# Extending AutoCompressors via Surprisal-Based Dynamic Segmentation

**Srivishnu Ramamurthi*** **Richard Xu***
**Raine Ma** **Dawson Park** **David Guo**
**Charles Duong**† **Vasu Sharma**† **Sean O'Brien**† **Kevin Zhu**†
Algoverse AI Research
srivishnur@ucla.edu, richardxu257@gmail.com, charles_duong@brown.edu

## Abstract

The long-context bottleneck of transformer-based language models can be addressed via context compression frameworks such as AutoCompressors, which distill tokens into **soft prompts** but silently assume uniform information density. We revisit this assumption and introduce dynamic segmentation by partitioning the input whenever the cumulative token-level **surprisal** exceeds a threshold $\tau$, yielding segments with balanced information before **summary vector** generation. We show that dynamically adjusting segment boundaries based on surprisal enables better alignment between the original and soft prompts for prediction and inference. Experimental results show that our surprisal-based segmentation outperforms a pretrained baseline model and the randomized segmentation AutoCompressor baseline with regard to cross-entropy loss and in-context learning (ICL) accuracy.

## 1 Introduction

Context window limitations hinder long-context fine-tuning and inference in transformer-based language models [Vaswani et al., 2017] due to memory and compute constraints [Wang et al., 2024]. To mitigate this, compression methods that reduce input complexity have been introduced, falling into two broad categories: either hard prompt or soft prompt techniques [Li et al., 2024].

**Hard prompts** are discrete natural language sequences consisting of tokens from a language model's vocabulary [Sennrich et al., 2016]. While hard prompts are easily interpretable, they often fail to concisely express semantic intent. **Soft prompts** are vectors with the same dimensions as token embeddings in the language model's dictionary [Zhao et al., 2023]. While soft prompts provide less interpretability compared to hard prompts, they capture semantic nuance more concisely.

Existing soft prompt methods such as GIST tokens [Mu et al., 2023], ICAE [Ge et al., 2024], and AutoCompressors [Chevalier et al., 2023] distill long inputs into soft tokens but assume uniform information density via constant token budgets or randomized segmentation—a flawed assumption given natural language's uneven semantic information distribution [Yu et al., 2016]. Information-theoretic approaches such as LLMLingua [Jiang et al., 2023] and Selective Context [Li et al., 2023] have shown that token-level perplexity or self-information can effectively identify semantically important input regions, but apply only to hard prompts.

DAST [Chen et al., 2025] similarly implements dynamic allocation of soft tokens and may be considered parallel work, however, it is built on the Activation Beacon framework [Zhang et al., 2024] utilizing a different compression schema. We believe the implementation of DAST leaves room for improvement in method details and depth of experiments.

*Equal contribution
†Senior author

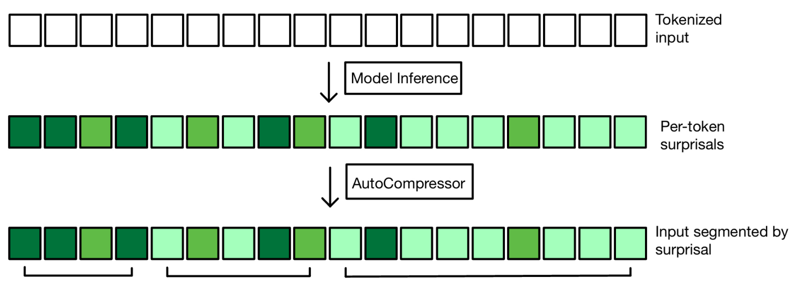


Figure 1: When fine-tuning AutoCompressors, we first quantify information density by obtaining per-token surprisals via baseline model inferencing on the tokenized input. We dynamically segment the input based on these surprisals, accumulating until a surprisal threshold $\tau$ is reached, resulting in variable-length segments and a variable number of segments per input sequence. These segments are then compressed into summary vectors, which are passed onto subsequent compression steps as soft prompts for the previous context.

We aim to bridge this gap by proposing a method that extends the AutoCompressor framework to incorporate surprisal-based dynamic segmentation. Specifically, we show that input segments of similar cumulative information produce more useful compressed representations when compressed into summary vectors, allowing for better language modeling performance on a variety of tasks.

## 2 Related Work

We adopt the AutoCompressor framework from Chevalier et al. [2023], which builds on the recurrent memory transformer (RMT) architecture [Bulatov et al., 2022] to compress plain text into short soft prompts known as **summary vectors** [Lester et al., 2021]. The tokenized input text is split into segments, with lengths randomly determined given a fixed hyperparameter of the number of segments. Segment lengths are guaranteed to be within the model's context window length. After generating summary vectors, the vectors are then prepended to all subsequent segments to recursively generate summary vectors over the entire segmented input.

Methods for extending a model's context window have been developed by previous work, such as RoPE-based scaling (Chen et al. [2023], Rozière et al. [2024], Ding et al. [2024], Zhu et al. [2025]) and utilizing different types of embeddings [Sun et al., 2023b]. Non-transformer based architectures have also been proposed (Peng et al. [2023], Sun et al. [2023a]), allowing for extended context windows. However, these modifications do not perform well at longer scales or at a foundational level.

Other compression methods have been explored to tackle long-context input. Semantic Compression [Fei et al., 2024] utilizes graph-based chunking based on topic to dynamically compress context, but focuses on hard prompt compression through summarization techniques. DoDo [Qin et al., 2024] approaches compression architecturally by dynamically compressing the context via a trainable selector and compressor module to select and compress the most important hidden states in each layer to reduce computational intensity while maintaining model performance. Our work addresses the issue from the perspective of reducing input complexity via soft prompts.

## 3 Method

We incorporate our custom segmentation methodology into the original AutoCompressors framework. This creates a distinct fine-tuning process from fine-tuning on randomly split input segments, allowing for better organization of the segments' information content.

### 3.1 Framework

AutoCompressors are fine-tuned on base models and split long documents into a series of segments $S_1, \ldots, S_n$ with variable lengths constrained to fit within the model's context window. For each token $x_t$ in a segment $S_i$ with $m_i$ tokens, the model is trained with the unsupervised objective of minimizing cross-entropy loss when conditioned on the previous tokens $x_1, \ldots, x_{t-1}$ and the previous summary vectors $\sigma_{<i}$ :

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{n} \sum_{t=1}^{m_i} \log p\left(x_t \mid x_1, \ldots, x_{t-1}, \sigma_{<i}\right)$$

over all segments and total number of tokens $N$. We follow this training objective when incorporating our method to fine-tune AutoCompressors.

### 3.2 Surprisal-Guided Segmentation

Our main contribution is implementing surprisal-based segmentation. The **surprisal** of a token $x_t$ is the negative log probability of the token appearing given the preceding context [Ji et al., 2023]:

$$\text{Surprisal}(x_t) = -\log P(x_t \mid x_{<t}).$$

Surprisal captures the model's uncertainty about the token in its generative context, thereby representing the information contained within the token; higher surprisal corresponds to more information, and lower surprisal corresponds to less information.

**Token-level Segmentation.** Given a tokenized input sequence $X = [x_1, x_2, \ldots, x_n]$, we define a segment $S_j = [x_{s_{j-1}+1}, \ldots, x_{s_j}]$ by computing per-token surprisal via baseline model inferencing and accumulating tokens until a fixed threshold $\tau$ is exceeded:

$$s_j := \min \left\{ k \in \{s_{j-1}+1, \ldots, n\} \;\middle|\; \sum_{i=s_{j-1}+1}^{k} \text{Surprisal}(x_i) \geq \tau \right\}$$

where $s_{j-1}+1$ and $s_j$ are the start and end indices respectively of segment $S_j$. If the length of the segment exceeds the model's context window before the threshold is reached, we simply end the segment and begin a new segment. This procedure creates segments of roughly equal cumulative surprisal. Unlike the original methodology, which specifies a fixed number of segments for each training substep, we allow for a variable number based on the information distribution of the input, creating more flexibility in the fine-tuning process.

## 4 Experiments

### 4.1 Experimental Setup

We fine-tune an AutoCompressor model on a pre-trained OPT model [Zhang et al., 2022] with 1.3 billion parameters, fine-tuning on 6K-token sequences from the Wikipedia subdomain of the Pile dataset [Gao et al., 2020] with a surprisal threshold of $\tau = 1500$. This threshold was heuristically determined based on the total cumulative surprisal across sample input sequences.

Fine-tuning was done with 2-3 NVIDIA H100 GPUs each with 80 GB of memory over 50 hours, with one GPU solely dedicated to baseline model inferencing to obtain surprisal calculations. To ensure that the input would not exceed the base model's context window length during inferencing, we apply the extended full attention methodology introduced in the original work via extension of positional embeddings. Specifically, positional embeddings are reused beyond the model's context window length to allow for longer input sequences.

We evaluate our model by evaluating the out of domain cross-entropy loss on 6K-token sequences from the Gutenberg subdomain (consisting of various works of literature) of the Pile dataset. We split the input sequences into segments of 2,048 tokens except for the last segment, which has fewer tokens. Then, we compress all segments except the last, pass their summary vectors forward as soft prompts, and evaluate cross-entropy loss on the final segment.

| Model | Cross-Entropy | 7 | 46 | 209 | 1071 | 4489 | 19972 |
|---|---|---|---|---|---|---|---|
| OPT-1.3b | 4.20 | 62.61 | 55.53 | 52.25 | 74.40 | 53.50 | 59.07 |
| Baseline AC | 2.66 | 67.16 | 63.50 | 60.46 | 71.30 | 55.30 | **62.71** |
| Dynamic AC | **2.61** | **69.12** | **69.82** | **64.88** | **76.80** | **58.50** | 62.70 |

Table 1: Evaluation results for a baseline OPT-1.3b model, baseline AutoCompressor (AC), and our dynamic AutoCompressor (AC). We evaluate cross-entropy loss on the Gutenberg subdomain and ICL accuracy on the AG News dataset with 6 different seeds.

We also evaluate in-context learning (ICL) accuracy on the AG News benchmark, which involves 4-way topic classification on news articles. Following the original implementation, we construct 10-shot prompts by sampling and concatenating 10 plain text training examples.

### 4.2 Results

We display our results in Table 1. We compare to a baseline OPT-1.3b model with extended full attention as well as an AutoCompressor model utilizing randomized segmentation as in the original methodology.

Our dynamic AutoCompressor achieves lower cross-entropy loss (2.61) compared to the baseline AutoCompressor (2.66) when evaluated on the out of domain Gutenberg dataset, demonstrating improved generative model prediction. On the AG News classification task, Dynamic AC outperforms the random segmentation baseline on most seeds, showing performance gain due to surprisal-aligned compression. For example, on Seed 7 and Seed 209, dynamic AutoCompressor improves accuracy in text classification by approximately +2% and +4.4%.

## 5 Limitations

We were unable to fine-tune larger models as AutoCompressors or fine-tune and evaluate on more subdomains due to budget and time constraints, potentially limiting generalizability. We were also unable to adjust the surprisal threshold $\tau$ as a hyperparameter for the same reasons. Future work should consider scaling to larger models and explore the effect of varying $\tau$ for optimization, as well as evaluation on a wider range of tasks. Furthermore, research is needed to consider the compression ratio presented by dynamic segmentation.

## 6 Discussion

We fine-tune an OPT model as an AutoCompressor using surprisal-based segmentation when partitioning input, determining segment boundaries by a surprisal threshold. We evaluate out of domain cross-entropy loss and ICL accuracy as compared to the pretrained baseline model and the randomized segmentation AutoCompressor, showing improved model performance. While gains are not uniform across all possible seeds or downstream tasks, future experiments may provide deeper insights.

## References


Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, volume 35, pages 11079–11091, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/47e288629a6996a17ce50b90a056a0e1-Paper-Conference.pdf`.

Shaoshen Chen, Yangning Li, Zishan Xu, Yinghui Li, Xin Su, Zifei Shan, and Hai-tao Zheng. Dast: Context-aware compression in llms via dynamic allocation of soft tokens. *arXiv preprint arXiv:2502.11493*, 2025. URL `https://arxiv.org/abs/2502.11493`.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023. URL `https://arxiv.org/abs/2306.15595`.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore, 2023. doi: 10.18653/v1/2023.emnlp-main.232. URL `https://aclanthology.org/2023.emnlp-main.232/`.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens, 2024. URL `https://arxiv.org/abs/2402.13753`.

Weizhi Fei, Xueyan Niu, Pingyi Zhou, Lu Hou, Bo Bai, Lei Deng, and Wei Han. Extending context window of large language models via semantic compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5169–5181, Bangkok, Thailand, 2024. doi: 10.18653/v1/2024.findings-acl.306. URL `https://aclanthology.org/2024.findings-acl.306/`.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL `https://arxiv.org/abs/2101.00027`.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model, 2024. URL `https://arxiv.org/abs/2307.06945`.

Shaoxiong Ji, Wei Sun, and Pekka Marttinen. Content reduction, surprisal and information density estimation for long documents. *arXiv preprint arXiv:2309.06009*, 2023. URL `https://arxiv.org/abs/2309.06009`.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL `https://aclanthology.org/2023.emnlp-main.825/`.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL `https://aclanthology.org/2021.emnlp-main.243/`.

Yucheng Li, Bo Dong, Frank Guérin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore, 2023. doi: 10.18653/v1/2023.emnlp-main.391. URL `https://aclanthology.org/2023.emnlp-main.391/`.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. *arXiv preprint arXiv:2410.12388*, 2024. URL `https://arxiv.org/abs/2410.12388`.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, pages 19327–19352, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3d77c6dcc7f143aa2154e7f4d5e22d68-Paper-Conference.pdf.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era, 2023. URL https://arxiv.org/abs/2305.13048.

Guanghui Qin, Corby Rosset, Ethan Chau, Nikhil Rao, and Benjamin Van Durme. Dodo: Dynamic contextual compression for decoder-only lms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 9961–9975, Bangkok, Thailand, 2024. doi: 10.18653/v1/2024.acl-long.536. URL https://aclanthology.org/2024.acl-long.536/.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024. URL https://arxiv.org/abs/2308.12950.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, 2016. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009/.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023a. URL https://arxiv.org/abs/2307.08621.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14590–14604, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.816. URL https://aclanthology.org/2023.acl-long.816/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv preprint arXiv:2402.02244*, 2024. URL https://arxiv.org/abs/2402.02244.

Shuiyuan Yu, Jin Cong, Junying Liang, and Haitao Liu. The distribution of information content in english sentences. *CoRR*, abs/1609.07681, 2016. URL http://arxiv.org/abs/1609.07681.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. *arXiv preprint arXiv:2401.03462*, 2024. URL https://arxiv.org/abs/2401.03462.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Wenbo Zhao, Arpit Gupta, Tagyoung Chung, and Jing Huang. SPC: Soft prompt construction for cross-domain generalization. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP)*, pages 118–130, Toronto, Canada, 2023. doi: 10.18653/v1/2023.repl4nlp-1.10. URL https://aclanthology.org/2023.repl4nlp-1.10/.

Wenqiao Zhu, Chao Xu, Lulu Wang, and Jun Wu. Psc: Extending context window of large language models via phase shift calibration, 2025. URL https://arxiv.org/abs/2505.12423.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claim that adjusting segment boundaries based on surprisal improves alignment between original and soft prompts and performance is reflected in the results section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion and conclusion section discuss the limitations (such as budget and time constraints that limited amount of experiments).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There are no theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We specify the theory, datasets, models, parameters, surprisal threshold, and seeds needed to reproduce the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We are unable to provide open access to our code at the moment.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: A lot of the core settings and details are present in the original AutoCompressors paper, and we include additional necessary information for the tests we run.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Experiment statistical significance is not shown with this paper. This was mainly due to the cost and time required to run each individual experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the specific GPUs used as well as the amount of time it took to fine-tune.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: No work in the paper overlaps with the ethical concerns present in the Code of Ethics. The datasets used are not deprecated.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There are no substantial direct societal impacts of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly use and credit the AutoCompressor framework as well as the models used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper currently does not release any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.