Precise Length Control for Large Language Models

Author A[1], **Author B**[1], **Author C**[1]
[1]Affiliation
Author.A@email.com, Author.B@email.com, Author.C@email.com

Abstract

Large Language Models (LLMs) have become integral components of production systems, with applications ranging from chatbots like ChatGPT to tasks such as summarisation and question answering. However, a significant challenge with LLMs is the unpredictability of response length, which is particularly problematic in tasks requiring varying levels of detail, such as document summarisation. Here we present a method to adapt existing LLMs to allow control of response length. We achieve this by extending the length-difference positional encoding (LDPE) proposed by (Takase and Okazaki, 2019) to decoder-only transformer architectures. Our approach, termed offset reverse positional encoding (ORPE), uses a positional encoding that counts down from a predetermined response length. Fine-tuning with ORPE enables the model to learn to structure its responses to terminate at a given length. Our results, obtained from tasks such as question answering and document summarisation, demonstrate that ORPE provides precise control of the response length during inference.

1 Introduction

In recent years, Large Language Models (LLMs) have become essential components of various production systems, revolutionising the way we interact with technology. From chatbots like ChatGPT to applications in document summarisation, question answering, and content rewriting, LLMs have demonstrated remarkable capabilities in understanding and generating human-like text (Devlin et al., 2018; Brown et al., 2020). These models, trained on vast amounts of data, have the potential to transform industries and enhance user experiences across a wide range of domains.

However, despite their impressive performance, LLMs are unpredictable in terms of the length of their generated responses even when prompted with length specifications. This is particularly problematic in tasks that require varying levels of detail, such as document summarisation (Gambhir and Gupta, 2017). In situations where concise summaries are needed, an LLM might generate overly lengthy responses, while in cases where more comprehensive summaries are desired, the model might produce insufficient detail. This lack of control over response length limits the practical applicability of LLMs in real-world scenarios.

To address this limitation, we present a novel method to adapt existing LLMs and enable the control of output length. Our approach, termed offset reverse positional encoding (ORPE), extends the length-difference positional encoding (LDPE) proposed by (Takase and Okazaki, 2019) to decoder-only transformer architectures. This is achieved by incorporating a countdown mechanism that starts from a predetermined response length. During the training process the model considers this countdown as a signal and implicitly learns the concept of "token budget", i.e. how many tokens are remaining to generate an appropriate response. A user-specified token budget can then be provided at inference time, thereby facilitating the generation of responses with specified lengths for downstream tasks.

The main contributions of our paper are as follows:

1. We introduce ORPE, a method to adapt existing LLMs for controlling output length by extending LDPE to decoder-only transformer architectures.

2. We demonstrate the effectiveness of our approach through experiments on various tasks, including question answering and document summarisation.

3. Our results confirm that ORPE successfully controls response length and generates varied response versions with specified lengths, enhancing the flexibility and applicability of LLMs in real-world scenarios.

The remainder of this paper is organised as follows. Section 2 provides an overview of related

work in the field of LLMs and output length control. Section 3 describes our proposed methodology, including the inversion of positional encodings and the countdown mechanism. Section 4 details our experimental setup and the tasks used for evaluation. Section 5 presents and discusses the results obtained from our experiments. Section 6 concludes the paper and outlines potential future directions for research. Finally, Section 7 details some of the limitations of the work which could be addressed to improve the method.

2 Related Work

Positional encoding is used to directly provide transformers with information about the absolute or relative position of each token (Vaswani et al., 2017; Shaw et al., 2018). Multiple techniques have been proposed to augment the standard positional encoding with a secondary encoding containing useful positional information. For example using a secondary encoding to align digit positions of operands has been shown to provide greatly improved numerical calculation capabilities (McLeish et al., 2024). More generally, an additional learned context dependent encoding allows models to perform a variety of tasks requiring different levels of positional understanding (Golovneva et al., 2024). Here we explore using a secondary encoding scheme to allow direct control over the models response length.

Previous approaches for length control in text generation have primarily focused on encoder-decoder models for tasks like summarisation. For instance, (Kikuchi et al., 2016) proposed methods to initialise the LSTM cell of the decoder with a learnable vector based on the desired length (LenInit) or input an embedding representing the remaining length at each decoding step (LenEmb). This work was extended in (Yu et al., 2021) (LenAtten) to separate the length information from the decoder hidden states to better exploit the remaining length information.

Most relevant to our work, (Takase and Okazaki, 2019) introduced the length-difference positional encoding (LDPE), which modifies the sinusoidal positional encoding in transformer models to incorporate the remaining length to the terminal position.

While effective for encoder-decoder architectures, these methods are not directly applicable to the decoder-only LLMs that have become prevalent in recent years. Decoder-only LLMs like GPT-3 (Brown et al., 2020) do not have an explicit length input during generation, making length control more challenging. Some recent work has explored length control in LLMs in the context of dialogue generation. (Roller et al., 2021) used a special end-of-text token to control response length, penalising or encouraging its generation based on the desired length. However, this provides only coarse-grained control and may lead to unnatural responses.

Work has also been performed exploring prompt-based methods of controlling LLM output lengths. For example, (Jie et al., 2023) uses reinforcement learning to finetune decoder only LLMs using a rule-based reward model. The type of prompt (more than, less than, equal to, between) determines the reward function, which is calculated using the target and output length. One limitation of this model is that it does not explicitly encode any information about the remaining token budget at each generation step, which could lead to less semantically complete responses.

3 Method

In this section, we describe our proposed method for controlling the output length of LLMs by adapting the length-difference positional encoding (LDPE) proposed by (Takase and Okazaki, 2019) to the decoder-only transformer architecture. Our approach consists of two main components: (1) adding offset reverse positional encodings during the fine-tuning process and (2) utilising the offset reverse positional encodings at inference time to control the length of the generated response.

3.1 Offset Reverse Positional Encodings

In the transformer architecture (Vaswani et al., 2017), positional encodings are used to provide information about the position of each token in the input sequence. The LDPE scheme, as introduced by (Takase and Okazaki, 2019), encodes the remaining distance to the end of the generated sequence. We propose an offset reverse positional encoding (ORPE) that adapts LDPE to the decoder-only LLM setting.

To represent the reverse positional encodings in the embedding space of a LLM we use the standard sinusoidal encodings (Takase and Okazaki, 2019; Vaswani et al., 2017):

$$PE_{(i,2k)} = \sin\left(\frac{i}{10000^{\frac{2k}{d}}}\right), \tag{1}$$

$$PE_{(i,2k+1)} = \cos\left(\frac{i}{10000^{\frac{2k}{d}}}\right). \tag{2}$$

Here i represents the tokens position in the text sequence, k is the index within the embedding dimension, and d is the total number of dimensions in the embedding.

In order to utilise the ORPE we adjust the order that the positional encodings are used by firstly reversing them to count down towards $i = 1$, and secondly introducing an offset so that the countdown starts from the end of the input prompt.

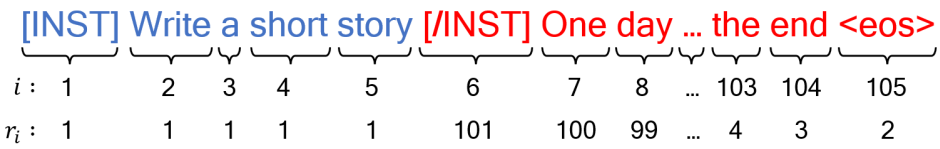

Figure 1: A simplified diagram of ORPE encoding for a target response length of 100 tokens and a question length of 5 tokens (i.e. $L = 100$ and $n = 5$). Offset reverse positional encodings $PE_{(r_i,k)}$ are added to the input token embeddings with a encoding of $r_i = 1$ added to each token in the prompt part of the text, and a countdown from $r_i = L + 1$ to $r_i = 2$ added to the response.

Formally, let $(X, Y) = x_1, \ldots, x_n, y_{n+1}, \ldots y_{n+L}$ be a sequence of tokens consisting of a prompt and response, where n is the prompt length and L is the response length. We augment the i^{th} token embedding by adding the corresponding ORPE encoding, $ORPE_i = PE_{(r_i,k)}$, where r_i is given by:

$$r_i = \begin{cases} 1, & \text{if } i \leq n \\ L - (i - n) + 2, & \text{otherwise} \end{cases} \quad (3)$$

Here L is the desired length of the generated response and $n+1$ is the position where the model's response begins. By offsetting the start of the reverse positional encoding to coincide with the beginning of the model's response, we ensure that the countdown only applies to the generated text, not the input prompt. This allows the model to distinguish between the context it is conditioning on and the text it should generate, while still benefiting from the length-controlling properties of LDPE.

The arrangement of the ORPE is demonstrated in Figure 1, using word-level tokenization for illustrative purposes only.

3.2 Fine-tuning with Offset Reverse Positional Encodings

To incorporate ORPE into the LLM, we propose a fine-tuning process that includes them as additional input features. During fine-tuning, we add ORPE to the token embeddings before passing them through the transformer layers.

Given a finetuning dataset of prompt-response pairs $\mathcal{D} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_m, Y_m)$, where X_i is the prompt and Y_i is the corresponding target response, we fine-tune the LLM to minimise the following loss function:

$$\mathcal{L} = -\sum_{i=1}^{m} \sum_{j=1}^{|Y_i|} \log P(y_{i,j} | X_i, R_{i,<j}, y_{i,<j}) \quad (4)$$

where $y_{i,j}$ is the j-th token of the target response Y_i, and $y_{i,<j}$ denotes the tokens preceding $y_{i,j}$ with corresponding ORPE encodings $R_{i,<j}$. By including ORPE during fine-tuning, the LLM learns to generate responses of the desired length, as the model can utilise the countdown information provided by the encodings. To further enhance the models ability to lock on to the ORPE signal, the response is considered to start at the instruct token, as illustrated in Figure 1.

3.2.1 Adapting Existing LLMs

Directly adding ORPE to the token embeddings of an off-the-shelf LLM may not yield optimal results due to differences in scale between the learned embeddings and the positional encodings. To address this, we introduce a scaling term that balances the magnitude of ORPE with the token embeddings. Let $E = e_1, e_2, \ldots, e_n$ be the token embeddings of the input sequence X. We compute the scaled ORPE R' as follows:

$$R' = R \cdot \frac{|E|_F}{|R|_F} \quad (5)$$

where $|E|_F$ and $|R|_F$ denote the Frobenius norm of the token embeddings and ORPE, respectively.

By scaling ORPE with the ratio of the Frobenius norms, we ensure that the positional encodings have a similar magnitude to the token embeddings, allowing the LLM to effectively incorporate the length control signal during fine-tuning and inference.

The final input to the LLM, $\hat{E}$, is computed by adding the scaled ORPE to the token embeddings:

$$\hat{E} = E + R' \quad (6)$$

This adaptation allows existing LLMs to benefit from the length-controlling properties of ORPE without requiring significant modifications to the model architecture.

3.3 Inference with Length Control

At inference time, we control the length of the generated response by providing the appropriate ORPE. Given a prompt X and a desired response length L, we generate ORPE R and normalize it according to Equation 5 before adding R to the token embeddings. The LLM then generates the response token by token, conditioning on the prompt, ORPE, and the previously generated tokens. The generation process continues until an end-of-sequence token is generated, which given the fine-tuning process should be approximately around the target length. By manipulating ORPE,

we can control the length of the generated response, enabling the LLM to produce concise or verbose answers as required by the application.

3.4 Max New Tokens++

While generating responses with exact lengths can be useful in certain applications, there are scenarios where specifying upper-bound on the desired length is more appropriate. The widely-used transformers library (Wolf et al., 2019) achieves this through a simple generation stopping criterion: generated tokens $\geq$ max new tokens. However, this approach does not provide the model with an awareness of its remaining "token budget" during the generation process, which may lead to suboptimal response quality.

To address this limitation, we propose a content-aware length control method that allows the model to learn when to terminate the response based on the input prompt and the generated content. During training, we introduce a random shift that is added to the target response length. This shift is sampled from a truncated half-normal distribution HalfNormal(σ), where the truncation point is determined by the maximum allowed response length $L_{\max}$ and the true response length L:

$$\text{shift} \sim \min(\text{HalfNormal}(\sigma), L_{\max} - L) \quad (7)$$

The sampled shift is then incorporated into the positional encoding of the response tokens as follows:

$$r_i = \begin{cases} 1, & \text{if } i \leq n \\ L - (i - n) + 2 + \text{shift}, & \text{otherwise} \end{cases} \quad (8)$$

where $n + 1$ is the position of the first response token.

By exposing the model to various target response lengths for the same input prompt during training, we encourage it to learn to generate coherent and relevant responses that can be shorter than the true response length. The half-normal distribution is chosen because it yields non-negative shifts with a higher probability for smaller values, ensuring that the model still observes the original target lengths more frequently, providing a valuable learning signal.

To further improve the model's ability to generate responses with varying lengths, we employ a curriculum learning approach (Bengio et al., 2009) by gradually increasing the scale parameter σ of the half-normal distribution throughout the training process. Initially, σ is set to a small value, resulting in shifts that are close to zero. This encourages the model to focus on learning to generate responses that closely match the true response length. As training progresses, σ is slowly increased according to an exponential schedule:

$$\sigma_t = \sigma_0 \exp\left(\frac{t}{T} \log\left(\frac{\sigma_{\max}}{\sigma_0}\right)\right) \quad (9)$$

where σ_0 and $\sigma_{\max}$ are the initial and maximum values of σ, respectively, t is the current training step, and T is the total number of training steps. The values of σ_0 and $\sigma_{\max}$ are treated as hyperparameters and can be tuned to control the level of length variability and the trade-off between conciseness and informativeness.

Our updated *max new tokens++* method offers a principled approach to generating responses with a desired maximum length while allowing the model to learn when to terminate the response based on the input prompt and generated content. This enables the model to produce more concise and relevant responses compared to the naive generation stopping criterion, enhancing the flexibility and applicability of language models in real-world scenarios.

4 Experimental Setup

This section provides an overview of the models, data and hyperparameters used to evaluate the performance of the proposed ORPE method.

Data A combination of the OpenOrca (Lian et al., 2023) and MMLU (Hendrycks et al., 2021) datasets was used for training. A training set of 110,000 samples was constructed by combining 100,000 OpenOrca samples with 10,000 MMLU samples (from across all topics). These datasets were combined to cover a wider range of topics and sequence lengths, and to utilise both human and synthetically generated data. An evaluation set of 200 samples was constructed in the same manner, taking 100 samples from each dataset. All training and evaluation data was in English, including the datasets and benchmarks discussed in Sections 5.1.2 and 5.1.3.

Models Experiments were performed by fine-tuning both Mistral 7B (Jiang et al., 2023) and Llama3 8B (Meta, 2024). These models were chosen as they are exemplary representations of modern LLMs with robust architectures and significant parameter counts, whilst still being small enough to train on a single NVIDIA RTX A6000 GPU. The instruct versions of both models were used, and their forward pass methods were modified to add the appropriate ORPE encodings.

Baseline Models As a baseline for general model response quality we simply used the pre-trained Mistral 7B Instruct and Llama3 8B Instruct models (without ORPE added). Additionally as a baseline method of length control we fine-

tuned Mistral with the target response length included in the prompt, for example "Answer the following question in 112 tokens:".

Hyperparameters Both models were trained for a single epoch using the entire training dataset and a batch size of 1 with gradient accumulation of 5. All models were trained using the AdamW optimiser (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 1 \times 10^{-8}$, with a learning rate of 0.0003 and a linear learning rate schedule. These hyperparameter settings were obtained via manual experimentation and evaluation until satisfactory results were obtained.

Low Rank Adaptation Low Rank Adaptation (LoRA) (Hu et al., 2021) was employed to efficiently train all models. LoRA allows for efficient incorporation of the ORPE without overwriting the pretraining weights. A rank of 16, an α of 32 and a dropout rate of 0.05 was used to train all models.

5 Results

5.1 Exact Length Control

5.1.1 Q&A

To demonstrate the effectiveness of our method we evaluated its performance on the OpenOrca dataset. A set of 20 samples were selected and different length responses were generated for each, ranging from 10 to 200 tokens. The generated responses length was then compared to the target length. This analysis was performed for three different approaches: prompting without length control, fine-tuned prompting, and our proposed ORPE method. Figure 2 presents the results of this comparison.

As shown in the top-left panel of Figure 2, prompting without any length control results in a wide spread of response lengths, with many responses significantly deviating from the ideal length. This highlights the need for an effective length control mechanism in question-answering systems. The top-right panel of Figure 2 demonstrates the performance of fine-tuned prompting, where the model is fine-tuned with length information. While this approach improves the alignment between the target and response lengths compared to prompting without length control, there is still a noticeable deviation from the ideal response length for many samples. Note that these two experiments were performed with Mistral only as we expect the performance to be very similar for Llama.

In contrast, our ORPE method, shown in the bottom panels of Figure 2, achieves a near-perfect alignment between the target and response lengths. This showcases the effectiveness of our proposed method in controlling the exact length of the generated responses in a question-answering setting. Some example responses for a range of target lengths can be seen in Table 1.

5.1.2 Summarisation

To evaluate the performance of our content-aware length control method, we generated a dataset of summaries using the CNN/DailyMail dataset (Nallapati et al., 2016). We selected a subset of 25 articles from the training split and generated summaries for each article using five different system prompts, each targeting a specific summary length or style:

- A one-sentence summary
- A one-paragraph summary
- A long, detailed summary not exceeding the original article length
- A 100-word summary
- A short, concise summary

The summaries were generated using the OpenAI API with the GPT-3.5-turbo-0125 model (OpenAI, 2024). For each combination of article and system prompt, we recorded the generated summary, its length, and the corresponding system prompt and article. The resulting dataset was then used to evaluate the summarisation quality of our ORPE finetuned models, as well as a baseline Mistral model that was finetuned for prompt based length control. Each model was prompted to generate a summary of a full article from the CNN/DailyMail dataset, and the target length of the summary was set to the length of each of the 5 GPT-3.5 summaries. For the ORPE finetuned models the target length was given to the model only via the ORPE encodings, whereas for the prompt finetuned model the target length as was added to the prompt. The quality of the summary was assessed by calculating the BERT scores between the generated summary and the corresponding GPT-3.5 summary.

Figure 3 presents the mean BERT scores for the summarisation tasks, binned into different target response length ranges. We compared both the Mistral and Llama models with the ORPE finetuning, as well as the baseline Mistral model finetuned for length control via prompting (Mistral-Prompted). The results show that the summary quality was stable across all three models and for different response lengths. The average BERT scores across all summaries were 0.691 for Mistral-ORPE, 0.687 for Llama-ORPE, and 0.686 the baseline Mistral-Prompted model. The length accuracy of the summaries was significantly better for the ORPE finetuned models as shown in Figure 3. The mean length error between the target summary length and the models summary was 3.6 and

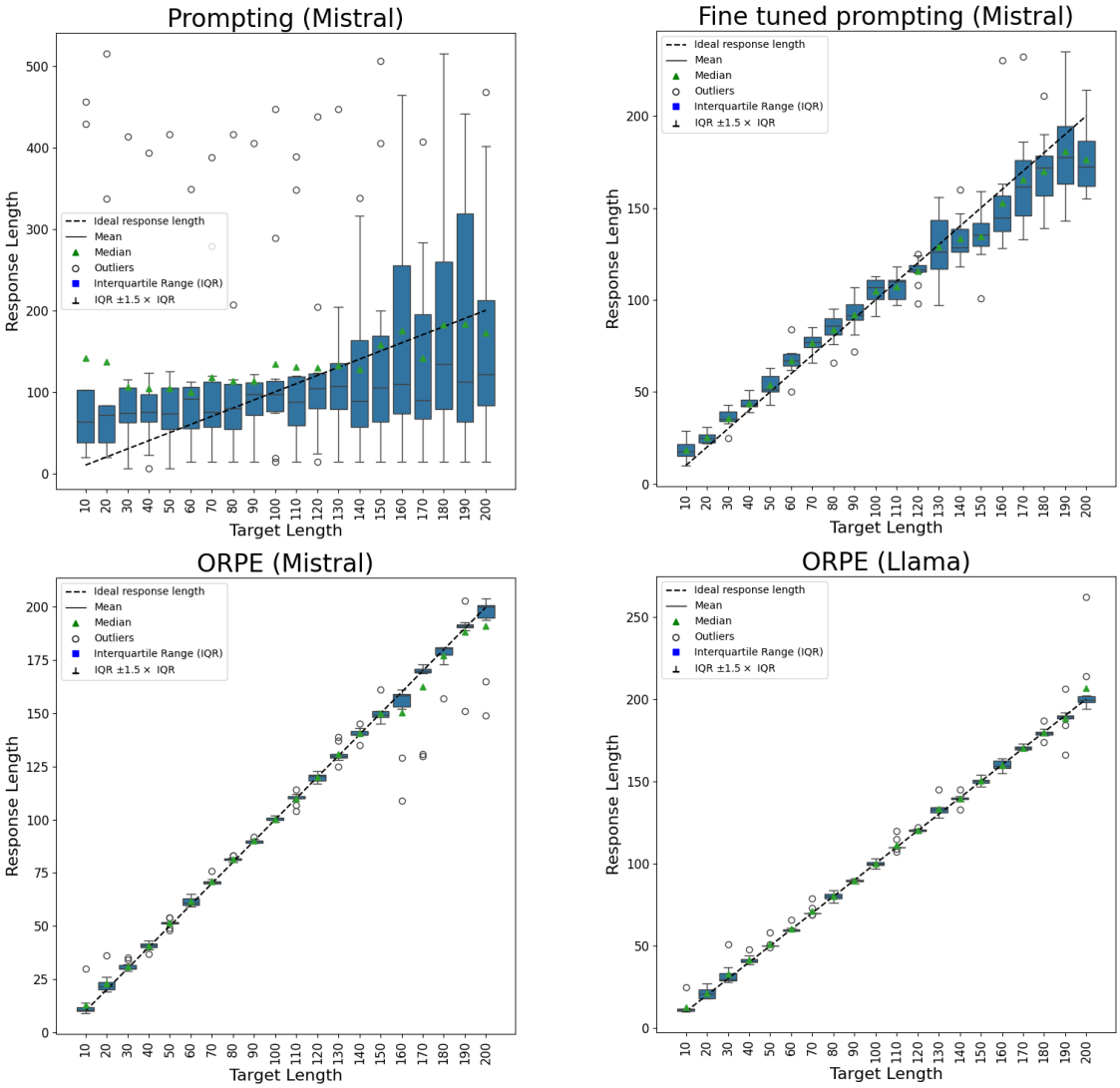

Figure 2: Comparison of target and response lengths for different length control approaches on a question-answering task. The ideal response length is indicated by the dashed line in each panel. **Top left:** results for prompting Mistral without length control. **Top right:** results for fine-tuned prompting Mistral. **Bottom left:** results for ORPE fine tuned Mistral 7B model. **Bottom right:** results for ORPE fine tuned Llama3 8B model.

4.6 tokens for Llama-ORPE and Mistral-ORPE respectively. The mean error for the Mistral-prompted was an order of magnitude larger at 28.1 tokens. Overall the results suggest that the models are still capable of generating high-quality summaries after finetuning with ORPE, and have developed the capability for fine-grained length control.

5.1.3 Benchmarking Response Quality

To identify any degradation in the quality of the model's responses after finetuning, we evaluated the models against a number of standard benchmarks. We ran all benchmarks using the LM Evaluation Harness (Gao et al., 2023). LM Evaluation Harness evaluates models against common benchmarks using the loglikehood of correct or incorrect responses from the model to score the task. The benchmarks evaluated were the Abstraction and Reasoning Corpus (ARC) (Lei et al., 2024), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2019), and WINOGRANDE (Sakaguchi et al., 2019).

We found that the ORPE finetuned models had an increased reliance on the instruction tokens used in finetuning. Thus instruction tokens were added into the appropriate places for each evaluation task. Futhermore, for question answer (QA) style tasks such as ARC and PIQA the ORPE funetuned models performed poorly unless the ORPE encodings were added to the 'answer' part of the evaluation. Likely the poor results were due to the loglikehood of correct or incorrect response being dominated by the additional model perplexity due

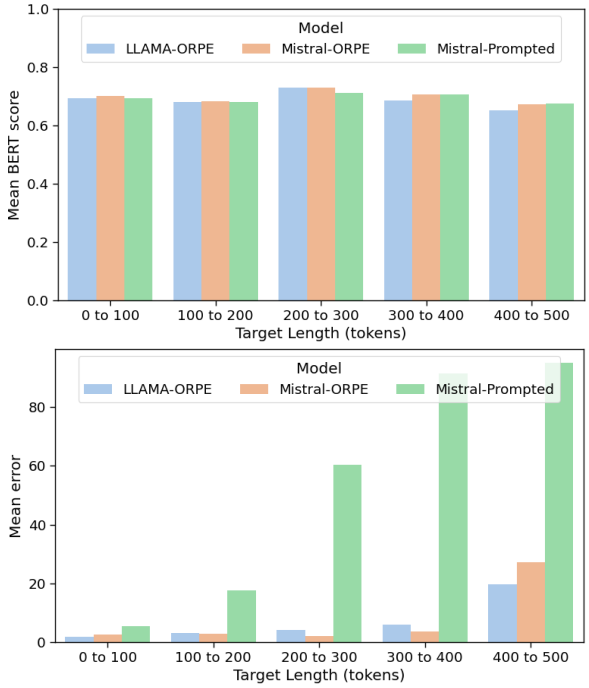

Figure 3: Results for length the length controlled summarisation task. Models used are ORPE finetuned Mistral and Llama (-ORPE), as well as the Mistral baseline model finetuned for prompted based length control (Mistral-Prompted). **top:** BERT scores between summaries and GPT-3.5 ground truth summaries. **bottom:** Average length errors for target summary length.

to the lack of ORPE embeddings. Therefore to evaluate the ORPE models on QA tasks we modified the LM Evaluation Harness to optionally apply the ORPE countdown to the answer part of the queries. Note that this gave the model additional information about the response length, but no information about whether a specific question response pair was correct or incorrect.

The results are shown in Figure 4 for the ORPE finetuned Mistral and Llama models (Mistral-ORPE and Llama-ORPE) as well as the base instruct versions (-base). The results show that the ORPE finetuned model's performance was generaly preserved after finetuning, except for a reduction in Mistral's performance on the Hellaswag benchmarks.

5.2 Max New Tokens++

Figure 5 shows the comparison between the token limit and response length for a Mistral model finetuned using the *max new tokens++* method with $\sigma_0 = 0.1$ and $\sigma_{\max} = 2048$. For shorter token limits the responses closely follow the identity line, while for longer token limits there is a deviation towards shorter responses. Even though there is a shift in median away from identity the 95% confidence interval still follows closely. This indicates that the model still possesses a notion of "token budget", and that it uses this token budget to smoothly terminate the longer responses as they approach the token limit.

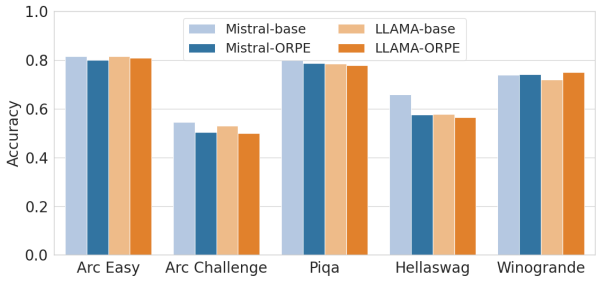

Figure 4: Performance of length controlled and baseline models against a range of standard benchmarks. All evaluations used zero-shot prompting. The labels Mistral and Llama correspond to Mistral-7B-Instruct and Llama-3-8B-Instruct respectively. The tag -ORPE means the model was finetuned with the ORPE. Additionally ORPE were added during the evaluation for QA type tasks (Arc and Piqa).

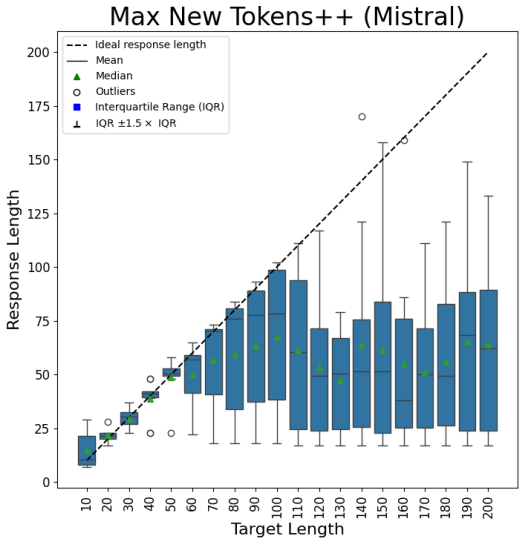

Figure 5: Plot of token limit vs response length for using a *max new tokens++* finetuned Mistral model on a question-answering task. This technique has not yet been applied to Llama but we expect the performance to be similar.

6 Conclusion

In this work, we introduce offset reverse positional encoding (ORPE), a method for controlling the output length of large language models (LLMs) without arbitrarily truncating the generation. This is achieved by adding an additional po-

sitional encoding to the transformer architecture, which counts down from a predetermined response length rather than counting up to the end of the sequence as is typically done. During training, the model learns to use this countdown to structure its responses to terminate at the end of the ORPE countdown. Our results demonstrate that ORPE effectively controls response length without degrading the underlying performance of the LLM.

Preliminary work was also presented for the *max new tokens++* method, a way of implementing an upper bound on the number of generated tokens rather than a target. The results show that this method is effective at training the model provide responses that terminate smoothly at or before the token limit is reached. This prevents the generation of needlessly long answers in an attempt to meet the target length imposed by the ORPE encoding.

Overall, this work shows that the addition of a specialized positional encoding during finetuning enables the model to output specific tokens on command without degrading response quality. While we focused on controlling the generation of the EOS token to terminate the response on command, this approach could be generalised to control the generation of other tokens or influence other aspects of LLM generation.

7 Limitations

We have identified several limitations in our work that could be addressed to improve the evaluation and effectiveness of the proposed length control method.

Firstly, due to time constraints, we fine-tuned our models on relatively small question-answer-focused datasets. The impact of dataset size and diversity on learning length control remains unclear. Secondly, our experiments were limited to the instruct versions of Mistral 7B and Llama3 8B models, both within the 7-8B parameter range and instruction fine-tuned. While this made them suitable for further fine-tuning with a QA dataset, it remains an open question how well the proposed length control techniques generalise to other models.

Another limitation is that the countdown mechanism in our approach is based on tokens rather than words or characters, the latter of which may be more relevant for user applications. Future research could explore using ORPE to count down words or characters instead of tokens. Additionally, incorporating the relative progress towards the termination length, rather than the absolute number of remaining tokens, might enhance the model's ability to generalise across various response lengths. Investigating dynamic control of the ORPE weighting within a beam search algorithm could further refine response length accuracy.

Finally, we utilised standard sinusoidal positional encodings in a reverse and offset order throughout this work. These encodings, originally designed to provide information about token ordering, might not be optimal for our specific use case where the focus is on counting down to a termination point.

Addressing these limitations in future work would provide a more robust evaluation of the proposed length control method and potentially enhance its applicability and performance across different models and datasets.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. pages 41–48.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. PIQA: reasoning about physical commonsense in natural language. *CoRR* abs/1911.11641. http://arxiv.org/abs/1911.11641.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* .

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47(1):1–66.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation. https://doi.org/10.5281/zenodo.10256836.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. 2024. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719* .

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR* abs/2106.09685. https://arxiv.org/abs/2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Llio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothe Lacroix, and William El Sayed. 2023. Mistral 7b.

Renlong Jie, Xiaojun Meng, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Prompt-based length controlled generation with reinforcement learning. *ArXiv* abs/2308.12030. https://api.semanticscholar.org/CorpusID:261076002.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1328–1338.

Chao Lei, Nir Lipovetzky, and Krista A. Ehinger. 2024. Generalized planning for the abstraction and reasoning corpus.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, et al. 2024. Transformers can do arithmetic with the right embeddings. *arXiv preprint arXiv:2405.17399* .

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date — ai.meta.com. `https://ai.meta.com/blog/meta-llama-3/`. [Accessed 13-05-2024].

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond.

OpenAI. 2024. GPT3.5 Turbo-0125. https://platform.openai.com/docs/models/gpt-3-5-turbo.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, et al. 2021. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637* .

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WINOGRANDE: an adversarial winograd schema challenge at scale. *CoRR* abs/1907.10641. http://arxiv.org/abs/1907.10641.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* .

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. 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 3999–4004.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* .

Zhongyi Yu, Zhenghao Wu, Hao Zheng, Zhe XuanYuan, Jefferson Fong, and Weifeng Su. 2021. Lenatten: An effective length controlling unit for text summarization. *CoRR* abs/2106.00316. https://arxiv.org/abs/2106.00316.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *CoRR* abs/1905.07830. http://arxiv.org/abs/1905.07830.

Appendix

Question
Extract the answer to the question from the following context. Question: What's the expensive rock mean Context: Buddhist schools vary on the exact nature of the path to liberation, the importance and canonicity of various teachings and scriptures, and especially their respective practices. Buddhism denies a creator deity and posits that mundane deities such as Mahabrahma are misperceived to be a creator. The foundations of Buddhist tradition and practice are the Three Jewels: the Buddha, the Dharma (the teachings), and the Sangha (the community). Taking ""refuge in the triple gem"" has traditionally been a declaration and commitment to being on the Buddhist path, and in general distinguishes a Buddhist from a non-Buddhist. Other practices are Ten Meritorious Deeds including, giving charity to reduce the greediness; following ethical precepts; renouncing conventional living and becoming a monastic; the development of mindfulness and practice of meditation; cultivation of higher wisdom and discernment; study of scriptures; devotional practices; ceremonies; and in the Mahayana tradition, invocation of buddhas and bodhisattvas.

Responses	**Target length**	**Response length**
There is no answer.	10	12
The context does not provide an answer to the question ""What's the expensive rock mean?"" The context is about Buddhist schools and their practices, and it does not mention anything about an expensive rock. Therefore, it is not possible to extract an answer to the question from the given context. The question seems to be unrelated to the context and requires a separate answer or information that is not provided in the given context.	90	89
The context does not provide an answer to the question ""What's the expensive rock mean?"" The context is about Buddhist schools and their practices, including the Three Jewels, Ten Meritorious Deeds, and other practices. It also mentions the denial of a creator deity and the importance of scriptures and teachings in different Buddhist schools. However, there is no mention of an ""expensive rock"" in the context. Therefore, it is not possible to extract an answer to the question from the given context. The question seems to be unrelated to the context and requires a separate answer or information that is not provided in the given context.	150	149
The context provided does not directly answer the question about the expensive rock. However, it does provide information about the nature of the path to liberation, the importance and canonicity of various teachings and scriptures, and especially their respective practices in Buddhist schools. The context also mentions the Three Jewels, which are the Buddha, the Dharma (the teachings), and the Sangha (the community). Taking ""refuge in the triple gem"" has traditionally been a declaration and commitment to being on the Buddhist path, and in general distinguishes a Buddhist from a non-Buddhist. Therefore, the answer to the question about the expensive rock is not provided in the given context. It is important to note that the context provided is about Buddhist schools and their practices, and it does not mention anything about an expensive rock. If you have any further questions or if you would like to know more about Buddhist schools and their practices, feel free to ask.	200	197

Table 1: Example generated responses using ORPE fine tuned Llama3 8B. Note that special tokens, instruction tokens and linebreaks have been removed from the text but not from the token count.