Forecasting Emerges from Auto-Regressive Pretraining: Latent Predictive Structure in Language Models

Alexis Roger [1 2 *] **Prateek Humane** [3 2 *] **Zhenghan Tai** [4 *]
Gwen Legate [5 2] **Andrei Mircea** [3 2] **Vasilii Feofanov** [6] **Irina Rish** [3 2]

Abstract

Predicting how a sequence will continue is a basic problem for intelligent systems. We show that large language models contain usable forecasting structure *before* any explicit time-series supervision. A single linear readout from frozen Qwen3-0.6B hidden states maps ordinary text sequences to numerical trajectories that resemble real time series, and those trajectories can be used for straightforward forecasts. The distribution over output tokens also gives coherent, non-crossing probabilistic forecasts in a single forward pass. After time-series specialization, pretrained models show aligned gradients and improve immediately, whereas randomly initialized models spend early training in a destructive-interference regime. These findings suggest that auto-regressive pretraining already shapes representations around temporal continuation; and finetuning adapts that structure to numerical forecasting rather than creating it from scratch.

1. Introduction

Forecasting is a basic ingredient of intelligence. Agents need to predict continuations, reason about uncertain futures, and act on expectations about what comes next. The same problem appears in time-series forecasting, such as weather and climate models, planning, and probabilistic reasoning under uncertainty.

Recent work has shown that large auto-regressive foundation models, trained only to predict the next token of text, can forecast in domains they were never trained on (Gruver et al., 2023; Jin et al., 2023; Zhou et al., 2023; Wolff et al., 2025). This transfer is empirically real but mechanistically unclear: some studies attribute it to architectural biases or tokenization (Tan et al., 2024; Zheng et al., 2025; Zhang et al., 2025), while others find substantial gains in low-data, cross-domain, and distribution-shift settings (Riachi et al., 2025; Qiu et al., 2026; Bayazi et al., 2024). We ask a more basic question:

What makes a model capable of forecasting?

We test the hypothesis that large-scale auto-regressive pretraining organizes representations into predictive temporal structure that can be rapidly specialized into a forecasting system. Language sequences contain repetition, trends, periodicity, and long-range dependencies. Predicting the next token therefore forces a model to encode information about continuation. The resulting hidden states, even before time-series exposure, can already trace trajectories that resemble real numerical signals.

Contributions. We give four pieces of evidence:

1. A linear projection from frozen Qwen3-0.6B hidden states decodes *realistic time-series trajectories* from English text, without paired supervision (Sec. 3).
2. These projected trajectories *forecast future values via retrieval*, beating a last-value baseline before any time-series finetuning (Sec. 4).
3. Auto-regressive token distributions already produce coherent, non-crossing *probabilistic forecasts* in a single forward pass (Sec. 5).
4. When forecasting supervision is added, pretrained models exhibit *aligned gradients and rapid convergence* from step one, indicating that finetuning specializes pre-existing predictive structure rather than building it (Sec. 6).

These results recast cross-modal forecasting transfer as a consequence of the sequential-prediction objective that defines language pretraining.

[1]McGill University [2]Mila - Quebec AI Institute [3]Université de Montréal [4]University of Toronto [5]Concordia University [6]42.com. Correspondence to: <alexis.roger@mila.quebec>.

Proceedings of the ICML Workshop on Forecasting as a New Frontier of Intelligence, Seoul, South Korea. 2026.

2. Forecasting Setup

We repurpose a Qwen3-0.6B auto-regressive language model (Yang et al., 2025) as a probabilistic time-series forecaster by casting forecasting as next-token prediction over discretized values. This minimal modification lets us study both the pretrained model's latent forecasting capability and the dynamics of later specialization.

Discretization. Given a univariate series, we extract sliding windows of length $C + L$ with C=512 and prediction horizon L=64. Each window is normalized using context-window statistics and quantized into V=1024 uniform bins over $[-5, 5]$. Following Roger et al. (2025), bin indices are mapped directly to the first V vocabulary slots of the pretrained model so that the auto-regressive distribution over the LLM's output tokens is, by construction, a distribution over future numerical bins.

Probabilistic objective. When finetuning, the model is trained with a weighted quantile loss over $\mathcal{Q} = \{0.1, \ldots, 0.9\}$ following Chronos Bolt (Ansari et al., 2024):

$$\mathcal{L} = \frac{1}{T|\mathcal{Q}|} \sum_{t=1}^{T} \sum_{\tau \in \mathcal{Q}} \rho_\tau(y_t - \hat{q}_{t,\tau}), \tag{1}$$

where $\rho_\tau(u) = u(\tau - \mathbf{1}[u < 0])$ is the standard quantile regression check function. Output logits are softmaxed into a categorical distribution over ordered bins and inverted to a CDF; arbitrary quantiles are then extracted in a single pass.

Adaptation regimes. We compare four regimes that differ only in trainable parameters: *full finetuning*, *IO only* (only input embeddings and output head trained), *LoRA attention* (Hu et al., 2021) (r=8), and *LoRA Attention+IO*. Models are trained on sliding windows from GiftEval (Aksu et al., 2024) with AdamW, batch size 128, bf16. Evaluation uses 1,000 held-out windows under CRPS, MASE, and MSE, following GluonTS conventions (Alexandrov et al., 2020). Full hyperparameters, metric definitions, and additional results are in Appendix A.

3. Latent Forecasting Structure in Pretrained LLMs

We first ask whether forecasting-compatible structure exists before any time-series training. Concretely: can a simple linear map, trained *without any paired text–time-series supervision*, project frozen LLM hidden states onto realistic time-series windows?

Method. We pass N=1,920 WikiText-103 sequences (Merity et al., 2016) of T=512 tokens through frozen Qwen3-0.6B. For each token t in sequence i we form

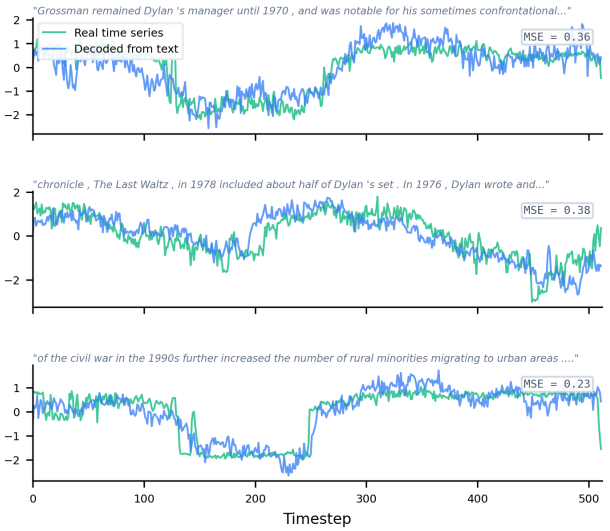

Figure 1. **Realistic time series decoded from frozen LLM text representations.** Three decoded outputs (blue) overlaid with their nearest real GiftEval window (green); input text shown above each plot (gray). No paired supervision was used. MSE $\in [0.23, 0.38]$.

$\mathbf{h}_{i,t} \in \mathbb{R}^D$ by concatenating the hidden states of all 28 layers ($D = 28 \times 1024 = 28{,}672$). A single linear layer

$$\hat{y}_{i,t} = \mathbf{w}^\top \mathbf{h}_{i,t} + b \tag{2}$$

produces a scalar at each timestep, yielding a predicted trajectory $\hat{\mathbf{y}}_i \in \mathbb{R}^T$ that we z-score normalize.

We train $\mathbf{w}$ with an EM-style nearest-neighbor matching procedure against a bank of M=10,000 real GiftEval windows. At each step we sample K=128 candidates per prediction, choose $j_i^\star = \operatorname{argmin}_j \frac{1}{T}\|\hat{\mathbf{y}}_i - \mathbf{Y}_j\|^2$, and minimize an MSE-to-nearest term plus a PSD diversity penalty:

$$\mathcal{L} = \tfrac{1}{B}\textstyle\sum_i \tfrac{1}{T}\|\hat{\mathbf{y}}_i - \mathbf{Y}_{j_i^\star}\|^2 + \tfrac{\lambda}{|\mathcal{P}|}\textstyle\sum_{(i,j)\in\mathcal{P}} \cos\big(\mathbf{s}_i, \mathbf{s}_j\big), \tag{3}$$

where $\mathbf{s}_i = |\mathrm{FFT}(\hat{\mathbf{y}}_i)|^2$, $\lambda = 0.5$; trained with Adam ($\eta = 10^{-3}$, B=32, 100 epochs).

Result. Across the 1,920 training predictions, nearest-neighbor matches span 686 distinct real time series; 36% of predictions map to a *unique* real signal rather than collapsing onto a single mode (Fig. 1). On 1,000 held-out WikiText sequences never seen during training, mean nearest-neighbor MSE is 0.72 (vs. 0.67 on training text), confirming the structure is a property of the pretrained representation space rather than an artifact of fit.

What creates this structure? A 2×2 ablation crossing model weights (pretrained vs. random init, same architecture) with input (WikiText vs. random tokens) gives a sharp

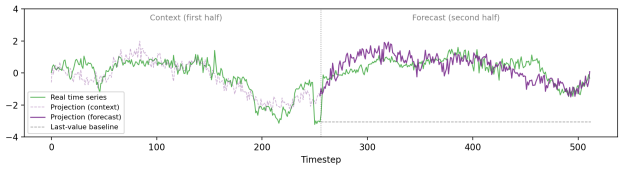

Figure 2. **Forecasting future values by retrieving from a bank of pretrained text projections.** The first 256 steps (left of the dashed line) are used to retrieve the closest WikiText projection; its second half (purple) forecasts the continuation, capturing the level shift and trend that last-value carry-forward (gray dashed) misses (retrieval MSE 0.42 vs. LV 11.7).

contrast: random initialization drops unique coverage to 2.8% (54 matches), and random tokens through pretrained weights collapse it to 0.2% (4 matches). Only the combination of *pretrained weights and meaningful text* produces diverse temporal trajectories (Appendix B). Auto-regressive language pretraining is therefore what produces this structure.

4. Retrieval-Based Forecasting from Pretrained Representations

The decoding result shows that pretrained hidden states can trace realistic trajectories. We next ask whether those trajectories carry *predictive* signal: given the first half of a real time series, can a retrieved text projection forecast its continuation?

Method. We take 500 real GiftEval series and observe only the first 256 of 512 timesteps. Among the 10,000 WikiText projections obtained in Sec. 3, we retrieve the one with lowest MSE on the observed first half and use its second half as the forecast for the remaining 256 steps. The baseline is Last-Value carry-forward (LV).

Result. Across all 500 queries, the retrieval-based forecast achieves MSE 1.91 versus last-value's 2.27—a 16% reduction with *no time-series training and no model parameters touched*. The mechanism behind this aggregate gap is asymmetric: retrieval wins on 37% of queries, but when it wins the median advantage has 4 times lower MSE because the retrieved projection captures trends and level shifts that last-value misses. Figure 2 shows a representative case.

Interpretation. This suggests that pretrained language models encode more than shape-level similarity to time series: they encode *continuation structure*. The same auto-regressive computation that predicts the next text token traces hidden-state trajectories whose continuations are informative about how unrelated numerical signals will continue. Formally, the residual stream defines a deterministic update $\mathbf{h}_{t+1} = F(\mathbf{h}_t, x_{t+1})$ whose trajectories $\{\mathbf{w}^\top \mathbf{h}_t\}$ recover realistic future-prediction signal under a linear readout. Forecasting capability, by this measure, predates time-series supervision.

5. Probabilistic Forecasting from Token Distributions

A second piece of pre-existing forecasting structure appears in the uncertainty estimates. After modest finetuning, the auto-regressive distribution over the LLM's output tokens defines a categorical distribution over ordered numerical bins. Cumulative summation yields a discrete CDF

$$\hat{F}(b \mid \mathbf{x}_{<t}) = \sum_{k \le b} p_\theta(k \mid \mathbf{x}_{<t}), \tag{4}$$

and arbitrary quantiles are extracted by inversion: $\hat{q}_\tau = \inf\{b : \hat{F}(b \mid \mathbf{x}_{<t}) \ge \tau\}$. Three properties follow directly:

- **Coherent, non-crossing quantiles by construction.** As all quantiles are read from a single CDF, $\hat{q}_{\tau_1} \le \hat{q}_{\tau_2}$ whenever $\tau_1 \le \tau_2$.
- **Any quantile from a single forward pass.** Unlike fixed-grid quantile regression or quantile-conditioned models, the cost of producing hundreds of quantiles is nearly constant after the logits are computed and can be changed at inference time.
- **No interpolation between sparse quantile heads.** The full predictive distribution is the model's native output.

This is not an auxiliary forecasting head added to the LLM. It is the auto-regressive distribution already produced by the model, reinterpreted as a forecasting distribution because the token inventory has been ordered along a numerical axis. Probabilistic forecasting, like trajectory structure, is present in rough form before specialization; supervision calibrates it for the numerical task.

6. Fast Specialization Through Finetuning

If forecasting structure already exists in the pretrained model, then finetuning should behave like *specialization*: a rapid adjustment of an already useful representation rather than construction from scratch. We test this in two ways.

Coherent gradients from step one. Mircea et al. (2025) attribute slow progress in language models to *zero-sum learning*: per-example gradients disagree, so reducing loss on one example raises it on another. We compute per-sample gradients $\mathbf{g}_i = \nabla_\theta \mathcal{L}(x_i; \theta)$ over 32 held-out time series at each checkpoint and measure the mean off-diagonal cosine alignment $\frac{1}{N(N-1)} \sum_{i \ne j} \cos(\mathbf{g}_i, \mathbf{g}_j)$. Figure 3 shows that pretrained models exhibit high gradient alignment from step one and CRPS descends immediately, whereas randomly initialized models sit near zero alignment for tens to hundreds of steps before loss begins to fall. The pattern holds in

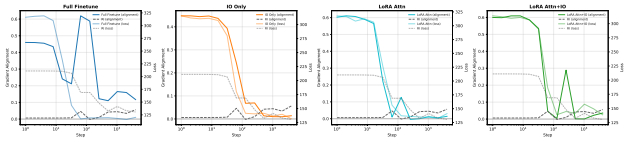

Figure 3. **Coherent gradients from step one.** Per-sample gradient alignment (left axis; mean off-diagonal cosine over 32 held-out series) and CRPS evaluation loss (right axis) versus training step (log scale). Solid: language-pretrained (LangInit). Dashed: random initialization (RandInit). Across all four regimes, pretrained models exhibit high alignment immediately and loss falls from step one, while random initialization remains near-zero alignment until step 64–256, at which point loss begins to decline.

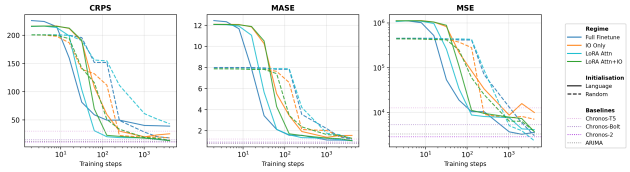

Figure 4. **Rapid specialization across four adaptation regimes.** Forecasting metrics versus training step at h=1. Solid lines: language-pretrained initialization; dashed lines: random initialization; dotted horizontals: baseline performance. Language-pretrained models converge faster and to lower error in every regime; LoRA-based methods match full finetuning.

all four regimes (including IO-only, where the transformer backbone is frozen) indicating that the coherent learning signal lives in the pretrained representation, not in any particular adapter.

Rapid convergence and low-rank specialization. Figure 4 compares the four adaptation regimes under both initializations. Pretrained models converge faster and reach lower forecasting error in every regime; the gap is largest in the first 10–100 steps. LoRA-based methods (which modify only a small low-rank subspace) match full finetuning, indicating that specialization requires only a low-dimensional adjustment of the pretrained representation rather than a wholesale rewrite. The effective-data analysis (Hernandez et al., 2021) formalizes this: full finetuning saves $D_T \approx 6.8\times$ in training steps, attention-only LoRA $5.2\times$, and LoRA+IO $3.4\times$ relative to random initialization (Appendix C).

The two panels show the same pattern from different angles: useful learning signal is available at step one, and aligning it with the time-series objective requires movement in only a small subspace.

7. Discussion

Forecasting as a consequence of auto-regressive pretraining. Our results support a single claim: language pretraining builds a system whose representations are organized for sequential prediction, and forecasting is one behavior this system can express without being explicitly taught. The same hidden states that predict the next text token (i) trace realistic numerical trajectories under a linear readout (Sec. 3); (ii) carry continuation signal usable for retrieval-based forecasting before any time-series training (Sec. 4); (iii) define a native probabilistic forecast over ordered bins (Sec. 5); and (iv) provide an aligned-gradient signal that lets adaptation succeed immediately (Sec. 6). Finetuning specializes this pre-existing predictive structure; it does not invent it.

Connection to forecasting-capable foundation models. This account reconciles disagreements about when language pretraining helps forecasting. If the asset being transferred is reusable *future-prediction structure*, the advantage should be largest where limited data cannot establish that structure on its own—low-data, cross-domain, and distribution-shift settings—matching the empirical picture (Riachi et al., 2025; Qiu et al., 2026). It also makes low-rank adaptation a principled choice: if specialization mainly redirects an already prepared system, the required update should be small and structured rather than a full rewrite.

Predictive world models. The mechanism we study is not specific to language. Any sufficiently trained auto-regressive sequence model whose representations carry temporal structure should support similar forecasting transfer; language is a particularly rich source of such structure, not the only one. This is consistent with reports that vision-pretrained backbones (Chen et al., 2024; Roschmann et al., 2025) and embedding geometries (Huh et al., 2024; Jha et al., 2025) share predictive structure across modalities. In this framing, forecasting capability is less tied to a particular domain than to large-scale next-step prediction.

Limitations. All experiments use a single backbone (Qwen3-0.6B), one pretraining corpus, and one tokenization scheme; this interpretation is consistent with our results but has not been verified at larger scale. The retrieval forecast beats last-value on 37% of queries, with the aggregate gap driven by large wins where projections capture trends and level shifts. Future work should characterize which series the latent structure already supports and which still require explicit supervision. Finally, "forecasting emerges from auto-regressive pretraining" describes a dominant mechanism we observe; fully separating contributions of pretraining data, architecture, and scale will require interventions beyond this paper's scope.

Conclusion. Large-scale auto-regressive pretraining appears to provide much of what forecasting requires. Pretrained language models already encode realistic temporal trajectories, support retrieval-based future prediction, and define coherent probabilistic forecasts, all before explicit forecasting supervision. Finetuning turns this latent predictive structure into a forecasting system through a low-dimensional, low-cost adjustment. On this view, forecasting is not constructed only at downstream training time; it is

partly inherited from the act of training a model to predict the next step.

8. Acknowledgment

We thank 42.com for providing computational resources that supported this work. We also acknowledge the AMD University Program AI & HPC Cluster for additional compute resources used in this research. This research was also made possible thanks to the computing resources on the Frontier supercomputer, provided as a part of the ALCC 2025 program award "Real-Time Adaptive Disruption Forecasting at DIII-D". These resources were provided by the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

Aksu, T., Woo, G., Liu, J., Liu, X., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Gift-eval: A benchmark for general time series forecasting model evaluation, 2024. URL https://arxiv.org/abs/2410.10393.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116): 1–6, 2020.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., and Wang, Y. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Bayazi, M. J. D., Ghonia, H., Riachi, R., Aristimunha, B., Khorasani, A., Arefin, M. R., Darabi, A., Dumas, G., and Rish, I. General-purpose brain foundation models for time-series neuroimaging data. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

Chen, M., Shen, L., Li, Z., Wang, X. J., Sun, J., and Liu, C. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters, 2024. URL https://arxiv.org/abs/2408.17253.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters, 2023. URL https://arxiv.org/abs/2310.07820.

Hernandez, D., Kaplan, J., Henighan, T., and McCandlish, S. Scaling laws for transfer, 2021. URL https://arxiv.org/abs/2102.01293.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Huh, M., Cheung, B., Wang, T., and Isola, P. The platonic representation hypothesis, 2024. URL https://arxiv.org/abs/2405.07987.

Jha, R., Zhang, C., Shmatikov, V., and Morris, J. X. Harnessing the universal geometry of embeddings, 2025. URL https://arxiv.org/abs/2505.12540.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., and Wen, Q. Time-llm: Time series forecasting by reprogramming large language models, 2023. URL https://arxiv.org/abs/2310.01728.

Li, M. Z., Agrawal, K. K., Ghosh, A., Teru, K. K., Santoro, A., Lajoie, G., and Richards, B. A. Tracing the representation geometry of language models from pretraining to post-training, 2025. URL https://arxiv.org/abs/2509.23024.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models, 2016.

Mircea, A., Chakraborty, S., Chitsazan, N., Rish, I., and Lobacheva, E. Training dynamics underlying language model scaling laws: Loss deceleration and zero-sum learning. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28154–28188, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1366. URL https://aclanthology.org/2025.acl-long.1366/.

Qiu, X., Tong, J., Sun, Y., Ma, Y., Zhang, W., and Shen, X. Rethinking the role of llms in time series forecasting, 2026. URL https://arxiv.org/abs/2602.14744.

Riachi, R., Rasul, K., Ashok, A., Humane, P., Roger, A., Williams, A. R., Nevmyvaka, Y., and Rish, I. Random initialization can't catch up: The advantage of language model transfer for time series forecasting, 2025. URL https://arxiv.org/abs/2506.21570.

Roger, A., Legate, G., Rasul, K., Nevmyvaka, Y., and Rish, I. Small vocabularies, big gains: Pretraining and tokenization in time series models, 2025. URL https://arxiv.org/abs/2511.11622.

Roschmann, S., Bouniot, Q., Feofanov, V., Redko, I., and Akata, Z. Time series representations for classification lie hidden in pretrained vision transformers. *arXiv preprint arXiv:2506.08641*, 2025.

Tan, M., Merrill, M. A., Gupta, V., Althoff, T., and Hartvigsen, T. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, pp. 60162–60191, 2024.

Wolff, M. L., Yang, S., Torkkola, K., and Mahoney, M. W. Using pre-trained llms for multivariate time series forecasting, 2025. URL https://arxiv.org/abs/2501.06386.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhang, X., Feng, S., and Li, X. From text to time? rethinking the effectiveness of the large language model for time series forecasting. *arXiv preprint arXiv:2504.08818*, 2025.

Zheng, L. N., Dong, C., Zhang, W. E., Yue, L., Xu, M., Maennel, O., and Chen, W. Understanding why large language models can be ineffective in time series analysis: The impact of modality alignment. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 4026–4037. Association for Computing Machinery, 2025.

Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. One fits all: Power general time series analysis by pretrained lm, 2023. URL https://arxiv.org/abs/2302.11939.

A. Experimental Setup and Metrics

A.1. Hyperparameters

Table 1. Hyperparameter configuration for all forecasting experiments.

Category	Parameter	Value
Model	Base model	Qwen3-0.6B
	Architecture	Qwen3ForCausalLM (default config)
Tokenizer	Scaling	Z-score (mean/std from context)
	Binning	Uniform
	Vocab size (V)	1024
	Bin range	$[-5, 5]$
	Use EOS token	No
Training	Optimizer	AdamW
	Learning rate	$\{1, 3\} \times 10^{-4}$, $\{1, 3\} \times 10^{-3}$
	LR schedule	Linear warmup + cosine decay
	Warmup ratio	3%
	Precision	bf16 (mixed)
Batching	Effective batch size	128
	Epoch length	5,000 steps
Data	Dataset	GiftEval Pretrain (152 sub-datasets)
	Context length	512
	Target length	64
	Window stride	600
Loss	Loss function	Quantile (pinball) loss
	Quantiles	$\{0.1, 0.2, \ldots, 0.9\}$
	Softmax temperature	10^{-2}
Evaluation	Eval samples	1,000 held-out windows
	Eval horizons	$h \in \{1, 64\}$
	Eval strategy	auto-regressive
LoRA	Rank (r)	4, 8
	Alpha (α)	16, 32
	Dropout	0.05
	Targets (Attn)	`q_proj, k_proj, v_proj, o_proj`
Reproducibility	Random seed	420
	Compute usage	$\approx$12 hours on 8$\times$A100 per model

A.2. Evaluation Metrics

Following GluonTS (Alexandrov et al., 2020), all metrics are computed per-sequence and macro-averaged across sequences. Time steps with NaN ground truth are masked. We report:

CRPS: approximated as $\mathrm{CRPS}_{\text{approx}} = 2 \sum_q w_q \mathrm{QL}_q(y, \hat{y})$ over quantile levels.

MSE / RMSE / MAE: standard point-forecast metrics on the median prediction.

MASE: $\mathrm{MAE}/[\frac{1}{H-m} \sum_{t=m+1}^{H} |y_t - y_{t-m}|]$ with seasonality m selected by frequency.

NRMSE / ND / MAPE / sMAPE / wMAPE: scale-normalized variants with $\epsilon = 10^{-8}$ to prevent division by zero.

Directional Accuracy: fraction of horizon points whose sign of change relative to the last observed value is predicted correctly.

Pearson: linear correlation between predicted and true horizon.

A.3. Full Result Tables

Table 2. Single-step (h=1) forecasting at training step 128. All NanoTS variants use Qwen3-0.6B. Best within each section in **bold**.

	Model	CRPS↓	MSE↓	MAE↓	MASE↓	RMSE↓	NRMSE↓	ND↓	MAPE↓	sMAPE↓	wMAPE↓	DA↑
Pretrained	Full Finetune	49.51	10461	27.60	**1.561**	27.60	0.286	0.286	0.286	0.175	0.199	0.453
	IO Only	59.56	84846	66.53	3.430	66.53	0.614	0.614	0.614	0.300	0.250	0.469
	LoRA Attn	**20.10**	**8818**	**24.68**	1.607	**24.68**	**0.243**	**0.243**	**0.243**	**0.168**	**0.107**	**0.478**
	LoRA Attn+IO	22.54	11220	26.11	1.702	26.11	0.270	0.270	0.270	0.174	0.124	0.450
Random Init	Full Finetune	151.8	415339	180.4	7.819	180.4	0.914	0.914	0.914	0.532	0.463	**0.542**
	IO Only	111.6	280932	142.3	6.565	142.3	0.706	0.706	0.706	0.462	0.313	0.540
	LoRA Attn	154.5	430387	182.2	7.872	182.2	0.917	0.917	0.917	0.532	0.481	0.535
	LoRA Attn+IO	**50.68**	**59864**	**62.90**	**3.438**	**62.90**	**0.599**	**0.599**	**0.599**	**0.281**	**0.220**	0.476
Baselines	Chronos-Bolt	14.94	5409	18.91	0.934	18.91	0.240	0.240	0.240	0.146	0.096	0.678
	Chronos-T5	29.88	12764	32.99	1.212	32.99	**0.173**	**0.173**	**0.173**	**0.124**	**0.073**	0.646
	Chronos-2	**10.89**	**2886**	**13.38**	0.830	**13.38**	0.192	0.192	0.192	0.133	0.077	**0.724**
	ARIMA	11.94	3327	14.86	**0.805**	14.86	0.201	0.201	0.201	0.138	0.083	0.708
	Qwen3 Text	213.5	2.6M	213.5	376667	213.5	39508	39508	39508	0.563	19754	0.503

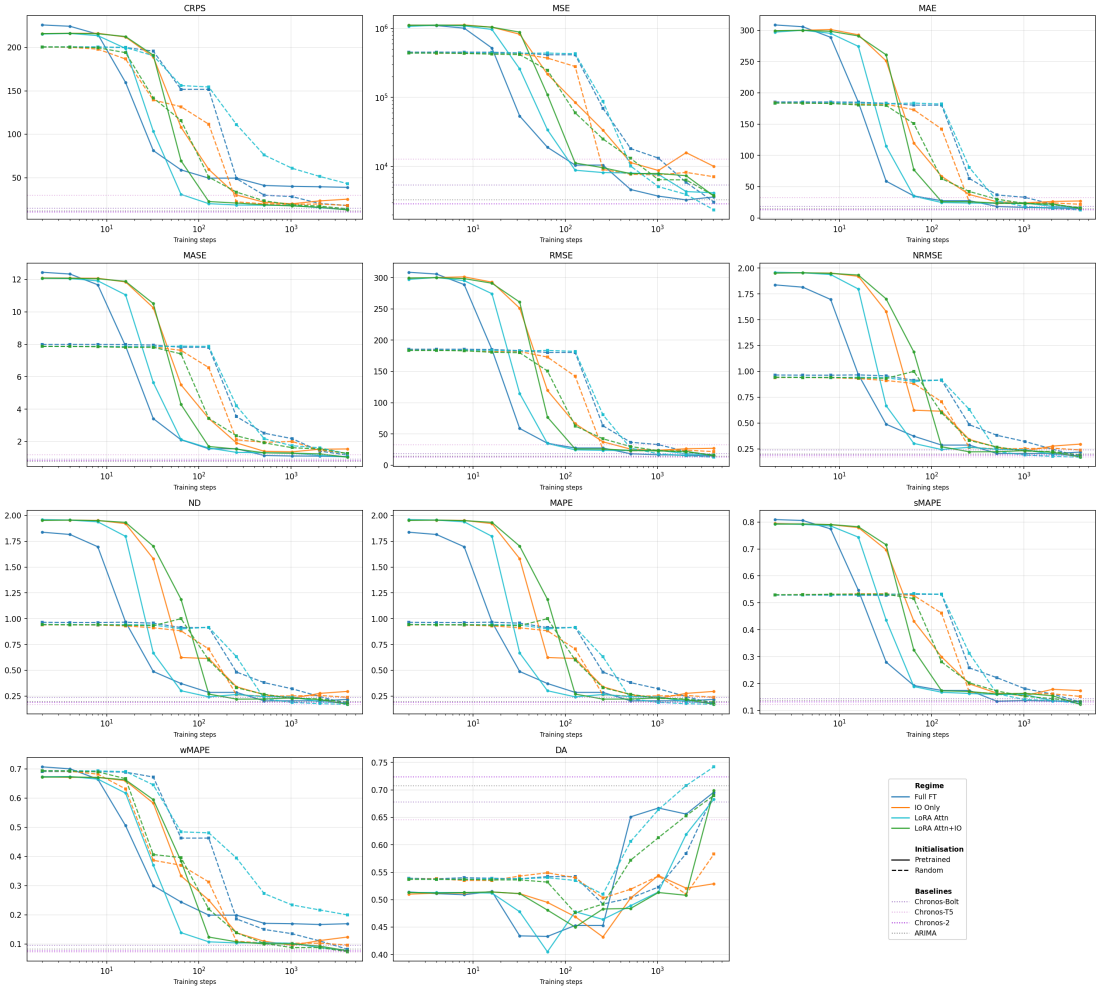

Figure 5. Training progression of all h=1 metrics across training steps for the four regimes. Solid: pretrained; dashed: random init; dotted horizontals: baselines. Pretrained initialization helps most in the first 100 steps and remains better after convergence.

B. Latent-Decoding Experiment: Additional Results

B.1. Fair top-K comparison

Table 3. Fair top-K comparison across ablation conditions. For each condition, we take the best-K unique nearest-neighbor matches and report mean MSE. "—" indicates fewer than K unique matches available. Text + PT achieves the lowest MSE at every K.

K	text + PT	text + RandInit	rand + PT	rand + RandInit
4	**0.254**	0.383	0.330	0.412
54	**0.350**	0.721	—	0.755
99	**0.383**	0.825	—	0.862
200	**0.441**	0.971	—	1.004
439	**0.535**	—	—	—
686	**0.639**	—	—	—

B.2. What creates the latent forecasting structure?

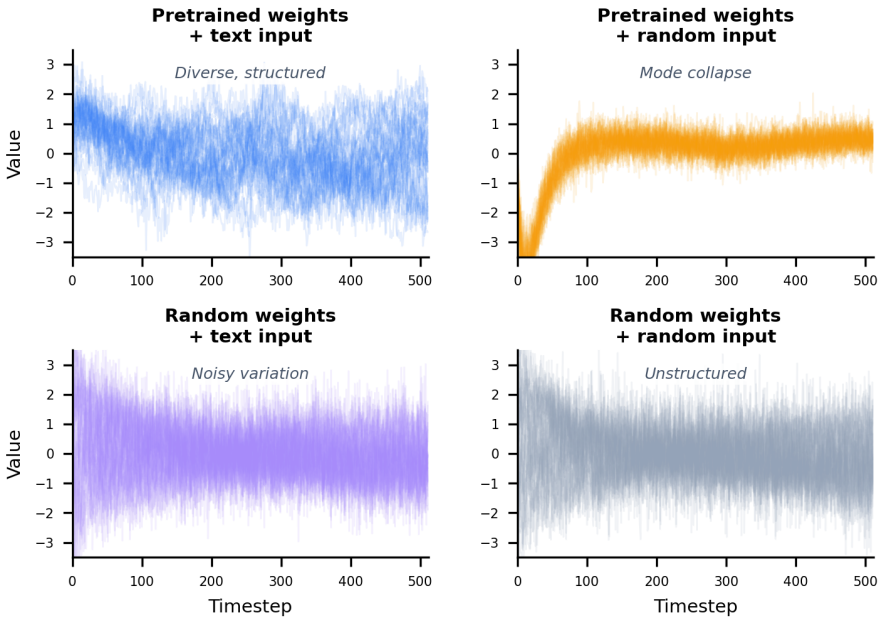

Figure 6. 2×2 **ablation crossing model weights and input.** *Top-left*: pretrained + text—diverse, structured trajectories. *Top-right*: pretrained + random tokens—outputs collapse to a few modes. *Bottom-left*: random weights + text—noisy variation without structure. *Bottom-right*: random weights + random tokens—unstructured. Only pretrained weights *and* meaningful text together produce diverse temporal signals.

C. Effective Data Transfer

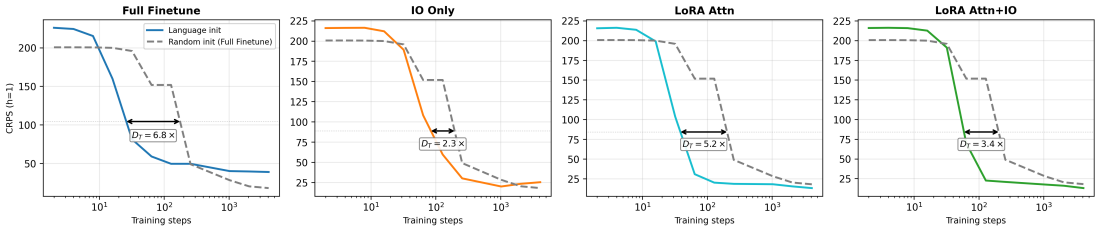

Figure 7. **Effective data transfer (Hernandez et al., 2021).** CRPS (h=1) versus training steps per regime, comparing language-pretrained initialization (solid) against the fully-finetuned random-init baseline (dashed). D_T is the multiplicative reduction in training steps needed to reach a given performance.

Let $\mathcal{L}_R(d)$ and $\mathcal{L}_P(d)$ denote validation losses after training for d tokens with random and pretrained initializations. For a target loss ℓ, the inverse $\mathcal{L}_\cdot^{-1}(\ell)$ gives the number of training tokens needed to reach it. Following Hernandez et al. (2021), the *effective data transferred* at ℓ is

$$D_T(\ell) := \mathcal{L}_R^{-1}(\ell) - \mathcal{L}_P^{-1}(\ell).$$

A positive D_T indicates that language pretraining reduces the forecasting data required to reach loss ℓ.

Across regimes, pretrained initialization reaches a target CRPS in substantially fewer steps than random initialization. Full finetuning saves $D_T = 6.8$, attention-only LoRA saves $D_T = 5.2$, LoRA+IO saves $D_T = 3.4$, and IO-only saves $D_T = 2.3$. A rank-8 LoRA adapter retains most of the pretrained advantage, which indicates that specialization moves in a small, structured subspace of the pretrained representation rather than reconfiguring it.

D. Catastrophic Forgetting Under Forecasting Specialization

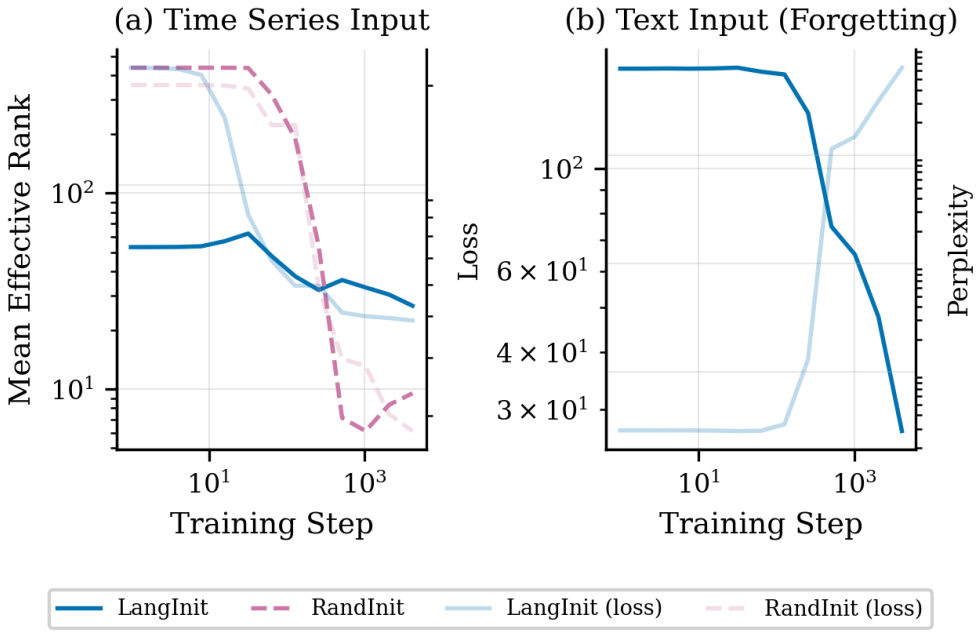

Figure 8. **Effective rank during specialization.** Mean effective rank across all 28 layers. **Left:** on time-series input, randomly initialized models start near-isotropic (∼440) and collapse to ∼10 within 500 steps; pretrained models decline more gradually from a much lower baseline (∼50) to ∼27. Transparent lines show CRPS on a secondary axis. **Right:** on text input, pretrained models' effective rank falls from ∼165 to ∼25 (85% reduction) while perplexity rises, consistent with catastrophic forgetting of language-specific directions as the model specializes into forecasting.

The trajectory analysis is consistent with the representation phases of Li et al. (2025): random initialization undergoes a warmup collapse followed by entropy-seeking expansion; pretrained initialization skips the warmup collapse and instead redistributes capacity, preserving the directions most useful for predicting the next step in a numerical sequence.