A Survey on Small Language Models in the Era of Large Language Models: Architecture, Capabilities, and Trustworthiness

Fali Wang
The Pennsylvania State University
University Park, USA
fqw5095@psu.edu

Minhua Lin
The Pennsylvania State University
University Park, USA
mfl5681@psu.edu

Yao Ma
Rensselaer Polytechnic Institute
Troy, USA
may13@rpi.edu

Hui Liu
Amazon
Palo Alto, USA
liunhu@amazon.com

Qi He
Amazon
Palo Alto, USA
qih@amazon.com

Xianfeng Tang
Amazon
Palo Alto, USA
tangxianfeng@outlook.com

Jiliang Tang
Michigan State University
East Lansing, USA
tangjili@msu.edu

Jian Pei
Duke University
Durham, USA
j.pei@duke.edu

Suhang Wang*
The Pennsylvania State University
University Park, USA
szw494@psu.edu

Abstract

Large language models (LLMs) based on Transformer architecture are powerful but face challenges with deployment, inference latency, and costly fine-tuning. These limitations highlight the emerging potential of small language models (SLMs), which can either replace LLMs through innovative architectures and technologies, or assist them as efficient proxy or reward models. Emerging architectures such as Mamba and xLSTM address the quadratic scaling of inference with window length in Transformers by enabling linear scaling. To maximize SLM performance, test-time compute scaling strategies reduce the performance gap with LLMs by allocating extra compute budget during test time. Beyond standalone usage, SLMs could also assist in LLMs via weak-to-strong learning, proxy tuning, and guarding, fostering secure and efficient LLM deployment. Lastly, the trustworthiness of SLMs remains a critical yet underexplored research area. However, there is a lack of tutorials on cutting-edge SLM technologies, prompting us to conduct one.

CCS Concepts

• **Computing methodologies → Natural language generation**.

Keywords

Small Language Models, Weak-to-strong, Trustworthiness

ACM Reference Format:
Fali Wang, Minhua Lin, Yao Ma, Hui Liu, Qi He, Xianfeng Tang, Jiliang Tang, Jian Pei, and Suhang Wang. 2025. A Survey on Small Language Models in the Era of Large Language Models: Architecture, Capabilities, and Trustworthiness. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3711896.3736563

1 Introduction

Large language models (LLMs) have revolutionized various fields such as AI for science [7, 67, 94, 102], programming [84], and human-centered interaction [121]. However, their massive parameter sizes and commonly used Transformer architecture often pose significant challenges: (1) common local devices cannot load the large-scale parameters and hard to handle the cache storage; (2) huge computational amount cause large inference latency unsuitable for real-time applications; and (3) they make domain-specific fine-tuning computationally demanding. Consequently, small language models (SLMs), which excel in efficiency, cost, and flexibility, have emerged with new potential in the era of LLMs, offering support or replacing LLMs to overcome these challenges. We collect a series of newly released SLMs in Table 1.

The Transformer architecture [93], driven by self-attention, remains the preferred framework for LLMs. To enable local deployment and enhance performance, smaller-scale Transformers are being explored [64, 90] by optimizing components including activation functions, attention mechanisms, layer normalization, and parameter reuse. Despite its strengths, the Transformer's reliance on historical key-value pairs causes inefficient inference, facing quadratic computational complexity $O(L^2)$ and high KV cache memory demands for long sequences. To address these issues, ongoing research is developing new architectures like the Mamba series [20, 22, 28, 30, 50, 115] and xLSTM [13] to improve inference efficiency and memory usage.

Deploying SLMs often results in inferior performance on specific tasks compared to LLMs. To bridge performance gaps, test-time compute scaling [61, 85, 104] proves effective by providing additional computation during inference, such as repeated sampling and verifier-based selection, significantly boosting model performance. Beyond simply replacing LLMs, SLMs could assist

[*] Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '25, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736563

in enhancing their usage and security via weak-to-strong learning [16, 55, 70, 86, 113, 129], proxy tuning [60, 71, 122], decoding [56, 84, 128], and guard [45, 48]. During weak-to-strong learning, SLMs generate weakly supervised datasets that activate LLMs' knowledge. They also function as proxies for fine-tuning by integrating log probabilities with behavioral deltas from smaller models and serve as reward models to guide LLMs during decoding without additional training. Additionally, SLMs enhance LLM security by filtering out potential attack samples, thus mitigating safety risks in conversational AI. Importantly, it is essential to address trustworthiness issues like susceptibility to adversarial attacks and privacy breaches [21, 38, 95]. To summarize, our key contributions include:

- In Section 2, we explore various techniques for enhancing Transformers for SLMs and introduce new architectures suitable for small models.
- In Section 3, we examine strategies for elevating SLMs from weak to strong through test-time scaling and discuss how weak SLMs can support strong LLMs in fine-tuning, decoding, and guarding.
- In Section 4, we assess the trustworthiness of SLMs, focusing on issues including robustness, toxicity, misinformation, hallucination, sycophancy, privacy, and fairness, while providing a comprehensive taxonomy of current evaluation methodologies.

2 SLM Architecture

SLMs commonly employ the Transformer architecture [93], which utilizes self-attention mechanisms to manage long-range text dependencies. Due to the quadratic-time inference associated with the attention mechanism, several subquadratic-time architectures, such as Mamba [30] and xLSTM [13], have been proposed. We have collected newly-released SLMs based on these architectures in Table 1. These architectures are detailed below.

Table 1: Overview of New SLMs (<7B) in 2024 (Excluding Mamba-1, Released in 2023). We correct several erroneous activations in the SLMs collection in [66]. Architecture Symbols: $\mathcal{T}$: Transformer, $\mathcal{M}$: Mamba, $\mathcal{X}$: xLSTM.

Model	Size	Date	Attention	Activation	Layer Norm
$\mathcal{T}$: Phi-4-mini [2]	3.8B	2025.03	GQA	GEGLU	RMSNorm
$\mathcal{T}$: SmolLM-2 [9]	135M, 360M, 1.7B	2025.02	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: DeepSeek-R1-Distill-Qwen-1.5B [32]	1.5B	2025.01	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: PhoneLM [116]	0.5B, 1.5B	2024.11	MHA	ReLU	RMSNorm
$\mathcal{M}$: Hymba [22]	1.5B	2024.11	SSM+TF	GEGLU	LayerNorm
$\mathcal{T}$: MiniCPM-3 [41]	4B	2024.09	MLA	UNK	RMSNorm
$\mathcal{T}$: Llama-3.2 [5]	1B, 3B	2024.09	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: Qwen-2.5 [112]	0.5B, 1.5B, 3B	2024.09	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: Phi-3.5-mini [1]	2.7B	2024.09	GQA	GEGLU	RMSNorm
$\mathcal{T}$: DCLM [54]	1.4B	2024.08	MHA	SwiGLU	LayerNorm
$\mathcal{T}$: H2O-Danube3 [77]	0.5B, 4B	2024.07	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: Fox-1 [89]	1.6B	2024.07	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: SmolLM [10]	135M 360M 1.7B	2024.07	GQA GQA MHA	SwiGLU SwiGLU SwiGLU	RMSNorm RMSNorm RMSNorm
$\mathcal{T}$: Minitron [73]	4B	2024.07	MHA	GELU	LayerNorm
$\mathcal{T}$: Gemma-2 [88]	2B	2024.07	GQA	GELU	RMSNorm
$\mathcal{T}$: Qwen-2 [111]	1.8B, 4B	2024.06	GQA	SwiGLU	RMSNorm
$\mathcal{M}$: Zamba [28]	1.2B	2024.05	SSM+TF	UNK	LayerNorm
$\mathcal{X}$: xLSTM [13]	125M, 350M, 760M, 1.3B	2024.05	LSTM	GELU	LayerNorm
$\mathcal{M}$: Mamba 2 [20]	2.7B	2024.05	SSM	SwiGLU	LayerNorm
$\mathcal{T}$: OpenELM [69]	270M, 450M 1.1B, 3B	2024.04	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: OLMo [29]	1.2B	2024.04	MHA	SwiGLU	RMSNorm
$\mathcal{T}$: Phi-3-mini [1]	3.8B	2024.04	GQA	GEGLU	RMSNorm
$\mathcal{T}$: MiniCPM [41]	1B, 2B	2024.04	GQA	UNK	RMSNorm
$\mathcal{T}$: MobiLlama [90]	0.5B, 1B	2024.02	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: Gemma-1 [87]	2B	2024.02	MQA	GELU	RMSNorm
$\mathcal{T}$: Qwen-1.5 [12]	0.5B	2024.02	GQA	SwiGLU	RMSNorm
$\mathcal{T}$: StableLM-2-zephyr [14]	1.6B	2024.01	MHA	SwiGLU	LayerNorm
$\mathcal{M}$: Mamba 1 [30]	130M, 370M, 790M, 1.4B, 2.8B	2023.12	SSM	SwiGLU	LayerNorm

2.1 Transformer for SLMs

Transformer architecture [93] remains the dominant framework for LLMs, utilized across platforms from open-source like Llama [91] to proprietary systems like GPT-4 [3]. Given the extensive use of LLMs, considerable efforts are focused on enhancing Transformer-based SLMs to closely match the performance of their larger counterparts. Generally, in the Transformer architecture, input tokens receive token embeddings and are processed through self-attention that dynamically weights input importance; this output feeds into a feed-forward neural network with an activation function for non-linear processing, and each output is normalized by layer normalization to stabilize learning and facilitate smooth gradient flow. This section examines how Transformers advance SLMs, focusing on core components, including the attention mechanism, activation functions in feed-forward networks, and layer normalization, as well as their strategic parameter sharing.

Self-Attention Mechanism. The self-attention mechanism enables models to assess the significance of all preceding token representations when encoding a current token. It is mathematically expressed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}^K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}^V$ denote the query, key, and value matrices, respectively. These matrices are scaled by $\sqrt{d_k}$ to ensure stability, with d_k denoting the dimension of the key matrices. The dot product $\mathbf{Q}\mathbf{K}^{\top}$ measures the similarity between the query and key vectors. For encoding the current position, the query might be represented as a vector $\mathbf{q}$, resulting in an output that is a weighted vector derived from prior values corresponding to previous token representations. **Multi-Head Attention (MHA)** leverages multiple "heads" to capture diverse information from various representation subspaces. Each head in the Multi-Head Attention mechanism operates independently, allowing the model to process information across different subspaces. This captures a broader range of data features.

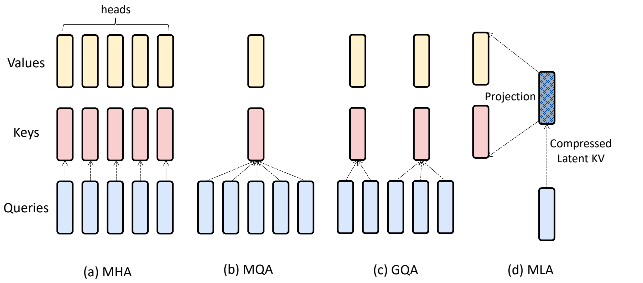


Figure 1: Multi-head Attention Mechanism and its variants.

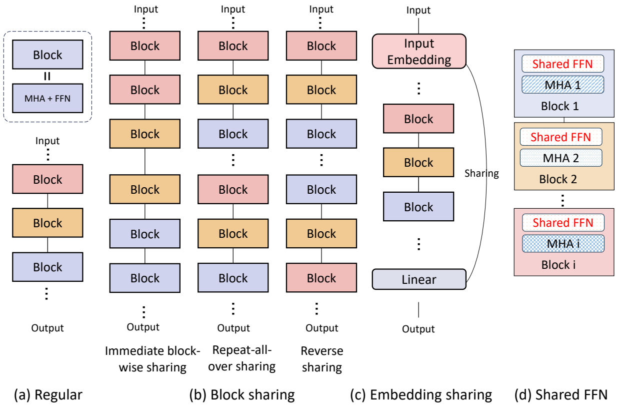
Table 2: Comparison of Activation Functions.

Name	Activation Function
ReLU [4]	$f(x) = \max(0, x)$
GELU [36]	$x \cdot \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$
GEGLU [83]	$x \cdot \mathrm{GELU}(Wx + b)$
SwiGLU [83]	$\mathrm{Swish}(x \cdot W + b) \odot (x \cdot V + c)$, $\mathrm{Swish}(x) = x \cdot \mathrm{sigmoid}(x)$

Building on this, many variants have improved the self-attention mechanism for memory efficiency and computational speed including MQA [82], GQA [6], and MLA [59], as shown in Figure 1. Multi-Query Attention (MQA) [82] addresses the KV cache bottleneck in MHA by proposing that all heads share the same keys and values, thus reducing memory and computational overhead. Grouped Query Attention (GQA) [6] strikes a balance by assigning subgroups of query heads to share a single key and value head, minimizing the number of key-value pairs. In contrast, Multi-Head Latent Attention (MLA) [59] compresses keys and values into a latent vector, enhancing management and boosting inference efficiency. *Recent SLMs often utilize GQA in their self-attention mechanisms, offering a flexible balance between reducing cache space and maintaining functionality.*

Feedforward Network (FFN). FFN comprises two linear transformations separated by a non-linear activation function, typically represented as: $\mathrm{FFN}(\mathbf{x}) = \sigma(\mathbf{x}\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2$, where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices, and b_1 and b_2 are bias terms. σ introduces non-linearity, allowing the model to learn complex patterns. Popular activation functions include ReLU [4], GELU [36], GEGLU [83], and SwiGLU [83], shown in Table 2. *ReLU promotes sparsity, which speeds up computation, while SwiGLU offers parameterized flexibility for various tasks, making it favored in SLMs for its effectiveness.*

Layer Normalization. Layer normalization [49] enhances training stability by normalizing the outputs of layers, thereby accelerating convergence. There are primarily two types of layer normalization techniques utilized: (i) **Non-Parametric Layer Norm**, which normalizes inputs based on the mean and variance calculated across the dimensions of a layer without any learnable parameters. (ii) **Parametric Layer Norm**, which incorporates learnable parameters, to allow for adaptive scaling and bias, thus enhancing the model's flexibility. In addition, **RMS Norm (Root Mean Square Layer Normalization)** [120] simplifies computations by utilizing the root mean square of the inputs. *Owing to its robustness in expressiveness, RMS Norm has gained popularity over traditional Layer Norm in small language models.*



(a) Regular (b) Block sharing (c) Embedding sharing (d) Shared FFN

Figure 2: Parameter sharing technologies.

Parameter Sharing. In SLM architectures, parameter-sharing techniques are crucial for reducing model size and space usage. As shown in Figure 2, parameter sharing techniques can be classified into three categories: (i) block sharing, which includes immediate block sharing, repeat-all-over sharing, and reverse sharing, (ii) embedding sharing where the input embedding and the final output weight are identical, and (iii) FFN sharing that involves using a common FFN module across all transformer layers. These approaches can be combined for enhanced efficiency. Repeat-all-over sharing generally performs the best in block sharing, whereas immediate block sharing offers memory savings due to the efficient use of shared cache among closely positioned blocks. Embedding sharing can reduce parameters by about 20% in SLMs, and FFN sharing can achieve a 60% reduction, significantly enhancing parameter efficiency while presenting a viable performance-parameter trade-off.

2.2 Mamba

Transformer architecture suffers from quadratic computational complexity. Mamba series [20, 22, 28, 30, 50, 115] aims to improve inference efficiency and memory usage.

2.2.1 From SSM to Mamba.

State space models [31] utilize the minimal number of variables to describe a dynamic system and can process sequential data as follows:

$$\boldsymbol{h}_k = \overline{\boldsymbol{A}}\boldsymbol{h}_{k-1} + \overline{\boldsymbol{B}}\boldsymbol{x}_k, \quad \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{h}_k \tag{2}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are the input-output sequence pairs, $\overline{\boldsymbol{A}}$, $\overline{\boldsymbol{B}}$, and $\boldsymbol{C}$ are learnable weight matrices, and $\boldsymbol{h}_k$ is the hidden state at time step k. However, traditional SSMs have fixed parameters and treat all input tokens uniformly, which constrains their utility in language modeling. To overcome this limitation, two key innovations transform SSMs into the Mamba architecture:

(1) Selective Scanning Mechanism. Traditional SSMs cannot perform selective copying or induction head tasks due to their Linear Time Invariant (LTI) nature, with fixed parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Delta}$ (where Δ denotes the discrete step size in SSMs) for all input tokens. This design inhibits content-aware inference, essential for language models, as it treats each token uniformly. To overcome this, Mamba introduces input-dependent parameters $\boldsymbol{B}(x), \boldsymbol{C}(x), \boldsymbol{\Delta}(x)$, allowing a unique set of $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Delta}$ for each input token ($\boldsymbol{A}$ remains unchanged).

(2) Hardware-Aware Algorithm. Frequent data transfers between GPUs' DRAM and SRAM reduce computational efficiency. To address this issue, a hardware-aware algorithm is designed to minimize these transfers by integrating discrete steps, selective scanning, and multiplication with $\boldsymbol{C}$ into a single kernel fusion operation.

2.2.2 Mamba Blocks.

Multiple Mamba blocks can be stacked and used sequentially, similar to transformer layers, as depicted in Figure 3. Initially, input tokens are linearly embedded into a hidden space, then processed by convolutional layers coupled with the SiLU activation function. Subsequently, a selective SSM module efficiently processes these embeddings to capture context-specific

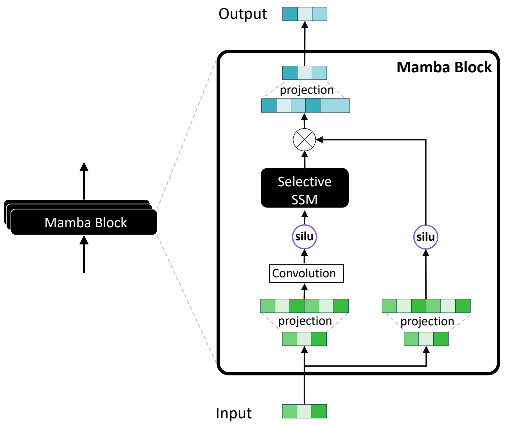


Figure 3: Mamba 1 architecture [30].

information. The resulting representations undergo additional nonlinear activation for enhanced expressivity, and a final linear projection maps the features into output space.

2.2.3 Mamba-Derivative Work. Mamba architecture has inspired various innovations focused on fuing Transformer and SSM. Mamba-2 [20] utilizes the structured state space duality to increase computational efficiency, while Jamba [50] alternates attention and SSM layers. In lightweight design, Hymba [22] combines attention and SSM heads in small models (1.5B) for enhanced memory efficiency, and Zamba [28] utilizes a Mamba backbone and a shared attention module to reduce memory use. LongMamba [115] improves long-context capabilities with a token filtering mechanism, enhancing document comprehension. Overall, these initiatives utilize Mamba's linear complexity to enhance efficient sequential modeling.

2.3 xLSTM

Long Short-Term Memory (LSTM) [37], similar in concept to Mamba [30] for its introduction of time-dependent weights, was pivotal in establishing foundational techniques for language modeling. However, the advent of Transformers [93] has marked a substantial paradigm shift in this field. This raises an intriguing question: Can LSTMs be effectively scaled to billion-parameter models with contemporary technologies? In response, the Extended Long Short-Term Memory (xLSTM) framework [13] has been developed to augment traditional LSTMs by incorporating exponential gating and innovative memory structures. This framework is bifurcated into two distinct components: sLSTM, which utilizes memory mixing for enhanced data integration, and mLSTM, which eschews mixing to facilitate parallel processing. Together, these elements comprise the comprehensive xLSTM architecture, as depicted in Figure 4 and further detailed below.

2.3.1 LSTM. LSTMs mitigate the vanishing gradient problem of RNNs through the constant error carousel and gating mechanisms by the following update rules at time step t:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{z}_t \qquad \text{(cell state)}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tilde{\mathbf{h}}_t, \qquad \tilde{\mathbf{h}} = \psi(\mathbf{c}_t) \qquad \text{(hidden state)}$$

$$\mathbf{z}_t = \varphi(\tilde{\mathbf{z}}_t), \qquad \tilde{\mathbf{z}}_t = \mathbf{w}_z^\top \mathbf{x}_t + \mathbf{r}_z \mathbf{h}_{t-1} + b_z \qquad \text{(cell input)}$$

$$\mathbf{i}_t = \sigma(\tilde{\mathbf{i}}_t), \qquad \tilde{\mathbf{i}}_t = \mathbf{w}_i^\top \mathbf{x}_t + \mathbf{r}_i \mathbf{h}_{t-1} + b_i \qquad \text{(input gate)}$$

$$\mathbf{f}_t = \sigma(\tilde{\mathbf{f}}_t), \qquad \tilde{\mathbf{f}}_t = \mathbf{w}_f^\top \mathbf{x}_t + \mathbf{r}_f \mathbf{h}_{t-1} + b_f \qquad \text{(forget gate)}$$

$$\mathbf{o}_t = \sigma(\tilde{\mathbf{o}}_t), \qquad \tilde{\mathbf{o}}_t = \mathbf{w}_o^\top \mathbf{x}_t + \mathbf{r}_o \mathbf{h}_{t-1} + b_o \qquad \text{(output gate)}$$

where weight vectors $\mathbf{w}_z$, $\mathbf{w}_i$, $\mathbf{w}_f$, $\mathbf{w}_o$ correspond to the inputs $\mathbf{x}_t$, and cell, input, forget, and output gates. Recurrent weights $\mathbf{r}_z$, $\mathbf{r}_i$, $\mathbf{r}_f$, $\mathbf{r}_o$ link the hidden state $\mathbf{h}_{t-1}$ to these gates. Bias terms are b_z, b_i, b_f, and b_o. Activation functions φ (usually tanh) and ψ normalize the otherwise unbounded cell state. All gates use sigmoid activation.

LSTMs have three key limitations: First, they struggle with revising dynamic storage decisions. Second, they have limited storage capacity, as all data is compressed into a single cell state $\mathbf{c}$. Third, their parallel processing capabilities are hindered by memory mixing, necessitating sequential hidden-to-hidden connections.

2.3.2 sLSTM. To improve the ability of LSTMs to revise storage decisions, exponential gates have been introduced. Particularly, input and forget gates can utilize exponential activation functions. For normalization purposes, a normalizer state is introduced that aggregates the product of the input gate with all subsequent forget gates. Based on the traditional LSTM, the sLSTM changes the input gate to $\mathbf{i}_t = \exp(\tilde{\mathbf{i}}_t)$, the forget gate to $\mathbf{f}_t = \sigma(\tilde{\mathbf{f}}_t)$ or $\exp(\tilde{\mathbf{f}}_t)$, adds a normalizer state $\mathbf{n}_t = \mathbf{f}_t \odot \mathbf{n}_{t-1} + \mathbf{i}_t$, and changes the hidden state to $\mathbf{h}_t = \mathbf{o}_t \odot \tilde{\mathbf{h}}_t$, where $\tilde{\mathbf{h}}_t = \frac{\mathbf{c}_t}{\mathbf{n}_t}$.

The sLSTM enables multiple memory cells similar to the original LSTM via recurrent connections from the hidden state h to memory cell inputs $\mathbf{z}$. This setup supports multiple heads without memory mixing across the heads, but only memory mixing across cells within each head.

2.3.3 mLSTM. To improve LSTM storage capacity, the memory cell dimensionality is increased from a vector $\mathbf{c}$ to a matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$. Based on the traditional LSTM, the mLSTM modifies the cell state equation to $\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot (\mathbf{v}_t \mathbf{k}_t^\top)$, introduces $\mathbf{n}_t = \mathbf{f}_t \odot \mathbf{n}_{t-1} + \mathbf{i}_t \odot \mathbf{k}_t$, and defines $\mathbf{q}_t = \mathbf{W}_q \mathbf{x}_t + \mathbf{b}_q$, $\mathbf{k}_t = \frac{1}{\sqrt{d}}(\mathbf{W}_k \mathbf{x}_t + \mathbf{b}_k)$, $\mathbf{v}_t = \mathbf{W}_v \mathbf{x}_t + \mathbf{b}_v$. It changes the input gate to $\mathbf{i}_t = \exp(\tilde{\mathbf{i}}_t)$, the forget

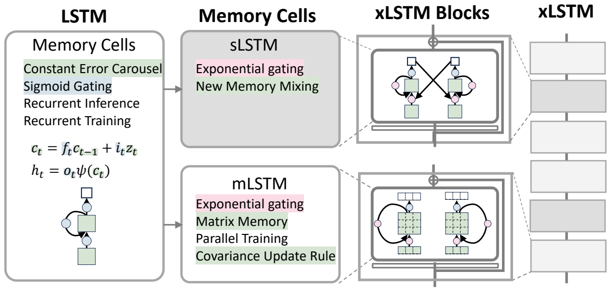


Figure 4: The extended LSTM (xLSTM) family [13].

gate to $\mathbf{f}_t = \sigma(\tilde{\mathbf{f}}_t)$ or $\exp(\tilde{\mathbf{f}}_t)$, and the hidden state to $\mathbf{h}_t = \mathbf{o}_t \odot \tilde{\mathbf{h}}_t$, where $\tilde{\mathbf{h}}_t = \frac{\mathbf{C}_t \mathbf{q}_t}{\max\{|\mathbf{n}_t^\top \mathbf{q}_t|, 1\}}$.

2.3.4 xLSTM. xLSTM has two block types: (i) post up-projection, using sLSTM with optional convolutions and gated MLPs, and (ii) pre up-projection, using mLSTM enclosed in two MLPs via convolution, learnable connections, and output gates. xLSTMs are built with these blocks and feature LayerNorm residual backbones. xLSTM is compared with Llama [45], and Mamba [30] across model sizes (125M, 350M, 760M, 1.3B). In the PALOMA benchmark [68], it excels in length extrapolation and outperforms same-size baselines in downstream tasks. xLSTM shows better scalability and higher throughput than Transformers and SSMs due to efficient memory use, allowing larger batches. These findings highlight xLSTM's potential in small model applications, though scaling to larger models remains unexplored. Due to xLSTM's unique efficiency in long-term sequence modeling, it has become the preferred architecture for many follow-up studies, including modeling long-range dependencies in biological and chemical sequences [80], time series forecasting [8], audio [110], and stock prediction [25].

3 Weak to Strong Methods

In the era of large language models, small language models maintain advantages due to their easy deployment, despite typically underperforming compared to large language models. To bridge this gap, methods have been developed to enhance SLMs, enabling them to beat LLMs in specific scenarios, noted as "SLMs from weak to strong." Moreover, SLMs' lightweight nature facilitates their usage in supporting LLMs through fine-tuning, decoding, and safeguarding, illustrating another aspect of the "weak-to-strong" strategy.

3.1 SLMs Beat LLMs in Specific Scenarios

Typically, SLMs underperform LLMs due to smaller parameter sizes. However, developers employ technologies such as supervised fine-tuning, distillation, and quantization to enhance SLMs. Among them, test-time scaling is a cutting-edge technology that can potentially beat LLMs, so we highlight recent work on it. For more details on other technologies enhancing SLMs, see Wang et al. [96].

Test-Time Compute Scaling. Test-time compute scaling (also known as inference scaling) enhances language models by allocating more computational resources during test time, utilizing strategies such as best-of-N [57], majority voting [100], and tree search [108]. A typical best-of-N approach involves sampling multiple outputs from the model, which are then assessed by a reward-based verifier to select the optimal one. Both Snell et al. [85] and Wu et al. [104] explore the trade-off between test-time scaling and training scaling. Snell et al. [85] found that SLMs with test-time compute scaling outperform models that are 14 times larger. Wu et al. [104] indicates that smaller models equipped with advanced test-time scaling algorithms offer superior performance. Further research by Liu et al. [61] shows that under test-time scaling strategies, smaller models excel all while requiring fewer FLOPs, e.g., a 0.5B SLM outperforms GPT-4o, a 3B LLM surpasses a 405B LLM, and a 7B LLM exceeds both o1 and DeepSeek-R1 [32].

Compute-optimal Test-Time Scaling. Compute-optimal test-time scaling aims to allocate the optimal compute for each problem, being defined as selecting optimal test-time scaling hyperparameters for a given prompt, such as scaling strategies, policy model selection, model size, and prompt designs. Various studies define the compute-optimal test-time scaling strategy, each differing in the selection space of hyperparameters [58, 61, 85]. Snell et al. [85] first formally defines $\text{Target}(\theta, N, q)$ as the distribution over output tokens induced by the language model for a given prompt q, using test-time compute hyper-parameters θ, and a compute budget of N. The goal is to select optimal hyperparameters θ that maximize accuracy for a given problem. Formally:

$$\theta^*_{q,a^*(q)}(N) = \arg\max_{\theta} \left(\mathbb{E}_{y\sim\text{Target}(\theta,N,q)} \left[\mathbf{1}_{y=y^*(q)} \right] \right) \quad (3)$$

where $y^*(q)$ is the ground-truth response for q, and $\theta^*_{q,a^*(q)}(N)$ indicates the strategy for problem q within a given budget N. This definition shows that scaling strategies are question-dependent, which slightly diverges from the concurrent study [104]. The study [61] suggests incorporating reward models in scaling decisions.

Strategies for Compute-optimal Scaling. Building on this problem, existing works design specific test-time compute-optimal scaling methods. Snell et al. [85] finds that the efficacy of methods varies with the computational budget and problem difficulty: Beam search excels with complex problems and limited compute budget; best-of-N is more effective for simpler problems and higher budgets. Inspired by this, they adapt search settings to take into account problem difficulty and compute budget, significantly enhancing performance. Additionally, Liu et al. [61] further refines this method to account for variability in reward models, as they perform differently at various inference lengths. For instance, tokens scaled with RLHFlow-PRM-Deepseek-8B are consistently larger than those of RLHFlowPRM-Mistral-8B—nearly double—owing to the longer training data length in DeepSeek PRM compared to Mistral-PRM-Data.

3.2 SLMs Help LLM Fine-tuning

This subsection introduces two paradigms by which SLMs can assist in fine-tuning of LLMs: **weak-to-strong learning** and **proxy fine-tuning**. *First, weak-to-strong learning can guide LLMs with SLM-generated datasets.* As LLMs advance and often outperform humans in complex tasks, the challenge of providing high-quality data emerges due to potentially simplistic or incorrect human annotations [16, 55, 113, 123, 129]. Burns et al. [16] introduced the *weak-to-strong learning* to explore if weak SLMs can effectively guide strong LLMs. This method is effective for two reasons: (1) strong models can replicate weak models, including their errors, while maintaining robust task representations, and (2) even inaccurate supervision from weak models can trigger pre-existing knowledge or capabilities. Strong models trained with weak supervision often underperform those fine-tuned on ground-truth data. To bridge this gap, Yang et al. [113] and Zhou et al. [129] focus on increasing data utilization effectiveness from weak models, employing strategies such as contrastive learning on incorrect samples and intensively learning on samples where LLMs are overconfident. SLMs could also enhance data quality. For example, Superfiltering [55] leverages SLMs to filter low-quality data for LLMs via difficulty-aware selection. *Second, proxy fine-tuning with SLMs can approximate gradients for fine-tuning large-scale LLMs on target* *datasets [60, 71, 122, 124]*. Fine-tuning LLMs is costly, while SLMs can effectively approximate the gradients required for fine-tuning LLMs. For instance, both Proxy-Tuning [60] and Emulated Fine-Tuning (EFT) [71] fine-tune a smaller LM and apply the predictive differences from the tuned and untuned small LMs to modify the original predictions of a larger, untuned model, while maintaining the benefits of large-scale pretraining. In a related direction, LoRAM [122] employs LoRA on a pruned SLM, obtaining incremental low-rank matrices that are subsequently restored onto the original large model during inference. Zhang et al. [124] propose training LLM agents without modifying weights by forging agent functions, which can potentially be implemented using small language models.

3.3 SLMs Guide LLM Decoding

Learning from human feedback has gained widespread attention due to its ability to utilize human-annotated data to align with human preferences, primarily by maximizing expected reward scores from implicit or explicit reward models. Alternatively, SLMs could serve as reward models for the alignment. Typically, this alignment process involves a greedy search at test time aimed at maximizing the log probability offsets between the tuned and untuned small models while sampling from a fixed large model [130]. Such weak-to-strong search strategy has been applied to various LLM applications, such as safety alignment [84], jailbreak attacks [128], and unlearning [43].

Multi-objective Decoding Considering the different alignment objectives across scenarios and users, there is a need for on-the-fly adaptation of language models to cater to various objective combinations. This raises a question: Given a set of policies corresponding to different reward models, can we find a way that does not require fine-tuning LLMs to correspond to the multi-policies, which would be time-consuming and difficult? Multi-Objective Decoding (MOD) [84] introduces a method for combining prediction distributions based on multiple reward models, each trained for an individual objective. This approach identifies a closed-form solution among f-divergence regularized alignment methods like PPO and DPO through Legendre transform [75]. This method allows for the combination of any reward models at inference time, eliminating the need for retraining reward models.

Weak-to-strong Jailbreak Attack The weak-to-strong search can also conduct jailbreak attacks on LLMs [128], revealing that most jailbreak tokens surface within the first ten tokens. The logits from strong, safe models, plus the unsafe logit offset between weak safe models and unsafe ones, can produce a jailbreak token from the strong model. This suggests that even secure LLMs can be misled by unsafe, weak SLMs into generating undesired outputs with targeted guidance. This method is computationally efficient, avoiding the need for extensive computations to find optimal decoding parameters or optimize prompts, and it can produce more harmful content than smaller attack models alone.

Weak-to-strong Unlearning The weak-to-strong decoding can also be applied to LLM unlearning issues. LLMs may cause privacy leaks and copyright infringement due to their memory of training data. Existing LLM unlearning methods require retraining models from scratch [126], which is impractical for black-box models. Hence, δ-Unlearning [43] is proposed, an offset unlearning framework for black-box LLMs that learns the required logit offsets by comparing the logits of a small, white-box model with the original logits without fine-tuning the black-box LLM.

3.4 SLMs Guard LLMs

As demonstrated in recent studies [33, 62, 78, 81, 119], LLMs exhibit significant vulnerabilities to adversarial attacks and jailbreaking attempts. For instance, Wang et al. [98] illustrate that ChatGPT underperforms when evaluated on adversarial datasets, highlighting ongoing risks associated with adversarial vulnerabilities. This reinforces the need for robust guardrails in generative AI systems. Beyond developing inherently trustworthy LLMs, the adoption of SLMs for reinforcing LLM safety [45, 48] and hallucination detection [92, 109] has gained considerable attention.

SLMs as Guardian. To address safety concerns of LLMs, several works [45, 48, 78, 99] have been conducted to use SLMs to help improve the safety of LLMs. For example, Llama Guard [45], fine-tuned on Llama2-7B, provides a publicly available tool designed explicitly for identifying safety risks in conversational prompts and responses. However, this tool primarily assesses the harmfulness of questions and answers without enabling the generation of fluent, safeguarded responses. To address this shortcoming, Kwon et al. [48] propose a specialized SLM capable of concurrently detecting harmful queries and generating safeguard-oriented, explanatory responses, thus significantly enhancing conversational AI safety. Additionally, Sawtell et al. [78] illustrates that SLMs can serve as robust feature extractors, effectively training simple classifiers with fewer than 100 high-quality examples. These classifiers support tasks including content safety classification, prompt injection detection, and simultaneous token generation for improved safety. Further advancements include STAND-Guard [99], which fine-tunes SLMs to monitor the safety of generated content, especially addressing out-of-distribution scenarios. Similarly, TorchOpera [34] integrates multiple SLMs serving as safety detectors and repair modules, enhancing prompt safety and response quality through enriched context and defined corrective guidelines.

SLMs as Hallucination Detectors. To address the challenge of hallucination detection in LLMs, recent works [11, 92, 109, 127] have focused on calibrating model confidence through auxiliary SLMs. These auxiliary models analyze both the original questions and the corresponding answers produced by LLMs to generate reliable confidence estimates. The calibration training minimizes discrepancies between predicted confidence scores and actual calibration errors. For example, APRICOT [92] leverages an auxiliary DeBERTaV3 model [35] to assess the confidence of LLM responses to improve the expression of uncertainty and the precision of response adjustments. Similarly, POLAR [127] proposes a self-supervised calibration technique to align LLM outputs with various weak supervision signals, which refines model confidence and reducing potential inaccuracies. Moreover, SLMs have also been employed to evaluate the internal states of LLMs, predicting the probability of hallucinated content. For example, Xu et al. [109] introduce a lightweight detector to analyze token-level contributions to hallucinations. SAPLMA [11] demonstrates that internal states within LLMs carry valuable signals about the truthfulness of their generated statements and achieves impressive classification accuracies ranging from 71% to 83%. Additionally, recent approaches [33, 40] have integrated SLMs into unified frameworks designed to enhance real-time hallucination detection. For instance, Hu et al. [42] propose using an SLM classifier for initial hallucination detection. Furthermore, Han et al. [33] employs customized SLMs to identify unsafe or hallucinated content for LLMs. Collectively, these approaches demonstrate the critical role of SLMs in fortifying LLM safety, emphasizing proactive risk mitigation and enhancing overall AI safety and user trust.

4 Trustworthiness in SLMs

Despite their widespread use, small language models pose risks in adversarial robustness, toxicity, privacy, and fairness. For a broader overview, we refer readers to see Wang et al. [96]. Below, we highlight recent advances in these areas for small language models.

Adversarial Robustness. Adversarial robustness in LMs refers to a model's ability to resist malicious inputs designed to manipulate its behavior or degrade its performance [63]. Recent studies have demonstrated that small language models are particularly vulnerable to various forms of adversarial attacks. These include malicious in-context demonstrations [72], adversarial word replacement [114], and adversarial suffix tokens [39]. However, there is currently no consensus on the relationship between model size and adversarial robustness. For instance, Yang et al. [114] observed that larger models exhibit improved robustness under adversarial word replacement, whereas other studies [39, 72] report different trends across different attack types. To tackle the adversarial attacks, several adversarial defense techniques have been proposed [47, 63, 95, 106, 118]. Yu et al. [118] introduced an adversarial training framework to enhance model robustness, and Xhonneux et al. [106] extended this work to defend against continuous adversarial perturbations. Furthermore, certifiably robust LMs have been proposed [47, 95], offering formal guarantees against specific classes of adversarial prompts. Howe et al. [39] found adversarial training more effective on larger models, implying greater training effort should be allocated to smaller models. These findings underscore **the need for increased defensive computing and thorough adversarial robustness evaluation in small language models to ensure their reliability in real-world applications.**

Toxicity and Refusal Behaviors. Toxicity in LMs refers to the generation of harmful, offensive, or inappropriate content that may cause harm to individuals or groups [97]. Modern language models are expected not only to avoid generating toxic outputs but also to actively refuse to respond to harmful or unsafe prompts.

Recent studies systematically benchmark toxicity and refusal behaviors in language models ranging from SLMs to LLMs [18, 24, 74, 101, 107]. A consistent finding across these studies is that *model size does not correlate strongly with safety performance in toxicity mitigation.* For instance, LLaMA-2 7B exhibits the highest refusal rate on harmful prompts in the Do-Not-Answer dataset [101], while in OR-Bench [18], LLaMA-3 8B demonstrates safer behavior than both LLaMA-3 70B and GPT-3.5-turbo-0125. Notably, Cui et al. [18] emphasize a trade-off between helpfulness and response safety. Similarly, SORRY-Bench [107] highlights that refusal performance varies significantly across models and fine-grained categories. A concerning trend emerges with on-device SLMs, particularly those around 3B parameters, such as Gemma-2B, and especially after aggressive quantization (e.g., 4-bit). These models often generate toxic, hateful, or illegal outputs without requiring jailbreak techniques. This vulnerability may result from quantization masking or removing safety layers, or from the lack of explicit refusal mechanisms in lightweight SLMs. These observations further reinforce that model size alone is not a reliable predictor of toxicity or safety in refusal behavior.

Several mitigation strategies address toxicity across different stages of LM development. During pretraining and fine-tuning, filtering toxic data [46, 65] and training on curated, safe datasets using techniques such as reinforcement learning from human feedback (RLHF) [105] show promise. At inference time, methods like contrastive prompting help steer generation away from harmful content [51]. In the post-processing phase, auditing tools and classifiers detect and filter toxic outputs before deployment [103]. Given the increasing deployment of SLMs in resource-constrained settings, future research should prioritize **toxicity mitigation techniques tailored to on-device models, especially under quantization**.

Jailbreak attacks, in adversarial settings, aim to craft sophisticated prompts that bypass a model's safety mechanisms, thereby coercing the model into responding to malicious queries and generating harmful or toxic content. Small language models often prioritize helpfulness over harmlessness, making them particularly susceptible to such attacks. Zhang et al. [125] conduct a comprehensive evaluation of SLMs under jailbreak attacks, assessing 63 models across 8 distinct attack strategies. Their findings reveal that approximately half of the evaluated SLMs are highly vulnerable to jailbreak prompts. Importantly, the study shows that model size has little correlation with jailbreak vulnerability, while training techniques play a critical role in determining the security posture of SLMs. These observations underscore **the importance of incorporating robust safety mechanisms during the design and training phases of SLM development**.

Privacy. Privacy-preserving capabilities of SLMs are essential, as even basic interactions can severely disseminate personally identifiable information (PII) [52, 53], potentially violating major privacy laws like the EU's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). The existing research on SLM privacy primarily focuses on (1) benchmarking SLM privacy issues [53] and (2) privacy-preserving strategies for SLM generation [23, 44, 53, 117]. Regarding evaluating SLMs' privacy, Li et al. [53] proposes a privacy evaluation benchmark, PrivLM-Bench, to assess the privacy issues in LMs. Three levels of attacks are implemented, including data extraction attacks [17], membership inference attacks [27], and embedding-level privacy attacks [76]. Their experimental results suggest that LMs have limited privacy-preserving capabilities. Regarding privacy-preserving strategies, research on privacy-preserving LMs (PPLMs) mainly focuses on differential privacy (DP) mechanisms [44, 53, 117]. Duan et al. explored DP prompt tuning by adding noise to soft prompts for enhanced privacy. This remains an active area of research, particularly as more organizations fine-tune small models on proprietary data, raising open questions about how to prevent unintended data exposure. **A promising direction is to integrate differential privacy into**

parameter-efficient tuning (e.g., LoRA), develop quantization-aware privacy controls, and build auditing tools to detect memorization or leakage in on-device settings.

Fairness, Bias, and Stereotype. Fairness in language models refers to the absence of unjust or prejudiced behavior toward specific groups (e.g., based on gender, race, etc.), and more broadly to the avoidance of harmful stereotypes or biased outputs [15]. Small language models (SLMs), like their larger counterparts, inherit biases from their training data. Cui et al. [19] introduce a benchmark focusing on identity, credit, criminality, and health-related questions to assess fairness, and find that 7B SLMs perform significantly worse in these aspects. Nakka et al. [74] study both on-server and on-device SLMs and show that quantized on-device models exhibit higher risks of stereotypical bias and unfair behaviors. AdvCoU [72] reveals that adversarial in-context demonstrations achieve 100% attack success rates on stereotype and sycophancy dimensions across both small and large models. Similar to other trustworthiness dimensions, there is no consistent relationship between model size and fairness vulnerabilities [24]. Some initial efforts aim to improve fairness in SLMs [26]. For example, Fayyazi et al. [26] propose a fairness-aware framework, FACTER, designed for LM-based recommendation systems. FACTER employs an adaptive semantic variance threshold and a violation-triggered tightening mechanism to automatically enhance fairness constraints when biased patterns are detected. It achieves up to a 95.5% reduction in fairness violations on SLMs such as Mistral-7B and LLaMA-2-7B. ROBBIE [24] demonstrates that the effectiveness of debiasing methods varies by model size: self-debiasing [79] is more effective on smaller models, while prompting methods are more effective on larger models. **A key open challenge is ensuring fairness in multilingual SLMs, where biases specific to both cultural contexts and low-resource languages are prevalent yet remain underexplored.**

5 OPEN CHALLENGES AND FUTURE DIRECTIONS

In this section, we discuss the open challenges and several corresponding promising directions to further advance SLM studies with technologies that are already applied in LLMs.

- **Retrieval Augmented Generation for SLMs** Recent advances in Retrieval-Augmented Generation enhance large language model integration of external knowledge, but do not transfer well to small language models, which struggle with complex queries and multi-step reasoning. A specialized RAG paradigm using graph structures offers a promising solution for SLMs by leveraging graph connections and hierarchical representations to reduce cognitive load and enhance reasoning. This approach necessitates developing lightweight graph-based retrieval algorithms and hybrid data systems that integrate text with graph structures.
- **Multi-agent with SLM Collaboration** The SLM collaboration-based multi-agent system is redefining development with its suitability for distributed deployment and improved performance in adaptive collaboration networks. Unlike a single large LLM that demands significant resources, a network of smaller SLMs offers dynamic calling for computational efficiency. This decentralized approach leverages expert models to enhance task execution, with systems like eight SLMs outperforming a large LLM and providing efficient advanced AI capabilities for edge devices.
- **Advancing Trustworthy SLMs**: Ensuring the trustworthiness of SLMs remains a critical challenge requiring deeper investigation. Key areas of focus include mitigating the generation of toxic content and misinformation, both of which are currently underexplored in SLMs. Additionally, no effective methods have been developed to address sycophancy. Another pressing concern is the design of fairness-aware SLMs that function effectively across diverse domains while minimizing biases, thereby promoting ethical and responsible AI deployment.

6 Conclusion

In this survey, we have reviewed recent advancements in small language models in the era of large language models. We begin by examining architectures tailored for SLMs, including Transformer design, Mamba, and xLSTM. We then explore weak-to-strong methods such as test-time scaling, which enhance SLMs by enabling them to surpass LLMs and assist in the fine-tuning, decoding, and safeguarding of LLMs. Additionally, we address the critical issue of trustworthiness in SLMs, a significant concern in contemporary language models. This survey concludes by offering key insights that will inform and guide future research on small language models.

Acknowledgments

This material is based upon work supported by, or in part by the Army Research Office (ARO) under grant number W911NF-21-10198, the Department of Homeland Security (DHS) under grant number 17STCIN00001-05-00, and Cisco Faculty Research Award.

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219* (2024).

[2] Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. 2025. Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs. *arXiv preprint arXiv:2503.01743* (2025).

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[4] Abien Fred Agarap. 2018. Deep Learning using Rectified Linear Units (ReLU). *CoRR* abs/1803.08375 (2018). arXiv:1803.08375 http://arxiv.org/abs/1803.08375

[5] Meta AI. 2024. *Llama 3.2: Revolutionizing edge AI and vision with open, customizable models.* https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/ Accessed: 2024-9-25.

[6] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. arXiv:2305.13245 [cs.CL] https://arxiv.org/abs/2305.13245

[7] Ali Al-Lawati, Jason Lucas, Zhiwei Zhang, Prasenjit Mitra, and Suhang Wang. 2025. Graph-based Molecular In-context Learning Grounded on Morgan Fingerprints. arXiv:2502.05414 [cs.LG] https://arxiv.org/abs/2502.05414

[8] Musleh Alharthi and Ausif Mahmood. 2024. xlstmtime: Long-term time series forecasting with xlstm. *AI* 5, 3 (2024), 1482–1495.

[9] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. 2025. SmolLM2: When Smol Goes Big–Data-Centric Training of a Small Language Model. *arXiv preprint arXiv:2502.02737* (2025).

[10] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. SmolLM - blazingly fast and remarkably powerful.

[11] Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 967–976.

[12] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. arXiv:2309.16609 [cs.CL] https://arxiv.org/abs/2309.16609

[13] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael K Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2024. xLSTM: Extended Long Short-Term Memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=ARAxPPIAhq

[14] Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. 2024. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834* (2024).

[15] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (Technology) is Power: A Critical Survey of "Bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5454–5476.

[16] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2024. Weak-to-strong generalization: eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning*. 4971–5012.

[17] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[18] Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. OR-Bench: An Over-Refusal Benchmark for Large Language Models. *arXiv preprint arXiv:2405.20947* (2024).

[19] Shiyao Cui, Zhenyu Zhang, Yilong Chen, Wenyuan Zhang, Tianyun Liu, Siqi Wang, and Tingwen Liu. 2023. Fft: Towards harmlessness evaluation and analysis for llms with factuality, fairness, toxicity. *arXiv preprint arXiv:2311.18580* (2023).

[20] Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060* (2024).

[21] Ricardo Dominguez-Olmedo, Moritz Hardt, and Celestine Mendler-Dünner. 2023. Questioning the survey responses of large language models. *arXiv preprint arXiv:2306.07951* (2023).

[22] Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, ZIJIA CHEN, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. 2025. Hymba: A Hybrid-head Architecture for Small Language Models. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=A1ztozypga

[23] Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. 2023. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems* 36 (2023), 76852–76871.

[24] David Esiobu, Xiaoqing Tan, Saghar Hosseini, Megan Ung, Yuchen Zhang, Jude Fernandes, Jane Dwivedi-Yu, Eleonora Presani, Adina Williams, and Eric Smith. 2023. ROBBIE: Robust bias evaluation of large generative language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 3764–3814. https://aclanthology.org/2023.emnlp-main.230

[25] Xiaojing Fan, Chunliang Tao, and Jianyu Zhao. 2024. Advanced stock price prediction with xlstm-based models: Improving long-term forecasting. In *2024 11th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE, 117–123.

[26] Arya Fayyazi, Mehdi Kamal, and Massoud Pedram. 2025. FACTER: Fairness-Aware Conformal Thresholding and Prompt Engineering for Enabling Fair LLM-Based Recommender Systems. *arXiv preprint arXiv:2502.02966* (2025).

[27] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2023. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *arXiv preprint arXiv:2311.06062* (2023).

[28] Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712* (2024).

[29] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838* (2024).

[30] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[31] Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*. https://openreview.net/forum?id=uYLFoz1vlAC

[32] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[33] Shanshan Han, Salman Avestimehr, and Chaoyang He. 2025. Bridging the Safety Gap: A Guardrail Pipeline for Trustworthy LLM Inferences. *arXiv preprint arXiv:2502.08142* (2025).

[34] Shanshan Han, Zijian Hu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Dimitris Stripelis, Zhaozhuo Xu, and Chaoyang He. 2024. TorchOpera: A Compound AI System for LLM Safety. *arXiv preprint arXiv:2406.10847* (2024).

[35] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=sE7-XhLxHA

[36] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[37] Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM can solve hard long time lag problems. *Advances in neural information processing systems* 9 (1996).

[38] Junyuan Hong, Jinhao Duan, Chenhui Zhang, Zhangheng Li, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian R. Bartoldson, Ajay Kumar Jaiswal, Kaidi Xu, Bhavya Kailkhura, Dan Hendrycks, Dawn Song, Zhangyang Wang, and Bo Li. 2024. Decoding Compressed Trust: Scrutinizing the Trustworthiness of Efficient LLMs Under Compression. In *Proceedings of the Forty-first International Conference on Machine Learning, ICML*. https://openreview.net/forum?id=e3Dpq3WdMv

[39] Nikolaus Howe, Ian McKenzie, Oskar Hollinsworth, Michał Zajac, Tom Tseng, Aaron Tucker, Pierre-Luc Bacon, and Adam Gleave. 2024. Effects of scale on language model robustness. *arXiv preprint arXiv:2407.18213* (2024).

[40] Mengya Hu, Rui Xu, Deren Lei, Yaxi Li, Mingyu Wang, Emily Ching, Eslam Kamal, and Alex Deng. 2024. SLM Meets LLM: Balancing Latency, Interpretability and Consistency in Hallucination Detection. *arXiv preprint arXiv:2408.12748* (2024).

[41] Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies. In *First Conference on Language Modeling*. https://openreview.net/forum?id=3X2L2TFr0f

[42] Xing Hu, Yuan Chen, Dawei Yang, Sifan Zhou, Zhihang Yuan, Jiangyong Yu, and Chen Xu. 2024. I-LLM: Efficient Integer-Only Inference for Fully-Quantized Low-Bit Large Language Models. *arXiv preprint arXiv:2405.17849* (2024).

[43] James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. Offset unlearning for large language models. *arXiv preprint arXiv:2404.11045* (2024).

[44] Timour Igamberdiev and Ivan Habernal. 2023. DP-BART for privatized text rewriting under local differential privacy. *arXiv preprint arXiv:2302.07636* (2023).

[45] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674* (2023).

[46] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*. PMLR, 17506–17533.

[47] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. 2024. Certifying LLM Safety against Adversarial Prompting. In *First Conference on Language Modeling*. https://openreview.net/forum?id=9Ik05cycLq

[48] Ohjoon Kwon, Donghyeon Jeon, Nayoung Choi, Gyu-Hwung Cho, Hwiyeol Jo, Changbong Kim, Hyunwoo Lee, Inho Kang, Sun Kim, and Taiwoo Park. 2024. SLM as Guardian: Pioneering AI Safety with Small Language Model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (Eds.). Association for Computational Linguistics, Miami, Florida, US, 1333–1350. doi:10.18653/v1/2024.emnlp-industry.99

[49] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *ArXiv e-prints* (2016), arXiv–1607.

[50] Barak Lenz, Opher Lieber, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, Daniel Gissin, Daniel Jannai, Dor Muhlgay, Dor Zimberg, Edden M. Gerber, Elad Dolev, Eran Krakovsky, Erez Safahi, Erez Schwartz, Gal Cohen, Gal Shachaf, Haim Rozenblum, Hofit Bata, Ido Blass, Inbal Magar, Itay Dalmedigos, Jhonathan Osin,

Julie Fadlon, Maria Rozman, Matan Danos, Michael Gokhman, Mor Zusman, Naama Gidron, Nir Ratner, Noam Gat, Noam Rozen, Oded Fried, Ohad Leshno, Omer Antverg, Omri Abend, Or Dagan, Orit Cohavi, Raz Alon, Ro'i Belson, Roi Cohen, Rom Gilad, Roman Glozman, Shahar Lev, Shai Shalev-Shwartz, Shaked Haim Meirom, Tal Delbari, Tal Ness, Tomer Asida, Tom Ben Gal, Tom Braude, Uriya Pumerantz, Josh Cohen, Yonatan Belinkov, Yuval Globerson, Yuval Peleg Levy, and Yoav Shoham. 2025. Jamba: Hybrid Transformer-Mamba Language Models. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=JFPaD7lpBD

[51] Chak Tou Leong, Yi Cheng, Jiashuo Wang, Jian Wang, and Wenjie Li. 2023. Self-Detoxifying Language Models via Toxification Reversal. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 4433–4449.

[52] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. 2023. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197* (2023).

[53] Haoran Li, Dadi Guo, Donghao Li, Wei Fan, Qi Hu, Xin Liu, Chunkit Chan, Duanyi Yao, Yuan Yao, and Yangqiu Song. 2024. PrivLM-Bench: A Multi-level Privacy Evaluation Benchmark for Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. 54–73.

[54] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems* 37 (2024), 14200–14282.

[55] Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024. Superfiltering: Weak-to-Strong Data Filtering for Fast Instruction-Tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 14255–14273.

[56] Tianlin Li, Qian Liu, Tianyu Pang, Chao Du, Qing Guo, Yang Liu, and Min Lin. 2024. Purifying large language models by ensembling a small language model. *arXiv preprint arXiv:2402.14845* (2024).

[57] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5315–5333.

[58] Minhua Lin, Hui Liu, Xianfeng Tang, Jingying Zeng, Zhenwei Dai, Chen Luo, Zheng Li, Xiang Zhang, Qi He, and Suhang Wang. 2025. How Far are LLMs from Real Search? A Comprehensive Study on Efficiency, Completeness, and Inherent Capabilities. *arXiv preprint arXiv:2502.18387* (2025).

[59] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434* (2024).

[60] Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565* (2024).

[61] Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling. *arXiv preprint arXiv:2502.06703* (2025).

[62] Zhengxiao Liu, Bowen Shen, Zheng Lin, Fali Wang, and Weiping Wang. 2023. Maximum Entropy Loss, the Silver Bullet Targeting Backdoor Attacks in Pre-trained Language Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 3850–3868. doi:10.18653/v1/2023.findings-acl.237

[63] Zhengxiao Liu, Fali Wang, Zheng Lin, Lei Wang, and Zhiyi Yin. 2020. DE-CO: A Two-Step Spelling Correction Model for Combating Adversarial Typos. In *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. 554–561. doi:10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00095

[64] Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *arXiv preprint arXiv:2402.14905* (2024).

[65] Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 3245–3276.

[66] Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790* (2024).

[67] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics* 23, 6 (2022), bbac409.

[68] Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, et al. 2023. Paloma: A benchmark for evaluating language model fit. *arXiv preprint arXiv:2312.10523* (2023).

[69] Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Seyed Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, et al. 2024. Openelm: An efficient language model family with open training and inference framework. In *Workshop on Efficient Systems for Foundation Models II@ ICML2024*.

[70] Dheeraj Mekala, Alex Nguyen, and Jingbo Shang. 2024. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *arXiv preprint arXiv:2402.10430* (2024).

[71] Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2024. An Emulator for Fine-tuning Large Language Models using Small Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=Eo7kv0sllr

[72] Lingbo Mo, Boshi Wang, Muhao Chen, and Huan Sun. 2024. How Trustworthy are Open-Source LLMs? An Assessment under Malicious Demonstrations Shows their Vulnerabilities. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2775–2792.

[73] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *Advances in Neural Information Processing Systems* 37 (2024), 41076–41102.

[74] Kalyan Nakka, Jimmy Dani, and Nitesh Saxena. 2024. Is On-Device AI Broken and Exploitable? Assessing the Trust and Ethics in Small Language Models. *arXiv preprint arXiv:2406.05364* (2024).

[75] Yurii Nesterov et al. 2018. *Lectures on convex optimization*. Vol. 137. Springer.

[76] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1314–1331.

[77] Pascal Pfeiffer, Philipp Singer, Yauhen Babakhin, Gabor Fodor, Nischay Dhankhar, and Sri Satish Ambati. 2024. H2O-Danube3 Technical Report. *arXiv preprint arXiv:2407.09276* (2024).

[78] Mason Sawtell, Tula Masterman, Sandi Besen, and Jim Brown. 2024. Lightweight safety classification using pruned language models. *arXiv preprint arXiv:2412.13435* (2024).

[79] Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. *Transactions of the Association for Computational Linguistics* 9 (2021), 1408–1424.

[80] Niklas Schmidinger, Lisa Schneckenreiter, Philipp Seidl, Johannes Schimunek, Sohvi Luukkonen, Pieter-Jan Hoedt, Johannes Brandstetter, Andreas Mayr, Sepp Hochreiter, and Günter Klambauer. 2024. Bio-xLSTM: Generative modeling, representation and in-context learning of biological and chemical sequences. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*. https://openreview.net/forum?id=rn0JKIvABP

[81] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. 2023. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844* (2023).

[82] Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150* (2019).

[83] Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202* (2020).

[84] Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A Smith, and Simon S Du. 2024. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems* 37 (2024), 48875–48920.

[85] Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=4FWAwZtd2n

[86] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 9275–9293.

[87] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295* (2024).

[88] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).

[89] TensorOpera Team. 2024. *TensorOpera Unveils Fox Foundation Model: A Pioneering Small Language Model (SLM) for Cloud and Edge.*

https://blog.tensoropera.ai/tensoropera-unveils-fox-foundation-model-a-pioneering-open-source-slm-leading-the-way-against-tech-giants/ Accessed: 2024-6-13.

[90] Omkar Thawakar, Ashmal Vayani, Salman Khan, Hisham Cholakal, Rao M Anwer, Michael Felsberg, Tim Baldwin, Eric P Xing, and Fahad Shahbaz Khan. 2024. Mobillama: Towards accurate and lightweight fully transparent gpt. *arXiv preprint arXiv:2402.16840* (2024).

[91] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[92] Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. 2024. Calibrating Large Language Models Using Their Generations Only. *arXiv preprint arXiv:2403.05973* (2024).

[93] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[94] Fali Wang, Runxue Bao, Suhang Wang, Wenchao Yu, Yanchi Liu, Wei Cheng, and Haifeng Chen. 2024. InfuserKI: Enhancing Large Language Models with Knowledge Graphs via Infuser-Guided Knowledge Integration. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 3675–3688.

[95] Fali Wang, Zheng Lin, Zhengxiao Liu, Mingyu Zheng, Lei Wang, and Daren Zha. 2021. Macrobert: Maximizing certified region of bert to adversarial word substitutions. In *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part II 26*. Springer, 253–261.

[96] Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, et al. 2024. A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness. *arXiv preprint arXiv:2411.03350* (2024).

[97] Huandong Wang, Wenjie Fu, Yingzhou Tang, Zhilong Chen, Yuxi Huang, Jinghua Piao, Chen Gao, Fengli Xu, Tao Jiang, and Yong Li. 2025. A Survey on Responsible LLMs: Inherent Risk, Malicious Use, and Mitigation Strategy. *arXiv preprint arXiv:2501.09431* (2025).

[98] Jindong Wang, HU Xixu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, et al. [n. d.]. On the Robustness of ChatGPT: An Adversarial and Out-of-distribution Perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

[99] Minjia Wang, Pingping Lin, Siqi Cai, Shengnan An, Shengjie Ma, Zeqi Lin, Congrui Huang, and Bixiong Xu. 2024. STAND-Guard: A Small Task-Adaptive Content Moderation Model. *arXiv preprint arXiv:2411.05214* (2024).

[100] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=1PL1NIMMrw

[101] Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024. Do-Not-Answer: Evaluating Safeguards in LLMs. In *Findings of the Association for Computational Linguistics: EACL 2024*. Association for Computational Linguistics, 896–911. https://aclanthology.org/2024.findings-eacl.61

[102] Zhepeng Wang, Runxue Bao, Yawen Wu, Guodong Liu, Lei Yang, Liang Zhan, Feng Zheng, Weiwen Jiang, and Yanfu Zhang. 2024. Self-guided Knowledge-Injected Graph Neural Network for Alzheimer’s Diseases. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 378–388.

[103] Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the Implicit Toxicity in Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 1322–1338.

[104] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Scaling Inference Computation: Compute-Optimal Inference for Problem-Solving with Language Models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS’24*. https://openreview.net/forum?id=j7DZWSc8qu

[105] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems* 36 (2023), 59008–59033.

[106] Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. 2024. Efficient adversarial training in llms with continuous attacks. *arXiv preprint arXiv:2405.15589* (2024).

[107] Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, et al. 2025. SORRY-bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*.

[108] Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Self-Evaluation Guided Beam Search for Reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=Bw82hwg5Q3

[109] Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna Martindale, and Marine Carpuat. 2023. Understanding and Detecting Hallucinations in Neural Machine Translation via Model Introspection. *Transactions of the Association for Computational Linguistics* 11 (2023), 546–564.

[110] Sarthak Yadav, Sergios Theodoridis, and Zheng-Hua Tan. 2024. Audio xlstms: Learning self-supervised audio representations with xlstms. *arXiv preprint arXiv:2408.16568* (2024).

[111] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024).

[112] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

[113] Yuqing Yang, Yan Ma, and Pengfei Liu. 2024. Weak-to-Strong Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 8350–8367.

[114] Zeyu Yang, Zhao Meng, Xiaochen Zheng, and Roger Wattenhofer. 2024. Assessing adversarial robustness of large language models: An empirical study. *arXiv preprint arXiv:2405.02764* (2024).

[115] Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. 2025. LongMamba: Enhancing Mamba’s Long-Context Capabilities via Training-Free Receptive Field Enlargement. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=fMbLszVO1H

[116] Rongjie Yi, Xiang Li, Weikai Xie, Zhenyan Lu, Chenghua Wang, Ao Zhou, Shangguang Wang, Xiwen Zhang, and Mengwei Xu. 2024. PhoneLM: an Efficient and Capable Small Language Model Family through Principled Pre-training. *arXiv preprint arXiv:2411.05046* (2024).

[117] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500* (2021).

[118] Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. 2024. Robust LLM safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089* (2024).

[119] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. 2024. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=MbfAK4s61A

[120] Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems* 32 (2019).

[121] He Zhang, Jingyi Xie, Chuhao Wu, Jie Cai, ChanMin Kim, and John M Carroll. 2024. The future of learning: Large language models through the lens of students. In *Proceedings of the 25th Annual Conference on Information Technology Education*. 12–18.

[122] Jun Zhang, Jue WANG, Huan Li, Lidan Shou, Ke Chen, Yang You, Guiming Xie, Xuejian Gong, and Kunlong Zhou. 2025. Train Small, Infer Large: Memory-Efficient LoRA Training for Large Language Models. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=s7DkcgpRxL

[123] Shaokun Zhang, Xiaobo Xia, Zhaoqing Wang, Ling-Hao Chen, Jiale Liu, Qingyun Wu, and Tongliang Liu. 2024. IDEAL: Influence-Driven Selective Annotations Empower In-Context Learners in Large Language Models. In *ICLR*.

[124] Shaokun Zhang, Jieyu Zhang, Jiale Liu, Linxin Song, Chi Wang, Ranjay Krishna, and Qingyun Wu. 2024. Training language model agents without modifying language models. *arXiv e-prints* (2024), arXiv–2402.

[125] Wenhui Zhang, Huiyu Xu, Zhibo Wang, Zeqing He, Ziqi Zhu, and Kui Ren. 2025. Can Small Language Models Reliably Resist Jailbreak Attacks? A Comprehensive Evaluation. *arXiv preprint arXiv:2503.06519* (2025).

[126] Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2025. Catastrophic Failure of LLM Unlearning via Quantization. arXiv:2410.16454 [cs.CL] https://arxiv.org/abs/2410.16454

[127] Theodore Zhao, Mu Wei, J Samuel Preston, and Hoifung Poon. 2023. Automatic calibration and error correction for large language models via pareto optimal self-supervision. *arXiv preprint arXiv:2306.16564* (2023).

[128] Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. 2024. Weak-to-Strong Jailbreaking on Large Language Models. In *ICML 2024 Next Generation of AI Safety Workshop*. https://openreview.net/forum?id=shrX5xIHCW

[129] Yucheng Zhou, Jianbing Shen, and Yu Cheng. 2024. Weak to strong generalization for large language models with multi-capabilities. In *The Thirteenth International Conference on Learning Representations*.

[130] Zhanhui Zhou, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. 2024. Weak-to-Strong Search: Align Large Language Models via Searching over Small Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=dOJ6CqWDf1