DECODING PARTIAL DIFFERENTIAL EQUATIONS: CROSS-MODAL ADAPTATION OF DECODER-ONLY MODELS TO PDES

Paloma García-de-Herreros[1] **Philipp Slusallek**[1, 2] **Dietrich Klakow**[1] **Vagrant Gautam**[3]
[1]Saarland University [2]DFKI [3]Heidelberg Institute for Theoretical Studies
pgherreros@lsv.uni-saarland.de vagrant.gautam@h-its.org

ABSTRACT

While large language models are primarily used on natural language tasks, they have also shown great promise when adapted to new modalities, e.g., for scientific machine learning tasks. Most proposed approaches for such cross-modal adaptation of language models focus on encoder-only transformer model architectures, despite decoder-only architectures being far more popular for language tasks in recent years, and being trained at much larger scales. This raises the question of how model architecture affects cross-modal adaptation approaches, and whether we can leverage the success of decoder-only models. In this paper, we systematically compare encoder-only and decoder-only language models on cross-modal adaptation for time-dependent simulation tasks based on partial differential equations (PDEs). We find that decoder-only models are far worse than encoder-only models, when existing approaches are applied unmodified. In contrast to several other domains, scaling decoder-only models also does not help. To enhance the performance of decoder-only models in this context, we introduce two novel approaches that mimic bidirectionality, Parallel Flipping and Sequence Doubling. Both our methods improve overall performance using decoder-only models for all tasks and all cross-modal adaptation methods, closing the gap to encoder-only model performance. We hope that our findings broaden the spectrum of models used on cross-modal adaptation tasks to further scientific machine learning.

1 INTRODUCTION

Pre-trained large language models (LLMs) have seen unprecedented improvements in processing natural language in recent years. These models can then be adapted to new tasks using different approaches like fine-tuning or in-context learning. Recent work has even applied such methods to adapt models to data modalities unseen during pre-training, an approach termed cross-modal adaptation. Cross-modal adaptation methods purport to take advantage of the knowledge, skills, or simply the already-optimized neural network of a pre-trained model, to adapt it to a new modality. Cross-modal adaptation achieves competitive performance across a wide range of tasks, including detecting atrial cardiac disease from ECG recordings, and time-dependent simulation tasks based on partial differential equations (Lu et al., 2022; Shen et al., 2023; Ma et al., 2024; Shen et al., 2024). These approaches can be of great utility for scientific machine learning tasks, currently used for tasks such as seismic monitoring (Wang et al., 2025) and time series forecasting (Liu et al., 2025).

However, it is unclear how general cross-modal adaptation methods are, since few studies vary the originally proposed configurations. For example, most approaches are based on encoder-only models, even though decoder-only models are by far the more popular transformer-based model architecture (Vaswani et al., 2017) for natural language tasks; they scale impressively, mimic human-like language convincingly, and can be used for a wide variety of tasks. Through better, more general representations of natural language, today's best decoder-only models may well provide a better starting point for cross-modal adaptation.

In our work, we attempt to leverage the potential of decoder-only models to broaden the range of models available for cross-modal adaptation. First, we directly apply two existing cross-modal

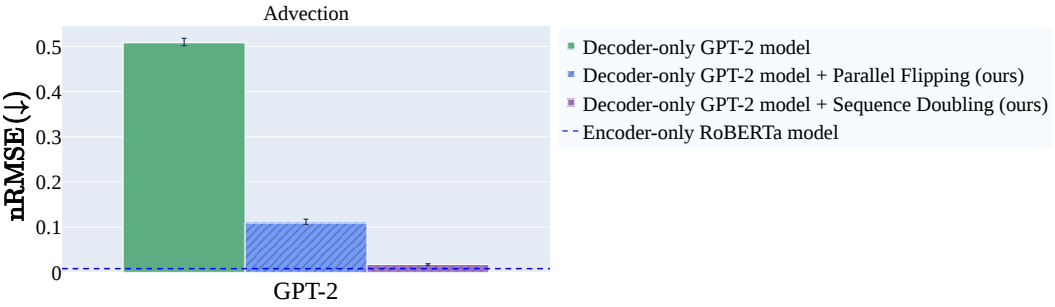

Figure 1: Cross-modal adaptation of GPT-2, a decoder-only model, with ORCA-based adaptation on the Advection dataset of time-dependent PDE simulation compared to the original performance of RoBERTa with the same cross-modal adaptation approach as a baseline. Although the original setup shows high error, our proposed methods (Parallel Flipping and Sequence Doubling) close the gap to encoder-only model performance.

adaptation methods to encoder-only and decoder-only models, and find that decoder-only models perform much worse (Section 4). Next, we test whether scaling up decoder-only models helps performance, but find that it does not (Section 5).

We hypothesize that the reasons for the lack of success of these approaches are due to autoregressive attention over the input, as well as how the outputs are computed, which is by averaging the representations of the last hidden layer, rather than generating outputs, as in natural language. Addressing these issues, we introduce two different methods to improve cross-modal adaptation with decoder-only models by simulating bidirectionality (Section 6):

Parallel Flipping: In parallel, we run the original setup and the same pipeline with the data inverted, and we then combine the predictions by taking the second half of both of them.

Sequence Doubling: We concatenate every sequence in the original data with itself before introducing it to the model, and use only the second half of the last hidden layer to compute predictions.

Both methods extend the sequence context that models can access at different points, and results show that our methods outperform the original setup for all tasks and cross-modal adaptation methods, closing the gap to encoder-only models (Section 7), as shown in Figure 1. Each method comes with particular tradeoffs, making this a promising direction for future work, which we discuss in Section 8. We hope that our findings broaden the spectrum of models used for cross-modal adaptation and further the field of scientific machine learning.

2 Related Work

2.1 Machine Learning and Partial Differential Equations

Solving partial differential equations has been considered a classical machine learning problem, given its scientific significance, with task-specific models proposed including the physics-informed neural network (PINN; Raissi et al., 2019), Fourier neural operator (FNO; Li et al., 2021). In recent years, foundational PDE models like PDE-FM (Soares et al., 2026), POSEIDON (Herde et al., 2024), Walrus (McCabe et al., 2025), and UNISOLVER (Zhou et al., 2025) have been introduced. All of these approaches focus on custom-built machine learning solutions to solve PDEs.

In contrast, other approaches try to leverage other types of machine learning models (such as large language models) to enhance analytical PDE approximations (Bhatnagar et al., 2025), find the analytical solution to differential equations (Surkov et al., 2024; Zakharov et al., 2025; Xiao et al., 2026). Some work has even used LLMs to generate code (Li et al., 2025) to simulate the data modeled by certain PDEs, as proposed in Takamoto et al. (2022), while other work uses in-context learning to simply prompt LLMs for PDE simulation tasks (Bao et al., 2026) Our work differs from

the above by using cross-modal adaptation of LLMs to solve these tasks, similar to Shen et al. (2023) and Shen et al. (2024). This data consists of time series predictions of continuous observations over a space domain, similar to other scientific machine learning data such as Satellite (Petitjean et al., 2012) and MegaFlow2D (Xu et al., 2023), which our PDE-focused work could also be relevant to.

2.2 Cross-Modal Adaptation

Although language model adaptation is an important feature within natural language applications (across domains, languages, varieties, for particular use cases, etc.), a more extreme approach known as cross-modal adaptation has recently been introduced, which involves adapting models to new modalities unseen by the model during pre-training. The concept of modality is not well-defined in this literature, but the key idea is to adapt models to a data modality that is entirely different from the modality that a model was pre-trained on, and which should, in theory, have no connection, unlike speech and text representing the same utterance. Most of the proposed methods focus on the fine-tuning stage of large language models. These methods include Frozen Pretrained Transformers (FPT; Lu et al., 2022), ORCA (Shen et al., 2023), Patch Replacement (PaRe; Cai et al., 2024), Modality kNowledge Alignment (MoNA; Ma et al., 2024), Unified PDE Solver (UPS; Shen et al., 2024), and more. All these methods take advantage of model pre-training, to minimize the amount of fine-tuning necessary to adapt it to a new modality. These techniques have a lot of potential to be used for various scientific machine learning tasks, and some practical applications have also been presented, including seismic monitoring (Wang et al., 2025), time series forecasting (Liu et al., 2025), and physics simulations (Ohana et al., 2024).

2.3 Architecture Differences

Virtually all modern large language models are based on the transformer architecture, consisting of an encoder and a decoder (Vaswani et al., 2017). This encoder-decoder architecture has been replaced by encoder-only architectures such as BERT (Devlin et al., 2018), as well as decoder-only architectures such as GPT (Radford et al., 2019), the latter two of which are more popular in modern natural language processing applications. Due to differences between the architectures, including pre-training objectives and attention mechanisms, several works compare them, finding differences in phenomena such as pronoun use (Gautam et al., 2024) and various linguistic probes (Waldis et al., 2024). In cross-modal adaptation, however, there have been no systematic architectural comparisons, to the best of our knowledge. Some papers, like us, also try to close the gap between encoder-only and decoder-only models in various contexts, including language model embeddings (BehnamGhader et al., 2024; Springer et al., 2025) and cognitively plausible language modelling (Charpentier & Samuel, 2024).

3 Experimental Setup

To evaluate the effects of model architecture and scaling on cross-modal adaptation with partial differential equation data, we experiment with several models, scales, and cross-modal adaptation methods as described below.

3.1 Methods

We choose two popular methods for cross-modal adaptation in the literature – Frozen Pretrained Transformers (FPT) (Lu et al., 2022) and ORCA (Shen et al., 2023). In both cases, a task-specific embedder and predictor are created to account for mismatches in dimensions between the target modality data and the original model. Then, FPT adapts the pre-trained models to new tasks by fine-tuning only the input and output layers, as well as the layer normalization parameters. ORCA, on the other hand, first trains the embedder on its own, minimizing the Optimal Transport Dataset Distance (OTDD) (Alvarez-Melis & Fusi, 2020) between the target task dataset and a pre-selected proxy dataset. After this, all parameters are trained on the target task dataset. We use ORCA's implementation for both ORCA and FPT, with the same hyperparameters.

3.2 MODELS

We select ROBERTA-BASE (Liu et al., 2019) and BERT as our encoder-only models, following ORCA (Shen et al., 2023), and GPT-2 (Radford et al., 2019) and PYTHIA (Biderman et al., 2023) as our decoder-only models, since GPT-2 is used in Lu et al. (2022) and PYTHIA-160M has a large range of model sizes. All of these models have similar sizes (respectively, 125M, 110M, 137M, and 160M parameters). For the scaling experiments, we consider the larger versions of the GPT-2 family: GPT-2 MEDIUM (380M), GPT-2 LARGE (812M), and GPT-2 XL (1.61B), as well as the PYTHIA family: PYTHIA-14M, PYTHIA-70M, PYTHIA-410M, PYTHIA-1B, and PYTHIA-1.4B. We do not consider larger PYTHIA model sizes to keep the comparison with the GPT-2 family fair.

3.3 DATASETS

We use four different datasets of time-dependent simulation tasks based on partial differential equations: Advection, Diffusion-Reaction, Diffusion-Sorption, and Navier-Stokes, all taken from PDEBench (Takamoto et al., 2022). We follow the configurations in Shen et al. (2023) as detailed in Appendix A. As in their work, we train the models to predict the last time step from the first time step, instead of a shorter one-step prediction or a multi-step rollout.

Proxy Datasets In addition to the target dataset, the ORCA method also requires a proxy dataset for training the embedder. For ROBERTA-BASE, we use the original proxy dataset generated by Shen et al. (2023) using CoNLL-2003. We follow their approach with CoNLL-2000 to generate proxy datasets for the rest of the models, as we explain in detail in Appendix B.

3.4 EVALUATION METRIC

As in previous literature (Shen et al., 2023; Ma et al., 2024; Shen et al., 2024; Cai et al., 2024; Li et al., 2025), we report normalized Root Mean Squared Errors (nRMSE) for all tasks, as it is scale-independent. As the metric is error-based, lower values are better, which we also note in all figure captions. We report averages over five runs; given the high variance for some configurations, we show best (minimum) and worst (maximum) performance with error bars.

4 DECODER-ONLY MODELS PERFORM MUCH WORSE THAN ENCODER-ONLY MODELS

In this section, we experiment with two transformer architectures, encoder-only and decoder-only models, represented by ROBERTA-BASE and BERT-BASE, and GPT-2 and PYTHIA-160M, respectively, plugged directly into the existing cross-modal adaptation approaches, FPT and ORCA. Prior work generally assumes that pre-training results in better cross-modal adaptation performance, but we ablate for this factor as well by including randomly-initialized versions of these models. This allows us to disentangle the effects of both architecture and pre-training.

We start by considering the performance of randomly-initialized versions of the models, to evaluate whether pre-training on language data actually helps at all with these tasks. Using ORCA, as Figure 2 shows, decoder-only models do not outperform their randomly-initialized counterparts for any of the tasks, while encoder-only models do so for all of the tasks, at least with one of the tested models. On the other hand, when using FPT, both encoder-only and decoder-only models do outperform their randomly-initialized versions for most of the tasks (except PYTHIA for Navier Strokes). However, both sets of models still show very large error compared to ORCA-based adaptation, and in some cases (for example, PYTHIA for Diffusion-Reaction), the performance gain is small. We contend that applying these approaches should only be done when the pre-training in the original modality is necessary; otherwise, there is no gain from pre-training a model at all.

When comparing the performance pre-trained models of different architectures, as Figure 2 shows, **encoder-only models outperform decoder-only models overall** for three of the four selected tasks (Advection, Diffusion-Reaction, and Navier-Stokes), with very different performance depending on the task. The remaining task, Diffusion-Sorption, shows equally good performance for all models and cross-modal adaptation methods, indicating that the task is simple enough to be solved without

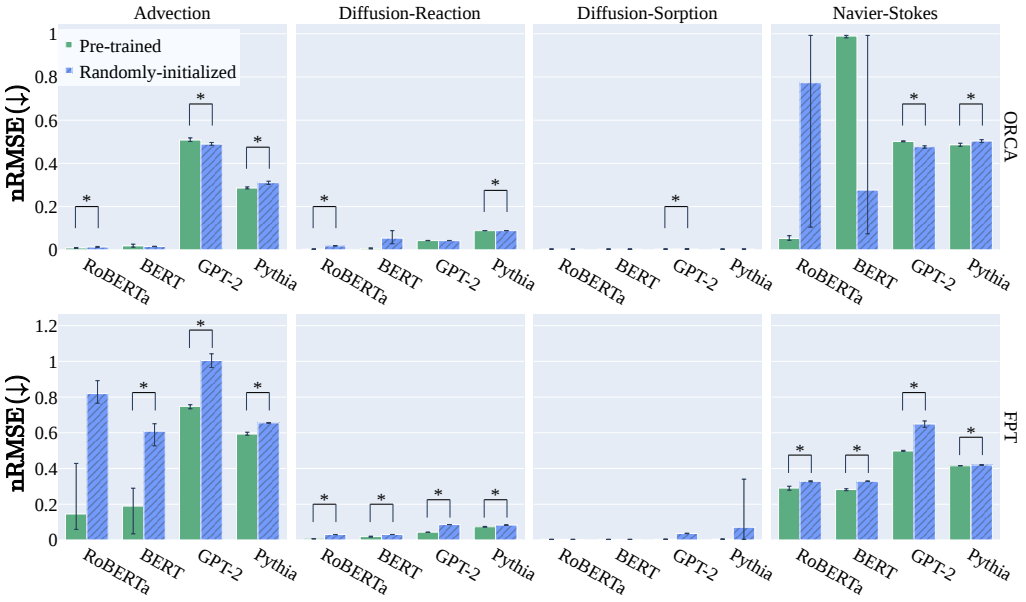

Figure 2: Comparison of model performance with ORCA- (above) and FPT-based (below) cross-modal adaptation, using both pre-trained and randomly-initialized versions of encoder-only models (RoBERTa, BERT) and decoder-only models (GPT-2, Pythia). Performance is measured using nRSME, where lower is better; the plots show average performance over five random seeds, and the error bars represent the best and worst runs. Statistically significant differences between pre-trained and randomly-initialized models are denoted with $*$); a detailed explanation can be found in Appendix G. Since the scale used can make Diffusion-Reaction and Diffusion-Sorption plots hard to read, complementary plots can be found in Appendix C.

pre-training. Similarly to what García de Herreros et al. (2024) report with the Satellite dataset for satellite image time series analysis, this highlights the importance of selecting tasks that allow us to better evaluate cross-modal adaptation methods. Broadly, we also observe that ORCA achieves better results than FPT on three of the four tasks, as previously described in Shen et al. (2023).

Lastly, it must be noted that for some of these tasks, there is large variance between runs. For ORCA, all tasks are stable except for Navier-Stokes, where we can see high variance when using encoder-only models that have been randomly initialized. When using their pre-trained counterparts, this variance reduces dramatically. On the other hand, when using FPT, all tasks except for Advection seem stable. For Advection, pre-trained encoder-only models show high variance between runs. As we discuss later, this fine-tuning instability—which could come from optimizers or simply bad regions in weight space—should be investigated more systematically in future work.

Overall, our results show that **decoder-only models cannot compete with encoder-only models for PDE tasks using cross-modal adaptation methods out of the box**. In the following sections, we try several approaches to close the performance gap between architectures.

5 Scaling Decoder-Only Models does not Improve Performance

The previous results motivated us to find potential ways in which decoder-only models can achieve comparable performance to encoder-only models. Since the compared models in the last section were all of similar size, in this section, we test scaling decoder-only models to see if this improves performance, as seen in other areas (Kaplan et al., 2020; Caillaut et al., 2024; Cai et al., 2025).

However, Figure 3 shows that **scaling barely reduces the performance gap between decoder-only and encoder-only models**, where RoBERTa-BASE represents encoder-only model performance. Below, we outline the trends we observe.

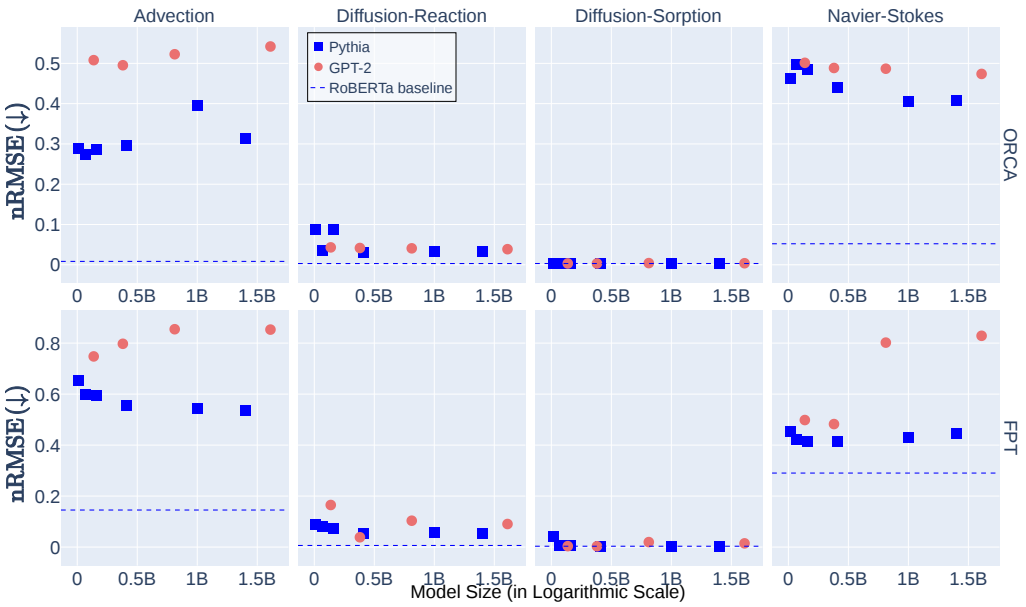

Figure 3: Performance of different sizes of models of the GPT-2 family and PYTHIA family using both ORCA (Shen et al., 2023) and FPT (Lu et al., 2022). The plots depict the average performance over five random seeds. Once again, performance is measured using nRSME, where lower is better. If scaling the models was improving the performance, downward trends could've been seen for the different model families. Since the scale used can make Diffusion-Reaction and Diffusion-Sorption plots hard to read, complementary plots can be found in Appendix C.

When using ORCA, there is no performance improvement on the Advection and Diffusion Sorption datasets; for Advection, there is even some deterioration for both model families. Diffusion-Reaction shows some improvement with the PYTHIA models, with some outliers, but no improvement with the GPT-2 models. On the other hand, Navier-Stokes shows some improvement for both model families; the relative percentage improvement when comparing the best model with the smallest model of each family is 5% for the GPT-2 models versus 12% for the PYTHIA models. Still, the trend is not smooth, and the parameter increase to achieve this performance is much bigger than the performance gain; the best GPT-2 model is approximately 12 times bigger than the original size, and the best PYTHIA model approximately 71 times bigger, without getting much closer to encoder-only model performance.

With FPT-based adaptation, GPT-2 models do not show consistent performance improvements, even deteriorating for Advection and Navier-Stokes. On the other hand, the PYTHIA family shows some improvements for Advection and Navier-Stokes, but once again, the gains are relatively small compared to the models' size difference.

Since scaling does not close the performance gap between architectures, we hypothesize that the stark differences in performance from plugging decoder-only models into these approaches are due to two reasons: First, they are penalized for being autoregressive, since each point in the sequence is treated as an individual token, and GPT-2 and PYTHIA cannot condition on the sequence bidirectionally, which is necessary for waveforms with symmetry. Secondly, the predictions are not computed generatively, but instead, the representations of the last hidden layer are simply averaged. This does not take advantage of the strong generative capabilities that decoder-only models possess.

In the following sections, we focus on addressing our first hypothesis as a means to improve the performance of decoder-only models using cross-modal adaptation approaches, and leave the exploration of generation for cross-modal adaptation to future work.

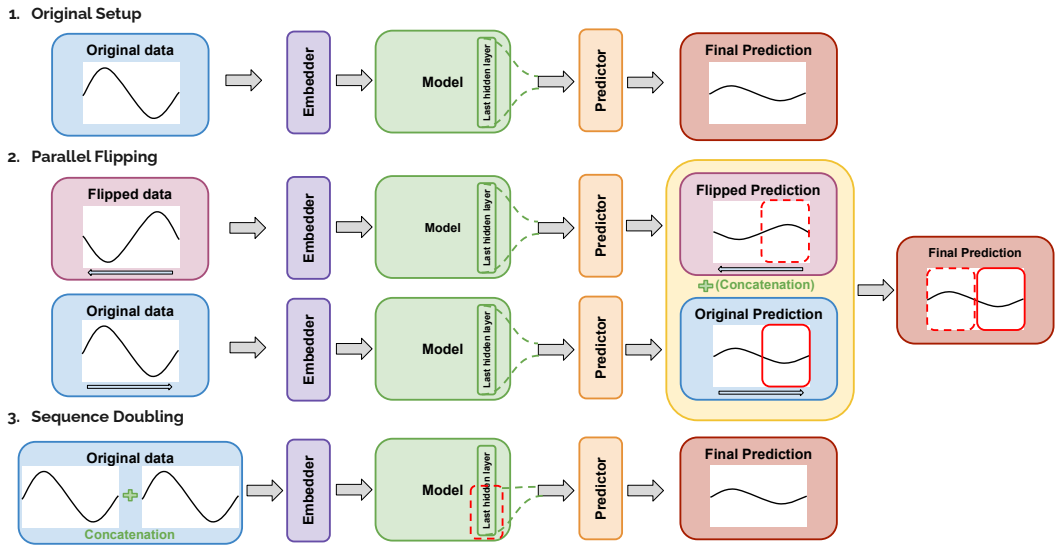

Figure 4: Pipeline comparison of the original setup and the two methods we introduce, Parallel Flipping and Sequence Doubling. For **Parallel Flipping**, the pipeline is run twice, with the original data and with the inverted sequences. For **Sequence Doubling**, each sequence is concatenated with itself before being introduced to the model, and then we only pass the second part of the last hidden layer to the predictor.

6 Simulating Bidirectionality With Decoder-Only Models

Since scaling decoder-only models does not improve their performance, we introduce two novel methods to counter the lack of bidirectional context in the models, illustrated in Figure 4.

6.1 Parallel Flipping

Through error analysis of the decoder-only model outputs, we observed that the beginnings of the output sequences were generally more spiky but they became smoother as the sequence progressed, since the model has more context to condition on. We show some examples of this in Appendix D.

Using this to our advantage, we design a new method to give both halves of the sequence equal opportunity to condition on the other. As shown in Figure 4, we run the same cross-modal pipeline twice in parallel (both for ORCA and FPT), once with the original data and once with the data sequences inverted. Then, we combine both predictions by taking the second half of each from the original run and the inverted one and concatenating them. In this way, both parts of the predicted sequence have access to the previous context and we obtain the smoother part of both runs, even though the point at which they are concatenated can still be spiky. Compared to the original cross-modal adaptation approach, the second half of the final prediction remains unchanged with Parallel Flipping, but the first half may now improve through conditioning on the flipped version.

6.2 Sequence Doubling

To expand the context window the model can use beyond half the sequence (as in Parallel Flipping) to the full sequence, we introduce sequence doubling. As shown in Figure 4, we concatenate all the sequences introduced to the model with themselves. Then, for the prediction, we take only the second half of the last hidden layer and introduce it to the predictor. This half of the hidden layer is conditioned on the first instance of the entire sequence and should therefore be a much richer representation of the data. Indeed, a similar kind of sequence repetition has also shown promising results in the context of language model embeddings (Springer et al., 2025). Compared to Parallel Flipping, this approach also does not have a hard concatenation point, which should result in

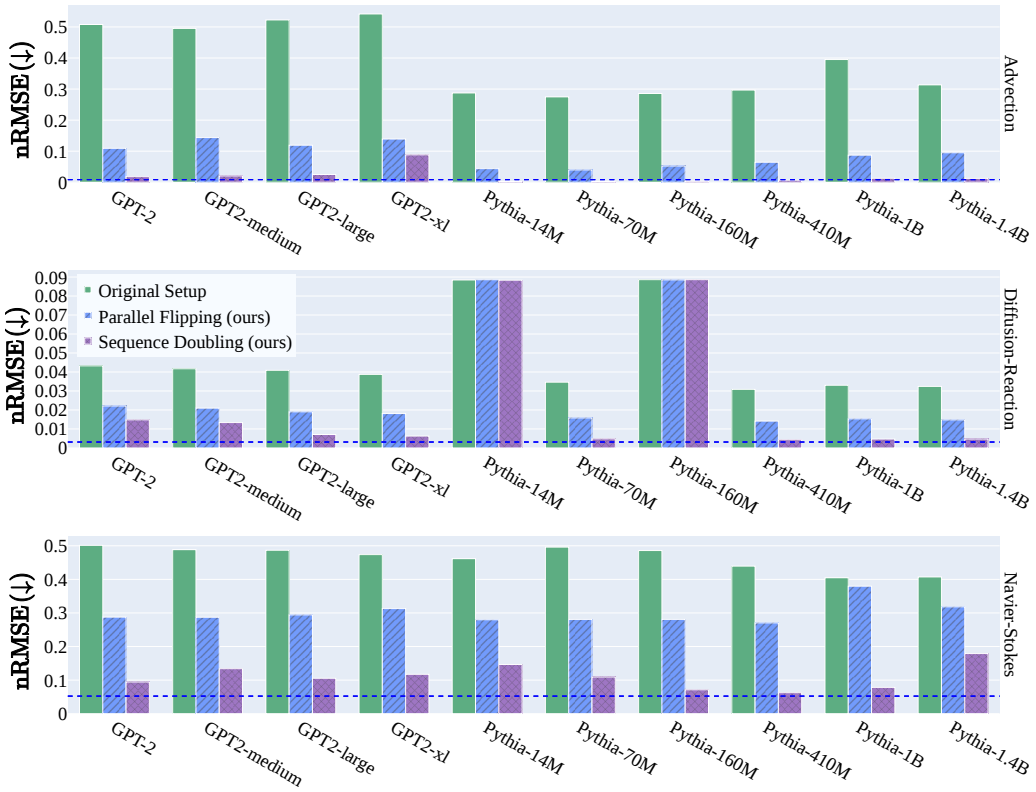

Figure 5: Performance comparison of the original setup versus our own two methods, Parallel Flipping and Sequence Doubling, using ORCA (Shen et al., 2023). We set RoBERTa with the original setup as a baseline for all the configurations. The plots depict the average performance over five random seeds. Performance is measured using nRSME, where lower is better. The performance comparison using and FPT (Lu et al., 2022) can be found in Appendix E.

smoother and better outputs overall, leading to bigger improvements. However, Sequence Doubling also doubles the sequence length of inputs, so we reduce the batch size for some configurations.

7 Simulating Bidirectionality Closes The Performance Gap Between Architectures

We compare our two newly-proposed methods with the original setup in Figure 5, using the results of our two new methods in the following section, compared to the original setup. As Diffusion-Sorption shows equally good performance for all configurations in previous sections, we focus on the other three datasets for these remaining experiments.

Overall, both **Parallel Flipping and Sequence Doubling outperform the original setup** for all tasks and cross-modal adaptation methods. As expected, we also see that Sequence Doubling generally outperforms Parallel Flipping, with the exception of FPT-based adaptation of GPT-2 models to Advection and Navier-Stokes.

For ORCA, we see larger improvements, with Sequence Doubling outperforming Parallel Flipping for all tasks. For both Navier-Stokes and Diffusion-Reaction, the performance approaches RoBERTa-BASE's original performance, and in some cases, our approaches even outperform RoBERTa-BASE, i.e., some Pythia models on Advection (Pythia-14M, Pythia-70M, Pythia-160M, and Pythia-410M).

For FPT (see Appendix E), improvements are less consistent than for ORCA, but still show good gains across most configurations. All models approach ROBERTA-BASE's performance on Diffusion-Reaction with Sequence Doubling, but the improvement with Parallel Flipping is smaller as before. For some configurations, decoder-only models outperform encoder-only models again (e.g., PYTHIA on Advection and Navier-Stokes with Sequence Doubling). In contrast, GPT-2 models show small gains.

Despite the improvements, we still do not consistently see neat scaling behaviour on cross-modal adaptation. This could be due to a task inherently not benefiting from scaling, particularly for datasets that can already be solved with a lower-capacity model. On the other hand, the lack of clear scaling could also come from randomness in the particular model checkpoints that we use, which could also cause some of the outlier runs that we see. Adaptation stability is therefore an important area for future work in cross-modal adaptation.

8 DISCUSSION AND FUTURE WORK

With a series of experiments to analyze the effect of model architecture and size on cross-modal adaptation approaches, we show that decoder-only models are consistently worse than encoder-only models and do not, at least with traditional approaches, exploit the potential of their pre-trained knowledge for the new tasks. We show that this is due to decoder-only models being penalized for their autoregressive attention over the input. To address this penalization, we introduce two different methods, both of which come with certain tradeoffs.

First, **Parallel Flipping** requires each instance to be run twice to obtain the final prediction, but by design it can be parallelized, either using double the resources to run in the same time or running it sequentially in double the time.

On the other hand, **Sequence Doubling** cannot be parallelized. Also, since the sequence length is doubled, it takes longer to run and increases the required memory. In some cases, particularly when using bigger models, this requires reducing the batch size or upgrading our resources.

Our primary motivation with both methods was to try to mimic the data processing of encoder-only models while using decoder-only models. We did so by introducing a kind of bidirectional context. Another way of achieving this would be to actually enable bidirectional attention in decoder-only models, as in LLM2Vec (BehnamGhader et al., 2024), or by merging encoder-only and decoder-only models as in (Charpentier & Samuel, 2024). We leave this to future work.

We also notice that cross-modal adaptation methods are very sensitive to the task, with performance varying greatly between the four PDEs we consider. To better assess the utility of methods in this setting, future work should investigate which properties of PDEs are responsible for this variance.

However, we see the most important direction for future work as being to diagnose the instabilities of cross-modal adaptation, given the high variance of performance with some configurations. As we point out in Section 4, optimizers might play a role (Kunstner et al., 2023), as might randomness in the checkpoints we begin with. One approach would be to try to disentangle when transfer capabilities emerge for these models (and whether that is stable), particularly decoder-only models, and the influence that they have on the variation (van der Wal et al., 2025).

9 CONCLUSION

In this paper, we aim to understand and minimize the impact of model architecture and size on cross-modal adaptation approaches with time-dependent simulation of partial differential equations. We find that decoder-only models perform much worse than encoder-only models, even when scaled up. Unidirectional attention plays a key role in this performance gap, preventing models from conditioning on the data overall. To mitigate the effects of the lack of bidirectionality, we introduce two novel approaches: Parallel Flipping and Sequence Doubling, both of which outperform the original setup, with Sequence Doubling showing much larger gains and closing the gap to encoder-only model performance. We encourage future research on scientific machine learning to build on our approach to leverage more capable decoder-only models in cross-modal adaptation research.

10 LIMITATIONS

We only experiment with two popular cross-modal adaptation methods, and leave it to future work to investigate whether the same patterns hold for PARE (Cai et al., 2024) and UPS (Shen et al., 2024).

Our experiments focus on 1-dimensional PDE datasets and tasks widely used in the field of cross-modal adaptation (Lu et al., 2022; Shen et al., 2023; Ma et al., 2024; Shen et al., 2024; Cai et al., 2024), even though different PDE tasks (like higher-dimensional PDEs) and PDE-specific evaluations (e.g., with physics-based metrics) could also be used. We caution against claims about generalization of our results on other PDE tasks and evaluations.

Additionally, given our difficulties replicating the original proxy dataset from ORCA (Shen et al., 2023), more testing is required to determine the potential influence this could have on all models.

11 ETHICS STATEMENT

All datasets and models are used in accordance with their licenses and intended use.

12 REPRODUCIBILITY STATEMENT

Our code is available here: `https://github.com/palomagh/DecodingPDEs`

We use Nvidia A100 GPUs to run all experiments. We perform five runs per configuration, using different random seeds, for a total of 232 different configurations. The running times spanned from 12 to 140 GPU-hours, depending on the dataset and model size.

REFERENCES

David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21428–21439. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f52a7b2610fb4d3f74b4106fb80b233d-Paper.pdf.

Jiajun Bao, Nicolas Boullé, Toni J.B. Liu, Raphaël Sarfati, and Christopher Earls. Text-trained LLMs can zero-shot extrapolate PDE dynamics, revealing a three-stage in-context learning mechanism. In *AI&PDE: ICLR 2026 Workshop on AI and Partial Differential Equations*, 2026. URL https://openreview.net/forum?id=BN8hrHQnkC.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IW1PR7vEBf.

Rohan Bhatnagar, Ling Liang, Krish Patel, and Haizhao Yang. From equations to insights: Unraveling symbolic structures in pdes with llms. *arXiv preprint arXiv:2503.09986*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Hongru Cai, Yongqi Li, Ruifeng Yuan, Wenjie Wang, Zhen Zhang, Wenjie Li, and Tat-Seng Chua. Exploring training and inference scaling laws in generative retrieval. *arXiv preprint arXiv:2503.18941*, 2025.

Lincan Cai, Shuang Li, Wenxuan Ma, Jingxuan Kang, Binhui Xie, Zixun Sun, and Chengwei Zhu. Enhancing cross-modal fine-tuning with gradually intermediate modality generation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and

Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5236–5257. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/cai24c.html.

Gaëtan Caillaut, Mariam Nakhlé, Raheel Qader, Jingshu Liu, and Jean-Gabriel Barthélemy. Scaling laws of decoder-only models on the multilingual machine translation task. In Barry Haddow, Tom Kocmi, Philipp Koehn, and Christof Monz (eds.), *Proceedings of the Ninth Conference on Machine Translation*, pp. 1318–1331, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.wmt-1.124. URL https://aclanthology.org/2024.wmt-1.124/.

Lucas Georges Gabriel Charpentier and David Samuel. GPT or BERT: why not both? In Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox (eds.), *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pp. 262–283, Miami, FL, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.conll-babylm.24/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Paloma García de Herreros, Vagrant Gautam, Philipp Slusallek, Dietrich Klakow, and Marius Mosbach. What explains the success of cross-modal fine-tuning with orca? In *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*, pp. 8–16, 2024.

Vagrant Gautam, Eileen Bingert, Dawei Zhu, Anne Lauscher, and Dietrich Klakow. Robust pronoun fidelity with english llms: Are they reasoning, repeating, or just biased? *Transactions of the Association for Computational Linguistics*, 12:1755–1779, 12 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00719. URL https://doi.org/10.1162/tacl_a_00719.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. *arXiv preprint arXiv:2304.13960*, 2023.

Shanda Li, Tanya Marwah, Junhong Shen, Weiwei Sun, Andrej Risteski, Yiming Yang, and Ameet Talwalkar. Codepde: An inference framework for llm-driven pde solver generation. *arXiv preprint arXiv:2505.08783*, 2025.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *In International Conference on Learning Representations*, 2021.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7628–7636, 2022.

Wenxuan Ma, Shuang Li, Lincan Cai, and Jingxuan Kang. Learning modality knowledge alignment for cross-modality transfer. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33777–33793. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ma24d.html.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Regaldo-Saint Blancard, Francois Rozet, Cristiana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, Irina Espejo, Rio Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Geraud Krawezik, Francois Lanusse, Rudy Morel, Ruben Ohana, Liam Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. Walrus: A cross-domain foundation model for continuum dynamics, 2025. URL https://arxiv.org/abs/2511.15684.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

François Petitjean, Jordi Inglada, and Pierre Gancarski. Satellite image time series analysis under time warping. *IEEE Transactions on Geoscience and Remote Sensing*, 50(8):3081–3095, 2012. doi: 10.1109/TGRS.2011.2179050. URL https://ieeexplore.ieee.org/document/6144005.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Erik F Sang and Sabine Buchholz. Introduction to the conll-2000 shared task: Chunking. *arXiv preprint cs/0009008*, 2000.

Junhong Shen, Liam Li, Lucio M. Dery, Corey Staten, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. Cross-modal fine-tuning: Align then refine. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31030–31056. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shen23e.html.

Junhong Shen, Tanya Marwah, and Ameet Talwalkar. Ups: Towards foundation models for pde solving via cross-modal adaptation. *arXiv preprint arXiv:2403.07187*, 2024.

Eduardo Almeida Soares, Emilio Ashton Vital Brazil, Victor Yukio Shirasuna, Breno William Santos Rezende de Carvalho, and Cristiano Malossi. Towards a foundation model for partial differential equations across physics domains. In *AAAI Conference on Artificial Intelligence*, 2026.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ahlrf2HGJR.

Anton Surkov, Vladimir Zakharov, Sergei Koltcov, and Vera Ignatenko. Application of large language models to solving differential equations: Constructing baseline models with lstm and gru. In *International Conference on Smart Technologies, Systems and Applications*, pp. 239–252. Springer, 2024.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

Oskar van der Wal, Pietro Lesci, Max Muller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. Polypythias: Stability and outliers across fifty language model pre-training runs. *arXiv preprint arXiv:2503.09543*, 2025.

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

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. Holmes: A benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647, 12 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00718. URL https://doi.org/10.1162/tacl_a_00718.

Xinghao Wang, Feng Liu, Rui Su, Zhihui Wang, Lei Bai, and Wanli Ouyang. Seismollm: Advancing seismic monitoring via cross-modal transfer with pre-trained large language model. *arXiv preprint arXiv:2502.19960*, 2025.

Qisong Xiao, Xinhai Chen, Qinglin Wang, Xiaowei Guo, Binglin Wang, Weifeng Chen, Zhichao Wang, Yunfei Liu, Rui Xia, Hang Zou, et al. Llm4fluid: Large language models as generalizable neural solvers for fluid dynamics. *arXiv preprint arXiv:2601.21681*, 2026.

Wenzhuo Xu, Noelia Grande Gutierrez, and Christopher McComb. Megaflow2d: A parametric dataset for machine learning super-resolution in computational fluid dynamics simulations. In *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, CPS-IoT Week '23, pp. 100–104, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700491. doi: 10.1145/3576914.3587552. URL https://doi.org/10.1145/3576914.3587552.

Vladimir Zakharov, Anton Surkov, and Sergei Koltcov. Agdes: a python package and an approach to generating synthetic data for differential equation solving with llms. *Procedia Computer Science*, 258:1169–1178, 2025.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: Pde-conditional transformers are universal neural pde solvers. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025.

A PDE Datasets Details and Configurations

As we saw in Section 3, we tested the models in a collection of PDE datasets from PDEBench (Takamoto et al., 2022). We follow Shen et al. (2023) for the download, pre-processing, and loading of the data.

The specifications of the selected datasets are shown in Table 1.

Table 1: List of PDE datasets used as target datasets and their corresponding specifications.

Dataset	Dimension	Resolution	Coefficients	Optimizer
Advection	1D	1024	$\beta = 0.4$	Adam
Diffusion-Reaction	1D	1024	$\nu = 0.5, \rho = 1.0$	SGD
Diffusion-Sorption	1D	1024	-	AdamW
Compressible Navier-Stokes	1D	1024	$\eta = \zeta = 0.1$, rand periodic	AdamW

A.1 1D Advection Equation

The Advection equations are defined as follows:

$$\partial_t u(t,x) + \beta\, \partial_x u(t,x) = 0 \tag{1}$$

$$u(0,x) = u_0(x) \tag{2}$$

where β represents a given advection speed and $x \in (0,1)$ and $t \in (0,2]$.

A.2 1D Diffusion-Reaction Equation

The Diffusion-Reaction equation is expressed as:

$$\partial_t u(t,x) - \nu \partial_{xx} u(t,x) = \rho u(t,x)(1 - u(t,x)) \tag{3}$$

where $x \in (0,1)$ and $t \in (0,1]$.

A.3 1D Diffusion-Sorption Equation

The Diffusion-Sorption equation is defined as follows:

$$\partial_t u(t,x) = \frac{D}{R(u)} \partial_{xx} u(t,x) \tag{4}$$

where $x \in (0,1)$, $t \in (0,500]$, and $D = 5 \times 10^{-4}$. We follow Takamoto et al. (2022) for the details of the function $R(u)$, describing the external force slowing down the diffusion process.

A.4 1D Compressible Navier-Stokes Equation

The Compressible Navier-Strokes equations are expressed as:

$$\partial_t p + \nabla \cdot (\rho u) = 0, \tag{5}$$

$$\rho(\partial_t u + u \cdot \nabla u) = -\nabla p + \eta \Delta u + \left(\xi + \frac{\eta}{3}\right) \nabla(\nabla \cdot u) \tag{6}$$

$$\partial_t \left(\epsilon + \rho \frac{\|u\|_2^2}{2}\right) + \nabla \cdot \left(\left(p + \epsilon + \rho \frac{u^2}{2}\right) u + u \cdot \sigma'\right) = 0 \tag{7}$$

where ρ represents the density, u represents the velocity, p represents the pressure, and ϵ the internal energy of the system. Here, $x \in (-1,1)$, and $t \in (0,1)$.

B Proxy Datasets

To create proxy datasets for GPT-2, GPT-2 Medium , GPT-2 Large , and GPT-2 XL, we follow the approach detailed in Shen et al. (2023). Due to discrepancies between the stated dataset and instructions in Shen et al. (2023), we use the CoNLL-2000 dataset (Sang & Buchholz, 2000) instead of CoNLL-2003. We select a random sample of 2000 sequences containing less than 32 tokens. We unify the length by padding to a sequence length of 32. Lastly, we calculate the embeddings using the selected models.

C Rescaled Results for Diffusion-Reaction and Diffusion-Sorption

We show results from Figure 2 and 3 for Diffusion-Reaction and Diffusion-Sorption at a smaller scale for better readability.

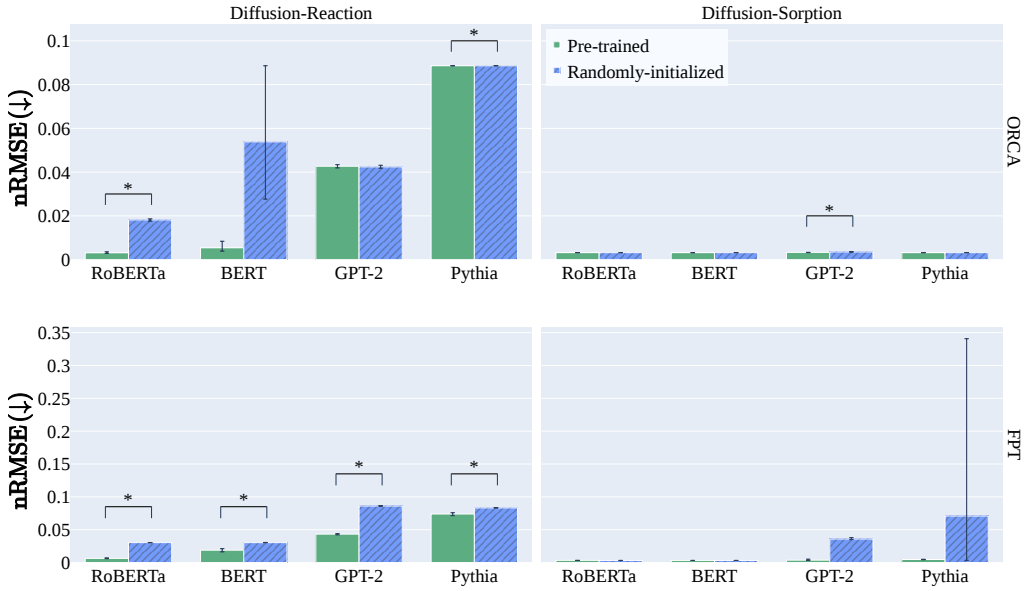

Figure 6: Comparison of model performance with ORCA- (above) and FPT-based (below) cross-modal adaptation, using both pre-trained and randomly-initialized versions of encoder-only models (RoBERTa, BERT) and decoder-only models GPT-2 and Pythia). Performance is measured using nRSME, where lower is better; the plots show average performance over five random seeds, and the error bars represent the best and worst runs. We denote with an asterisk (*) the cases in which there is a statistically significant difference between the pre-trained model and the randomly initialized one; a detailed explanation can be found in Appendix G.

D Error Analysis Examples

We show the comparison of the predicted waves for different examples of Advection and Diffusion-Reaction with different models and the ground truth wave. In Figures 8 and 9 we can see that the predictions are more spiky and irregular in the first half of the wave than in the second half.

E Our Methods with Frozen Pre-Trained Transformers

In this appendix, we include the complementary plot to Figure 5, using FPT.

F Complete Results

In this appendix, we include the results of all the runs performed. Tables 2 and 3 include the runs comparing the performance of the encoder-only models (RoBERTa, BERT) versus the decoder-only GPT-2 and Pythia), both pre-trained and randomly-initialized. Tables 4, 5, and 6 include the comparison of the different sizes of the decoder-only models (GPT-2 and Pythia), with the original setup and both of our newly introduced methods (Parallel Flipping and Sequence Doubling).

Table 2: Model performance of all five runs with ORCA- and FPT-based cross-modal adaptation, using both pre-trained and randomly-initialized versions of encoder-only models (RoBERTa, BERT) and decoder-only models GPT-2 and Pythia) for Advection and Diffusion-Reaction.

Methods	Models	Advection		Diffusion-Reaction	
		Pre-Trained	Random	Pre-Trained	Random
ORCA	RoBERTa	**8,43E-03 ± 2,86E-04**	1,28E-02 ± 2,40E-03	**3,11E-03 ± 2,97E-04**	1,82E-02 ± 3,51E-04
	BERT	1,80E-02 ± 5,08E-03	1,54E-02 ± 6,42E-04	5,40E-03 ± 1,74E-03	5,38E-02 ± 3,19E-02
	GPT-2	5,08E-01 ± 6,94E-03	4,89E-01 ± 5,05E-03	4,32E-02 ± 5,11E-04	4,24E-02 ± 6,40E-04
	Pythia	2,86E-01 ± 3,90E-03	3,11E-01 ± 5,96E-03	8,86E-02 ± 2,15E-06	8,86E-02 ± 1,03E-05
FPT	RoBERTa	1,45E-01 ± 1,59E-01	8,19E-01 ± 5,75E-02	6,27E-03 ± 4,48E-04	3,03E-02 ± 1,24E-04
	BERT	1,89E-01 ± 9,98E-02	6,09E-01 ± 5,31E-02	1,88E-02 ± 1,74E-03	3,03E-02 ± 1,92E-04
	GPT-2	7,48E-01 ± 1,00E-02	1,00E+00 ± 2,73E-02	1,65E-01 ± 7,33E-02	8,61E-02 ± 1,06E-04
	Pythia	5,93E-01 ± 6,60E-03	6,56E-01 ± 1,31E-03	7,39E-02 ± 1,98E-03	8,44E-02 ± 1,02E-04

Table 3: Model performance of all five runs with ORCA- and FPT-based cross-modal adaptation, using both pre-trained and randomly-initialized versions of encoder-only models (RoBERTa, BERT) and decoder-only models GPT-2 and Pythia) for Diffusion-Sorption and Navier-Stokes.

Methods	Models	Diffusion-Sorption		Navier-Stokes	
		Pre-Trained	Random	Pre-Trained	Random
ORCA	RoBERTa	**3,18E-03 ± 2,82E-06**	**3,18E-03 ± 7,57E-07**	**5,22E-02 ± 7,92E-03**	7,74E-01 ± 3,84E-01
	BERT	**3,18E-03 ± 1,28E-06**	3,19E-03 ± 7,75E-06	9,89E-01 ± 5,04E-03	2,76E-01 ± 4,01E-01
	GPT-2	3,29E-03 ± 4,41E-05	3,52E-03 ± 5,18E-05	5,01E-01 ± 1,64E-03	4,77E-01 ± 4,30E-03
	Pythia	**3,18E-03 ± 2,00E-06**	**3,18E-03 ± 1,33E-06**	4,86E-01 ± 6,54E-03	5,03E-01 ± 4,80E-03
FPT	RoBERTa	3,29E-03 ± 4,26E-05	3,20E-03 ± 8,11E-06	2,90E-01 ± 9,37E-03	3,28E-01 ± 2,40E-03
	BERT	3,20E-03 ± 9,70E-06	3,20E-03 ± 5,54E-06	2,82E-01 ± 4,17E-03	3,29E-01 ± 1,40E-03
	GPT-2	3,62E-03 ± 8,84E-04	3,59E-02 ± 1,22E-03	7,03E-01 ± 1,79E-02	6,49E-01 ± 1,40E-02
	Pythia	4,66E-03 ± 2,94E-04	7,09E-02 ± 1,51E-01	4,15E-01 ± 7,96E-04	4,20E-01 ± 4,84E-04

Table 4: Model performance of all five runs with ORCA- and FPT-based cross-modal adaptation, with different sizes of decoder-only models GPT-2 and Pythia) for **Advection**, using both the Original setup and our two new approaches.

Model Family	Size	ORCA			FPT		
		Original Setup	Parallel Flipping (ours)	Sequence Doubling (ours)	Original Setup	Parallel Flipping (ours)	Sequence Doubling (ours)
GPT-2	137M	5,08E-01 ± 6,94E-03	1,09E-01 ± 4,99E-03	**1,81E-02 ± 1,98E-03**	7,48E-01 ± 1,00E-02	**5,51E-01 ± 1,88E-02**	6,25E-01 ± 3,90E-02
	380M	4,95E-01 ± 4,07E-03	1,44E-01 ± 8,39E-02	**2,04E-02 ± 3,71E-03**	7,97E-01 ± 1,34E-02	**5,32E-01 ± 4,28E-02**	6,77E-01 ± 1,92E-02
	812M	5,23E-01 ± 1,59E-02	1,19E-01 ± 9,63E-03	**2,49E-02 ± 8,03E-03**	8,54E-01 ± 9,31E-03	**6,29E-01 ± 2,28E-02**	7,65E-01 ± 1,71E-02
	1.61B	5,42E-01 ± 1,01E-02	1,39E-01 ± 9,11E-03	**8,83E-02 ± 6,41E-03**	8,53E-01 ± 6,69E-03	**5,98E-01 ± 3,87E-02**	7,69E-01 ± 1,96E-02
Pythia	14M	2,87E-01 ± 3,44E-03	4,38E-02 ± 1,55E-03	**2,83E-03 ± 9,42E-05**	6,52E-01 ± 3,72E-03	2,95E-01 ± 6,70E-03	**2,54E-02 ± 1,92E-03**
	70M	2,75E-01 ± 1,68E-02	3,98E-02 ± 1,39E-03	**3,02E-03 ± 1,39E-04**	5,97E-01 ± 3,78E-03	2,35E-01 ± 7,53E-03	**7,25E-03 ± 9,94E-04**
	160M	2,86E-01 ± 3,90E-03	5,20E-02 ± 9,22E-04	**3,72E-03 ± 3,51E-04**	5,93E-01 ± 6,60E-03	2,29E-01 ± 1,44E-04	**1,58E-02 ± 5,84E-03**
	410M	2,97E-01 ± 3,21E-03	6,43E-02 ± 4,46E-03	**6,47E-03 ± 2,81E-04**	5,55E-01 ± 3,37E-03	2,07E-01 ± 3,19E-03	**8,88E-03 ± 2,74E-04**
	1B	3,95E-01 ± 4,64E-02	8,72E-02 ± 7,45E-03	**1,27E-02 ± 2,95E-04**	5,43E-01 ± 6,10E-03	2,16E-01 ± 6,19E-03	**2,04E-02 ± 9,10E-03**
	1.4B	3,14E-01 ± 4,22E-03	9,50E-02 ± 2,10E-02	**1,23E-02 ± 2,08E-04**	5,37E-01 ± 8,54E-03	2,19E-01 ± 2,83E-03	**1,69E-02 ± 5,83E-03**

Table 5: Model performance of all five runs with ORCA- and FPT-based cross-modal adaptation, with different sizes of decoder-only models GPT-2 and Pythia) for **Diffusion-Reaction**, using both the Original setup and our two new approaches.

Model Family	Size	ORCA			FPT		
		Original Setup	Parallel Flipping (ours)	Sequence Doubling (ours)	Original Setup	Parallel Flipping (ours)	Sequence Doubling (ours)
GPT-2	137M	4,32E-02 ± 5,11E-04	2,21E-02 ± 7,95E-04	**1,48E-02 ± 2,07E-03**	1,65E-01 ± 7,33E-02	7,18E-02 ± 6,40E-04	**1,57E-02 ± 1,32E-03**
	380M	4,17E-02 ± 1,20E-03	2,10E-02 ± 5,83E-04	**1,34E-02 ± 6,08E-04**	3,85E-02 ± 8,26E-04	6,99E-02 ± 1,40E-03	**1,25E-02 ± 1,06E-03**
	812M	4,30E-02 ± 1,90E-03	1,90E-02 ± 9,89E-04	**7,08E-03 ± 3,19E-04**	1,03E-01 ± 3,91E-02	7,29E-02 ± 4,77E-04	**6,85E-03 ± 4,91E-04**
	1.61B	3,89E-02 ± 5,93E-04	1,81E-02 ± 3,32E-04	**6,25E-03 ± 3,33E-04**	9,05E-02 ± 1,80E-02	7,41E-02 ± 5,38E-04	**6,97E-03 ± 4,26E-04**
Pythia	14M	8,84E-02 ± 3,07E-04	8,86E-02 ± 5,00E-04	**8,83E-02 ± 3,97E-04**	8,87E-02 ± 6,75E-05	**8,86E-02 ± 9,50E-05**	8,87E-02 ± 1,24E-04
	70M	3,46E-02 ± 2,26E-03	1,58E-02 ± 1,60E-03	**4,82E-03 ± 2,75E-04**	8,11E-02 ± 6,21E-04	7,51E-02 ± 1,11E-03	**2,02E-02 ± 7,62E-04**
	160M	**8,86E-02 ± 2,15E-06**	8,85E-02 ± 1,70E-05	**8,86E-02 ± 2,18E-06**	7,39E-02 ± 1,98E-03	6,20E-02 ± 1,75E-03	**1,37E-02 ± 1,12E-03**
	410M	3,09E-02 ± 9,22E-04	1,42E-02 ± 4,95E-04	**4,33E-03 ± 5,96E-04**	5,43E-02 ± 1,53E-03	3,84E-02 ± 1,65E-03	**9,06E-03 ± 3,37E-04**
	1B	3,30E-02 ± 1,16E-03	1,53E-02 ± 3,81E-04	**4,69E-03 ± 3,15E-04**	5,71E-02 ± 6,68E-03	4,18E-02 ± 7,66E-03	**1,04E-02 ± 1,45E-03**
	1.4B	3,24E-02 ± 8,12E-04	1,50E-02 ± 5,06E-04	**4,71E-03 ± 2,87E-04**	5,33E-02 ± 7,01E-03	3,99E-02 ± 9,06E-03	**1,25E-02 ± 2,88E-03**

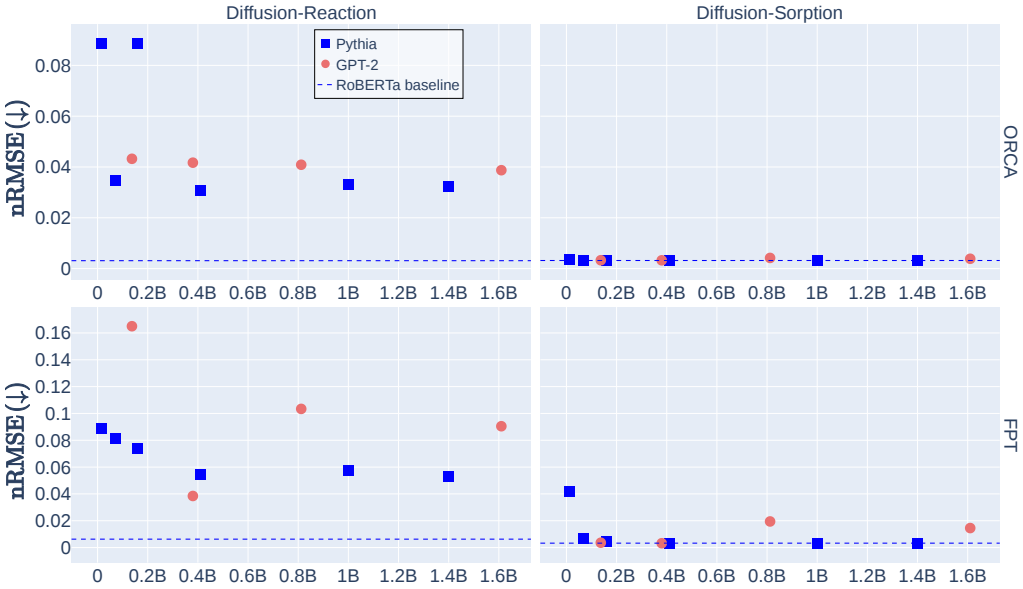

Figure 7: Performance of different sizes of models of the GPT-2 family and PYTHIA family using both ORCA (Shen et al., 2023) and FPT (Lu et al., 2022). The plots depict the average performance over five random seeds. Once again, performance is measured using nRSME, where lower is better. If scaling the models was improving the performance, downward trends could've been seen for the different model families.

Table 6: Model performance of all five runs with ORCA- and FPT-based cross-modal adaptation, with different sizes of decoder-only models GPT-2 and PYTHIA) for **Navier-Stokes**, using both the Original setup and our two new approaches.

Model Family	Size	ORCA			FPT		
		Original Setup	Parallel Flipping (ours)	Sequence Doubling (ours)	Original Setup	Parallel Flipping (ours)	Sequence Doubling (ours)
GPT-2	137M	5,01E-01 ± 1,64E-03	2,88E-01 ± 2,50E-03	**9,44E-02 ± 5,63E-03**	7,03E-01 ± 1,79E-02	**5,37E-01 ± 5,59E-03**	5,79E-01 ± 3,59E-02
	380M	4,89E-01 ± 1,30E-03	2,87E-01 ± 1,53E-03	**1,34E-01 ± 4,92E-03**	**4,82E-01 ± 2,35E-03**	5,10E-01 ± 8,90E-03	4,92E-01 ± 1,83E-02
	812M	4,87E-01 ± 2,60E-03	2,94E-01 ± 1,03E-03	**1,05E-01 ± 5,95E-03**	8,02E-01 ± 8,25E-03	**5,45E-01 ± 1,11E-02**	7,10E-01 ± 4,69E-02
	1.61B	4,74E-01 ± 3,30E-03	3,13E-01 ± 1,32E-02	**1,17E-01 ± 1,25E-02**	8,29E-01 ± 1,21E-02	**5,67E-01 ± 1,28E-02**	7,90E-01 ± 1,73E-02
Pythia	14M	4,62E-01 ± 6,25E-03	2,80E-01 ± 1,91E-03	**1,47E-01 ± 9,26E-03**	4,52E-01 ± 4,80E-04	4,45E-01 ± 7,73E-04	**3,87E-01 ± 1,69E-03**
	70M	4,96E-01 ± 2,04E-03	2,81E-01 ± 1,89E-03	**1,09E-01 ± 4,01E-03**	4,23E-01 ± 8,73E-04	4,10E-01 ± 8,61E-04	**3,45E-01 ± 2,29E-03**
	160M	4,86E-01 ± 6,54E-03	2,81E-01 ± 1,34E-03	**7,12E-02 ± 4,92E-03**	4,15E-01 ± 7,96E-04	3,80E-01 ± 1,48E-02	**3,11E-01 ± 1,26E-02**
	410M	4,39E-01 ± 4,66E-03	2,70E-01 ± 2,24E-03	**6,25E-02 ± 8,78E-03**	4,14E-01 ± 9,43E-04	3,24E-01 ± 4,41E-03	**2,49E-01 ± 4,71E-03**
	1B	4,05E-01 ± 7,73E-04	3,80E-01 ± 6,76E-02	**7,82E-02 ± 5,19E-03**	4,30E-01 ± 7,57E-03	3,45E-01 ± 8,66E-03	**2,69E-01 ± 6,57E-03**
	1.4B	4,07E-01 ± 1,11E-03	3,53E-01 ± 7,83E-02	**1,79E-01 ± 5,79E-02**	4,45E-01 ± 3,36E-03	3,34E-01 ± 1,11E-02	**2,64E-01 ± 8,38E-03**

G STATISTICAL SIGNIFICANCE

To be able to select the proper test for statistical significance, we first performed the Shapiro-Wilk test to assess if the obtained results from the five runs for each configuration were normally distributed. Once we determined this, we utilized two different statistical significance testing methods, the Wilcoxon test (for the cases in which one of the two distributions was not normally distributed) and the t-test (when both distributions were normally distributed), both of them two-sided. For all statistical test ran, we set the p-value to 0.05.

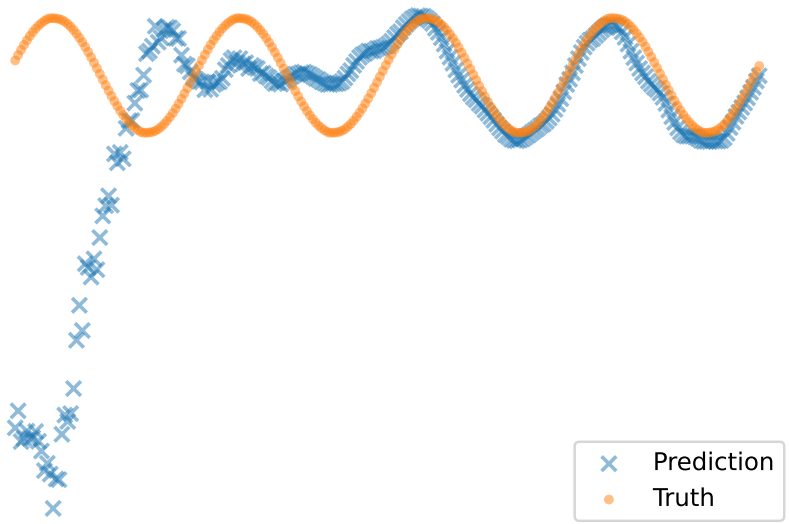

Figure 8: Comparison between GPT-2 prediction on an Advection example using ORCA as the cross-modal adaptation method and the ground truth.

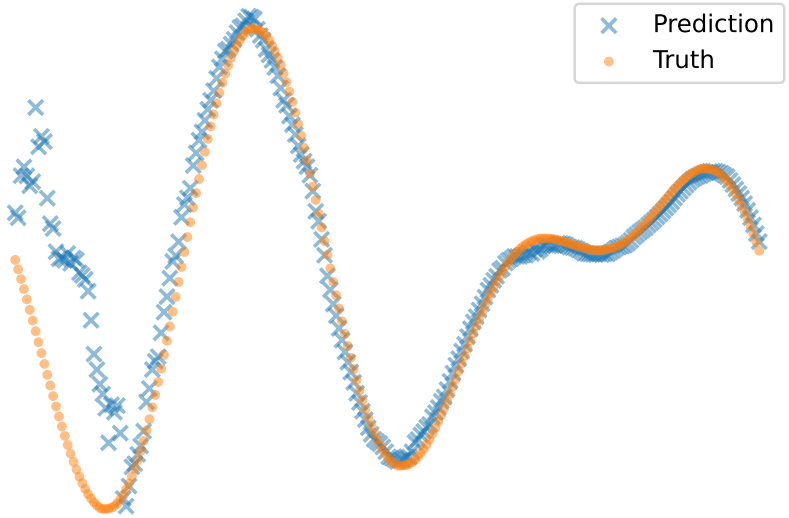

Figure 9: Comparison between PYTHIA prediction on an Advection example using ORCA as the cross-modal adaptation method and the ground truth.

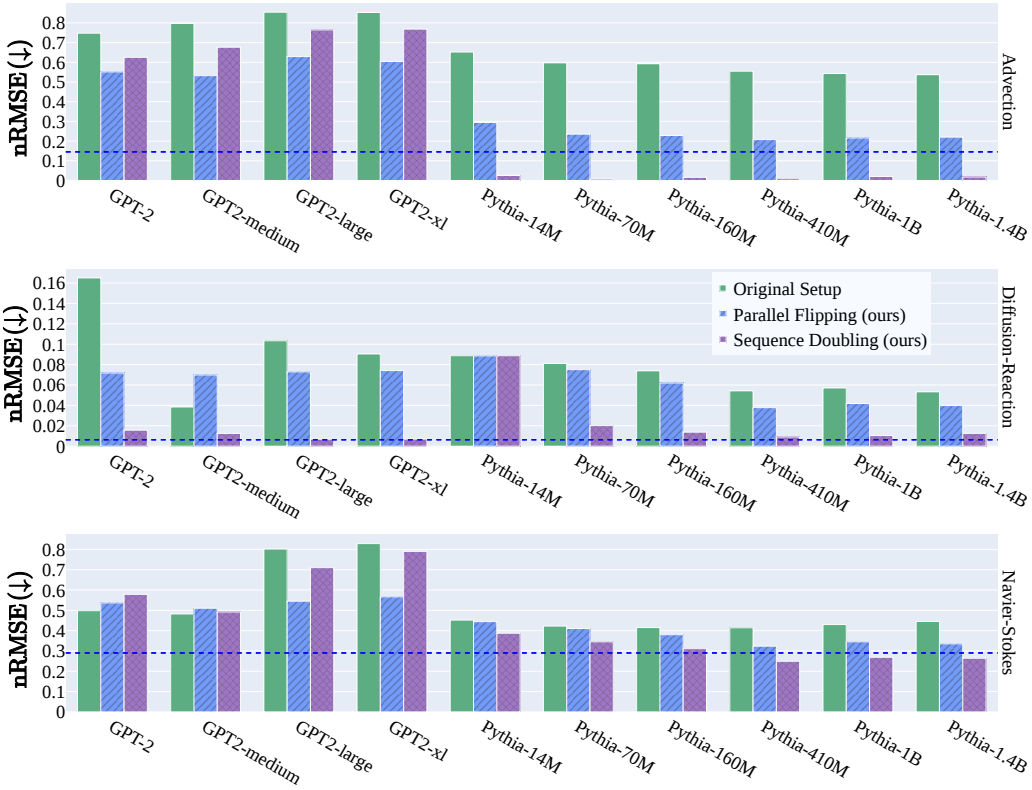

Figure 10: Performance comparison of the original setup versus our own two methods, Parallel Flipping and Sequence Doubling, using FPT (Lu et al., 2022). We set RoBERTa with the original setup as a baseline for all the configurations. The plots depict the average performance over five random seeds. Performance is measured using nRSME, where lower is better.