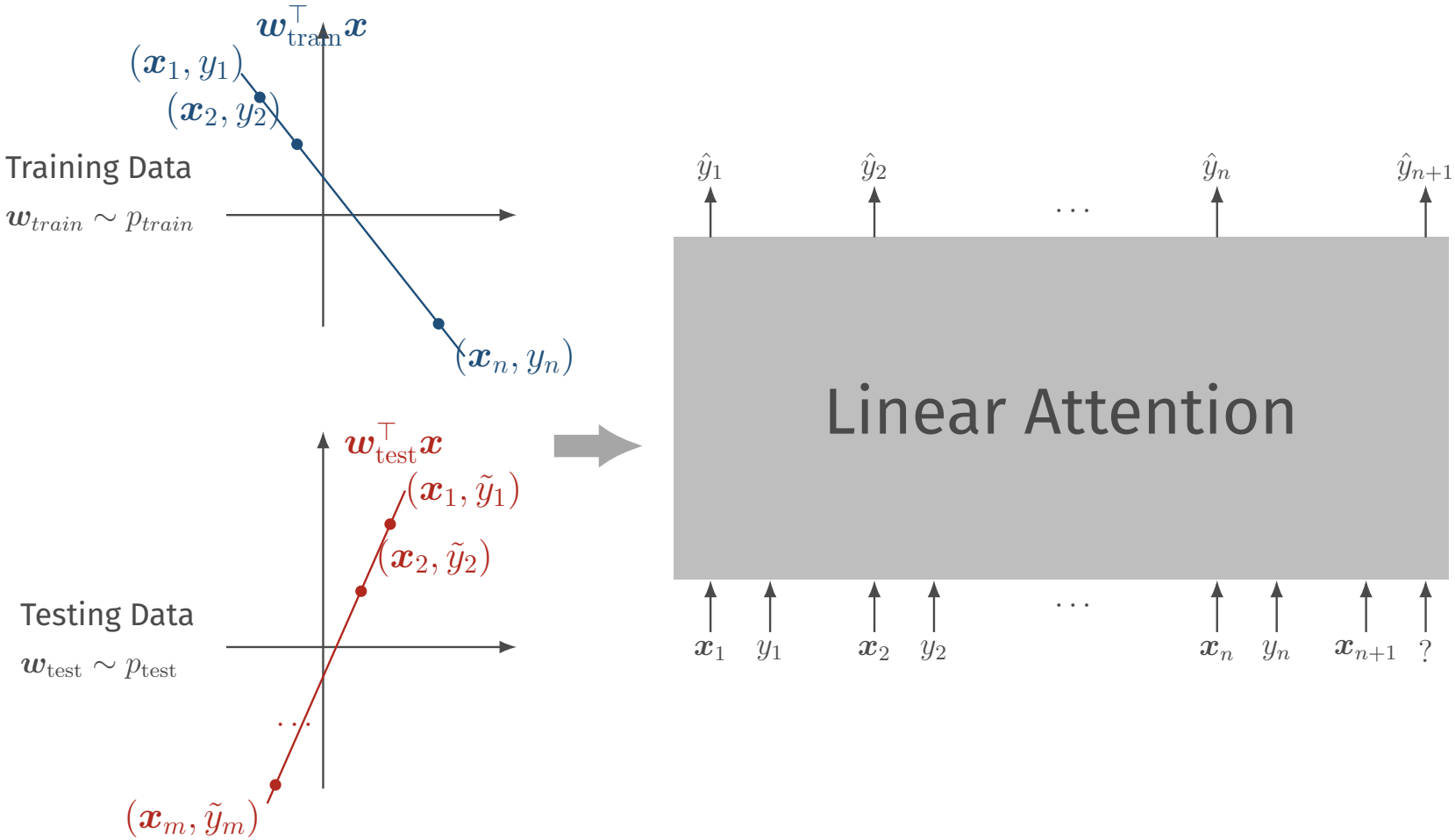
$$y = f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + \xi,$$

where we have

- w is the **task vector** drawn from **low-dimensional subspaces**.
- $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d),\ \xi \sim \mathcal{N}(0, \sigma^2)$.

[1]Zhang et al. Trained Transformers Learn Linear Models In-Context. JMLR, 2024.

Linear Regression: A Simple Testbed



Linear regression problem with $x \sim \mathcal{N}(0_d, I_d)$, $\xi \sim \mathcal{N}(0, \sigma^2)$, and

$$y = w_{\text{train}}^\top x + \xi, \quad \text{and} \quad \tilde{y} = w_{\text{test}}^\top x + \xi,$$

where w_{train} and w_{test} are **task vectors** from different subspaces.

Task Diversity: Union of Low-Dimensional Subspaces

Task Vector Distribution: Mixture of Low-Rank Gaussians

For each linear-regression task

$$y = \boldsymbol{w}^\top \boldsymbol{x} + \xi,$$

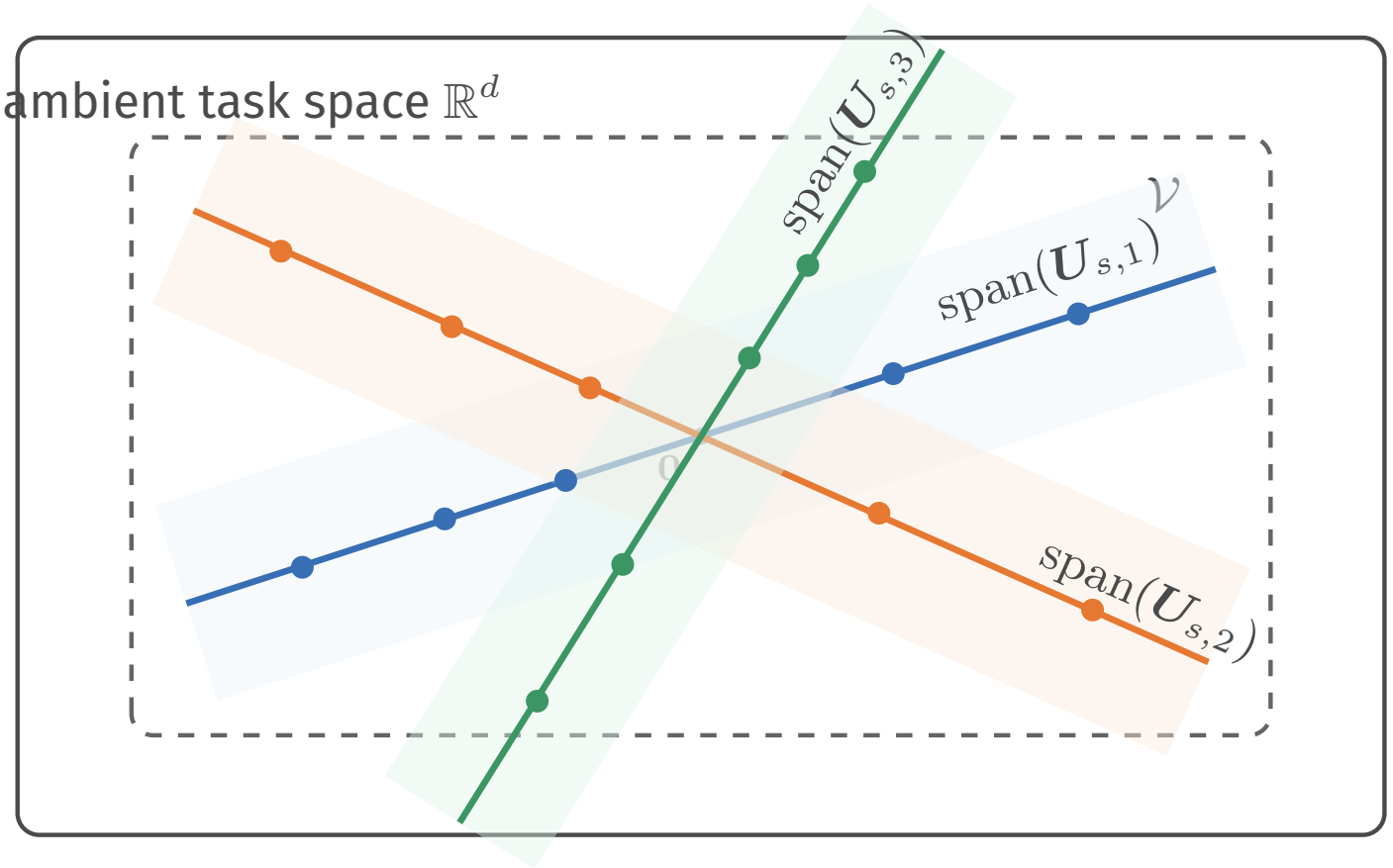
we assume the **task vector** w is drawn from

$$p_{\text{data}}(\boldsymbol{w}) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}\left(\boldsymbol{w}; \mathbf{0}, \boldsymbol{\Sigma}_{s,k}\right), \qquad \boldsymbol{\Sigma}_{s,k} = \boldsymbol{U}_{s,k} \boldsymbol{U}_{s,k}^\top.$$

Here, each **low-rank basis** $U_{s,k}$ captures one task family with

$$\boldsymbol{U}_{s,k} \in \mathbb{R}^{d \times r}, \qquad \boldsymbol{U}_{s,k}^\top \boldsymbol{U}_{s,k} = \boldsymbol{I}_r, \qquad r \ll d.$$

Task Diversity: Union of Low-Dimensional Subspaces

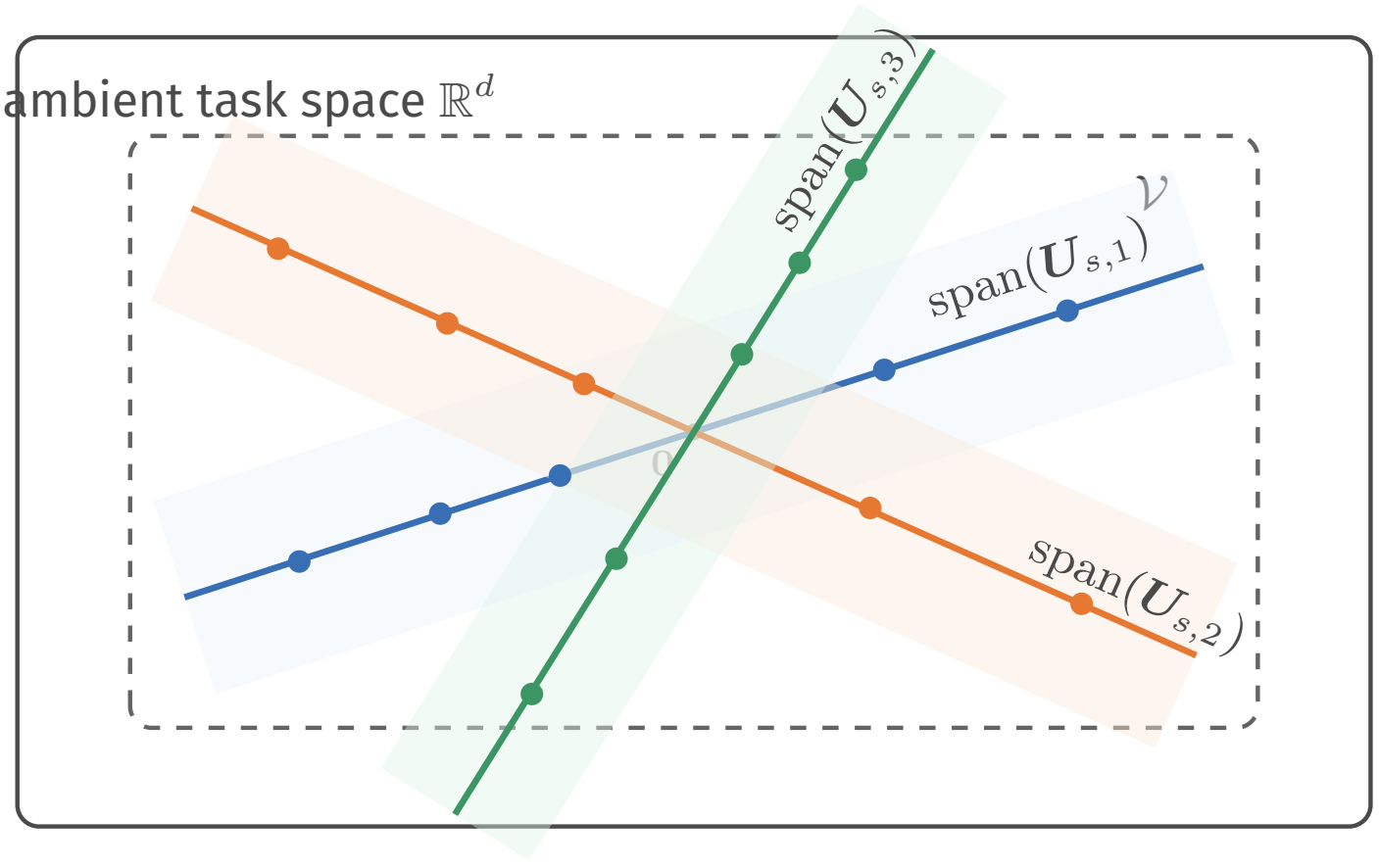


During training phase, we draw the task vector $\boldsymbol{w}_{\mathrm{train}}$ with

$$\boldsymbol{w}_{\mathrm{train}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{s,k}), \text{ with prob. } 1/K, \quad k = 1, \cdots, K,$$

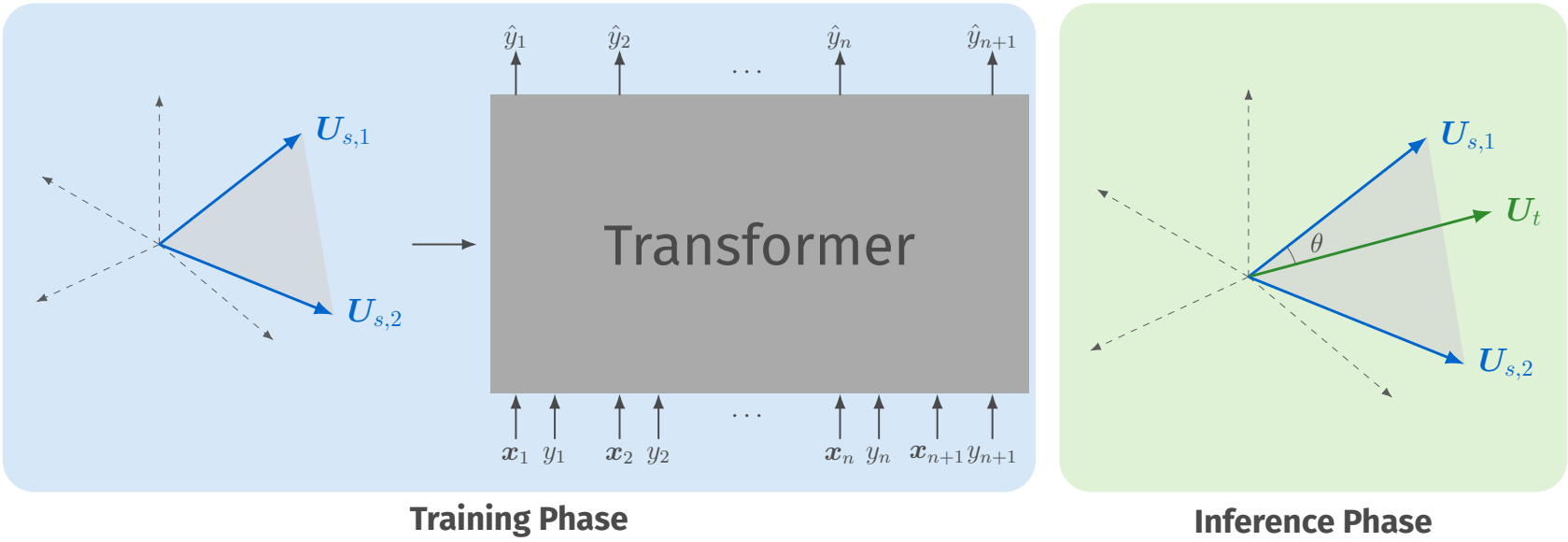
so that $\boldsymbol{w}_{\mathrm{train}} \in \mathcal{V} := \mathrm{span}\left(\begin{bmatrix} \boldsymbol{U}_{s,1} & \dots & \boldsymbol{U}_{s,K} \end{bmatrix}\right)$.

Task Diversity: Union of Low-Dimensional Subspaces



The **task diversity** is captured by (i) **the number of subspaces** K and (ii) **the co-dimension** $n_{\mathrm{com}} = \dim\left(\bigcap_{k=1}^{K} \mathrm{span}(\boldsymbol{U}_{s,k})\right)$.

Task Diversity: Union of Low-Dimensional Subspaces



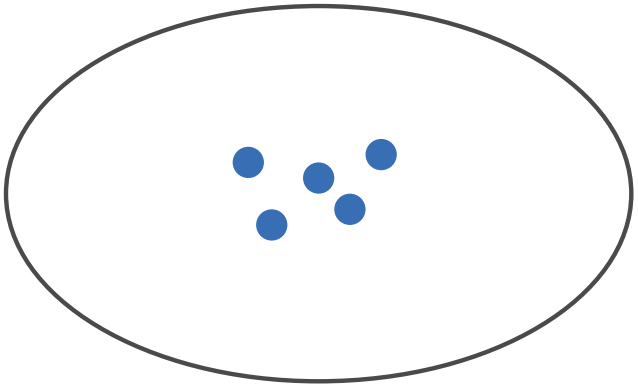
During **inference phase**, we predict on samples drawn from

$$\tilde{y} = \boldsymbol{w}_{\text{test}}^{\top} \boldsymbol{x} + \xi$$

where $\boldsymbol{w}_{\text{test}} \in \text{span}(\boldsymbol{U}_{\text{test}})$ with $\boldsymbol{U}_{\text{test}} \neq \boldsymbol{U}_{\text{train}}$.

Claim I: Training Task Diversity Matters for OOD Generalization

Low task diversity

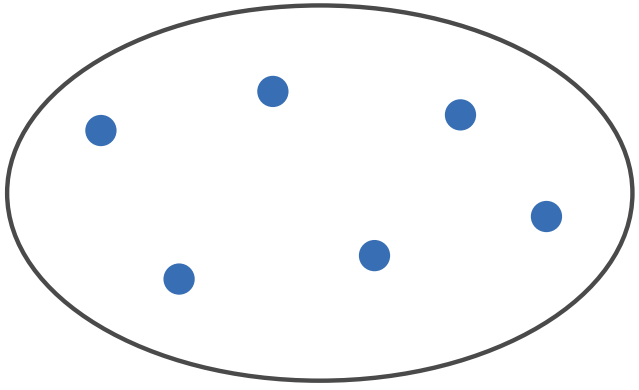


narrow task coverage

⇓

Leads to **task memorization**

High task diversity



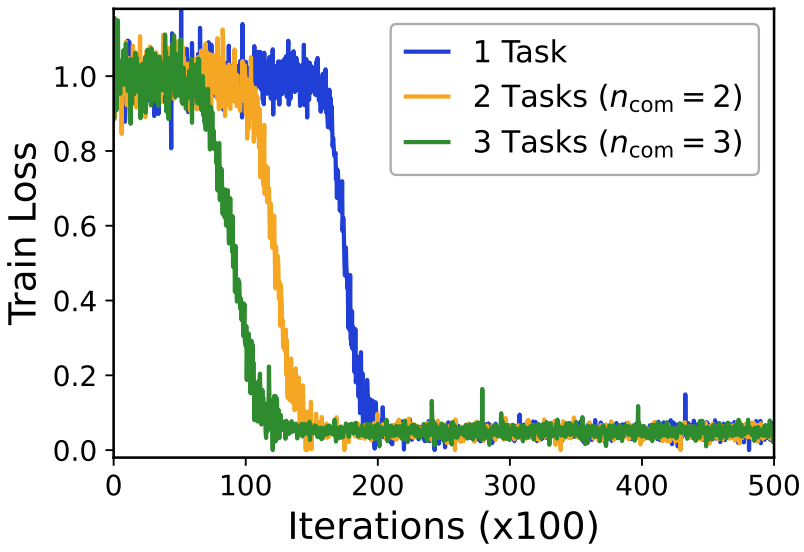
broad task coverage

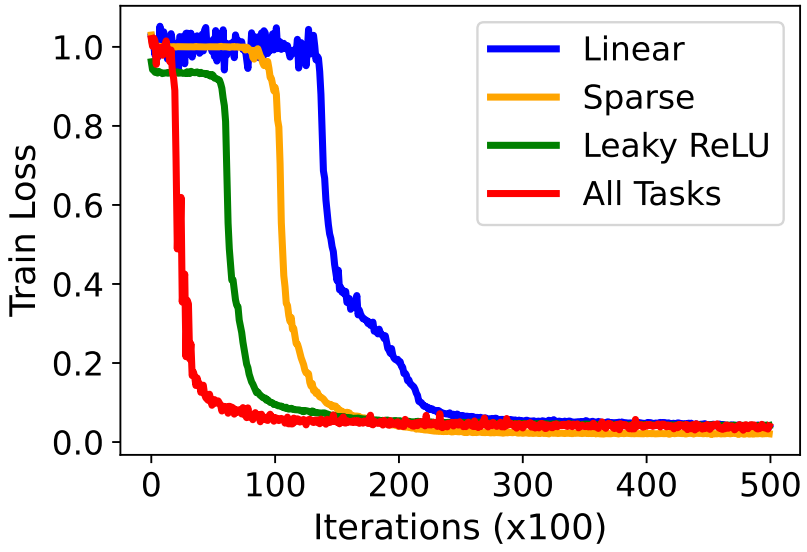
⇓

encourage **pattern learning**

Task diversity (i) is a **key driver** for OOD generalization ability.

Claim II: Task Diversity Matters for Model Training





Better task diversity (ii) can **accelerate model pretraining**.

Outline of This Talk

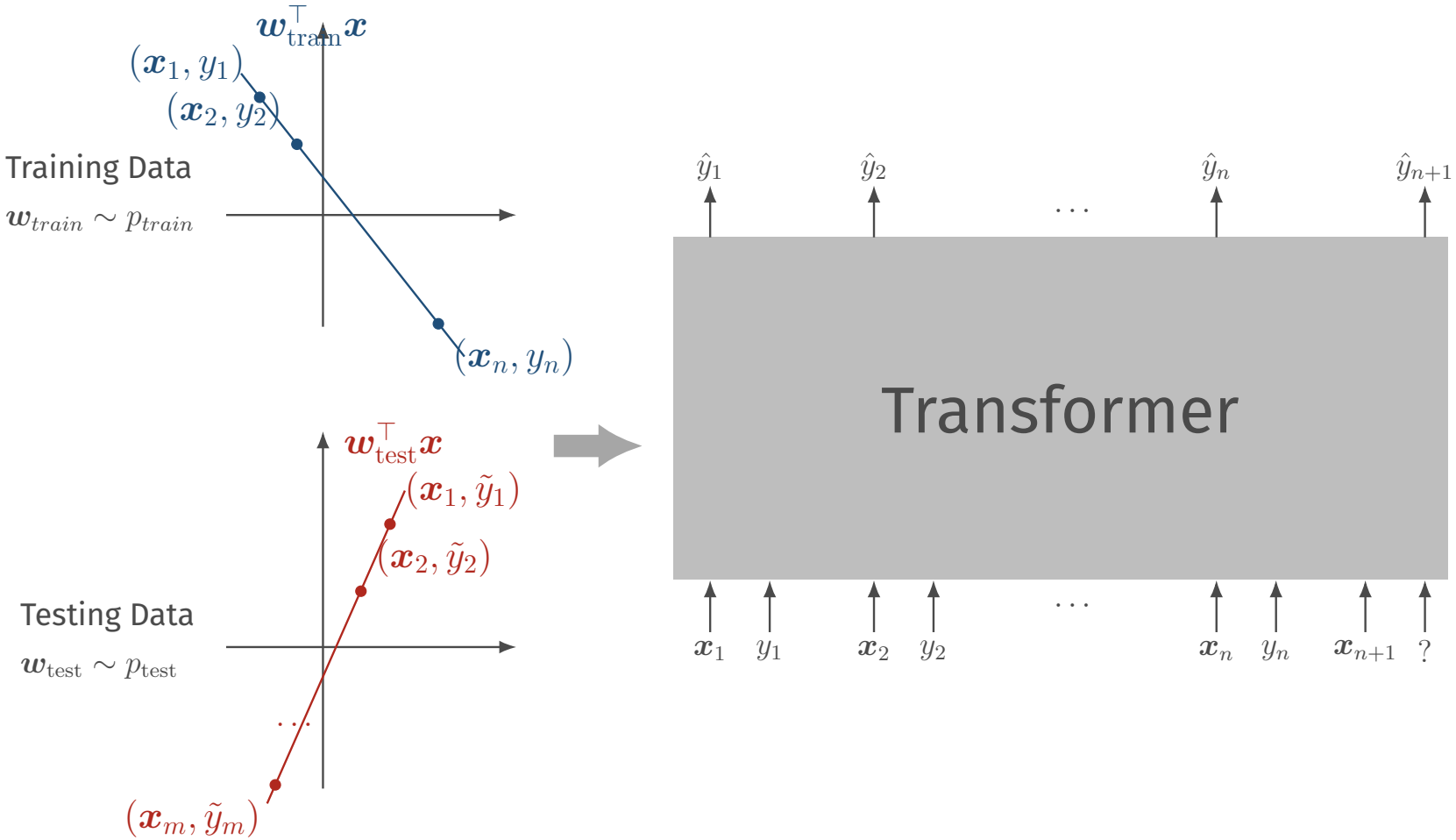
1. Task Diversity Enables OOD Generalization

2. Task Diversity Enables Faster Training

3. Conclusion

Task Diversity Enables OOD Generalization

Linear Regression: A Simple Testbed



Linear regression problem with $x \sim \mathcal{N}(0_d, I_d)$, $\xi \sim \mathcal{N}(0, \sigma^2)$, and

$$y = w_{\text{train}}^\top x + \xi, \quad \text{and} \quad \tilde{y} = w_{\text{test}}^\top x + \xi,$$

where w_{train} and w_{test} are **task vectors** from different subspaces.

Training Phase: A Case Study with Two Subspaces

Given n pairs of in-context examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ with

$$y_i = \boldsymbol{w}_{\text{train}}^\top \boldsymbol{x}_i + \xi_i,$$

with $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$ and $\xi_i \sim \mathcal{N}(0, \sigma^2)$.

Training Phase: A Case Study with Two Subspaces

Given n pairs of in-context examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ with

$$y_i = \boldsymbol{w}_{\text{train}}^\top \boldsymbol{x}_i + \xi_i,$$

with $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$ and $\xi_i \sim \mathcal{N}(0, \sigma^2)$.

We consider the diversity of **training task vector** w_{train}:

- **Case 1. With Diversity:** w_{train} is drawn from

$$\boldsymbol{w}_{\text{train}} \sim \begin{cases} \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{s,1}), & w.p.\ 1/2, \\ \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{s,2}), & w.p.\ 1/2, \end{cases}$$

  where $\boldsymbol{\Sigma}_{s,k} = \boldsymbol{U}_{s,k}\boldsymbol{U}_{s,k}^\top + \epsilon \cdot \boldsymbol{I}_d$ with and $\boldsymbol{U}_{s,k} \in \mathbb{R}^{d \times r}$ and $\boldsymbol{U}_{s,1} \perp \boldsymbol{U}_{s,2}$.

Training Phase: A Case Study with Two Subspaces

Given n pairs of in-context examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ with

$$y_i = \boldsymbol{w}_{\text{train}}^\top \boldsymbol{x}_i + \xi_i,$$

with $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$ and $\xi_i \sim \mathcal{N}(0, \sigma^2)$.

We consider the diversity of **training task vector** w_{train}:

- **Case 1. With Diversity:** w_{train} is drawn from

$$\boldsymbol{w}_{\text{train}} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,1}), & w.p.\ 1/2, \\ \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,2}), & w.p.\ 1/2, \end{cases}$$

  where $\boldsymbol{\Sigma}_{s,k} = \boldsymbol{U}_{s,k}\boldsymbol{U}_{s,k}^\top + \epsilon \cdot \boldsymbol{I}_d$ with and $\boldsymbol{U}_{s,k} \in \mathbb{R}^{d\times r}$ and $\boldsymbol{U}_{s,1} \perp \boldsymbol{U}_{s,2}$.

- **Case 2. Without Diversity:** w_{train} is drawn from a single Gaussian $\boldsymbol{w}_{\text{train}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,1})$ with $\boldsymbol{U}_{s,1} \in \mathbb{R}^{d\times r}$.

Training Phase: Learning One-Layer Linear Attention Model

We write in-context examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and query $\boldsymbol{x}_{n+1}$ as

$$\boldsymbol{Z}_{\mathcal{M}} := \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_n & \boldsymbol{0}_d \\ y_1 & \dots & y_n & 0 \end{bmatrix}^\top \text{ and } \boldsymbol{z}_q := \begin{bmatrix} \boldsymbol{x}_{n+1} \\ 0 \end{bmatrix}$$

Training Phase: Learning One-Layer Linear Attention Model

We write in-context examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and query $\boldsymbol{x}_{n+1}$ as

$$\boldsymbol{Z}_{\mathcal{M}} := \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_n & \boldsymbol{0}_d \\ y_1 & \dots & y_n & 0 \end{bmatrix}^\top \text{ and } \boldsymbol{z}_q := \begin{bmatrix} \boldsymbol{x}_{n+1} \\ 0 \end{bmatrix}$$

- **Model parameterization:** We train a one-layer linear attention

$$\hat{y}_{n+1} = g_{\boldsymbol{\Theta}}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q) = \frac{1}{n} \boldsymbol{z}_q^\top \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{Z}_{\mathcal{M}}^\top \boldsymbol{Z}_{\mathcal{M}} \boldsymbol{W}_V \boldsymbol{p},$$

  where $\boldsymbol{\Theta} := \{\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{p}\}$ are trainable parameters.

Training Phase: Learning One-Layer Linear Attention Model

We write in-context examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and query $\boldsymbol{x}_{n+1}$ as

$$\boldsymbol{Z}_{\mathcal{M}} := \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_n & \boldsymbol{0}_d \\ y_1 & \dots & y_n & 0 \end{bmatrix}^\top \text{ and } \boldsymbol{z}_q := \begin{bmatrix} \boldsymbol{x}_{n+1} \\ 0 \end{bmatrix}$$

- **Model parameterization:** We train a one-layer linear attention

$$\hat{y}_{n+1} = g_{\boldsymbol{\Theta}}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q) = \frac{1}{n} \boldsymbol{z}_q^\top \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{Z}_{\mathcal{M}}^\top \boldsymbol{Z}_{\mathcal{M}} \boldsymbol{W}_V \boldsymbol{p},$$

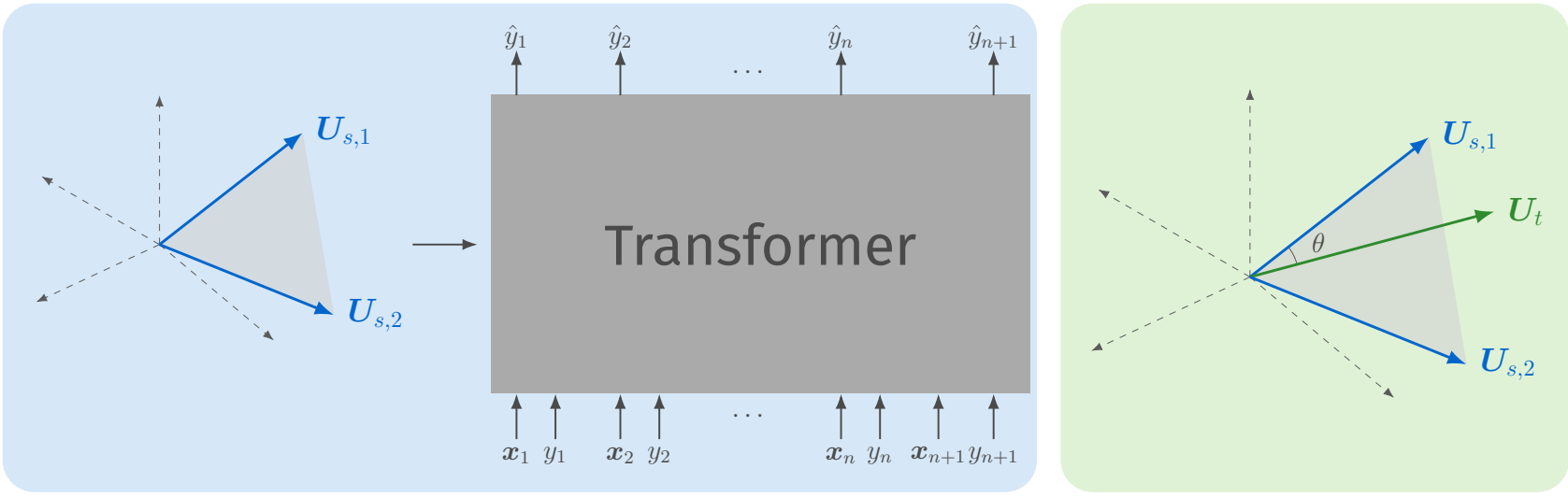
  where $\boldsymbol{\Theta} := \{\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{p}\}$ are trainable parameters.

- **Training loss:** we minimize the population loss

$$\min_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q; \boldsymbol{\Theta}) := \mathbb{E}_{\boldsymbol{Z}} \left[\left(y_{n+1} - \underbrace{g_{\boldsymbol{\Theta}}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q)}_{=\widehat{y}_{n+1}} \right)^2 \right],$$

  with the optimal weight $\boldsymbol{\Theta}_\star$ that has closed-form solutions.

Inference Phase: OOD Generalization Error

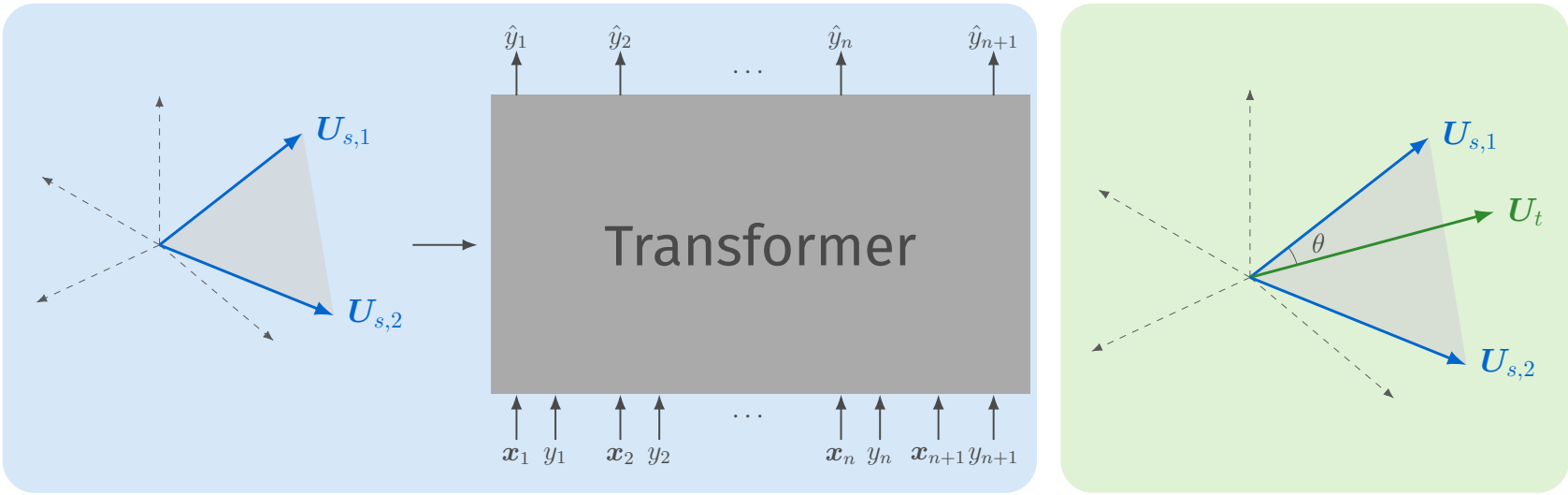


During inference, we consider m pairs of $\{(\boldsymbol{x}_i, \widetilde{y}_i)\}_{i=1}^m \subset \mathbb{R}^d \times \mathbb{R}$ with

$$\widetilde{y}_i = \boldsymbol{w}_{\text{test}}^\top \boldsymbol{x}_i + \xi_i.$$

- Here, $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_t)$ with $\boldsymbol{\Sigma}_t = \boldsymbol{U}_t \boldsymbol{U}_t^\top + \epsilon \boldsymbol{I}_d$.

Inference Phase: OOD Generalization Error



During inference, we consider m pairs of $\{(\boldsymbol{x}_i, \widetilde{y}_i)\}_{i=1}^m \subset \mathbb{R}^d \times \mathbb{R}$ with

$$\widetilde{y}_i = \boldsymbol{w}_{\text{test}}^\top \boldsymbol{x}_i + \xi_i.$$

- Here, $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_t)$ with $\boldsymbol{\Sigma}_t = \boldsymbol{U}_t \boldsymbol{U}_t^\top + \epsilon \boldsymbol{I}_d$.
- For any $\theta \in [0, \pi/2]$, the test subspace $\boldsymbol{U}_t$ satisfies

$$\boldsymbol{U}_t := \cos(\theta) \cdot \boldsymbol{U}_{s,1} + \sin(\theta) \cdot \boldsymbol{U}_{s,2}.$$

Inference Phase: OOD Generalization

During inference, we have $\{\boldsymbol{x}_i, \widetilde{y}_i\}_{i=1}^m$ and query $\boldsymbol{x}_{m+1}$ as

$$\widetilde{\boldsymbol{Z}}_{\mathcal{M}} := \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_m & \mathbf{0}_d \\ \widetilde{y}_1 & \dots & \widetilde{y}_m & 0 \end{bmatrix}^\top \text{ and } \widetilde{\boldsymbol{z}}_q := \begin{bmatrix} \boldsymbol{x}_{m+1} \\ 0 \end{bmatrix}$$

Inference Phase: OOD Generalization

During inference, we have $\{\boldsymbol{x}_i, \widetilde{y}_i\}_{i=1}^m$ and query $\boldsymbol{x}_{m+1}$ as

$$\widetilde{\boldsymbol{Z}}_{\mathcal{M}} := \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_m & \boldsymbol{0}_d \\ \widetilde{y}_1 & \dots & \widetilde{y}_m & 0 \end{bmatrix}^\top \text{ and } \widetilde{\boldsymbol{z}}_q := \begin{bmatrix} \boldsymbol{x}_{m+1} \\ 0 \end{bmatrix}$$

- **Model prediction:** we make predictions through

$$\hat{y}_{m+1} = g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{Z}}_{\mathcal{M}}, \widetilde{\boldsymbol{z}}_q) = \frac{1}{m} \widetilde{\boldsymbol{z}}_q^\top \boldsymbol{W}_Q^\star \boldsymbol{W}_K^{\star\top} \widetilde{\boldsymbol{Z}}_{\mathcal{M}}^\top \widetilde{\boldsymbol{Z}}_{\mathcal{M}} \boldsymbol{W}_V^\star \boldsymbol{p}^\star,$$

Inference Phase: OOD Generalization

During inference, we have $\{\boldsymbol{x}_i, \widetilde{y}_i\}_{i=1}^m$ and query $\boldsymbol{x}_{m+1}$ as

$$\widetilde{\boldsymbol{Z}}_{\mathcal{M}} := \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_m & \boldsymbol{0}_d \\ \widetilde{y}_1 & \dots & \widetilde{y}_m & 0 \end{bmatrix}^\top \text{ and } \widetilde{\boldsymbol{z}}_q := \begin{bmatrix} \boldsymbol{x}_{m+1} \\ 0 \end{bmatrix}$$

- **Model prediction:** we make predictions through

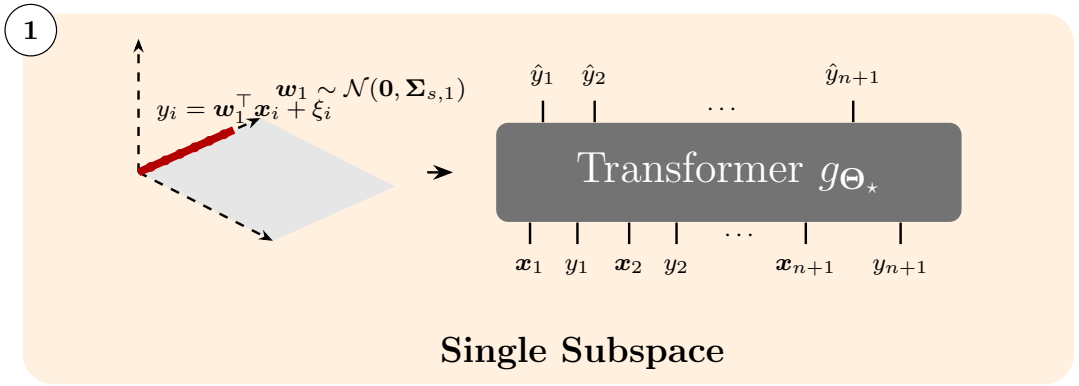
$$\hat{y}_{m+1} = g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{Z}}_{\mathcal{M}}, \widetilde{\boldsymbol{z}}_q) = \frac{1}{m} \widetilde{\boldsymbol{z}}_q^\top \boldsymbol{W}_Q^\star \boldsymbol{W}_K^{\star\top} \widetilde{\boldsymbol{Z}}_{\mathcal{M}}^\top \widetilde{\boldsymbol{Z}}_{\mathcal{M}} \boldsymbol{W}_V^\star \boldsymbol{p}^\star,$$

- **OOD generalization error:** we quantify the error by

$$\mathcal{E}_{\mathrm{OOD}} := \lim_{m\to\infty} \lim_{n\to\infty} \lim_{\epsilon\to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{Z}}_{\mathcal{M}}, \widetilde{\boldsymbol{z}}_q) \right)^2 \right].$$

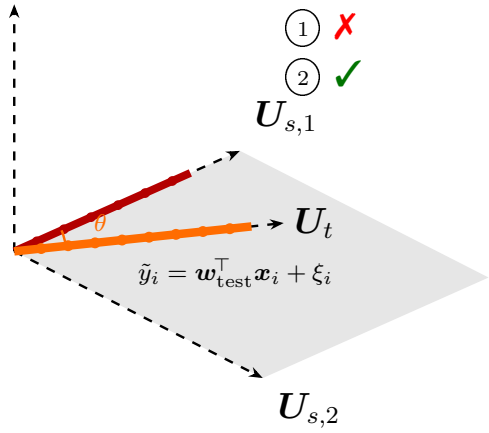
A Geometric Understanding of OOD Generalization

Training Phase



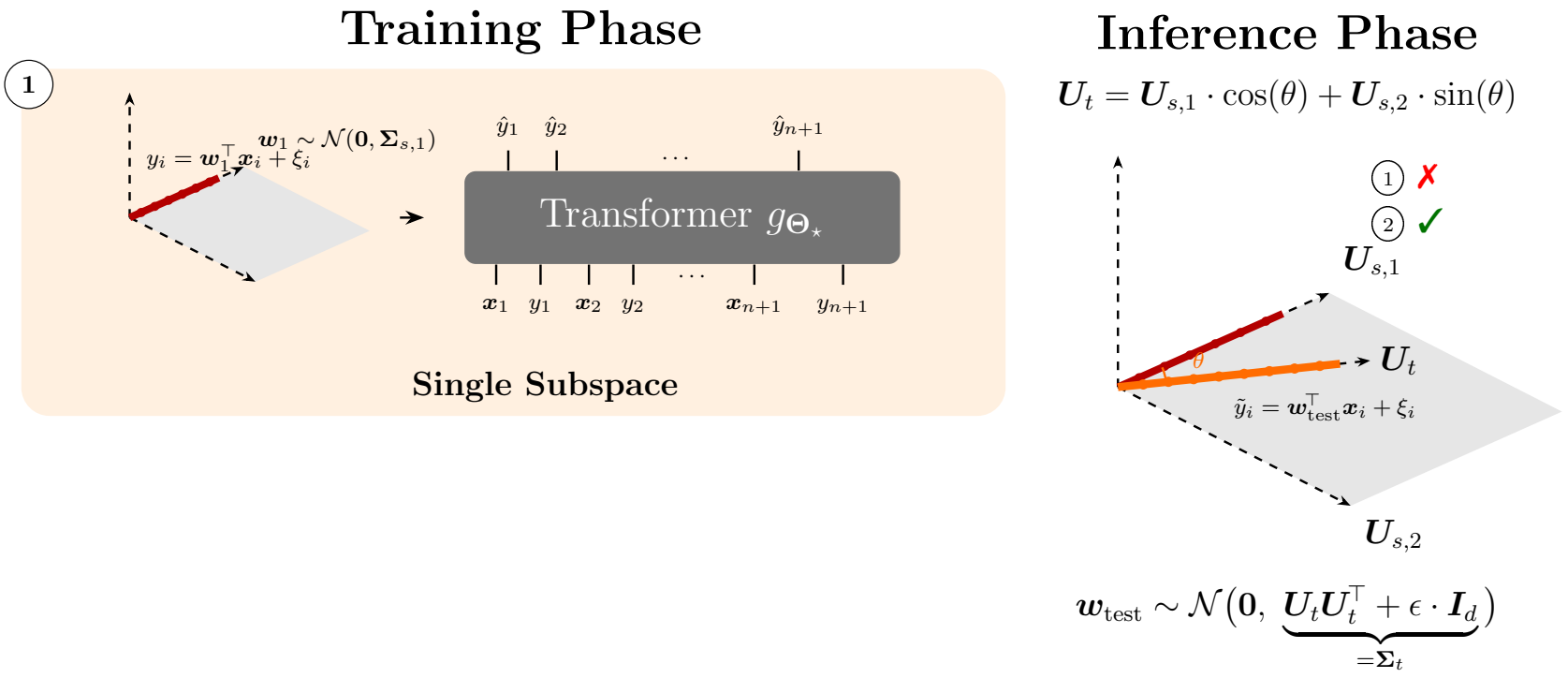
Inference Phase

$$\boldsymbol{U}_t = \boldsymbol{U}_{s,1} \cdot \cos(\theta) + \boldsymbol{U}_{s,2} \cdot \sin(\theta)$$



$$\boldsymbol{w}_{\text{test}} \sim \mathcal{N}\big(\mathbf{0},\ \underbrace{\boldsymbol{U}_t \boldsymbol{U}_t^\top + \epsilon \cdot \boldsymbol{I}_d}_{=\boldsymbol{\Sigma}_t}\big)$$

A Geometric Understanding of OOD Generalization



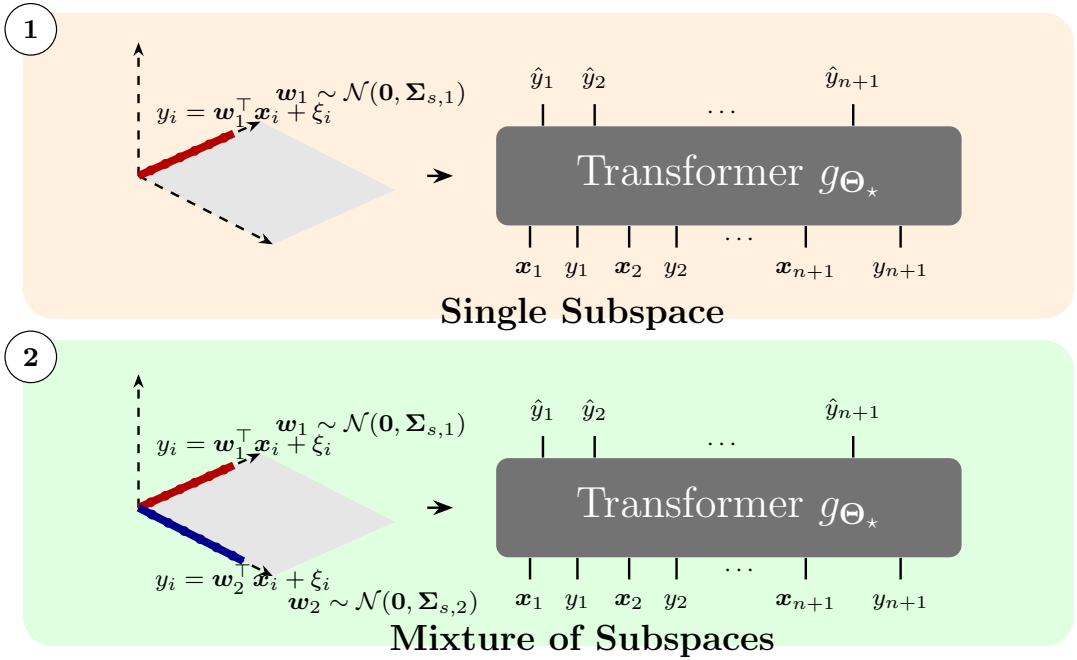
- **Case 1: training with no task diversity.** When $\boldsymbol{w}_{\text{train}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{s,1})$, we encounter a large error

$$\boxed{\mathcal{E}_{\text{OOD}} \propto r \sin^2(\theta)},$$

proportional to the subspace angle θ and dimension r.

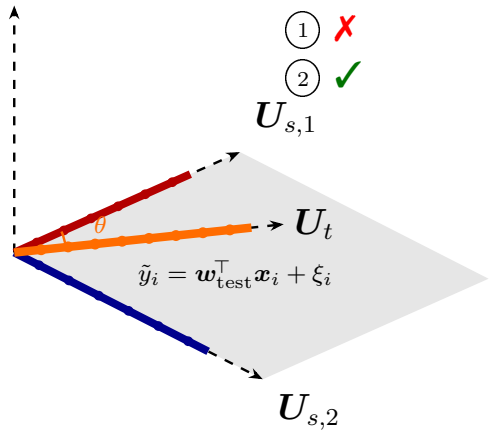
A Geometric Understanding of OOD Generalization

Training Phase



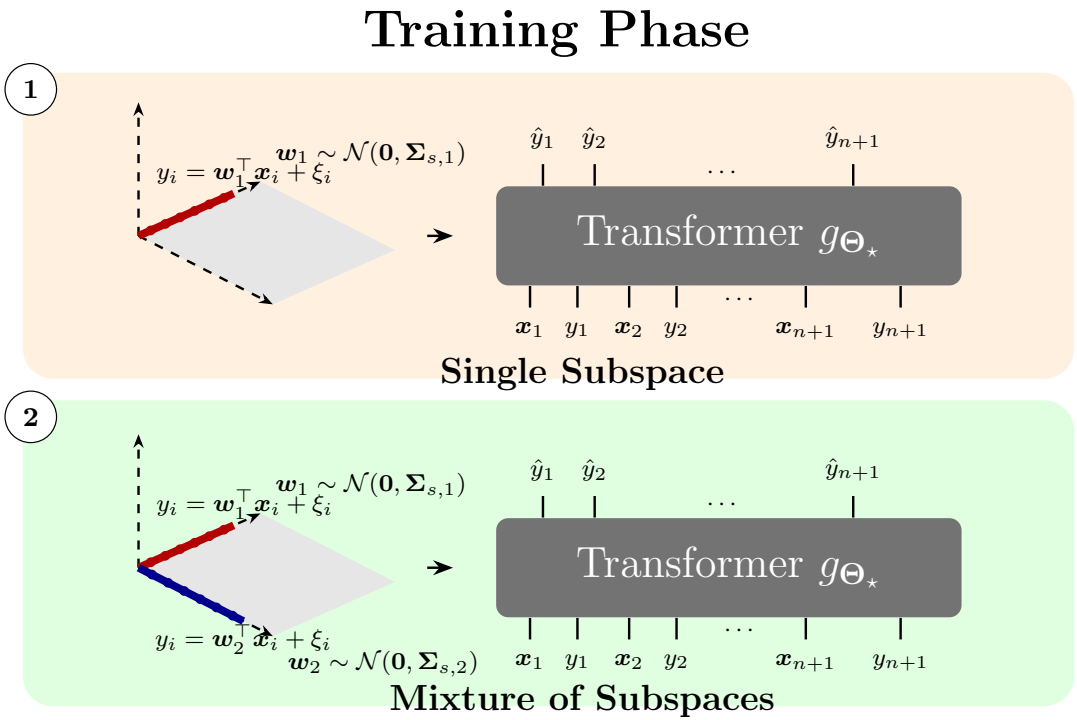
Inference Phase

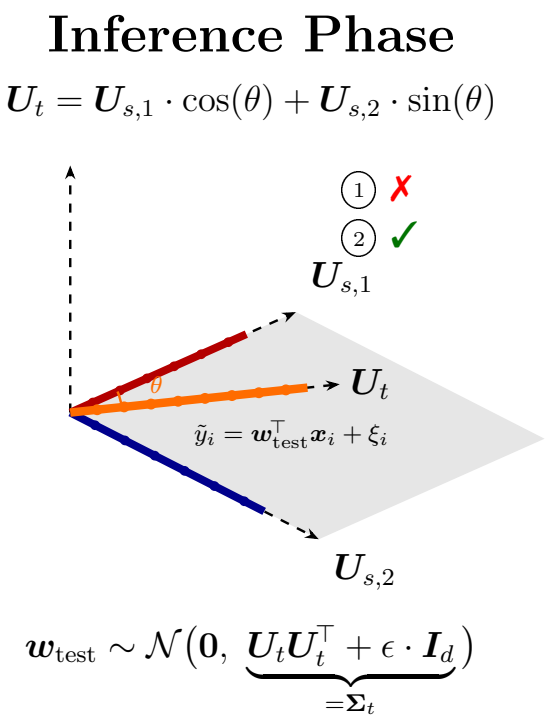
$$\boldsymbol{U}_t = \boldsymbol{U}_{s,1} \cdot \cos(\theta) + \boldsymbol{U}_{s,2} \cdot \sin(\theta)$$



$$\boldsymbol{w}_{\text{test}} \sim \mathcal{N}\big(\boldsymbol{0},\ \underbrace{\boldsymbol{U}_t \boldsymbol{U}_t^\top + \epsilon \cdot \boldsymbol{I}_d}_{=\boldsymbol{\Sigma}_t}\big)$$

A Geometric Understanding of OOD Generalization



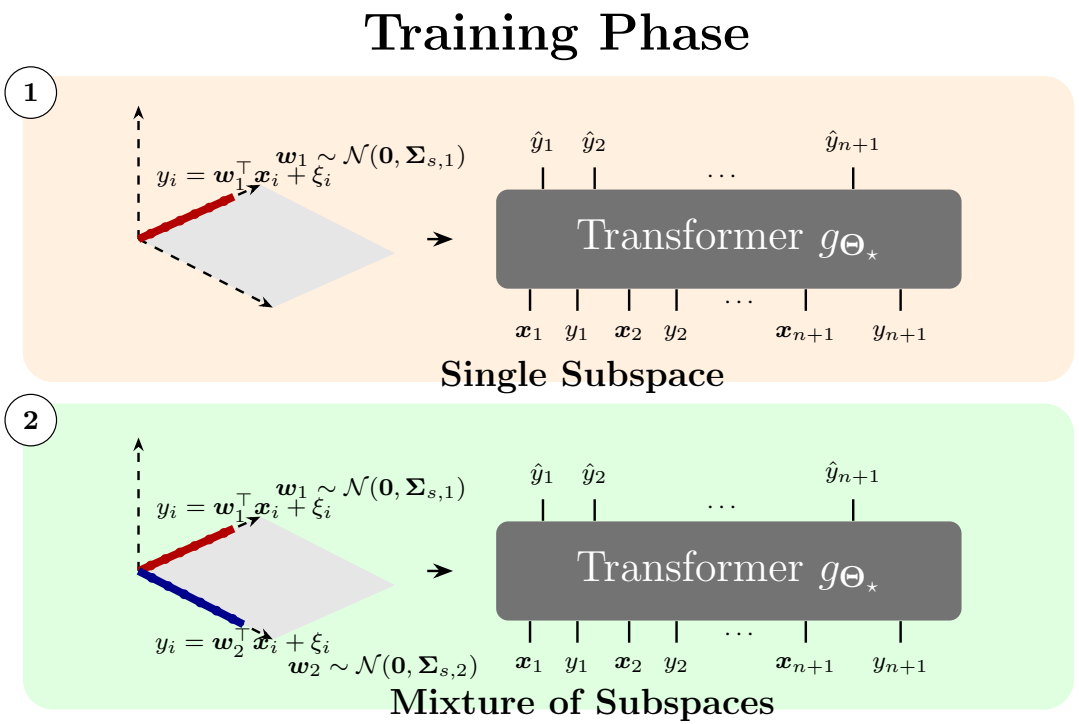


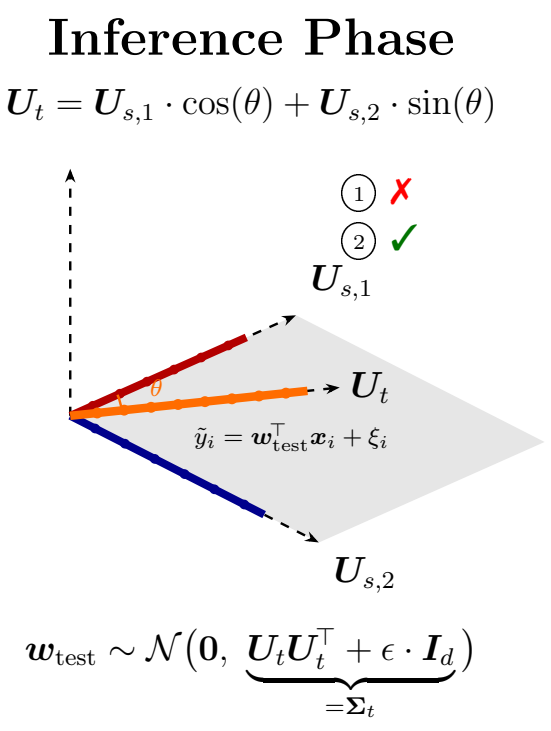
- **Case 2: Training with task diversity.** When trained on diverse tasks $w_{\text{train}} \sim \frac{1}{2}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{s,1}) + \frac{1}{2}\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{s,2})$, we achieve

$$\boxed{\mathcal{E}_{\text{OOD}} \approx \sigma^2},$$

so that we generalize to $\text{span}(U_{s,1}, U_{s,2})$ up to noise level σ^2.

A Geometric Understanding of OOD Generalization

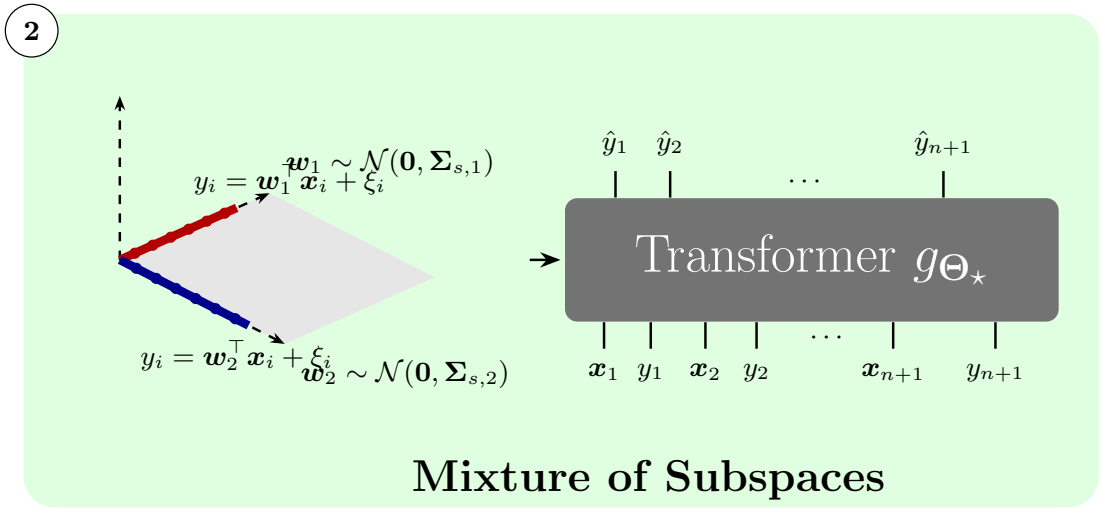




Training task diversity enables OOD generalization!

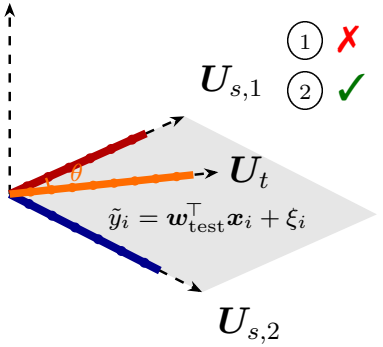
OOD Generalization with Training Task Diversity

Training Phase



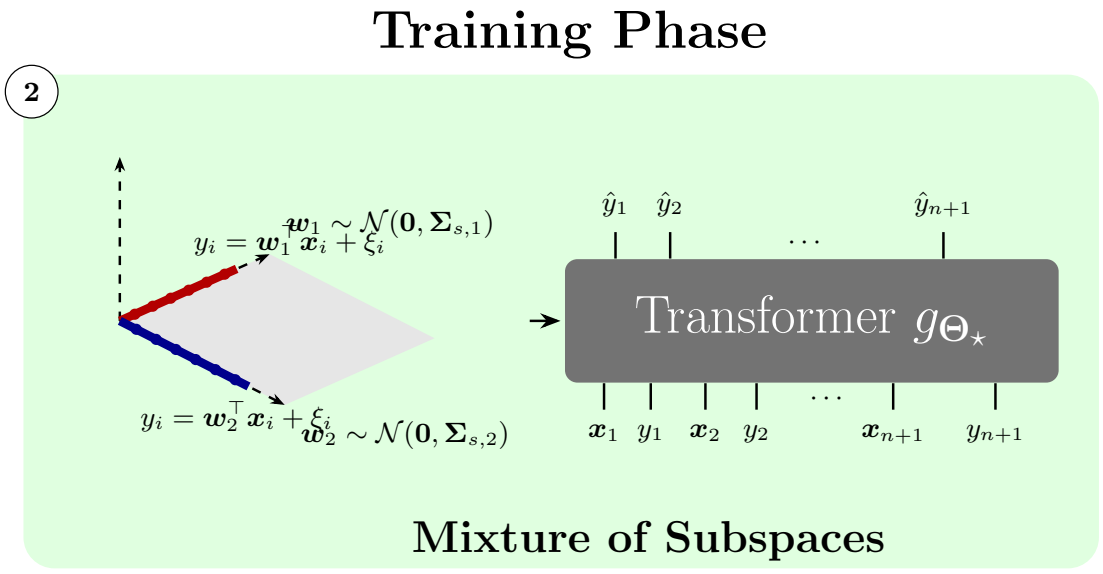
Inference Phase

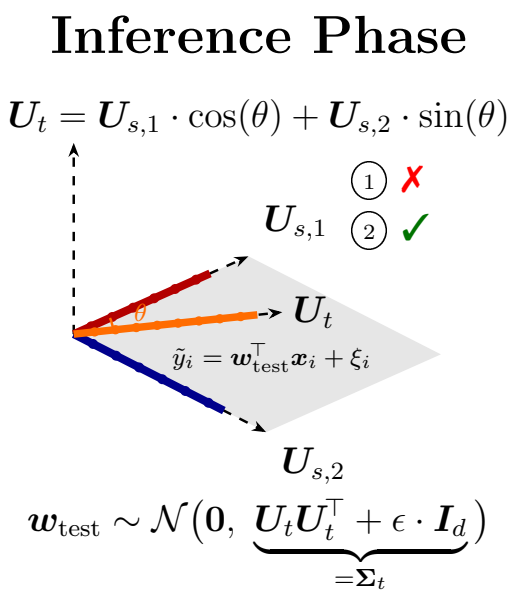
$$\boldsymbol{U}_t = \boldsymbol{U}_{s,1} \cdot \cos(\theta) + \boldsymbol{U}_{s,2} \cdot \sin(\theta)$$



$$\boldsymbol{w}_{\text{test}} \sim \mathcal{N}\big(\boldsymbol{0}, \underbrace{\boldsymbol{U}_t \boldsymbol{U}_t^\top + \epsilon \cdot \boldsymbol{I}_d}_{=\boldsymbol{\Sigma}_t}\big)$$

OOD Generalization with Training Task Diversity





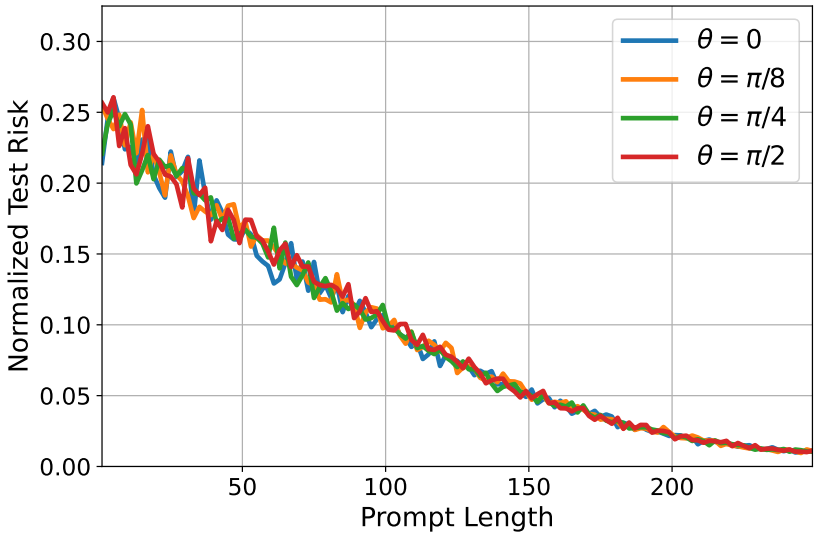
Theorem (Training on Two Subspaces)

Suppose $\boldsymbol{w}_{\text{train}} \sim \frac{1}{2}\mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,1}) + \frac{1}{2}\mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,2})$ and $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_t)$. Let $g_{\boldsymbol{\Theta}_\star}$ denote the optimal linear attention model,

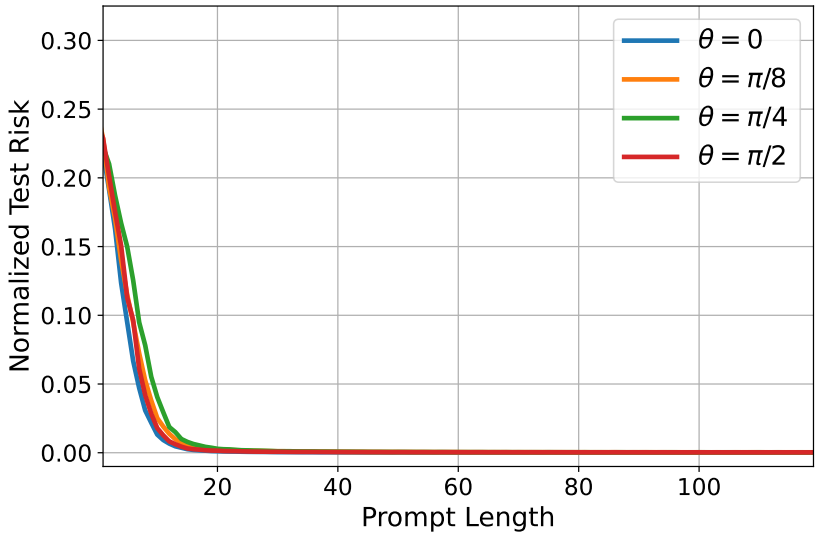
$$m \geq n > \left(2(r+\sigma^2)+1\right)\left(\frac{r}{\delta} - 1\right), \quad \forall \theta \in \left(0, \frac{\pi}{2}\right), \ \delta \in (0, r),$$

then $\lim_{\epsilon \to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{z}}_q, \widetilde{\boldsymbol{Z}}_{\mathcal{M}})\right)^2\right] < \sigma^2 + \delta.$

Experiments and Observations

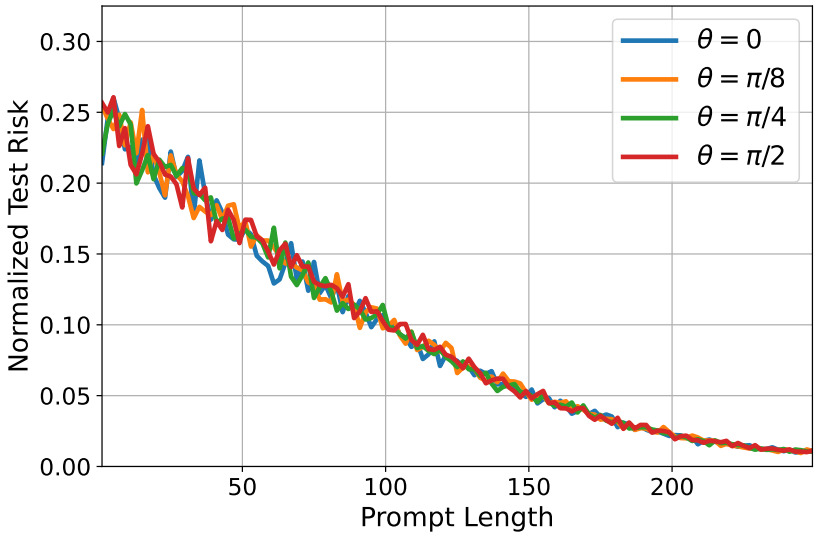


Linear Attention

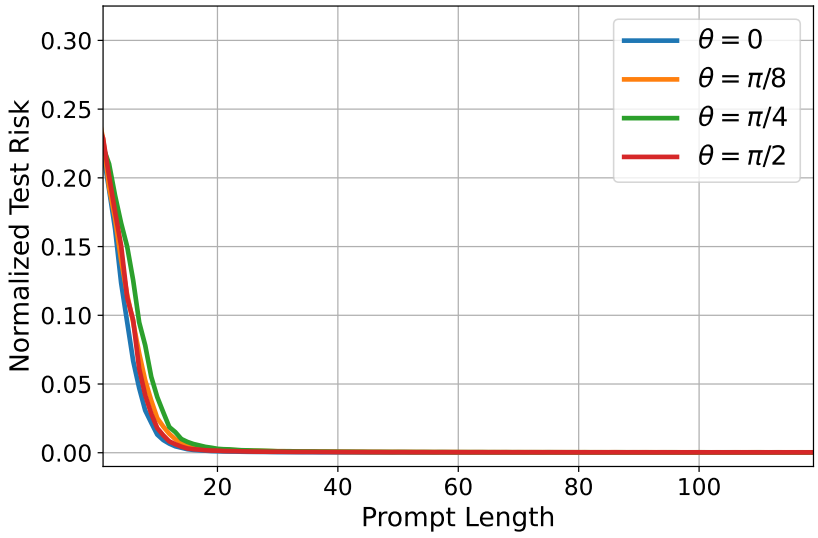


GPT-2 Model

Experiments and Observations



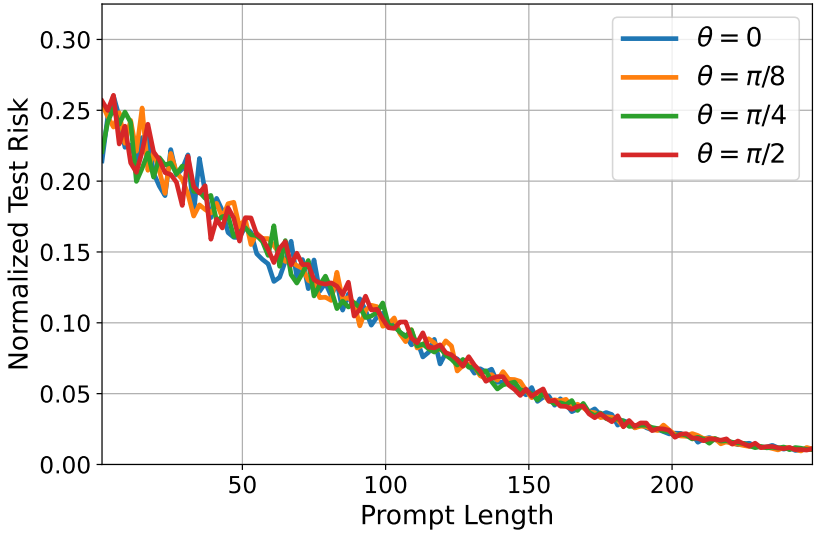
Linear Attention



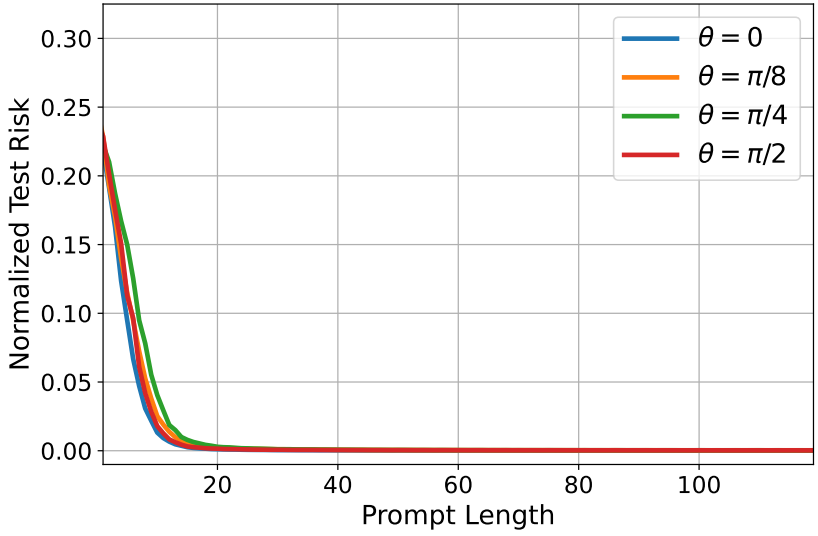
GPT-2 Model

- **OOD generalization beyond training data distribution.** Although we only trained on data from each subspace $U_{s,k}$, ICL generalizes well to the whole span$(U_{s,1}, U_{s,2})$.

Experiments and Observations



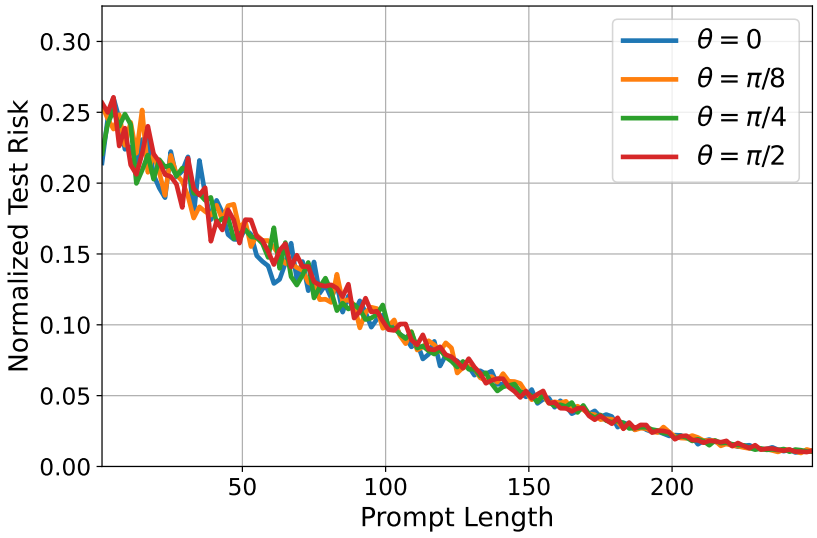
Linear Attention



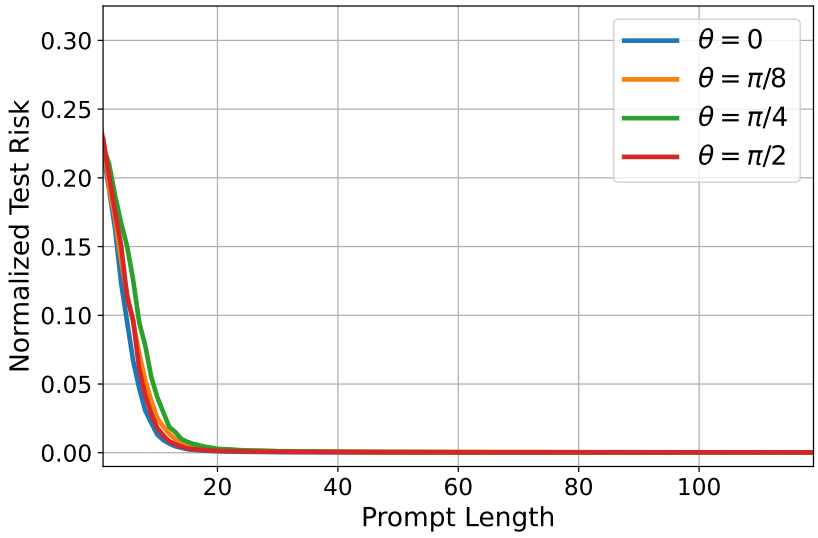
GPT-2 Model

- **OOD generalization beyond training data distribution.** Although we only trained on data from each subspace $U_{s,k}$, ICL generalizes well to the whole span$(U_{s,1}, U_{s,2})$.
- **Required number of in-context examples.** Train and inference time in-context examples (n, m) is governed by r.

Experiments and Observations



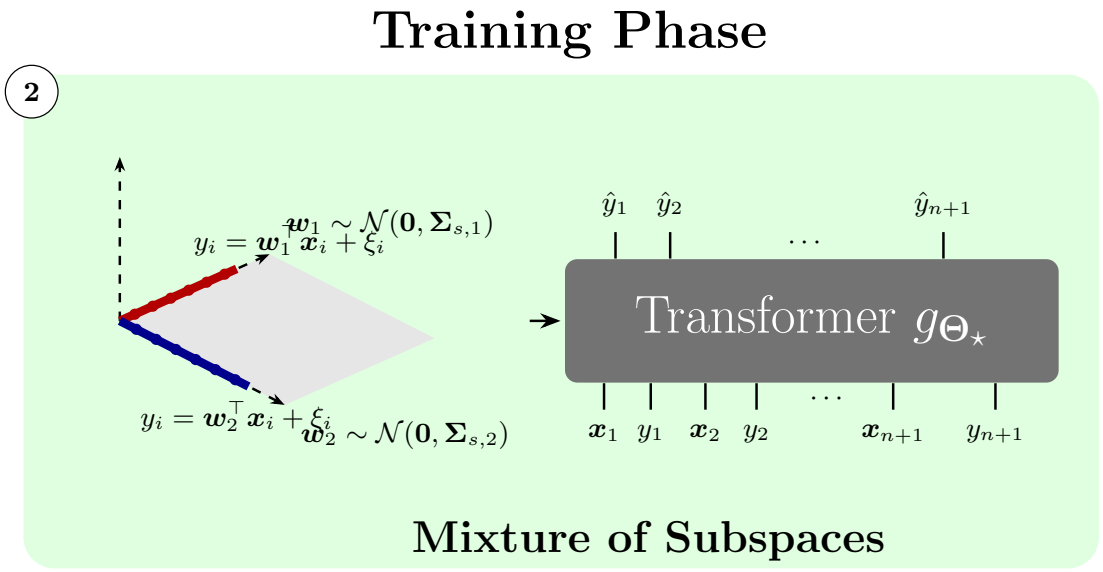
Linear Attention

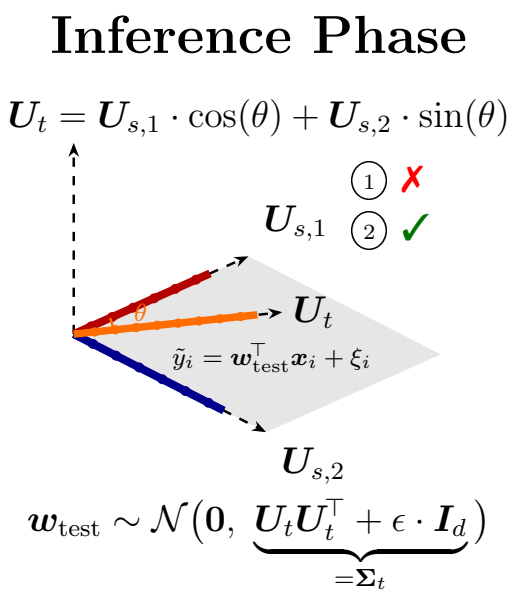


GPT-2 Model

- **OOD generalization beyond training data distribution.** Although we only trained on data from each subspace $U_{s,k}$, ICL generalizes well to the whole span$(U_{s,1}, U_{s,2})$.
- **Required number of in-context examples.** Train and inference time in-context examples (n, m) is governed by r.
- **Nonlinear vs. linear models.** Nonlinear models require fewer in-context examples than linear models.

OOD Generalization with Training Task Diversity





Theorem (Training on Two Subspaces)

Suppose $\boldsymbol{w}_{\text{train}} \sim \frac{1}{2}\mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,1}) + \frac{1}{2}\mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,2})$ and $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_t)$. Let $g_{\boldsymbol{\Theta}_\star}$ denote the optimal linear attention model,

$$m \geq n > \left(2(r+\sigma^2)+1\right)\left(\frac{r}{\delta} - 1\right), \quad \forall \theta \in \left(0, \frac{\pi}{2}\right), \ \delta \in (0, r),$$

then $\lim_{\epsilon \to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{z}}_q, \widetilde{\boldsymbol{Z}}_{\mathcal{M}})\right)^2\right] < \sigma^2 + \delta.$

Proving Ideas

- The model prediction $g_{\Theta_\star}$ is essentially a **single step of projected gradient descent (PGD)** (Oswald et al. 2023):

$$\hat{y}_{m+1} = g_{\Theta_\star}(\widetilde{\boldsymbol{Z}}_{\mathcal{M}}, \widetilde{\boldsymbol{z}}_q) = \frac{1}{m} \boldsymbol{x}_{m+1}^\top \boldsymbol{A} \boldsymbol{X}^\top \underbrace{\boldsymbol{X} \boldsymbol{w}_{\text{test}}}_{\widetilde{\boldsymbol{y}}},$$

  where $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is the PGD projection with optimal weights $\Theta_\star$, and $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_m \end{bmatrix}^\top$, $\widetilde{\boldsymbol{y}} = \begin{bmatrix} \widetilde{y}_1 & \cdots & \widetilde{y}_m \end{bmatrix}^\top$.

Proving Ideas

- The model prediction $g_{\boldsymbol{\Theta}_\star}$ is essentially a **single step of projected gradient descent (PGD)** (Oswald et al. 2023):

$$\hat{y}_{m+1} = g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{Z}}_{\mathcal{M}}, \widetilde{\boldsymbol{z}}_q) = \frac{1}{m} \boldsymbol{x}_{m+1}^\top \boldsymbol{A} \boldsymbol{X}^\top \underbrace{\boldsymbol{X} \boldsymbol{w}_{\text{test}}}_{\widetilde{\boldsymbol{y}}},$$

  where $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is the PGD projection with optimal weights $\boldsymbol{\Theta}_\star$, and $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_m \end{bmatrix}^\top$, $\widetilde{\boldsymbol{y}} = \begin{bmatrix} \widetilde{y}_1 & \cdots & \widetilde{y}_m \end{bmatrix}^\top$.

- With $m, n \to \infty, \epsilon \to 0$ and $\boldsymbol{U}_{2r} = [\boldsymbol{U}_{s,1}, \boldsymbol{U}_{s,2}]$, we have

$$\begin{aligned} \hat{y}_{m+1} = \frac{1}{m} \boldsymbol{x}_{m+1}^\top \boldsymbol{A} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w}_{\text{test}} &\to \boldsymbol{x}_{m+1}^\top \boldsymbol{U}_{2r} \boldsymbol{U}_{2r}^\top \boldsymbol{w}_{\text{test}} \\ &= \boldsymbol{x}_{m+1}^\top \boldsymbol{U}_{2r} \boldsymbol{U}_{2r}^\top \boldsymbol{U}_t \boldsymbol{g}, \end{aligned}$$

  where $\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_r)$.

Proving Ideas

- With $m, n \to \infty, \epsilon \to 0)$ and $\boldsymbol{U}_{2r} = [\boldsymbol{U}_{s,1}, \boldsymbol{U}_{s,2}]$, we have

$$\begin{aligned}\hat{y}_{m+1} = \frac{1}{m}\boldsymbol{x}_{m+1}^\top \boldsymbol{A}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w}_{\text{test}} &\to \boldsymbol{x}_{m+1}^\top \boldsymbol{U}_{2r}\boldsymbol{U}_{2r}^\top \boldsymbol{w}_{\text{test}} \\ &= \boldsymbol{x}_{m+1}^\top \boldsymbol{U}_{2r}\boxed{\boldsymbol{U}_{2r}^\top \boldsymbol{U}_t}\boldsymbol{g}, \\ &= \boldsymbol{x}_{m+1}^\top \boldsymbol{w}_{\text{test}}.\end{aligned}$$

- Since $\text{span}(\boldsymbol{U}_t) \in \text{span}(\boldsymbol{U}_{2r})$ for all $\theta \in [0, \pi/2]$, the trained model prediction $\hat{y}_{m+1} = \boldsymbol{w}_{\text{test}}^\top \boldsymbol{x}_{m+1}$ perfectly recovers $\widetilde{y}_{m+1}$.

OOD Generalization with Training Task Diversity

Theorem (Training on Multiple Subspaces)

Suppose

- $\boldsymbol{w}_{\text{train}} \sim \sum_{k=1}^{K} \gamma_k \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,k})$ with $\gamma_k = 1/K$.
- $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \overline{\boldsymbol{\Sigma}}_t)$ with $\overline{\boldsymbol{\Sigma}}_t = \overline{\boldsymbol{U}}_t \overline{\boldsymbol{U}}_t^\top + \epsilon \cdot \boldsymbol{I}_d$ and $\overline{\boldsymbol{\Sigma}}_t = \sum_{k=1}^{K} \alpha_k \boldsymbol{U}_{s,k}$.

Let $g_{\boldsymbol{\Theta}_\star}$ denote the optimal linear attention model. For any $\{\alpha_k\}$ s.t. $\sum_{k=1}^{K} \alpha_k^2 = 1$ and $\delta \in (0, r)$, if

$$m \geq n > (\frac{r}{\delta} - 1)\left(K(r + \sigma^2) + 1\right),$$

then $\lim_{\epsilon \to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{z}}_q, \widetilde{\boldsymbol{Z}}_{\mathcal{M}})\right)^2\right] < \sigma^2 + \delta.$

OOD Generalization with Training Task Diversity

Theorem (Training on Multiple Subspaces)

Suppose

- $\boldsymbol{w}_{\text{train}} \sim \sum_{k=1}^{K} \gamma_k \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,k})$ with $\gamma_k = 1/K$.
- $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \overline{\boldsymbol{\Sigma}}_t)$ with $\overline{\boldsymbol{\Sigma}}_t = \overline{\boldsymbol{U}}_t \overline{\boldsymbol{U}}_t^\top + \epsilon \cdot \boldsymbol{I}_d$ and $\overline{\boldsymbol{\Sigma}}_t = \sum_{k=1}^{K} \alpha_k \boldsymbol{U}_{s,k}$.

Let $g_{\boldsymbol{\Theta}_\star}$ denote the optimal linear attention model. For any $\{\alpha_k\}$ s.t. $\sum_{k=1}^{K} \alpha_k^2 = 1$ and $\delta \in (0, r)$, if

$$m \geq n > (\frac{r}{\delta} - 1)\left(K(r + \sigma^2) + 1\right),$$

then $\lim_{\epsilon \to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{z}}_q, \widetilde{\boldsymbol{Z}}_{\mathcal{M}})\right)^2\right] < \sigma^2 + \delta.$

- This implies that we can generalize to $\text{span}(\boldsymbol{U}_{s,1}, \cdots, \boldsymbol{U}_{s,K})$.

OOD Generalization with Training Task Diversity

Theorem (Training on Multiple Subspaces)

Suppose

- $\boldsymbol{w}_{\text{train}} \sim \sum_{k=1}^{K} \gamma_k \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_{s,k})$ with $\gamma_k = 1/K$.
- $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\boldsymbol{0}, \overline{\boldsymbol{\Sigma}}_t)$ with $\overline{\boldsymbol{\Sigma}}_t = \overline{\boldsymbol{U}}_t \overline{\boldsymbol{U}}_t^\top + \epsilon \cdot \boldsymbol{I}_d$ and $\overline{\boldsymbol{\Sigma}}_t = \sum_{k=1}^{K} \alpha_k \boldsymbol{U}_{s,k}$.

Let $g_{\boldsymbol{\Theta}_\star}$ denote the optimal linear attention model. For any $\{\alpha_k\}$ s.t. $\sum_{k=1}^{K} \alpha_k^2 = 1$ and $\delta \in (0, r)$, if

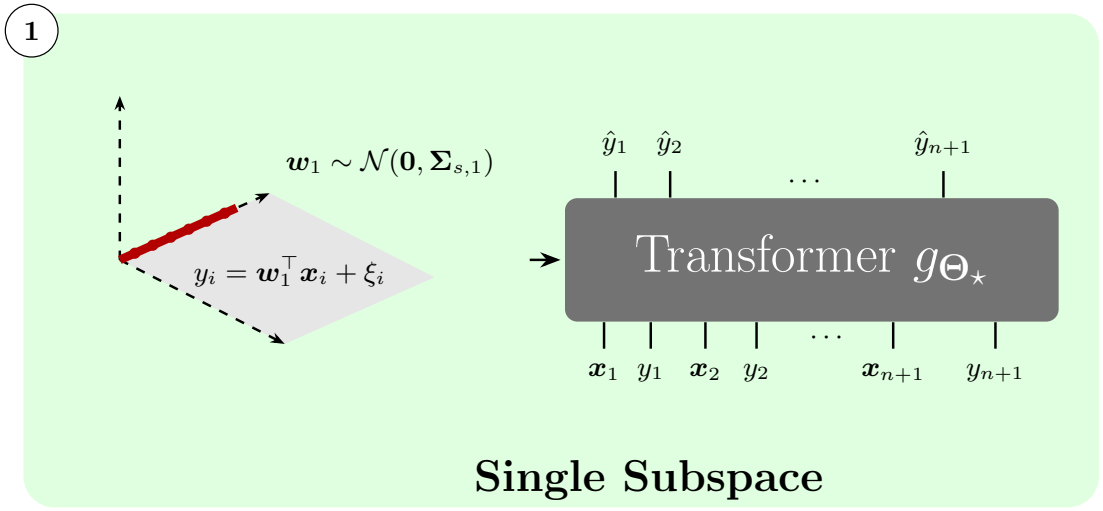
$$m \geq n > (\frac{r}{\delta} - 1)\left(K(r + \sigma^2) + 1\right),$$

then $\lim_{\epsilon \to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\boldsymbol{\Theta}_\star}(\widetilde{\boldsymbol{z}}_q, \widetilde{\boldsymbol{Z}}_{\mathcal{M}})\right)^2\right] < \sigma^2 + \delta.$

- This implies that we can generalize to $\text{span}(\boldsymbol{U}_{s,1}, \cdots, \boldsymbol{U}_{s,K})$.
- If the subspaces have angles θ', the dependence on subspace angles can be increased by a factor of $1/\sin(\theta')$.

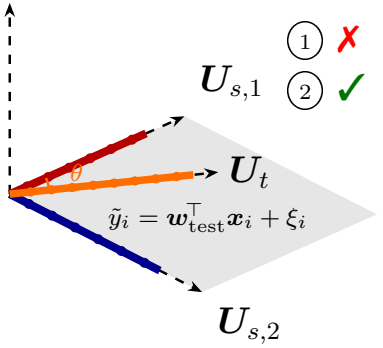
OOD Generalization with No Training Task Diversity

Training Phase



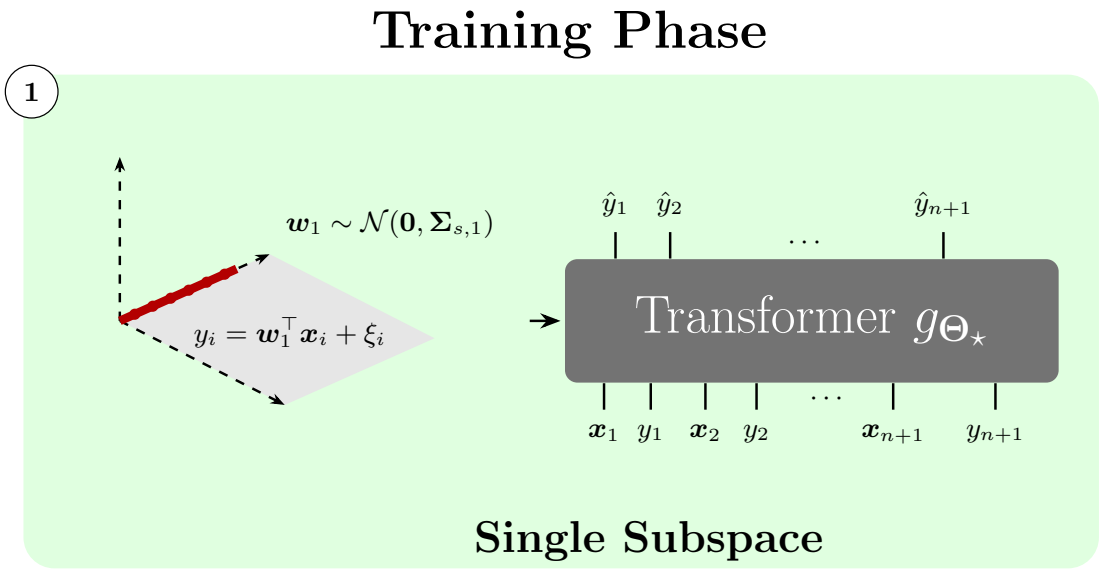
Inference Phase

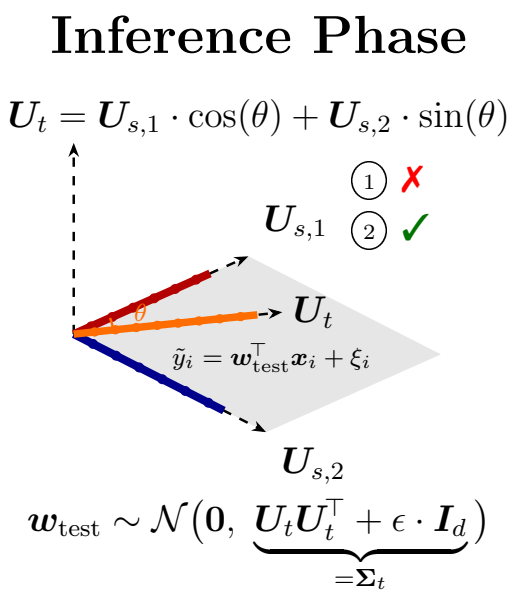
$$\boldsymbol{U}_t = \boldsymbol{U}_{s,1} \cdot \cos(\theta) + \boldsymbol{U}_{s,2} \cdot \sin(\theta)$$



$$\boldsymbol{w}_{\text{test}} \sim \mathcal{N}\big(\mathbf{0},\ \underbrace{\boldsymbol{U}_t \boldsymbol{U}_t^\top + \epsilon \cdot \boldsymbol{I}_d}_{=\boldsymbol{\Sigma}_t}\big)$$

OOD Generalization with No Training Task Diversity





Theorem (Training on Single Subspace)

Suppose $w_{\text{train}} \sim \mathcal{N}(0, U_{s,1}U_{s,1}^\top)$ and $w_{\text{test}} \sim \mathcal{N}(0, U_t U_t^\top)$ with angle θ. Let $g_{\Theta_\star}$ denote the optimal linear attention model, then

$$\lim_{m,n\to\infty} \lim_{\epsilon\to 0} \mathbb{E}\left[\left(\widetilde{y}_{m+1} - g_{\Theta_\star}(\widetilde{z}_q, \widetilde{Z}_{\mathcal{M}})\right)^2\right] = r \cdot \sin^2(\theta) + \sigma^2.$$

Proving Ideas

- With $m, n \to \infty, \epsilon \to 0)$, we have

$$\hat{y}_{m+1} = \frac{1}{m} \boldsymbol{x}_{m+1}^\top \boldsymbol{A} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{w}_{\text{test}} \to \boldsymbol{x}_{m+1}^\top \boldsymbol{U}_{s,1} \boldsymbol{U}_{s,1}^\top \boldsymbol{w}_{\text{test}}$$
$$= \boldsymbol{x}_{m+1}^\top \boldsymbol{U}_{s,1} \boxed{\boldsymbol{U}_{s,1}^\top \boldsymbol{U}_t} \boldsymbol{g}.$$

- Since $\boldsymbol{U}_t$ and $\boldsymbol{U}_{s,1}$ have angles

$$\boldsymbol{U}_t := \cos(\theta) \cdot \boldsymbol{U}_{s,1} + \sin(\theta) \cdot \boldsymbol{U}_{s,2},$$

the model prediction will have an error depending on θ and r.

OOD Generalization with No Training Task Diversity

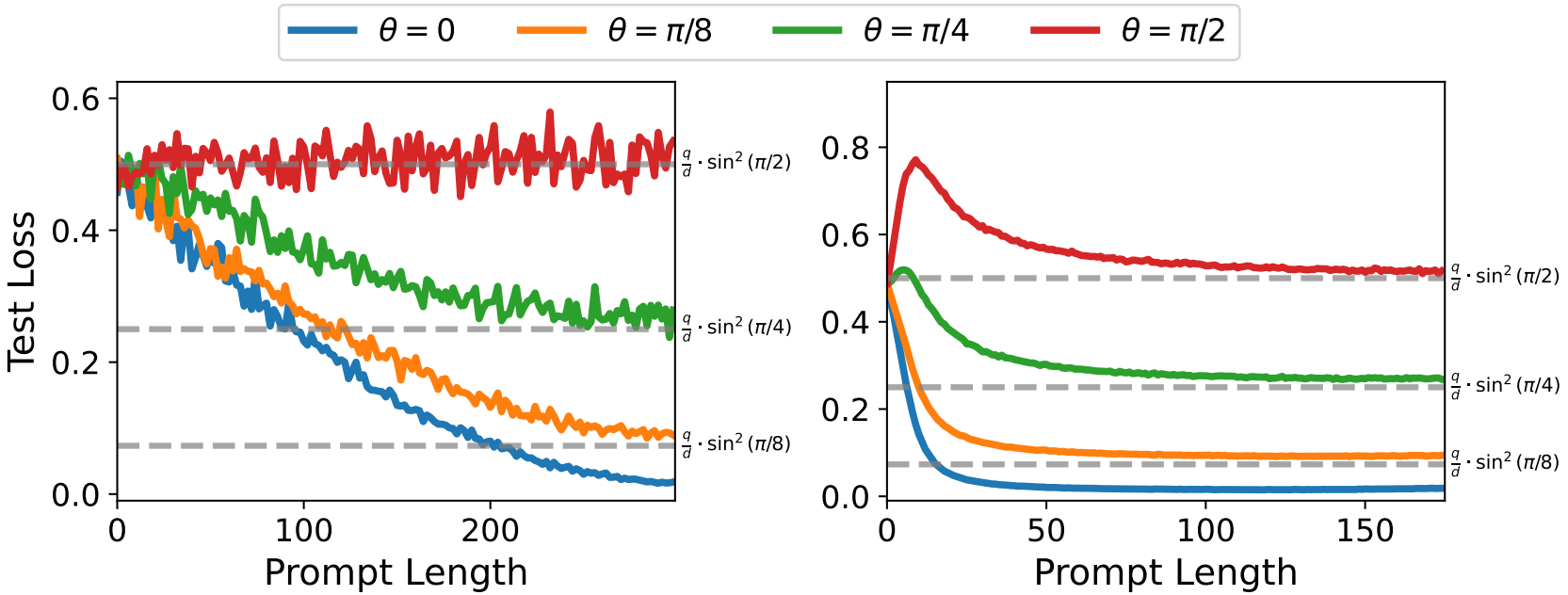


Figure 1: Left: Linear Attention; **Right:** GPT2 Model

- **ICL cannot generalize beyond the span of the training subspaces.** The error depends on (i) the subspace angle θ between training and test, and (ii) the subspace dimension r.
- **It is a fundamental limitation of ICL rather than linear models.** The GPT2 model exhibits the same error, but this occurs with a reduced prompt length.

Can LoRA Be Used to Adapt Subspace Shifts?

Theorem (Training on Single Subspace)

Suppose $\boldsymbol{w}_{\text{train}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{U}_{s,1}\boldsymbol{U}_{s,1}^\top)$ and $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{U}_t\boldsymbol{U}_t^\top)$ with angle θ. Define the **adapted model** $h_{\boldsymbol{\Theta}^\star}\left(\widetilde{\boldsymbol{Z}}, \boldsymbol{B}_1, \boldsymbol{B}_2\right)$ as

$$h_{\boldsymbol{\Theta}^\star} := \frac{1}{m}\left(\tilde{\boldsymbol{z}}_q^\top \left(\boldsymbol{W}_Q^\star \boldsymbol{W}_K^{\star\top} + \boldsymbol{B}_2\boldsymbol{B}_1^\top\right)\widetilde{\boldsymbol{Z}}_{\mathcal{M}}^\top\right)\widetilde{\boldsymbol{Z}}_{\mathcal{M}}\boldsymbol{W}_V^\star \boldsymbol{p}^\star.$$

Can LoRA Be Used to Adapt Subspace Shifts?

Theorem (Training on Single Subspace)

Suppose $\boldsymbol{w}_{\text{train}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{U}_{s,1}\boldsymbol{U}_{s,1}^\top)$ and $\boldsymbol{w}_{\text{test}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{U}_t\boldsymbol{U}_t^\top)$ with angle θ. Define the **adapted model** $h_{\boldsymbol{\Theta}^\star}\left(\widetilde{\boldsymbol{Z}}, \boldsymbol{B}_1, \boldsymbol{B}_2\right)$ as

$$h_{\boldsymbol{\Theta}^\star} := \frac{1}{m}\left(\tilde{\boldsymbol{z}}_q^\top \left(\boldsymbol{W}_Q^\star \boldsymbol{W}_K^{\star\top} + \boldsymbol{B}_2\boldsymbol{B}_1^\top\right)\widetilde{\boldsymbol{Z}}_{\mathcal{M}}^\top\right)\widetilde{\boldsymbol{Z}}_{\mathcal{M}}\boldsymbol{W}_V^\star \boldsymbol{p}^\star.$$

Then there exist $\boldsymbol{B}_1^\star, \boldsymbol{B}_2^\star \in \mathbb{R}^{(d+1)\times r}$ such that for any $\theta \in \left[0, \frac{\pi}{2}\right]$ and $\delta \in (0, r)$, if

$$m \geq n > \frac{2(r+\sigma^2+1)(r-\delta) - r}{\delta} + (r+\sigma^2+1)\sqrt{\frac{r-\delta}{\delta}},$$

we have $\lim_{\epsilon \to 0} \mathbb{E}\left[\left(\tilde{y}_{m+1} - h_{\boldsymbol{\Theta}^\star}\left(\widetilde{\boldsymbol{Z}}, \boldsymbol{B}_1^\star, \boldsymbol{B}_2^\star\right)\right)^2\right] < \sigma^2 + \delta.$

Extension to Nonlinear Function Classes

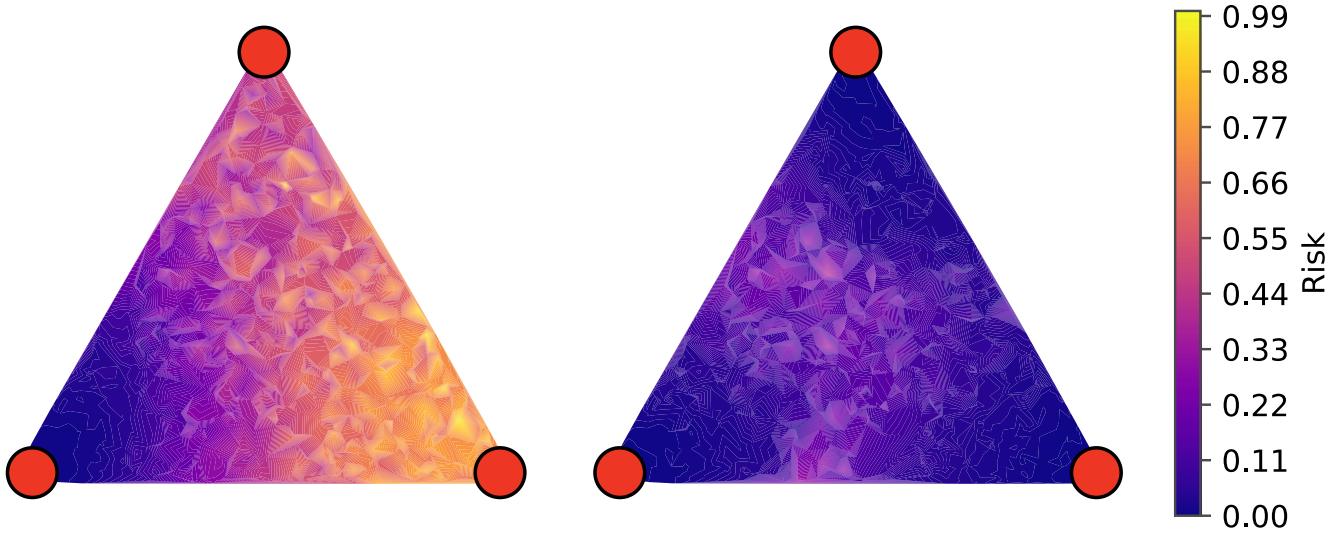


Figure 2: Left: training with a single polynomial; **Right:** with a union of basis.

Hermite Polynomials with orthogonal basis:

$$\psi_k^H(x) = \frac{(-1)^k}{\sqrt{k!}} e^{x^2/2} \frac{d^k(e^{-x^2/2})}{dx^k} \quad \text{for } k \in \mathbb{N}$$

.

Extension to Nonlinear Function Classes

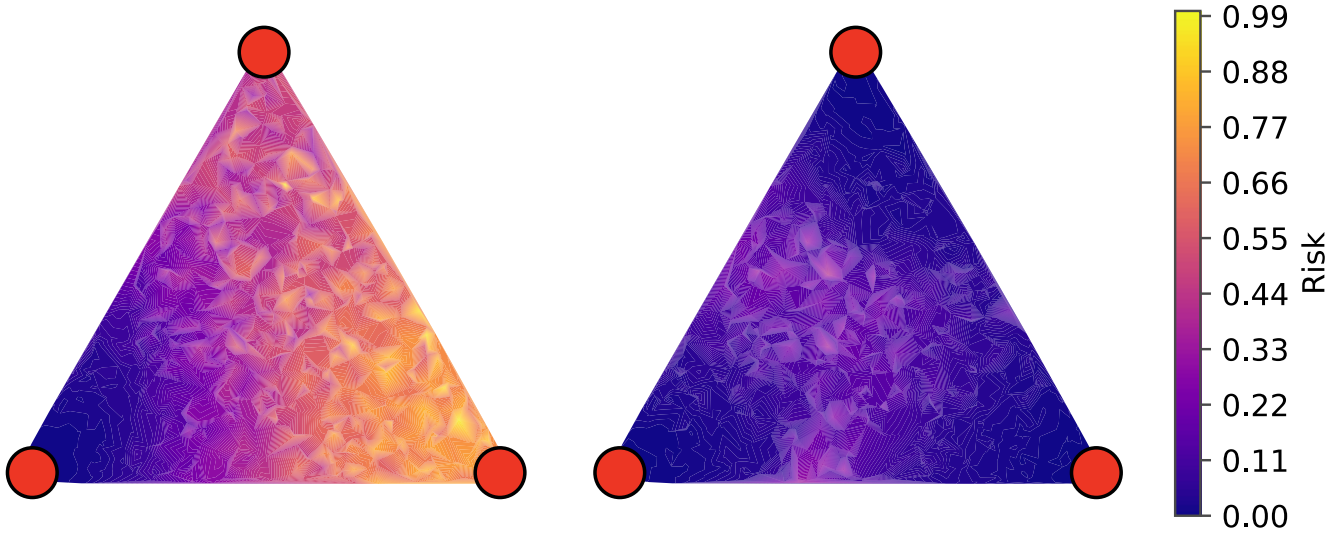


Figure 2: Left: training with a single polynomial; **Right:** with a union of basis.

Hermite Polynomials with orthogonal basis:

$$\psi_k^H(x) = \frac{(-1)^k}{\sqrt{k!}} e^{x^2/2} \frac{d^k(e^{-x^2/2})}{dx^k} \quad \text{for } k \in \mathbb{N}$$

.

Similar findings can be empirically verified for **nonlinear** function classes!

Task Diversity Enables Faster Training

The Training Loss Plateau and Two Stage Training

What does the training loss look like when we train a transformer to learn linear regression via ICL?

$$\min_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q; \boldsymbol{\Theta}) := \mathbb{E}_{\boldsymbol{Z}}\left[\left(y_{n+1} - \underbrace{g_{\boldsymbol{\Theta}}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q)}_{=\widehat{y}_{n+1}}\right)^2\right],$$

The Training Loss Plateau and Two Stage Training

What does the training loss look like when we train a transformer to learn linear regression via ICL?

$$\min_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q; \boldsymbol{\Theta}) := \mathbb{E}_{\boldsymbol{Z}} \left[\left(y_{n+1} - \underbrace{g_{\boldsymbol{\Theta}}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q)}_{=\widehat{y}_{n+1}} \right)^2 \right],$$

- Given the **linear regression**:

  $$y_i = \boldsymbol{w}^\top \boldsymbol{x}_i + \xi_i, \quad \xi_i \sim \mathcal{N}(0, \sigma^2),$$

  with task vector $\boldsymbol{w} \in \mathbb{R}^d$, and $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$.

The Training Loss Plateau and Two Stage Training

What does the training loss look like when we train a transformer to learn linear regression via ICL?

$$\min_{\Theta} \mathcal{L}(Z_{\mathcal{M}}, z_q; \Theta) := \mathbb{E}_{Z}\left[\left(y_{n+1} - \underbrace{g_{\Theta}(Z_{\mathcal{M}}, z_q)}_{=\widehat{y}_{n+1}}\right)^2\right],$$

- Given the **linear regression**:

  $$y_i = \boldsymbol{w}^\top \boldsymbol{x}_i + \xi_i, \quad \xi_i \sim \mathcal{N}(0, \sigma^2),$$

  with task vector $\boldsymbol{w} \in \mathbb{R}^d$, and $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$.
- **Two-stage training:** Training loss $\mathcal{L}$ first exhibits a "plateau", followed by a sudden drop.

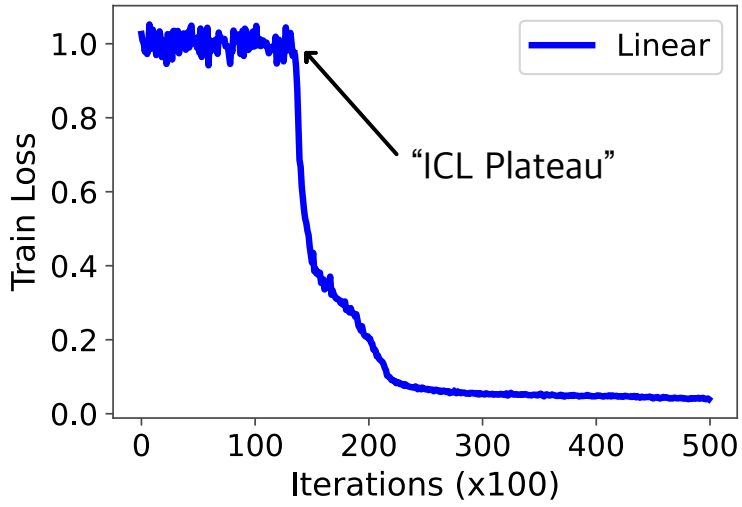


Figure 3: Training GPT-2 to learn linear regression tasks via ICL

The ICL Plateau and Training Task Diversity

- Recent work (Kim et al. 2024) empirically found **task diversity shortens the plateau**!

The ICL Plateau and Training Task Diversity

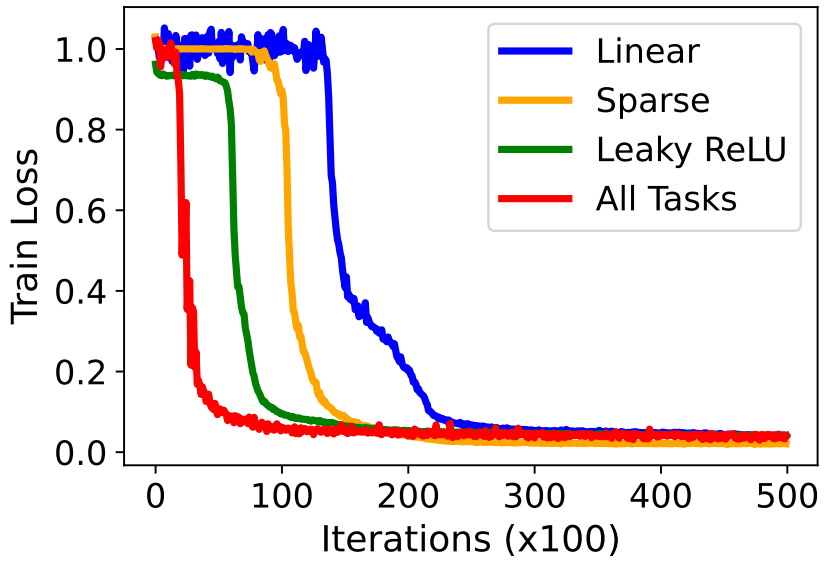
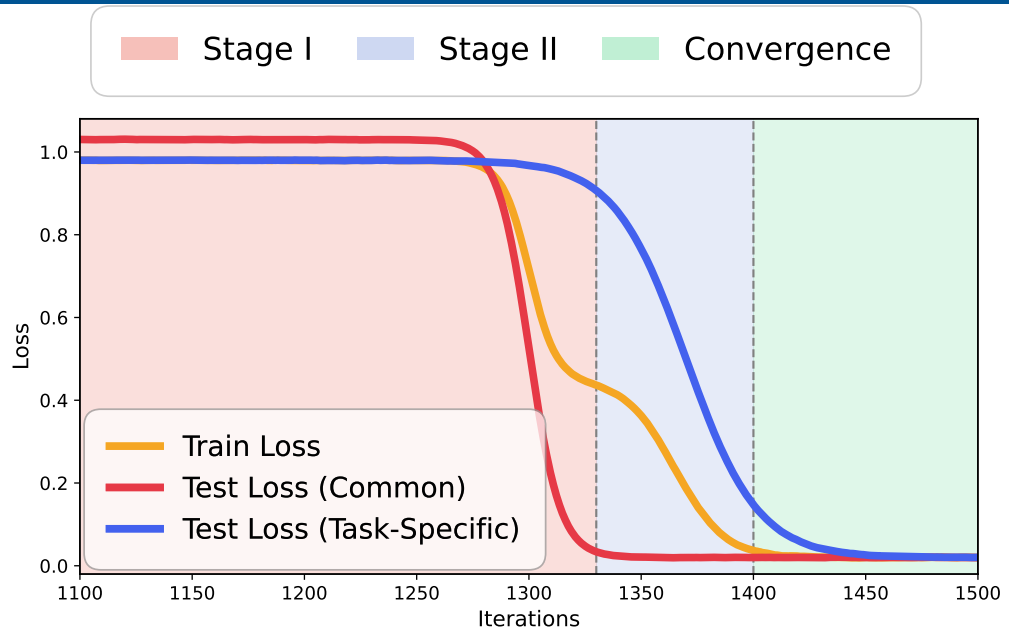


Figure 4: Training GPT-2 to learn multiple tasks via ICL

- Recent work (Kim et al. 2024) empirically found **task diversity shortens the plateau**!
- **Key Idea:** Train with **multiple (diverse) tasks** instead of just one:

$$y_i = f(\boldsymbol{x}_i) + \xi_i, \quad \text{for} \quad f \sim \begin{cases} \text{linear reg.} & \text{w.p. } 1/3 \\ \text{sparse reg.} & \text{w.p. } 1/3 \\ \text{leaky ReLU reg.} & \text{w.p. } 1/3 \end{cases}$$

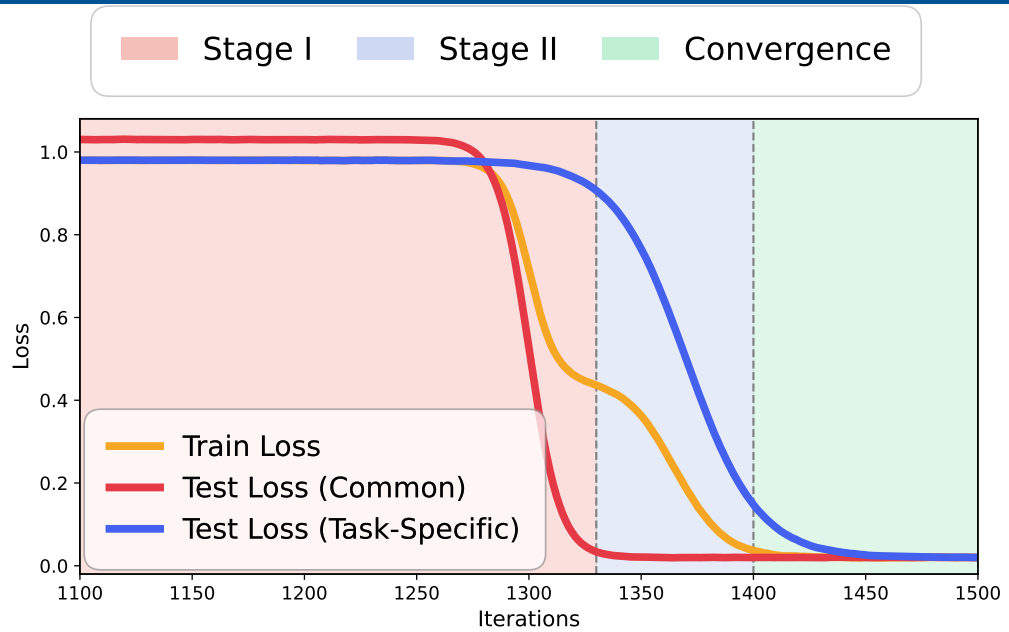
Why Training Task Diversity Shortens the Plateau?



Intuition: When tasks share common structures,

- Gradient flow **learns the shared structure first**, followed by task-specific structures.

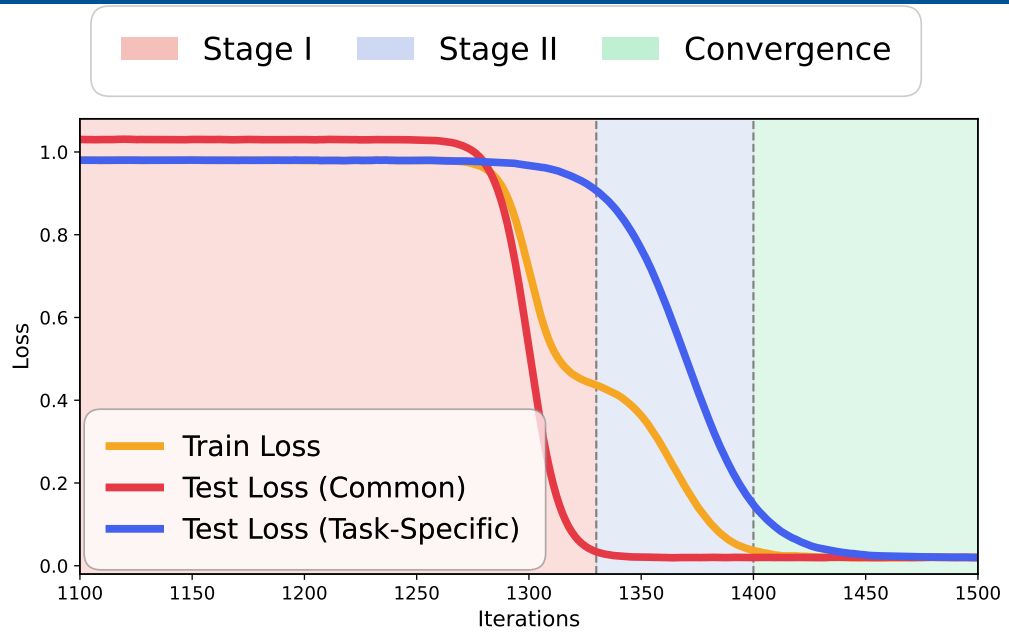
Why Training Task Diversity Shortens the Plateau?



Intuition: When tasks share common structures,

- Gradient flow **learns the shared structure first**, followed by task-specific structures.
- The training plateau is **dominated** by the difficulty of learning the shared structure.

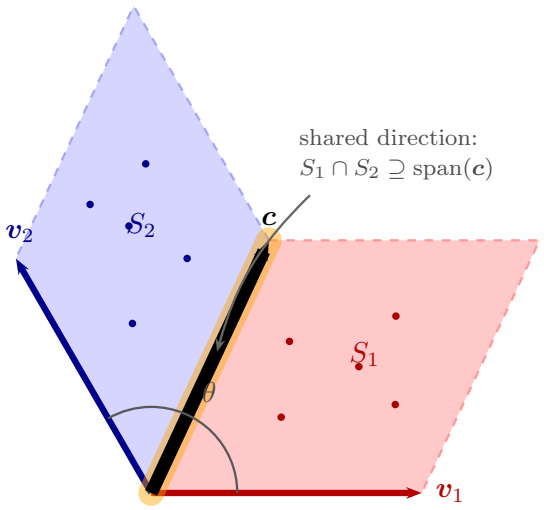
Why Training Task Diversity Shortens the Plateau?

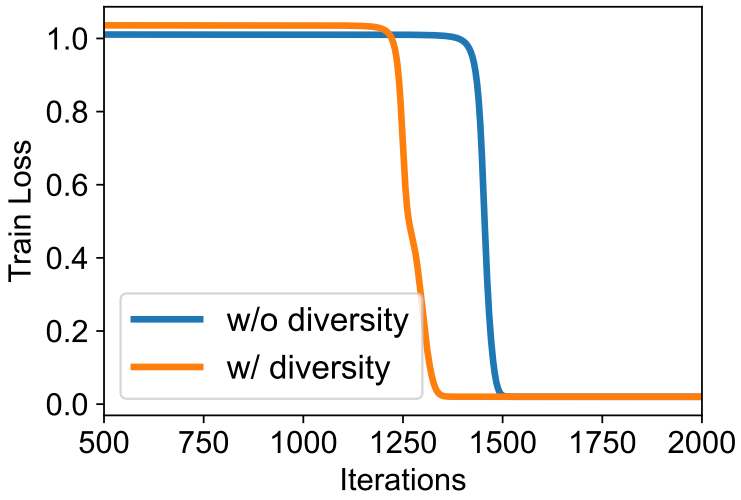


Intuition: When tasks share common structures,

- Gradient flow **learns the shared structure first**, followed by task-specific structures.
- The training plateau is **dominated** by the difficulty of learning the shared structure.
- More tasks ($K \uparrow$) **amplify** the shared structure, thereby accelerating training.

Why Training Task Diversity Shortens the Plateau?





Intuition: When tasks share common structures,

- Gradient flow **learns the shared structure first**, followed by task-specific structures.
- The training plateau is **dominated** by the difficulty of learning the shared structure.
- More tasks ($K \uparrow$) **amplify** the shared structure, thereby accelerating training.

How to Justify Our Intuition Theoretically?

Again, consider the **linear regression** problem

$$y = \boldsymbol{w}_{\text{train}}^{\top} \boldsymbol{x} + \xi.$$

with $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$, $\xi \sim \mathcal{N}(0, \sigma^2)$.

How to Justify Our Intuition Theoretically?

Again, consider the **linear regression** problem

$$y = \boldsymbol{w}_{\text{train}}^\top \boldsymbol{x} + \xi.$$

with $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$, $\xi \sim \mathcal{N}(0, \sigma^2)$.

- **Case 1: diverse training data with shared structures.**

$$\boldsymbol{w}_{\text{train}} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,1}) & \text{w.p. } 1/2 \\ \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,2}) & \text{w.p. } 1/2 \end{cases} \quad \text{and} \quad \boldsymbol{\Sigma}_{s,k} = \boldsymbol{U}_{s,k}^\top \boldsymbol{U}_{s,k},$$

where $q = \dim(\mathcal{V})$ with $\mathcal{V} = \operatorname{span}(\boldsymbol{U}_{s,1}, \boldsymbol{U}_{s,2})$ and

$$n_{\text{com}} := \dim\left(\operatorname{span}\left(\boldsymbol{U}_{s,1}\right) \cap \operatorname{span}\left(\boldsymbol{U}_{s,2}\right)\right) > 0.$$

How to Justify Our Intuition Theoretically?

Again, consider the **linear regression** problem

$$y = \boldsymbol{w}_{\text{train}}^{\top} \boldsymbol{x} + \xi.$$

with $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$, $\xi \sim \mathcal{N}(0, \sigma^2)$.

- **Case 1: diverse training data with shared structures.**

$$\boldsymbol{w}_{\text{train}} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,1}) & \text{w.p. } 1/2 \\ \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,2}) & \text{w.p. } 1/2 \end{cases} \quad \text{and} \quad \boldsymbol{\Sigma}_{s,k} = \boldsymbol{U}_{s,k}^{\top} \boldsymbol{U}_{s,k},$$

  where $q = \dim(\mathcal{V})$ with $\mathcal{V} = \operatorname{span}(\boldsymbol{U}_{s,1}, \boldsymbol{U}_{s,2})$ and

$$n_{\text{com}} := \dim\left(\operatorname{span}\left(\boldsymbol{U}_{s,1}\right) \cap \operatorname{span}\left(\boldsymbol{U}_{s,2}\right)\right) > 0.$$

- Define the **common task ratio**:

$$\kappa \;:=\; \frac{n_{\text{com}}}{q} \;\in\; [0, 1],$$

  which measures the fraction of $\mathcal{V}$ shared by the two tasks.

How to Justify Our Intuition Theoretically?

Consider the **linear regression** problem

$$y = \boldsymbol{w}_{\text{train}}^\top \boldsymbol{x} + \xi.$$

with $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$, $\xi \sim \mathcal{N}(0, \sigma^2)$.

- **Case 1: diverse training tasks with shared structures.**

$$\boldsymbol{w}_{\text{train}} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,1}) & \text{w.p. } 1/2 \\ \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{s,2}) & \text{w.p. } 1/2 \end{cases} \quad \text{and} \quad \boldsymbol{\Sigma}_{s,k} = \boldsymbol{U}_{s,k}^\top \boldsymbol{U}_{s,k},$$

where $q = \dim(\mathcal{V})$ with $\mathcal{V} = \operatorname{span}(\boldsymbol{U}_{s,1}, \boldsymbol{U}_{s,2})$ and

$$n_{\text{com}} := \dim\left(\operatorname{span}\left(\boldsymbol{U}_{s,1}\right) \cap \operatorname{span}\left(\boldsymbol{U}_{s,2}\right)\right) > 0.$$

- **Case 2: Single training task with no shared structures.**

$$\boldsymbol{w}_{\text{train}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_s\right) \quad \text{and} \quad \boldsymbol{\Sigma}_s = \boldsymbol{U}_s \boldsymbol{U}_s^\top$$

where $\boldsymbol{U}_s \in \mathbb{R}^{d \times q}$ with $\operatorname{span}(\boldsymbol{U}_s) = \mathcal{V}$ of the **same dimension** q.

Studying the Gradient Flow From Small Initializations

- **Compare the convergence time** t to stationary points from initializations: (i) **diverse-task** t_{diverse} vs. (ii) **single-task** t_{single}.

Studying the Gradient Flow From Small Initializations

- **Compare the convergence time** t to stationary points from initializations: (i) **diverse-task** t_{diverse} vs. (ii) **single-task** t_{single}.
- Consider **gradient flow dynamics** of optimizing the loss

$$\min_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q; \boldsymbol{\Theta}) := \mathbb{E}_{\boldsymbol{Z}}\left[\left(y_{n+1} - g_{\boldsymbol{\Theta}}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q)\right)^2\right],$$

for $\left(\dot{\boldsymbol{W}}_Q(t), \dot{\boldsymbol{W}}_K(t), \dot{\boldsymbol{W}}_V(t)\right)$ from **small initializations**

$$\boldsymbol{W}_Q(0) = \boldsymbol{W}_K(0) = \alpha \cdot \boldsymbol{I}_{d+1}, \qquad \boldsymbol{W}_V(0) = \phi \cdot \boldsymbol{I}_{d+1}.$$

where $\alpha > 0$ is a small constant and $\phi = \alpha\sqrt{q}$.

Task Diversity Accelerates Training

Theorem (Task Diversity Accelerates Training)

Compare training for **Case I (diverse-task)** and **Case II (single-task)**, and define the **common task ratio** of the two tasks:

$$\kappa \coloneqq \frac{n_{\text{com}}}{q} \in [0,1],$$

which measures the fraction of $\mathcal{V}$ shared by the two tasks. Consider the convergence times t_{diverse} (**Case I**) and t_{single} (**Case II**) for the gradient flow dynamics, we have

$$\boxed{\frac{t_{\text{diverse}}}{t_{\text{single}}} \approx \rho(\kappa) \coloneqq \frac{1+\kappa}{1+\sqrt{\kappa}}.}$$

Task Diversity Accelerates Training

Theorem (Task Diversity Accelerates Training)

Compare training for **Case I (diverse-task)** and **Case II (single-task)**, and define the **common task ratio** of the two tasks:

$$\kappa \coloneqq \frac{n_{\text{com}}}{q} \in [0,1],$$

which measures the fraction of $\mathcal{V}$ shared by the two tasks. Consider the convergence times t_{diverse} (**Case I**) and t_{single} (**Case II**) for the gradient flow dynamics, we have

$$\boxed{\frac{t_{\text{diverse}}}{t_{\text{single}}} \approx \rho(\kappa) \coloneqq \frac{1+\kappa}{1+\sqrt{\kappa}}.}$$

- **Training acceleration:** $\rho(\kappa) < 1$ for all $\kappa \in (0,1)$.

Task Diversity Accelerates Training

Theorem (Task Diversity Accelerates Training)

Compare training for **Case I (diverse-task)** and **Case II (single-task)**, and define the **common task ratio** of the two tasks:

$$\kappa \coloneqq \frac{n_{\text{com}}}{q} \in [0,1],$$

which measures the fraction of $\mathcal{V}$ shared by the two tasks. Consider the convergence times t_{diverse} (**Case I**) and t_{single} (**Case II**) for the gradient flow dynamics, we have

$$\boxed{\frac{t_{\text{diverse}}}{t_{\text{single}}} \approx \rho(\kappa) \coloneqq \frac{1+\kappa}{1+\sqrt{\kappa}}.}$$

- **Training acceleration:** $\rho(\kappa) < 1$ for all $\kappa \in (0,1)$.
- **Optimal diversity** $\kappa_\star = 3 - 2\sqrt{2}$ implies a 17% time reduction, indicating more significant acceleration for $K \gg 2$.

Key Insights from Gradient Flow Dynamics

- **Model simplifications.** From **small identity initializations**,

$$g_{\Theta}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q) = v \cdot \boldsymbol{x}_{n+1}^{\top} \boldsymbol{Q} \boldsymbol{K}^{\top} \boldsymbol{c}, \quad \text{where} \quad \boldsymbol{c} = \frac{1}{n} \sum_{i=1}^{n} y_i \boldsymbol{x}_i,$$

  where $v \in \mathbb{R}$ and $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{d \times d}$.

Key Insights from Gradient Flow Dynamics

- **Model simplifications.** From **small identity initializations,**

$$g_{\Theta}(\boldsymbol{Z}_{\mathcal{M}}, \boldsymbol{z}_q) = v \cdot \boldsymbol{x}_{n+1}^{\top} \boldsymbol{Q} \boldsymbol{K}^{\top} \boldsymbol{c}, \quad \text{where} \quad \boldsymbol{c} = \frac{1}{n} \sum_{i=1}^{n} y_i \boldsymbol{x}_i,$$

  where $v \in \mathbb{R}$ and $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{d \times d}$.

- **Gradient flow dynamics.** Suppose $\boldsymbol{\Sigma} = \mathbb{E}\left[\boldsymbol{w}_{\text{train}} \boldsymbol{w}_{\text{train}}^{\top}\right] = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{U}^{\top}$

$$\underline{\dot{\boldsymbol{Q}}} := \lim_{n \to \infty} \dot{\boldsymbol{Q}} = \frac{1}{\tau} \boldsymbol{U} \underbrace{\left(v \left(\boldsymbol{\Lambda} - v \boldsymbol{\Lambda}_Q \boldsymbol{\Lambda}_K \boldsymbol{\Lambda}_s\right) \boldsymbol{\Lambda}_K\right)}_{\dot{\boldsymbol{\Lambda}}_Q} \boldsymbol{U}^{\top},$$

$$\underline{\dot{\boldsymbol{K}}} := \lim_{n \to \infty} \dot{\boldsymbol{K}} = \frac{1}{\tau} \boldsymbol{U} \underbrace{\left(v \boldsymbol{\Lambda}_Q \left(\boldsymbol{\Lambda} - v \boldsymbol{\Lambda}_Q \boldsymbol{\Lambda}_K \boldsymbol{\Lambda}\right)\right)}_{\dot{\boldsymbol{\Lambda}}_K} \boldsymbol{U}^{\top},$$

$$\underline{\dot{v}} := \lim_{n \to \infty} \dot{v} = \frac{1}{\tau} \operatorname{tr}\left(\boldsymbol{\Lambda}_Q \left(\boldsymbol{\Lambda} - v \boldsymbol{\Lambda}_Q \boldsymbol{\Lambda}_K \boldsymbol{\Lambda}\right) \boldsymbol{\Lambda}_K\right).$$

Key Insights from Gradient Flow Dynamics

Because of the shared basis $\boldsymbol{U}$ between $\boldsymbol{Q}$ and $\boldsymbol{K}$, it suffices to track how **eigenvalues** λ_i of $\boldsymbol{Q}, \boldsymbol{K}$ evolve:

$$\underline{\dot{\lambda}}_i = \left(\sigma_i - v\sigma_i\lambda_i^2\right) \cdot v\lambda_i, \quad \underline{\dot{v}} = \frac{1}{\tau}\sum_{i=1}^{d}\left(\sigma_i - v\sigma_i\lambda_i^2\right) \cdot \lambda_i^2, \quad \forall i \in [d].$$

where σ_i is the i-th diagonal entry of $\boldsymbol{\Lambda}$, **depending on the training diversity**.

Key Insights from Gradient Flow Dynamics

Because of the shared basis U between Q and K, it suffices to track how **eigenvalues** λ_i of Q, K evolve:

$$\dot{\lambda}_i = \left(\sigma_i - v\sigma_i\lambda_i^2\right) \cdot v\lambda_i, \quad \dot{v} = \frac{1}{\tau}\sum_{i=1}^{d}\left(\sigma_i - v\sigma_i\lambda_i^2\right) \cdot \lambda_i^2, \quad \forall i \in [d].$$

where σ_i is the i-th diagonal entry of Λ, **depending on the training diversity**.

- **For Case I with diverse tasks**, task diversity with $0 < \kappa < 1$ **amplifies** σ_i of the **shared principal directions**, leading to faster convergence.

Key Insights from Gradient Flow Dynamics

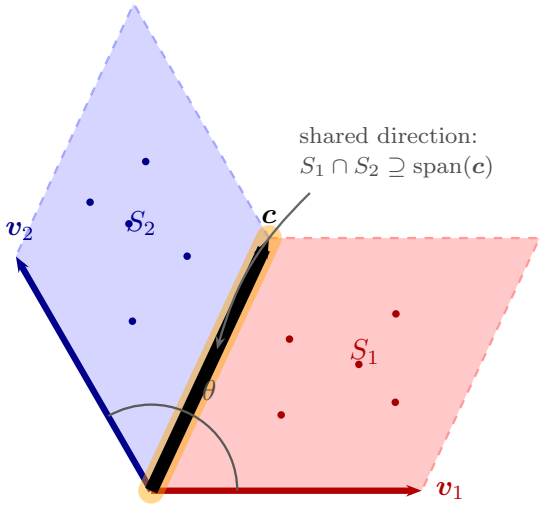
Because of the shared basis U between Q and K, it suffices to track how **eigenvalues** λ_i of Q, K evolve:

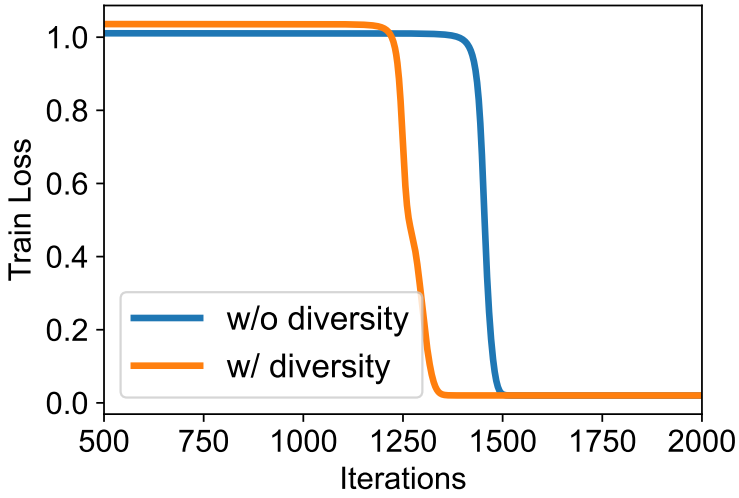
$$\underline{\dot{\lambda}}_i = \left(\sigma_i - v\sigma_i\lambda_i^2\right) \cdot v\lambda_i, \quad \underline{\dot{v}} = \frac{1}{\tau}\sum_{i=1}^{d}\left(\sigma_i - v\sigma_i\lambda_i^2\right) \cdot \lambda_i^2, \quad \forall i \in [d].$$

where σ_i is the i-th diagonal entry of Λ, **depending on the training diversity**.

- **For Case I with diverse tasks**, task diversity with $0 < \kappa < 1$ **amplifies** σ_i of the **shared principal directions**, leading to faster convergence.
- **For Case II with a single task**, it weighs all principal directions σ_i **equally** with **uniform** slow convergence.

Key Insights from Gradient Flow Dynamics

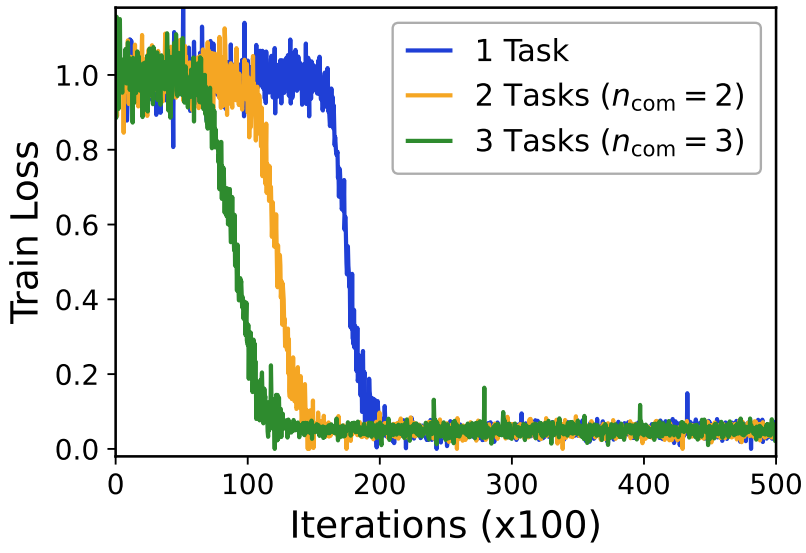


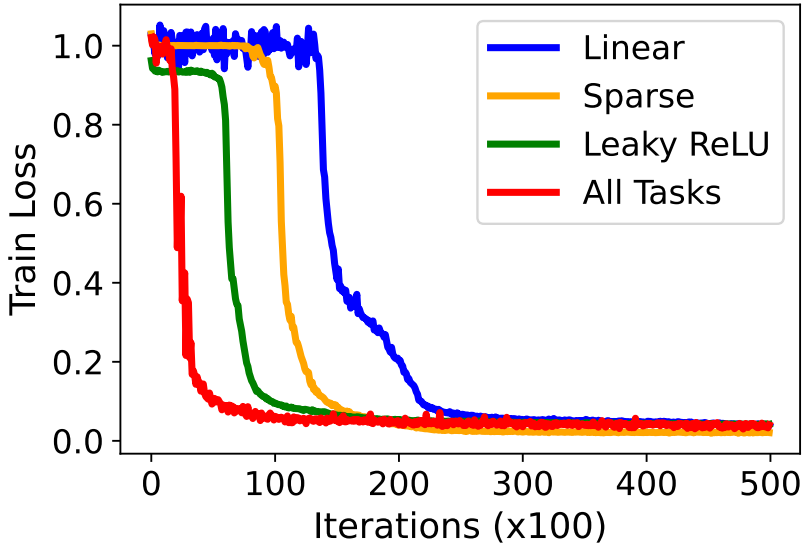


- **For Case I with diverse tasks**, task diversity with $0 < \kappa < 1$ **amplifies** σ_i of the **shared principal directions**, leading to faster convergence.
- **For Case II with a single task**, it weighs all principal directions σ_i **equally** with **uniform** slow convergence.

Experimental Results

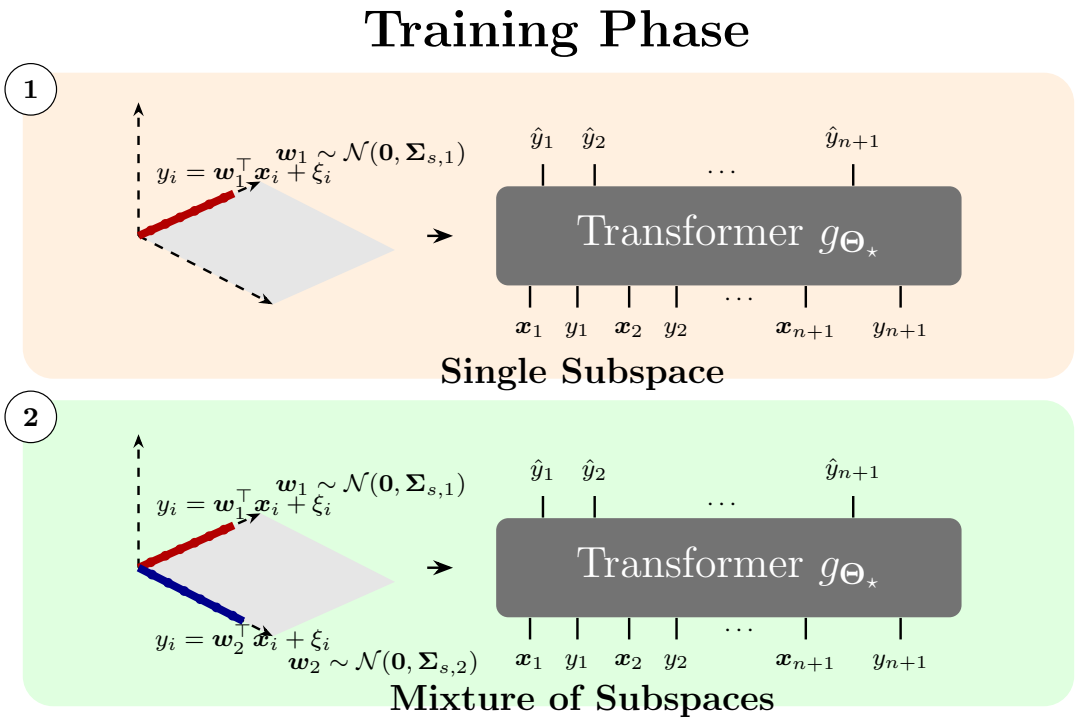
Training GPT-2 Model for ICL under different settings:

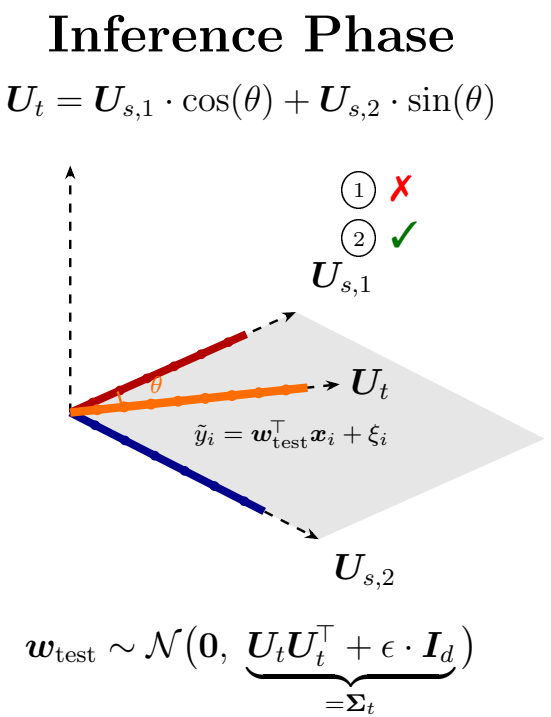




Conclusion

Take-Home Messages (Without Task Overlap)



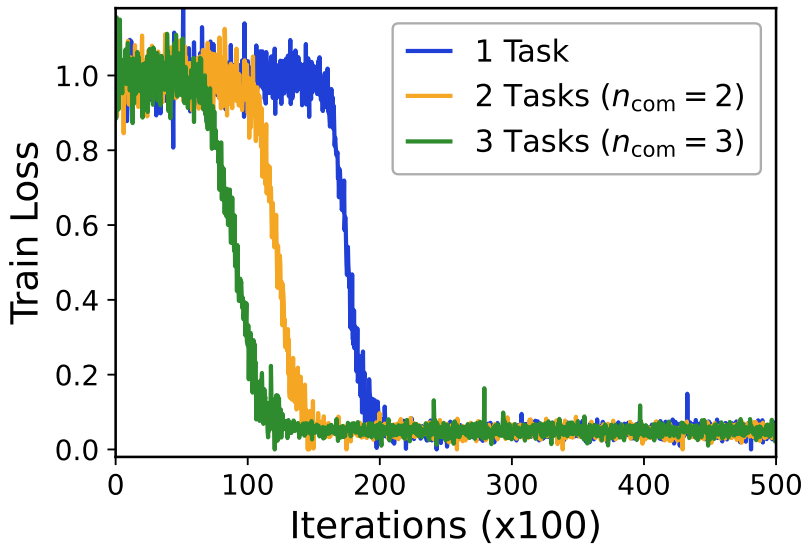


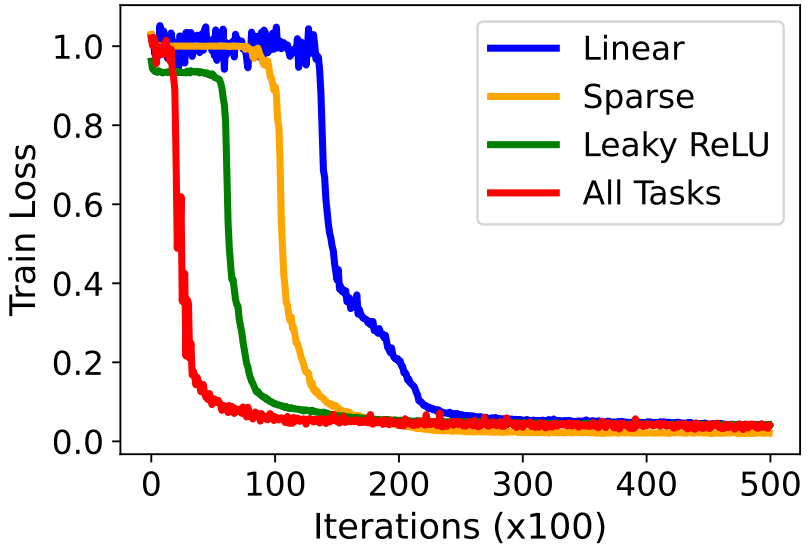
Training task diversity enables OOD generalization!

References:

- Kwon et al. Out-of-Distribution Generalization of In-Context Learning: A Low-Dimensional Subspace Perspective. AISTATS'26. (**Best paper** at CauScale Workshop'26).

Take-Home Messages (With Task Overlap)





Training task diversity accelerates convergence!

References:

- Kwon et al. The Effect of Training Task Diversity for In-Context Learning: A Subspace Analysis. arXiv 2606.06814 (Under review at ACM Journal of Data Science), 2026.

Acknowledgments



Laura Balzano
University of Michigan



Soo-Min Kwon
Microsoft



Dogyoon Song
UC Davis



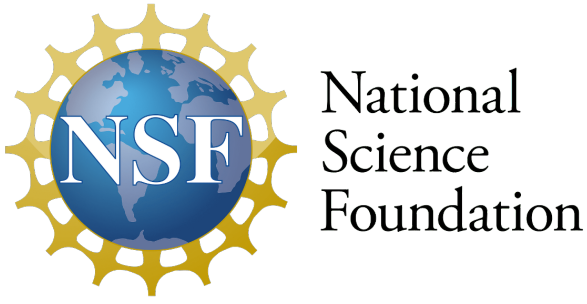
Alec Xu
University of Michigan



Can Yaras
Princeton University

Acknowledgements













Thank You!