SCALABLE HIERARCHICAL TRANSFORMERS FOR MULTI-TURN JAILBREAK DETECTION IN LONG CONVERSATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn jailbreaks can evade turn-level moderation by spreading unsafe intent across a dialogue through gradual escalation, reframing, and role manipulation. We address multi-turn jailbreak detection as a conversation-level classification problem and introduce an efficient hierarchical detector that reduces full-context attention cost while retaining cross-turn reasoning. The model encodes individual turns to form compact turn representations and applies a lightweight conversation module that captures dialogue dynamics and selectively attends to fine-grained evidence when needed. On a challenging evaluation benchmark of $14{,}038$ conversations, our approach achieves a positive-class F1 of 0.9394, outperforming Claude Opus 4.7, the strongest competing baseline, by 0.0994 while halving its false-positive rate. Ablation studies confirm that each architectural component contributes meaningfully, with combining cross-attention and self-attention in the conversation module yielding a 2.26 percentage point reduction in false-positive rate over the self-attention-only variant.

1 INTRODUCTION

Large language models (LLMs) deployed in interactive settings are increasingly exposed to multi-turn jailbreak attempts, where an adversary does not issue a single overtly malicious prompt, but instead incrementally steers the conversation toward disallowed behavior. In these attacks, unsafe intent can be distributed across turns via gradual escalation, topic drift, role manipulation, and reframing of earlier instructions. As a result, effective defense requires conversation-level reasoning rather than isolated, turn-by-turn filtering.

A straightforward solution is to concatenate the entire dialogue history into a single sequence and apply a Transformer encoder to classify the conversation as safe or jailbreak. Modern long-context models can process thousands of tokens, but applying full-token attention to every moderation decision can be inefficient in high-throughput, low-latency settings. For a typical setting of 24 turns with 256 tokens per turn, concatenation yields a 6,144-token sequence, and the quadratic cost of self-attention increases training and deployment cost. Moreover, long-context concatenation spends computation uniformly across all tokens, whereas multi-turn jailbreak evidence is often structured by speaker turns and localized to a few phrases. This motivates a hierarchy that encodes local turn semantics first and then applies an explicit conversation-level inductive bias.

We propose an efficient two-level hierarchical architecture for multi-turn jailbreak detection that preserves rich cross-turn reasoning while avoiding the computational bottleneck of full-sequence attention. At the first level, a TurnEncoder processes each turn independently using a pre-trained multilingual encoder (`intfloat/multilingual-e5-base`). Each turn is formatted with explicit speaker markers (two added special tokens, `[USER]` and `[ASST]`) and is padded/truncated to a fixed 256-token budget. The TurnEncoder produces (i) a compact turn summary embedding via the CLS representation and (ii) token-level embeddings for the full turn. Crucially, turns are encoded independently with no cross-turn token sharing at this stage, enabling efficient batched encoding across turns and conversations (with gradient checkpointing used during training to manage memory).

At the second level, a lightweight ConvTransformer reasons over the sequence of turn summaries to capture conversation dynamics that are characteristic of multi-turn jailbreaks. A learnable

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

[CONV_CLS] token is prepended to aggregate global information, and learned position embeddings (turn order) and role embeddings (user/assistant/CLS) allow the model to differentiate early vs. late turns and speaker identity. The ConvTransformer applies bidirectional self-attention over turn summaries to model cross-turn patterns (e.g., escalation and returns to previously refused topics). In addition, each layer performs cross-attention over the token-level embeddings produced by the TurnEncoder, allowing the model to "zoom in" on specific words or phrases within any turn when fine-grained cues are needed. Finally, an attentive pooling mechanism complements the [CONV_CLS] representation by learning to weight turns by importance—useful, for example, when later turns carry stronger jailbreak signals than early benign setup turns.

This hierarchical design yields an attractive efficiency–accuracy trade-off: per-turn encoding remains bounded by a short context window, while the conversation-level module operates on a small sequence of turn summaries, yet retains access to token-level details through cross-attention. In experiments on a challenging 14,038-sample dataset, the proposed model achieves strong performance (95.90% precision, 92.10% recall, and 0.9394 F1), indicating that efficient conversation-level modeling can be highly effective for multi-turn jailbreak detection.

This paper makes the following contributions:

- We formulate multi-turn jailbreak detection as a conversation-level classification problem that demands cross-turn reasoning under realistic length constraints.
- We introduce a hierarchical TurnEncoder–ConvTransformer architecture that avoids quadratic scaling from naive concatenation while preserving access to fine-grained token evidence via cross-attention.
- We demonstrate strong empirical performance on a multi-turn jailbreak dataset, showing that our approach outperforms strong proprietary models such as Claude Opus 4.7.
- We validate the importance of the cross-turn encoder via an ablation study, showing that replacing it with mean pooling substantially degrades jailbreak detection performance.

2 Related Work

Moderation classifiers and jailbreak attacks. Safety moderation is commonly implemented with supervised toxicity or policy classifiers (Borkan et al., 2019; Hanu & team, 2020). Jailbreaks, however, often exploit instruction-following failures rather than explicit toxicity: optimization-based attacks such as GCG (Zou et al., 2023) and AutoDAN (Liu et al., 2024) yield non-toxic surface forms that evade content-level filters, and indirect prompt injection extends the attack surface to retrieved and tool-provided content (Greshake et al., 2023; Zhan et al., 2024).

Multi-turn jailbreaks. Multi-turn attacks distribute intent across a dialogue: Crescendo escalates from benign to malicious requests (Russinovich et al., 2024), PAIR and TAP iteratively refine attacks using target feedback (Chao et al., 2023; Mehrotra et al., 2024), and many-shot jailbreaking exploits long interaction histories (Anil et al., 2024; Ackerman & Panickssery, 2025). Automated multi-turn attack generation continues to scale (Narula et al., 2025; Kumarappan & Mujoo, 2025; Alobaid et al., 2026), and multi-turn benchmarks show that single-turn evaluation underestimates real-world risk (Yang et al., 2025). Recent benchmarks such as MultiBreak (Song et al., 2026) and CoSafe (Yu et al., 2024) measure the safety of the *target* LLM under multi-turn and coreference-based attacks, whereas we evaluate external conversation-level detectors. On the defense side, LAD (Kulkarni, 2026) probes the target model's internal activations for multi-turn attack trajectories; this requires white-box access to the deployed model, whereas our detector operates on the conversation text alone and is model-agnostic. These works motivate defenses that model evolving intent across turns rather than filtering each turn in isolation, which is the setting we target.

Hierarchical and efficient long-input models. Hierarchical Attention Networks (Yang et al., 2016) encode local units before aggregating them globally, whereas efficient Transformers reduce the cost of attending over a single long token stream via sparse attention (Beltagy et al., 2020; Zaheer et al., 2020), hashing or low-rank approximation (Kitaev et al., 2020; Wang et al., 2020; Choromanski et al., 2021), or recurrence and memory (Dai et al., 2019; Rae et al., 2020). We build on the hierarchical view and use the turn structure of safety conversations as an inductive bias: turns are encoded locally,

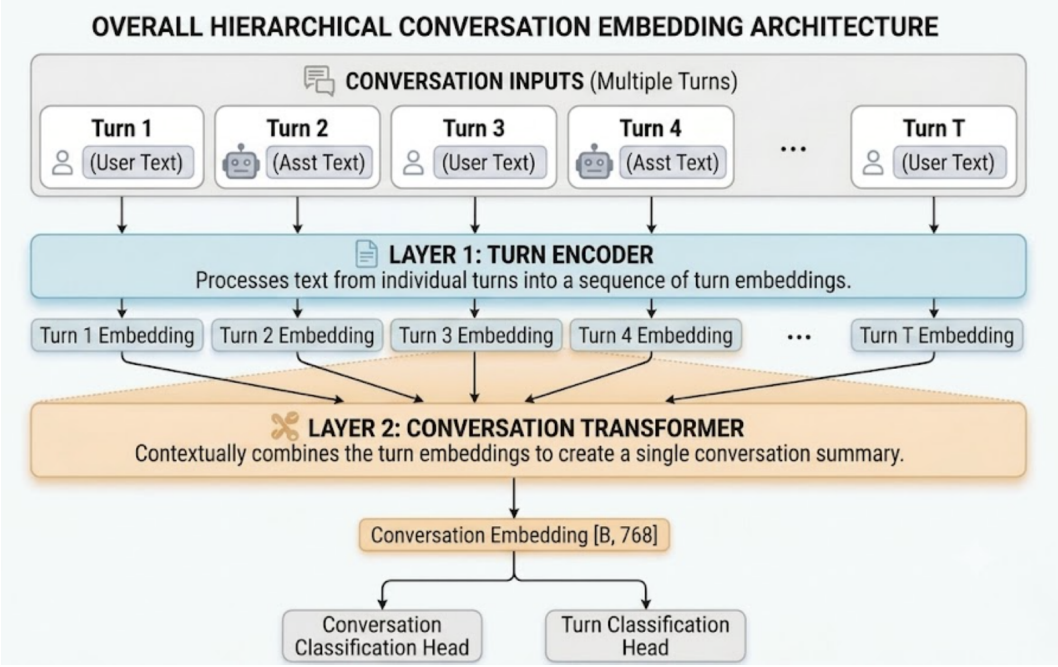


Figure 1: Multi-turn jailbreak detection based on a hierarchical conversation embedding architecture.

conversation-level attention models escalation and reframing, and cross-attention selectively revisits token-level evidence. Extended discussion is provided in Appendix A.1.

3 Problem Formulation

We represent a dialogue as role-annotated turns $\mathcal{C} = (x_1, \ldots, x_T)$, where $x_t = (r_t, u_t)$ and $r_t \in \{\text{user}, \text{assistant}\}$. The binary label $y \in \{0, 1\}$ marks safe conversations (0) versus jailbreak conversations (1); as in the underlying datasets, positive labels may reflect malicious intent or adversarial payloads even when the target assistant refuses, while some sources mark only successful policy violations; this semantics is fixed by the evaluation source.

Given an observed conversation or prefix, the detector estimates

$$p(\mathcal{C}) = \Pr(y = 1 \mid \mathcal{C}), \qquad \hat{y} = \mathbf{1}\{p(\mathcal{C}) \geq \tau\}. \tag{1}$$

Our experiments evaluate an offline/prefix classifier that scores the supplied prefix with bidirectional attention over that prefix; it is not evaluated as a causal streaming early-warning system. Training minimizes binary cross-entropy,

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^{B} \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right]. \tag{2}$$

The defender observes the available dialogue and must identify adaptive multi-turn escalation, reframing, and role manipulation by the user while treating assistant responses as observed context. Formal notation, threat-model details, and deployment caveats are in Appendix A.2.

4 Method

Efficiency motivation. A flattened Transformer over $N = \sum_t n_t$ tokens incurs $\mathcal{O}(N^2)$ self-attention. With 24 turns and 256 tokens per turn, this is a 6,144-token sequence with 37.7M pairwise interactions per layer. With T turns of at most L tokens, our hierarchy instead costs approximately $\mathcal{O}(TL^2 + T^2 + T^2L)$ (per-turn encoding, self-attention over turn summaries, and cross-attention

from turn summaries to the TL token memories) rather than $\mathcal{O}((TL)^2)$; for $T = 24$ and $L = 256$ this is about 1.72M interactions per layer, roughly $22\times$ fewer (an analytical estimate, not a measured speedup). It bounds local encoding to each turn and applies a small conversation module over turn summaries, improving throughput/latency while imposing an inductive bias aligned with dialogue structure. The full complexity derivation is in Appendix A.3.

Architecture. Figure 1 summarizes the model. A multilingual E5-base TurnEncoder processes each role-marked turn independently using added `[USER]` and `[ASST]` tokens, a 256-token cap, padding masks, and gradient checkpointing. It returns a CLS turn summary and token embeddings. A 4-layer ConvTransformer prepends a learned `[CONV_CLS]` token, adds turn-position and role embeddings, applies bidirectional self-attention over the supplied prefix, and uses cross-attention to access the TurnEncoder token memories. Attentive pooling over turn outputs is added to the conversation CLS vector for classification.

4.1 Level 1: TurnEncoder

Each role-marked turn is padded or truncated to $L = 256$ tokens and encoded independently by the TurnEncoder ($L_{\text{enc}} = 12$ layers, $d = 768$). For each turn t it returns a turn summary $\mathbf{s}_t = \mathbf{h}_{t,1}^{(L_{\text{enc}})} \in \mathbb{R}^d$ (the CLS output of the last layer) and token-level embeddings $\mathbf{H}_t = \left[\mathbf{h}_{t,1}^{(L_{\text{enc}})}, \ldots, \mathbf{h}_{t,L}^{(L_{\text{enc}})}\right]^\top \in \mathbb{R}^{L\times d}$. The summaries $\{\mathbf{s}_t\}_{t=1}^T$ feed Level 2 self-attention, while $\{\mathbf{H}_t\}_{t=1}^T$ serve as a cross-attention memory for "zooming in" on salient phrases. There is no cross-turn token sharing at Level 1, so all cross-turn reasoning is deferred to Level 2 and all turns are encoded in one batched forward pass; gradient checkpointing keeps memory manageable for up to 24 turns per conversation. The full configuration is given in Table 5 (Appendix A.3).

4.2 Level 2: ConvTransformer

Overview. Level 2 performs conversation-level reasoning over the sequence of turn summaries and optionally retrieves token-level details through cross-attention. This captures multi-turn patterns such as gradual escalation, topic shifts, instruction hierarchy manipulation, and role changes.

4.2.1 Input sequence construction

Given turn summaries $\{\mathbf{s}_t\}_{t=1}^T$, we prepend a learnable conversation-level token $\mathbf{s}_0$ corresponding to `[CONV_CLS]`:

$$\mathbf{S}^{(0)} \;=\; [\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_T] \in \mathbb{R}^{(T+1)\times d}. \tag{3}$$

The final representation of `[CONV_CLS]` serves as the primary conversation embedding.

4.2.2 Position and role embeddings

Each element in the input sequence is augmented with learned position and role embeddings:

$$\tilde{\mathbf{s}}_i \;=\; \mathbf{s}_i + \mathbf{p}_i + \mathbf{e}_{\rho_i}, \qquad i = 0, \ldots, T, \tag{4}$$

where $\mathbf{p}_i$ is a learned position embedding and $\mathbf{e}_{\rho_i}$ is a learned role embedding; these position, role, and `[CONV_CLS]` parameters are trainable during Stage 1. We use learned position embeddings for 25 positions (position 0 for `[CONV_CLS]`, positions 1–24 for turns), and learned role embeddings for 3 roles (user, assistant, and a CLS placeholder role).

4.2.3 Self-attention over turn summaries

The ConvTransformer is a 4-layer Transformer decoder (pre-layer normalization, 8 heads, Feed-Forward Network (FFN) hidden size 2048, dropout 0.2). It applies bidirectional self-attention (no causal mask) over the augmented summary sequence

$$\tilde{\mathbf{S}}^{(0)} \;=\; [\tilde{\mathbf{s}}_0, \tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_T], \tag{5}$$

because the classifier can observe the full conversation at inference time in the current setting. This supports detecting global multi-turn structures (e.g., returning to a previously refused topic several turns later).

4.2.4 CROSS-ATTENTION TO TOKEN-LEVEL MEMORY

In addition to self-attention over summaries, each decoder layer performs cross-attention over token-level embeddings from Level 1. We flatten all per-turn token embeddings into one memory:

$$\mathbf{M} \;=\; [\mathbf{H}_1; \mathbf{H}_2; \ldots; \mathbf{H}_T] \in \mathbb{R}^{(T \cdot L) \times d}, \tag{6}$$

where $L = 256$ and thus $|\mathbf{M}| = T \cdot L$ tokens (with $T \leq 24$). Tokens in $\mathbf{M}$ are augmented with turn-aware positional signals (reusing the same turn position table) so attention heads can infer which turn each token belongs to. A padding mask is passed to cross-attention so padded token positions introduced by the 256-token turn cap are excluded from the key/value memory. This mechanism allows Level 2 to "zoom in" on specific words or phrases that may carry jailbreak signal.

4.2.5 ATTENTIVE POOLING

After the decoder layers, we compute an additional conversation summary via attentive pooling: a learned query attends over all turn outputs to produce a weighted combination, which is then added to the `[CONV_CLS]` output via a residual connection. This lets the model weight turns differently depending on their importance (e.g., later turns in an escalation attack). The full conversation module configuration is given in Table 6 (Appendix A.3).

4.2.6 CLASSIFICATION HEADS

The final 768-dimensional conversation embedding is passed through dropout and a linear projection to produce a single jailbreak logit:

$$z \;=\; \mathbf{w}^\top \mathbf{h}_{\text{conv}} + b \in \mathbb{R}, \tag{7}$$

where $\mathbf{h}_{\text{conv}} \in \mathbb{R}^{768}$ is the final conversation embedding. A sigmoid yields the jailbreak probability

$$p(\mathcal{C}) \;=\; \sigma(z). \tag{8}$$

Objective and inference. The classifier outputs $p = \sigma(z)$ and is trained with the BCE objective in Section 3. At inference, scores are thresholded at $\tau = 0.5$ unless otherwise stated.

5 DATASETS

Training uses 98,626 multi-turn conversations, and evaluation uses a held-out 14,038-conversation benchmark (Table 1). Safe examples combine public benign dialogue corpora, namely UltraChat (Ding et al., 2023), WildChat (Zhao et al., 2024), ShareGPT (ShareGPT Community, 2023), and OASST2 (Köpf et al., 2023), used as released without additional safety filtering, with generated hard negatives that resemble security or refusal-adjacent interactions but lack adversarial intent. Training jailbreaks cover eight attack-strategy categories: framing and disguise (24,392), escalating manipulation (9,839), gradual escalation (4,986), refusal persistence (4,657), task decomposition (2,095), sudden adversarial insertion (903), context narrowing (253), and prompt injection/other (1,214).

The benchmark combines external corpora, including XGuard (Abishethvarman et al., 2025), Red Queen (Jiang et al., 2025), SafeDialBench (Cao et al., 2025), FRACTURED-SORRY-Bench (Priyanshu & Vijay, 2024), and LAD (Kulkarni, 2026), with synthetic conversations targeting underrepresented attacks (gradual escalation, sudden insertion, and context narrowing) and structurally attack-like hard negatives. To limit train–test leakage, it is filtered against the training set with a

Table 1: Dataset summary. Benchmark conversations have 2–102 turns (mean 9.3, median 8) and about 1,962 tokens on average.

Split	Conversations	Safe	Jailbreak
Training	98,626	50,287	48,339
Evaluation benchmark	14,038	8,182	5,856

Table 2: Performance comparison on the full evaluation benchmark (14,038 conversations). τ denotes the classification threshold. Best results in **bold**.

Model	Recall	Precision	Positive-class F1	FPR
Concatenation	36.20%	88.50%	0.5136	3.40%
ToT OR	47.10%	85.30%	0.6072	5.80%
ToT Majority	37.00%	89.10%	0.5226	3.20%
Claude Sonnet 4	59.10%	91.70%	0.7189	3.80%
Claude Sonnet 4.5	66.10%	91.70%	0.7682	4.30%
Claude Opus 4.7	78.30%	90.60%	0.8400	5.80%
Our Model ($\tau = 0.50$)	**92.10%**	**95.90%**	**0.9394**	**2.80%**

five-rule protocol: exact whole-conversation match, any matching prefix or suffix of at least two turns, and matching first or final user-turn hashes. The benchmark is intended to be out-of-distribution relative to the training mixture; results should therefore be read as evidence of robustness under the tested domain shifts rather than universal coverage. Detailed provenance and per-source counts are in Appendix A.4 and Appendix B.

6 Experimental Setup

Model and training. The primary model uses Multilingual E5-base (Wang et al., 2022) (278M parameters) as the TurnEncoder and a 4-layer ConvTransformer; conversations are truncated to the most recent 24 turns of at most 256 tokens each. Training has two stages: the ConvTransformer is first trained for 2 epochs with a frozen TurnEncoder (learning rate 10^{-4}), after which both components are fine-tuned jointly for 4 epochs (learning rate 5×10^{-6} for the TurnEncoder and 10^{-4} for the ConvTransformer).

Baselines. *Mean Pool* keeps the same TurnEncoder but replaces the ConvTransformer with mean pooling over turn embeddings and a linear classifier. *Concatenation* prompts Llama 3.3 70B (Dubey et al., 2024) with the most recent turns concatenated up to a window of 4. *ToT OR* and *ToT Majority* prompt the same Llama model to assess the full conversation through three lenses (user intent, potential harm, and jailbreak patterns), predicting a jailbreak if any lens (OR) or at least two lenses (Majority) are positive. We also evaluate the proprietary Claude Sonnet 4, Sonnet 4.5, and Opus 4.7 models (Anthropic, 2026). All baselines except Mean Pool are prompted rather than fine-tuned (see Section 9).

Metrics. All systems are evaluated on the same benchmark, with our model thresholded at $\tau = 0.5$. Precision, recall, and F1 refer to the positive (jailbreak) class, and FPR is the fraction of safe conversations flagged as jailbreak. Full configurations are in Appendix A.5.

7 Main Results

Table 2 reports performance on the full evaluation benchmark (14,038 conversations). Our model substantially outperforms all baselines across every metric.

Compared to Claude Opus 4.7—the strongest competing system—our model improves recall by $+13.8$ points (92.10% vs. 78.30%), precision by $+5.3$ points (95.90% vs. 90.60%), and positive-class F1 by $+0.0994$ (0.9394 vs. 0.8400), while simultaneously reducing the false-positive rate from 5.80% to 2.80%. The advantage over the Llama 3.3 70B–based baselines (Concatenation, ToT OR, ToT Majority) is even more pronounced: our model improves F1 by 0.33–0.43 absolute and substantially lowers FPR. Notably, the ToT OR variant trades a higher false-positive rate (5.80%) for moderate recall (47.10%), yet still falls short of our model by 45 recall points—indicating that simply expanding the decision surface of a prompted classifier cannot compensate for the lack of conversation-level reasoning.

These results demonstrate that our hierarchical ConvTransformer generalizes well within the evaluated out-of-distribution benchmark, outperforming even the most capable proprietary models while maintaining the lowest false-positive rate among all systems evaluated.

8 Ablations

ConvTransformer depth. We vary the number of ConvTransformer decoder layers from $L = 1$ to $L = 5$ to study the effect of model depth on performance. All variants use the same Multilingual E5-base TurnEncoder and training procedure. As shown in Table 3, increasing depth from $L = 1$ to $L = 4$ yields the best overall performance, while deeper models provide diminishing returns.

$L = 4$ achieves the highest AUC-ROC (98.66%), accuracy (95.04%), and jailbreak recall (92.06%), while $L = 3$ attains the best jailbreak precision (96.18%) and FPR (2.55%). The advantage over $L = 1$ is meaningful: $+1.7$ percentage points (pp) JB recall and $+0.8$ pp accuracy, demonstrating that deeper models better capture the multi-step reasoning required for detecting subtle escalation patterns.

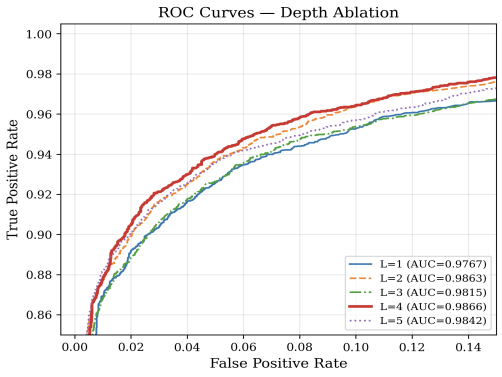
$L = 3$ shows a slight dip relative to $L = 2$ (89.81% vs. 91.27% recall), suggesting an intermediate-depth instability. $L = 5$ marginally underperforms $L = 4$, indicating diminishing returns beyond 4 layers for this task and dataset size.

All depths maintain FPR between 2.55% and 2.98%. The deeper models' recall gains do not come at the cost of additional false positives.

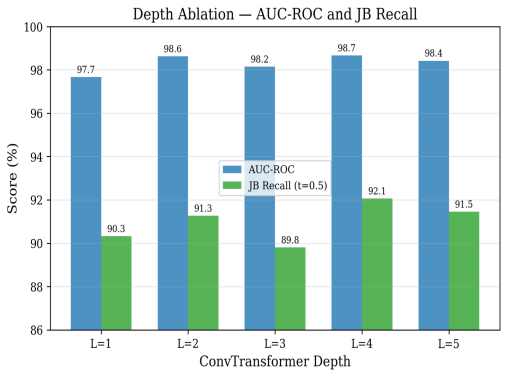
Figure 2 visualizes the impact of model depth on classification quality. Figure 2a shows that all ConvTransformer variants achieve strong ROC performance, with the deeper models generally tracing curves closer to the upper-left corner, indicating better discrimination. Figure 2b further highlights the depth-performance trade-off: performance improves from $L = 1$ to $L = 4$, where both AUC-ROC

Table 3: ConvTransformer depth ablation on the evaluation benchmark (14,038 conversations). All models were trained on 98.6k data. Values are percentages ($\tau = 0.5$). F1 Macro is the unweighted mean of the per-class F1 scores.

Layers	AUC-ROC	Acc	F1 Macro	JB Rec	JB Prec	FPR
$L = 1$	97.67	94.23	94.02	90.33	95.59	2.98
$L = 2$	98.63	94.76	94.57	91.27	95.96	2.75
$L = 3$	98.15	94.26	94.04	89.81	96.18	2.55
$L = 4$	98.66	95.04	94.87	92.06	95.89	2.82
$L = 5$	98.42	94.73	94.54	91.46	95.71	2.93



(a) ROC curves by ConvTransformer depth ($L = 1$ to $L = 5$).



(b) AUC-ROC and JB Recall by ConvTransformer depth ($L = 1$ to $L = 5$).

Figure 2: Performance trends across ConvTransformer depths.

Table 4: Attention component ablation on the evaluation benchmark. $L = 4$ ConvTransformer, trained on 98.6k data. Values are percentages ($\tau = 0.5$).

Variant	AUC-ROC	Acc	JB Rec	JB Prec	FPR
Full (SA+CA)	98.66	95.04	92.06	95.89	2.82
Self-Attn Only	98.02	93.18	90.76	92.74	5.08
Cross-Attn Only	96.70	89.22	78.30	94.99	2.96
Mean Pool	96.02	87.11	75.56	92.13	4.62

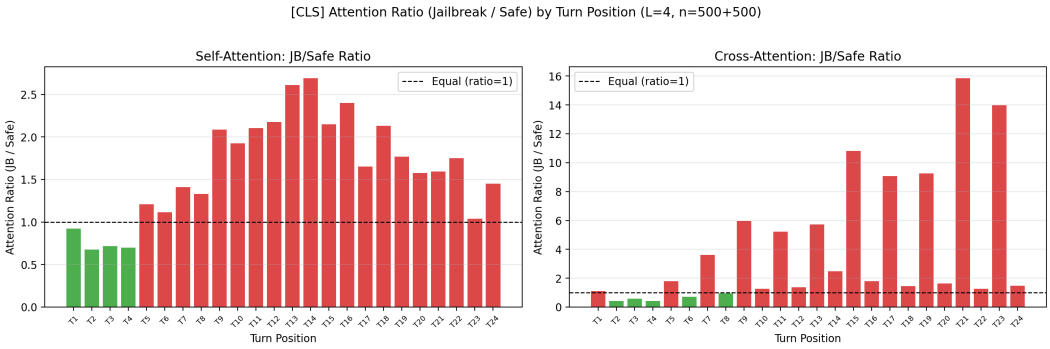


Figure 3: JB/Safe ratio of `[CONV_CLS]` attention by turn position for the ConvTransformer decoder (L=4, n=500+500 conversations). Left: self-attention over turn summaries; right: cross-attention over token memories. Ratios above 1 indicate positions that receive more attention in jailbreak than in safe conversations; the most extreme values occur in cross-attention at late odd-numbered user turns, where adversarial payloads typically appear. Raw attention weights are shown in Appendix Figure 5.

and jailbreak recall reach their strongest values, while $L = 5$ shows a slight decline, suggesting diminishing returns beyond four layers.

Attention components. We ablate the attention components of the $L = 4$ ConvTransformer by selectively disabling self-attention or cross-attention. This reveals the distinct contributions of each mechanism at the chosen model depth. Quantitative results are shown in Table 4, and ROC comparisons are shown in Appendix Figure 4.

At $L = 4$ depth, the attention components show strongly differentiated roles. As shown in Table 4, self-attention captures inter-turn dynamics and achieves competitive recall (90.76%) but at nearly double the false positive rate (5.08%). It detects multi-turn escalation patterns by modeling how turns relate to each other, but this sensitivity to structural patterns also triggers on benign instructional conversations.

Cross-attention alone degrades dramatically, reaching 78.30% recall, a drop of 13.8 percentage points relative to the full model in Table 4. Without self-attention to provide conversation-level context, cross-attention's token-level content matching cannot recover the distributed signal of gradual escalation attacks. However, it maintains a low FPR (2.96%), suggesting that its errors are primarily missed detections rather than false alarms.

Combining the two mechanisms yields the best trade-off: relative to self-attention only, the full model improves recall by 1.3 percentage points (92.06% vs. 90.76%) while reducing FPR by 2.26 points (2.82% vs. 5.08%).

To further isolate the contribution of attention-based conversation-level aggregation, we additionally compare against a non-attentive baseline in which the ConvTransformer cross-turn encoder is replaced with simple mean pooling over turn embeddings, followed by a linear classifier. Both variants share the same Multilingual E5-base TurnEncoder, so any performance difference is attributable to the aggregation mechanism rather than to the turn-level representation. As shown in Table 4, the full model substantially outperforms this Mean Pool baseline across all metrics, improving AUC-ROC

by 2.64 percentage points (98.66% vs. 96.02%), accuracy by 7.93 points (95.04% vs. 87.11%), and jailbreak recall by 16.50 points (92.06% vs. 75.56%), while simultaneously reducing FPR from 4.62% to 2.82%. The large recall gap is particularly informative: mean pooling treats all turns equally and dilutes localized adversarial signals, whereas the attention-based ConvTransformer can selectively emphasize the turns and tokens most indicative of escalation. These results indicate that attention-based aggregation, rather than turn-level encoding alone, is the primary driver of the hierarchical model's effectiveness on multi-turn jailbreak detection.

Attention analysis: jailbreak vs. safe conversations. We extract attention weights from the $L{=}4$ ConvTransformer decoder on 1,000 held-out conversations (500 jailbreak and 500 safe, each with at least four turns). The visualization focuses on the attention assigned by `[CONV_CLS]` to turn positions, separately for self-attention over turn summaries and cross-attention over token memories.

Figure 3 and the raw weights in Appendix Figure 5 show that safe conversations concentrate more attention on early context, whereas jailbreak conversations allocate relatively more mass to later turns. The effect is strongest in cross-attention: the JB/Safe ratio peaks at late odd-numbered user turns, consistent with multi-turn attacks that build a benign prefix before delivering the adversarial payload. Because safe and jailbreak conversations can differ in length, later turn positions are averaged over different subsets of conversations, so we treat these plots as qualitative diagnostics rather than causal explanations.

9 Limitations

Our detector is fine-tuned on in-domain data, whereas the Llama and Claude baselines are prompted; the gaps in Table 2 therefore reflect both the architecture and task-specific supervision, and a long-context classifier (e.g., Longformer) fine-tuned on the same data remains an important control. We also do not compare against activation-probing detectors such as LAD (Kulkarni, 2026), which require white-box access to the target model. The Concatenation baseline sees only a 4-turn window. Our efficiency argument is analytical, and we do not yet report measured latency, memory, or throughput. All results are single-run point estimates, so small differences, such as those between adjacent depths in Table 3, may not be statistically robust. The open-domain safe corpora are used without additional safety filtering, so some safe labels may be noisy. The 256-token turn cap and 24-turn window are also an evasion surface: an adaptive attacker could place a payload beyond the token cap or push early framing out of the window. Neither cap is an architectural limit; long turns could be chunked with overlapping windows and aggregated into a turn summary by an additional ConvTransformer-style level, and more turns can be retained at low cost, but we have not evaluated these extensions. Finally, the evaluation is offline over supplied conversation prefixes rather than a measured streaming deployment, and source distributions may not cover every future adaptive attack, so domain-shift conclusions should be read as qualified evidence rather than exhaustive coverage. More details are in Appendix A.6.

10 Conclusion

We introduced a conversation-level jailbreak detector that assigns each dialogue a jailbreak likelihood and enables straightforward deployment via a single decision threshold. Across a challenging out-of-distribution evaluation, our approach consistently outperformed a broad set of baselines, including Claude Opus 4.7, demonstrating strong generalization within the evaluated domain-shift setting. The proposed hierarchical design reduces the attention cost of long-context concatenation by encoding turns independently and performing lightweight conversation-level reasoning. It also supports selectively attending to fine-grained token evidence when needed, improving detection of multi-turn escalation and reframing behaviors.

Future work will focus on improving robustness to adaptive jailbreak strategies, refining calibration and threshold selection across domains, and further reducing false positives without sacrificing detection coverage.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic multi-turn conversations used in the dataset construction process described in Appendix A.4; supporting data analysis and interpretation of experimental results; summarizing existing literature; and editing, polishing, and organizing the manuscript text. The Claude and Llama models evaluated in Section 6 are baselines under study rather than tools used to produce this work. We did not use generative AI tools to generate the paper's core research ideas, final experimental results, or peer reviews. All AI-assisted outputs were reviewed by the authors, including manual inspection and filtering of generated conversations, verification of data-analysis outputs against the underlying experimental results, and author review of manuscript edits. The authors take full responsibility for the final content of this work, including the text, claims, data artifacts, analyses, and conclusions.

ETHICS STATEMENT

This work targets a defensive goal: detecting multi-turn jailbreak attempts to strengthen content moderation. We describe attack *categories* (e.g., gradual escalation, context narrowing) only at the level needed to motivate the detector's design, and we do not release novel attack techniques, optimized adversarial prompts, or step-by-step harmful content. The synthetic adversarial conversations used for training and evaluation carry dual-use risk and are not publicly released. External corpora are used under their original licenses. Because a deployed detector can produce both false negatives (missed attacks) and false positives (over-blocking benign users), we report false-positive rate alongside recall and caution against using the model as a sole safety control.

REFERENCES

Vigneswar Abishethvarman, Usman Naseem, et al. XGUARD: A graded benchmark for evaluating safety failures of large language models on extremist content. *arXiv preprint arXiv:2506.00973*, 2025.

Christopher M. Ackerman and Nina Panickssery. Mitigating many-shot jailbreaking, 2025. arXiv preprint arXiv:2504.09604.

Ahmad Alobaid, Martí Jordà Roca, Carlos Castillo, and Joan Vendrell. The echo chamber multi-turn LLM jailbreak, 2026. arXiv preprint arXiv:2601.05742.

Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, et al. Many-shot jailbreaking. In *Advances in Neural Information Processing Systems*, 2024.

Anthropic. Claude: Model card and evaluations. `https://docs.anthropic.com/en/docs/about-claude/models`, 2026. Claude Sonnet 4, Sonnet 4.5, and Opus 4.7 model family.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. arXiv preprint.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Proceedings of the World Wide Web Conference (WWW)*, 2019.

Hongyu Cao, Yuyang Wang, Shuo Jing, et al. SafeDialBench: A fine-grained safety evaluation benchmark for large language models in multi-turn dialogues with diverse jailbreak attacks. *arXiv preprint arXiv:2502.11090*, 2025.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *Advances in Neural Information Processing Systems*, 2023.

Krzysztof Choromanski et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Abhimanyu Dubey, Akhil Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Tom Gibbs, Ethan Kosak-Hine, George Ingebretsen, Jason Zhang, Julius Broomfield, Sara Pieri, Reza Iranmanesh, Reihaneh Rabbany, and Kellin Pelrine. Emerging vulnerabilities in frontier models: Multi-turn jailbreak attacks. *arXiv preprint arXiv:2409.00137*, 2024.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2023.

Laura Hanu and Unitary team. Detoxify. Software (GitHub repository), 2020.

Yifan Jiang, Kriti Aggarwal, Tanusree Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. Red queen: Exposing latent multi-turn risks in large language models. In *Findings of the Association for Computational Linguistics (ACL)*, 2025. Also available as arXiv:2409.17458.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, ES Shahul, Sameer Suri, David Glushkov, Arnav Varma Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. OpenAssistant conversations – democratizing large language model alignment. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Prashant Kulkarni. Latent adversarial detection: Adaptive probing of LLM activations for multi-turn attack detection, 2026. arXiv preprint arXiv:2604.28129.

Adarsh Kumarappan and Ananya Mujoo. Automating deception: Scalable multi-turn LLM jailbreaks, 2025. arXiv preprint.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.04451.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Sidhant Narula, Javad Rafiei Asl, Mohammad Ghasemigol, Eduardo Blanco, and Daniel Takabi. HarmNet: A framework for adaptive multi-turn jailbreak attacks on large language models, 2025. arXiv preprint.

Aman Priyanshu and Supriti Vijay. FRACTURED-SORRY-Bench: Framework for revealing attacks in conversational turns undermining refusal efficacy and defenses over SORRY-Bench. *arXiv preprint arXiv:2408.16163*, 2024.

Jack W. Rae et al. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. Derail yourself: Multi-turn LLM jailbreak attack through self-discovered clues. *arXiv preprint arXiv:2410.10700*, 2024.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo multi-turn LLM jailbreak attack, 2024. arXiv preprint.

ShareGPT Community. ShareGPT: Shared ChatGPT conversations dataset. `https://sharegpt.com`, 2023. Community-collected dataset.

Jialin Song, Xiaodong Liu, Weiwei Yang, Wuyang Chen, Mingqian Feng, Xuekai Zhu, and Jianfeng Gao. MultiBreak: A scalable and diverse multi-turn jailbreak benchmark for evaluating LLM safety. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2605.01687.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. arXiv preprint.

Xianjun Yang, Liqiang Xiao, Shiyang Li, Faisal Ladhak, Hyokun Yun, Linda Ruth Petzold, Yi Xu, and William Yang Wang. Many-turn jailbreaking, 2025. arXiv preprint.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, 2016.

Erxin Yu, Jing Li, Ming Liao, Siqi Wang, Zuchen Gao, Fei Mi, and Lanqing Hong. CoSafe: Evaluating large language model safety in multi-turn dialogue coreference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17494–17508, 2024. doi: 10.18653/v1/2024.emnlp-main.968.

Manzil Zaheer et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of ACL 2024*, 2024.

Jiahao Zhang, Michael Kuo, Yuheng Chen, and Hai Li. Malicious-educator: A benchmark for stress-testing AI safety protocols. *arXiv preprint arXiv:2502.12836*, 2025. Duke Center for Computational Evolutionary Intelligence.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT interaction logs in the wild. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. In *Advances in Neural Information Processing Systems*, 2023.

A Supplementary Details

A.1 Detailed Related Work

Jailbreak / Prompt-Injection Detection and Moderation Classifiers Safety moderation for LLM systems is often implemented using supervised classifiers trained to detect toxic or policy-violating content, including widely used toxicity benchmarks and models (e.g., Jigsaw/Conversation AI datasets; (Borkan et al., 2019) and Detoxify; (Hanu & team, 2020)). While such classifiers are effective for overtly harmful content, jailbreaks frequently rely on instruction-following failures rather than explicit toxicity, motivating research on adversarial prompting and "guard" models that predict policy-violation risk.

A complementary line of work studies attack methods that generate jailbreak prompts through optimization or search. Universal adversarial suffix attacks such as GCG (Zou et al., 2023) show that iterative optimization can yield transferable jailbreak triggers across prompts and models. AutoDAN

(Liu et al., 2024) similarly generates stealthy jailbreak prompts via iterative optimization/evolution strategies. These approaches are important baselines and threat models for detection because they can produce non-obvious, non-toxic surface forms that nonetheless induce unsafe behavior.

Multi-Turn Safety / Dialogue Risk Detection Multi-turn jailbreaks introduce temporal structure: an adversary can begin benignly and gradually move the model toward disallowed behavior through escalation, reframing, or role manipulation (Russinovich et al., 2024; Chao et al., 2023; Mehrotra et al., 2024). Several recent attacks explicitly operationalize this multi-step structure. Crescendo (Russinovich et al., 2024) demonstrates a benign-to-malicious escalation strategy that unfolds across turns and can evade turn-local defenses. PAIR (Chao et al., 2023) jailbreaks black-box LLMs through iterative attacker–target refinement, using feedback from previous responses to craft subsequent queries. Tree of Attacks (TAP) (Mehrotra et al., 2024) extends iterative refinement into a tree search, branching attack trajectories and selecting promising continuations.

Long-context multi-turn threats also appear in many-shot jailbreaking (Anil et al., 2024), where a long conversation history (often with many demonstrations) biases the model toward compliance. Defense-oriented follow-ups such as *Mitigating Many-Shot Jailbreaking* (Ackerman & Panickssery, 2025) are relevant when positioning detection in the broader mitigation landscape (detectors, training-time alignment, inference-time policies).

On the benchmarking side, multi-turn evaluation suites such as MTJ-Bench / Multi-Turn Jailbreak Benchmark (Yang et al., 2025) reflect growing recognition that single-turn benchmarks underestimate real-world risk, and they provide more realistic settings for measuring conversation-level detection and early-warning capabilities.

Finally, newer work continues to push adaptive multi-turn attack automation at scale, including HarmNet (Narula et al., 2025), *Automating Deception* (Kumarappan & Mujoo, 2025), and *Echo Chamber* (Alobaid et al., 2026). Collectively, these works strengthen the case that robust defenses must (i) model evolving attacker intent across turns and (ii) remain effective under adaptive adversaries (Narula et al., 2025; Kumarappan & Mujoo, 2025; Alobaid et al., 2026).

Indirect Prompt Injection and Tool-/Agent-Integrated Settings Beyond direct user prompting, indirect prompt injection arises when models ingest untrusted retrieved or tool-provided content that contains adversarial instructions (Greshake et al., 2023; Zhan et al., 2024). *Not What You've Signed Up For* (Greshake et al., 2023) highlights real-world compromise vectors in LLM-integrated applications via malicious content embedded in retrieved documents or other inputs, often manifesting as multi-step interactions in agentic or retrieval-augmented pipelines. More recently, InjecAgent (Zhan et al., 2024) proposes threat models and benchmarking for indirect prompt injections in tool-integrated LLM agents. These settings reinforce the importance of multi-turn and multi-source context modeling: the effective "conversation" may include tool outputs, retrieved passages, and intermediate reasoning steps, all of which can participate in a gradual jailbreak (Greshake et al., 2023; Zhan et al., 2024).

Hierarchical Modeling for Long Documents and Dialogue Hierarchical architectures are a standard approach for modeling structured long inputs by separating local encoding from global aggregation. Classic examples include Hierarchical Attention Networks (Yang et al., 2016), which encode tokens into sentence representations and then apply higher-level attention over sentences for document classification. Related hierarchical dialogue modeling traditions similarly encode utterances/turns independently and then model the sequence of turns to capture conversational dynamics. This perspective aligns closely with multi-turn jailbreak detection: the salient signal is often how turns relate (escalation, contradiction, instruction manipulation), rather than token-level interactions uniformly across a flattened sequence. Our approach fits this family by combining turn-wise representations with conversation-level attention, while still retaining access to token-level evidence through cross-attention.

Efficient Transformers and Long-Context Modeling (Not Just "Long Context") A separate body of work aims to scale Transformers to long sequences by reducing attention cost or adding memory mechanisms, including sparse attention (Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020)), hashing or approximation methods (Reformer (Kitaev et al., 2020), Linformer (Wang et al., 2020), Performer (Choromanski et al., 2021)), and recurrence/memory (Transformer-XL (Dai

et al., 2019), Compressive Transformer (Rae et al., 2020)). These methods primarily treat the input as a single long token stream.

In contrast, many jailbreak settings are naturally structured by turns (and sometimes by tool calls / retrieved documents). Multi-turn attacks such as Crescendo (Russinovich et al., 2024), PAIR (Chao et al., 2023), TAP (Mehrotra et al., 2024), and many-shot jailbreaking (Anil et al., 2024) exploit exactly this conversational structure—e.g., building a benign prefix, iteratively refining instructions, or using long interaction histories. This motivates architectures that allocate computation to turn-level attention while selectively revisiting token-level detail, rather than relying solely on generic long-context Transformers.

A.2 Detailed Problem Formulation

Conversation and Turn Notation We model a conversation as a sequence of discrete turns:

$$\mathcal{C} = (x_1, x_2, \ldots, x_T), \tag{9}$$

where T is the number of turns and each turn x_t is a role-annotated message

$$x_t = (r_t, u_t), \tag{10}$$

with role $r_t \in \{\text{user}, \text{assistant}\}$ and textual content u_t. When role tags are not needed, we refer to the content-only sequence $(u_1, u_2, \ldots, u_T)$. We denote the prefix of the conversation up to turn t as

$$\mathcal{C}_{\leq t} = (x_1, x_2, \ldots, x_t). \tag{11}$$

Labels and Tasks We use a binary conversation-level label $y \in \{0, 1\}$, where $y = 0$ indicates a safe conversation and $y = 1$ denotes a jailbreak conversation. Note that datasets differ in whether $y = 1$ reflects jailbreak *attempts* (malicious user intent is present at some point) or only *successful* jailbreaks (the assistant produces disallowed content); we follow the definition specified by each data source; training labels are success-based unless a source explicitly labels attempts (Appendix A.4).

Conversation-level classification. Given the full conversation $\mathcal{C}$, the primary task is to estimate the posterior

$$p(\mathcal{C}) \;=\; \Pr(y = 1 \mid \mathcal{C}), \tag{12}$$

and produce a hard decision by thresholding,

$$\hat{y} \;=\; \mathbf{1}\{\, p(\mathcal{C}) \geq \tau \,\}, \tag{13}$$

where $\tau \in [0, 1]$ is an operating threshold and $\mathbf{1}\{\cdot\}$ denotes the indicator function (equal to 1 if its argument is true and 0 otherwise).

Prefix classification and deployment mode. The experiments in this paper evaluate an offline/prefix classifier: for each example, the model scores an observed conversation prefix and produces a single conversation-level decision. Although the same architecture can be applied repeatedly to prefixes $\mathcal{C}_{\leq t}$ to obtain per-turn risk scores, the ConvTransformer self-attention used here is bidirectional over the supplied prefix rather than a causal streaming mechanism. Consequently, we do not claim an evaluated early-detection objective in this work; measuring alert time and streaming latency under incremental updates is left to future work.

Threat Model We assume an adversarial user who controls the user turns and can adaptively choose messages based on prior assistant outputs. The attacker is described by an adaptive policy π such that, for each user-role turn,

$$u_t \sim \pi(\cdot \mid \mathcal{C}_{\leq t-1}). \tag{14}$$

Assistant-role turns are produced by the target model (or an environment/tool wrapper) and are treated as exogenous to the defender.

The defender observes the available conversation prefix and produces a risk score

$$g(\mathcal{C}_{\leq t}) \;=\; p_t, \tag{15}$$

from which a hard decision is derived by thresholding. In the experiments below, this decision is made for the provided prefix/full conversation rather than evaluated as a causal streaming alert. The defender does not control the attacker and must remain robust to multi-turn escalation strategies and other adaptive behaviors expressed across turns.

A.3 METHOD DETAILS

Baseline and Complexity Motivation

Naive concatenation baseline. A straightforward approach concatenates all turns into a single long token sequence and runs a Transformer encoder over the full conversation. Let turn t contain n_t tokens. The total concatenated length is

$$N \;=\; \sum_{t=1}^{T} n_t. \tag{16}$$

For standard Transformer self-attention, the dominant cost scales quadratically with sequence length:

$$\text{cost} \;=\; \mathcal{O}(N^2) \quad \text{(up to constant factors and hidden-size terms)}. \tag{17}$$

Concrete example ($24 \times 256 \to 6144$ tokens). With up to $T = 24$ turns and a per-turn cap of $n_t \le 256$ tokens,

$$N \;=\; 24 \cdot 256 \;=\; 6144, \tag{18}$$

and attention scales as

$$N^2 \;=\; 6144^2 \;=\; 37{,}748{,}736 \tag{19}$$

pairwise interactions per layer (up to constant factors and number of heads). This quadratic scaling motivates an architecture that avoids attention over the full concatenated context while preserving cross-turn reasoning for high-throughput, low-latency moderation. For a hierarchical design with per-turn cap L and T turns, the dominant attention cost is approximately $\mathcal{O}(TL^2 + T^2 + T^2L)$ when cross-attention from T turn summaries to TL token memories is used, rather than $\mathcal{O}((TL)^2)$ for full concatenation; it also imposes an inductive bias that first models local turn semantics and then dialogue-level structure.

Level 1: TurnEncoder

Overview. As shown in Figure 1, Level 1 encodes each turn independently using a pre-trained language model, producing (i) a compact summary embedding for cross-turn reasoning and (ii) token-level embeddings for optional fine-grained retrieval by Level 2. Encoding turns independently keeps computation roughly linear in the number of turns and enables efficient batching.

Backbone and role markers. The TurnEncoder wraps the pretrained multilingual E5-base model (`intfloat/multilingual-e5-base`), a 12-layer Transformer with hidden size $d = 768$. We add two special tokens to the vocabulary, `[USER]` and `[ASST]`, which act as role markers; their embeddings are frozen together with the rest of the TurnEncoder during Stage 1 and are updated only during Stage 2 joint fine-tuning. Each turn is tokenized independently as

$$\tilde{x}_t \;=\; [\texttt{ROLE}(r_t)] \,\|\, u_t, \tag{20}$$

where $\|$ denotes token-sequence concatenation, then padded or truncated to a fixed maximum length $L = 256$ tokens.

Outputs. For each turn t, the encoder outputs:

- **Turn summary embedding (CLS embedding):**

$$\mathbf{s}_t \in \mathbb{R}^d, \qquad \mathbf{s}_t \;=\; \mathbf{h}_{t,1}^{(L_{\text{enc}})}, \tag{21}$$

  defined as the first token's output (CLS) from the last encoder layer ($L_{\text{enc}} = 12$).

- **Token-level embeddings (last-layer hidden states):**

$$\mathbf{H}_t \in \mathbb{R}^{L \times d}, \qquad \mathbf{H}_t \;=\; \big[\mathbf{h}_{t,1}^{(L_{\text{enc}})}, \ldots, \mathbf{h}_{t,L}^{(L_{\text{enc}})}\big]^\top. \tag{22}$$

The summaries $\{\mathbf{s}_t\}_{t=1}^T$ feed Level 2 self-attention, while $\{\mathbf{H}_t\}_{t=1}^T$ serve as a cross-attention memory for "zooming in" on salient phrases.

Independence across turns. There is no cross-turn token sharing at Level 1; all cross-turn reasoning (e.g., recognizing escalation from turn 3 to turn 5) is deferred to Level 2. This independence also enables encoding all turns in a batch in one batched forward pass.

Memory optimization. Gradient checkpointing is enabled during training to reduce memory usage, which is important when processing up to 24 turns per conversation per GPU. Table 5 summarizes the TurnEncoder configuration.

Table 5: TurnEncoder configuration.

Parameter	Value
Base model	`intfloat/multilingual-e5-base` (12 layers, 768-dim)
Special tokens	`[USER]`, `[ASST]`
Max tokens per turn	256
Dropout	0.2
Gradient checkpointing	Enabled

Level 2: ConvTransformer

Overview. Level 2 performs conversation-level reasoning over the sequence of turn summaries and optionally retrieves token-level details through cross-attention. This captures multi-turn patterns such as gradual escalation, topic shifts, instruction hierarchy manipulation, and role changes.

A.3.1 Input sequence construction

Given turn summaries $\{\mathbf{s}_t\}_{t=1}^{T}$, we prepend a learnable conversation-level token $\mathbf{s}_0$ corresponding to `[CONV_CLS]`:

$$\mathbf{S}^{(0)} \;=\; [\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_T] \in \mathbb{R}^{(T+1)\times d}. \tag{23}$$

The final representation of `[CONV_CLS]` serves as the primary conversation embedding.

A.3.2 Position and role embeddings

Each element in the input sequence is augmented with learned position and role embeddings:

$$\tilde{\mathbf{s}}_i \;=\; \mathbf{s}_i + \mathbf{p}_i + \mathbf{e}_{\rho_i}, \qquad i = 0, \ldots, T, \tag{24}$$

where $\mathbf{p}_i$ is a learned position embedding and $\mathbf{e}_{\rho_i}$ is a learned role embedding; these position, role, and `[CONV_CLS]` parameters are trainable during Stage 1. We use learned position embeddings for 25 positions (position 0 for `[CONV_CLS]`, positions 1–24 for turns), and learned role embeddings for 3 roles (user, assistant, and a CLS placeholder role).

A.3.3 Self-attention over turn summaries

The ConvTransformer is a 4-layer Transformer decoder (pre-layer normalization, 8 heads, Feed-Forward Network (FFN) hidden size 2048, dropout 0.2). It applies bidirectional self-attention (no causal mask) over the augmented summary sequence

$$\tilde{\mathbf{S}}^{(0)} \;=\; [\tilde{\mathbf{s}}_0, \tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_T], \tag{25}$$

because the classifier can observe the full conversation at inference time in the current setting. This supports detecting global multi-turn structures (e.g., returning to a previously refused topic several turns later).

A.3.4 Cross-attention to token-level memory

In addition to self-attention over summaries, each decoder layer performs cross-attention over token-level embeddings from Level 1. We flatten all per-turn token embeddings into one memory:

$$\mathbf{M} \;=\; [\mathbf{H}_1; \mathbf{H}_2; \ldots; \mathbf{H}_T] \in \mathbb{R}^{(T\cdot L)\times d}, \tag{26}$$

where $L = 256$ and thus $|\mathbf{M}| = T \cdot L$ tokens (with $T \leq 24$). Tokens in $\mathbf{M}$ are augmented with turn-aware positional signals (reusing the same turn position table) so attention heads can infer which turn each token belongs to. A padding mask is passed to cross-attention so padded token positions introduced by the 256-token turn cap are excluded from the key/value memory. This mechanism allows Level 2 to "zoom in" on specific words or phrases that may carry jailbreak signal.

A.3.5 ATTENTIVE POOLING

After the decoder layers, we compute an additional conversation summary via attentive pooling: a learned query attends over all turn outputs to produce a weighted combination, which is then added to the `[CONV_CLS]` output via a residual connection. This lets the model weight turns differently depending on their importance (e.g., later turns in an escalation attack). We list the conversation module configuration in Table 6.

Table 6: ConvTransformer configuration.

Parameter	Value
Decoder layers	4
Attention heads	8
FFN hidden dim	2,048
Normalization	Pre-LN
Position embeddings	Learned, 25 positions
Role embeddings	Learned, 3 roles
Dropout	0.2

A.3.6 CLASSIFICATION HEADS

The final 768-dimensional conversation embedding is passed through dropout and a linear projection to produce a single jailbreak logit:

$$z \;=\; \mathbf{w}^\top \mathbf{h}_{\text{conv}} + b \in \mathbb{R}, \tag{27}$$

where $\mathbf{h}_{\text{conv}} \in \mathbb{R}^{768}$ is the final conversation embedding. A sigmoid yields the jailbreak probability

$$p(\mathcal{C}) \;=\; \sigma(z). \tag{28}$$

A secondary turn-level classification head projects each turn's output embedding to a single logit (with a sigmoid) to support an auxiliary per-turn loss (multiple-instance learning). In the described experiment, the auxiliary loss weight is set to zero, so this head has no effect on training or inference (though logits may still be computed).

Training Objective and Inference

A.3.7 INPUT CONSTRAINTS

We cap conversation length by truncating at both the turn and conversation levels: we limit each turn to at most 256 tokens, and we limit each conversation to at most 24 turns (keeping only the most recent 24 turns when a conversation is longer). This cap is an engineering choice that supports fixed-size batching, not a claim that longer contexts are infeasible; Appendix A.6 discusses the resulting loss of evidence in very long conversations or turns, its implications for adaptive evasion, and possible extensions.

A.3.8 TRAINING OBJECTIVE

Let $y \in \{0, 1\}$ denote the conversation label (0=safe, 1=jailbreak). We compute a predicted jailbreak probability $p(\mathcal{C})$ and optimize binary cross-entropy:

$$\mathcal{L}_{\text{conv}} \;=\; -\,y \log p(\mathcal{C}) \;-\; (1-y) \log \big(1 - p(\mathcal{C})\big). \tag{29}$$

When turn-level labels are available, an auxiliary per-turn binary cross-entropy loss can be added to provide finer-grained supervision, though we do not activate it in the experiments presented herein.

A.3.9 INFERENCE

At inference time, we encode each turn independently with the TurnEncoder to obtain turn summaries and (optionally) token-level embeddings, then run the Level 2 conversation module over the resulting sequence. Following common moderation practice, the final assistant response is dropped during both training and inference so the decision is conditioned only on the conversation history up to the last user turn. The classifier outputs a jailbreak probability via a sigmoid, and we predict *jailbreak* if the probability exceeds a threshold τ (set on a validation set); turn-level logits are optional and are not used for the final decision.

A.4 DATASET PROVENANCE DETAILS

Training Set We train all models on a curated dataset of $98{,}626$ multi-turn conversations, approximately balanced between jailbreak (49%) and safe (51%) examples. A conversation is labeled *jailbreak* ($y = 1$) if any assistant turn contains harmful content elicited through adversarial manipulation, and *safe* ($y = 0$) otherwise. Under this success-based labeling, failed attacks in which the assistant consistently refuses are treated as safe/hard-negative examples unless the dataset source explicitly labels attempts; this distinction matters when comparing attempt detectors with success detectors.

Safe conversations. The safe subset ($50{,}287$ conversations) combines open-domain dialogues with purpose-built hard negatives. The open-domain portion combines real user–assistant logs (WildChat, ShareGPT, and OASST2) with synthetic dialogues (UltraChat); these corpora are used as released, without additional safety filtering, and teach the detector to tolerate normal multi-turn interaction patterns. Hard negatives are benign conversations that share structural features with attacks—such as progressive complexity, polite follow-ups, or extended length—and reduce false alarms on legitimate conversations.

Jailbreak conversations. The jailbreak subset ($48{,}339$ conversations) covers eight distinct multi-turn attack-strategy categories. These include framing and disguise attacks ($24{,}392$) that wrap harmful intent in persona, academic, historical, fictional, or hypothetical contexts; escalating manipulation ($9{,}839$) and gradual escalation ($4{,}986$), where successive turns become increasingly specific or harmful; refusal persistence ($4{,}657$), where the user repeatedly reframes a refused request until compliance; task decomposition ($2{,}095$), where a harmful objective is split into apparently innocuous sub-questions; sudden adversarial insertion (903), where a malicious turn is embedded late in an otherwise benign exchange; context narrowing (253), where a benign prefix is reformatted into harmful content; and prompt injection/other strategies ($1{,}214$), including instruction overrides, attempts to suppress safety guidance, and short context-shifting attacks. This taxonomy is intended to describe attack mechanisms rather than mutually exclusive semantic harms, and individual conversations may exhibit more than one tactic.

Evaluation Benchmark We evaluate on a comprehensive multi-source benchmark of $14{,}038$ conversations ($8{,}182$ safe and $5{,}856$ jailbreak) constructed to assess detector performance across a broad spectrum of attack strategies and benign interaction patterns. The benchmark combines established external corpora with targeted synthetic conversations, ensuring coverage of attack vectors that no single existing dataset provides in isolation. The full per-source composition is reported in Tables 7 and 8 (Appendix B).

Safe conversations. The $8{,}182$ safe conversations are drawn from five public dialogue corpora spanning open-domain discussion, task-oriented instruction, and user–assistant interaction. UltraChat (Ding et al., 2023) and WildChat (Zhao et al., 2024) contribute the bulk of safe examples, representing diverse multi-turn exchanges between users and LLMs. ShareGPT (ShareGPT Community, 2023) and OpenAssistant (OASST2) (Köpf et al., 2023) provide shorter community-collected dialogues, and SafeMTData (Ren et al., 2024) contributes benign conversations on sensitive-adjacent topics where user intent is genuinely benign. To calibrate false-positive rates, we additionally include synthetically generated hard negatives—benign conversations that share structural features with jailbreak attacks (progressive complexity, topic shifts, extended length) but contain no adversarial intent.

Jailbreak conversations. The 5,856 jailbreak conversations span eight distinct attack-strategy categories. External corpora provide ecologically valid attacks: XGuard (Abishethvarman et al., 2025) contributes persona-adoption red-teaming, Red Queen (Jiang et al., 2025) supplies short targeted manipulations, carl213 (Gibbs et al., 2024) provides cipher-encoded attacks via user-defined word-substitution mappings, and SafeDialBench (Cao et al., 2025) and the Duke CEI Malicious-Educator benchmark (Zhang et al., 2025) contribute additional multi-turn attack scenarios. FRACTURED-SORRY-Bench (Priyanshu & Vijay, 2024) contributes multi-step attacks with benign intermediary turns, and LAD (Kulkarni, 2026) contributes gradual escalation in technical domains (SCADA systems, autonomous vehicles, network security). To cover attack vectors underrepresented in existing corpora, we additionally include synthetic conversations implementing gradual escalation (progressive steering across 9–32 turns), sudden adversarial insertion (a single malicious turn embedded within an otherwise benign conversation), and context narrowing (benign prefixes reframed toward harmful requests via the final user turn). Prompt-injection attacks (2–12 turns) test robustness to single-shot instruction-override attempts within a multi-turn context.

Deduplication. To prevent train–test leakage, the benchmark is filtered against the training set using a strict five-rule protocol: (1) exact whole-conversation match; (2) any prefix of ≥ 2 turns matching; (3) any suffix of ≥ 2 turns matching; (4) first user-turn hash match; and (5) final user-turn hash match. This eliminates both verbatim repetition and template-level overlap that arises when synthetic generators draw from shared seed pools.

Rationale for combining external and synthetic data. No single jailbreak corpus covers the full range of known multi-turn attack strategies. External datasets provide ecological validity—these are attacks that were actually attempted against deployed systems—but they cluster around specific tactics (e.g., XGuard is predominantly persona-based; Red Queen is uniformly five turns). Synthetic augmentation addresses three gaps: (i) controlled variation of conversation length and escalation speed, enabling fine-grained evaluation of detector sensitivity; (ii) coverage of structurally distinct attack types (sudden insertion, context narrowing) that are rare in existing corpora; and (iii) hard negatives that share surface-level features with attacks, which are critical for measuring false-positive robustness, since naturally occurring safe conversations rarely resemble jailbreaks in structure.

Summary statistics. The benchmark comprises 14,038 conversations (58.3% safe, 41.7% jailbreak), with turn counts ranging from 2 to 102 (mean 9.3, median 8) and approximately 1,962 tokens per conversation on average. The 24 distinct source identifiers are grouped into eight attack-strategy categories and three safe-conversation categories, drawing on published external datasets and synthetic conversations.

A.5 Experimental Details

Experimental Setup

Model configuration. Our primary model uses Multilingual E5-base (Wang et al., 2022) (278M parameters) as the TurnEncoder with a 4-layer ($L = 4$) ConvTransformer decoder. Each turn is tokenized to a maximum of 256 tokens, and conversations are truncated to 24 turns (retaining the most recent turns when a conversation exceeds this limit). The ConvTransformer uses 8 attention heads with a feed-forward dimension of 2,048.

Training procedure. Training proceeds in two stages. In the first stage, we train the ConvTransformer with the entire TurnEncoder frozen (including the `[USER]`/`[ASST]` embeddings) for 2 epochs (learning rate 1×10^{-4}). In the second stage, we jointly fine-tune both components for 4 additional epochs, using a reduced learning rate of 5×10^{-6} for the TurnEncoder and 1×10^{-4} for the ConvTransformer. Joint fine-tuning enables the TurnEncoder and ConvTransformer to adapt to each other, improving overall end-to-end performance compared to keeping the encoder frozen.

Baselines. We compare against the following baselines:

- **Mean Pool.** Uses the same Multilingual E5-base TurnEncoder (Wang et al., 2022) but replaces the ConvTransformer with simple mean pooling over turn embeddings, followed by a linear classifier. This ablation isolates the contribution of the attention-based conversation-level aggregation.
- **Concatenation.** A Llama 3.3 70B model (Dubey et al., 2024) that concatenates the most recent turns up to a fixed window size of 4, then classifies the resulting truncated context.
- **ToT OR / ToT Majority.** Tree-of-Thought baselines built on the same Llama 3.3 70B model (Dubey et al., 2024). The full conversation is examined from three complementary analytical lenses: (i) *user intent*—inferring the user's true goal and whether it is manipulative; (ii) *potential harm*—assessing worst-case misuse and safety impact; and (iii) *jailbreak patterns*—detecting known tactics such as roleplay framing, hypotheticals, authority claims, or gradual escalation. Each lens produces an independent binary decision. **ToT OR** predicts jailbreak if *any* of the three lenses is positive; **ToT Majority** predicts jailbreak if at least two of the three are positive.
- **Claude Sonnet 4 / Sonnet 4.5 / Opus 4.7** (Anthropic, 2026). State-of-the-art proprietary models from Anthropic. Claude Opus 4.7 is Anthropic's flagship Opus-tier model. These represent strong, widely deployed safety-tuned baselines; outperforming them provides compelling evidence for the effectiveness of our approach.

A.6 Detailed Limitations

The model is evaluated as an offline/prefix classifier with bidirectional attention over the supplied conversation prefix; we do not evaluate causal streaming updates, alert latency, or early-detection time. The 24-turn conversation cap and 256-token per-turn cap enable efficient batching and predictable memory use, but they can discard evidence in very long dialogues or verbose turns. This limitation is especially relevant for many-shot or document-augmented attacks where harmful intent may be distributed across a longer history.

Truncation and adaptive evasion. The caps also create an evasion surface that we do not evaluate: an adaptive attacker could bury a payload beyond the first 256 tokens of a long turn behind benign padding, or extend the conversation so that the initial framing of a gradual escalation falls outside the 24 most recent turns. Neither cap is intrinsic to the architecture. The TurnEncoder backbone accepts up to 512 positions, so the per-turn cap can be raised directly, with cost that remains linear in the number of turns. For longer turns, the hierarchy can be extended by one level: a turn is split into overlapping windows, each window is encoded by the TurnEncoder, and a small ConvTransformer-style aggregator combines the window summaries into the turn summary $\mathbf{s}_t$, while the window token embeddings join the cross-attention memory $\mathbf{M}$. Encoding cost then grows linearly with the number of windows rather than quadratically with turn length. At the conversation level, self-attention over turn summaries is inexpensive ($\mathcal{O}(T^2)$ with small T), so more turns can be retained, and the earliest turns can be kept alongside the most recent ones so that the initial framing is not discarded. We have not implemented or evaluated these extensions, and robustness to truncation-aware adaptive attacks remains an open question.

Our evaluation combines external and synthetic sources and includes deduplication against the training set, but it remains a finite benchmark. Results should therefore be interpreted as evidence of robustness within the evaluated distribution shifts rather than proof of universal domain generalization. Additional public datasets, newly released multi-turn benchmarks such as MultiBreak (Song et al., 2026) or CoSafe (Yu et al., 2024), and broader proprietary or open-source guardrail baselines are important future work. We also do not claim that the model distinguishes all failed jailbreak attempts from benign conversations: under success-based labels, consistently refused attacks can appear as safe or hard-negative examples, while attempt-labeled datasets may use different semantics. Finally, the open-domain safe corpora (UltraChat, WildChat, ShareGPT, and OASST2) are used as released, without additional safety filtering; because in-the-wild logs can contain toxic or adversarial content, a small fraction of safe labels may be noisy.

Comparability of baselines and empirical efficiency. Our detector is fine-tuned on the in-domain training mixture, whereas the proprietary Claude baselines and the Llama 3.3 70B Concatenation/ToT baselines are evaluated via prompting without equivalent in-domain fine-tuning; the reported gap therefore reflects both the hierarchical architecture and the benefit of task-specific supervision, and

it should not be read as an architecture-only comparison. Isolating the architectural contribution against a supervised long-context sequence classifier (e.g., a Longformer fine-tuned on the identical training set) and evaluating the concatenation baseline on the full conversation context rather than a fixed 4-turn window are important controls we leave to future work. Relatedly, the Concatenation baseline is deliberately restricted to a short recent-turn window to reflect a low-latency operating point; a full-context variant would provide a more favorable, and more informative, comparison. Our efficiency argument is currently analytical (Appendix A.3) rather than measured; direct comparisons of inference latency, peak memory, and throughput against full-sequence concatenation are left to future work. Finally, the results in Tables 2–4 are single-run point estimates; reporting variance across multiple random seeds would help confirm that the observed differences, particularly the narrow gaps between adjacent ConvTransformer depths, are statistically robust.

A.7 Additional Analysis Figures

This section collects secondary visualizations that complement the ablation tables in Section 8.

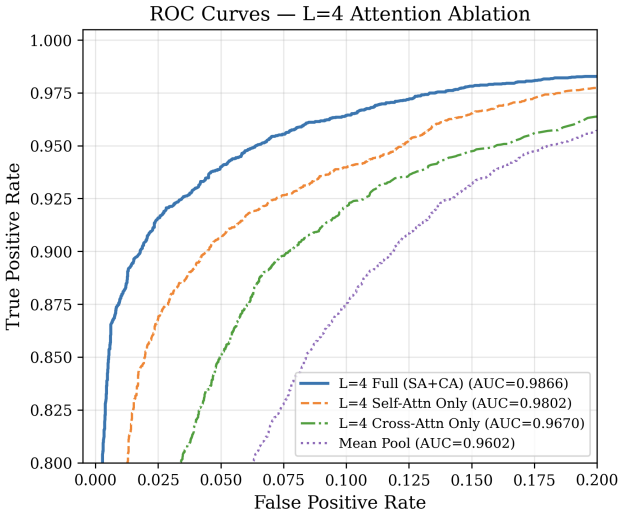


Figure 4: ROC comparison across attention component ablation variants.

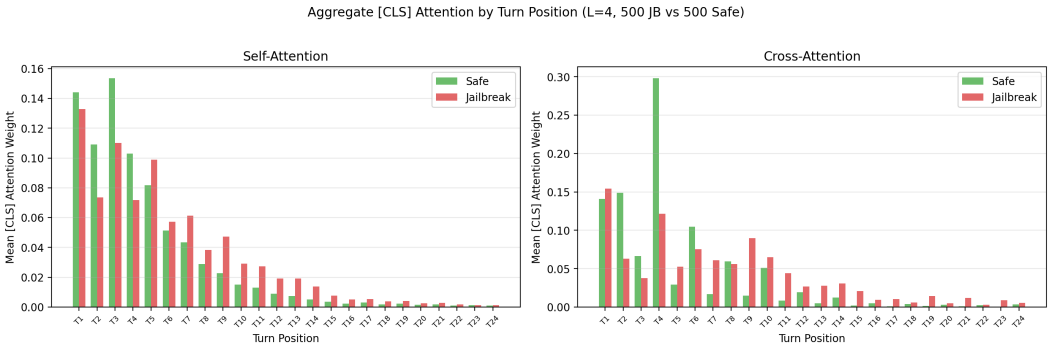


Figure 5: Raw aggregate `[CONV_CLS]` attention weights by turn position for jailbreak and safe conversations (L=4 ConvTransformer, n=500+500). Left: self-attention; right: cross-attention. Jailbreak conversations shift attention toward later turns. The corresponding JB/Safe ratios are shown in Figure 3.

B Evaluation Benchmark Composition

Tables 7 and 8 provide a complete per-source breakdown of the 14,038-conversation evaluation benchmark introduced in Section 5.

Table 7: Safe conversations in the evaluation benchmark (label $= 0$): 8,182 total.

Source	N	Turns	Description	Reference
UltraChat	2,937	4–14	Synthetic open-domain dialogues on science, daily life, technology, and creative topics.	(Ding et al., 2023)
WildChat	2,800	4–86	Real user–ChatGPT interactions with follow-up questions, clarifications, and diverse intents.	(Zhao et al., 2024)
ShareGPT	666	4–72	Community-shared ChatGPT conversations spanning coding help, writing, and general Q&A.	(ShareGPT Community, 2023)
OASST2	262	4–6	Crowd-sourced assistant dialogues covering factual questions and task completion.	(Köpf et al., 2023)
Synthetic hard neg.	388	4–58	LLM-generated benign conversations that structurally resemble attacks but contain no adversarial intent.	Generated
Long hard neg.	278	4–102	Extended (20+-turn) benign conversations; controls for length bias in the detector.	Generated
Benign follow-ups	390	4	Short benign conversations with polite user follow-ups; targets gratitude patterns confused with post-refusal jailbreak turns.	Generated
SafeMTData (safe)	192	4–12	Multi-turn conversations on sensitive-adjacent topics (e.g., DNS, cybersecurity) with genuinely benign intent.	(Ren et al., 2024)
Security discussions	190	4–30	Benign conversations on penetration testing and red-teaming methodology.	Generated
Other	79	4–5	Empathetic dialogues and safe adversarial-ML research conversations.	Mixed

Table 8: Jailbreak conversations in the evaluation benchmark (label $= 1$): 5,856 total.

Source	N	Turns	Description	Reference
XGuard	1,003	4–14	Red-team attacks using persona adoption and role-play framing.	(Abishethvarman et al., 2025)
Red Queen	802	5–7	Short targeted manipulation exploiting trust or authority claims.	(Jiang et al., 2025)
SafeDialBench	408	6–16	Multi-turn safety dialogues testing gradual boundary-pushing across harm categories.	(Cao et al., 2025)
carl213	111	4–26	Cipher-encoded attacks using user-defined word-substitution mappings.	(Gibbs et al., 2024)
Duke Safety	75	5	Short gradual-escalation attacks across 37 harm categories.	(Zhang et al., 2025)
Long escalation	828	4–32	Extended (15+-turn) conversations incrementally steered toward harmful requests.	Generated
Synth. escalation	465	9–20	LLM-generated gradual-escalation attacks with controlled escalation speed.	Generated
FRACTURED-SORRY	94	7–15	Multi-step attacks with benign intermediary turns fragmenting harmful intent.	(Priyanshu & Vijay, 2024)
Synth. sudden	461	9–21	Benign conversations with a single adversarial turn inserted at a random late position.	Generated
Long sudden	268	19–31	Long benign conversations with adversarial insertion; tests detection under signal dilution.	Generated
Ctx. narrowing	469	7–15	Benign multi-turn prefix followed by a final user turn demanding harmful reformatting of the context.	Generated
Prompt injection	318	2–12	Short instruction-override attacks ("ignore previous instructions", system-prompt extraction, hypothetical framing).	Generated
LAD	152	4–18	Gradual escalation in technical domains (SCADA, AVs, network security) from benign inquiry to exploit requests.	(Kulkarni, 2026)
Gradual technical	402	4–30	Benign technical questions (cybersecurity, programming) escalating into explicit exploit/attack-code requests.	Generated