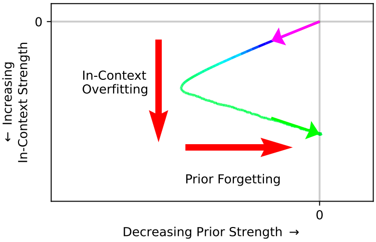
Prior Forgetting and In-Context Overfitting

Sungyoon Lee

Department of Computer Science, Hanyang University



NeurIPS 2025

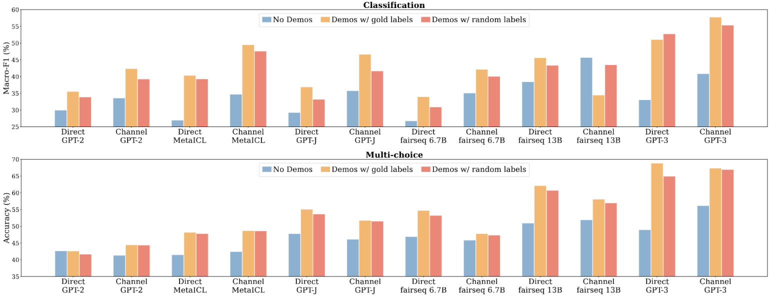
What Makes LLMs Successful (In-Context Learning; ICL)

$$Z = \left[\underbrace{\begin{matrix} \text{sea otter} & \text{peppermint} & \text{plush giraffe} \\ \downarrow & \downarrow & \downarrow \\ \text{loutre de mer} & \text{menthe poivrée} & \text{girafe peluche} \end{matrix}}_{\text{demos}} \;\; \underbrace{\begin{matrix} \text{cheese} \\ \downarrow \\ ? \end{matrix}}_{\text{query}} \right]$$

$Z \mapsto T_\theta(Z) \leftrightarrow$ "fromage" ("cheese" in French)

In-Context Learning (ICL): a model's ability to learn (without updating any model parameter) and perform a new unseen task from a few demonstrations of input-output pairs given at test time.

Do Demos Matter?



No Demos $<$ Good Demos $\approx$ Bad Demos [Min+22]

[Min+22; Lyu+23; Yoo+22; Wei+23; Shi+23]

ICL = TR + TL

-	(pre)training	inference (forward pass)
classical(?) DNN	feature learning	feature extraction
Transformer	TR+TL learning	ICL ("*weight(?)*" update)

- TR (Task Recognition): the model recalls similar functions and concepts learned (priorly) in the pretraining phase
- TL (Task Learning): the model smoothly adapts to and implicitly learns the (observed) in-context task.

How do ICL abilities (TR+TL)
emerge (and disappear) during pretraining?

Overview

- Simple Dynamics
  - in-context linear regression
  - single-layer linear self-attention model $\rightarrow$ **two-parameter transformer**
- TR-TL Decomposition
  - **demonstration-query task independence**
  - **noncentral task distribution**
- New Phenomena
  - **Prior Forgetting**
  - **In-Context Overfitting**

In-Context "Linear Regression"

Task: English-to-French Translation

$$Z = \begin{bmatrix} \text{sea otter} & \text{peppermint} & \text{plush giraffe} & \text{cheese} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \text{loutre de mer} & \text{menthe poivrée} & \text{girafe peluche} & ? \end{bmatrix}$$

$$Z \mapsto T_\theta(Z) \leftrightarrow \text{"fromage" ("cheese" in French)}$$

Task: $x \mapsto w^\top x$

$$Z = \begin{bmatrix} x^{(1)} & x^{(2)} & \dots & x^{(n)} & x^{(n+1)} \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ y^{(1)}{}_{(= w^\top x^{(1)})} & y^{(2)} & \dots & y^{(n)} & 0 \end{bmatrix}$$

$$Z \mapsto T_\theta(Z) \leftrightarrow w^\top x^{(n+1)} \text{ (target)}$$

Single-Layer Transformer w/ "Linear" Self-Attention (LSA)

$$Z \mapsto T_\theta(Z) = -\left[Z + \frac{1}{n}\mathrm{LSA}_\theta(Z)\right]_{-1,-1}$$

$$\begin{aligned}\mathrm{LSA}_\theta({\color{red}Z}) &= \underbrace{W_V {\color{red}Z}}_{\text{Value}} \underbrace{M}_{\text{Mask}} \cancel{\mathrm{softmax}}((\underbrace{W_Q {\color{red}Z}}_{\text{Query}})^\top \underbrace{W_K {\color{red}Z}}_{\text{Key}}) \\ &= \underbrace{W_V}_{P} {\color{red}Z} M {\color{red}Z}^\top \underbrace{W_Q^\top W_K}_{Q} {\color{red}Z}\end{aligned}$$

$$P = \begin{bmatrix} 0_{d\times d} & 0_d \\ p^\top & \kappa \end{bmatrix}, Q = \begin{bmatrix} \bar{Q} & 0_d \\ q^\top & 0 \end{bmatrix}$$

Here, we put the first d rows $P_{1:d,:}$ of P and the last column $Q_{:,-1}$ of Q as 0 as they do not affect the output.

Two-Parameter Transformer

$$\mathrm{LSA}_\theta(Z) = \underbrace{W_V}_{P} ZMZ^\top \underbrace{W_Q^\top W_K}_{Q} Z$$

$$P = \begin{bmatrix} 0_{d\times d} & 0_d \\ p^\top & \kappa \end{bmatrix} = \begin{bmatrix} 0_{d\times d} & 0_d \\ \alpha\mu^\top & \kappa \end{bmatrix} \qquad (\underbrace{\alpha}_{\text{TR}}, \underbrace{\kappa}_{\text{TL}} \in \mathbb{R})$$

$$Q = \begin{bmatrix} \bar{Q} & 0_d \\ q^\top & 0 \end{bmatrix} = \begin{bmatrix} I_d & 0_d \\ 0_d^\top & 0 \end{bmatrix}$$

Demonstration-Query Task Independence (Concept Shift)

During pretraining, we assume

$$w = w_q \sim_{iid} \mathcal{D}_{\mathcal{W}}.$$

Task: $x \mapsto {w_q}^\top x$

Good Demos

$$Z = \begin{bmatrix} \boxed{\begin{matrix} x^{(1)} & x^{(2)} & \cdots & x^{(n)} \\ \downarrow & \downarrow & & \downarrow \\ w^\top x^{(1)} & w^\top x^{(2)} & \cdots & w^\top x^{(n)} \end{matrix}} & \begin{matrix} x^{(n+1)} \\ \downarrow \\ 0 \end{matrix} \end{bmatrix}$$

$Z \mapsto T_\theta(Z) \leftrightarrow {w_q}^\top x^{(n+1)}$

Demonstration-Query Task Independence (Concept Shift)

At test time, we assume

$$w, w_q \sim_{iid} \mathcal{D}_{\mathcal{W}}.$$

Task: $x \mapsto {w_q}^\top x$

Bad Demos

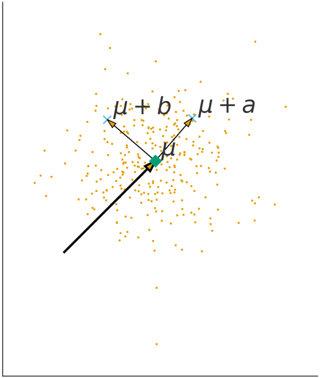
$$Z = \left[\boxed{\begin{array}{cccc} x^{(1)} & x^{(2)} & \dots & x^{(n)} \\ \downarrow & \downarrow & & \downarrow \\ w^\top x^{(1)} & w^\top x^{(2)} & \dots & w^\top x^{(n)} \end{array}} \begin{array}{c} x^{(n+1)} \\ \downarrow \\ 0 \end{array}\right]$$

$$Z \mapsto T_\theta(Z) \leftrightarrow {w_q}^\top x^{(n+1)}$$

Noncentral Task Distribution

The task center μ explains the prior task distribution in the sense that $w = \mu + a$ and $w_q = \mu + b$ **share the (non-zero) prior knowledge** μ and **they have their own knowledge** a **and** b.

$$w, w_q \sim_{iid} \mathcal{N}(\mu, \sigma^2 I)$$



Quadratic Training Objective w/ Two Parameters

Under the two-parameter model with $\theta = [\alpha, \kappa]^\top$, we have

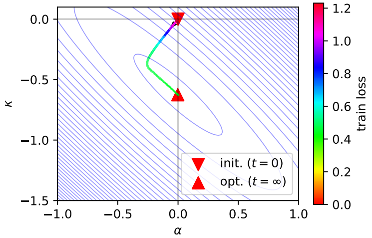
$$T_\theta(Z) = \hat{w}^\top x^{(n+1)} \leftrightarrow w_q^\top x^{(n+1)}$$

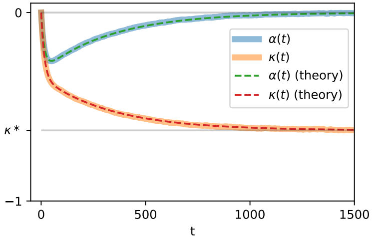
$$\hat{w} \propto \underbrace{\alpha\mu}_{\text{shared}} + \underbrace{\kappa w}_{\text{context-specific}} .$$

The training objective

$$\begin{aligned} L_{\text{train}}(\theta) &\equiv \mathbb{E}_{w,X}\left[\left(w^\top x^{(n+1)} - T_\theta(Z)\right)^2\right] \\ &= \mathbb{E}_{w,X}\left[\left((w - \hat{w})^\top x^{(n+1)}\right)^2\right] \end{aligned}$$

is **quadratic** wrt $\theta = [\alpha, \kappa]^\top$.





$$\hat{w} \propto \underbrace{\alpha\mu}_{\text{shared}} + \underbrace{\kappa w}_{\text{context-specific}}$$

- initial phase
  - in-context strength $|\kappa| \uparrow$
  - prior strength $|\alpha| \uparrow$
- later phase
  - in-context strength $|\kappa| \uparrow$: **in-context overfitting**
  - prior strength $|\alpha| \downarrow \mathbf{0}$: **prior forgetting**

Prior Forgetting and In-Context Overfitting

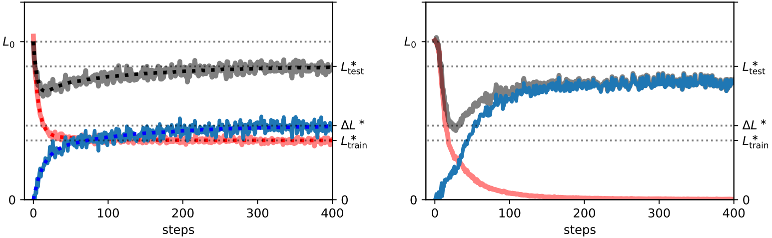
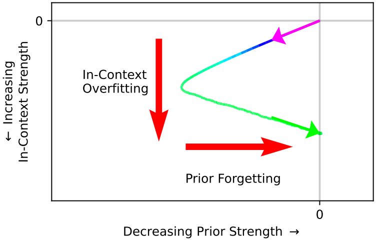
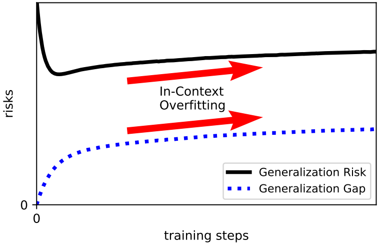


Figure: Left: two-parameter model, Right: practical Transformer

Training loss ($w = w_q$) monotonically decreases, but **the gap increases** and **test loss ($w \neq w_q$) shows a u-shape curve**.

Summary

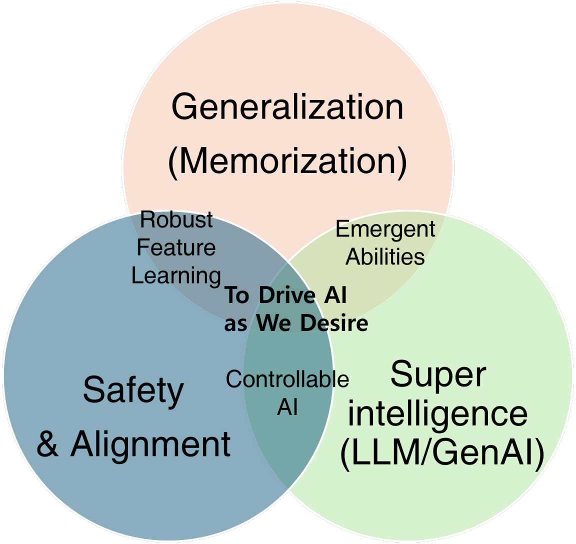




$$\hat{w} \propto \alpha\mu + \kappa w$$

- $\alpha\mu$: to learn a shared concept (TR)
- κw: to learn a given task (TL)

Sungyoon Lee



References

[Lyu+23] Xinxi Lyu et al. "Z-ICL: Zero-Shot In-Context Learning with Pseudo-Demonstrations". In: *The 61st Annual Meeting Of The Association For Computational Linguistics*. 2023.

[Min+22] Sewon Min et al. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11048–11064. DOI: 10.18653/v1/2022.emnlp-main.759. URL: https://aclanthology.org/2022.emnlp-main.759.

[Shi+23] Freda Shi et al. "Large language models can be easily distracted by irrelevant context". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 31210–31227.

[Wei+23] Jerry Wei et al. "Larger language models do in-context learning differently". In: *arXiv preprint arXiv:2303.03846* (2023).

[Yoo+22] Kang Min Yoo et al. "Ground-Truth Labels Matter: A Deeper Look into Input-Label Demonstrations". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 2422–2437.