Scaling-laws for Large Time-series Models

Thomas D. P. Edwards
Johns Hopkins University
tedwar42@jhu.edu

James Alvey
University of Cambridge
University of Amsterdam
j.b.g.alvey@uva.nl

Justin Alsing
Stockholm University
Calda AI
justin@calda.ai

Nam H. Nguyen
Capital One
nam.nguyen@capitalone.com

Benjamin D. Wandelt
Institut d'Astrophysique de Paris
CCA, Flatiron Institute
bwandelt@iap.fr

Abstract

Scaling laws for large language models (LLMs) have provided useful guidance in training ever larger models for predictable performance gains. Time series forecasting shares a similar sequential structure to language, and is amenable to large-scale transformer architectures. Here we show that foundational decoder-only time series transformer models exhibit analogous scaling-behavior to LLMs, with architectural details (aspect ratio and number of heads) having a minimal effect over broad ranges. We assemble a large corpus of heterogenous time series data on which to train, and establish for the first time power-law scaling with parameter count, dataset size, and training compute, spanning five orders of magnitude.

1 Introduction

Time-series forecasting is fundamental to decision-making and scientific inference across all domains involving time-ordered observations. In fact, making probabilistic forecasts given past data (whether explicitly or implicitly) arguably underpins every human decision [1–5]. In industrial and scientific settings, time-series forecasting has traditionally involved supervised training of either statistical models (e.g., ARIMA, GARCH, state-space models, and others; see [6, 7] for reviews), bespoke dynamical models based on domain-specific knowledge, or more recently deep-learning based approaches trained for a specific forecasting task (see [8] for a review). While these approaches have formed the bedrock of time-series analysis up until now, key challenges and limitations remain: statistical models often fail to describe and capture the latent processes underlying the data, hampering their predictive utility; developing specialized problem-specific models requires considerable investment in human time and resources; and supervised deep-learning approaches trained on a single dataset are typically only useful in the data-rich regime, and generalize poorly to other problems.

The emergence of large language (LLMs; [9–12]) and computer vision models [13–19] with zero-shot prediction capabilities has sparked an interest in developing foundation models for time-series — general purpose forecasting models, pre-trained on a large and diverse corpus of time-series data, aimed at achieving state-of-the-art (SOTA) zero-shot forecasting performance across many domain areas [20–35]. Large time-series models (LTMs) are already achieving zero-shot prediction capability similar to or better than baseline statistical or domain-specific models in many areas [20–28, 34].

Underpinning the investment and subsequent success of LLMs and large-scale computer vision models was the demonstration of neural scaling laws [36–40, 18, 41]. The observed power-law scaling of test loss with model size, compute resources, and training set size, has provided a basis for predicting expected gains from different efforts, aiding the community in allocating resources

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

appropriately to achieve performance breakthroughs. Given qualitative differences in data and modeling challenges, existence of neural scaling-laws for time-series is not guaranteed from the language and computer vision results. Establishment of similarly favourable scaling laws for LTMs would serve as a motivation and guide in the pursuit of foundation models for time-series forecasting.

Contributions: We establish for the first time that LTMs enjoy similar power-law scaling laws to language and computer vision. We train decoder-only transformer models (with architectures tailored to time-series forecasting; §2) on a large, diverse, and well-balanced dataset comprising around 8 billion data points across 30,211,687 individual time-series, drawn from 38 qualitatively distinct data sources from varied areas (see §2). We demonstrate power-law like scaling behavior of model performance with model size, compute, and dataset size (Fig. 1), finding similar scaling behavior in three key measures of model performance: the mean-square error (MSE) characterizing the accuracy of point (posterior mean) forecasts; the Continuous Ranked Probability Score (CRPS [42]) characterizing the fidelity of the probabilistic predictions (ie., coverage of the forecast posterior density); and the log-likelihood loss characterizing the Kullback-Leibler (KL) divergence between the model and data generative distributions.

2 Data and Methods

Data: The development of a foundation model for time-series forecasting is predicated on the availability of a sufficiently large, diverse, and well-balanced dataset to train on. We constructed a corpus of time-series data comprising around 8 billion data points drawn from 38 varied data sources (see Tab. 1). For the purpose of this study, our focus was to ensure our dataset is: large enough so that for our largest models (~ 100M parameters) we are still operating in the $\sim$infinite data limit (c.f. [36]); as diverse as possible given the practical limitations on publicly available data; and well-balanced, such that no individual dataset comprises more than roughly 15% of the total number of data points. Our resulting dataset is competitive with the SOTA in terms of both diversity and size[1], while covering a wide variety of sampling frequencies, record lengths, dynamic ranges, and underlying latent process phenomenology. We focus exclusively on univariate time-series forecasting, and leave the study of scaling-laws for multivariate LTMs to future work. Data sources, balancing procedure and pre-processing steps are detailed in Appendix A.

Model Architecture: We use decoder-only transformer models with self-attention as the primary architecture throughout, with a context length of 256 data points and ReLU activation functions. Following the performance gains shown in Ref. [26], we use a learnable encoding rather than the sinusoidal positional encoding used in the original transformer model [43]. Both the learned positional encoding and embedding are simple linear layers going from one input to d_{m} outputs.

Distribution Head: We use a Student's-t distribution head, where the mean, variance, and degrees of freedom are each modelled by a separate dense network with four hidden layers of dimension d_{m}. The Student's-t distribution allows us to model heavy-tailed data, which we find in experiments enables significantly more stable training than a Gaussian head or MSE loss. We note that in reality, time-series data and processes exhibit diverse distributional characteristics, and a more expressive distribution head (e.g., mixture model, normalizing flow or diffusion model) is well-motivated. We leave the exploration of scaling-laws under more expressive distribution heads to future work. We use a negative log-likelihood (KL) loss for training throughout.

Parameter Counting: With this setup, the model architecture is defined by the following parameters: the number of output dimensions θ_{out}, the input/output size of the linear layers in the self-attention d_{m}, the number of heads N_{heads}, the hidden layer size of the linear layers directly after the self-attention d_{ff}, and the number of decoder layers N_{l}. Throughout this work we fix $d_{\mathrm{m}} = d_{\mathrm{ff}}$ and treat all trainable parameters (including weights and biases of all layers) equally in the parameter counting. As shown in Fig. 1, we explore models with $\sim 10^3$ to $\sim 10^8$ trainable parameters.

Learning rate and architecture sensitivity: To extract reliable scaling laws, we need to determine sensitivity and robustness to the learning rate (LR) schedule and architecture choices. We use a linear warm up followed by sinusoidal decay for the learning rate scheduling, and find that the model

[1]Where other SOTA time-series datasets from recent studies are larger, this discrepency is mostly accounted for by the use of synthetic data (which we deliberately do not include), or reduction in our total data count from re-balancing the data to ensure it is not dominated by a single source.

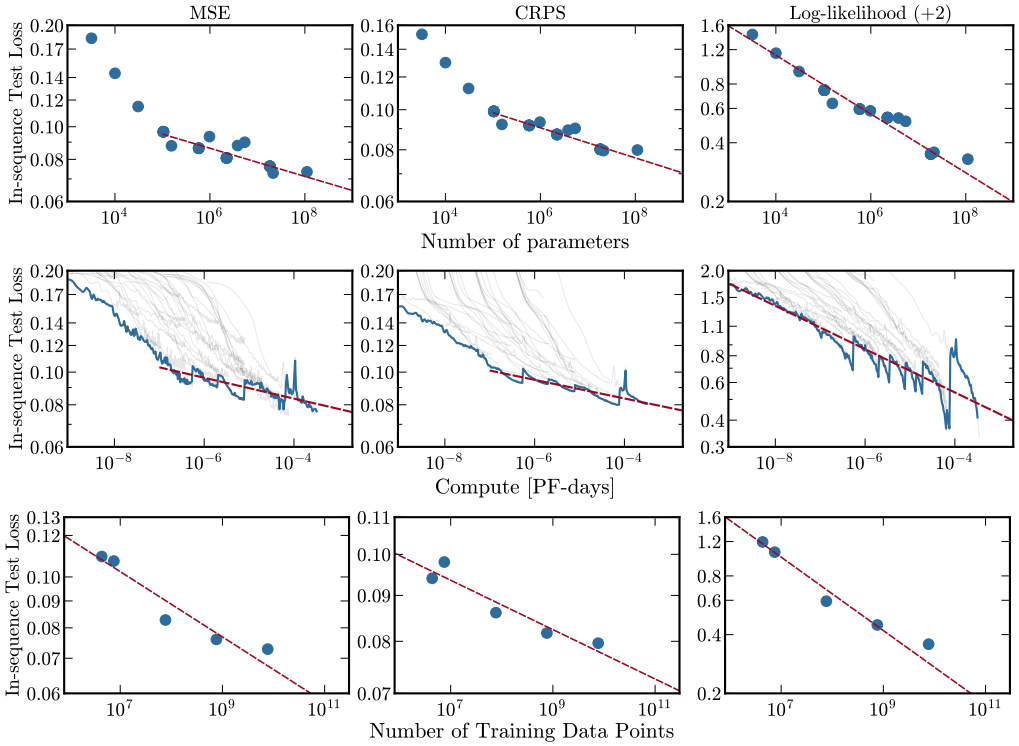

Figure 1: **Test Loss Scaling Laws:** Minimum MSE (left), CRPS (middle) and log-likelihood (right) in-sequence test metrics as a function of the number of parameters (top), compute (middle), and dataset size (bottom).

performance clearly depends on the maximum LR reached at the end of the warm-up. To ensure robustness to the maximum LR, we fit a power-law to the best model at each parameter size to estimate the optimal maximum LR as a function of parameter count, shown in Fig. 3 (Appendix C).

Figure 4 (Appendix C) shows how the minimum CRPS varies as a function of aspect ratio d_m/N_l (left panel) and the number of attention heads, N_{heads} (right panel). Performance is $\sim$insensitive to the number of heads, and only weakly sensitive to aspect ratio for aspect ratios $\lesssim 100$ (after which performance drops steeply). We note that this is analogous to the weak architecture sensitivity observed for LLMs [36]. For the main parameter-, compute-, and data-scaling runs, we fix the number of heads to four, and keep the aspect ratio < 70. See Appendix B for further training details.

3 Results

Scaling as a function of parameter count N_p, dataset size $\mathcal{D}$, and compute $\mathcal{C}$ is summarized in Fig.1. For each scaling-relation, we fit a power law of the form $\ln L(A) = -B_0 \ln A + B_0 \ln A_0$, where L is the objective function (MSE, CRPS, or log-likelihood) and A is the scaled quantity (i.e., parameter count, dataset size, or compute). The fitted parameter values are given in Tab. 7 (Appendix D). Where broken power-law like scaling is observed, we report the power law fit after the break only, since this is the relevant quantity to motivate extrapolation to larger models / datasets / compute resources.

Parameter scaling: Fig. 1 (top row) shows the minimum in-sequence test loss (MSE, CRPS, and log-likelihood[2]) as a function of parameter count, showing $\sim$power-law behavior over nearly five orders of magnitude in model size. A mild break is observed in the power-law behavior in both the MSE and CRPS test losses, indicating qualitatively different behavior for smaller models. In contrast, little or no break is seen in the log-likelihood scaling; this qualitative difference relative to MSE and

[2]We add a constant factor of two to the log-likelihood to ensure values are always positive, enabling us to examine power-law scaling. A constant additive factor can change the slope of the fitted power-law; to remain as agnostic as possible we choose to add the smallest integer required to make all values of the loss positive.

CRPS is likely due to the log-likelihood being more sensitive to variations in the tails of the forecast distribution (see e.g., [44]). The observed scaling over many orders of magnitude demonstrates that LTMs are likely to reach SOTA performance given enough data and model size.

Data Scaling: Extracting reliable scaling behavior with dataset size requires keeping the data-diversity fixed, i.e., each dataset's relative contribution to the total data count should remain the same under scaling (see Tab. 1). For time-series that are significantly longer than our context length, we use a randomly chosen portion, f_d, of each time-series, while for series that would become shorter than our context length once cut, we instead randomly drop the entire series with probability equal to $1 - f_d$. We compute the test loss over the full test set to allowing direct comparison between runs with different values of f_d, and reduce the noise on the test loss in the small (scaled) dataset limit.

Results are shown in the bottom row of Fig. 1 where we train a ~ 20M parameter model using the optimum max learning rate found during the parameter scaling exploration, with early stopping. We find power-law scaling across four orders of magnitude in all three performance measures.

Compute Scaling: The compute at any given stage in the training process is given by $\mathcal{C} = 6BN_{\mathrm{p}}L_{\mathrm{seq}}$, where B is the batch size, N_{p} is the number of parameters in the model, and L_{seq} is the context length [36]. Test losses as a function of compute are shown in Fig. 1 (middle row), where the scaling law is obtained from the minimum test loss attained at any given value of $\mathcal{C}$. Although we see a significant amount of noise in the loss functions during training, there is a clear overall trend towards lower test losses for higher compute, which is well-described by a power law. Similarly to the parameter scaling we see a mild break at low compute values for both the MSE and CRPS test losses. Note that while the MSE and CRPS metrics appear to be approximately converged over the compute range considered, the log-likelihood may not be fully converged; additional training may be needed to obtain accurate compute scaling-law fits for the log-likelihood in the large model limit.

4 Discussion

We have focused on evaluating models by their in-sequence (next step) test loss, rather than explicitly assessing model's ability to forecast further into the future. The implicit assumption that a model with good in-sequence predictions should naturally be able to forecast into the future is theoretically and empirically well-motivated: as the modelled posterior predictive distribution for the next value improves, any accumulated errors from auto-regressive roll-out those predictions into the future should also improve. In App. E, we show some clear examples of how forecast roll-out becomes increasingly coherent with increasing model size. We leave the study of scaling-laws based on forecasting ability on different time horizons to future work.

We have detailed the specific scaling laws for a decoder-only transformer with self-attention. However, it would be interesting to explore how modifications to this architecture might improve model scaling. In particular, much of the recent progress in using LTMs [20–28, 34] has involved various changes to transformer architectures to make them more suited to time-series data. We advocate for comparative scaling law studies as new architectures are introduced, to allow the community to evaluate which model architectures will eventually reach SOTA zero-shot prediction capabilities.

When experimenting with data scaling, we found it was critical to scale the training data in such a way as to preserve the data diversity; approaches to data scaling that did not preserve data diversity failed to reveal any clear scaling behaviour. Given the importance of data diversity in establishing data scaling laws, and in training SOTA pre-trained foundation models in general, developing a robust framework for quantifying data diversity would be of great utility to the field.

One scaling law that we have not explored in this work (due to computational limitations) is performance as a function of increasing context length. Multiple studies (e.g., [34, 45]), both for LLMs and LTMs, have shown that increasing the context length significantly improves both in-sequence prediction and forecasting, and a recent study [46] find interesting scaling behaviour of LTMs with context length. We will explore context-length scaling in future work.

We have focused on univariate time-series data. However, a general purpose foundation model for time-series forecasting should be able to cope with the more general setting of multivariate time-series prediction, with multiple exogeneous covariates. Establishing scaling-laws for multivariate time-series forecasting will be an important extension to this work; this demands the assembly of a large and diverse training set of multivariate data, each with their own exogeneous factors.

References

[1] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, no. 6971, pp. 244–247, 2004.

[2] K. Doya, *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

[3] K. Doya, "Modulators of decision making," *Nature neuroscience*, vol. 11, no. 4, pp. 410–416, 2008.

[4] A. Funamizu, B. Kuhn, and K. Doya, "Neural substrate of dynamic bayesian inference in the cerebral cortex," *Nature neuroscience*, vol. 19, no. 12, pp. 1682–1689, 2016.

[5] C. Lindig-León, N. Kaur, and D. A. Braun, "From bayes-optimal to heuristic decision-making in a two-alternative forced choice task with an information-theoretic bounded rationality model," *Frontiers in Neuroscience*, vol. 16, p. 906198, 2022.

[6] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*. Springer Series in Statistics, Springer New York, 2013.

[7] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[8] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[12] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[15] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*, pp. 8821–8831, Pmlr, 2021.

[16] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[17] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.

[18] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[19] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, pp. 19730–19742, PMLR, 2023.

[20] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," *arXiv e-prints*, p. arXiv:2310.10688, Oct. 2023.

[21] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, "MOMENT: A Family of Open Time-series Foundation Models," *arXiv e-prints*, p. arXiv:2402.03885, Feb. 2024.

[22] K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. V. Hassen, A. Schneider, S. Garg, A. Drouin, N. Chapados, Y. Nevmyvaka, and I. Rish, "Lag-Llama: Towards Foundation Models for Time Series Forecasting," *arXiv e-prints*, p. arXiv:2310.08278, Oct. 2023.

[23] A. Garza and M. Mergenthaler-Canseco, "TimeGPT-1," *arXiv e-prints*, p. arXiv:2310.03589, Oct. 2023.

[24] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," *arXiv e-prints*, p. arXiv:2211.14730, Nov. 2022.

[25] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified Training of Universal Time Series Forecasting Transformers," *arXiv e-prints*, p. arXiv:2402.02592, Feb. 2024.

[26] G. Woo, C. Liu, A. Kumar, and D. Sahoo, "Pushing the Limits of Pre-training for Time Series Forecasting in the CloudOps Domain," *arXiv e-prints*, p. arXiv:2310.05063, Oct. 2023.

[27] W. Xue, T. Zhou, Q. Wen, J. Gao, B. Ding, and R. Jin, "CARD: Channel Aligned Robust Blend Transformer for Time Series Forecasting," *arXiv e-prints*, p. arXiv:2305.12095, May 2023.

[28] R. Ilbert, A. Odonnat, V. Feofanov, A. Virmaux, G. Paolo, T. Palpanas, and I. Redko, "Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention," *arXiv e-prints*, p. arXiv:2402.10198, Feb. 2024.

[29] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[30] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[31] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "Meta-learning framework with applications to zero-shot time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 9242–9250, 2021.

[32] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[33] Q. Ma, Z. Liu, Z. Zheng, Z. Huang, S. Zhu, Z. Yu, and J. T. Kwok, "A survey on time-series pre-trained models," *arXiv preprint arXiv:2305.10716*, 2023.

[34] A. Fatir Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. Sundar Rangapuram, S. Pineda Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, "Chronos: Learning the Language of Time Series," *arXiv e-prints*, p. arXiv:2403.07815, Mar. 2024.

[35] M. Kunz, S. Birr, M. Raslan, L. Ma, Z. Li, A. Gouttes, M. Koren, T. Naghibi, J. Stephan, M. Bulycheva, M. Grzeschik, A. Kekić, M. Narodovitch, K. Rasul, J. Sieber, and T. Januschowski, "Deep Learning based Forecasting: a case study from the online fashion industry," *arXiv e-prints*, p. arXiv:2305.14406, May 2023.

[36] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," *arXiv e-prints*, p. arXiv:2001.08361, Jan. 2020.

[37] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[38] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*, pp. 10096–10106, PMLR, 2021.

[39] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," *Advances in neural information processing systems*, vol. 34, pp. 12116–12128, 2021.

[40] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby, "Scaling vision with sparse mixture of experts," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8583–8595, 2021.

[41] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, *et al.*, "Scaling laws for autoregressive generative modeling," *arXiv preprint arXiv:2010.14701*, 2020.

[42] H. Hersbach, "Decomposition of the continuous ranked probability score for ensemble prediction systems," *Weather and Forecasting*, vol. 15, no. 5, pp. 559–570, 2000.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[44] M. B. Bjerregård, J. K. Møller, and H. Madsen, "An introduction to multivariate probabilistic forecast evaluation," *Energy and AI*, vol. 4, p. 100058, 2021.

[45] Gemini Team, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv e-prints*, p. arXiv:2403.05530, Mar. 2024.

[46] J. Shi, Q. Ma, H. Ma, and L. Li, "Scaling Law for Time Series Forecasting," *arXiv e-prints*, p. arXiv:2405.15124, May 2024.

[47] R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, and P. Montero-Manso, "Monash time series forecasting archive," in *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[48] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, and P. Battaglia, "Learning skillful medium-range global weather forecasting," *Science*, vol. 382, pp. 1416–1421, Dec. 2023.

[49] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar, "FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators," *arXiv e-prints*, p. arXiv:2202.11214, Feb. 2022.

[50] P. Emami, A. Sahu, and P. Graf, "Buildingsbench: A large-scale dataset of 900k buildings and benchmark for short-term load forecasting," *Advances in Neural Information Processing Systems*, 2023.

[51] E. J. H. Wilson, A. Parker, A. Fontanini, E. Present, J. L. Reyna, R. Adhikari, C. Bianchi, C. CaraDonna, M. Dahlhausen, J. Kim, A. LeBar, L. Liu, M. Praprost, L. Zhang, P. DeWitt, N. Merket, A. Speake, T. Hong, H. Li, N. M. Frick, Z. Wang, A. Blair, H. Horsey, D. Roberts, K. Trenbath, O. Adekanye, E. Bonnema, R. El Kontar, J. Gonzalez, S. Horowitz, D. Jones, R. T. Muehleisen, S. Platthotam, M. Reynolds, J. Robertson, K. Sayers, and Q. Li, "End-use load profiles for the u.s. building stock: Methodology and results of model calibration, validation, and uncertainty quantification," *Technical Report*, 3 2022.

[52] X. Liu, Y. Xia, Y. Liang, J. Hu, Y. Wang, L. Bai, C. Huang, Z. Liu, B. Hooi, and R. Zimmermann, "Largest: A benchmark dataset for large-scale traffic forecasting," in *Advances in Neural Information Processing Systems*, 2023.

[53] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv e-prints*, p. arXiv:1804.03209, Apr. 2018.

[54] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 368–380, 2019.

Table 1: **Dataset summary**. M indicates million and B indicates billion.

	Monash	Climate	Energy	Traffic	Finance	Audio	Total
Datasets	23	15	2	5	2	3	38
# of data points	503M	1.56B	2.5B	1.5B	42.6M	1.98B	8.13B
% of data	6.18%	19.19%	30.75%	18.45%	0.52%	24.35%	100%

A Dataset Details

In this section we detail the various sources that form the basis of our dataset and the choices made during its construction and re-balancing. Constructing a training set for establishing foundational scaling relations requires three key considerations. Firstly, the dataset should be large enough so that for the largest models trained, we are still operating in the $\sim$infinite data limit (see e.g., 36). Secondly, the dataset needs to be sufficiently diverse so that any results are representative of the foundation-model regime, covering a large volume of the space of time-series phenomenology. Thirdly, the dataset needs to be balanced, so that any scaling results are representative of foundation model behaviour and not tied to performance gains for a single or handful of dominant dataset(s).

Taking inspiration from large language models [36], we therefore aimed to gather around $\mathcal{O}(10^{10})$ data points from a variety of domains. We note that treating a single floating point number on a similar footing to a language token is not necessarily a good comparison; language tokens can contain significantly more semantic meaning than a floating point number can. The continual growth of open-source time-series datasets in both size and diversity will enable increasingly robust neural scaling studies.

Before detailing our particular sources, we would like to emphasize that there is a large corpus of time series data publicly available but is not currently formatted for easy downloading and processing. Ref. [25] was the first paper to open-source a large dataset[3], setting a trend for improved training and benchmarking of foundational time-series models. However, significant work is still required to expand available datasets in size and diversity to reach the same maturity as LLMs (large-scale, SOTA language models are trained on well over a trillion tokens).

We now discuss each dataset presented in Tab. 1.

All data used throughout this work has been labelled/licensed as free to use for non-commercial purposes with the appropriate citations. We have included the appropriate citations where necessary below.

A.1 Monash

The Monash dataset has been the default repository of open-source time series data used by the academic community for some time [47]. It contains data from a huge variety of sources and contains a wide variety of characteristics. For this work we exclude series that are either too short or are particularly noisy.[4] We are then left with a total of 23 different sources which add up to a total of $\sim 500\text{M}$ data points; details are given in Tab. 2.

A.2 Climate

Our climate dataset, made up of around 1.5B data points, has two primary sources: the National Oceanic and Atmospheric Administration (NOAA) and the fifth generation European Centre for Medium-Range Weather Forecasts atmospheric reanalysis of the global climate (ERA5). Each source provides approximately 750M data points split across a variety of observables and time frames.

We note here that since the global climate is a correlated system, forecasting a single variable into the future whilst ignoring the evolution of the rest of the system is intrinsically difficult (maybe

[3]Note that by the time this data became open-source we had already fixed our dataset for the production runs completed for this study.

[4]We found through experimentation that removing very noisy datasets significantly improved training stability.

Table 2: **Monash Data:** For each dataset we list the sampling frequency, the total number of series, and the total number of data points.

Dataset	Frequency	Number of Series	Number of Data Points
London Smart Meters	Half Hourly	5,560	166.5M
Wind Farms	Every Minute	339	172.1M
Wind Power	4 Seconds Intervals	1	7.4M
Solar Power	4 Second Intervals	1	7.4M
Oikolab Weather	Hourly	8	0.8M
Elecdemand	Half Hourly	1	17.5k
Kaggle Web Traffic	Daily	145,063	116.5M
Tourism Quarterly	Quarterly	427	42.5k
Tourism Monthly	Monthly	366	109.3k
CIF 2016	Monthly	72	7.1k
Traffic Weekly	Weekly	862	89.6k
Traffic Hourly	Hourly	862	15.1M
Australian Electricity	Half Hourly	5	1.2M
Sunspot	Daily	1	73.9k
Hospital	Monthly	767	64.4k
NN5 Daily	Daily	111	87.8k
NN5 Weekly	Weekly	111	12.5k
M4 Hourly	Hourly	414	373.4k
Fred MD	Monthly	107	77.9k
Solar Weekly	Weekly	137	7.1k
Solar 10 Minutes	10 Minute Intervals	137	7.2M
Electricity Weekly	Weekly	321	50.1k
Electricity Hourly	Hourly	321	8.4M

impossible in some cases). Nevertheless, each time series can provide important information from which the foundation model can learn correlations. Moreover, some seasonal trends are very stable and predictable from a single time series. Future work should carefully consider how to include climate data in a way that allows the model to exploit correlations inherent in the data [48, 49].

Table 3: **NOAA Data:** For each dataset we list the sampling frequency, the total number of series, the length of each series, and the total number of data points.

Dataset	Frequency	Number of Series	Length	Number of Data Points
SST Mean	Daily	582241	365	212.5M
SST Anomalies	Daily	581249	365	212.1M
SST Long Term Average	Daily	218211	365	79.6M
SST Monthly Average	Monthly	72730	509	37M
SST Weekly Average	Monthly	72689	2214	161M
Ice Mean	Daily	63971	365	23M
Ice Long Term Average	Daily	12451	365	4.5M
Ice Monthly Average	Daily	5363	509	2.7M
Radiation Long Term Average	Daily	6622	365	2.4M

NOAA: We primarily gather data from NOAA high-resolution blended analysis of daily sea surface temperature (SST) which includes both temperature measurements and ice level measurements on a 0.25° grid worldwide.[5] Weather at different points of the grid are intrinsically correlated, especially on such small grid sizes. We therefore downsample the data by a factor of three by randomly choosing grid points without replacement (we do this independently for each dataset).

To ensure we have data that covers a wide range of time scales and variability we pick a variety of observables shown in Tab. 3. For the daily data we pick 8 years of data, each separated by 5 years (spread out to maximize data diversity i.e., minimize year to year correlations) but skip leap years for easier data processing (so all arrays are 365 elements long). The final year selection is 1985, 1990, 1995, 2001, 2005, 2010, 2015, and 2021. This size of this dataset could easily be supplemented simply by adding more of the 40 years of available data.

For additional diversity we use the same method to extract outgoing long wave radiation time series from `https://downloads.psl.noaa.gov/Datasets/uninterp_OLR/`. This is the shown in the final row of Tab. 3.

Table 4: **ERA5 Data:** Similar to above. The different number of series for each dataset is due to the randomness in the subsampling.

Dataset	Frequency	Number of Series	Length	Number of Data Points
Sea Level Pressure	4 Hour Intervals	63094	2190	138M
2m Temp.	4 Hour Intervals	63190	2190	138M
2m Dewpoint Temp.	4 Hour Intervals	63123	2190	138M
Surface Pressure	4 Hour Intervals	63263	2190	139M
10m V Wind Comp.	4 Hour Intervals	63263	2190	139M
10m U Wind Comp.	4 Hour Intervals	63220	2190	138M

ERA5: We take a similar approach to above when processing/gathering ERA5 data. Here though, we focus on higher frequencies by using a single year of data (2001) sampled every four hours. We additionally use different data variables (the six most popular variables) to ensure that the data features are likely different to those present in the NOAA data. ERA5 data is also originally on a 0.25° global grid which we randomly down sample by a factor of four. Details are given in Tab. 4.

A.3 Energy

For the energy dataset, we use the benchmark dataset prepared in the `BuildingsBench` data release [50]. In particular, we choose to sample 2.5B data points from the full dataset (which totals over 15B individual data points). These 2.5B data points, which overall constitute approximately 30% of our full dataset, are all taken from the Buildings-900K database. These time series represent a large-scale sample of simulated US building energy demand and are designed to be broadly representative of US commercial and residential building stock. As described in [50], the dataset is originally sourced from the NREL EULP database [51], which provides 15- minute resolution, appliance-level consumption for 550K residential and 350K commercial buildings spread across all climate regions in the U.S. For more finer-grained details, see App. B.3 in Ref. [50].

A.4 Traffic

We consider the public LargeST [52] dataset which is a collection of 8600 time series recorded from traffic sensors in the California area. The data spans over 5 years, from 2017 to 2021, and is sampled at 15 minute resolution. To reduce the data size, we down-sample the data to hourly resolution and remove series that contains over 50% missing entries. This gives us a total of 8520 series all with length 175296, which translates to 1.46B data points.

[5]The original data can be found here `https://downloads.psl.noaa.gov/Datasets/noaa.oisst.v2.highres/`.

A.5 Finance

We include daily stock returns and volume data, treated as separate one-dimensional time-series respectively, for 5038 stocks listed across the Nasdaq, NYSE, and AMEX stock exchanges. Daily stock returns and volume tickers are obtained for 7230 stocks from `yahoo finance`, from the beginning of each listing up to 1st January 2024. We discard any stocks that have fewer than 512 ticks (recorded trading days), and any series containing `NaN` or `inf`. This results in time-series for 5038 stocks, with both returns and volume data, and a total of 42.6M data points (Tab. 5).

Table 5: **Finance Data:** Daily stock returns and volume data for 5038 stocks listed across the Nasdaq, NYSE and AMEX exchanges, obtained from `yahoo finance`.

Dataset	Frequency	Number of Series	Number of Data Points
Stock Returns	Daily	5038	26.3M
Stock Volume	Daily	5038	26.3M

A.6 Audio

Audio data is intrinsically a one dimensional time series rich with structure and features; it is therefore perfectly suited for our study. We have three primary sources of audio data, all from the DagsHub Open-Source Audio Datasets repository (`https://github.com/DagsHub/audio-datasets`). Again, the total volume of data here is extremely large and can be used to supplement future datasets for larger models. Here we use three particular sources each from a different domain to enhance its diversity. As presented in Tab 1, these three sources add up to approximately 2B data points and $\sim 25\%$ of our overall dataset. A summary of the three sources can be found in Tab. 6

Table 6: **Audio Data:** Similar to above.

Dataset	Frequency	Number of Series	Length	Number of Data Points
Commands	16 kHz	47650	16,000	762.4M
Arabic Speech	24 kHz	1813	Varied	329.9M
Bird Audio	22 kHz	4000	Varied	888.3M

Commands: The speech command dataset [53] is made up of a series of short audio files with different voices saying a collection of common English words (e.g., "happy" and "five"). From all the data provided `https://github.com/DagsHub/audio-datasets/blob/main/Speech_Commands_Dataset/README.md` we take a random half of the data and exclude any clips that are not 16k long (again for easy saving). We are then left with 47650 series, making a total of ~ 750M data points.

Arabic Speech: This dataset contains 1813 time series of high quality (studio recorded) spoken Arabic utterances sampled at 48kHz – `https://github.com/DagsHub/audio-datasets/tree/main/Arabic-Speech-Corpus`. To reduce the data size without dramatically affecting its quality, we down sample the data by a factor of two (human speech is typically below 24kHz). This gives us a total of ~ 300M data points.

Birds: Finally, we use the bird detection dataset from `https://github.com/DagsHub/audio-datasets/blob/main/Bird-Audio-Detection-challenge/README.md` [54]. This dataset contains a combination of bird and other sounds designed to train machine learning algorithms to detect bird noises. Here we ignore the labels and use the entire dataset in training. Again, to reduce data volumes we down sample by a factor of two, and only use a randomly chosen half of the data. This leaves us with 4000 time series sampled at 22 kHz for a total of ~ 900M data points.

A.7 Dataset balancing and pre-processing

Each dataset is made up of a large number of individual time series of varying lengths. We use 95% of the set of time series for training and the remaining 5% for testing. Since the majority of the series are significantly longer than our context window, during training and testing we visit each series with probability $p_i = t_i/T$, where t_i is the number of data points in that series and T is the total number of data points in the training set. Additionally, each time we visit a series we choose a random starting index. This strategy ensures that the model sees each section of the data once (on average) in a given epoch. We normalize each time-series in the training set to have zero mean and unit standard deviation.[6]

B Training details and compute requirements

We use the `AdamW` optimizer with a batch size of 512, a cosine learning rate scheduler with a linear warm up of 3000 training steps, and train for a total of 10^5 steps. When training on the entire dataset ($\sim$ 8B data points), this equates to roughly two epochs. To reduce computational costs we compute the test loss every $\mathcal{O}(200)$ steps and average over a random 10% of the test data each time.

To produce the results in this paper requires $\mathcal{O}(50-70)$ individual production runs. Apart from the 100M parameter run, these were all carried out on single A100 NVIDIA GPU instances, each taking between 1 and 3 days to complete. As such, overall, the work presented here required $\mathcal{O}(150)$ GPU-days of compute. To host the full dataset, we also required a CPU RAM allocation of approximately 250 GB.

C Learning-rate and architecture dependence

In Fig. 3 we show the effect of changing the maximum learning rate reached at the end of the warm up. The performance of the model (CRPS) clearly depends on the maximum learning rate, and that dependence is itself a function of parameter count. The dependence on maxmimum learning rate is strong enough that it is possible to get better performance with a smaller model, if the maximum learning rate is too small (or too large) for the larger model. Moreover, for a fixed model size we see a clear optimum learning rate above which models diverge (shown as crosses on Fig. 3). To ensure that we used an optimal maximum learning rate as a function of model size, we fit a power law with a constant offset to the best models (at each parameter size) shown in Fig. 3. In the few cases for the largest models where our power law fit overestimates the optimal maximum learning rate (leading to divergence), we slowly reduce the learning rate until we achieve convergence.

In Figure 4 we show the dependence of model performance on architecture choices, showing that performance is largely invariant to the number of heads, while being only weakly sensitive to the aspect ratio (for aspect ratios $\lesssim 100$).

D Power-law fits

In 7 we provide the power law fits to teh scaling relations shown in Fig. 1.

Table 7: **Power-law fits.**

	MSE		CRPS		Log-Likelihood	
	$\log_{10}(A_0)$	B_0	$\log_{10}(A_0)$	B_0	$\log_{10}(A_0)$	B_0
Number of Parameters, N_p	-19.47	0.042	-22.64	0.036	4.33	0.151
Training Compute, $\mathcal{C}$	-38.88	0.031	-43.03	0.028	-6.65	0.101
Dataset Size, $\mathcal{D}$	-8.91	0.062	-30.42	0.027	7.00	0.188

[6] In rare instances where input time-series are constant (and hence have zero standard deviation), we set them to a constant value of zero.

E In-Sequence Predictions to Forecasting

Here we simply show an example of how in-sequence test loss correlates with forecasting prediction from roll-out. In particular, in Fig. 2 we show forecasts for three different datasets as a function of model size. Here we use the best weights (i.e., the model that achieved the lowest test loss during training) for each model size and show both in-sequence and forecasting along with the true data. For both the in-seqeunce and forecasting predictions we show the 1σ range of predictions. Although not perfect, it's clear that as one scales up model size (and therefore in-sequence test loss decreases), forecasting performance also improves substantially. Although we only show three examples here, we observe a similar trend in the forecasting power of our models for a variety of datasets. We leave a more detailed exploration to future work.

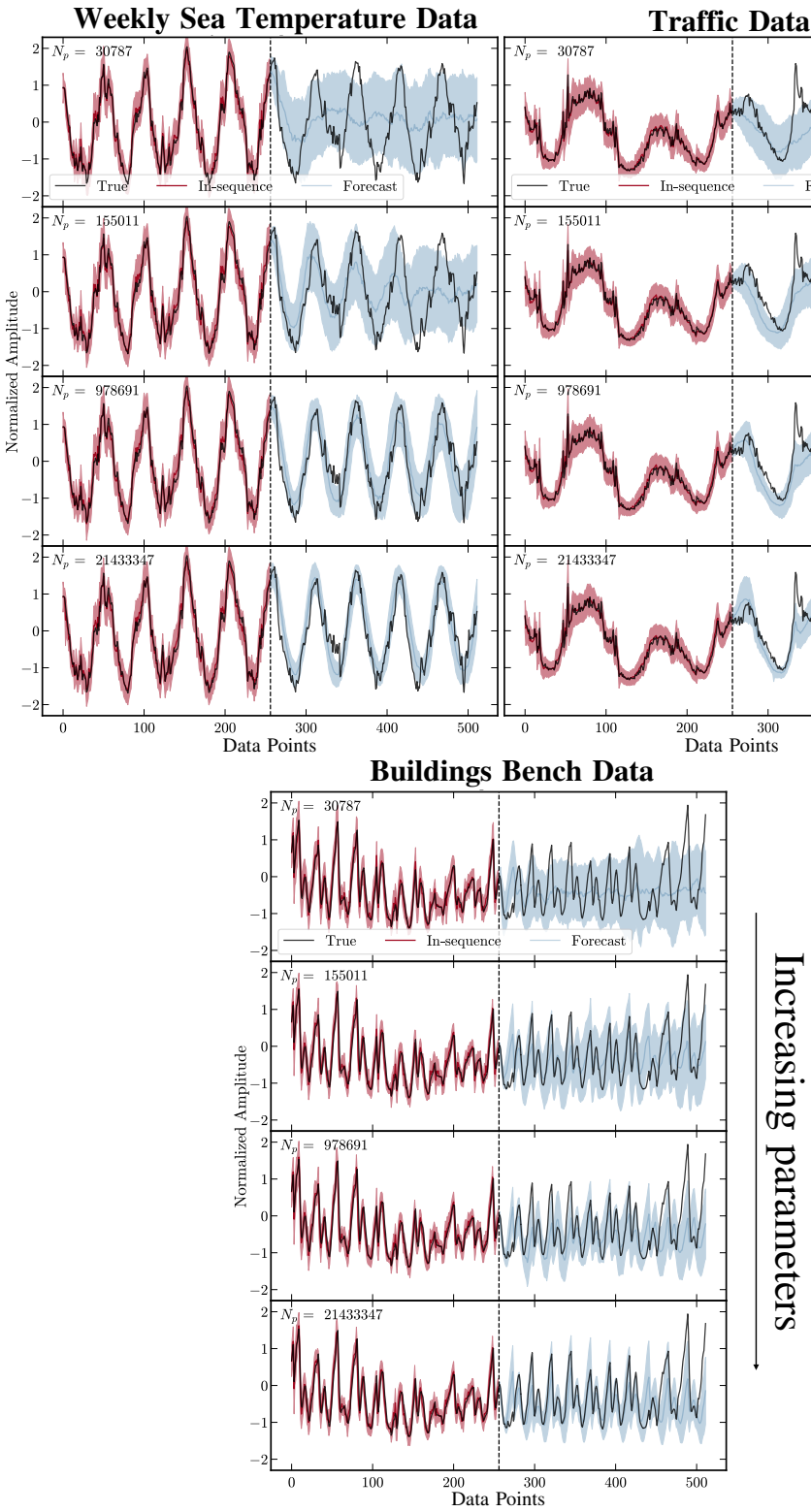

Figure 2: **In-sequence test loss to forecasting:** Here we show the connection between improved in-sequence test loss and forecasting performance as a function of model size. In particular, we show the true data in black with 1σ ranges for both in-sequence and forecasting predictions. It is clear that as in-sequence test loss decreases, forecasting also becomes substantially more predictive.

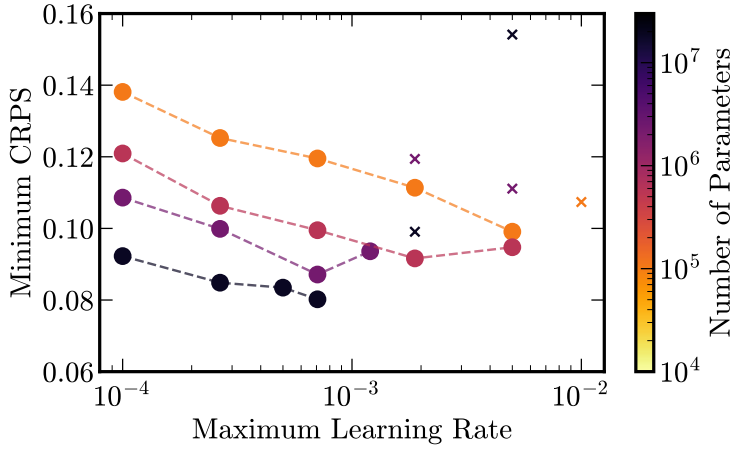

Figure 3: **Importance of Learning Rate:** Here we show the minimum CRPS measured on the test data as a function of the maximum learning rate reached at the end of the linear warm up schedule. Crosses indicate that the model diverged before training was complete. There is a clear optimum max learning rate which decreases as a function of model size/number of parameters.

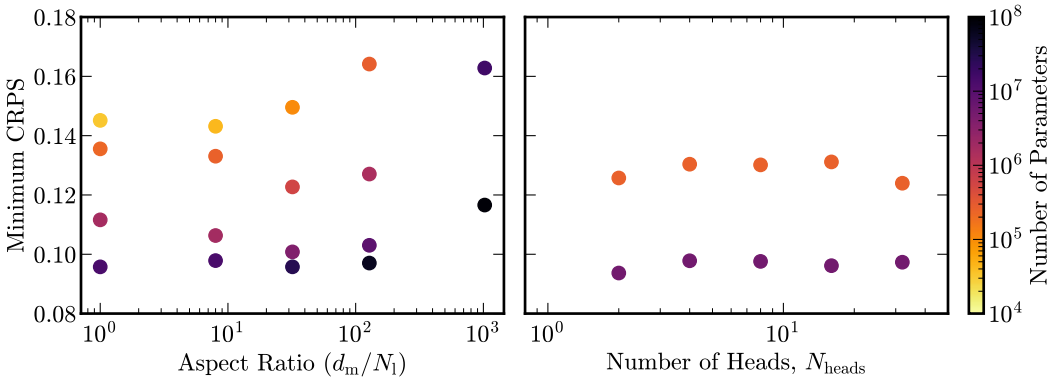

Figure 4: **Importance of Transformer Architecture:** We show the minimum CRPS on the test set as a function of architecture choices and number of parameters. *Left:* Performance on the test data has a weak dependence on aspect ratio below < 100 but degrades significantly > 128. We therefore keep aspect ratios < 70 for all scaling runs. *Right:* Here we see that the number of attention heads has no noticeable affect on the performance for both model sizes tested. We fix the number of heads to four for the scaling runs.