Smarter, Not Harder: Training-Free Adaptive Computation for Transformers

Romain Storaï[*], **Jaeseong Lee**[*], **Seung-won Hwang**[†]
Computer Science and Engineering, Seoul National University
{romsto,tbvj5914,seungwonh}@snu.ac.kr

Abstract

Adaptive Computation in Transformers (ACT) has been pursued in two directions: efficiency- and performance-focused. We study performance-focused ACT, or PACT, which invests more computation on hard steps to improve performance, such as by adding forward passes. We first discuss beam search and hesitation-based methods as PACT and their limitations. While the hesitation-based approach outperforms beam search by perturbing input embeddings, it suffers from inefficiency due to invalidating KVCache and exhibits instability due to its reliance on randomness. To address this, we propose IMPACT, a novel PACT method that perturbs network weights rather than input embeddings. This approach enables the reuse of KVCache, offers deterministic predictions, and significantly improves memory and computational efficiency. By achieving a better balance between performance and efficiency, IMPACT makes PACT accessible to communities with consumer-grade hardware.

1 Introduction

Scaling the size and pretraining of Large Language Models (LLMs) has revolutionized their capabilities, enabling remarkable performance across a wide range of tasks (Brown et al., 2020). However, this scaling also makes retraining LLMs prohibitively expensive, motivating recent approaches for improving inference.

One promising direction for this is Adaptive Computation in Transformers (ACT), a paradigm that dynamically adjusts computation at the token level to handle varying complexities during inference. ACT has been explored along two primary directions: efficiency-focused and performance-focused approaches. The former aims to reduce computation for simpler inferences, while the latter seeks to enhance performance by allocating more computation to complex inference steps.

Beam search (Lowerre, 1976), a constrained variant of breadth-first search that explores multiple hypotheses at each step, is often adopted for enhancing performance at the cost of increased inference time for enumerating multiple hypotheses. While originally designed to mitigate the label bias problem in greedy decoding, rather than for such performance-focused ACT (PACT) uses, its effectiveness as a PACT has been explained through the Uniform Information Density (UID) hypothesis (Jaeger and Levy, 2006; Meister et al., 2020). According to UID, human-generated sentences tend to distribute information evenly, and beam search aids in making the surprisal values of model outputs more uniform, resulting in more human-like text.

A stronger example of PACT is hesitation-based methods, such as HARP (Storaï and Hwang, 2024), as more directly aligned with UID. HARP introduces ‘hesitation’ into high surprisal token generation, mimicking how humans deliberate on complex decisions. Specifically, it first identifies points of hesitation and then performs additional forward passes to generate reframed predictions, which can be implemented using random dropout on input embeddings (Srivastava et al., 2014).

Despite their potential, existing hesitation-based methods face significant drawbacks, including increased latency and memory overhead. By randomly perturbing input embeddings, these methods invalidate key-value caches (KVCache), leading to inefficiencies. Furthermore, randomness can sometimes make the performance of HARP inferior.

To address these challenges, we propose IMPACT, a PACT method designed to better balance performance and efficiency. **First**, IMPACT enables **KVCache reuse**, critical for complex tasks requiring a long generation, a significant demand

[*] Equal contribution.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 8147–8155
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

for recent reasoning-focused models. IMPACT builds on the principles of HARP but replaces the reframing through input embedding dropout with an approximation of the original network. **Second**, by eliminating random perturbations, IMPACT ensures a fully **deterministic** forward pass, improving both reliability and efficiency. Unlike HARP, which often requires multiple KVCache-invalidated runs due to its stochastic nature, we show comparable gains without compromising generalization.

We empirically validate IMPACT on reasoning-intensive tasks and modern approaches to reasoning, such as chain-of-thought. Qualitatively, we demonstrate that hesitation-based methods, such as IMPACT, align closely with the UID hypothesis. Additionally, IMPACT is effective across models of various scales–from Llama-3.1-8B-Instruct to Llama-3.3-70B-Instruct. The code for this method is publicly available.[1]

2 Preliminary and Related Work

This section overviews token-level PACT approaches, though other PACT approaches such as Pause Tokens (Goyal et al., 2024), or Test-time scaling (TTS; Snell et al., 2025), can be applied in conjunction for further improvements.

Uniform Information Density Hypothesis To illustrate UID, consider: *'How big is the family (that) you should cook for?'* "that" is optional but often preferred by UID, as it helps reduce the surprisal of the next token "you".

Greedy decoding may omit optional words when having slightly lower probability–by immediately committing to the most probable next token, while beam search may choose such words, as later contexts from multiple hypothesis can improve earlier decisions.

Hesitation-based PACT Hesitation-Aware Reframed Forward Pass (HARP; Storaï and Hwang, 2024) is more directly aligned for the UID hypothesis. It consists of two main components: (1) token-level uncertainty estimation to identify hard–surprising–steps, and (2) an additional Transformer forward step for those steps. We interpret step (1) as identifying high-surprisal steps, and step (2) as reducing surprisal. HARP components and functioning are detailed in Appendix A.

By directly defining hard steps as those with high entropy (i.e., high surprisal), hesitation-based PACT, such as our proposal IMPACT, aligns more strongly with the UID principle. We empirically confirm in Section 4 that IMPACT lowers entropy, thus reducing surprisal.

3 IMPACT: Efficiency IMproved PACT

This section discusses the inefficiencies associated with UID computation and introduces IMPACT as a solution.

3.1 Challenges of Cache-Invalidating HARP

KVCache significantly improves efficiency in Transformer-based models (Vaswani et al., 2017) by storing key-value pairs from the self-attention mechanism during the forward pass, allowing instant retrieval of previously computed token representations and avoiding redundant computations in subsequent steps.

By applying dropout on input embeddings, HARP makes the KVCache reuse impossible. We formally prove that dropout frequently alters a significant portion of tokens, continuously disrupting the cache (proof in Appendix B). Consequently, this process increases both latency and memory usage, undermining the efficiency benefits of KVCache (Appendix D). In such a constrained setting, where HARP can be applied on top of the original model for the given resources, users may prefer to use a larger variant of the original model to achieve better performance.

3.2 KVCache-friendly Reframing

Model Approximation for Reframing In HARP, two forward passes on $\mathbf{e}$ and $\hat{\mathbf{e}}$ generate $logits$ and $logits_r$ which contain predictions for each position. The next-token prediction corresponds to the last position.

To address the inefficiencies of multiple passes, we propose a cache-aware reframing technique. Instead of performing costly passes on different inputs, which cannot utilize the cache, we leverage the same input $\mathbf{e}$ to generate both original and reframed predictions, i.e., $logits$ and $logits_r$. This reframing offers an alternate perspective on predictions without sacrificing quality, all while preserving computational efficiency.

To achieve this, we employ an approximation of the original model rather than introducing different inputs. It would enable both KVCache reuse

[1] https://anonymous.4open.science/r/IMPACT-739B

and the generation of reliable reframed predictions. Specifically, we use *contextual sparsity*, building a structure-pruned network that selectively ignores input activations with the smallest magnitudes.

Contextual Sparsity for Approximation Contextual sparsity has been widely studied in systems research to enhance computational efficiency in deep learning by pruning small-magnitude activations. Our approach repurposes these methods beyond efficiency, utilizing sparsity as a mechanism for controlled reframing while preserving the KV-Cache structure.

Formally, consider a linear layer in a network with a weight matrix W and input $\mathbf{v}$. The original layer output is given by:

$$h(\mathbf{v}) = W\mathbf{v} \tag{1}$$

Given a threshold t, we zero-out the i-th column of W, denoted as W_i, if the corresponding input v_i has a magnitude below t. t is pre-calculated with a validation set, to meet the desired sparsity of W. That is,

$$\hat{W_i} = \begin{cases} 0 & \text{if } |v_i| < t, \\ W_i & \text{otherwise} \end{cases} \tag{2}$$

$$\hat{h}(\mathbf{x}) = \hat{W}\mathbf{v} \tag{3}$$

t simply becomes the s-percentile of the distribution of the magnitudes of elements in $\mathbf{v}$, where $s\%$ is the desired sparsity ratio. Then with this pre-calculated t, we apply Eq. 3 during the inference time. The sparsity s is conventionally selected as the highest value showing a negligible drop in the quality of the output (Liu et al., 2024; Lee et al., 2024).

When hesitation occurs, applying contextual sparsity to every linear layer in the network generates the reframed logits $logits_r$. These logits are then linearly combined with the original $logits$ as described in Eq. 6, similar to HARP.

In summary, by changing the reframing process of HARP, IMPACT preserves the **deterministic** behavior by eliminating the randomness introduced by dropout in HARP (Eq. 5), while also enabling **KVCache reuse**. When diversity needs to be added, IMPACT can easily be combined with nucleus sampling, or any other stochastic decoding method, for decoding.

4 Experiments

We use Llama-3.1-8B-Instruct and Llama-3.3-70B-Instruct (Dubey et al., 2024) as our base models. All evaluations are conducted using a batch size of 1, in a greedy decoding setting (temperature of 0). Latency is evaluated on the whole dataset using a batch size of 1 and a concurrency of 1, following the convention of HARP. For beam search, we use a beam size b of 3, and length-normalization (Wu et al., 2016) of 0.6, to be consistent with previous work (Storaï and Hwang, 2024). We detail the experimental settings in Appendix E.

	GSM8K		LogiEval		CsQA	
Method	Lat.	Acc.	Lat.	EM.	Lat.	EM.
Original	1.00×	77.71	1.00×	54.39	1.00×	75.92
HARP	6.54×	77.86	2.98×	**56.21**	1.32×	**76.14**
Beam search	1.37×	78.17	2.05×	54.45	1.73×	75.92
IMPACT	1.57×	**78.85**	1.36×	54.58	1.18×	75.92

Table 1: GSM8K, LogiEval and Commonsense QA results of Llama-3.1-8B-Instruct with different PACT methods.

	GPQA		BBH		MMLUPro	
Method	Lat.	Acc.	Lat.	EM.	Lat.	EM.
Original	1.00	28.86	1.00	69.56	1.00	43.80
HARP	3.85	33.00	8.09	70.13	9.56	45.20
IMPACT	1.14	32.63	1.19	69.84	1.81	44.00

Table 2: Accuracy of Llama-3.1-8B-Instruct over three reasoning datasets using chain-of-thought.

IMPACT vs. Other PACT Methods: Balancing Performance and Efficiency Table 1 shows that IMPACT outperforms both HARP and beam search, in terms of balancing performance and efficiency. For GSM8K, the random nature of HARP leads to suboptimal average performance compared to beam search. Moreover, HARP requires almost 6.5 times longer time to generate an answer because of its inability to cache. On the other hand, IMPACT outperforms beam search while maintaining a comparable latency. Results over LogiEval and Commonsense QA (CsQA) further highlight the efficiency of IMPACT.

Performance on Complex Reasoning Tasks Chain-of-thought prompting (Wei et al., 2024) encourages the model to generate long reasoning steps, often prone to hesitation, making the generation process more complex. Table 2 presents the results using this modern approach across three datasets. While HARP achieves slightly higher accuracy improvements, it comes at a significant computational cost, leading to a large increase in inference latency. In contrast, IMPACT achieves a

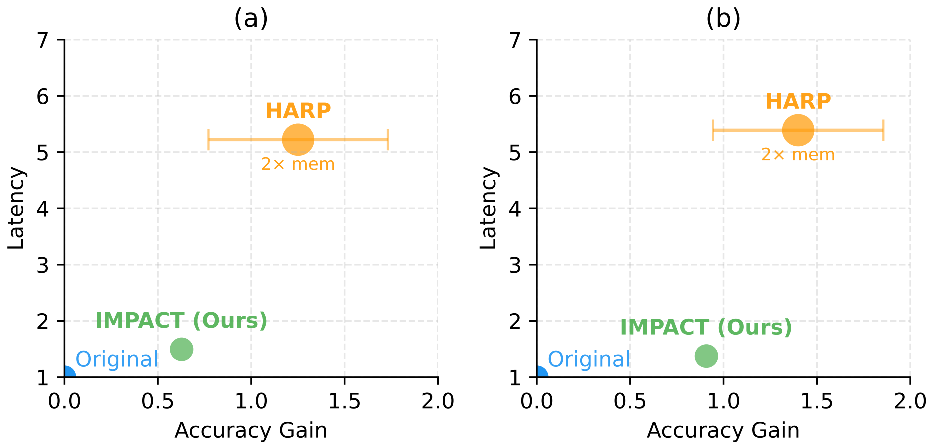

Figure 1: IMPACT introduces a new Pareto frontier in the accuracy-latency trade-off space. Circle sizes indicate the memory requirements of each method. In the left figure (a), the accuracy is averaged across all tasks described in Section E, including reasoning and non-reasoning tasks. In contrast, the right figure (b) reports accuracy averaged only over reasoning tasks

better balance between performance and efficiency, preserving accuracy gains while significantly reducing computational overhead.

	LAMBADA		CNN/DM	
Method	Latency	EM.	Latency	Rouge-1
Original	1.00×	55.48	1.00×	29.52
HARP	2.20×	56.62	7.23×	29.99
IMPACT	2.30×	55.52	1.41×	29.05

Table 3: Results of PACT methods, using Llama-3.1-8B-Instruct, on two non-reasoning tasks.

Limitation on Non-Reasoning Tasks We claimed that IMPACT is especially effective on reasoning tasks, where precision is essential. We observe mixed results on non-reasoning tasks (Table 3), confirming our argument.

Method	GSM8K	LogiEval
Original	93.63	72.07
Beam search	92.95	71.88
IMPACT (Ours)	**93.71**	**72.14**

Table 4: GSM8K accuracy and LogiEval exact match score of Llama-3.3-70B-Instruct with different PACT methods.

Scaling IMPACT to Larger Models Table 4 shows that IMPACT continues to outperform other PACT methods in experiments, even with more parameters, validating the benefits of hesitation-based methods. While beam search lowers the performance, IMPACT consistently improves it. The excessive latency and memory constraints of HARP prevented the evaluation of hesitation on larger models.

5 Analysis

New Pareto-frontier Figure 1 shows how the reframing process of IMPACT leads to providing a new Pareto-frontier in the performance-latency trade-off space. By offering a more cost-effective alternative to HARP while maintaining comparable performance, IMPACT expands the Pareto solution space, much like LLMs are released in varying sizes to balance efficiency and performance. The contrast between Figure 1a and Figure 1b highlights that the overall accuracy gain of IMPACT diminishes when non-reasoning tasks are included in the average, suggesting that its advantage is most pronounced on reasoning tasks.

Stochastic vs Deterministic PACT In Figure 1, the variance of HARP, shown as an error bar, illustrates the downside of HARP's stochastic nature. While HARP improves average performance, its variability makes it challenging to ensure consistent improvements– For example, two of the five runs of HARP for GSM8K in Table 1 showed worse performance than the original model (76.58 and 77.32). Keeping the deterministic aspect of the original forward pass is a strong advantage of IMPACT, ensuring the results are consistently improved.

IMPACT for UID We hypothesized that IMPACT helps the model generate outputs consistent with the UID hypothesis, where hesitation occurs when surprisal or entropy becomes large. We validate this empirically: IMPACT decreases the average entropy by 0.011 every hard step during the GSM8K generation of Llama-3.1-8B-Instruct. Furthermore, IMPACT reduces surprisal kurtosis–a widely used statistic for detecting outliers–by nearly half, from 0.605 to 0.363. This reduction suggests a more even distribution of surprisals, contributing to more stable and consistent model performance.

6 Conclusion

To address the limitations of the hesitation-based PACT method, we introduced a cache-aware alternative that perturbs network weights instead of the inputs. Our method, IMPACT, preserves the training-free advantage of HARP while improving efficiency and ensuring determinism by eliminating the randomness caused by dropout. We validated improvements in both performance and efficiency, complementing existing approaches focused on performance and efficiency. This makes higher performance and computational speed accessible to real-world applications and to a wider audience, including those with consumer-grade GPUs and limited memory.

7 Limitations

A more precise tuning of the perturbation magnitude to balance the trade-off between performance and computational efficiency could further enhance performance. We consider more advanced model compression techniques as a separate direction for future work.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00414981), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00077/RS-2022-II220077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data).

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda et al. Askell. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation. Zenodo.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.

T. Jaeger and Roger Levy. 2006. Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Donghyun Lee, Jaeyong Lee, Genghan Zhang, Mo Tiwari, and Azalia Mirhoseini. 2024. CATS: Context-Aware Thresholding for Sparsity in Large Language Models. In *First Conference on Language Modeling*.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4. *Preprint*, arXiv:2304.03439.

James Liu, Pragaash Ponnusamy, Tianle Cai, Han Guo, Yoon Kim, and Ben Athiwaratkun. 2024. Training-Free Activation Sparsity in Large Language Models. *Preprint*, arXiv:2408.14690.

Bruce T. Lowerre. 1976. *The Harpy Speech Recognition System*. Ph.D. thesis, Carnegie Mellon University, USA.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling test-time compute optimally can be more effective than scaling LLM parameters. In *The Thirteenth International Conference on Learning Representations*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Romain Storaï and Seung-won Hwang. 2024. HARP: Hesitation-Aware Reframing in Transformer Inference Pass. *Preprint*, arXiv:2412.07282.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, and Klaus Macherey et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Preprint*, arXiv:1609.08144.

Dataset	Standard Deviation
GSM8K	0.86
LogiEval	0.63
CsQA	0.27
GPQA	0.28
BBH	0.24
MMLUPro	–
CNN/DM	0.45
LAMBADA	0.63

Table 5: Standard deviation of Llama-3.1-8B-Instruct using HARP across all the datasets. MMLU Pro has not yet been evaluated on multiple seeds.

A HARP details

The standard Transformer forward pass processes an input sequence $\mathbf{x}$ by mapping it to token embeddings $\mathbf{e}$ and generating logits to predict the next token. HARP modifies this process by computing uncertainty on the logits during inference. Uncertainty is quantified using Shannon entropy (Shannon, 1948), computed from the normalized logits $\sigma(logits)$:

$$H(\sigma(logits)) = -\sum_{i=1}^{|V|} P(v_i \mid \mathbf{x}) \log_2 P(v_i \mid \mathbf{x}) \tag{4}$$

where $P(v_i \mid \mathbf{x})$ represents the probability of token v_i as the next token, given the input $\mathbf{x}$.

If the entropy H remains below a predefined threshold θ, the model is considered confident and the original logits $logits$ are returned. Otherwise, the model identifies the step as hard and triggers an additional forward pass.

HARP reframes the input embeddings $\mathbf{e}$ by applying dropout at a rate δ, producing reframed embeddings $\hat{\mathbf{e}}$:

$$\hat{\mathbf{e}} = \textsc{Dropout}(\mathbf{e}, \delta). \tag{5}$$

The second forward pass uses the reframed input embeddings $\hat{\mathbf{e}}$ and generates reframed logits $logits_r$, which are combined using a β-weighted sum with the original logits, as follows:

$$\beta \cdot logits + (1 - \beta) \cdot logits_r \tag{6}$$

B Proof of Dropout Altering Significant Portion of Tokens

Proof. Let the random variable X_i be defined as

$$X_i = \begin{cases} 1 & \text{at least one } x_i \text{ dimension is dropped,} \\ 0 & \text{otherwise.} \end{cases}$$

for $i \in [1, n]$, where n is the length of the input sequence.

$$Pr[X_i = 1] = 1 - (1 - \delta)^{|hidden|}$$

where δ is the dropout rate and $|hidden|$ is the dimensionality of the embeddings.

Let Y denote the total number of impacted tokens in the sequence. Then,

$$Y = \sum_{i=1}^{n} X_i$$

By linearity of expectation,

$$\begin{aligned} E[Y] &= \sum_{i=1}^{n} E[X_i] = n \cdot Pr[X_i = 1] \\ &= n \cdot \left(1 - (1 - \delta)^{|hidden|}\right) \end{aligned}$$

Since $|hidden|$ is typically very large in practice (4096 for Llama-3.1), $(1 - \delta)^{|hidden|}$ approaches 0 rapidly, making $Pr[X_i = 1]$ close to 1. Thus, the expected value of Y (the number of impacted tokens) approaches n for sufficiently large $|hidden|$. □

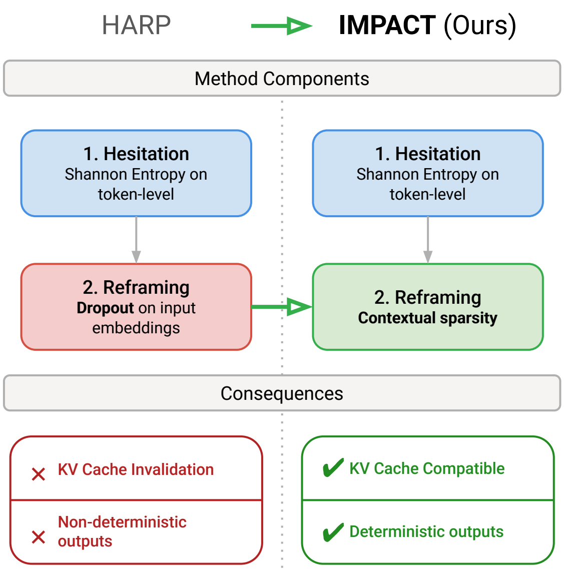

Figure 2: Side to side comparison of our method (IMPACT) with the other hesitation-based PACT method (HARP). Our method overcomes the two limitations of HARP by changing its reframing process.

C Illustration of HARP vs IMPACT

Figure 2 compares HARP and IMPACT.

D Memory usage

HARP requires twice the memory of the original model–Llama-3.1-8B-Instruct's GSM8K generation demands 16GB for the original model but 32GB for HARP. In contrast, IMPACT achieves performance improvements with only an 18GB requirement, making it particularly valuable in scenarios where HARP's memory overhead is prohibitive.

E Experimental Settings

Baselines We use Llama-3.1-8B-Instruct and Llama-3.3-70B-Instruct (Dubey et al., 2024) as our base models. We compare IMPACT with three baselines: the original model using KVCache, HARP (without KVCache), and beam search. All evaluations are conducted using a greedy decoding setting (temperature of 0). All experiments are conducted on A6000 GPUs. Latency is evaluated on the whole dataset using a batch size of 1 and a concurrency of 1, following the convention of HARP.

Datasets Our evaluation is implemented using the LM-Evaluation-Harness (Gao et al., 2021) framework. We focus on tasks that require reasoning, as we hypothesize that hesitation during inference is particularly beneficial for such tasks. Our evaluation includes the following datasets: **GSM8K** (Cobbe et al., 2021), a mathematical problems benchmark; **LogiEval** (Liu et al., 2023), a logical reasoning dataset; and **Commonsense QA** (Talmor et al., 2019), a general commonsense understanding evaluation. We also evaluate with chain-of-thought prompting (Wei et al., 2024)–a modern approach to reasoning–using the following datasets: **BIG-Bench Hard** (BBH) (Suzgun et al., 2023), **GPQA** (Rein et al., 2024), and **MMLU Pro** (Wang et al., 2024). Additionally, we include two non-reasoning tasks for continuous comparison and generalization: **CNN DailyMail** (CNN/DM) (Nallapati et al., 2016), a summarization task; and **LAMBADA** (Paperno et al., 2016), a short-word generation task. Following Storaï and Hwang (2024), we use subsets of Commonsense QA, MMLU Pro, and CNN DailyMail. We specify the settings used for each dataset in Appendix H. These datasets cover a range of output formats, such as free-text reasoning, multiple-choice questions, and one-word generation. Licenses of the artifacts are specified in Appendix I.

Metrics For each task, we report the accuracy and the inference time on the full datasets. For some datasets, we report the exact match (EM), while we report the rouge-1 score for CNN DailyMail. Since HARP introduces randomness in its process, we perform evaluations with multiple random seeds and report the average. The standard deviation of each dataset is reported in Table 5.

Hyperparameters The hyperparameters for HARP are set as follows: hesitation threshold $\theta = 1.2$, dropout rate $\delta = 0.2$, and $\beta = 0.5$ weighted sum. Storaï and Hwang (2024) conducted their experiments using quantized models and subsets of datasets. After conducting preliminary experiments, we observed that $\theta = 1.2$ resulted in better accuracy compared to the value of $\theta = 1.0$ used in their work when evaluated on the full model and full dataset.

For IMPACT, each t for contextual sparsity is pre-calculated with TEAL (Liu et al., 2024) to target 30% of sparsity. We set the hesitation threshold as $\theta = \log 2$.[2] These hyperparameter values have been determined to perform well in our setting, but we leave an exploration of their broader influence to future work.

For beam search, we use a beam size b of 3, and length-normalization (Wu et al., 2016) of 0.6, to be consistent with previous work (Storaï and Hwang, 2024).

F Stability of IMPACT over θ

θ	**GSM8K**
$2 \log 2$	78.70
1	78.77
$\log 2$	78.85

Table 6: GSM8K performance of Llama-3.1-8B-Instruct over different θ values.

Table 6 shows that the performance of IMPACT is stable across various θ values.

G Choice of Teal Sparsity For IMPACT

Table 7 validates the choice of 30% TEAL sparsity for IMPACT.

[2] We used $\log_e$ for entropy calculation in our implementation, while HARP formulated entropy with base 2, which we followed in the writing. Conversion between these two formulas introduced $\log 2$.

TEAL sparsity	GSM8K
10%	78.09
20%	78.01
30%	78.85
40%	77.26

Table 7: GSM8K accuracy of Llama-3.1-8B-Instruct with different TEAL sparsity levels.

H Datasets Settings

- GSM8K: 5-shot
- LogiEval: 1-shot
- Commonsense QA: zero-shot. We adapted the evaluation from loglikelihood to generation, to match with Storaï and Hwang, 2024.
- BIG-Bench Hard: 3-shot
- GPQA: 1-shot
- MMLU Pro: zero-shot
- CNN DailyMail: zero-shot.
- LAMBADA: zero-shot. Similarly as for Commonsense QA, we adapted the evaluation to word generation to match with HARP's evaluation.

I Artifact Licenses

Datasets The datasets are downloaded from the Hugging Face Datasets library (Apache License 2.0). The LM-Evaluation-Harness is based on the MIT License. The licenses for the specific datasets used are as follows:

- GSM8K: MIT License
- LogiEval: Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
- Commonsense QA: MIT License
- BIG-Bench Hard: MIT License
- GPQA: Creative Commons Attribution 4.0
- MMLU Pro: Apache License 2.0
- CNN DailyMail: Apache License 2.0
- LAMBADA: Modified MIT License (GPT2)

Llama Models The Llama models (Llama 3 Community License) are used with the Hugging Face Transformers library (Apache License 2.0). TEAL is licensed under the MIT License.