Token–Context Attention for NLI: An Alternative to Self-Attention

Xin Zhang, Victor S. Sheng

Department of Computer Science, Texas Tech University
Xin.Zhang@ttu.edu, Victor.Sheng@ttu.edu

Abstract

Despite the rapid progress in large language models (LLMs), even sub-billion-scale systems perform at chance level on challenging natural language inference (NLI) benchmarks such as Adversarial Natural Language Inference (ANLI), while training larger models is often impractical due to limited computational resources. We address this parameter-efficiency bottleneck in NLI with a *Complex–Vector Token Representation* that explicitly decouples each token from its context, and a *Token–Context Attention* mechanism that updates each token based on the most informative contextual semantics. On ANLI, a 0.8B-parameter Token–Context Attention model achieves higher parameter efficiency (accuracy per parameter) than all 1B and comparable 0.8B self-attention baselines; it also suffers smaller performance degradation under Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attacks and achieves the largest few-shot gains on SNLI and MNLI while exhibiting no significant degradation in ANLI accuracy after adaptation. These results suggest that explicitly disentangling token and context offers a viable alternative to standard self-attention for NLI tasks.

Introduction

The Adversarial Natural Language Inference (ANLI) benchmark is designed to target the semantic weaknesses of natural language models by iteratively collecting human-written adversarial examples (Nie et al. 2020). Pretrained Transformer models smaller than GPT-3 ($\approx$ 175B parameters) typically perform at chance level on ANLI in the zero-shot setting, and even GPT-3 itself does not substantially exceed chance without task-specific fine-tuning or supervised training (Brown et al. 2020). Moreover, simply increasing model size does not guarantee improved performance on ANLI-like tasks (Wang et al. 2023). Meanwhile, large-scale pretraining is prohibitively expensive (Yao et al. 2022). This raises a natural question: how can we improve the parameter efficiency (accuracy per parameter; APP; see Canziani, Paszke, and Culurciello 2016) of Transformer-based NLI models under limited computational resources?

To answer this question, we propose a Complex-Vector Token Representation whose magnitude encodes global semantics and its phase captures each token's alignment to that global meaning (in polar form). After converting to Cartesian coordinates, the real and imaginary parts correspond to the token itself and the L2-norm–based context vector (from all other tokens), respectively, each regulated by a transformation-induced cosine-similarity gate. Building on this, we compute token–context attention weights and update each token with multiple high-weight contexts.

On the ANLI benchmark, our 0.8B-parameter *Token-Context Attention* (T-C Attn) achieves higher APP than all 1B- and comparable 0.8B-scale self-attention baselines. When transferring *Token-Context Attention*'s trained checkpoints to SNLI (Bowman et al. 2015) and MNLI (Williams, Nangia, and Bowman 2018), zero-shot accuracy shows only modest drops, while fine-tuning on just ten samples yields the largest few-shot gains among all models. Finally, under FGSM (Goodfellow, Shlens, and Szegedy 2015) and PGD (Madry et al. 2018) embedding perturbations, its accuracy declines less than that of any self-attention baseline, suggesting a promising trend toward improved stability.

Background

Benchmarks

Madaan et al. (2025) show that ANLI remains unsaturated, discriminates between similarly sized models, and yields consistent gains with training and scaling, making it a strong benchmark for LLM evaluation.

Architecture-Level Approaches

(1) Transformer Architecture Improvements

Several Transformer variants reweight attention toward more reliable signals: Adversarial Self-Attention (ASA) applies binary masks during training (Wu et al. 2023a); Dynamic Attention adjusts logits at test time via masking and resampling (Shen et al. 2024); Gaussian-biased attention adds distance-aware priors (Guo, Zhang, and Liu 2019); context vectors are fused into the query–key projection (Gururangan et al. 2018); and Multi-Token Attention expands each query's receptive field (Golovneva et al. 2025). However, these designs still entangle global and local semantics, making models vulnerable to spurious lexical cues.

(2)Alternative Architectures

Recent architectures pursue *linear-time* or *Fourier-style* token mixing, e.g., RetNet's gated retention (Sun

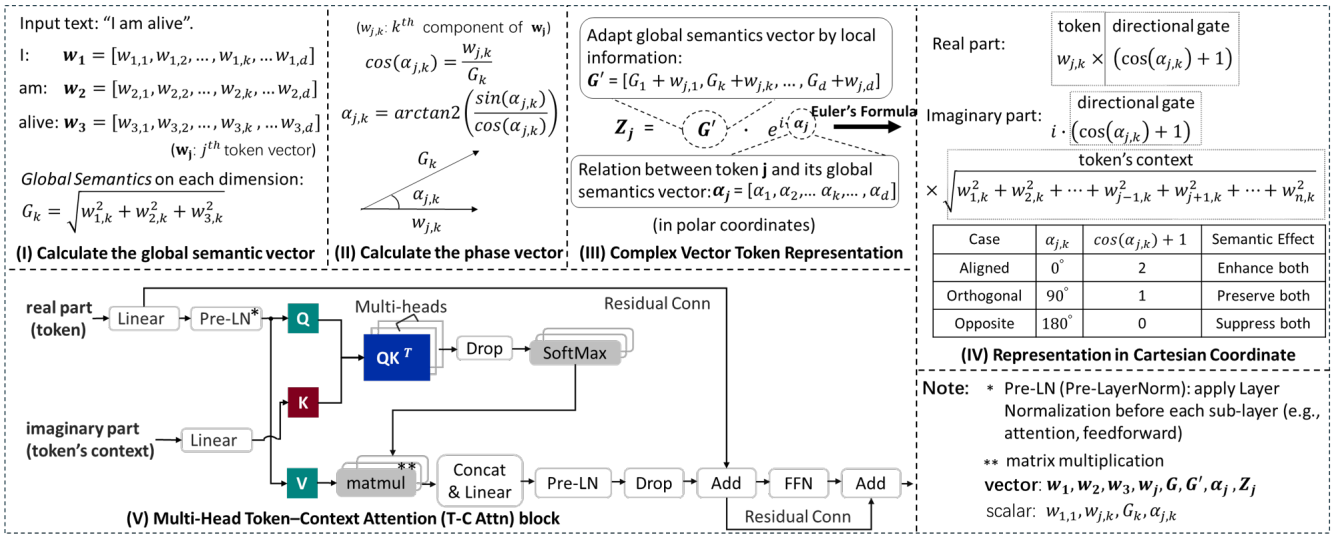

Figure 1: A Token–Context architecture based on Complex–Vector Token Representations

et al. 2023), the selective state-space (SSM) modeling of Mamba (Gu and Dao 2023), data-dependent Gated Linear Attention (Yang et al. 2024), and FFT-based mixers such as FNet (Lee-Thorp et al. 2022). These methods reduce the quadratic cost of self-attention, but typically incur accuracy gaps on fine-grained language-understanding tasks.

Representation-Space Variants

Tian et al. (2024) introduce EulerFormer, which uses Euler's formula to represent token embeddings as polar-form complex vectors, with real and imaginary parts taken from the two halves of the embedding. Lee, Hasegawa, and Gao (2022) survey complex-vector constructions, including rectangular (magnitude–phase) and polar (rotation-invariant) decompositions. Yet, these approaches do not explicitly and interpretably disentangle global from local semantics.

Complementary Approaches

Training-time regularization includes information bottlenecks (e.g., InfoBERT (Wang et al. 2021)), adversarial or perturbed pretraining (e.g., ALUM (Liu et al. 2020)), hidden-state noise (e.g., CreAT (Wu et al. 2023b)), and rationale-guided policy optimization (e.g., GRPO (Miralles-González et al. 2025)). *Inference-time post-processing* uses mixture-of-agents ensembling (Wang et al. 2025), instruction-tuned prompts (Kavumba et al. 2023), explanation-aware rerankers (Koulakos et al. 2024), and self-consistency voting (Abdaljalil et al. 2025; Jiang et al. 2025). *Data-centric augmentation* leverages GPT-3–expanded WANLI (Liu et al. 2022; Williams, Nangia, and Bowman 2018), logic-based perturbations (e.g., LogicAttack (Nakamura et al. 2023)), and rationale-augmented corpora (e.g., SMANLI/EMANLI (Pieper et al. 2024)). While effective for NLI, these methods still rely on extra prompts, adversarial data, or ensemble machinery.

We present a Complex–Vector Token Representation with *Token-Context Attention* that improves APP without extra prompt engineering, adversarial data, or reinforcement learning from human feedback (RLHF), while remaining training costs comparable to, or even lower than standard Transformers.

Method

We draw on ideas from signal processing, where signals are often represented in polar coordinates using a magnitude and a phase to decouple different attributes. In this framework, a signal's magnitude encodes its strength, while its phase represents the position relative to a reference signal (Feynman, Leighton, and Sands 2010). Analogously, we decouple a global semantic vector $\mathbf{G}$ and per-token local semantic vectors $\boldsymbol{\alpha}$ from the input text, and re-encode each token by its dimension-wise angular deviation $\boldsymbol{\alpha}$ relative to $\mathbf{G}$ as $\mathbf{G} \cdot e^{i \cdot \boldsymbol{\alpha}}$, where $\mathbf{G}$ summarizes the overall semantics and serves as a relatively stable background. After converting this polar form to Cartesian coordinates, the real part serves as the token representation, while the imaginary part encodes its contextual representation. Both parts are transformation-induced and modulated by a semantic gate $\cos(\boldsymbol{\alpha}) + 1$ applied elementwise, which amplifies semantic alignment and suppresses contradictions, thereby mitigating spurious correlations.

Complex–Vector Token Representations

Global Semantics Representation Token-level interpretation depends heavily on the global semantics of the input, which provides the disambiguating context necessary for improved understanding. Formally, given an input text W with n tokens $[\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_j, \dots, \mathbf{w}_n]$, we can denote the j-th token's embedding vector by $\mathbf{w}_j = [w_{j,1}, w_{j,2}, \dots, w_{j,d}] \in \mathbb{R}^d, j \in \{1, 2, \dots, n\}, d \in \mathbb{N}$ (e.g., $d = 768$). Then, we can formulate W as shown in Eq. 1, where each element $w_{j,k}$ denotes the k-th component of the j-th token's embedding vector. From a physical

perspective, the magnitude of each signal component is defined as $G_{j,k} = |w_{j,k}|$ (Oppenheim, Willsky, and Nawab 1996). Based on these definitions, the token magnitude matrix shown in Eq. 2 contains the magnitude of each token in every embedding dimension and is computed from the token embedding matrix in Eq. 1.

$$\mathbf{W}=\begin{bmatrix}\mathbf{w}_1\\ \vdots\\ \mathbf{w}_n\end{bmatrix}=\begin{bmatrix}w_{1,1} & w_{1,2} & \dots & w_{1,k} & \dots & w_{1,d}\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ w_{n,1} & w_{n,2} & \dots & w_{n,k} & \dots & w_{n,d}\end{bmatrix}\in\mathbb{R}^{n\times d} \quad (1)$$

For example, $G_{j,k}$ corresponds to the magnitude of the j-th token along the k-th dimension, and is defined as $G_{j,k} = |w_{j,k}|$. Once all per-token magnitudes are computed, we aggregate them across tokens along each embedding dimension to obtain the **global semantic vector**: $\mathbf{G}=[G_1, \dots, G_k, \dots, G_d]$.

$$\begin{bmatrix}G_{1,1}\dots G_{1,k}\dots G_{1,d}\\ \vdots\ \ddots\ \vdots\ \ddots\ \vdots\\ G_{n,1}\dots G_{n,k}\dots G_{n,d}\end{bmatrix}=\begin{bmatrix}|w_{1,1}|\dots|w_{1,k}|\dots|w_{1,d}|\\ \vdots\ \ddots\ \vdots\ \ddots\ \vdots\\ |w_{n,1}|\dots|w_{n,k}|\dots|w_{n,d}|\end{bmatrix} \quad (2)$$

For each dimension k, the magnitude component G_k is the L2 norm of the k-th column of the token magnitude matrix in Eq. 2, formally: $G_k = \left\|\mathbf{G}_{:,k}\right\|_2 = \sqrt{\left(\sum_{j=1}^{n} G_{j,k}^2\right)} = \sqrt{\left(\sum_{j=1}^{n} w_{j,k}^2\right)}$, where $G_{j,k} = |w_{j,k}|$.

Local Semantics Representation Local semantics capture token-level meanings that are essential for modeling contextual dependencies and subtle variations. When each token embedding is viewed as a discrete signal, its phase encodes the token's dimension-wise position relative to the global semantic vector. Formally, for each token $\boldsymbol{w_j}$ in an input text, its **phase vector** is $\boldsymbol{\alpha_j} = (\alpha_{j,1}, \alpha_{j,2}, \dots, \alpha_{j,k}, \dots, \alpha_{j,d}) \in \mathbb{R}^d$, where $\alpha_{j,k}$ is defined as $\operatorname{atan2}\left(\sqrt{1-\left(\frac{w_{j,k}}{G_k}\right)^2},\ \frac{w_{j,k}}{G_k}\right) \in [0, \pi]$ based on the corresponding element G_k in the **global semantic vector** of the input text $\mathbf{G} = [G_1, G_2, \dots, G_k, \dots, G_d]$. With these definitions, we can derive the token phase matrix in Eq. 3 from the token embedding matrix in Eq. 1.

$$A=\begin{bmatrix}\boldsymbol{\alpha}_1\\ \vdots\\ \boldsymbol{\alpha}_n\end{bmatrix}=\begin{bmatrix}\alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,k} & \dots & \alpha_{1,d}\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ \alpha_{n,1} & \alpha_{n,2} & \dots & \alpha_{n,k} & \dots & \alpha_{n,d}\end{bmatrix}\in\mathbb{R}^{n\times d} \quad (3)$$

Adapt Magnitude with Local Semantics We adapt the global semantic vector $\mathbf{G}$ using each token embedding $\mathbf{w}_j$ as $\mathbf{G'} = [G_1 + w_{j,1},\ G_2 + w_{j,2},\ \dots,\ G_d + w_{j,d}]$ prior to constructing the Complex–Vector Token Representation, which is finally given by $\mathbf{Z_j} = \mathbf{G'} \cdot e^{i\cdot\boldsymbol{\alpha_j}} \in \mathbb{C}^d$. For example, steps one to three in Fig. 1 illustrate how each token in the input text "I am alive" is transformed into three Complex–Vector Token Representations.

Dual Directional Gating

As Step IV in Fig.1 shows, converting Complex–Vector Token Representation from polar to Cartesian coordinates via Euler's formula $e^{i\theta} = \cos(\theta) + i \cdot \sin(\theta)$(Ahlfors 1966) enables both real and imaginary parts to implement directional gates, modulating a token's individual and contextual contributions through the cosine alignment (i.e., $\cos(\alpha_{j,k})$) with global semantics in each dimension k. As Tab. 1 shows,

Case	$\boldsymbol{\alpha_{j,k}}$	$\cos\alpha_{j,k}+1$	Semantic Effect
Aligned	$0°$	2	Enhance both
Orthogonal	$90°$	1	Preserve both
Opposite	$180°$	0	Suppress both

Table 1: Real- & imaginary-part directional gates.

when a token is fully aligned with its global semantics, its overall coefficient approaches 2, thereby amplifying both the token's own and its contextual semantic contributions. When a token is orthogonal to its global semantics, the coefficient becomes 1, preserving the original weight of both components. When a token points in the opposite direction, the coefficient drops to 0, effectively suppressing both semantics.

Token-Context Attention

As Step V in Fig. 1 shows, we adopt cross-attention and use tokens (real parts) as the queries (Q) and values (V), while the contexts of all tokens (imaginary parts) serve as the keys (K). **Attention weights are computed between the queries and all contextual keys, and each token is updated by aggregating the corresponding token values under these weights.** This mechanism can be formally defined as: $w_j^{(t+1)} = \text{torch.complex}\Big(\sum_i \text{Attn}\Big[\text{Re}\big(w_j^{(t)}\big), \text{Im}\big(w_i^{(t)}\big)\Big] \cdot \text{Re}\big(w_i^{(t)}\big), \text{Im}\big(w_j^{(t)}\big)\Big)$.

Experiments

In this work we focus on three widely used NLI datasets: ANLI, SNLI, and MNLI. We train and select models on ANLI, the most challenging of the three, and then evaluate zero- and few-shot transfer to SNLI and MNLI to probe cross-domain generalization. We additionally include Mamba as a strong non-Transformer baseline. We compare *Token–Context Attention* with both 0.8B- and 1B-parameter baselines in terms of APP, while FGSM/PGD and generalization experiments are conducted against only the 1B-parameter group using ΔAccuracy as the evaluation metric. We conduct all experiments on H100 GPUs, and list detailed library versions in the supplementary material.

We set `use_deterministic_algorithms(True)` to enforce deterministic operations, and set configure `cudnn.deterministic=True` and `cudnn.benchmark=False` to disable dynamic algorithm selection in cuDNN. We then disable TF-32 by setting `cuda.matmul.allow_tf32=False` and `cudnn.allow_tf32=False` to enforce FP32 computation. We further export the environment variable `cublas_workspace_config=:4096:8` to enable reproducible results in cuBLAS. We also initialize each DataLoader instance with a seeded `torch.Generator`,

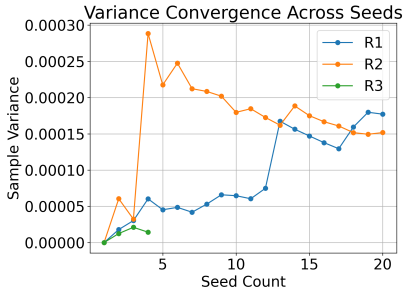

Figure 2: T-C Attn convergence curves on R1/R2/R3.

and synchronize all relevant pseudorandom number generators, such as `random`, `numpy`, `torch`, `torch.cuda`. Additionally, `num_workers` is set to zero for all DataLoaders to eliminate nondeterminism multiprocessing.

All baselines follow the learning-rate schedules specified in their original papers, and we perform an independent peak learning-rate search on ANLI-R1. The optimal rate is then applied uniformly across all ANLI subsets. Since some baselines vary weight decay across tasks, we evaluate 0.01, 0.05, and 0.1 on ANLI-R1 with five seeds. For *Phi-1.5* 0.8B and *Mamba* 790M, 0.05 slightly outperforms 0.1, but average APP gaps over 20 seeds are negligible (≤ 0.5 pp). We thus fix weight decay to 0.1 throughout. To preserve baseline fidelity, each model retains its original dropout setting.

For *Token-Context Attention*, we adopt *Llama3.2-1B*'s tokenizer with a two-stage learning rate schedule by a $LambdaLR$ scheduler consisting of linear warm-up (10% epochs) followed by cosine annealing. The base, peak and lowest learning rate are set to 4e-6, 8e-5 and 2e-5 after a grid search on ANLI R1. Our method is relatively sensitive to dropout rate and weight decay due to its asymmetric attention mechanism, therefore, we conduct small-scale tuning of these parameters independently on each of R1, R2, and R3.

For architecture comparisons, all models share the same initialization: LayerNorm weights are set to 1.0 with zero bias; linear and convolutional layers use Xavier uniform with zero bias; embeddings follow a normal distribution with mean 0.0 and standard deviation 0.02 for baselines, or 1.2 for *Token-Context Attention*. In addition, *Mamba* initializes its state-space mixer following its original design: $A\_log \sim \log(\mathcal{U}(0, 0.5))$ and the dt_proj bias is drawn from the inverse softplus of $\mathcal{U}(0.001, 0.1)$. To minimize the influence of scaling effects, we separate the comparisons into two groups: 0.8B (by pruning to 11 layers for *Llama3.2-1B* and 15 layers for *Phi-1.5*) and 1B group. For ANLI-R1 and R2, we report results averaged over 20 independent random seeds shared across all models. As illustrated in Fig. 2, the sample variance on the R3 subset stabilizes below 2.5×10^{-5} once four random seeds are averaged, yielding a standard deviation of merely $\sigma \approx 0.47\,\%$. By contrast, the variances for R1 and R2 do not plateau until at least six seeds and stay around 1.4–1.6×10^{-4} (i.e., $\sigma \approx 1.2\,\%$) even after twenty seeds. Considering these negligible variance reductions beyond four seeds, together with the prohibitive cost of a full 20-seed protocol on R3 (e.g., 21 GPU·h per seed for *Token-Context Attention* and 27 GPU·h for *Mamba*, totaling $\approx$ 2800 GPU·h or 15 days on an $8 \times$ H100 node), we therefore cap R3 to four seeds, which still affords statistically reliable and cross-model-comparable estimates. We nonetheless include this four-seed run to verify whether the variance pattern observed on R1/R2 persists under a different data distribution and to probe potential shifts in relative model ranking. For deeper analysis, we also report 95 % confidence intervals from 1000 bootstrap resamples for the 0.8B–R3 subset in the results section.

As Tab. 2 shows, on the 0.8B-R1/R2 group and all three subsets of the 1B-parameter group, *Token–Context Attention* consistently outperforms all baselines of comparable or larger scale in terms of APP, with statistical significance at the $\alpha = 0.05$ level. Regarding training efficiency, our method achieves the lowest GPU·h and FLOPs on the 0.8B-R1 and 1B-R1/R2 group, and remains within a modest margin of the most compute-efficient baseline on the remaining subsets. Taken together, these results suggest that our method offers favorable parameter efficiency by achieving higher APP while using comparable or lower FLOPs in most of the evaluated settings. Fig. 3 illustrates the distribution of APP across all ANLI subsets.

On the 0.8B-R3 group, *Mamba* surpasses our method in APP by 1.24 pp, but requires approximately 11 pp more FLOPs (3.39k vs. 3.04k). Compared to *LLaMA*, our method achieves a 3.04 pp gain in APP at the cost of approximately 60 pp more FLOPs (3.04k vs. 1.89k). When compared to *Phi*, the FLOPs are nearly identical (3.04k vs. 3.02k), with our method showing a 1.65 pp advantage in APP. Similarly, our method outperforms *LLaMA* with ΔAPP = 3.05 pp and $p = 0.022$, which is also statistically significant. In contrast, the difference with *Phi* (ΔAPP = 1.65 pp, $p = 0.136$) can be considered a borderline case.

To further validate performance discriminability on the 0.8B-R3 subset, we computed 95% confidence intervals (CI) via 1000 bootstrap resamples: *Token–Context Attention* [48.90, 51.51], *LLaMA* [45.98, 49.22], *Phi* [47.87, 49.24], and *Mamba* [50.90, 54.51]. The intersections between our method's confidence interval and those of *LLaMA*, *Phi*, and *Mamba* account for 12.3%, 13.0%, and 23.4% of our method's interval width, respectively. These results indicate that four random seeds provide statistically stable estimates of model performance on this subset. Our *Token-Context Attention* model consistently outperforms or matches baseline models across R1 and R2 subsets, validating its effectiveness in general. On the more challenging R3 subset, although occasional overlaps in confidence intervals exist, such cases remain statistically insignificant given the overlapping intervals. Further investigation into these subtle variations, potentially arising from different data distributions or model-specific sensitivities, could be beneficial in future work. Nevertheless, these minor fluctuations do not detract from the overall demonstrated effectiveness of our proposed approach.

On the 1B-R3 group, *Pythia* attains a slightly higher raw accuracy than our model (42.02% vs. 41.65%), but lags behind in APP.

Scale	Data	Models	APP*	VRAM	Time	P-value	ΔAcc	95% CI	GPU·h**	FLOPs***	Acc
0.8B	R1	**T-C Attn** (0.83B)	46.60%	12GB	488s	–	–	–	37.14	2.23k	38.68%
		Phi 1.5 (0.82B)	43.93%	8.8GB	493s	1.29×10^{-6}	2.67%	[1.86%,3.47%]	42.89	2.75k	36.03%
		Llama3.2 (0.85B)	42.36%	10GB	441s	2.01×10^{-7}	4.24%	[3.12%,5.37%]	43.12	2.48k	36.46%
		Mamba (0.79B)	44.06%	9.2GB	578s	5.84×10^{-4}	2.55%	[1.25%,3.84%]	43.83	2.34k	34.81%
	R2	**T-C Attn** (0.83B)	51.87%	13GB	1273s	–	–	–	117.40	5.35k	43.05%
		Phi 1.5 (0.82B)	50.52%	8.8GB	1331s	1.00×10^{-2}	1.34%	[0.36%,2.33%]	131.84	5.87k	41.74%
		Llama3.2 (0.85B)	48.56%	10GB	1178s	5.53×10^{-7}	3.30%	[2.36%,4.24%]	117.94	4.78k	41.28%
		Mamba (0.79B)	49.02%	9.2GB	1887s	3.05×10^{-2}	2.85%	[0.30%,5.40%]	170.35	5.90k	38.73%
	R3	**T-C Attn** (0.83B)	50.20%	13GB	3141s	–	–	–	83.76	3.04k	41.67%
		Phi 1.5 (0.82B)	48.55%	10GB	3307s	1.36×10^{-1}	1.65%	[-0.94%,4.24%]	64.63	3.02k	39.81%
		Llama3.2 (0.85B)	47.16%	11GB	2770s	2.21×10^{-2}	3.05%	[0.83%, 5.26%]	57.87	1.89k	39.22%
		Mamba (0.79B)	51.44%	10GB	3829s	3.87×10^{-2}	-1.78%	[-3.26%,-0.29%]	101.04	3.39k	40.64%
1B	R1	**T-C Attn** (0.83B)	46.60%	12GB	488s	–	–	–	37.14	2.23k	38.68%
		Llama3.2 (1.23B)	29.03%	14GB	686s	3.09×10^{-21}	17.57%	[16.80%,18.35%]	65.36	4.21k	35.71%
		Pythia (1.08B)	31.96%	11GB	556s	3.33×10^{-17}	14.64%	[13.58%,15.69%]	43.24	3.52k	34.52%
		Phi 1.5 (1.3B)	27.81%	15GB	892s	3.91×10^{-5}	18.79%	[17.85%,19.74%]	84.74	5.20k	36.03%
	R2	**T-C Attn** (0.83B)	51.87%	13GB	1611s	–	–	–	148.57	5.35k	43.05%
		Llama3.2 (1.23B)	33.93%	14GB	1846s	2.23×10^{-20}	17.94%	[17.06%,18.81%]	171.27	8.22k	41.74%
		Pythia (1.08B)	38.33%	11GB	1508s	4.79×10^{-15}	15.53%	[12.26%,14.81%]	208.61	12.15k	40.70%
		Phi 1.5 (1.3B)	31.37%	15GB	2348s	1.08×10^{-21}	20.49%	[19.64% ,21.35%]	200.88	9.41k	40.79%
	R3	**T-C Attn** (0.83B)	50.18%	13GB	3126s	–	–	–	83.36	3.04k	41.65%
		Llama3.2 (1.23B)	33.22%	15GB	4121s	3.35×10^{-5}	16.96%	[14.66%, 19.25%]	68.68	2.97k	40.87%
		Pythia (1.08B)	38.91%	11GB	3370s	1.69×10^{-3}	11.29%	[7.97%, 14.62%]	109.53	5.67k	42.02%
		Phi 1.5 (1.3B)	30.74%	15GB	5119s	1.95×10^{-5}	19.44%	[17.12% ,21.74%]	106.65	4.53k	39.97%

* Accuracy per Parameters (in billions), ** For all seeds in one subset, *** FLOPs(PF) = non-embedding params × total tokens × $6/10^{15}$.

Table 2: Performance comparison between Token-Context Attention (T-C Attn) and 0.8B & 1B baselines.

FGSM/PGD

All models were evaluated on ANLI-R1 using early-stopped checkpoints trained with the same 10 seeds. Fixing one seed, we selected *LLaMA* and *Pythia* as representative models and conducted a grid search over perturbation magnitudes $\epsilon \in \{10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}, 10^{-2}, 3\times10^{-2}, 10^{-1}\}$. For each ϵ, we computed clean and perturbed accuracies as $\Delta_{\text{acc}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{FGSM}}$. We then selected the perturbation magnitude that caused the largest degradation (20–30 percentage points) in either model, which corresponded to $\epsilon = 10^{-4}$ for all comparisons. We enabled deterministic PyTorch behavior with a batch size of 32 and a max sequence length of 256. The attack used a single FGSM step on input embeddings, adding ϵ times the sign of the gradient (masked to avoid perturbing paddings) under a cross-entropy loss.

We performed a grid search over PGD hyperparameters on ANLI-R1 early-stopped checkpoints (10 seeds), varying perturbation magnitude $\epsilon \in \{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$, step size fraction $\alpha \in \{0.1\epsilon, 0.25\epsilon, 0.5\epsilon\}$, and iteration count $T \in \{1, 3, 5\}$. For each (ϵ, α, T), we computed clean accuracy $\text{Acc}_{\text{clean}}$, adversarial accuracy Acc_{PGD}, and the drop $\Delta_{\text{acc}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{PGD}}$. We selected the parameters that caused the largest degradation (20–30 percentage points) in either model. For baselines with embedding std $\sigma = 0.02$, we set $\epsilon = \sigma/200 = 10^{-4}$; for our method ($\sigma = 1.2$), $\epsilon = \sigma/200 = 6 \times 10^{-4}$. We run PGD for $T = 5$ steps with $\alpha = 2 \times 10^{-5}$, batch size 32, and max sequence length 256. Perturbations apply only to input embeddings and are masked to exclude paddings. All runs use fixed random seeds and fully deterministic PyTorch/CuDNN.

For the 20-restart PGD test, we tokenize each batch once and fix the embeddings throughout. To increase difficulty, we reduce the embedding std-to-ε ratio from 200 to 20. PGD is run with $R = 20$ random restarts, initializing $\delta \sim \mathcal{U}[-\varepsilon, \varepsilon]$ and updating for $T = 20$ steps using $\delta \leftarrow \text{clip}_{[-\varepsilon,\varepsilon]}\left(\delta + \alpha\,\text{sign}(\nabla_\delta \mathcal{L}) \odot \text{mask}\right)$ with cross-entropy loss and step size $\alpha = 5\times10^{-5}$. We further compute logits from perturbed embeddings, with padding masked out. We then record the argmax prediction for each restart, and for each input, define the worst-case output as the first restart that flips the predicted label. If none flip it, we retain the clean prediction. All other settings match those in the single-step PGD test.

As Fig. 4 shows, with ΔAcc = Clean − Attack, FGSM (left) causes only a tiny drop for *Token-Context Attention* and *Mamba* (≈ 0–1 pp), while *LLaMA* degrades the most (≈ 18–20 pp) and *Pythia*/*Phi* fall in between (≈ 10–15 pp). Under single-run PGD (middle), *LLaMA* again suffers the largest loss (≈ 23–26 pp), *Pythia* and *Phi* show moderate drops (≈ 13–17 pp), *Token-Context Attention* degrades by

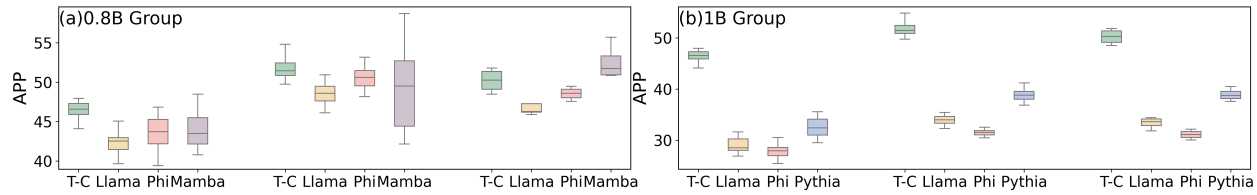

Figure 3: Test accuracy per billion parameters.

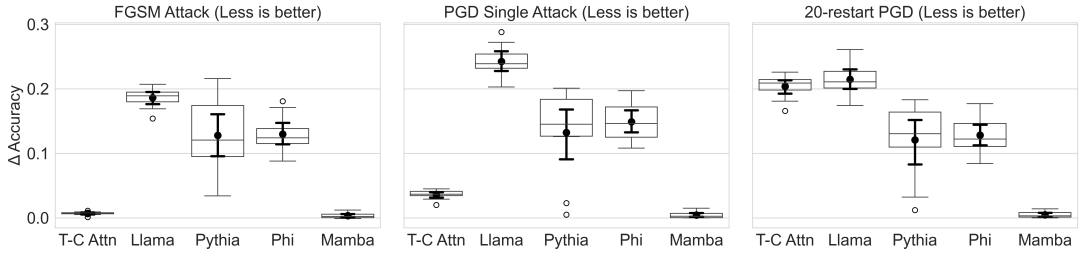

Figure 4: Gradient attack (FGSM/PGD) comparisons.

only $\approx$4–5 pp, and *Mamba* remains near zero. The stronger 20-restart PGD (right) yields the largest drops for *LLaMA* and our method ($\approx$ 19–25 pp), *Pythia*/*Phi* are moderate ($\sim$ 11–15 pp), and *Mamba* consistently incurs a sub-1 pp drop.

Our results suggest that, for the *Token-Context Attention* model, the seemingly small effect of FGSM and single-step PGD is unlikely to arise from gradient masking, because accuracy declines smoothly as attack strength increases (FGSM $\ll$ single-step PGD $<$ 20-restart PGD) and additional random restarts further reduce performance. Nevertheless, the model experiences a larger drop once stronger, properly optimized PGD is applied, indicating that damaging directions remain discoverable. In contrast, LLaMA shows consistently large drops, whereas *Mamba* shows near-zero degradation across attacks, which is consistent with prior observations (Malik et al. 2025). Under the σ*-scaled* budget, both *Token-Context Attention* and *Mamba* show a similarly small accuracy drop ($\leq$ 3 pp), whereas the other baselines degrade by $\approx$ 10–20 pp. Because the absolute ϵ differs across models, this observation should be viewed as a *trend* rather than a conclusive advantage.

Generalization

Starting from ANLI-trained checkpoints saved via early stopping (10 seeds), we first evaluate each on the ANLI R1 test set. We then select another 10 random seeds to perform zero-shot evaluation on the SNLI/MNLI test sets, fine-tune each on 10 SNLI/MNLI examples, re-evaluate on the full SNLI/MNLI test sets, and finally re-test on ANLI R1 to assess forgetting. This ANLI$\rightarrow$SNLI/MNLI (zero-shot)$\rightarrow$SNLI/MNLI (10-shot)$\rightarrow$ANLI sequence is applied uniformly across all models, and we track accuracy at each stage for comparison.

On zero-shot ANLI$\rightarrow$SNLI transfer (upper of Fig. 5), *Token-Context Attention* has a modest negative median in ΔAccuracy, and *Pythia* and *Mamba* exhibit negative shifts of similar magnitude. *LLaMA* has a slightly negative median and marginally positive mean with 95% CI overlapping zero, while *Phi* has a slightly positive median but slightly negative mean. In 10-shot evaluation, all models improve by less than 0.5 pp: *Token-Context Attention* attains the highest median and mean gains; *Mamba* shows the widest spread with large negative outliers, whereas *LLaMA*, *Pythia*, and *Phi* change minimally. After SNLI fine-tuning, when ANLI is retested, average forgetting remains within $\pm$0.3 pp for all models but lacks statistical significance, since each mean 95% CI still overlaps zero. *Token-Context Attention* has a wider interquartile range, *LLaMA* shows the tightest distribution, and *Mamba* has isolated severe drops.

Under the MNLI-matched setting (bottom of Fig. 5), only *LLaMA* shows both a clearly positive mean and median ($\approx$2 pp); *Phi* improves modestly ($\approx$1 pp); *Token-Context Attention* and *Mamba* both show mild negative shifts of about 1–1.5 pp below parity, while *Pythia* stays close to parity with both mean and median near zero. On the mismatched split, *LLaMA* continues to lead, *Phi* stays slightly positive, *Pythia* again stays close to parity with both mean and median near zero, while *Token-Context Attention* and *Mamba* both again decline by roughly 1–1.5 pp. With 10-shot support, all models improve by no more than 0.5 pp: *Token-Context Attention* attains the highest median and mean gains on both splits. Re-evaluated on ANLI, all models show mean ΔAccuracy within $\pm$0.3 pp; *Token-Context Attention* exhibits a slightly wider spread across seeds, *LLaMA* is the most stable, and *Mamba* remains tightly clustered with a few extreme negative outliers.

For *Token-Context Attention*, the trend is clear. Across SNLI and MNLI, zero-shot transfer shows only small drops in ΔAccuracy, while with just ten labeled examples the model attains the largest few-shot gains among all baselines and keeps its post-adaptation ANLI accuracy within about 0.3 pp of the original score. One possible reason is that the gating mechanism of the real and imaginary parts tends to enhance or suppress semantic dimensions based on the

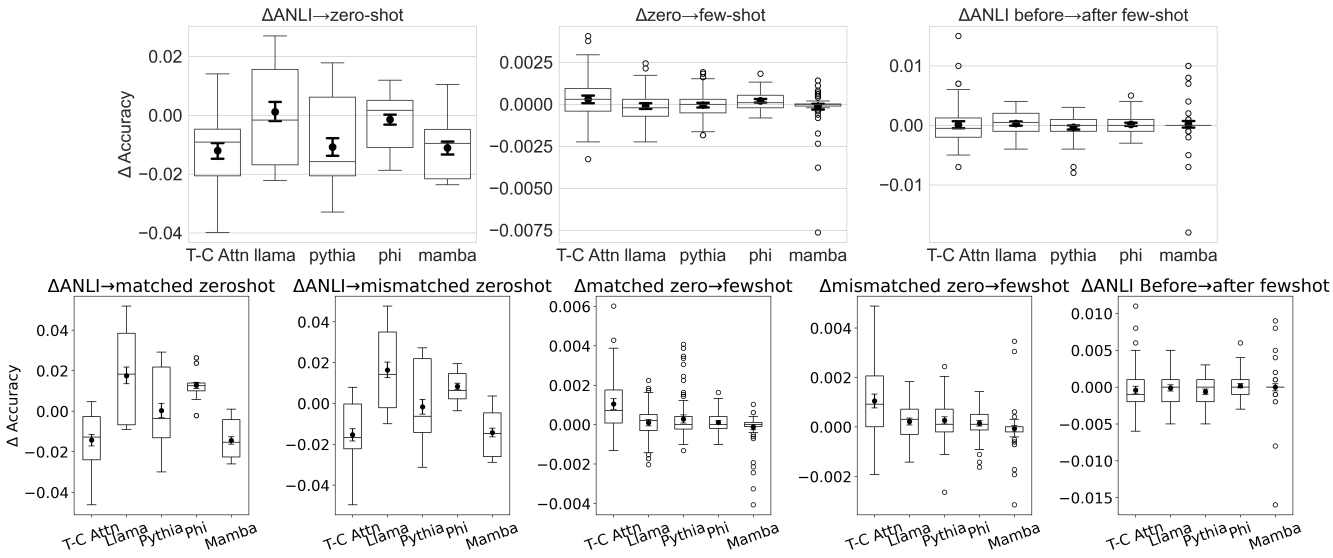

Figure 5: **Top:** Generalization on SNLI; **Bottom:** Generalization on MNLI.

alignment between tokens and global semantics. Moreover, computing attention weights between each token and the contexts of all tokens in the *Token-Context Attention* shifts the attention pattern from being driven by local lexical similarity toward being aligned with the semantics of the entire input text, because a token's context is naturally closer to the global semantics. As a result, this mechanism provides one possible explanation for the generalization behavior we observe on SNLI and MNLI.

Ablation Study

All ablation studies are conducted on ANLI R1. As shown in Fig. 6, enabling backpropagation (BP) through the real part raises mean validation accuracy from 33.57 to 38.68 pp. While gradients through the imaginary part do not affect final accuracy, Fig. 7 shows they significantly accelerate convergence: the median (mean) number of epochs to reach the best score drops by 2 (2.65). This may be because the imaginary component, involved in attention weight computation, introduces an additional gradient path that speeds up training. We further compare our full *Complex–Vector Token Representation* with a variant lacking directional gates in both real and imaginary parts. On ANLI R1 (20 seeds), both perform similarly (38.68 pp vs. 38.64 pp). However, on four challenging ANLI R2 seeds (46–144), the gated model scores 43.15 pp vs. 42.10 pp; on two difficult ANLI R3 seeds (857 and 4385), it reaches 41.63 pp vs. 40.42 pp. These results suggest that dual directional gates may offer greater benefits on more adversarial instances. We also replaced *Token-Context Attention* with self-attention, yielding 35.95 pp over 20 seeds, showing a 2.73 pp drop. Fig. 6 further shows that using a fixed global semantic vector (i.e., 1) or random phase values (to test token–global alignment) leads to chance-level performance (≈33 pp), highlighting the critical role of both token and global semantics in the *Complex–Vector Token Representation*.

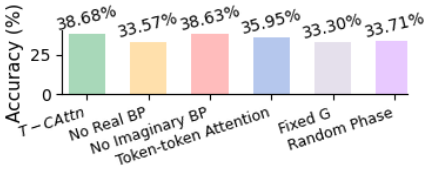

Figure 6: T-C Attn: ablation study on ANLI-R1

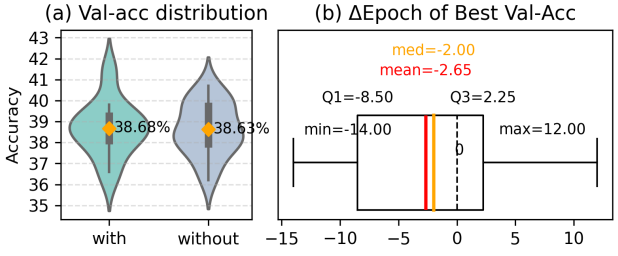

Figure 7: Acc distribution & Δepoch of no imaginary BP

Conclusion

We propose a *Complex–Vector Token Representation* and a *Token-Context Attention* mechanism that improve parameter efficiency of Transformer-based NLI models. Empirical results show that higher parameter efficiency not only increases accuracy but also contributes to reducing computation cost (FLOPs). In addition, our approach exhibits promising trends toward the largest few-shot generalization from ANLI to SNLI and MNLI, and often incurs smaller or comparable accuracy drops than the self-attention baselines under embedding-level gradient attacks.

Acknowledgments

This work used DeltaAI at the National Center for Supercomputing Applications (NCSA) and ACES at Texas A&M University through allocation CIS250625 (ACCESS) program (Boerner et al. 2023), which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. We also acknowledge support from the U.S. National Science Foundation through the project "Category II: REPACSS: Empowering Scientific Discovery through Renewable Energy Powered Advanced Computing Systems and Services" (Award No. 2404438).

References

Abdaljalil, S.; Kurban, H.; Qaraqe, K.; and Serpedin, E. 2025. Theorem-of-Thought: A Multi-Agent Framework for Abductive, Deductive, and Inductive Reasoning in Language Models. In Zhang, Y.; Chen, C.; Li, S.; Geva, M.; Han, C.; Wang, X.; Feng, S.; Gao, S.; Augenstein, I.; Bansal, M.; Li, M.; and Ji, H., eds., *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)*, 111–119. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-283-1.

Ahlfors, L. V. 1966. *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*. New York: McGraw-Hill, 2 edition.

Boerner, T. J.; Deems, S.; Furlani, T. R.; Knuth, S. L.; and Towns, J. 2023. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and experience in advanced research computing 2023: Computing for the common good*, 173–176.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In Màrquez, L.; Callison-Burch, C.; and Su, J., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 632–642. Lisbon, Portugal: Association for Computational Linguistics.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Canziani, A.; Paszke, A.; and Culurciello, E. 2016. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

Feynman, R. P.; Leighton, R. B.; and Sands, M. 2010. *The Feynman lectures on physics; New millennium ed.* New York, NY: Basic Books. Originally published 1963-1965.

Golovneva, O.; Wang, T.; Weston, J.; and Sukhbaatar, S. 2025. Multi-Token Attention. In *Conference on Language Modeling (COLM 2025)*. Montreal, Canada.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Guo, M.; Zhang, Y.; and Liu, T. 2019. Gaussian transformer: a lightweight approach for natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6489–6496.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 107–112. New Orleans, Louisiana: Association for Computational Linguistics.

Jiang, J.; Bewley, T.; Amoukou, S. I.; Leofante, F.; Rago, A.; Mishra, S.; and Toni, F. 2025. Representation Consistency for Accurate and Coherent LLM Answer Aggregation. *arXiv preprint arXiv:2506.21590*.

Kavumba, P.; Brassard, A.; Heinzerling, B.; and Inui, K. 2023. Prompting for explanations improves Adversarial NLI. Is this true?{Yes} it is {true} because {it weakens superficial cues}. In *Findings of the Association for Computational Linguistics: EACL 2023*, 2165–2180.

Koulakos, A.; Lymperaiou, M.; Filandrianos, G.; and Stamou, G. 2024. Enhancing adversarial robustness in Natural Language Inference using explanations. In Belinkov, Y.; Kim, N.; Jumelet, J.; Mohebbi, H.; Mueller, A.; and Chen, H., eds., *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, 105–117. Miami, Florida, US: Association for Computational Linguistics.

Lee, C.; Hasegawa, H.; and Gao, S. 2022. Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, 9(8): 1406–1426.

Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; and Ontañón, S. 2022. FNet: Mixing Tokens with Fourier Transforms. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4296–4313. Seattle, United States: Association for Computational Linguistics.

Liu, A.; Swayamdipta, S.; Smith, N. A.; and Choi, Y. 2022. WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022*, 6826–6847. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Liu, X.; Cheng, H.; He, P.; Chen, W.; Wang, Y.; Poon, H.; and Gao, J. 2020. Adversarial Training for Large Neural Language Models. *arXiv preprint arXiv:2004.08994*. Published 20 April 2020.

Madaan, L.; Esiobu, D.; Stenetorp, P.; Plank, B.; and Hupkes, D. 2025. Lost in Inference: Rediscovering the Role of Natural Language Inference for Large Language Models. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Pa-*

pers), 9229–9242. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-189-6.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR) 2018 (Poster Track)*.

Malik, H. S.; Shamshad, F.; Naseer, M.; Nandakumar, K.; Khan, F. S.; and Khan, S. 2025. Towards Evaluating the Robustness of Visual State Space Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 3544–3553.

Miralles-González, P.; Huertas-Tato, J.; Martín, A.; and Camacho, D. 2025. Pushing the boundary on Natural Language Inference. *arXiv preprint arXiv:2504.18376*.

Nakamura, M.; Mashetty, S.; Parmar, M.; Varshney, N.; and Baral, C. 2023. Logicattack: Adversarial attacks for evaluating logical consistency of natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 13322–13334.

Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J.; and Kiela, D. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4885–4901.

Oppenheim, A. V.; Willsky, A. S.; and Nawab, S. H. 1996. *Signals & systems (2nd ed.)*. USA: Prentice-Hall, Inc. ISBN 0138147574.

Pieper, T.; Ballout, M.; Krumnack, U.; Heidemann, G.; and Kühnberger, K.-U. 2024. Enhancing Small Language Models via ChatGPT and Dataset Augmentation. In *International Conference on Applications of Natural Language to Information Systems*, 269–279. Springer.

Shen, L.; Pu, Y.; Ji, S.; Li, C.; Zhang, X.; Ge, C.; and Wang, T. 2024. Improving the Robustness of Transformer-based Large Language Models with Dynamic Attention. In *NDSS*.

Sun, Y.; Dong, L.; Huang, S.; Ma, S.; Xia, Y.; Xue, J.; Wang, J.; and Wei, F. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Tian, Z.; Zhao, W. X.; Zhang, C.; Zhao, X.; Ma, Z.; and Wen, J.-R. 2024. EulerFormer: Sequential User Behavior Modeling with Complex Vector Attention. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1619–1628.

Wang, B.; Wang, S.; Cheng, Y.; Gan, Z.; Jia, R.; Li, B.; and Liu, J. 2021. InfoBERT: Improving Robustness of Language Models from An Information Theoretic Perspective. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.

Wang, J.; Hu, X.; Hou, W.; Chen, H.; Zheng, R.; Wang, Y.; Yang, L.; Ye, W.; Huang, H.; Geng, X.; Jiao, B.; Zhang, Y.; and Xie, X. 2023. On the Robustness of ChatGPT: An Adversarial and Out-of-distribution Perspective. In *ICLR Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*. Available at https://openreview.net/forum?id=uw6HSkgoM29.

Wang, J.; WANG, J.; Athiwaratkun, B.; Zhang, C.; and Zou, J. 2025. Mixture-of-Agents Enhances Large Language Model Capabilities. In *The Thirteenth International Conference on Learning Representations*.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1112–1122. New Orleans, Louisiana: Association for Computational Linguistics.

Wu, H.; Ding, R.; Zhao, H.; Xie, P.; Huang, F.; and Zhang, M. 2023a. Adversarial self-attention for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 13727–13735.

Wu, H.; Liu, Y.; Shi, H.; Zhao, H.; and Zhang, M. 2023b. Toward Adversarial Training on Contextualized Language Representation. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*.

Yang, S.; Wang, B.; Shen, Y.; Panda, R.; and Kim, Y. 2024. Gated Linear Attention Transformers with Hardware-Efficient Training. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yao, X.; Zheng, Y.; Yang, X.; and Yang, Z. 2022. Nlp from scratch without large-scale pretraining: A simple and efficient framework. In *International Conference on Machine Learning*, 25438–25451. PMLR.