An in-context approach to estimate the parameters of a stochastic process

Aziz Temirkhanov[a], Kseniia Kuvshinova[a], Dmitry Simakov[a]

[a] *Sber AI Lab, Moscow, Russia,* temirkhanovmail@gmail.com

1. Introduction

Stochastic processes [1] are essential in finance [2, 3], engineering [4, 5], and natural sciences [6, 7, 8, 9, 10]. In financial mathematics, the use of stochastic processes, mainly modeled by SDEs, depends on accurate parameter estimation [11].

MLE is widely used but relies on careful parameter initialization and knowledge of the exact parent distribution [12]. Building on [13, 14], we propose a zero-shot model to estimate the volatility σ, mean reversion rate θ, and long-term mean μ from observed points of an Ornstein–Uhlenbeck (OU) process. We focus on the OU process due to its broad applications in physics [15] and finance [16, 17].

2. Related Work

Estimating the parameters of stochastic differential equations (SDE) is a problematic task [18]. This research problem is important in the field of finance [19]. MLE has been a foundational method for parameter estimation in stochastic processes. For instance, [20] discusses MLE within the context of stochastic dynamical models. Recent work has shown that combining the Least squares estimator (LSE) with other estimation frameworks can enhance its performance in stochastic settings [21, 22]. Studies have demonstrated the effectiveness of generalized method of moments (GMM) in estimating the parameters of SDE driven by Brownian motion [23].

Advancements in neural networks have sparked growing interest among researchers in this approach. In [24, 25, 26, 27], the authors propose a neural network-based approach to estimate SDE parameters, addressing the limitations of MLE and GMM approaches, which often require parameter tuning.

Recent foundation models for time series [28, 29] automate representation learning, yet challenges persist, particularly in the zero-shot inference of random process parameters [30].

3. Framework

General setup. We adopt the in-context learning approach from [14] and [13], leveraging a general framework for training transformer models [31] on synthetic data with several key modifications. Unlike [13], our experiments employ a transformer encoder rather than a decoder. In contrast to [14], we pre-train our model not on data with seasonal patterns but on realizations of a modified random walk, represented by OU process. Additionally, instead of focusing on forecasting, as in [14], we address stochastic process parameter extraction, similar to [30].

Synthetic data generation. In the experiments, we train and evaluate our method using the OU process realization [32]. The OU process is defined by the stochastic differential equation:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t,$$

where $\theta > 0$ is the rate of mean reversion, μ is a long-term mean toward which the process reverts, and σ is the volatility coefficient. W_t denotes the Wiener process. We sample θ, μ, and σ uniformly from the interval $(0, 20]$. The training dataset consists of a 15000 time series with a time horizon of $T = 5$ and time increments of $dt = 0.1$, yielding 500 samples per trajectory. All trajectories start at zero. To assess the model's robustness in handling time series of varying lengths, we introduce an additional parameter — the sequence length — sampled uniformly from the interval $[10, 500]$.

Model. In our setup, in-context learning refers to providing the model with points from an OU process realization as input, allowing it to infer the underlying process parameters. Our model follows a standard Transformer encoder design, as illustrated in Figure A4. More details are also described in Appendix A.

4. Experiments

Baselines. We use the MLE algorithm implemented in [33] as a baseline. We test three options of MLE: (1) vanilla MLE with initial guess = [1, 1, 1] and parameter bounds almost equal to the bounds of the sampling of the training dataset $[(0, 20), (0, 20), (0.01, 20)]$; (2) MLE with initial guess $\hat{\Psi} = [\hat{\mu}, \hat{\theta}, \hat{\sigma}]$, where $\hat{\Psi}$ is our zero-shot model's prediction for OU parameters, and bounds for each parameter $\hat{\psi}$ as $(0.75 \times \hat{\psi}, 1.25 \times \hat{\psi})$ for a corresponding parameter in $\hat{\Psi}$; (3) MLE with initial guess [1, 1, 1] and bounds obtained like in option 2 via zero-shot model predictions. These MLE modifications, as demonstrated later, highlight our primary contribution and the unique application of the in-context learning method for SDE parameter extraction.

Pipeline. The pipeline is as follows. First, we obtain the dataset consisting of OU process realizations of different lengths. Next, we train our transformer model on these synthetics to predict the underlying process parameters' MSE loss between parameters. We test the model in three different scenarios.

To evaluate the quality of our model, we estimate the parameters of the OU and then use those estimated parameters $\hat{\mu}, \hat{\theta}, \hat{\sigma}$ and the saved Wiener real-

ization to generate a new reconstructed trajectory $\hat{X}$ of the process. Then, we measured the MSE between true and reconstructed time series to conclude our experiment.

Influence of parameters. We vary a single parameter from μ, θ, σ from 0.1 to 20 in the separate test synthetic dataset while remaining the other fixed at 3. We report the changes in the MSE for the varied parameters. The results of this experiment are shown in Figure A1. These plots illustrate how the estimation quality in terms of MSE changes as one parameter is varied while all other parameters remain fixed. In the upper set of the graphs, MSE is measured between true and estimated parameters for 30 equally spaced parameter values from the $(0, 20]$. In the lower set, MSE is measured between a true and reconstructed trajectory.

Interpolation and extrapolation ability. In the second experiment, we investigate the model's ability to accurately predict parameters from the unseen regions during the training procedure. For that purpose, we train our model on realizations of the OU process obtained with a different set of parameters sampled uniformly from the interval union $\mathcal{K} \in (0, 5] \cup (10, 15]$. Next, we test our model on realizations with *known* set $\mathcal{K}$ and *unknown* set $\mathcal{U} \in (5, 10] \cup (15, 20]$ of the parameters. The result is shown in Table 1. The model's prediction errors in the unknown region $\mathcal{U}$ closely resemble those observed in the first experiment (see A1), albeit with more significant errors in extrapolation, as expected. However, we believe that further investigation is required.

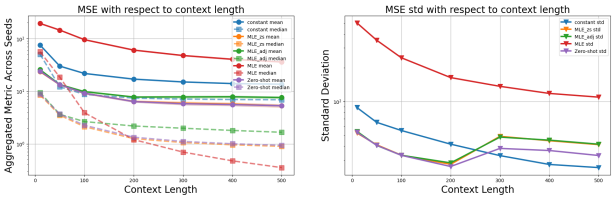


Fig. 1: Ablation on different context lengths. All values are in the log scale. The first plot reports median, mean, and 95% confidence intervals for mean aggregation. `Constant` stands for mean prediction on points in context, `MLE_zs` is for case 2 from **Baselines** (Section 4) and `MLE_adj` is for case 3. `MLE` is for vanilla MLE (case 1). For MSE on parameter values, see Figure A3 in the Appendix. For visualization of zero-shot model errors for different contexts, see Figure A2.

Context length influence. The third experiment evaluates the model's ability to handle varying context sizes. We adjust the input length of the time series within the range of 10 to 500 by applying a mask of the corresponding length to the original trajectory. The model then estimates the parameters based on the truncated trajectory. We generate a full-length reconstructed trajectory using these estimated parameters and compute the mean squared error (MSE) between the full original time series and its reconstruction.

The results are presented in Figure 1. The observation that increasing context length improves the performance of all models is fairly trivial. However, these plots offer a more valuable insight. The key takeaway is that MLE parameter estimation lacks robustness. This is evident when comparing the mean, median, and standard deviation of the MSEs on OU process realizations sampled with different seeds, as shown in Figure 1, and in the histograms of these MSEs presented in Figure B. In the first plot, we observe that, when aggregating by the mean, pure MLE with the default guess and parameter bounds performs the worst. Conversely, vanilla MLE outperforms all its modifications for median aggregation as context length increases. This suggests that vanilla MLE is influenced by significant outliers in MSE scores, resulting in heavy tails in the metric distribution and skewing the mean MSE across seeds. This hypothesis is supported by the histogram in Figure B. The histograms show MSE distributions, highlighting vanilla MLE's non-robustness with heavy tails, which worsen as context length decreases.

Nonetheless, the zero-shot approach remains the most robust overall. While vanilla MLE produces the best results when it converges, the zero-shot method becomes the optimal alternative when dealing with a large number of series, where maintaining good mean scores is essential. The zero-shot approach is more robust and yields scores comparable to MLE methods. Furthermore, by leveraging zero-shot predictions from our model to provide an adjusted guess and parameter bounds (see experiment cases 2 and 3 from the **Baselines** (Section 4) for details), we demonstrate a significant improvement in MLE's robustness.

region	(0, 5]	**(5, 10]**	(10, 15]	**(15, 20]***
parameters	5.69	**14.16**	2.93	**27.30***
trajectories	2.39	**4.36**	1.38	**5.37***

Table 1: Interpolation test. MSE between true and predicted values are reported. **Bold** font indicates test for unknown region. * indicates extrapolation.

5. Conclusion

In this paper, we propose an in-context learning-based approach for estimating the parameters of Ornstein-Uhlenbeck processes from their realizations. By leveraging our method to refine the initial conditions of classical MLE, we significantly improve its performance. Moreover, it is an independent method to extract parameters, giving comparable scores with tuned MLE (case 2). Along with similar works [14, 13, 30], it opens the new direction of in-context learning for different tasks in Artificial Intelligence (AI) for Science, such as chemistry [34], physics [30] and financial mathematics [35].

Acknowledgments

We would like to thank Maxim Kaledin and Artem Chubov for the stimulating discussion of ideas.

References

[1] John Lamperti. *Stochastic processes: a survey of the mathematical theory*, volume 23. Springer Science & Business Media, 2012.

[2] J Michael Steele. *Stochastic calculus and financial applications*, volume 1. Springer, 2001.

[3] Marek Musiela and Marek Rutkowski. *Martingale methods in financial modelling*, volume 36. Springer Science & Business Media, 2006.

[4] François Baccelli, Bartłomiej Błaszczyszyn, et al. Stochastic geometry and wireless networks: Volume ii applications. *foundations and trends® in networking*, 4(1–2):1–312, 2010.

[5] Edward R Dougherty. Random processes for image and signal processing. *(No Title)*, 1999.

[6] Paul C Bressloff. *Stochastic processes in cell biology*, volume 41. Springer, 2014.

[7] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

[8] Russell Lande, Steinar Engen, and Bernt-Erik Saether. *Stochastic population dynamics in ecology and conservation*. Oxford University Press, USA, 2003.

[9] Carlo Laing and Gabriel J Lord. *Stochastic methods in neuroscience*. Oxford University Press, 2010.

[10] Donald Lee DeAngelis. *Individual-based models and approaches in ecology: populations, communities and ecosystems*. CRC Press, 2018.

[11] Jaya PN Bishwal. *Parameter estimation in stochastic volatility models*. Springer, 2022.

[12] David Arthur Sprott. Robustness and maximum likelihood estimation. *Communications in Statistics-Theory and Methods*, 11(22):2513–2529, 1982.

[13] Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36:2403–2426, 2023.

[14] Kseniia Kuvshinova, Olga Tsymboi, Alina Kostromina, Dmitry Simakov, and Elizaveta Kovtun. Towards foundation time series model: To synthesize or not to synthesize? *arXiv preprint arXiv:2403.02534*, 2024.

[15] Don S Lemons and Paul Langevin. *An introduction to stochastic processes in physics*. JHU Press, 2002.

[16] Tomas Björk. *Arbitrage theory in continuous time*. Oxford university press, 2009.

[17] Jirat Suchato, Sean Wiryadi, Danran Chen, Ava Zhao, and Michael Yue. An application of the ornstein-uhlenbeck process to pairs trading. *arXiv preprint arXiv:2412.12458*, 2024.

[18] Sanmitra Ghosh, Paul J Birrell, and Daniela De Angelis. Differentiable bayesian inference of sde parameters using a pathwise series expansion of brownian motion. In *International Conference on Artificial Intelligence and Statistics*, pages 10982–10998. PMLR, 2022.

[19] Kalok C Chan, G Andrew Karolyi, Francis A Longstaff, and Anthony B Sanders. An empirical comparison of alternative models of the short-term interest rate. *The journal of finance*, 47(3):1209–1227, 1992.

[20] Timothy DelSole and Xiaosong Yang. State and parameter estimation in stochastic dynamical models. *Physica D: Nonlinear Phenomena*, 239(18):1781–1788, 2010.

[21] Solym Manou-Abi. Parameter estimation for a class of stable driven stochastic differential equations. 2023.

[22] Ankur Gupta and James B Rawlings. Comparison of parameter estimation methods in stochastic chemical kinetic models: examples in systems biology. *AIChE Journal*, 60(4):1253–1268, 2014.

[23] Semi Ergişi. Various parameter estimation techniques for stochastic differential equations. Master's thesis, Middle East Technical University, 2019.

[24] S. Samarasinghe Z. Xie, D. Kulasiri and C. Rajanayaka. The estimation of parameters for stochastic differential equations using neural networks. *Inverse Problems in Science and Engineering*, 15(6):629–641, 2007.

[25] Bálint Csanády, Lóránt Nagy, Dániel Boros, Iván Ivkovic, Dávid Kovács, Dalma Tóth-Lakits, László Márkus, and András Lukács. Parameter estimation of long memory stochastic processes with deep neural networks. In *ECAI 2024*, pages 2548–2555. IOS Press, 2024.

[26] Shuaiyu Li, Yang Ruan, Changzhou Long, and Yuzhong Cheng. Efficient cnn-lstm based parameter estimation of lévy driven stochastic differential equations. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 316–323. IEEE, 2023.

[27] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[28] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[29] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[30] David Berghaus, Kostadin Cvejoski, Patrick Seifner, Cesar Ojeda, and Ramses J Sanchez. Foundation inference models for markov jump processes. *arXiv preprint arXiv:2406.06419*, 2024.

[31] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[32] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[33] Justin Kirkby, Dang Nguyen, Duy Nguyen, and Nhu N Nguyen. pymle: A python package for maximum likelihood estimation and simulation of stochastic differential equations. *Journal of Statistical Software, Forthcoming*, 2025.

[34] Zekai Li, Mauricio Barahona, and Philipp Thomas. Moment-based parameter inference with error guarantees for stochastic reaction networks. *arXiv preprint arXiv:2406.17434*, 2024.

[35] Aashrit Cunchala. A basic overview of various stochastic approaches to financial modeling with examples. *arXiv preprint arXiv:2405.01397*, 2024.

[36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, 2019.

[38] Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. Generalization in nli: Ways (not) to go beyond simple heuristics, 2021.

[39] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962, 2019.

[40] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

[41] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[42] Shanghua Gao, Teddy Koker, Owen Queen, Tom Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2025.

[43] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[44] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[45] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

Appendix A. Framework

Model. Initially, all realizations of stochastic processes are processed in parallel through embedding and linear layers, producing both a [CLS] token representation and point-wise token embeddings of the series. These representations are concatenated and subsequently passed through positional encoding, Layer Normalization [36], and a transformer encoder initialized with BERT-Tiny [37, 38, 39] weights. This design choice is motivated by prior research demonstrating that initializing transformer modules with weights from pre-trained language models enhances performance in time series tasks [40, 41, 28, 42]. The outputs of the transformer module are then subjected to average pooling [43, 44], followed by a ReLU activation [45] and a final linear layer, which maps the outputs to the target dimensionality of three parameters.

Appendix B. Experiments

Results. Third experiment. Below, we present histograms of the MSE between true and reconstructed trajectories. The results highlight the significant non-robustness of vanilla MLE, as evidenced by the heavy tails in the MSE distribution. Furthermore, this issue becomes more pronounced as the context length decreases. The X-axis represents MSE values, while the Y-axis denotes their corresponding frequencies.

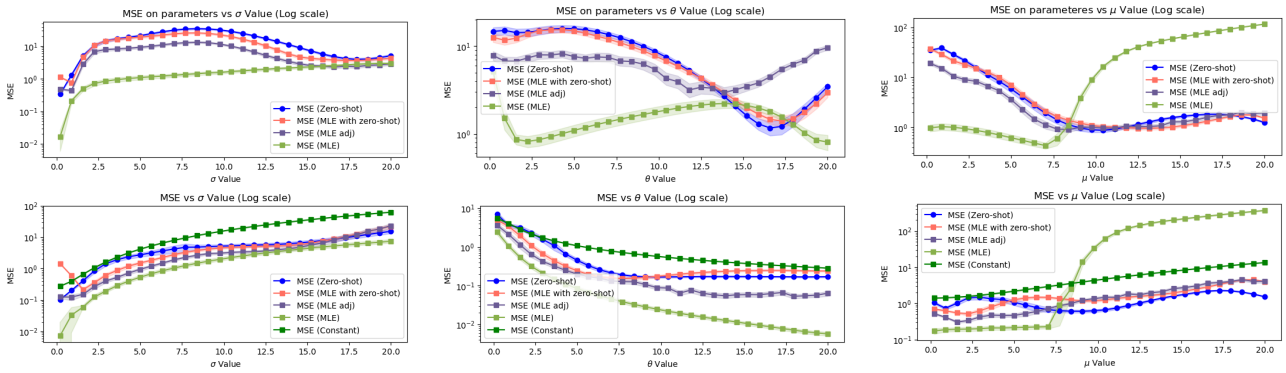


Fig. A1: Ablation on different parameter values. Mean and 95% confidence intervals are reported.

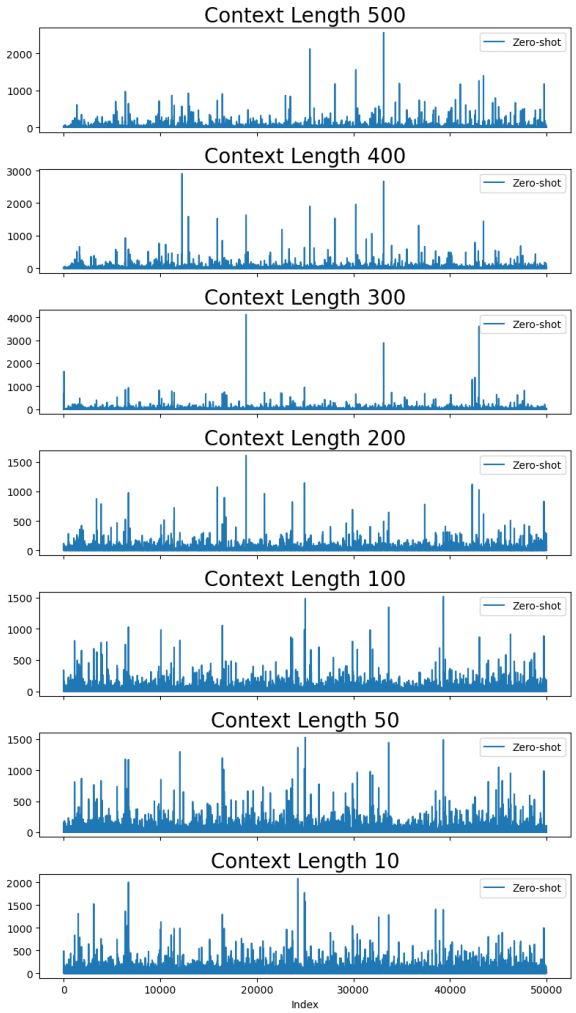


Fig. A2: Zero-shot errors for different contexts visualized for the same index. This figure explains the increase in MSE for large contexts in the second plot in Figure 1.

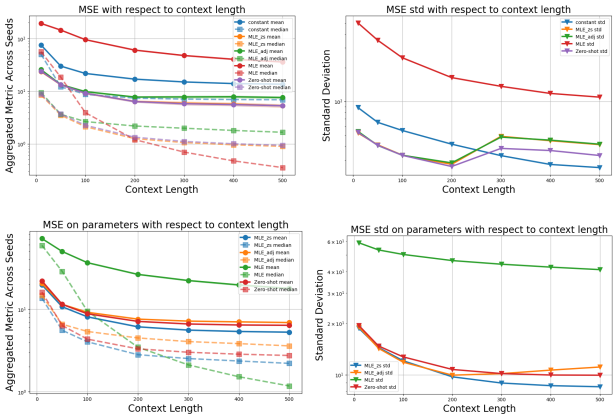


Fig. A3: Ablation on different context lengths. All values are in the log scale. The first plot reports median, mean, and 95% confidence intervals for mean aggregation. `Constant` stands for mean prediction on points in context, `MLE_zs` is for case 2 from **Baselines** (Section 4) and `MLE_adj` is for case 3. `MLE` is for vanilla MLE (case 1). For visualization of zero-shot model errors for different contexts, see Figure A2.

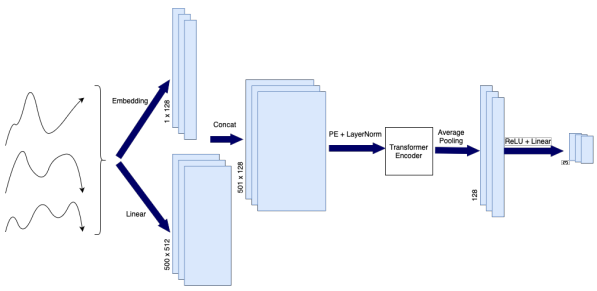
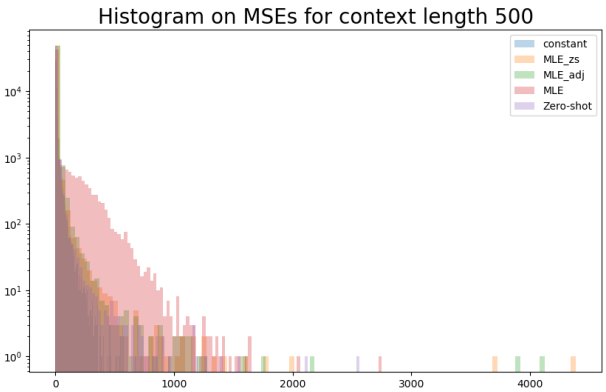
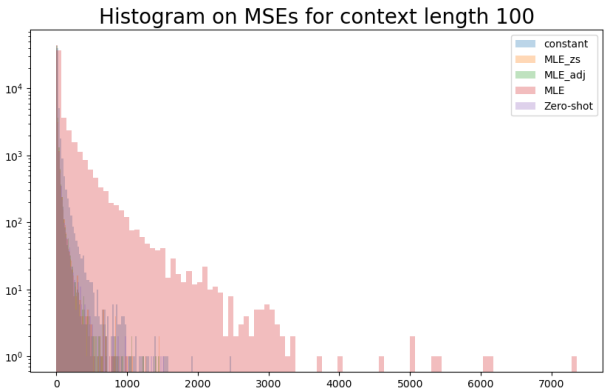
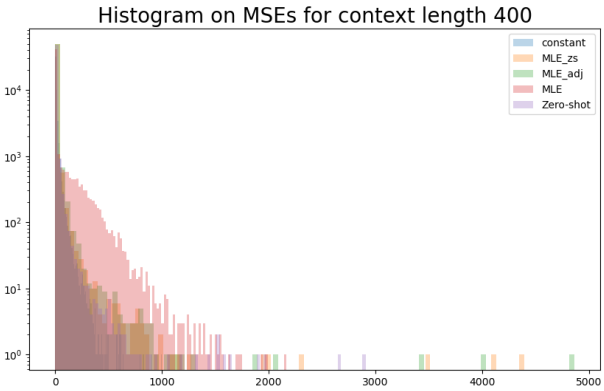
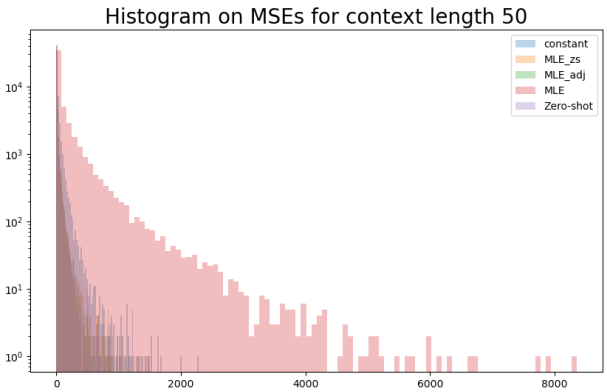


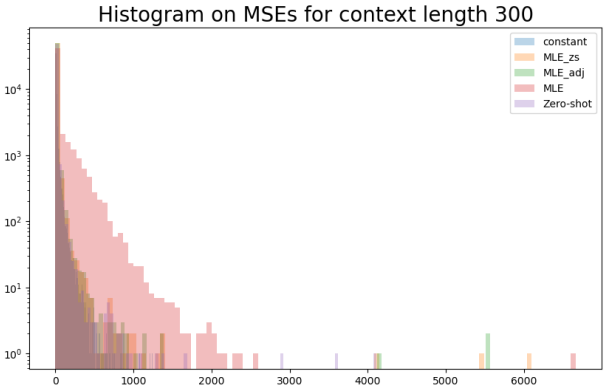
Fig. A4: Our architecture

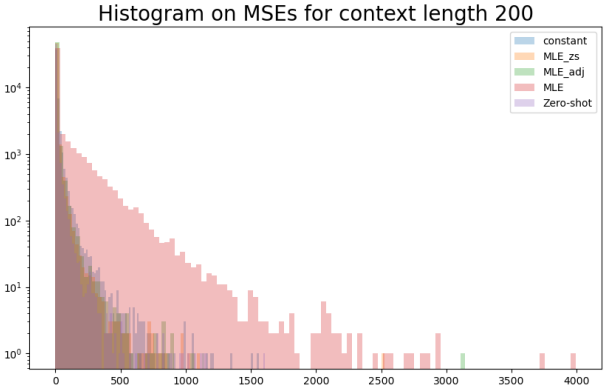

Histogram on MSEs for context length 500
constant
MLE_zs
MLE_adj
MLE
Zero-shot

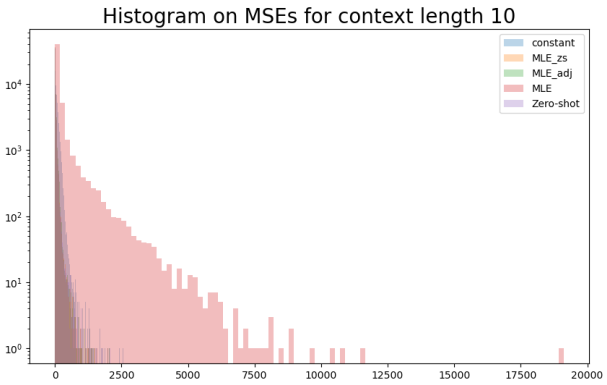

Histogram on MSEs for context length 100
constant
MLE_zs
MLE_adj
MLE
Zero-shot


Histogram on MSEs for context length 400
constant
MLE_zs
MLE_adj
MLE
Zero-shot


Histogram on MSEs for context length 50
constant
MLE_zs
MLE_adj
MLE
Zero-shot


Histogram on MSEs for context length 300
constant
MLE_zs
MLE_adj
MLE
Zero-shot


Histogram on MSEs for context length 200
constant
MLE_zs
MLE_adj
MLE
Zero-shot


Histogram on MSEs for context length 10
constant
MLE_zs
MLE_adj
MLE
Zero-shot