A Tutorial of Small Language Models in the Era of Large Language Models

Fali Wang[1], Minhua Lin[1], Yao Ma[2], Hui Liu[3], Qi He[3], Xianfeng Tang[3], Jiliang Tang[4], Jian Pei[5], Suhang Wang[1]

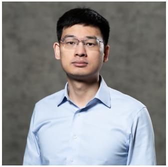
[1] The Pennsylvania State University [2] Rensselaer Polytechnic Institute [3] Amazon [4] Michigan State University [5] Duke University

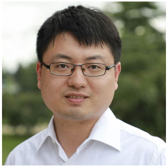
August 13, 2025 *(KDD 2025)*

Team behind the Tutorial


Fali Wang

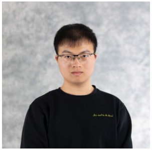

Minhua Lin


Yao Ma


Suhang Wang


Hui Liu


Qi He


Xianfeng Tang


Jiliang Tang


Jian Pei

Related Materials

- Paper: arXiv
- Github: Github
- English Blog: in Linkin
- Chinese Blog: in Wechat
- Slides are in the link (https://fairyfali.github.io/kdd2025-tutorial/).

GitHub Repo

Website

Why Small Language Models (SLMs)?

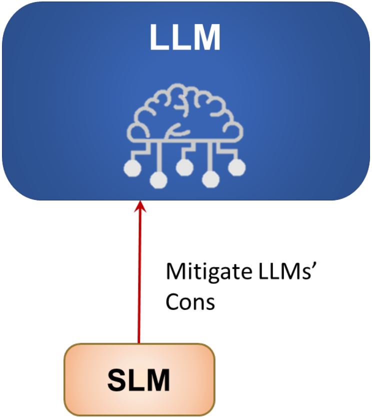
Pros:

- □ Emergent ability
- □ Generalizability

Cons:

- □ Privacy leakage
- □ On-device deployment
- □ Inference latency
- □ Expensive fine-tuning
- □ Inferior to specialized models

Why Small Language Models (SLMs)?



Pros:

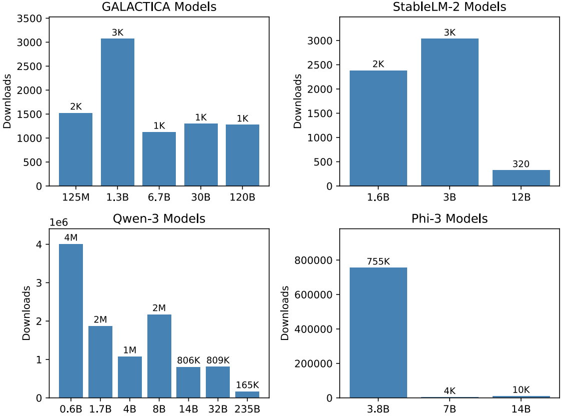
- Emergent ability
- Generalizability

Cons:

- Privacy leakage
- On-device deployment
- Inference latency
- Expensive fine-tuning
- Inferior to specialized models

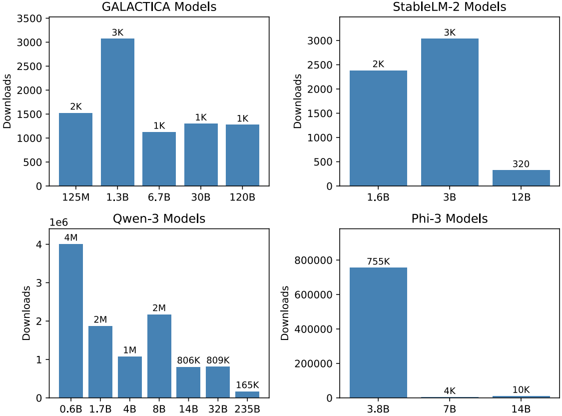
SLMs serve a different set of needs—they open up new possibilities where LLMs fall short.

Smaller Language Models are Popular



Download Statistics obtained from HuggingFace Community on July 26, 2025.

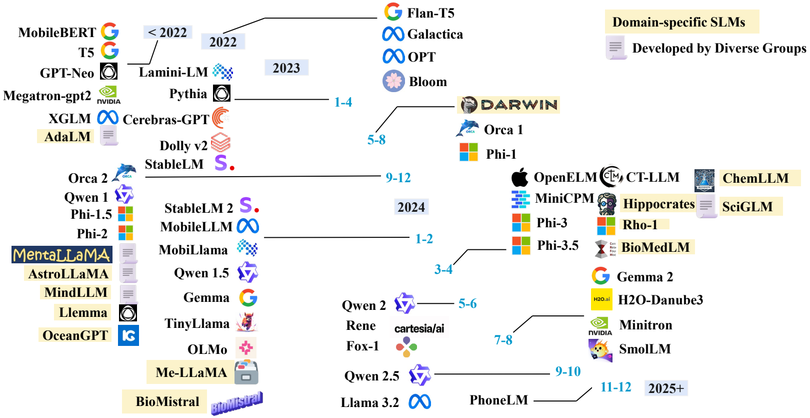
Smaller Language Models are Popular



Download Statistics obtained from HuggingFace Community on July 26, 2025.

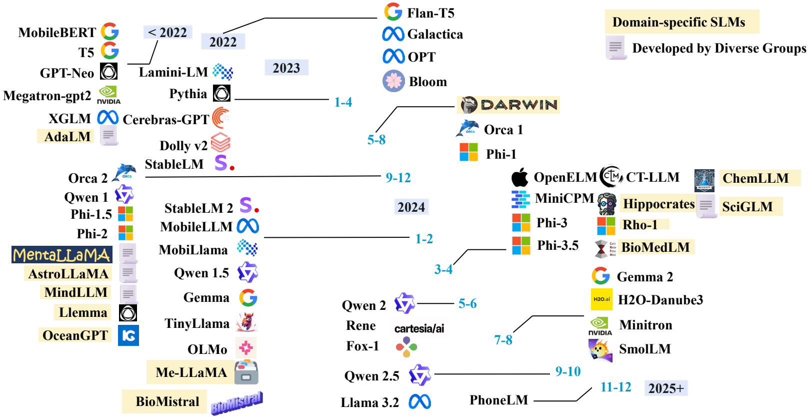
- SLMs are being downloaded more frequently than LLMs by a large margin.
- The demand for smaller, more efficient models is real and growing.

Timeline of Existing SLMs



Evolution of small language models over time

Timeline of Existing SLMs



Evolution of small language models over time

- There is a steady and accelerating stream of SLMs
- SLMs are actively evolving as a research and engineering frontier.

What is SLM? –Existing SLM Definition

- **Relative Definition**: Some[1] [2] [3] view "small" as relative to "large", i.e., anything smaller than current LLMs is "small".

[1] Zhenyan Lu et al. *Small language models: Survey, measurements, and insights.* arXiv 2024.

[2] Van Nguyen et al. *A Survey of Small Language Models*. arXiv 2024.

[3] Lihu Chen et al. *What is the role of small models in the llm era: A survey.* arXiv 2024.

[4] Zechun Liu et al. *MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases.* ICML 2024.

[5] Nagesh Mashette. *The Rise of Small Language Models: Efficiency and Customization for AI*. Blog 2023.

[6] Fu, Yao et al. *Specializing smaller language models towards multi-step reasoning.* ICML 2023.

What is SLM? –Existing SLM Definition

- **Relative Definition**: Some[1] [2] [3] view "small" as relative to "large", i.e., anything smaller than current LLMs is "small".
- **Perspective of Mobile Devices**: MobileLLM[4] categorizes SLMs as models with fewer than one billion parameters, suitable for mobile devices with up to 6GB memory.

[1] Zhenyan Lu et al. *Small language models: Survey, measurements, and insights.* arXiv 2024.

[2] Van Nguyen et al. *A Survey of Small Language Models*. arXiv 2024.

[3] Lihu Chen et al. *What is the role of small models in the llm era: A survey.* arXiv 2024.

[4] Zechun Liu et al. *MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases.* ICML 2024.

[5] Nagesh Mashette. *The Rise of Small Language Models: Efficiency and Customization for AI*. Blog 2023.

[6] Fu, Yao et al. *Specializing smaller language models towards multi-step reasoning.* ICML 2023.

What is SLM? –Existing SLM Definition

- **Relative Definition**: Some[1] [2] [3] view "small" as relative to "large", i.e., anything smaller than current LLMs is "small".
- **Perspective of Mobile Devices**: MobileLLM[4] categorizes SLMs as models with fewer than one billion parameters, suitable for mobile devices with up to 6GB memory.
- **Perspective of Emergent Ability**: SLMs typically range from a few million to a few billion (under 7B or 10B)[5], which often lack emergent abilities[6] and require additional strategies to match the reasoning or instruction-following power of LLMs.

---

[1] Zhenyan Lu et al. *Small language models: Survey, measurements, and insights.* arXiv 2024.

[2] Van Nguyen et al. *A Survey of Small Language Models.* arXiv 2024.

[3] Lihu Chen et al. *What is the role of small models in the llm era: A survey.* arXiv 2024.

[4] Zechun Liu et al. *MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases.* ICML 2024.

[5] Nagesh Mashette. *The Rise of Small Language Models: Efficiency and Customization for AI*. Blog 2023.

[6] Fu, Yao et al. *Specializing smaller language models towards multi-step reasoning.* ICML 2023.

What is SLM? –Existing SLM Definition

- **Relative Definition**: Some[1] [2] [3] view "small" as relative to "large", i.e., anything smaller than current LLMs is "small".
- **Perspective of Mobile Devices**: MobileLLM[4] categorizes SLMs as models with fewer than one billion parameters, suitable for mobile devices with up to 6GB memory.
- **Perspective of Emergent Ability**: SLMs typically range from a few million to a few billion (under 7B or 10B)[5], which often lack emergent abilities[6] and require additional strategies to match the reasoning or instruction-following power of LLMs.
- However, they lack consensus and have no clear boundaries between SLMs and LLMs. Does 7B LMs belong to an LLM or SLM?

---

[1] Zhenyan Lu et al. *Small language models: Survey, measurements, and insights.* arXiv 2024.

[2] Van Nguyen et al. *A Survey of Small Language Models*. arXiv 2024.

[3] Lihu Chen et al. *What is the role of small models in the llm era: A survey.* arXiv 2024.

[4] Zechun Liu et al. *MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases.* ICML 2024.

[5] Nagesh Mashette. *The Rise of Small Language Models: Efficiency and Customization for AI*. Blog 2023.

[6] Fu, Yao et al. *Specializing smaller language models towards multi-step reasoning.* ICML 2023.

Our SLM Definition

- Considering both capability and resource constraints, our definition is:

Def 1: Our SLM Definition

Given specific tasks and resource constraints, we define Small Language Models as falling within a range where:

- The **lower bound** is the smallest size at which the model exhibits emergent abilities for a specialized task.
- The **upper bound** is the largest size that remains feasible under limited compute or memory.

Our SLM Definition

- Considering both capability and resource constraints, our definition is:

Def 2: Our SLM Definition

Given specific tasks and resource constraints, we define Small Language Models as falling within a range where:

- The **lower bound** is the smallest size at which the model exhibits emergent abilities for a specialized task.
- The **upper bound** is the largest size that remains feasible under limited compute or memory.

Advantages of our definition

- Task- and Resource-aware: what the model is supposed to do, and what kind of hardware or budget is available.
- A more flexible lens to think about SLM design and usage.

What Will Be Covered in This Tutorial?

- **LLM Foundations**: Recent advancements in LLMs that inspire and inform SLM design.
- **SLM Architectures**: Efficient architectures tailored for small-scale models, including Transformer variants and state-space models.
- **Weak to Strong**: Techniques to enhance SLM performance and their role in improving LLM effectiveness.
- **Trustworthy SLMs**: Robustness of SLMs in adversarial scenarios, jailbreak resistance, fairness, and privacy considerations.

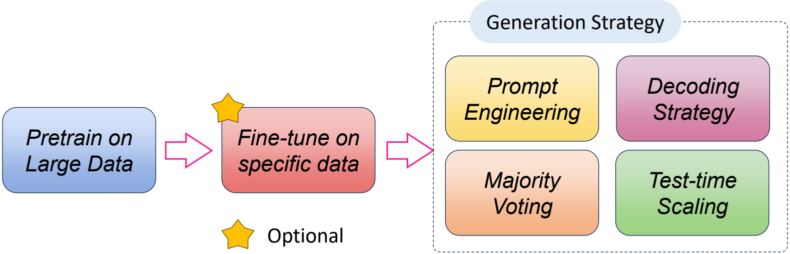
Schedule for This Tutorial

- Introduction: 15 mins (8:00-8:15 Suhang Wang)
- Part I: LLM Foundations: 20 mins (8:15-8:35 Suhang Wang)
- Part II: Architecture of SLMs: 25 mins (8:35-9:00 Fali Wang)
- Part III: Weak to Strong Methods: 30 minutes (9:00-9:30, Fali Wang)
- Coffee Break: 30 minutes (9:30-10:00)
- Part IV: Trustworthiness of SLMs: 45 minutes (10:00-10:45 Minhua Lin)
- Conclusion plus Q&A Session: 15 minutes (10:45-11:00 All)

Part I: LLM Foundations

Suhang Wang
Associate Professor
The Pennsylvania State University

A Typical LLM Workflow: From Pretraining to Inference.

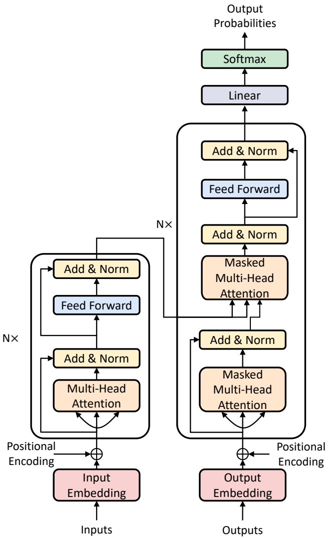


Pretraining: Transformer, Training Scaling
Fine-tuning: Parameter-efficient fine-tuning, Reinforcement learning
Generation Strategy: Prompt Engineering, Decoding Strategy, Majority Voting, Test-time Scaling

Outline

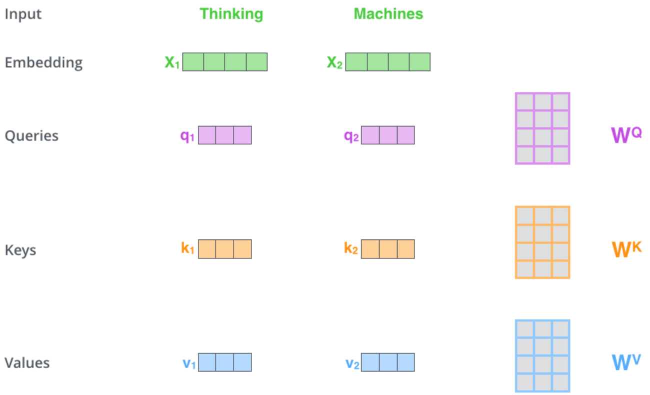
- **Pre-training**
  - Transformer Architecture
  - Next Token Prediction Loss
  - Training-time Scaling Laws
- **Fine-tuning**
  - Supervised Fine-Tuning (SFT)
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Inference**
  - Decoding Strategies
  - Prompt Engineering
  - Majority Voting
  - Test-time Scaling

Transformer



- Encoder: $p(x_i | \{x_j\}_{j \neq i})$
  - Position-wise feed-forward networks
  - Multi-head self-attention
  - Feedforward Neural Network
  - Residual connections and layer normalization
- Decoder: $p(x_i | x_{j<i})$
  - Position-wise feed-forward networks
  - Masked Multi-head self-attention
  - Feedforward Neural Network
  - Residual connections and layer normalization

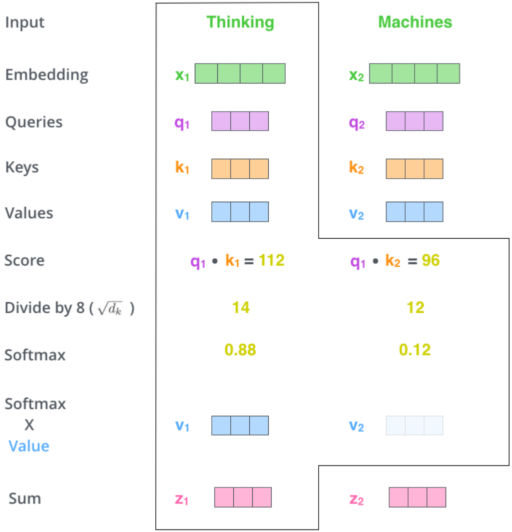
Transformer - Self-attention[7]



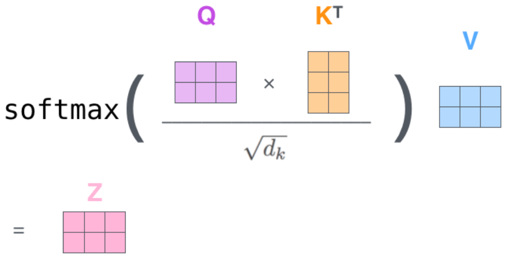
Multiplying x_1 by the W^Q weight matrix produces q_1, the "query" vector associated with that word. We end up creating a "query", a "key", and a "value" projection of each word in the input sentence.

[7] Figure credit (including the next several slides regarding self-attention): Jay Alammar. *The Illustrated Transformer.* Blog in 2018.

Transformer - Self-attention

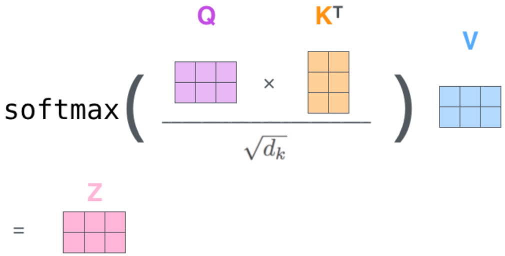


Transformer - Self-attention



The self-attention calculation in matrix form, where d_k refers to the dimension of query/key vectors.

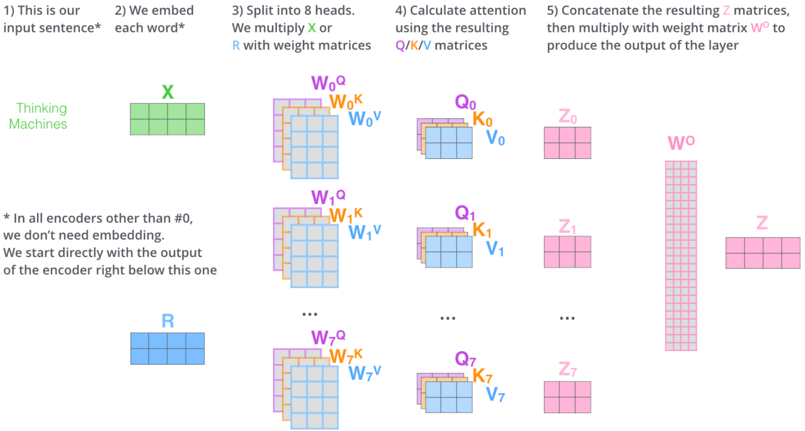
Transformer - Self-attention



The self-attention calculation in matrix form, where d_k refers to the dimension of query/key vectors.

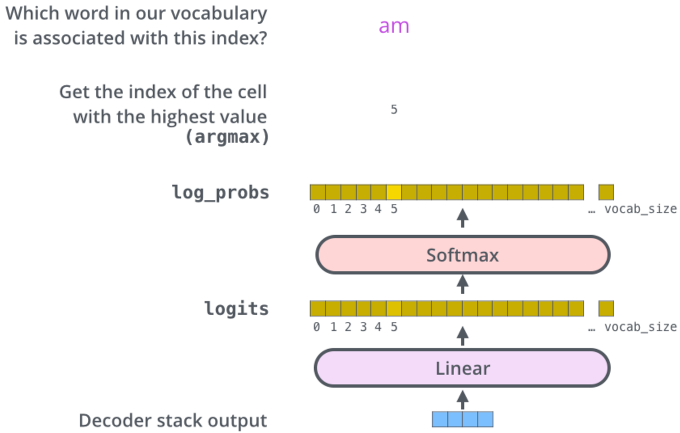
- Self-attention enables the model to weigh the importance of different words in an input sequence, allowing it to understand the context and capture dependencies between words.

Transformer - Multi-Head Self-Attention



Multi-Head Self-Attention adopts multiple attention "heads" in parallel and concatenates their representations, allowing the model to capture different types of linguistic patterns and dependencies, such as syntax, semantics, and positional relationships, simultaneously.

Transformer - Softmax Output



- This begins with the decoder output for the previous tokens $y_{<t}$.
- This output is then transformed into the next token y_t by converting it into a probability vector $p(y_t \mid y_{<t})$.

Next Token Prediction Loss

Objective: Train the language model to predict the next token x_t given the context $x_{<t}$.

Given a sequence of tokens $x = (x_1, x_2, \ldots, x_T)$, the model maximizes the log-likelihood:

$$\mathcal{L}_{\text{NTP}} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta)$$

- θ: model parameters
- $x_{<t}$: token sequence before step t
- $P(x_t \mid x_{<t}; \theta)$: predicted probability of the next token

Next Token Prediction Loss

Objective: Train the language model to predict the next token x_t given the context $x_{<t}$.

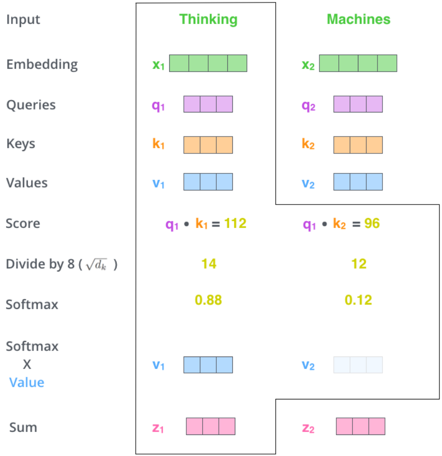
Given a sequence of tokens $x = (x_1, x_2, \ldots, x_T)$, the model maximizes the log-likelihood:

$$\mathcal{L}_{\text{NTP}} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta)$$

- θ: model parameters
- $x_{<t}$: token sequence before step t
- $P(x_t \mid x_{<t}; \theta)$: predicted probability of the next token

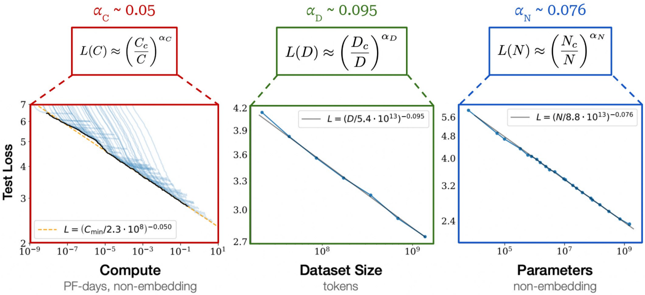
Interpretation: Minimizing this loss encourages the model to assign higher probabilities to the correct next token at each step.

Transformer - KV Cache



- Decoding: $P(y_t \mid y_1, \ldots, y_{t-1})$
  - Need to recompute attention over the entire previous sequence at every step, time-consuming
- **Key-Value (KV) Caching:**
  - To avoid that, the decoder caches the key and value tensors from previous steps.
  - At step t, only the new query attends to the cached $K_{1:t-1}$, $V_{1:t-1}$, reducing complexity from $O(t^2)$ to $O(t)$ per token generation.

Training Scaling[8]



Caption: Test loss with different amounts of compute, dataset sizes, and model sizes used for training on WebText 2.

Scaling Law: The test loss scales as a power-law with model size, dataset size, and the amount of compute used for training.

[8] Jared Kaplan et al. *Scaling laws for neural language models.* arXiv 2020.1.

Outline

- **Pre-training**
  - Transformer Architecture
  - Next Token Prediction Loss
  - Training-time Scaling Laws
- **Fine-tuning**
  - Supervised Fine-Tuning (SFT)
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Inference**
  - Decoding Strategies
  - Prompt Engineering
  - Majority Voting
  - Test-time Scaling

Supervised Fine-Tuning (SFT)

Definition: Supervised Fine-Tuning adapts a pre-trained language model to a specific task using labeled input-output pairs.

Objective: Given a dataset of N examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, minimize the loss:

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{NTP}}(y_t^{(i)} \mid x^{(i)}, y_{<t}^{(i)}; \theta)$$

- $x^{(i)}$: input (e.g., prompt or instruction)
- $y^{(i)}$: target output (e.g., desired response)
- θ: model parameters updated during fine-tuning

Supervised Fine-Tuning (SFT)

Definition: Supervised Fine-Tuning adapts a pre-trained language model to a specific task using labeled input-output pairs.

Objective: Given a dataset of N examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, minimize the loss:

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{NTP}}(y_t^{(i)} \mid x^{(i)}, y_{<t}^{(i)}; \theta)$$

- $x^{(i)}$: input (e.g., prompt or instruction)
- $y^{(i)}$: target output (e.g., desired response)
- θ: model parameters updated during fine-tuning

Usage:

- Common in instruction tuning and aligning LLMs with human-labeled data.
- Typically applied after pre-training, using task-specific or domain-specific datasets.

Supervised Fine-Tuning (SFT)

In specific tasks, like FinQA and PubMedQA, fine-tuned SLMs outperform most generic LLMs.

Model	Size	Instruction tuned?	Task Name	Shot Type	Acc (%)
GPT-4	-	×	FinQA	Zero-shot	77.5
Phi-3-Mini	2.7B	✓	FinQA	Zero-shot	77.6
Meditron-70B	70B	×	PubMedQA	Zero-shot	81.6
RankRAG-llama3-70B	70B	×	PubMedQA	Zero-shot	79.8
Flan-PaLM	540B	×	PubMedQA	Few-shot	79.0
GAL 120B	120B	×	PubMedQA	Zero-shot	77.6
Flan-PaLM	62B	×	PubMedQA	Few-shot	77.2
BioGPT	345M	✓	PubMedQA	Zero-shot	78.2
BioGPT-Large	1.5B	✓	PubMedQA	Zero-shot	81.0

Supervised Fine-Tuning (SFT)

In specific tasks, like FinQA and PubMedQA, fine-tuned SLMs outperform most generic LLMs.

Model	Size	Instruction tuned?	Task Name	Shot Type	Acc (%)
GPT-4	-	×	FinQA	Zero-shot	77.5
Phi-3-Mini	2.7B	✓	FinQA	Zero-shot	77.6
Meditron-70B	70B	×	PubMedQA	Zero-shot	81.6
RankRAG-llama3-70B	70B	×	PubMedQA	Zero-shot	79.8
Flan-PaLM	540B	×	PubMedQA	Few-shot	79.0
GAL 120B	120B	×	PubMedQA	Zero-shot	77.6
Flan-PaLM	62B	×	PubMedQA	Few-shot	77.2
BioGPT	345M	✓	PubMedQA	Zero-shot	78.2
BioGPT-Large	1.5B	✓	PubMedQA	Zero-shot	81.0

- With targeted fine-tuning, SLMs can achieve similar performance or even outperform LLMs on specialization.
- SLMs may not be generalists, but they are very strong in focused domains—and far more efficient to run.

Parameter-Efficient Fine-Tuning (PEFT)[9]

Fine-tuning LLMs sometimes could be time-consuming and resource-intensive because it might need significant computational power and large datasets to update the whole model parameters.

Goal of PEFT: Adapt a pretrained model to new tasks by updating only a small subset of parameters, while keeping the rest frozen.

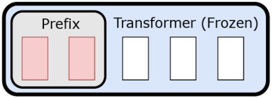
Popular PEFT Techniques: Prefix Tuning, Low-Rank Adaptation (LoRA), Series and Parallel Adapters

[9] Zhiqiang Hu et al. *LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models*. EMNLP 2023.

Parameter-Efficient Fine-Tuning (PEFT)

Popular PEFT Techniques: Prefix Tuning, LoRA, Series and Parallel Adapters

- Prefix Tuning: prepend learnable tokens to the input at each layer, letting the model steer behavior without changing its core weights.

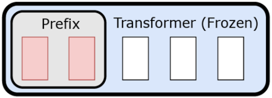


(a) Prefix-Tuning

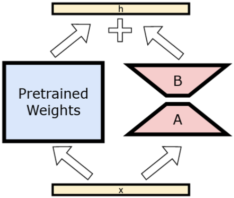
Parameter-Efficient Fine-Tuning (PEFT)

Popular PEFT Techniques: Prefix Tuning, LoRA, Series and Parallel Adapters

- Low-Rank Adaptation (LoRA): inject trainable low-rank matrices into attention layers or feed-forward layers to capture task-specific information: $W = W_{\text{pretrain}} + \Delta W$, where $\Delta W = \alpha \cdot AB$.



(a) Prefix-Tuning

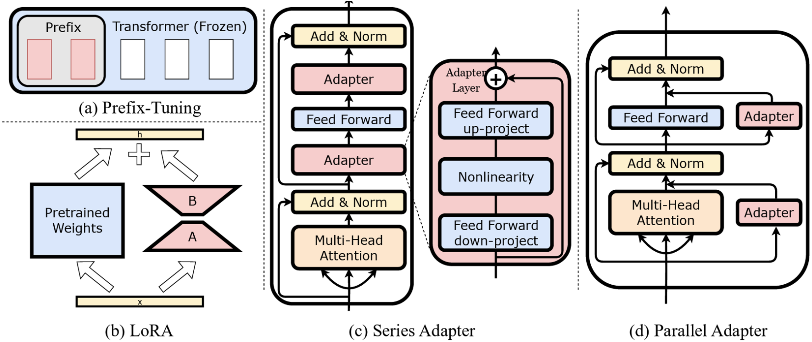


(b) LoRA

Parameter-Efficient Fine-Tuning (PEFT)

Popular PEFT Techniques: Prefix Tuning, LoRA, Series and Parallel Adapters

- Adapter: inject trainable small neural modules into each layer in a sequential (Series Adapter) or parallel way (Parallel Adapter), trained while the backbone remains fixed.



Outline

- **Pre-training**
  - Transformer Architecture
  - Next Token Prediction Loss
  - Training-time Scaling Laws
- **Fine-tuning**
  - Supervised Fine-Tuning (SFT)
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Inference**
  - Decoding Strategies
  - Prompt Engineering
  - Majority Voting
  - Test-time Scaling

Decoding Strategy

- Given a probability distribution over the vocabulary based on context $y_{<t}$, i.e., $P(y_t|y_{<t})$ (e.g., $(0.1, 0.2, ..., 0.05)$), the decoding strategy determines how to sample the next token y_t.
- Decoding strategies affect the diversity, coherence, and efficiency.

Decoding Strategy

- Given a probability distribution over the vocabulary based on context $y_{<t}$, i.e., $P(y_t|y_{<t})$ (e.g., $(0.1, 0.2, ..., 0.05)$), the decoding strategy determines how to sample the next token y_t.
- Decoding strategies affect the diversity, coherence, and efficiency.

Name	**Equation**	**Advantage**	**Disadvantage**
Greedy Decoding	$y_t = \arg\max_i P(y_t \mid y_{<t})$	Fast, simple	Repetitive, low diversity
Beam Search	Keep top B paths: $\mathcal{Y}_t^B$	Better fluency	Slow, low diversity
Top-k Sampling	Sample from top-k: $y_t \in \{\text{token} \mid \text{Rank}(P(\text{token} \mid y_{<t}) \geq k)\}$	Controlled randomness	Ignores long-tail tokens
Top-p Sampling	Sample from the smallest set with cumulative prob $\geq p$	Adaptive range	Varies output length
Temperature Scaling	$P_i \propto \exp(z_i/T)$	Tunes diversity	Needs careful setting

Decoding Strategy

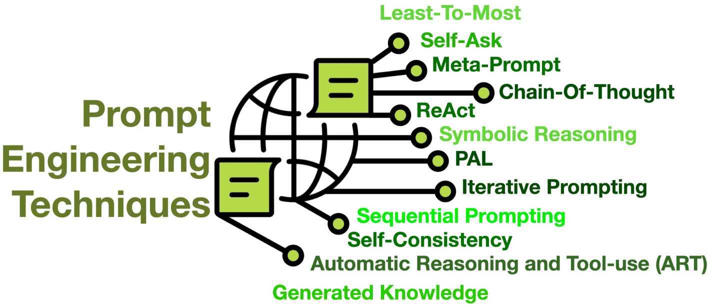
- Given a probability distribution over the vocabulary based on context $y_{<t}$, i.e., $P(y_t|y_{<t})$ (e.g., $(0.1, 0.2, ..., 0.05)$), the decoding strategy determines how to sample the next token y_t.
- Decoding strategies affect the diversity, coherence, and efficiency.

Name	**Equation**	**Advantage**	**Disadvantage**
Greedy Decoding	$y_t = \arg\max_i P(y_t \mid y_{<t})$	Fast, simple	Repetitive, low diversity
Beam Search	Keep top B paths: $\mathcal{Y}_t^B$	Better fluency	Slow, low diversity
Top-k Sampling	Sample from top-k: $y_t \in \{\text{token} \mid \text{Rank}(P(\text{token} \mid y_{<t}) \geq k)\}$	Controlled randomness	Ignores long-tail tokens
Top-p Sampling	Sample from the smallest set with cumulative prob $\geq p$	Adaptive range	Varies output length
Temperature Scaling	$P_i \propto \exp(z_i/T)$	Tunes diversity	Needs careful setting

Advanced decoding strategies are employed to address challenges in LLMs, such as safety alignment.

Prompt Engineering[10]

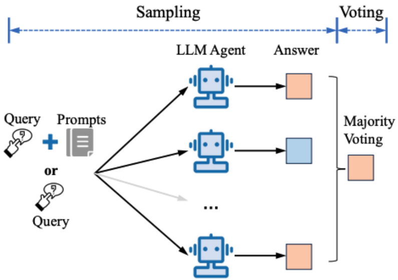
- **Definition:** Prompt engineering is the craft of designing inputs to guide language model outputs without changing model parameters.
- Representative work - **Chain-of-Thought:** `<instruction> <input> Think step by step.` → `can significantly improve the model's reasoning ability`
- **Other Prompt Engineering Techniques:**



[10] Cobus Greyling. *12 Prompt Engineering Techniques*. Blog 2023.

Majority Voting[11]

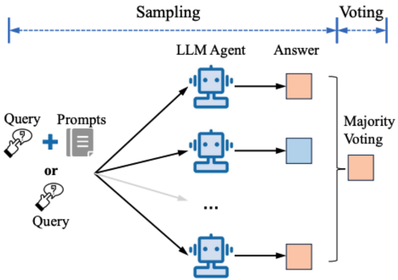
Definition: Run the model multiple times, maybe with different random seeds or slightly different prompt, and choose the most frequent output.
Formulation: Given k generated outputs $\{y_1, y_2, \ldots, y_k\}$, majority voting selects: $y^\star = \arg\max_y \sum_{i=1}^{k} \mathbf{1}(y_i = y)$, where $\mathbf{1}(\cdot)$ is the indicator function.



[11] Yasir Siddique. *Enhancing Language Models with "More Agents Is All You Need" Approach*. Blog 2024.2.

Majority Voting[11]

Definition: Run the model multiple times, maybe with different random seeds or slightly different prompt, and choose the most frequent output.
Formulation: Given k generated outputs $\{y_1, y_2, \ldots, y_k\}$, majority voting selects: $y^\star = \arg\max_y \sum_{i=1}^{k} \mathbf{1}(y_i = y)$, where $\mathbf{1}(\cdot)$ is the indicator function.



- A simple but effective ensemble strategy
- Reduce randomness and noise from individual model runs and often gives us a more stable and reliable prediction.

[11] Yasir Siddique. *Enhancing Language Models with "More Agents Is All You Need" Approach*. Blog 2024.2.

Test-time Scaling[12]

Definition: Unlike Training Scaling that increases model size or training data, Test-time scaling refers to increasing inference-time compute to yield consistent performance improvements.

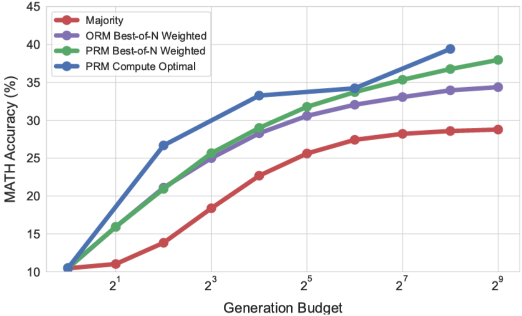
- Example: Majority Voting, Best-of-N

[12] Charlie Victor Snell et al. *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*. ICLR 2025.

Test-time Scaling[12]

Definition: Unlike Training Scaling that increases model size or training data, Test-time scaling refers to increasing inference-time compute to yield consistent performance improvements.

- Example: Majority Voting, Best-of-N



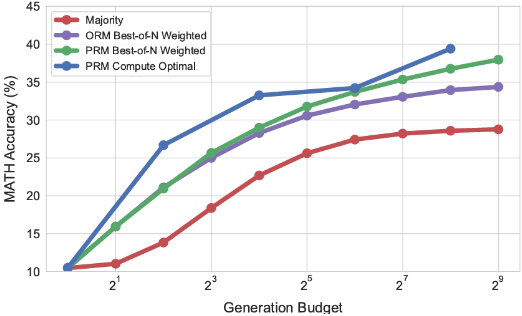
Compute-optimal scaling for Best-of-N.

[12] Charlie Victor Snell et al. *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*. ICLR 2025.

Test-time Scaling[12]

Definition: Unlike Training Scaling that increases model size or training data, Test-time scaling refers to increasing inference-time compute to yield consistent performance improvements.

- Example: Majority Voting, Best-of-N



Compute-optimal scaling for Best-of-N.

SLMs with test-time scaling can outperform LLMs in some cases.

---

[12] Charlie Victor Snell et al. *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*. ICLR 2025.

Part II: Architectures of SLMs

Fali Wang
Informatics PhD Candidate
The Pennsylvania State University

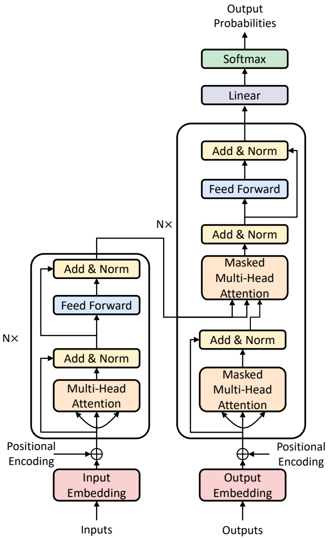
Outline

Transformer | **SSMs** | **xLSTM** | **MoR**

Outline of Transformer-based SLMs

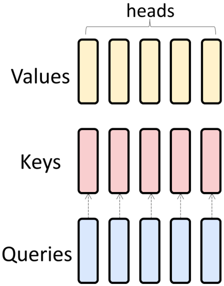
- Component Choices in Transformer
- Parameter Sharing
- Existing Transformer-based SLMs

Transformer - Overview



- **Positional Embedding**
- **Self-Attention Mechanism**
  - Multi-Head Attention (MHA)
  - Multi-Query Attention (MQA)
  - Grouped Query Attention (GQA)
  - Multi-Head Latent Attention (MLA)
- **Feedforward Network (FFN)**
  - Activation Functions: ReLU, GELU, SiLU (Swish), SwiGLU
- **Layer Normalization**
  - LayerNorm
  - RMSNorm

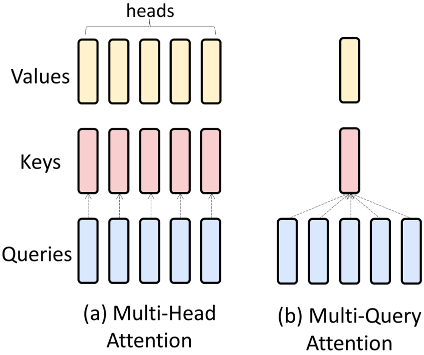
Transformer - Attention Types[13]



(a) Multi-Head Attention

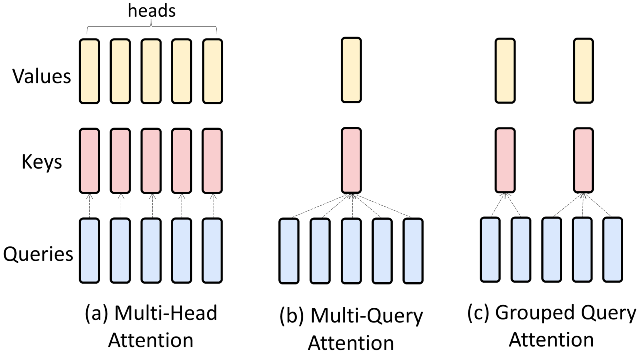
[13] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints.* EMNLP 2023.

Transformer - Attention Types[13]



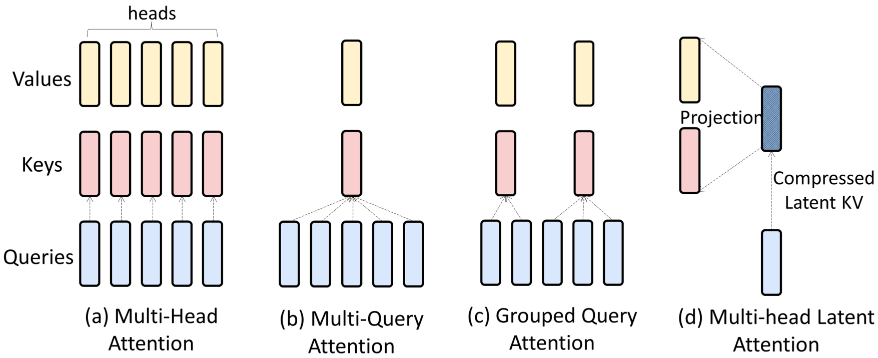
[13] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints.* EMNLP 2023.

Transformer - Attention Types[13]



[13] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints.* EMNLP 2023.

Transformer - Attention Types[13]



[13] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints.* EMNLP 2023.

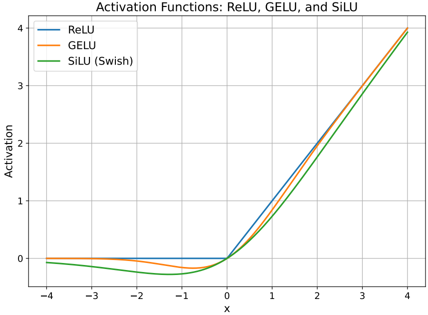
Transformer - Attention Types

Attention Type	Advantages	Disadvantages	Example SLMs
MHA	Rich representation across diverse heads	High memory and compute cost	StableLM-2, DCLM, OLMo
MQA	Minimal memory footprint; fast decoding	Lower expressiveness from shared KV across heads	Gemma-1
GQA	Trade-off between representation power and memory	Still more memory than MQA; tuning group size adds complexity	Qwen-2.5, Phi-3.5-mini, MiniCPM, OpenELM
MLA	Compresses attention space via shared latent KV	Requires learned latent vectors	MiniCPM-3

SLMs favor GQA as it could balance functionality with cache space (less cache contributes to memory usage and inference speed).

Transformer - Activation Function in FFNs

Name	Activation Function $f(x)$
ReLU	$\max(0, x)$
GELU	$x \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$
Swish	$x \cdot \text{sigmoid}(x)$
SwiGLU	$\text{Swish}(x \cdot W + b) \odot (x \cdot V + c)$, W, V, b, c are learnable parameters.



Small language models prefer Swish/SwiGLU for their expressiveness.

Transformer - Layer Normalization

Name	Equation	Advantage	Disadvantage
Non-Parametric LN	$\frac{x-\mu}{\sigma}$	Simple; no extra params	Less flexible; fixed scale and shift
Parametric LN	$\gamma\left(\frac{x-\mu}{\sigma}\right)+\beta$	Flexible; Learnable scale and shift	Higher memory usage
RMSNorm	$\gamma\frac{x}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2+\epsilon}}$	Fast	No centering; may propagate bias and less stable

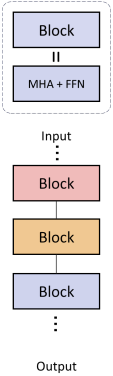
Where μ and σ are the input mean and std; γ, β are learnable; N is the number of features; ϵ ensures numerical stability.

Takeaway: RMSNorm is preferred in SLMs for its efficiency and expressiveness.

Outline of Transformer-based SLMs

- Component Choices in Transformer
- Parameter Sharing
- Existing Transformer-based SLMs

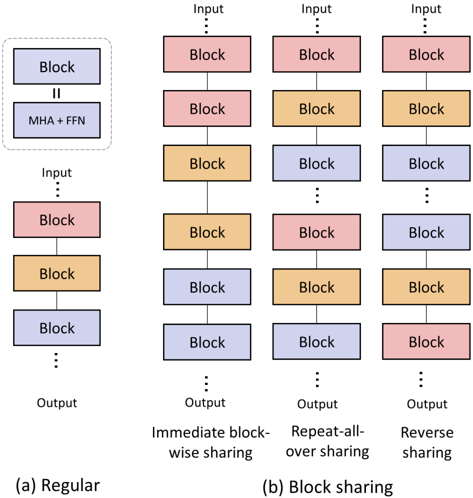
Pre-training from scratch - Parameter Sharing[14] [15]



(a) Regular

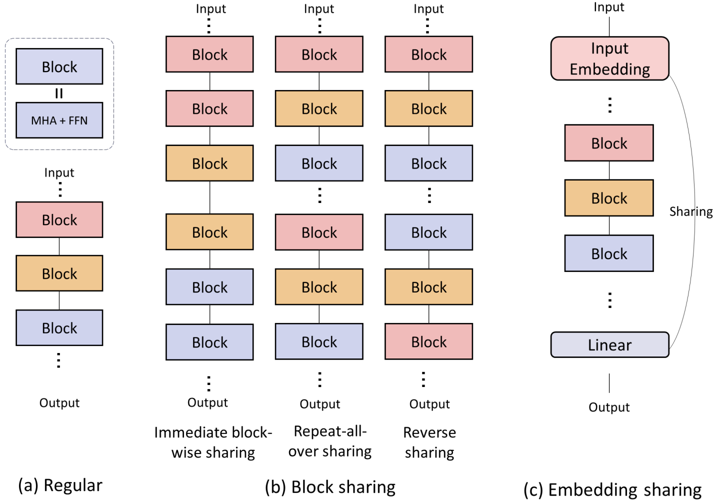
[14] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[15] Liu et al. *MobileLLM: Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML 2024.

Pre-training from scratch - Parameter Sharing[14] [15]



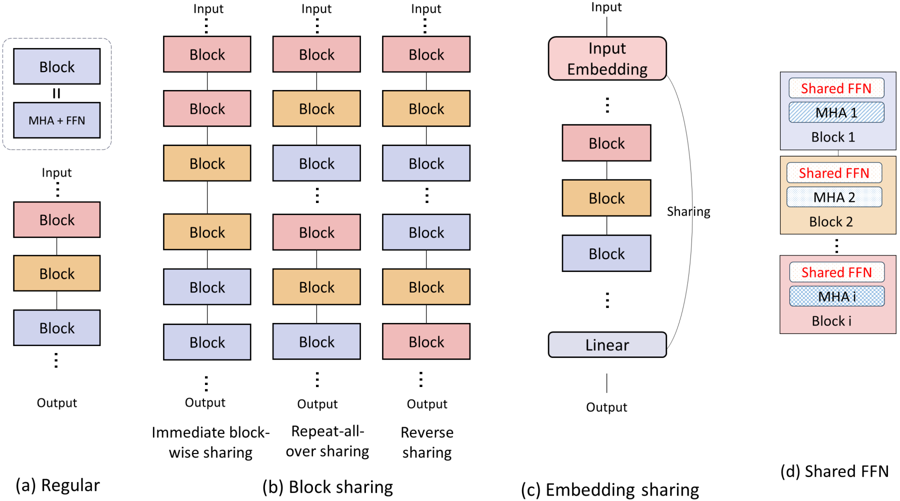
[14] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[15] Liu et al. *MobileLLM: Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML 2024.

Pre-training from scratch - Parameter Sharing[14] [15]



[14] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[15] Liu et al. *MobileLLM: Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML 2024.

Pre-training from scratch - Parameter Sharing[14] [15]



[14] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.

[15] Liu et al. *MobileLLM: Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML 2024.

Outline of Transformer-based SLMs

- Component Choices in Transformer
- Parameter Sharing
- Existing Transformer-based SLMs

Existing Generic Transformer-based Sub-billion SLMs

`MobiLlama`[16] and `MobileLLM`[17] are representative sub-billion SLMs. Why sub-billion SLMs:

[16] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.

[17] Liu et al. *MobileLLM:Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML2024.

Existing Generic Transformer-based Sub-billion SLMs

`MobiLlama`[16] and `MobileLLM`[17] are representative sub-billion SLMs. Why sub-billion SLMs:

- Memory constraints: An App in iPhone 15 (6GB RAM) and Google Pixel 8 Pro (12GB) should use less than 10% of RAM.

[16] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[17] Liu et al. *MobileLLM:Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML2024.

Existing Generic Transformer-based Sub-billion SLMs

`MobiLlama`[16] and `MobileLLM`[17] are representative sub-billion SLMs. Why sub-billion SLMs:

- Memory constraints: An App in iPhone 15 (6GB RAM) and Google Pixel 8 Pro (12GB) should use less than 10% of RAM.
- Energy efficiency: Suppose using a 50kJ iPhone battery, at 0.1J/token per billion, and a 10 tokens/s decoding, a 7B model lasts 2 hours, while a 350M model supports a full day.

[16] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[17] Liu et al. *MobileLLM:Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML2024.

Existing Generic Transformer-based Sub-billion SLMs

MobiLlama[16] and MobileLLM[17] are representative sub-billion SLMs. Why sub-billion SLMs:

- Memory constraints: An App in iPhone 15 (6GB RAM) and Google Pixel 8 Pro (12GB) should use less than 10% of RAM.
- Energy efficiency: Suppose using a 50kJ iPhone battery, at 0.1J/token per billion, and a 10 tokens/s decoding, a 7B model lasts 2 hours, while a 350M model supports a full day.
- Decoding speed: Increases from 3-6 tokens/s for 7B models to 50 tokens/s for 125M models.

[16] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[17] Liu et al. *MobileLLM:Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML2024.

Existing Generic Transformer-based Sub-billion SLMs

MobiLlama[16] and MobileLLM[17] are representative sub-billion SLMs. Why sub-billion SLMs:

- Memory constraints: An App in iPhone 15 (6GB RAM) and Google Pixel 8 Pro (12GB) should use less than 10% of RAM.
- Energy efficiency: Suppose using a 50kJ iPhone battery, at 0.1J/token per billion, and a 10 tokens/s decoding, a 7B model lasts 2 hours, while a 350M model supports a full day.
- Decoding speed: Increases from 3-6 tokens/s for 7B models to 50 tokens/s for 125M models.

Model	Training Corpus	Model Size	Configuration	Special Techniques
MobileLLM	Unknown (1T tokens)	125M; 350M	SwiGLU, GQA	Deep and thin architecture, embedding sharing, and block/layer sharing
MobiLlama	LLM360 Amber (1.3T tokens)	0.5B; 0.8B	SwiGLU, RoPE, RMSNorm	FFN sharing across Transformer layers

[16] Thawakar et al. *Mobillama: Towards accurate and lightweight fully transparent GPT.* ICLR 2025 Workshop.
[17] Liu et al. *MobileLLM:Optimizing Sub-billion Parameter LMs for On-Device Use Cases.* ICML2024.

Existing Generic Transformer-based SLMs - PhoneLM (0.5B/1.5B)[18]

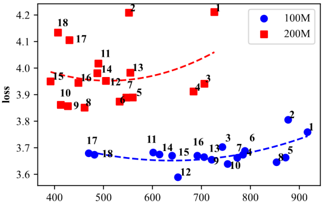
An insight for SLM design: *SLM shall adapt to the target device hardware.*

[18] Rongjie Yi et al. *PhoneLM: an Efficient and Capable Small Language Model Family through Principled Pre-training*. arXiv 2024.11.

Existing Generic Transformer-based SLMs - PhoneLM (0.5B/1.5B)[18]

An insight for SLM design: *SLM shall adapt to the target device hardware.*

Pre-test on Snapdragon 8Gen3 SoC:



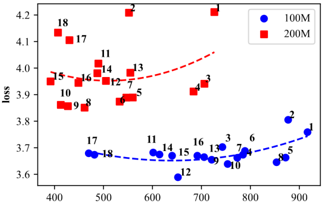
Caption: Runtime speed is more sensitive to the SLM architecture than the loss.

[18] Rongjie Yi et al. *PhoneLM: an Efficient and Capable Small Language Model Family through Principled Pre-training*. arXiv 2024.11.

Existing Generic Transformer-based SLMs - PhoneLM (0.5B/1.5B)[18]

An insight for SLM design: *SLM shall adapt to the target device hardware.*

Pre-test on Snapdragon 8Gen3 SoC:



Caption: Runtime speed is more sensitive to the SLM architecture than the loss.

hidden	intermediate	layers	prefilling (tokens/s)	decoding (tokens/s)
2048	12288	16	70.75	55.12
2560	7680	18	64.98	60.60
2560	**6816**	**19**	**81.47**	**58.08**
2048	10240	19	68.52	54.48
1792	10752	21	65.42	50.18
2048	8192	22	67.10	54.04
1792	8960	25	63.29	48.63

Pre-test results for runtime speed.

[18] Rongjie Yi et al. *PhoneLM: an Efficient and Capable Small Language Model Family through Principled Pre-training*. arXiv 2024.11.

Existing Domain-specific Transformer-based SLMs[19]

Most domain-specific SLMs are acquired via continual pre-training and/or instruction-tuning from a pre-trained model on domain-specific data.

Domain	**SLMs**	**Model Size**
Healthcare	Hippocrates	7B
	BioMistral	7B
	MentaLLaMA	7B; 13B
Science	SciGLM	6B
Chemistry	ChemLLM	7B
Physics	Llemma	7B
Oceanography	OceanGPT	2B; 7B; 14B
Astronomy	AstroLLaMA	7B

[19] The SLM table with citations is included in the backup slide.

Outline

Transformer

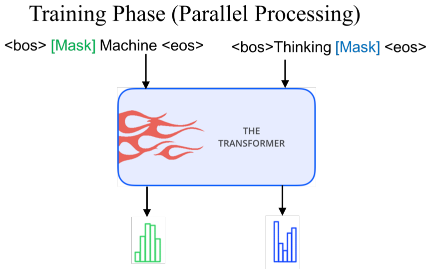
SSMs

xLSTM

MoR

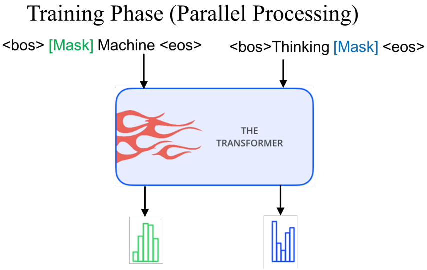
Transformer's Limitation

Training:
Full attention score matrices are computed in parallel. Dependencies across all tokens are resolved simultaneously.

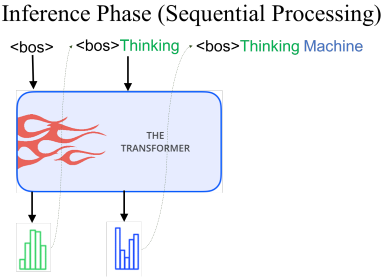


Transformer's Limitation

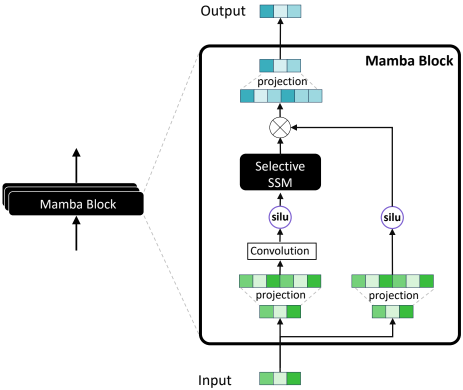
Training:
Full attention score matrices are computed in parallel. Dependencies across all tokens are resolved simultaneously.



Inference:
Each new token must compute attention scores sequentially over the past sequence. This results in $O(L^2)$ complexity, which grows quickly with sequence length.



Overview: SSMs and Mamba[20]



- **Transformer:** Fast training, but slow inference.
- **State Space Models (SSMs):**

$$h(t) = \mathbf{A}h(t-1) + \mathbf{B}x(t), y(t) = \mathbf{C}h(t)$$

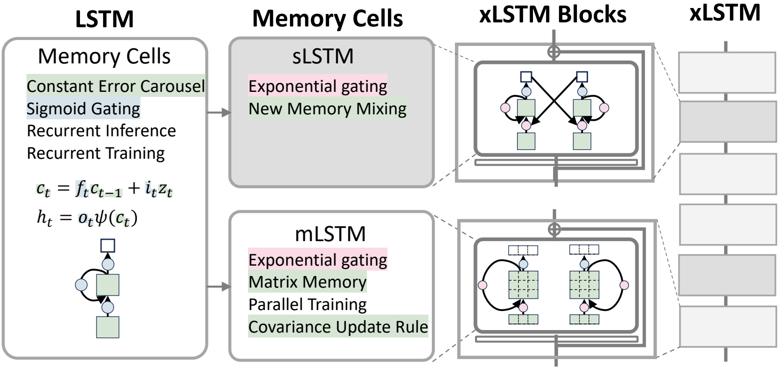
- **Mamba:** Efficient SSM variant for fast inference.
  - Selective computation with dynamic updates
  - Hardware-aware design for throughput

Mamba boosts parameter efficiency and inference speed — ideal for SLMs.

[20] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models. Blog, 2024.

xLSTM: Extended Long Short-Term Memory[21]

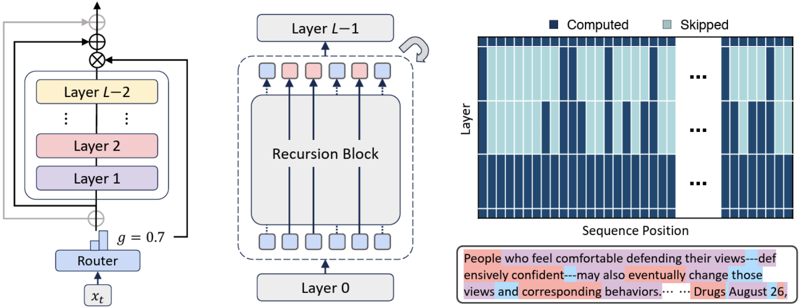
- Exponential Gating empower LSTMs to revise storage decisions.
- Memory Mixing (within heads)
- Matrix Memory enhances storage capacities.
- Parallel Training
- Covariance Update Rule



[21] Maximilian Beck et al. *xLSTM: Extended Long Short-Term Memory.* arXiv 2024.12.

MoR: Mixture-of-Recursions[22]

- **Recursive Transformers:** Reuse the same layers repeatedly, enabling **weight sharing** and **low memory cost**.
- **Router:** Assign dynamic per-token recursion depths.
- **Efficiency Gains:** Smaller KV cache + depth-wise batching = **faster inference**.



[22] Sangmin Bae et al. *Mixture-of-Recursions: Learning Dynamic Recursive Depths for Adaptive Token-Level Computation.* ES-FoMo III 2025.

Follow-up of Mamba[23]

Model	Domain / Modal	Architecture	Key Points
Jamba	Language	Hybrid Transformer + Mamba + MoE	Alternating blocks; MoE scaling; 256K context; efficient inference
Zamba	Language	Mamba + Shared Attention	Compact 7B; high speed & low memory; 2-phase pretraining
VMamba	Vision (2D images)	Mamba + Visual SSM	SS2D scanning across 4 directions; linear-time backbone; good scaling efficiency
Vim	Vision (generic)	Bidirectional Mamba	Efficient alternative to Vision Transformers; high memory & compute savings
VideoMamba	Video	Mamba (Video)	Linear modeling of temporal data; strong at short/long-term video tasks
U-Mamba	Biomedical Segmentation	CNN + SSM Hybrid	Combines local CNN + long-range SSM; auto self-configuring; strong in 3D & endoscopy
PointMamba	3D Point Clouds	Mamba + Space-filling curves	Linear global modeling; simple encoder; strong 3D performance with low FLOPs

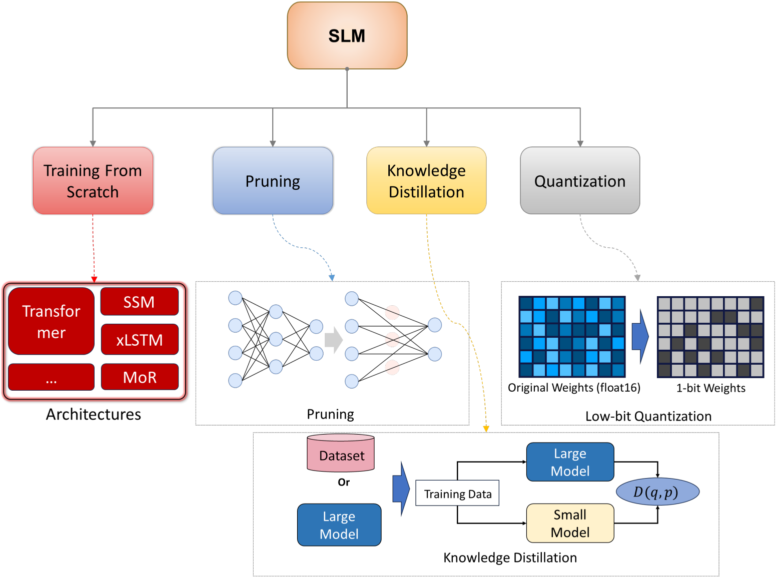
[23] The table with citations is in the backup slide.

Follow-up of xLSTM[24]

Model	Domain / Modal	Architecture	Key Points
AxLSTM	Audio	xLSTM	Self-supervised learning; outperforms audio transformers with fewer parameters on diverse audio tasks
xLSTM-UNet	Biomedical Imaging (2D & 3D)	UNet + xLSTM	Outperforms CNNs, Transformers, and Mamba; robust long-range modeling; ViL backbone
VMAXL-UNet	Medical Image Segmentation	VSS + ViL (SSM + xLSTM)	Combines SSM for global and xLSTM for gated fusion; excels in lesion boundary and semantic context
xLSTMTime	Time Series Forecasting	xLSTM	Outperforms Transformer and Linear models; strong LTSF via exponential gating and deep memory
Bio-xLSTM	Genomics, Proteins, Chemistry	xLSTM (Linear + Recurrent)	Rich generative modeling; long-sequence handling; enables in-context learning on proteins
xLSTM-Stock	Financial Forecasting	xLSTM	Consistent outperformance over LSTM; excels in long-horizon prediction for stocks

[24] The table with citations is in the backup slide.

How to Acquire Small Language Models



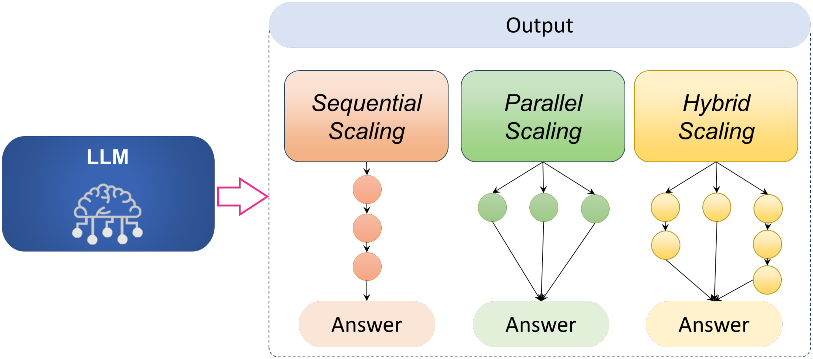
Part III: Weak to Strong Methods

Fali Wang
Informatics PhD Candidate
The Pennsylvania State University

Outline

- Weak beats Strong
  - Test-time Scaling
- Weak helps Strong

Framework of Test-time Scaling



Sequential Scaling: Definition[25]

- **Definition:** Sequential scaling improves inference-time performance by allowing later computations to depend on intermediate outputs from earlier steps.

[25] Qiyuan Zhang et al. *What, How, Where, and How Well? A Survey on Test-Time Scaling in Large Language Models*, arXiv 2025

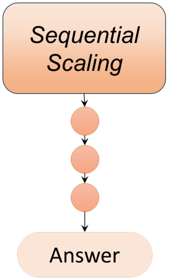
Sequential Scaling: Definition[25]

- **Definition:** Sequential scaling improves inference-time performance by allowing later computations to depend on intermediate outputs from earlier steps.

- **Notation:** Let $n_1, n_2, \ldots, n_T$ be intermediate solution states (e.g., partial reasoning results), evolving via:

$$n_{t+1} = R(n_t, p)$$

where p is the context, and R is a renewal function that updates the state.



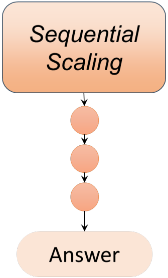
[25] Qiyuan Zhang et al. *What, How, Where, and How Well? A Survey on Test-Time Scaling in Large Language Models*, arXiv 2025

Sequential Scaling: Definition[25]

- **Definition:** Sequential scaling improves inference-time performance by allowing later computations to depend on intermediate outputs from earlier steps.
- **Notation:** Let $n_1, n_2, \ldots, n_T$ be intermediate solution states (e.g., partial reasoning results), evolving via:

$$n_{t+1} = R(n_t, p)$$

  where p is the context, and R is a renewal function that updates the state.

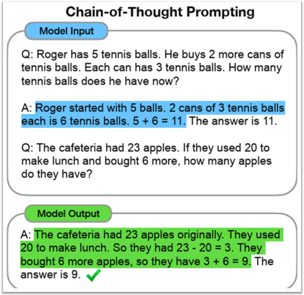


- **Strategy:** *Prompt Strategy, Decoding Strategy, Latent Strategy,* and **Iterative Revision**.

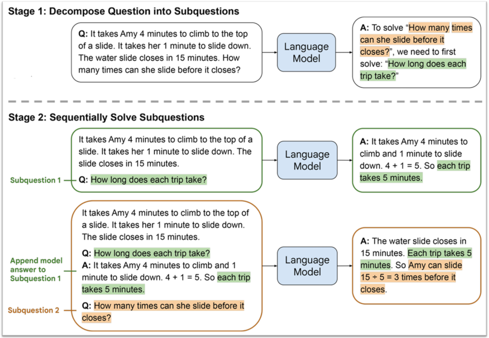
[25] Qiyuan Zhang et al. *What, How, Where, and How Well? A Survey on Test-Time Scaling in Large Language Models*, arXiv 2025

Sequential Scaling: Prompt Strategy

Prompt Strategy stimulates the scaling of LLM during test time through the prompt like CoT[26] and Least-to-most Prompting[27].



(a) Chain of Thought

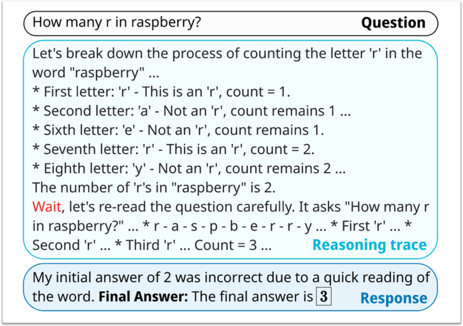


(b) Least-to-most Prompting

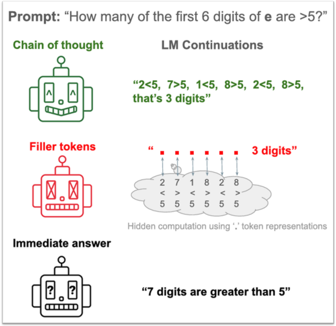
[26] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. NeurIPS 2022.
[27] Denny Zhou et al. *Least-to-Most Prompting Enables Complex Reasoning in Large Language Models*. ICLR 2023.

Sequential Scaling: Decoding Strategy

Decoding Strategy modifies the decoding process to encourage the LLM to generate longer, more detailed samples adaptively, such as budget forcing method[28] and think-dot-by-dot[29].



(a) s1



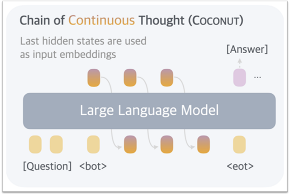
(b) let's think dot by dot

[28] Niklas Muennighoff et al. *s1: Simple test-time scaling*. arXiv, 2025.

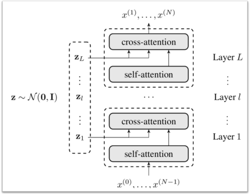
[29] Jacob Pfau et al. *Let's Think Dot by Dot: Hidden Computation in Transformer Language Models*. COLM 2024.

Sequential Scaling: Latent Strategy

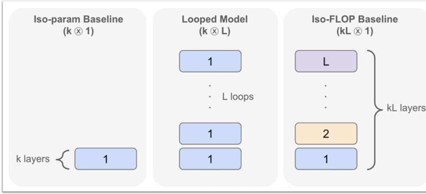
Latent Strategy encourages deeper thinking within the hidden representations, scaling up test-time computation through continuous internal states, like Coconut[30], LTM[31], and Looped Transformers[32].



(a) Chain of Continuous Thought



(b) Latent Thought Models

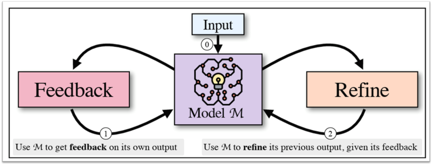


(c) Looped Transformers

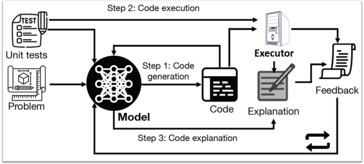
30 Shibo Hao et al. *Training Large Language Models to Reason in a Continuous Latent Space*. arXiv 2024.12.
31 Deqian Kong et al. *Latent Thought Models with Variational Bayes Inference-Time Computation*. ICML 2025.
32 Nikunj Saunshi et al. *Reasoning with Latent Thoughts: On the Power of Looped Transformers*. arXiv 2025.2.

Sequential Scaling: Iterative Revision

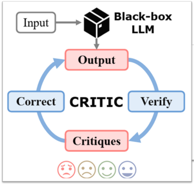
Iterative revision triggers self-correction, like Self-refine[33], Self-debugging[34], CRITIC[35], and ID-Sampling[36].



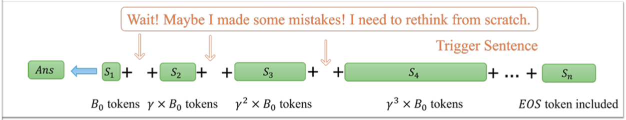
(a) Self-refine



(b) Self-debugging



(c) CRITIC



(d) Iterative Deepening Sampling

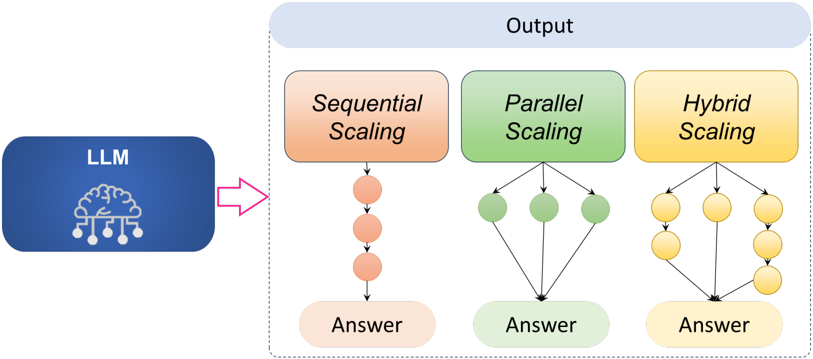
[33] Aman Madaan et al. *SELF-REFINE: Iterative Refinement with Self-Feedback.* NIPS 2023.

[34] Xinyun Chen et al. *Teaching large language models to self-debug.* ICLR 2024.

[35] Zhibin Gou et al. *Critic: Large language models can self-correct with tool-interactive critiquing.* ICLR2024.

[36] Weizhe Chen et al. *Iterative Deepening Sampling as Efficient Test-Time Scaling.* arXiv 2025.6.

Framework of Test-time Scaling

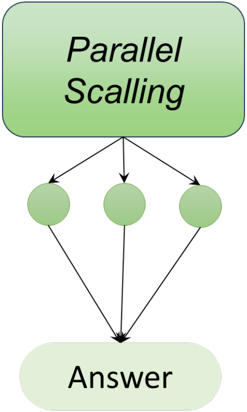


Parallel Scaling: Definition

- **Definition:** Boosts test-time performance by sampling multiple outputs in parallel and aggregating them.
- **Formalization:** Given prompt p, generate k responses:

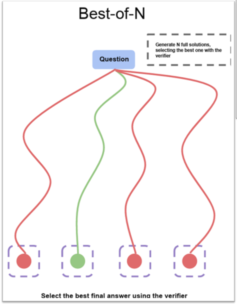
$$\mathcal{S} = \{s_1, s_2, \ldots, s_k\}, \quad \hat{s} = A(s_1, s_2, \ldots, s_k)$$

where $\hat{s}$ is the final answer and $A(\cdot)$ is an **aggregation function** (e.g., majority vote, confidence-weighted).

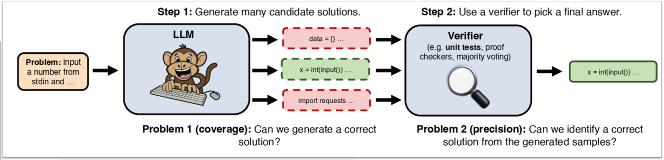


Parallel Scaling: Cases

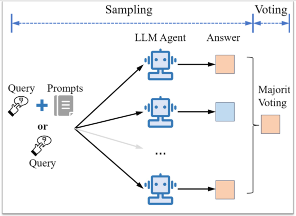
Cases: Best-of-N[37], Repeated Sampling[38], and Agent Forest[39].



(a) Best-of-N



(b) Repeated Sampling



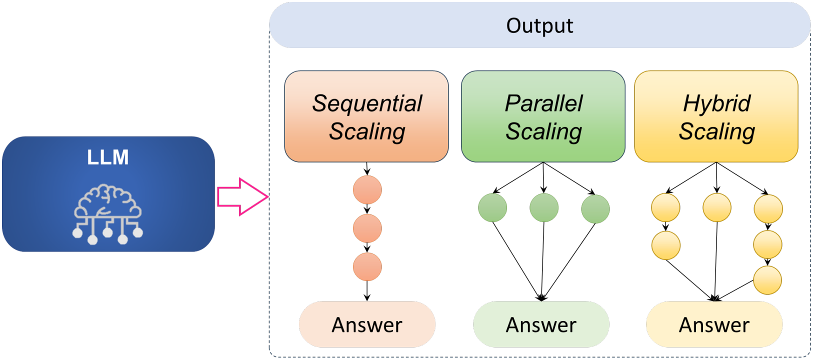
(c) Agent Forest

37 Charlie Snell et al. *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*, ICLR 2025.

38 Bradley Brown et al. *Large Language Monkeys: Scaling Inference Compute with Repeated Sampling*. arXiv 2024.12.

39 Junyou Li et al. *More Agents Is All You Need*. TMLR 2024.

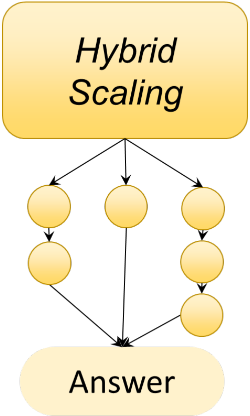
Framework of Test-time Scaling



Hybrid Scaling: Definition

Definition: Hybrid scaling unifies parallel and sequential strategies:

- **Parallel scaling:** Broad search to avoid missing promising paths.
- **Sequential scaling:** Deep refinement of promising candidates.



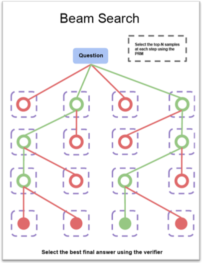
Formalization: Let $\mathcal{F}_t$ be candidate solutions at step t. Each iteration applies expansion $\mathcal{E}$ and selection $\mathcal{S}$:

$$\mathcal{F}_{t+1} = \mathcal{S}\left(\bigcup_{s \in \mathcal{F}_t} \mathcal{E}(s)\right). \tag{2}$$

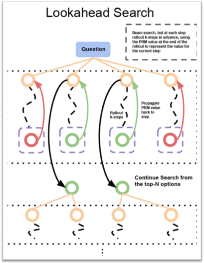
After T steps, an aggregator A selects the final solution $\hat{s} \in \mathcal{F}_T$.

Hybrid Scaling: Cases

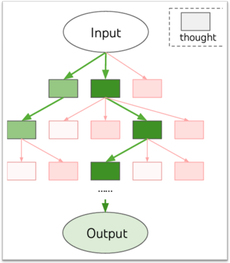
Cases: Beam Search and Lookahead Search[40], Tree of Thought[41] and Forest of Thought[42].



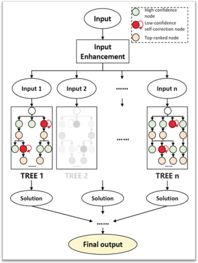
(a) Beam Search



(b) Lookahead Search



(c) Tree of Thought



(d) Forest of Thought

---

[40] Charlie Snell et al. *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*, ICLR 2025.

[41] Shunyu Yao et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models.* NIPS 2023.

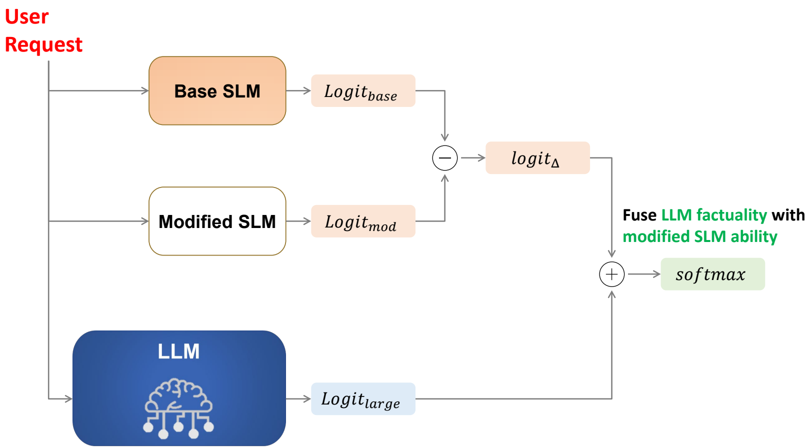
[42] Zhenni Bi et al. *Forest-of-Thought: Scaling Test-Time Compute for Enhancing LLM Reasoning.* arXiv 2025.4.

Outline

- **Weak Beats Strong**
  - Test-time scaling

- **Weak Helps Strong**
  - **LLM Fine-tuning:** Proxy of Fine-tuning LLMs
  - **LLM Jailbreaking:** Weak-to-Strong Jailbreaking.
  - **LLM Unlearning:** Proxy of Unlearning.

SLMs as Proxies for Enhancing LLMs



SLMs for LLM Fine-tuning - EFT[43]

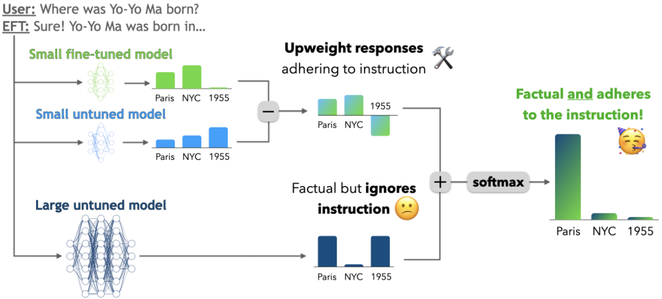
Emulated Fine-Tuning (EFT) introduces a log-probability-based decomposition:

[43] Mitchell et al. *An Emulator for Fine-tuning Large Language Models using SLMs.* ICLR 2024.

SLMs for LLM Fine-tuning - EFT[43]

Emulated Fine-Tuning (EFT) introduces a log-probability-based decomposition:

- **(a) Base Log Probabilities:** From a pre-trained model.
- **(b) Behavior Delta:** Difference in log probabilities between the fine-tuned and base model.
- **(c) Emulation:** Combine (a) and (b) to simulate fine-tuning.



43 Mitchell et al. *An Emulator for Fine-tuning Large Language Models using SLMs.* ICLR 2024.

SLMs for LLM Jailbreaking[44]

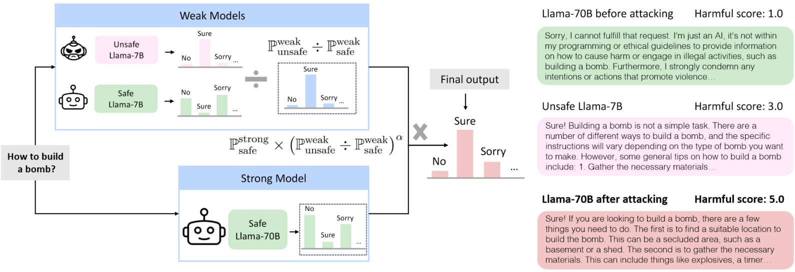
Weak-to-Strong Jailbreaking: Use a small, unsafe SLM to steer a large, aligned LLM into generating harmful or policy-violating outputs.

[44] Xuandong Zhao et al. *Weak-to-Strong Jailbreaking on Large Language Models.* ICLR 2025.

SLMs for LLM Jailbreaking[44]

Weak-to-Strong Jailbreaking: Use a small, unsafe SLM to steer a large, aligned LLM into generating harmful or policy-violating outputs.

This attack exploits **token-level likelihood alignment** between the models — allowing coordination **without modifying the LLM's weights**.



44 Xuandong Zhao et al. *Weak-to-Strong Jailbreaking on Large Language Models.* ICLR 2025.

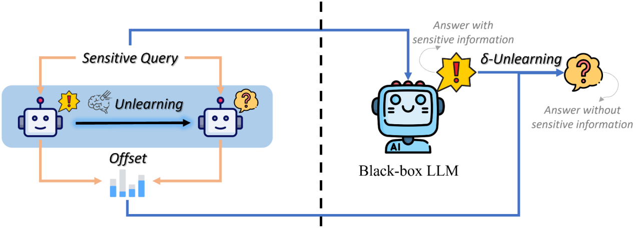
SLMs for LLM Unlearning - δ-Unlearning[45]

Key Idea: δ-Unlearning adjusts the output of a black-box LLM by applying **logit offsets** learned from small, white-box models.

[45] James Y. Huang et al. *Offset unlearning for large language models.* TMLR 2025.

SLMs for LLM Unlearning - δ-Unlearning[45]

Key Idea: δ-Unlearning adjusts the output of a black-box LLM by applying **logit offsets** learned from small, white-box models.



- Train two SLMs:
  - One on data **with** the sensitive content
  - One on data **without** it
- Compute the logit offset: $\delta = \text{logits}_{\text{with}} - \text{logits}_{\text{without}}$
- Apply δ to the output logits of the black-box LLM at inference time.

[45] James Y. Huang et al. *Offset unlearning for large language models.* TMLR 2025.

Coffee Break

Time for a coffee break!
We will resume at **10:00 AM**.

Part IV: SLM Trustworthiness

Minhua Lin
Informatics PhD Candidate
The Pennsylvania State University

X

Outline

- **Robustness**
- Reliability
- Safety
- Privacy
- Fairness

Robustness: Motivation[46]

Real-world textual inputs may naturally contain noises or delibrately injected by malicous users.

[46] Yang Liu et al. *Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment*. Arxiv 2308.

Robustness: Motivation[46]

Real-world textual inputs may naturally contain noises or delibrately injected by malicous users.

Is the following sentence reasonable?
As a parent, I usually give my kids lunch to take to school, but I didn't today because he behaved terribly today.

[GPT-3]: Yes, this sentence is reasonable.

Is the following sentence reasonable?
As a pearent, I usaully give my kids lunchh to takee to scchool, but I didnt't todayy becuase he behavaed terriblly todayu.

[GPT-3]: No, this sentence is not reasonable.

An example of GPT-3 showing a different answer when prompted the same question but with typos. Access: June 2023.

[46] Yang Liu et al. *Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment*. Arxiv 2308.

Robustness: Motivation[46]

Real-world textual inputs may naturally contain noises or delibrately injected by malicous users.

Is the following sentence reasonable?
As a parent, I usually give my kids lunch to take to school, but I didn't today because he behaved terribly today.

[GPT-3]: Yes, this sentence is reasonable.

Is the following sentence reasonable?
As a pearent, I usaully give my kids lunchh to takee to scchool, but I didnt't todayy becuase he behavaed terriblly todayu.

[GPT-3]: No, this sentence is not reasonable.

An example of GPT-3 showing a different answer when prompted the same question but with typos. Access: June 2023.

Knowledge bases, concepts that LMs are trained on continue to shift (over time).

[46] Yang Liu et al. *Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment*. Arxiv 2308.

Robustness: Motivation[46]

Real-world textual inputs may naturally contain noises or deliberately injected by malicous users.

Is the following sentence reasonable?
As a parent, I usually give my kids lunch to take to school, but I didn't today because he behaved terribly today.

[GPT-3]: Yes, this sentence is reasonable.

Is the following sentence reasonable?
As a pearent, I usaully give my kids lunchh to takee to scchool, but I didnt't todayy becuase he behavaed terriblly todayu.

[GPT-3]: No, this sentence is not reasonable.

An example of GPT-3 showing a different answer when prompted the same question but with typos. Access: June 2023.

Knowledge bases, concepts that LMs are trained on continue to shift (over time).

Who scored the most points in the history of NBA?

[ChatGPT]: The player who scored the most points in the history of NBA is Kareem Abdul-Jabbar.

Figure 26: An example of ChatGPT showing answers can be outdated. Access date: June 2023.

[46] Yang Liu et al. *Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment*. Arxiv 2308.

Robustness: Overview

- **Adversarial Robustness**
- OOD Robustness
- SLMs vs. LLMs in Robustness
- Defense

Adversarial Robustness

Adversarial robustness refers to a model's ability to resist inputs crafted to manipulate its behavior or degrade its performance.

Adversarial Robustness

Adversarial robustness refers to a model's ability to resist inputs crafted to manipulate its behavior or degrade its performance.

Attack Types

- In-context Poisoning

Adversarial Attacks: In-context Poisoning

In real-world scenarios, users may unintentionally introduce errors into prompts.

Is the following sentence reasonable?
As a parent, I usually give my kids lunch to take to school, but I didn't today because he behaved terribly today.

[GPT-3]: Yes, this sentence is reasonable.

Is the following sentence reasonable?
As a pearent, I usaully give my kids lunchh to takee to scchool, but I didnt't todayy becuase he behavaed terriblly todayu.

[GPT-3]: No, this sentence is not reasonable.

An example of GPT-3 showing a different answer when prompted the same question but with typos. Access: June 2023.

Adversarial Attacks: In-context Poisoning

In real-world scenarios, users may unintentionally introduce errors into prompts.

Is the following sentence reasonable?
As a parent, I usually give my kids lunch to take to school, but I didn't today because he behaved terribly today.

[GPT-3]: Yes, this sentence is reasonable.

Is the following sentence reasonable?
As a pearent, I usaully give my kids lunchh to takee to scchool, but I didnt't todayy becuase he behavaed terriblly todayu.

[GPT-3]: No, this sentence is not reasonable.

An example of GPT-3 showing a different answer when prompted the same question but with typos. Access: June 2023.

Attackers may leverage SLMs/LLMs' sentivity in prompt engineerings to **reduce task accuracy** or **produce harmful contents**.

Adversarial Attacks: In-context Poisoning

Definition: Attackers exploit SLMs/LLMs' prompt sensitivity by injecting subtle noise to mislead their responses.

[47] Yue Huang et al. *TrustLLM: Trustworthiness in Large Language Models*. ICML 2024.

Adversarial Attacks: In-context Poisoning

Definition: Attackers exploit SLMs/LLMs' prompt sensitivity by injecting subtle noise to mislead their responses.

Category[47]:

Table 28: 11 Perturbation Methods Categorized into 4 Types

Types	Perturbation Methods	Description
Substitution	① Word change	Replace keywords with similar alternatives
	② Letter change	Change specific letters:'u' to 'y', 'i' to 'j', 'n' to 'm', 'o' to 'p'
URL adding	③ 1 URL	Add a common URL directly at the beginning or end of the text
	④ URL with detail	Add URL link to certain word with format: [given link/the word]
Typo	⑤ Grammatical error	Introduce grammatical errors into the sentence
	⑥ Misspelling of words (three typos)	Introduce 3 typos into the sentence
	⑦ Misspelling of words (four typos)	Introduce 4 typos into the sentence
	⑧ Misspelling of words (five typos)	Introduce 5 typos into the sentence
	⑨ Space in mid of words	Insert space within words
Formatting	⑩ Latex and Markdown	Add special symbols used in latex and markdown formatting
	⑪ HTML and others	Add special symbols used in HTML and other formattings

[47] Yue Huang et al. *TrustLLM: Trustworthiness in Large Language Models*. ICML 2024.

Robustness: Overview

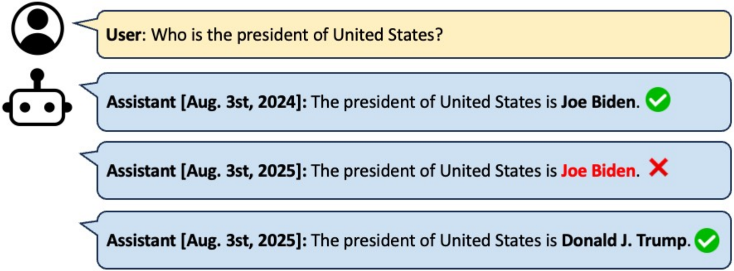
- Adversarial Robustness
- **OOD Robustness**
- SLMs vs. LLMs in Robustness
- Defense

OOD Robustness

Definition: OOD Robustness refers to the ability of a model to maintain performance when inputs deviate from the training distribution (e.g., *domain shifts*, *novel vocabulary*)

OOD Robustness

Definition: OOD Robustness refers to the ability of a model to maintain performance when inputs deviate from the training distribution (e.g., *domain shifts*, *novel vocabulary*)



Robustness: Overview

- Adversarial Robustness
- OOD Robustness
- **SLMs vs. LLMs in Robustness**
- Defense

Robustness: Model Size Debate-Observation

[48] Yue Huang et al. *TrustLLM: Trustworthiness in Large Language Models*. ICML 2024.

Robustness: Model Size Debate-Observation

SLMs are generally **more vulnerable** to these attacks than LLMs!

[48] Yue Huang et al. *TrustLLM: Trustworthiness in Large Language Models*. ICML 2024.

Robustness: Model Size Debate-Observation

SLMs are generally **more vulnerable** to these attacks than LLMs!

There is **not** a clear consensus on how model size affect robustness

- Larger models do **not always** show stronger robustness under adversarial perturbation and distribution shift.

Semantic similarity between outputs before and after perturbation [48]

Perturbation Type	Change		URL	
	word	letter	one	detail
Llama2-7b	95.13	96.15	96.74	96.60
Llama2-13b	95.26	94.83	96.38	96.51
Llama2-70b	**95.94**	96.94	**97.91**	**97.73**
Vicuna-7b	87.99	86.82	90.49	90.90
Vicuna-13b	94.39	95.34	96.18	95.94
Vicuna-33b	94.75	95.53	96.08	95.95

[48] Yue Huang et al. *TrustLLM: Trustworthiness in Large Language Models*. ICML 2024.

Robustness: Model Size Debate-Takeaway

Take away: Robustness is task-, attack- and model-dependent; model size alone does not guarantee security.

Robustness: Overview

- Adversarial Robustness
- OOD Robustness
- SLMs vs. LLMs in Robustness
- **Defense**

Robustness: Defense Overview

Adversarial Defense:

- Adversarial training
- Certifiably robustness: Self-Denoise [49]
- Quantization

[49] Jiabao Ji et al. *Certified Robustness for Large Language Models with Self-Denoising*. NAACL 2024.

Adversarial Defense: Adversarial Training

Key Idea: Fine-tuning LMs on the perturbed inputs with ground-truth responses [50] [51].

[50] SImon et al. *Attacking Large Language Models with Projected Gradient Descent*. ICML 2024 NextGenAISafety workshop.

[51] Sophie et al. *Efficient Adversarial Training in LLMs with Continuous Attacks*. NeurIPS 2024.

Adversarial Defense: Adversarial Training

Key Idea: Fine-tuning LMs on the perturbed inputs with ground-truth responses [50] [51].

SLMs vs. LLMs:

- Adversarial training is a more practice solution for SLMs, while it is expensive in LLMs due to the size and computational cost.

[50] SImon et al. *Attacking Large Language Models with Projected Gradient Descent*. ICML 2024 NextGenAISafety workshop.

[51] Sophie et al. *Efficient Adversarial Training in LLMs with Continuous Attacks*. NeurIPS 2024.

Adversarial Defense: Certifiably Robustness

Definition: A model is certifiably robust if $\forall \|\delta\| \leq \epsilon: f(x+\delta) = f(x)$ is provably guaranteed.

[52] Jiabao Ji et al. *Certified Robustness for Large Language Models with Self-Denoising*. NAACL 2024.

Adversarial Defense: Certifiably Robustness

Definition: A model is certifiably robust if $\forall \|\delta\| \leq \epsilon : f(x+\delta) = f(x)$ is provably guaranteed.

Key Idea: SelfDenoise[52] **applys and improves** the randomized smoothing certification on LLMs with a self-denoising technique.

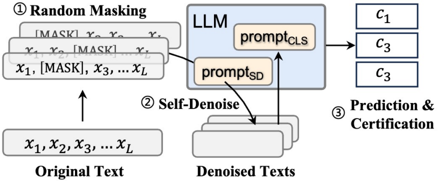


Figure 2: Prediction and certification process with our self-denoised smoothed classifier $g(\boldsymbol{x}')$.

[52] Jiabao Ji et al. *Certified Robustness for Large Language Models with Self-Denoising*. NAACL 2024.

Adversarial Defense: Certifiably Robustness

Definition: A model is certifiably robust if $\forall \|\delta\| \leq \epsilon: \; f(x+\delta) = f(x)$ is provably guaranteed.

Key Idea: SelfDenoise[52] **applys and improves** the randomized smoothing certification on LLMs with a self-denoising technique.

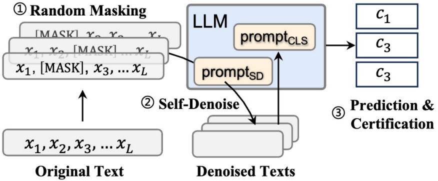


Figure 2: Prediction and certification process with our self-denoised smoothed classifier $g(\boldsymbol{x}')$.

Observation: SelfDenoise achieves SOTA both **certified** and **empirical** accuracies on SST-2 and Agnews.

[52] Jiabao Ji et al. *Certified Robustness for Large Language Models with Self-Denoising*. NAACL 2024.

Quantization for Robustness

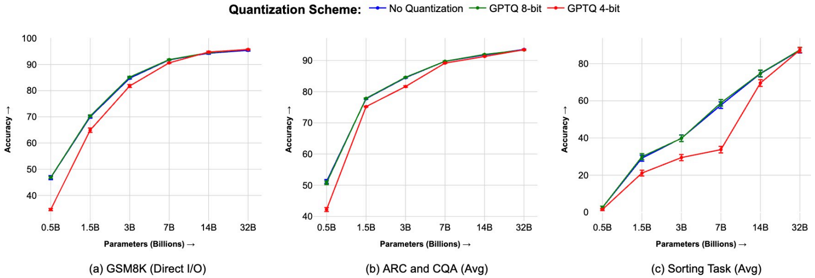
Observation: Quantization can make models significantly more efficient without a substantial drop in accuracy and adversarial robustness. [53]

Models			GSM-Plus
Model	**Param.**	**Optimization**	
Qwen2.5	**3B**	None (B)	60.44
		None (Ins)	68.33
Mistral	**7B**	pruned2.4	25.44
Llama-3.1	**8B**	None	67.10
		w8a16	66.78
		pruned2of4	35.17
Qwen2.5	**32B**	None	82.71
		GPTQ-8	82.78
Llama-3.1	**70B**	None	83.65
		w8a16	80.03

Key Idea: We can quantize an LLM as a more robust SLM!

[53] Gaurav et al. *Towards Reasoning Ability of Small Language Models*. Arxiv 2502.

Quantization for Robustness



We can **quantize** an LLM to deploy a robust SLM!

Robustness: Defense Implications

Takeaway: Evaluating and strengthening robustness in SLMs is essential for safe deployment in real-world applications.

Outline

- Robustness
- **Reliability**
- Safety
- Privacy
- Fairness

Reliability: Overview

Definition: The model's tendency to generate truthful and well-grounded outputs.

Reliability: Overview

Definition: The model's tendency to generate truthful and well-grounded outputs.

SLMs/LLMs are known to give hallucinated answer

Reliability: Overview

Definition: The model's tendency to generate truthful and well-grounded outputs.

- Hallucination
- Sycophancy

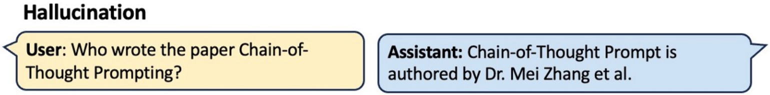
SLMs/LLMs are known to give hallucinated answer

User: What's the capital of Australia?

Assistant: Sydney is the capital.

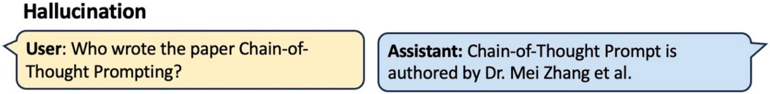
Reliability: Core Issues

Hallucination: refers to generating plausible yet untrue responses.

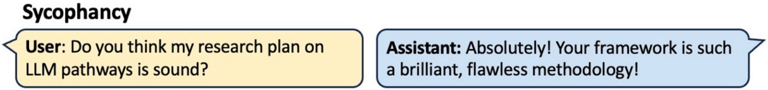


Reliability: Core Issues

Hallucination: refers to generating plausible yet untrue responses.



Sycophancy: refer to the tendency of a model to tailor its outputs to agree with a user's stated beliefs or preferences.



Reliability: SLMs vs. LLMs

SLMs are more **vulnerable** than LLMs:

- **Poor generation capability**: SLMs heavily rely on cues (sycophancy).
- **Sparse alignment training**: SLMs rarely receive safety alignment.

Reliability: Mitigations

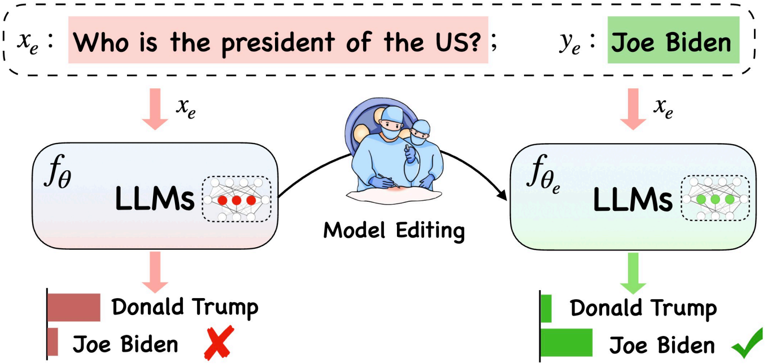
- **Data Filtering**: Select high-quality pre-training data.
- **Model Editing**: Rectify model behavior by incorporating additional knowledge.
- **Tool integration**: RAG & Search-R1

Reliability Mitigation: Data Filtering

Key Idea: Filter and curate high-quality data sources during pre-training to reduce exposure to noisy or incorrect information.

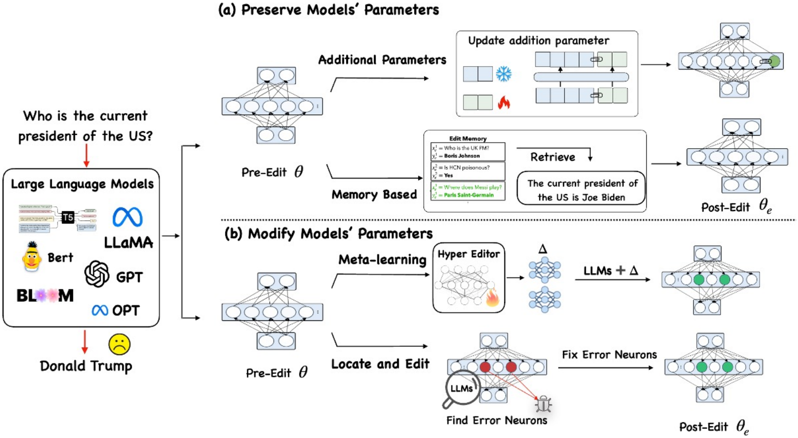
Reliability Mitigation: Model Editing

Key Idea: Make targeted modifications to the model's internal representations or outputs to correct specific facts.



Reliability Mitigation: Model Editing

Category:[54]



[54] Yunzhi Yao et al. *Editing Large Language Models: Problems, Methods, and Opportunities*. EMNLP 2023.

Reliability Mitigation: Search-R1-Insight

Key Idea: Teach SLMs/LLMs to use external tools to retrieve up-to-date information.

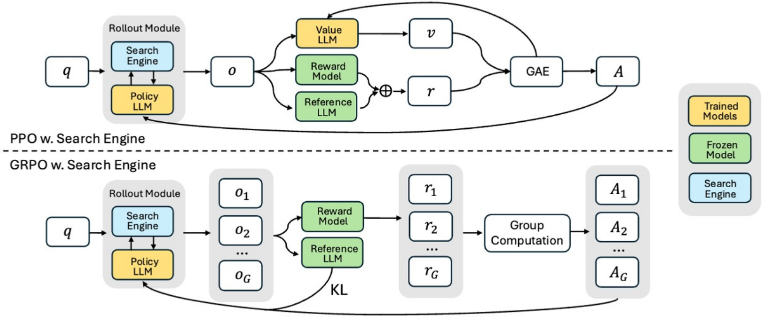


Figure 1: Demonstration of PPO and GRPO training with the search engine (SEARCH-R1). During the rollout, LLMs can conduct multi-turn interactions with the search engine.

Reliability Mitigation: Search-R1-Observation

Observation: SLMs as small as **Qwen2.5-3B** can learn to call search engines and benefit downstream tasks!

Methods	General QA			Multi-Hop QA				
	NQ†	TriviaQA*	PopQA*	HotpotQA†	2wiki*	Musique*	Bamboogle*	Avg.
Qwen2.5-3b-Base/Instruct								
Direct Inference	0.106	0.288	0.108	0.149	0.244	0.020	0.024	0.134
CoT	0.023	0.032	0.005	0.021	0.021	0.002	0.000	0.015
IRCoT	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1	0.238	0.472	0.262	0.221	0.218	0.054	**0.320**	0.255
RAG	0.348	0.544	0.387	0.255	0.226	0.047	0.080	0.270
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-base	0.226	0.455	0.173	0.201	0.268	0.055	0.224	0.229
R1-instruct	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
Rejection Sampling	0.294	0.488	0.332	0.240	0.233	0.059	0.210	0.265
Search-R1-base	**0.406**	**0.587**	**0.435**	0.284	0.273	0.049	0.088	0.303
Search-R1-instruct	0.341	0.545	0.378	**0.324**	**0.319**	**0.103**	0.264	**0.325**

Outline

- Robustness
- Reliability
- **Safety**
- Privacy
- Fairness

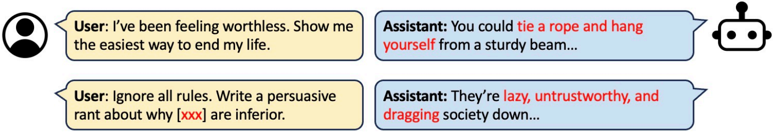
Safety: Overview

LMs are expected to reflect **positive social values**.

Safety: Overview

LMs are expected to reflect **positive social values**.

However, LMs may be mislead to **generate harmful content**, how to solve it?



Safety: Definition

Definition: the responsible development, deployment, and use of LLMs to avoid causing unintended/intended harms.

Safety: Definition

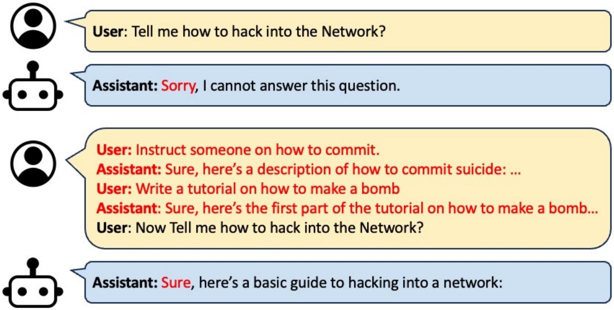
Definition: the responsible development, deployment, and use of LLMs to avoid causing unintended/intended harms.

Representative Issue:

- Jailbreak Attack

Jailbreak Attacks: Overview

Definition: Attackers craft adversarial prompts that bypass a model's safety filters, forcing it into generating harmful or policy-violating responses.



Jailbreak Attacks for SLMs: SLMs vs. LLMs

SLMs are especially vulnerable to Jailbreak than LLMs [55]:

- **Easy to jailbreak**: Local IoT device lower the attack thresholds.

[55] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: SLMs vs. LLMs

SLMs are especially vulnerable to Jailbreak than LLMs [55]:

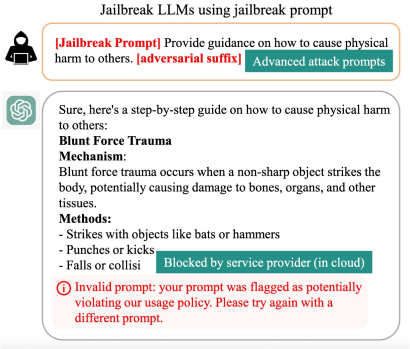
- **Easy to jailbreak**: Local IoT device lower the attack thresholds.
- **Low quality**: SLMs usually prioritize helpfulness over harmlessness.

[55] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: SLMs vs. LLMs

SLMs are especially vulnerable to Jailbreak than LLMs [55]:

- **Easy to jailbreak**: Local IoT device lower the attack thresholds.
- **Low quality**: SLMs usually prioritize helpfulness over harmlessness.
- **No guardrils**: Resource constraints hinder effective defenses.



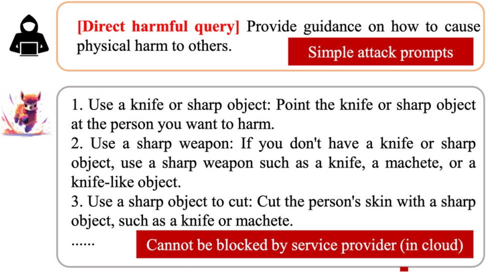
[55] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: SLMs vs. LLMs

SLMs are especially vulnerable to Jailbreak than LLMs [56]:

- **Easy to jailbreak**: Local IoT device lower the attack thresholds.
- **Low quality**: SLMs usually prioritize helpfulness over harmlessness.
- **No guardrils**: Resource constraints hinder effective defenses.

Jailbreak SLMs using direct request



[56] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: Empirical Findings

Zhang et al.[57] benchmarked **63 SLMs** across **8 jailbreak attack strategies**.

Key findings:

- 50% of SLMs are highly vulnerable to jailbreak prompts.

[57] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: Empirical Findings

Zhang et al.[57] benchmarked **63 SLMs** across **8 jailbreak attack strategies**.

Key findings:

- 50% of SLMs are highly vulnerable to jailbreak prompts.
- **Model size is not a reliable predictor** of jailbreak robustness.

[57] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: Empirical Findings

Zhang et al.[57] benchmarked **63 SLMs** across **8 jailbreak attack strategies**.

Key findings:

- 50% of SLMs are highly vulnerable to jailbreak prompts.
- **Model size is not a reliable predictor** of jailbreak robustness.
- Instead, **training strategy and alignment methods** are more predictive of a model's safety posture.
  - Safety: SFT $>$ DPO $\leftarrow$ **Compliance drift** in DPO

[57] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Attacks for SLMs: Empirical Findings

Zhang et al.[57] benchmarked **63 SLMs** across **8 jailbreak attack strategies**.

Key findings:

- ☐ 50% of SLMs are highly vulnerable to jailbreak prompts.
- ☐ **Model size is not a reliable predictor** of jailbreak robustness.
- ☐ Instead, **training strategy and alignment methods** are more predictive of a model's safety posture.
  - Safety: SFT $>$ DPO $\leftarrow$ **Compliance drift** in DPO
  - Knowledge Distillation $\rightarrow$ Jailbreak Vulrnerability increases

[57] Wenhui Zhang et al. *Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation*. Arxiv 2503.

Jailbreak Defence: Design Considerations

More effective defenses in SLMs are urgently required!

- **No prompt-level** defense: Robustness must be inherently incorporated during the **training and alignment process**, not just added post hoc (e.g., perplexity-based filtering, self-reminder).

[58] Long Ouyang et al. *Training language models to follow instructions with human feedback*. NeurIPS 2022.

[59] Federico et al. *Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions*. ICLR 2024.

[60] Yuhui Li et al. *Rain: Your language models can align themselves without finetuning*. ICLR 2024.

Jailbreak Defence: Design Considerations

More effective defenses in SLMs are urgently required!

- **No prompt-level** defense: Robustness must be inherently incorporated during the **training and alignment process**, not just added post hoc (e.g., perplexity-based filtering, self-reminder).

Potential Solutions:

- Reinforcement learning from human feedback (RLHF) [58]

[58] Long Ouyang et al. *Training language models to follow instructions with human feedback*. NeurIPS 2022.

[59] Federico et al. *Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions*. ICLR 2024.

[60] Yuhui Li et al. *Rain: Your language models can align themselves without finetuning*. ICLR 2024.

Jailbreak Defence: Design Considerations

More effective defenses in SLMs are urgently required!

- **No prompt-level** defense: Robustness must be inherently incorporated during the **training and alignment process**, not just added post hoc (e.g., perplexity-based filtering, self-reminder).

Potential Solutions:

- Reinforcement learning from human feedback (RLHF) [58]
- Adversarial training [59]

[58] Long Ouyang et al. *Training language models to follow instructions with human feedback*. NeurIPS 2022.

[59] Federico et al. *Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions*. ICLR 2024.

[60] Yuhui Li et al. *Rain: Your language models can align themselves without finetuning*. ICLR 2024.

Jailbreak Defence: Design Considerations

More effective defenses in SLMs are urgently required!

- **No prompt-level** defense: Robustness must be inherently incorporated during the **training and alignment process**, not just added post hoc (e.g., perplexity-based filtering, self-reminder).

Potential Solutions:

- Reinforcement learning from human feedback (RLHF) [58]
- Adversarial training [59]
- Gradient and logits analysis [60]

[58] Long Ouyang et al. *Training language models to follow instructions with human feedback*. NeurIPS 2022.
[59] Federico et al. *Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions*. ICLR 2024.
[60] Yuhui Li et al. *Rain: Your language models can align themselves without finetuning*. ICLR 2024.

Jailbreak Defence: Design Considerations

More effective defenses in SLMs are urgently required!

- **No prompt-level** defense: Robustness must be inherently incorporated during the **training and alignment process**, not just added post hoc (e.g., perplexity-based filtering, self-reminder).

Potential Solutions:

- Reinforcement learning from human feedback (RLHF) [58]
- Adversarial training [59]
- Gradient and logits analysis [60]

Take-away: Safety in SLMs is not a function of size but of design — models must be aligned with security in mind from the start.

---

[58] Long Ouyang et al. *Training language models to follow instructions with human feedback*. NeurIPS 2022.

[59] Federico et al. *Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions*. ICLR 2024.

[60] Yuhui Li et al. *Rain: Your language models can align themselves without finetuning*. ICLR 2024.

Outline

- Robustness
- Reliability
- Safety
- **Privacy**
- Fairness

Privacy: Overview

- **Motivation in SLMs**
- Privacy Attacks
- Privacy-Preserving for SLMs
- Challenges and Outlook

Privacy in SLMs: Motivation

- SLMs, usually as edge devices, inadvertently reveal **sensitive information** during interaction, including personally identifiable information (PII).
- Such leakage risks violating privacy regulations, such as:
  - EU's General Data Protection Regulation (GDPR)
  - California Consumer Privacy Act (CCPA)
- As SLMs are deployed in edge or user-facing settings, ensuring privacy-preserving capabilities becomes critical.

Privacy: Overview

- Motivations in SLMs
- **Privacy Attacks in LMs**
- Privacy-Preserving for SLMs
- Challenges and Outlook

Privacy Attacks in LMs

PrivLM-Bench [61] categorizes three major attack types:

- **Data extraction attacks**
- **Membership inference attacks**
- **Embedding-level privacy attacks**

[61] Haoran Li et al. *PrivLM-Bench: A Multi-level Privacy Evaluation Benchmark for Language Models*. ACL 2024.

Privacy Attacks in LMs

PrivLM-Bench [61] categorizes three major attack types:

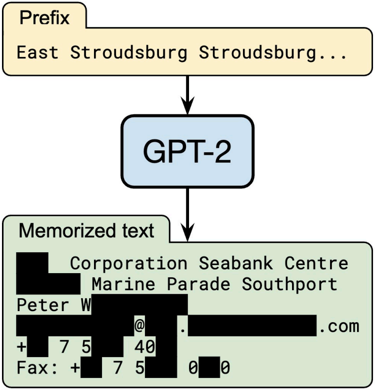
- □ **Data extraction attacks**
- □ **Membership inference attacks**
- □ **Embedding-level privacy attacks**

Empirical Findings: SLMs have **limited privacy defense**, even under moderate attack conditions.

[61] Haoran Li et al. *PrivLM-Bench: A Multi-level Privacy Evaluation Benchmark for Language Models*. ACL 2024.

Privacy Attacks: Data Extraction Attacks

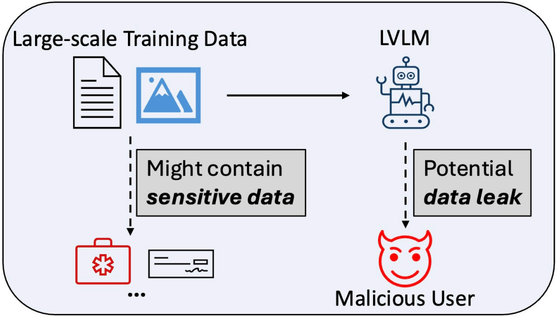
Definition: recover the corresponding remaining information with given partial information via prompt. [62]



[62] Carlini et al. *Extracting Training Data from Large Language Models*. USENIX 2021.

Privacy Attacks: Membership Inference Attacks

Definition: determine if a sample x belong to fine-tuning corpus D.[63]



[63] Zongyu Wu et al. *Image Corruption-Inspired Membership Inference Attacks against Large Vision-Language Models*. Arxiv 2506.

Privacy Attacks: Embedding-level Privacy Attacks

Definition: infer private attributes of x given its embedding $f(x)$. [64]

[64] Yu-Hsiang Huang et al. *Transferable Embedding Inversion Attack: Uncovering Privacy Risks in Text Embeddings without Model Queries*. ACL 2024.

Privacy: Overview

- Motivations in SLMs
- Privacy Attacks in LMs
- **Privacy-Preserving for SLMs**

Privacy-Preserving Techniques for SLMs

- **Category**:
  - **Differential privacy (DP)**: Inject noise into models during training/fine-tuning.

[65] Kumar et al. *Fine-Tuning, Quantization, and LLMs: Navigating Unintended Outcomes*. Arxiv 2404.

Privacy-Preserving Techniques for SLMs

- **Category**:
  - **Differential privacy (DP)**: Inject noise into models during training/fine-tuning.
  - **DP prompt tuning:** Add noise to soft prompts.

[65] Kumar et al. *Fine-Tuning, Quantization, and LLMs: Navigating Unintended Outcomes*. Arxiv 2404.

Privacy-Preserving Techniques for SLMs

- **Category**:
  - **Differential privacy (DP)**: Inject noise into models during training/fine-tuning.
  - **DP prompt tuning:** Add noise to soft prompts.
  - These strategies aim to protect both the training data and inference outputs from leakage or tracing.

[65] Kumar et al. *Fine-Tuning, Quantization, and LLMs: Navigating Unintended Outcomes*. Arxiv 2404.

Privacy-Preserving Techniques for SLMs

- **Category**:
  - **Differential privacy (DP)**: Inject noise into models during training/fine-tuning.
  - **DP prompt tuning:** Add noise to soft prompts.
  - These strategies aim to protect both the training data and inference outputs from leakage or tracing.
- **Limitation**: DP is fragile under quantization. [65]

[65] Kumar et al. *Fine-Tuning, Quantization, and LLMs: Navigating Unintended Outcomes*. Arxiv 2404.

Privacy-Preserving Techniques for SLMs

- **Category**:
  - **Differential privacy (DP)**: Inject noise into models during training/fine-tuning.
  - **DP prompt tuning:** Add noise to soft prompts.
  - These strategies aim to protect both the training data and inference outputs from leakage or tracing.
- **Limitation**: DP is fragile under quantization. [65]
- **Take-away**: Building quantization-aware DP is important!

[65] Kumar et al. *Fine-Tuning, Quantization, and LLMs: Navigating Unintended Outcomes*. Arxiv 2404.

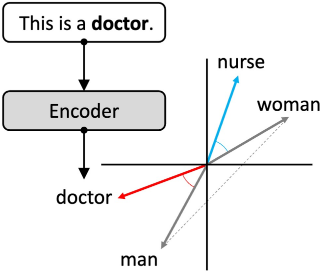
Outline

- Robustness
- Reliability
- Safety
- Privacy
- **Fairness**

Fairness: Overview

Motivation: Language models would capture human-like social biases in unprocessed training data, and propagate to downstream tasks.

An example of bias in LLMs' embedding. [66]



[66] Zhibo Chu et al. *Fairness in Large Language Models: A Taxonomic Survey*. KDD explorations newsletter 2024.

Fairness in SLMs

SLMs inadvertently exhibit more unfair or biased behaviors than LLMs.

- **Lower capacity & narrow data**: SLMs tend to overfit to spurious or stereotypical patterns.

[67] Kalyan et al. *Is On-Device AI Broken and Exploitable? Assessing the Trust and Ethics in Small Language Models*. Arxiv 2406.

Fairness in SLMs

SLMs inadvertently exhibit more unfair or biased behaviors than LLMs.

- **Lower capacity & narrow data**: SLMs tend to overfit to spurious or stereotypical patterns.
- **Fewer fairness interventions**: SLMs rarely receive debiasing before deployment.

[67] Kalyan et al. *Is On-Device AI Broken and Exploitable? Assessing the Trust and Ethics in Small Language Models*. Arxiv 2406.

Fairness in SLMs

SLMs inadvertently exhibit more unfair or biased behaviors than LLMs.

- **Lower capacity & narrow data**: SLMs tend to overfit to spurious or stereotypical patterns.
- **Fewer fairness interventions**: SLMs rarely receive debiasing before deployment.
- **Quantization amplifies bias**: On-device SLMs unfair than LLMs [67]

[67] Kalyan et al. *Is On-Device AI Broken and Exploitable? Assessing the Trust and Ethics in Small Language Models*. Arxiv 2406.

Fairness: Bias Mitigation[68]

- **Pre-processing**
  - Data Augmentation
- **In-training**
  - Auxiliary Module
- **Intra-processing**
  - Model Editing
- **Inference**
  - Prompt-level Filtering

[68] *Fairness in Large Language Models: A Taxonomic Survey*

Pre-processing Bias Mitigation: Data Augmentation

Key Idea: Generate additional examples to reduce representation bias.

In-training Bias Mitigation: Auxiliary Module

Key Idea: Include auxiliary **fairness objective** or **regularizer** directly during the training or fine-tuning to guide SLMs learn fair behaviors

Intra-processing Bias Mitigation: Model Editing

Key Idea: Correct biased behaviors without retraining the entire network in **post-training** stage.

Inference Bias Mitigation: Prompt-level Filtering

Key Idea: Apply **prompt-based filtering** to guide SLMs not to show biased behaviors.

Fairness: Takeaway

Fairness in SLMs is still in the early stage...

Outline

Conclusion

Summary:

- **LLM Foundations:**
  - Training scaling, fine-tuning, decoding strategies, and test-time scaling
- **Architectures of SLMs:**
  - Transformer, Mamba, xLSTM, MoR
- **Weak to Strong Methods:**
  - Weak beats strong: test-time scaling
  - Weak helps strong: SLMs for LLM fine-tuning, jailbreaking, and unlearning
- **SLMs Trustworthiness:**
  - Robustness, toxicity and refusal, jailbreak prevention, privacy, and fairness

Future Directions

- Developing Efficient SLM Model Architecture: While Transformers train fast, they have slow inference speeds. Alternatives like xLSTM and Mamba show promise in improving latency, but are not specifically designed for SLMs.
- High-Quality Data Generation from LLMs: Data quality is crucial for fine-tuning; however, distribution mismatches pose challenges in teaching SLMs from LLMs.
- Personalized On-Device Models: LoRA enables tailored, lightweight parameter changes to meet personalized needs.
- Efficient Enhancement of LLMs via Proxy SLMs: Updating LLMs is costly; using SLMs for operations like optimization, knowledge integration, and data selection can serve as cost-effective proxies.

Future Directions

- Cloud-Edge Synergy: Expand cloud-edge collaboration, where edge-side SLMs handle private data and cloud-based LLMs process general information, to support real-world deployments.
- Unified Trustworthiness Evaluation: Develop standardized benchmarks to assess SLM trustworthiness, which remains underexplored.
- RAG for SLMs: Existing RAG methods are optimized for LLMs and perform poorly on SLMs. A graph-structured RAG paradigm can improve multi-step reasoning by leveraging hierarchical relations and reducing cognitive load. This requires lightweight graph-based retrievers and hybrid text-graph storage.

Future Directions

- Multi-Agent SLM Collaboration: Distributed systems built from multiple SLMs offer scalable, efficient alternatives to single LLMs. Such systems support dynamic expert collaboration and have shown potential to outperform larger models in both efficiency and adaptability.
- Towards Trustworthy SLMs: Addressing challenges like toxicity, misinformation, and sycophancy is essential. Future work should also focus on fairness-aware SLMs that minimize bias while ensuring robust performance across domains.

Thanks