Alignment-Constrained Dynamic Pruning for LLMs: Identifying and Preserving Alignment-Critical Circuits

Dev Patel, Gabrielle Gervacio, Diekola Raimi, Kevin Zhu , Ryan Lagasse, Gabriel Grand, Ashwinee Panda, Maheep Chaudhary[*]

Algoverse AI Research
pateld@stpaulsschool.org.uk, maheepchaudhary.research@gmail.com

Abstract

Large Language Models require substantial computational resources for inference, posing deployment challenges. While dynamic pruning offers superior efficiency over static methods through adaptive circuit selection, it exacerbates alignment degradation by retaining only input-dependent safety-critical circuit preservation across diverse inputs. As a result, addressing these heightened alignment vulnerabilities remains critical. We introduce Alignment-Aware Probe Pruning (AAPP), a dynamic structured pruning method that adaptively preserves alignment-relevant circuits during inference, building upon Probe Pruning. Experiments on LLaMA 2-7B, Qwen2.5-14B-Instruct, and Gemma-3-12B-IT show AAPP improves refusal rates by 50% at matched compute, enabling efficient yet safety-preserving LLM deployment.

Introduction

LLMs deliver impressive capabilities yet impose high computational costs, with inference costs scaling directly with model size (Kaplan et al. 2020). Pruning offers a promising route to reduce these costs (Han, Mao, and Dally 2016), using different techniques, including static structured pruning (Ma, Fang, and Wang 2023) as well as dynamic probe-guided pruning (PP) (Le et al. 2025) which improves the accuracy-efficiency frontier by pruning columns of the learnable linear transformation that maps intermediate hidden state to the output hidden state, referred to as an input channel. However, these methods risk pruning alignment-critical structures, potentially weakening safety guardrails and degrading behaviors such as refusal of harmful instructions. Recent analyses (Wei et al. 2024) show that removing as little as 3% of parameters is enough to compromise safety. This brittleness motivates the development of Alignment-Aware Probe Pruning (AAPP)—a method that explicitly preserves alignment-critical circuits.

AAPP uses the average activation value for each input channel. By comparing these scores obtained from benign and harmful prompts to the scores obtained from our probe pass, our method detects adversarial inputs and enforces hard exclusions on alignment-critical structures. This structured pruning approach yields an improved efficiency-alignment frontier: AAPP outperforms PP, having refusal rates up to 50% greater for the same computational budget. These findings suggest constraint-satisfying pruning as a practical route to efficient yet safe LLMs. Our key contributions are as follows:

- We develop a pruning framework that preserves interpretable circuits
- We evaluate our framework on refusal rate, toxicity, accuracy, and computational cost (FLOPs)

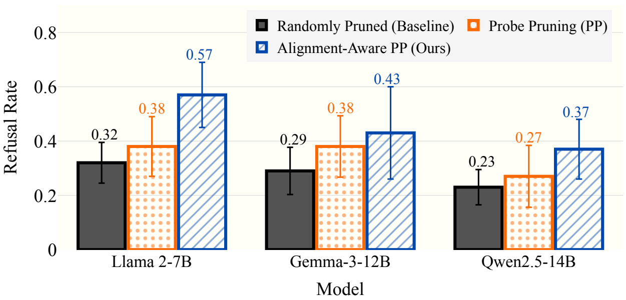

Figure 1: Refusal rates of LLaMA-2-7B, Qwen-2.5-14B, and Gemma-3-12B models on the WildJailbreak dataset (Jiang et al. 2024) under pruning ratio $r = 0.3$. We compare our Alignment-Aware Probe Pruning (AAPP) against two baselines: Probe Pruning (PP) (Le et al. 2025) and random pruning. Across all three models, AAPP consistently achieves higher refusal rates, demonstrating that preserving alignment-critical circuits upon the detection of adversarial prompts improves safety behavior under pruning.

Related Work

Structured Pruning

Structured pruning is a key approach for reducing the computational cost of LLMs. LLM-Pruner (Ma, Fang, and Wang 2023) removes entire attention heads and MLP neurons via gradient-based importance, while Wanda (Sun et al. 2024) prunes weights with small magnitude and activation values post-hoc, achieving high sparsity without retraining. Probe Pruning (Le et al. 2025) extends this line by using probed hidden states to guide batch-wise pruning, improving the

[*] Project Lead

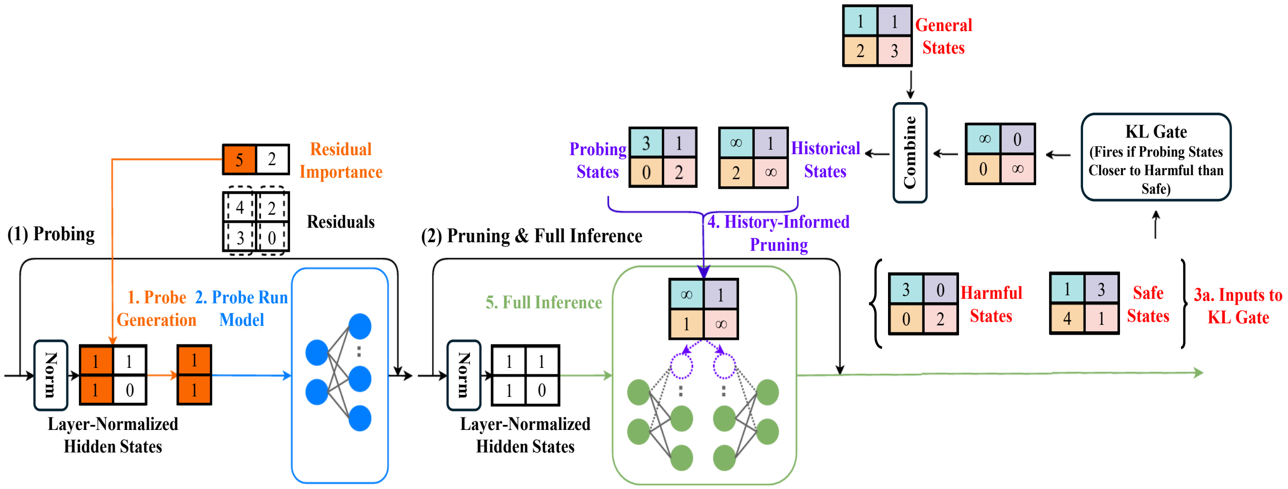

Figure 2: Alignment-Aware Probe Pruning (PP) is executed in five stages: (1) From the layer-normalized hidden states, pick tokens based on residual-importance and build a small probe. (2) Run the probe a few layers ahead to produce probing states (3a) A KL Gate compares them to historical states from safe and harmful prompts and fires when closer to harmful, ensuring the preservation of alignment-critical structures. If the gate does not fire, the probe states are just fused with the general historical states (4) Using the integrated states to calculate the pruning metric (Le et al. 2025), prune low-score channels. (5) Perform full inference on the remaining weights.

accuracy-efficiency frontier. However, these methods risk pruning the preservation of alignment-critical structures.

Alignment Preservation

Several methods aim to preserve alignment during model modification. Safe LoRA (Hsu et al. 2024) and SaLoRA (Li et al. 2025) constrain LoRA updates to remain within safety-aligned subspaces, while LoRI (Zhang et al. 2025) and LoTA (Panda et al. 2024) apply structural sparsity to reduce catastrophic forgetting. These works show that constraining fine-tuning helps preserve desirable behaviors in LLMs. NLSR (Yi et al. 2025) restores safety by transplanting safety-critical neurons from an aligned reference model. These approaches show that explicit parameter constraints and neuron transplantation can maintain refusal, honesty, and toxicity safeguards even under structural changes. Layer-level analyses further support targeted preservation: Shi et al. (Shi et al. 2024) showing that alignment changes concentrate in late-stage layers and that compression can focus on non-critical regions.

Methods

As shown in Figure 2, Alignment-Aware Probe Pruning consists of five stages, namely probe generation; probing, recording activations; comparison to our historical activation scores; history-informed pruning; and inference.

Activations and Scoring

For each target with C input channels, we create 3 tensors: general, benign, and harmful using sets of prompts: (1) general prompts to maintain linguistic functionality from C4 dataset (Raffel et al. 2020); (2) benign prompts from wild adversarial dataset; and (3) harmful prompts from wild adversarial dataset. ((Jiang et al. 2024)). Each set of scores stores the squared ℓ_2 norm of channel activations compressed across the batch and sequence dimensions. We refer to this value as the "channel's energy".

For structured pruning, we adopt the PP_{sp} importance metric from Probe Pruning (Le et al. 2025), which computes per-channel pruning scores using the ℓ_2 norms of each input channel's activations. Here, W^{final} denotes the learnable linear transformation between hidden states, and X^{int} the intermediate hidden state. A lower PP_{sp} score, I_k, indicates less important channels.

$$I_k = \left\| \left\{ |W^{\text{final}}_{i,k}|^2 \cdot \|X^{\text{int}}_{:,:,k}\|_2^2 \right\}_{i=0}^{C_{\text{out}}} \right\|_2 , \qquad (1)$$

Finally, we blend live scores with stored activation scores obtained from the set of general prompts.

Risk-aware gate and channel selection

We keep $k = \lceil (1-r)C \rceil$ channels, reserving $k_{align} = \lfloor \text{align\_frac} \cdot C \rfloor$ channels for safety. Probing states; and historical states from benign and harmful prompts are normalized into distributions: 'p'; and 'q_{safe}, and q_{jail}', respectively, using Equation 2.

$$KL_{\text{harm}} = \sum_c p_c \log \frac{p_c}{q^c_{\text{jail}}}, \quad KL_{\text{safe}} = \sum_c p_c \log \frac{p_c}{q^c_{\text{safe}}}. \qquad (2)$$

If $KL_{\text{harm}} - KL_{\text{safe}} \geq \tau_{\text{margin}}$, we preserve the top k_{align} channels by $hist_{\text{jail}}$ as we wish to protect channels most active under harmful prompts because they include refusal circuitry. We then fill the remainder by descending score. Otherwise, we retain the top k channels by score. Using these scores, binary masks are generated for pruning and then materialized to obtain real compute reductions.

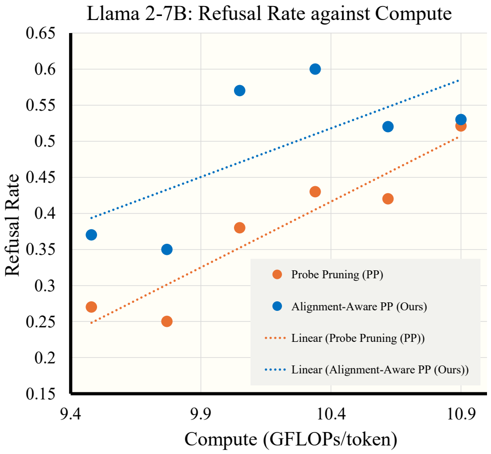

Figure 3: Refusal rate vs compute (GFLOPs/token) on Llama-2-7B-chat. AAPP maintains substantially higher refusal rates at comparable compute budgets, achieving safer behavior with fewer FLOPs compared to standard PP.

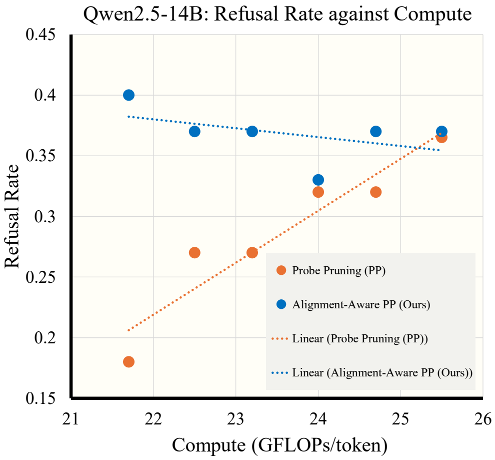

Figure 4: Refusal rate vs compute (GFLOPs/token) on Qwen2.5-14B-Instruct. AAPP preserves refusal performance as compute decreases, improving the refusal-compute trade-off relative to PP across the efficiency spectrum.

Experimentation and Results

We evaluate on HuggingFace implementations of Llama-2-7B-chat, Qwen2.5-14B-Instruct, and Gemma-3-12B-IT, using prompts from the WildJailbreak dataset ((Jiang et al. 2024)) which were not used for the generation of historical states. Workloads contain prompts of avg. length 300 tokens with 120 tokens generated. Unless stated otherwise, we fix hyperparameters to align frac $= 0.3$, refresh window $= 20$, and batch size $= 20$ for prompts.

We estimate inference FLOPs calculated using 2 FLOPs/-MAC ((Hoffmann et al. 2022)) taking into account the number of layers, attention heads, hidden size, intermediate size, and vocabulary size for the given model. We prune only in the input channels of attention o_{proj} and MLP $down_{\text{proj}}$, excluding the first 6 and last 3 layers. Outputs are post-hoc labeled for refusal and toxicity. Metrics include throughput compute (FLOPs/token), refusal rate (trained classifier), classification accuracy and toxicity (Perspective API (Lees et al. 2022)).

Across the two methods (AAPP and PP), We first consider the model's ability to classify harmful and unharmful prompts and act accordingly. This is investigated across various compute budgets and prune ratios. Following this, we assess the safety of the model's responses for AAPP and PP using toxicity as the measure.

Refusal Rates at Fixed Prune Ratio

Figure 1 presents refusal rates at prune ratio $r = 0.3$. Across all three models, AAPP achieves higher refusal rates (implicit and explicit) than both Randomly Pruned and Probe Pruning (PP) baselines, preserving alignment behavior. On Llama-2-7B-chat, AAPP attains a refusal rate (0.57) 50% and 78% greater than PP (0.38) and Random Pruning (0.32), respectively. Similar improvements hold for Llama-2-7B-chat (37% and 61%) and Gemma-3-12B-IT (13% and 48%), confirming the robustness of our approach across architectures.

Refusal Rates against Compute (FLOPs per Token)

Extending the investigation, we vary compute budgets to look into the alignment-efficiency frontiers created using either method. Figure 3 and 4 illustrates alignment (refusal rate) as a function of computational efficiency (GFLOPs/token) for the Llama-2-7B-chat and Qwen2.5-14B-Instruct models, respectively, under Probe Pruning (PP) and Alignment-aware PP. Given the same computational budget, our method achieves a higher refusal rate, shifting the efficiency-alignment frontier upward. For example, on Llama-2-7B-chat (3), to achieve a target refusal rate of 0.5, our method requires only 10.3 GFLOPs/token, compared to a higher cost with PP. Qwen2.5-14B-Instruct exhibits the same pattern, demonstrating that AAPP maintains safety more efficiently across various compute levels. These results show that AAPP improves the alignment-efficiency tradeoff, achieving safer behavior while reducing inference cost, and generalizing across diverse model families.

Alignment Accuracy

The accuracy of these refusals and the behavior of the model, more generally, is shown in Table 1. It indicates that AAPP outperforms PP across prune ratios on Llama-2-7B-Chat and Qwen2.5-14B-Instruct. The results for the pruned models are compared to the unpruned model, which we consider to have a maximum for these metrics, as our pruned models cannot exceed the performance of the base model. We use F1 to balance recall and precision, accuracy and False Acceptance Rate to indicate how often the model does not refuse prompts. PP's accuracy and F1 decline as pruning increases, dropping to 0.575 and 0.585 at a 0.3 ratio for

Model	Prune Ratio	Method	F1 (↑)	Accuracy (↑)	FAR (↓)
Llama-2-7B-chat	0	PP	1.000	1.000	0.000
		AAPP	1.000	1.000	0.000
	0.15	PP	0.725	0.702	0.290
		AAPP	0.834	0.808	0.201
	0.3	PP	0.645	0.624	0.313
		AAPP	0.760	0.741	0.254
Qwen2.5-14B-Instruct	0	PP	1.000	1.000	0.000
		AAPP	1.000	1.000	0.000
	0.15	PP	0.876	0.891	0.058
		AAPP	0.880	0.916	0.05
	0.3	PP	0.730	0.820	0.169
		AAPP	0.786	0.858	0.092

Table 1: Comparison of F1, Accuracy and FAR for PP and AAPP across prune ratios on Llama-2-7B-Chat and Qwen2.5-14B-Instruct: AAPP has a lower False Acceptance Rate with higher classification accuracy, behaving more similarly to the unpruned models.

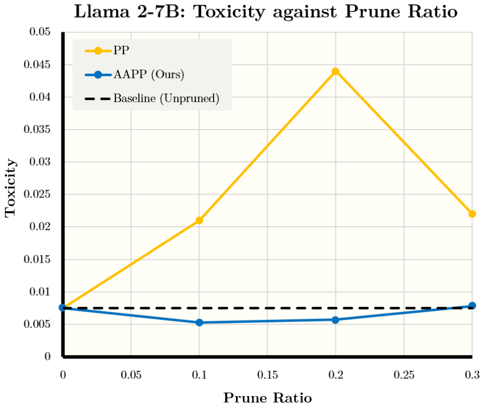

Figure 5: Toxicity vs prune ratio on Llama-2-7B-chat. AAPP maintains toxicity levels closer to the unpruned baseline compared to PP, demonstrating better preservation of safety alignment under aggressive pruning.

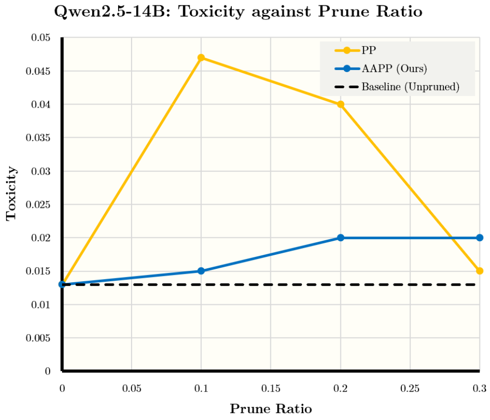

Figure 6: Toxicity vs prune ratio on Qwen2.5-14B-Instruct. AAPP sustains lower toxicity scores closer to the unpruned model across pruning ratios, outperforming PP in safety preservation.

Llama2-7B-Chat. In contrast, AAPP retains higher values, 0.741 accuracy and 0.760 F1, indicating stronger classification stability. Additionally, AAPP maintains a lower False Acceptance Rate (FAR) (e.g. 0.216 vs 0.353 at 0.3). Similar results can be seen for Qwen2.5-14B-Instruct. Overall, these results demonstrate AAPP's ability to preserve safety and behavior near to the unpruned models at reduced compute.

Toxicity against Prune Ratio

Through toxicity, we can understand how safely the model responds. Figure 5 and 6 indicates that across both models, AAPP shows clear safety gains over PP. On Llama-2-7B-Chat, PP's toxicity peaks at 0.044 at a 0.2 prune ratio, while AAPP stays nearly constant near 0.0075, matching the unpruned baseline. Similarly, on Qwen2.5-14B-Instruct, PP reaches 0.08, but AAPP remains below 0.02. This demonstrates that AAPP preserves alignment even under heavy pruning. Although toxicity scores decrease at high pruning ratios, this may reflect linguistic degradation rather than improved safety. Pruning can suppress expressive activations, yielding flatter, less coherent text that is rated as less toxic.

Conclusion

We a pruning method that preserves alignment while reducing inference cost. By integrating a risk-aware gate with probe-guided pruning, we prevent the removal of alignment-critical structures upon the input of an adversarial prompt and improves the efficiency-alignment frontier. Experiments on Llama-2-7B-chat, Qwen2.5-14B-Instruct, and Gemma-3-12B-IT show that AAPP sustains lower toxicity and greater classification accuracy at lower FLOP budgets, offering a practical route to safer and more efficient LLMs.

References

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D. d. L.; Hendricks, L. A.; Welbl, J.; Clark, A.; Hennigan, T.; Noland, E.; Millican, K.; Dyer, E.; Irving, G.; Rae, J. W.; van den Driessche, G.; de Haas, B.; Battaglia, P.; Malinowski, M.; Guy, A.; Osindero, S.; Kavukcuoglu, K.; Ring, R.; Cain, A.; Hillier, C.; Winter, R.; Hutter, O.; Lillicrap, T.; Green, S.; Cassirer, A.; Jones, C.; Cherepanova, V.; Rutherford, A.; Mensch, F.; Crampton, N.; Manning, S.; van Steenkiste, S.; and Sifre, L. 2022. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hsu, C.-Y.; Tsai, Y.-L.; Lin, C.-H.; Chen, P.-Y.; Yu, C.-M.; and Huang, C.-Y. 2024. SPLoRA: The silver lining of reducing safety risks when fine-tuning large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jiang, L.; Rao, K.; Han, S.; Ettinger, A.; Brahman, F.; Kumar, S.; Mireshghallah, N.; Lu, X.; Sap, M.; Choi, Y.; and Dziri, N. 2024. WildTeaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Le, Q.; Diao, E.; Wang, Z.; Wang, X.; Ding, J.; Yang, L.; and Anwar, A. 2025. Probe pruning for efficient large language models. In *International Conference on Learning Representations (ICLR)*.

Lees, A.; Tran, V. Q.; Tay, Y.; Sorensen, J.; Gupta, J.; Metzler, D.; and Vasserman, L. 2022. A new generation of Perspective API: Efficient multilingual character-level transformers. *arXiv preprint arXiv:2202.11176*.

Li, M.; Si, W. M.; Backes, M.; Zhang, Y.; and Wang, Y. 2025. SaLoRA: Safety-alignment preserved low-rank adaptation. In *International Conference on Learning Representations (ICLR)*.

Ma, X.; Fang, G.; and Wang, X. 2023. LLM-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Panda, A.; Isik, B.; Qi, X.; Kojejo, S.; Weissman, T.; and Mittal, P. 2024. Lottery Ticket Adaptation: Mitigating Destructive Interference in LLMs. *arXiv preprint arXiv:2406.16797*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Shi, G.; Lu, Z.; Dong, X.; Zhang, W.; Zhang, X.; Feng, Y.; and Wu, X.-M. 2024. Understanding layer significance in LLM alignment. *arXiv preprint arXiv:2410.17875*.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2024. Wanda: A simple and effective pruning approach for large language models. In *International Conference on Learning Representations (ICLR)*.

Wei, B.; Huang, K.; Huang, Y.; Xie, T.; Qi, X.; Xia, M.; Mittal, P.; Wang, M.; and Henderson, P. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *ICLR Workshop on Understanding of Foundation Models (ME-FoMo)*.

Yi, X.; Zheng, S.; Wang, L.; de Melo, G.; Wang, X.; and He, L. 2025. NLSR: Neuron-level safety realignment of large language models against harmful fine-tuning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Zhang, J.; You, J.; Panda, A.; and Goldstein, T. 2025. LoRI: Reducing Cross-Task Interference in Multi-Task Low-Rank Adaptation. *arXiv preprint arXiv:2504.07123*.