

---

# Context–Answer Metamodels: Forecasting LLM Activations Before the First Answer Token

---

Tommy He  
Tufa Labs

Thomas Jiralerspong  
Mila, Université de Montréal

## Abstract

Under stochastic decoding, a single prompt can elicit many different responses from a language model. Characterizing this variation typically requires many expensive autoregressive rollouts, each revealing only one realization. We ask whether the activations produced while reading the prompt can directly forecast the model’s rollout activations. We train *context–answer metamodels* that map a prompt activation sequence to point or distribution forecasts of mean-pooled answer activations. We find that (i) the four metamodel types we study forecast mean-pooled answer activations and their rollout variability with high fidelity; (ii) performance improves as the training set grows to 500,000 prompts and generalizes zero-shot to out-of-domain tasks; (iii) these metamodels retain behavior-predictive signal; and (iv) prompt activations from a weaker model predict mean-pooled answer activations from a stronger model, suggesting cross-model transfer from weaker to stronger models. Together, these results suggest that prompt activation traces can support scalable forecasts of behaviors in LLM answers before the first answer token is generated.

## 1 Introduction

How much of a language model’s future output is predictable before it emits its first answer token? Under stochastic decoding, a fixed prompt can produce different answer trajectories even when the model and decoding configuration are held fixed. Estimating this variability normally requires sampling multiple expensive rollouts. We ask whether we can forecast it directly from context activations. Such a forecast is useful in applications where a single completion is an unreliable indicator of a model’s possible behavior, such as pre-generation rare-behavior screening and pass@ $k$  estimation. Behavior screening often samples many answers and scores them with a probe or another language model. This is costly for rare events, and it provides no warning before the sampled behavior occurs. Rather than model combinatorially many complete token sequences directly, we forecast their fixed-dimensional answer representations.

Prior work predicts later tokens and hidden states from earlier activations [Pal et al., 2023], while other work uses prompt activations to predict response attributes, hallucination risk, and answer correctness [Dong et al., 2025, Ji et al., 2024, Moreno Cencerrado et al., 2025]. Most directly, Jiralerspong et al. [2026] fit linear and nonlinear context–answer maps on 500,000 training prompts, with one sampled answer per prompt. We study point and distribution forecasts under repeated-rollout supervision, evaluating both point and distribution prediction fidelity.

We train *context–answer metamodels*<sup>1</sup> on prompt activation traces and repeated answer activation traces from rollouts of a language model. *Point* metamodels forecast the expectation of the mean-pooled answer activation; *distribution* metamodels forecast its rollout-induced distribution. Figure 1

---

<sup>1</sup>Code repository: <https://github.com/Tufalabs/context-answer-metamodels>.

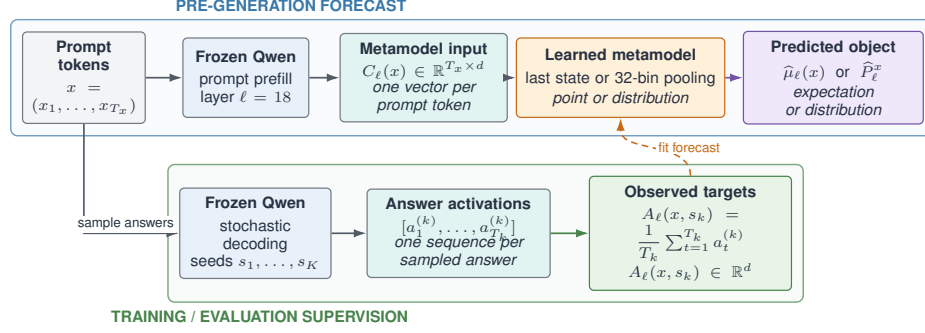


Figure 1: **Forecasting a future activation distribution.** The metamodel receives prompt activations from a language model and produces either a point forecast of the expected mean-pooled answer activation (linear or MLP) or a distribution forecast over mean-pooled answer activations (Gaussian or flow). Multiple decoding seeds provide training and evaluation targets.

summarizes the training and evaluation procedure. We separately test whether the forecasts preserve behavior-predictive directions present in real answer activations.

Our contributions are:

- We compare linear and MLP predictors of the conditional mean with Gaussian and flow-matching models of rollout variability.
- We show continued scaling through 500,000 training prompts and zero-shot transfer from LMSYS to WeirdChat and IFEval.
- We show that forecasted answer activations retain behavior-predictive signal.
- We show that prompt activations from Qwen2.5 can forecast Qwen3.5 and Gemma 4 answer activations, retaining most of the accuracy of matched target-model conditioners at scale.

## 2 Methodology

### 2.1 Problem formulation

Let  $M$  be a frozen autoregressive language model with hidden dimension  $d$ ,  $D$  a fixed stochastic decoding configuration, and  $x$  a prompt with token length  $T_x$ . Let  $s \sim \mathbb{P}_D$  denote a pseudorandom seed. Prefill at residual layer  $\ell$  yields a prompt activation trace  $C_\ell(x) \in \mathbb{R}^{T_x \times d}$ . A rollout under  $D$  yields a variable-length answer activation trace that we mean-pool into  $A_\ell(x, s) \in \mathbb{R}^d$  [Wang et al., 2026, Der et al., 2026]. This stochastic decoding process defines both a conditional expectation and a conditional law:

$$\begin{aligned} \mu_\ell(x) &:= \mathbb{E}_{s \sim \mathbb{P}_D} [A_\ell(x, s) \mid x] \\ P_\ell^x &:= \text{Law}_{M,D}(A_\ell(x, s) \mid x). \end{aligned} \quad (1)$$

We accordingly study two prediction targets: a *point metamodel* models the expectation  $\mu_\ell$ , and a *distribution metamodel* models the whole rollout-induced law,  $P_\ell^x$ . We write  $\hat{\mu}_\ell(x)$  and  $\hat{P}_\ell^x$  for the corresponding learned forecasts, appending a model-family subscript only when needed.

### 2.2 Four metamodel architectures

We train two point metamodels (linear ridge and MLP) and two distribution metamodels (Gaussian and rectified flow). We select the former by validation  $R^2$  and the latter by validation energy score. For inputs, the linear ridge model takes the final prompt-token activation, whereas the MLP and rectified flow take in binned activations (the Gaussian extends a frozen MLP). We bin the sequence of activations into  $B = 32$  bins and average within each bin to obtain bin representations  $r_1, \dots, r_B \in \mathbb{R}^d$ . We then learn a query  $q$  and pool them as

$$\alpha_j = \text{softmax}_j \left( \frac{q^\top \text{LN}(r_j)}{\sqrt{d}} \right), \quad z = \sum_{j=1}^B \alpha_j r_j. \quad (2)$$

We use single-query, query-only attention pooling:  $q$  is one learned global query, the normalized bins are used directly to compute its attention scores, and the unprojected bins are the values. The resulting  $z$  is passed directly to the downstream MLP or flow layers as the learned first stage of the architecture; its query is trained jointly with the downstream model. For outputs, the former predict the expected mean-pooled answer activation, whereas the latter capture its rollout-induced distribution.

**Linear ridge point metamodel.** The linear baseline receives only the final prompt token activation  $h_{\text{last}} \in \mathbb{R}^d$  and predicts  $\hat{\mu}_{\ell, \text{lin}}(x) = W\tilde{h}_{\text{last}} + b$ , where the tilde denotes coordinatewise standardization fit on the training set. We fit the full-dimensional ridge map in closed form to each prompt’s empirical rollout mean. We select the ridge penalty  $\lambda$  by validation  $R^2$ . Appendix Figure 5 ablates the choice of input representation.

**MLP point metamodel.** The MLP forecast  $\hat{\mu}_{\ell, \text{MLP}}(x)$  maps  $z$  through a three-layer  $4096 \rightarrow 8192 \rightarrow 8192 \rightarrow 4096$  GELU MLP (134.2M parameters) and minimizes MSE to each prompt’s empirical mean over training rollouts. We select the final model with the highest validation  $R^2$ . Appendix Figure 6 ablates the choice of input representation.

**Gaussian distribution metamodel.** This model extends a frozen MLP point map as  $\hat{P}_{\ell, G}^x = \mathcal{N}(\hat{\mu}_{\ell, \text{MLP}}(x), \kappa^2 \gamma(z)^2 \Sigma)$ . We estimate  $\Sigma$  from within-prompt residuals  $A(x, s) - \bar{A}(x)$ . Candidate covariances are isotropic, diagonal, or low-rank-plus-diagonal. The scale is either global or prompt-dependent;  $\gamma(z)$  is a two-layer  $4096 \rightarrow 256 \rightarrow 1$  SiLU head trained to predict log within-prompt variance. Covariance family, rank, scale mode, and the scalar multiplier  $\kappa$  are selected by validation energy score.

**Rectified flow distribution metamodel.** The flow is trained on individual rollout targets while conditioned on  $z$ . For a Gaussian base draw  $\varepsilon$ , integrating the rectified flow vector field  $v_\theta(y_t, t, z)$  [Lipman et al., 2023] from  $t = 0$  to 1 produces one forecast sample  $\tilde{A}_\ell \sim \hat{P}_{\ell, F}^x$ . Varying  $\varepsilon$  while holding  $z$  fixed samples this learned forecast law. To train this map, for a mean-pooled answer activation target  $y$ , base draw  $\varepsilon$ , and  $t \sim \mathcal{U}[0, 1]$ , we set  $y_t = (1 - t)\varepsilon + ty$  and minimize

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E} \left[ \|v_\theta(y_t, t, z) - (y - \varepsilon)\|_2^2 \right]. \quad (3)$$

The vector field  $v_\theta$  has width 4096, sinusoidal time embeddings, and eight conditioned residual blocks (352.8M parameters). At inference, 64 Gaussian particles are integrated for 32 Euler steps. We select the final model with the lowest validation energy score. Appendix Figure 7 ablates the choice of input representation.

## 2.3 Experimental setup

Our primary study uses post-block layer 18 of Qwen3.5-9B [Qwen Team, 2026], as selected in Appendix Figure 9. For the training dataset, we use LMSYS Chat-1M [Zheng et al., 2023], with 500,000 training prompts, 1,000 validation prompts, and 3,000 test prompts. Each prompt has 4 independently seeded rollouts. The rollout-count ablation uses 100,000 training prompts and 1,000/3,000 validation/test prompts, with a common 16-rollout evaluation target. For OOD evaluation, we use WeirdChat [Chowdhury et al., 2026] and IFEval [Zhou et al., 2023]. We evaluate each frozen LMSYS metamodel on all 2,660 WeirdChat prompts and all 541 IFEval prompts, with 4 observed rollouts per prompt. We evaluate the linear and MLP forecasts and the means of the Gaussian and flow with held-out  $R^2$  against the empirical rollout mean. For the Gaussian and flow, we add the energy score [Gneiting and Raftery, 2007] divided by  $\sqrt{d}$ , Spearman correlation between predicted and observed prompt-level variance, and the ratio of generated to observed variance. Appendix C defines these metrics and reports additional secondary metrics. For cross-model forecasting, we predict Qwen3.5 layer-18 and Gemma 4 layer-47 answer activations from Qwen2.5 layer-19 prompt activations [Yang et al., 2024], with matched target-model conditioners as controls (Appendix G). Appendix D gives the full experimental settings, including activation boundaries and target construction.

### 3 Results

#### 3.1 Prompt activation traces forecast answer activations at scale and generalize OOD

Figure 2 shows a clear scale-dependent reversal in mean  $R^2$ . The linear ridge map initially outperforms the nonlinear MLP and flow mean on all three evaluation sets, but the nonlinear models overtake it as prompt coverage grows. The linear curves then plateau much earlier. At the largest scale, the flow mean approaches the MLP on both OOD sets, with a modest gap in domain. Every-token pooling further improves both neural models (Appendix A).

Table 1 reports endpoint distribution metrics: the Gaussian has a slightly better LMSYS energy score and variance ranking, whereas the flow has lower OOD energy scores. Appendix Figures 10 and 11 show the corresponding scaling curves, and Appendix F visualizes the forecasts. Increasing prompt coverage is more valuable than adding repeated targets: eight rollouts capture most of the gain from sixteen (Appendix Figure 8).

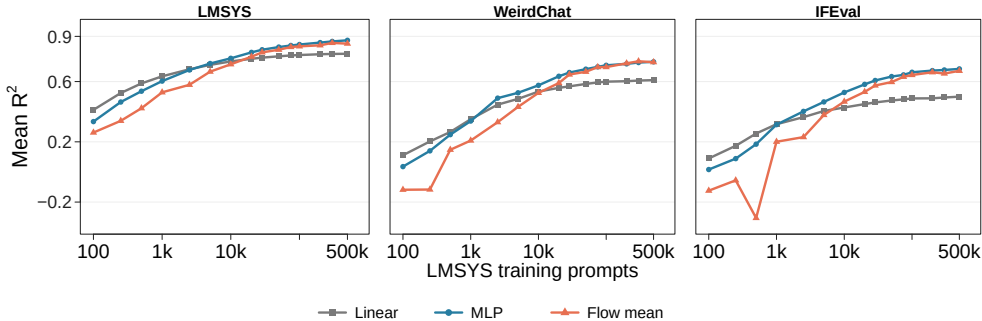


Figure 2: **Prompt coverage drives forecast quality through the largest tested scale, in and out of domain.** Columns correspond to evaluation sets and show mean  $R^2$  for the linear, MLP, and flow means. All metamodels are trained only on LMSYS; WeirdChat (2,660 prompts) and IFEval (541) are zero-shot, with 4 rollouts per prompt. Linear uses the final prompt-token activation and neural models use the 32-bin trace. Other endpoint metrics are in Table 1.

Table 1: Distribution metrics at the final 500,000-prompt LMSYS-only checkpoint. WeirdChat and IFEval are zero-shot evaluations.

| Evaluation set | Energy score ↓ |        | Variance $\rho$ ↑ |      | Variance ratio → 1 |       |
|----------------|----------------|--------|-------------------|------|--------------------|-------|
|                | Gaussian       | Flow   | Gaussian          | Flow | Gaussian           | Flow  |
| LMSYS          | .06686         | .06747 | .829              | .690 | 1.532              | 1.299 |
| WeirdChat      | .06978         | .06851 | .610              | .633 | 1.961              | 1.857 |
| IFEval         | .10330         | .10032 | .679              | .611 | 1.547              | 1.833 |

#### 3.2 Forecasted activations retain behavior-predictive signal

We fit behavior readouts on real answer activations, then apply them to predicted means or average their outputs over Gaussian and flow samples. Figure 3 shows AUPRC on held-out WeirdChat answers judged by GPT-6 Luna. The MLP mean improves on direct context in all four groups, although some gains are small. Gaussian performs similarly to MLP on self-harm and deception, while flow’s relative performance varies by behavior. Appendix H gives the setup and complementary AUROC results.

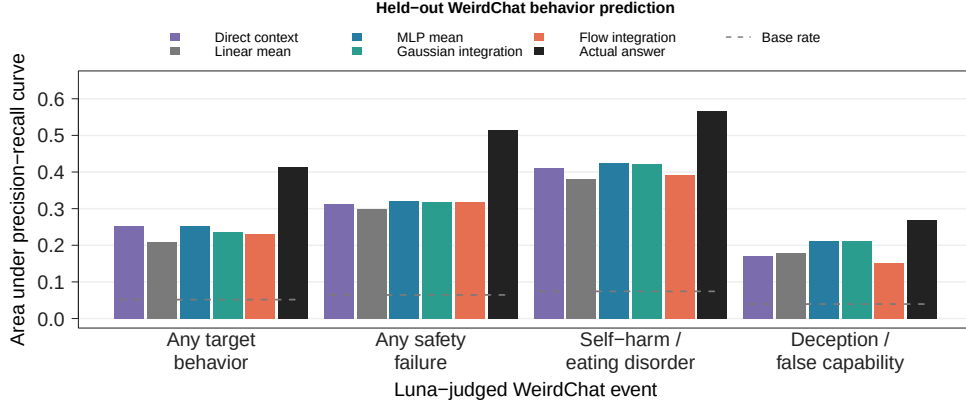


Figure 3: **Forecasted activations retain behavior probe signal.** AUPRC on held-out WeirdChat prompts. Dashed segments show each group’s positive-rollout base rate. The actual-answer probe observes generated answers.

### 3.3 Weaker model traces forecast stronger model rollout distributions

We train all four metamodel families from Qwen2.5 layer-19 prompt activations to Qwen3.5 layer-18 and Gemma 4 layer-47 answer activations, with matched target-model conditioners as controls. Conditioning on Qwen2.5 yields slightly lower but comparable mean-forecast accuracy on both targets (Figure 4). This transfer spans architectures for Qwen3.5, whose hybrid blocks differ from Qwen2.5’s conventional Transformer [Yang et al., 2024, Qwen Team, 2026], and model training for Gemma 4.

The distribution comparison depends on the target (Table 6). For Qwen3.5, the Gaussian ranks prompt-level variance more strongly, while the flow is closer to the ideal unit variance ratio. For Gemma 4, the flow has lower energy scores and stronger variance ranking under both conditioners. Appendix G gives protocols and additional metrics.

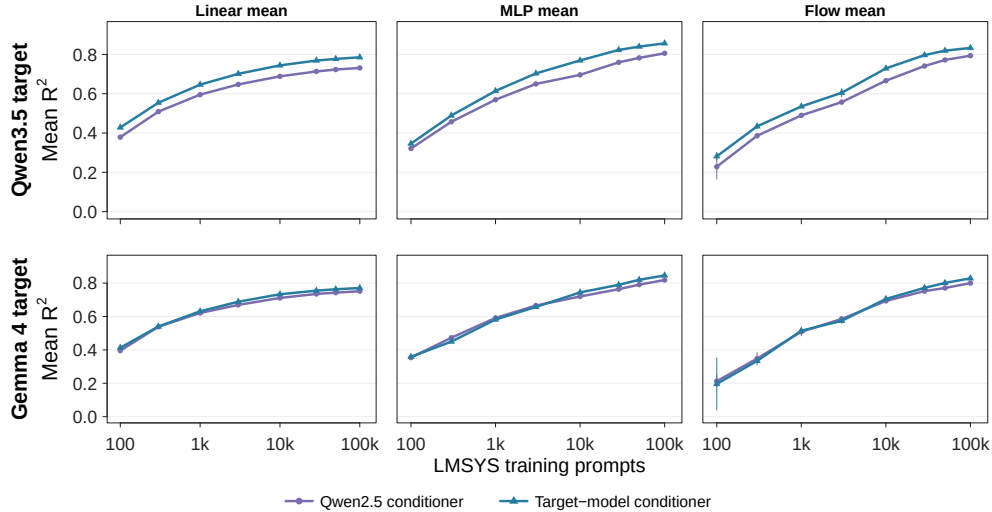


Figure 4: **Qwen2.5 traces preserve most Qwen3.5 and Gemma 4 forecast quality.** Mean  $R^2$  for linear, MLP, and flow metamodels. Rows show target models; curves compare Qwen2.5 and matched target-model conditioning. Linear models receive the final prompt-token activation; MLP and flow receive 32-bin traces. Error bars show one standard deviation over three rollout partitions for Qwen3.5 and Gemma 4.

## 4 Related work

**Reading and predicting future computation.** Prompt-time probes predict response structure and content [Dong et al., 2025], hallucination risk [Ji et al., 2024], and answer correctness [Moreno Cencerrado et al., 2025]. Reasoning-trace probes predict later behavior [Kortukov et al., 2026], while refusal directions identify behavior-linked geometry [Arditi et al., 2024]. These methods target chosen labels. The Tuned Lens decodes the current token prediction [Belrose et al., 2023]; Future Lens predicts later tokens and hidden states [Pal et al., 2023]; and EAGLE predicts near-future features for speculative decoding [Li et al., 2024]. Closest to our target, Jiralerspong et al. [2026] predict a pooled answer activation from the last prompt activation. We instead predict a rollout-induced distribution with multi-rollout supervision and out-of-domain tests. We mean-pool answers because information is generally spread across generated positions [Wang et al., 2026, Der et al., 2026].

**Metamodels and neural artifacts as data.** Metamodels take neural-network artifacts as training examples for a second model. Hypernetworks generate the weights of another network [Ha et al., 2017]; newer work learns representations or generative models of checkpoint populations [Peebles et al., 2022, Schürholt et al., 2024, Wang et al., 2024]. Generative Latent Priors extend this view to activation distributions and use the learned model as a prior and feature extractor [Luo et al., 2026]. We study a different artifact population: the conditional relationship between an observed prompt trace and a later answer summary.

**Generative models of activations.** Activation decoders map hidden states to descriptions of model knowledge or behavior [Costarelli et al., 2024, Pan et al., 2024]. UniSteer learns a text-conditioned activation flow for editing and classification [Shi et al., 2026]. We instead condition on an earlier trace from the same model and predict the distribution of a later pooled state under stochastic decoding. We use conditional flow matching [Lipman et al., 2023] and evaluate it with both point and distribution metrics [Gneiting and Raftery, 2007, Pacchiardi et al., 2024].

## 5 Discussion

We built point and distributional context–answer metamodels and found that they scaled reliably with prompt count, generalized out-of-domain, and retained behavior signal for probing. The point-model results show that MLP regression can provide accurate mean forecasts that retain behavioral signal without explicitly modeling rollout variability. The distributional-model results support Gaussian forecasts as a useful baseline for capturing rollout variability, with the benefits of flow matching depending on the target model and the distributional properties being evaluated.

**Limitations** The main distributional targets came from Qwen3.5 layer 18; the cross-model extension adds Gemma 4 layer 47. We selected layer 18 for in-domain conditional mean accuracy, so it might not have been optimal for these scores, transfer, or behavior prediction. Training and analysis used only 4–16 rollouts, leaving room for further scaling robustness. Behavior labels come from GPT-6 Luna rubric judgments but have not been extensively validated against human annotations. The learned law was also specific to the decoding rule, chat template, generation cap, activation boundary, and answer-pooling choice.

**Future Directions** Future work should explore how activation forecasts can support behavioral monitoring and intervention, for example by anticipating reward hacking. More ambitious targets, such as activation trajectories, could capture how behavior unfolds during generation. Extending forecasts beyond the initial prompt could enable monitoring over longer and multi-agent interactions. Scaling to larger and more diverse datasets could broaden their usefulness across tasks and domains.

## Acknowledgments

We thank Jerome Sieber for detailed feedback on the paper as well as technical discussions. We additionally thank Stephen Lu and Jungyoon Lee for technical discussions on flow matching models.

## References

- Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4322. URL [https://proceedings.neurips.cc/paper\\_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html](https://proceedings.neurips.cc/paper_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html).
- Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023. URL <https://arxiv.org/abs/2303.08112>.
- Neil Chowdhury, Cassidy Laidlaw, Kaiying Hou, Daniel Johnson, Sarah Schwettmann, and Jacob Steinhardt. WeirdChat: A catalog of unexpected AI behaviors, discovered automatically. Translucence dataset, July 2026. URL <https://translucence.org/weirdchat>.
- Anthony Costarelli, Mat Allen, and Severin Field. Meta-models: An architecture for decoding LLM behaviors through interpreted embeddings and natural language, 2024. URL <https://arxiv.org/abs/2410.02472>.
- Kevin Der, Harish Kamath, and Ben Thompson. Turn-averaged SAEs for feature discovery and long-context attribution, 2026. URL <https://arxiv.org/abs/2606.28548>.
- Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 14301–14320. PMLR, 2025. URL <https://proceedings.mlr.press/v267/dong25p.html>.
- Tilman Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.
- Google DeepMind. Gemma-4-12B-it. Hugging Face model card, 2026. URL <https://huggingface.co/google/gemma-4-12B-it>.
- David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=rkpACe1lx>.
- Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. LLM internal states reveal hallucination risk faced with a query, 2024. URL <https://arxiv.org/abs/2407.03282>.
- Thomas Jiralerspong, Mukesh Ramanathan, Christopher Ackerman, Christina Lu, Guillaume Lajoie, and Dan Mossing. Affine anticipation: LLM answer representations are largely linearly predictable from context. Preprint, 2026. URL <https://thomas.jiralerspong.com/assets/docs/affine-anticipation.pdf>.
- Evgenii Kortukov, Piotr Komorowski, Florian Klein, Paula Engl, Gabriele Sarti, Seong Joon Oh, Sebastian Lapuschkin, and Wojciech Samek. Predicting future behaviors in reasoning models enables better steering, 2026. URL <https://arxiv.org/abs/2606.11172>.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28935–28948. PMLR, 2024. URL <https://proceedings.mlr.press/v235/li24bt.html>.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- Grace Luo, Jiahai Feng, Trevor Darrell, Alec Radford, and Jacob Steinhardt. Learning a generative meta-model of LLM activations, 2026. URL <https://arxiv.org/abs/2602.06964>.

- Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. No answer needed: Predicting LLM answer accuracy from question-only linear probes, 2025. URL <https://arxiv.org/abs/2509.10625>.
- Lorenzo Pacchiardi, Rilwan A. Adewoyin, Peter Dueben, and Ritabrata Dutta. Probabilistic forecasting with generative networks via scoring rule minimization. *Journal of Machine Learning Research*, 25(45):1–64, 2024. URL <https://www.jmlr.org/papers/v25/23-0038.html>.
- Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning*, pages 548–560. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.conll-1.37. URL <https://aclanthology.org/2023.conll-1.37/>.
- Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language, 2024. URL <https://arxiv.org/abs/2412.08686>.
- William Peebles, Ilija Radosavovic, Tim Brooks, Alexei A. Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints, 2022. URL <https://arxiv.org/abs/2209.12892>.
- Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026. URL <https://huggingface.co/Qwen/Qwen3.5-9B>.
- Konstantin Schürholt, Michael W. Mahoney, and Damian Borth. Towards scalable and versatile weight space learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 43947–43966. PMLR, 2024. URL <https://proceedings.mlr.press/v235/schurholt24a.html>.
- Yingdong Shi, Ruiming Zhang, Changming Li, Zhiyu Yang, Kaixing Zhang, Jingyi Yu, and Kan Ren. UniSteer: Text-guided flow matching in activation space for versatile LLM steering, 2026. URL <https://arxiv.org/abs/2605.30076>.
- Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion, 2024. URL <https://arxiv.org/abs/2402.13144>.
- Sophie L. Wang, Phillip Isola, and Brian Cheung. The truth lies somewhere in the middle (of the generated tokens), 2026. URL <https://arxiv.org/abs/2605.09969>.
- An Yang et al. Qwen2.5 Technical Report, 2024. URL <https://arxiv.org/abs/2412.15115>.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A large-scale real-world LLM conversation dataset, 2023. URL <https://arxiv.org/abs/2309.11998>.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL <https://arxiv.org/abs/2311.07911>.

## A Scaling and context-representation ablations

The representation ablations use Qwen3.5-9B layer 18. Linear maps compare three inputs: 32 average bins concatenated, the last prompt token, and the unweighted mean of prompt tokens. MLP and flow additionally compare every prompt token; neural sequence inputs use learned-query pooling. All fits use LMSYS-only training, common 1,000-prompt validation and 3,000-prompt test splits, plus zero-shot evaluation on 2,660 WeirdChat and 541 IFEval prompts. The Gaussian is omitted because its analytic mean is the MLP forecast.

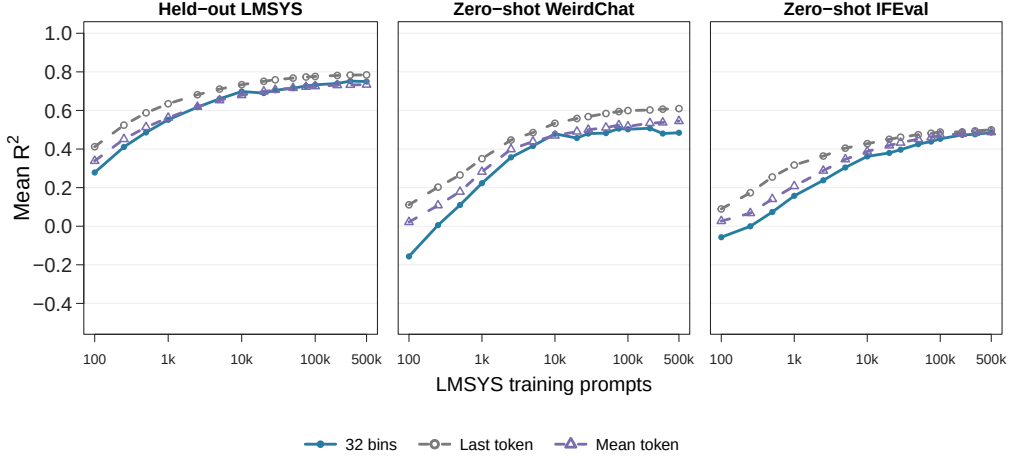


Figure 5: **Linear prompt-input comparison.** Mean  $R^2$  for LMSYS-only linear maps from 100 to 500,000 training prompts. The three inputs are the final token, the unweighted mean of prompt tokens, and all 32 bins concatenated. Among the three 500k fits, the last-token map is strongest on all three domains; its LMSYS  $R^2$  is 0.784, versus 0.750 for 32 bins and 0.734 for the mean token. WeirdChat and IFEval are zero-shot; selection uses only LMSYS validation.

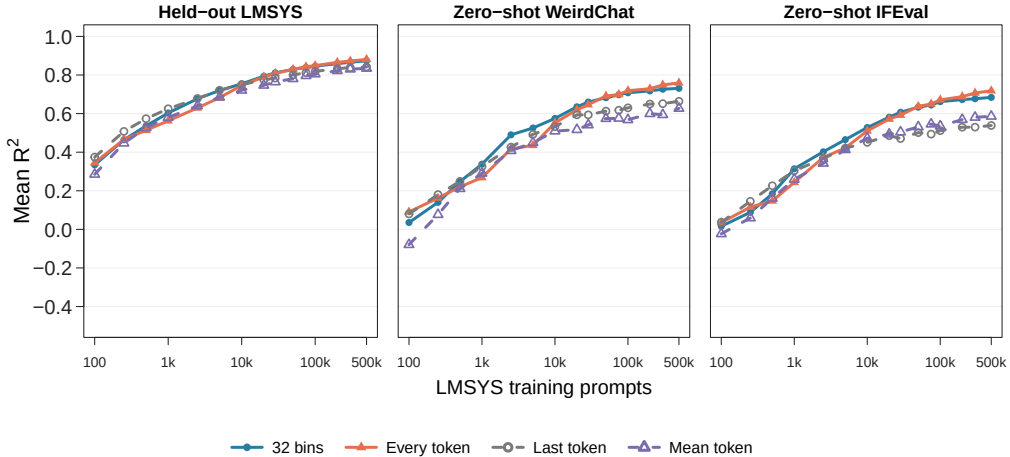


Figure 6: **Point-MLP prompt conditioner comparison.** Mean  $R^2$  from 100 to 500,000 training prompts. Columns show held-out LMSYS, zero-shot WeirdChat, and zero-shot IFEval. At 500k, every-token learned-query pooling reaches LMSYS  $R^2 = 0.881$ , compared with 0.874 for 32 bins, 0.842 for the last token, and 0.834 for the mean token. Every-token pooling has the highest endpoint  $R^2$  on all three domains (Table 2).

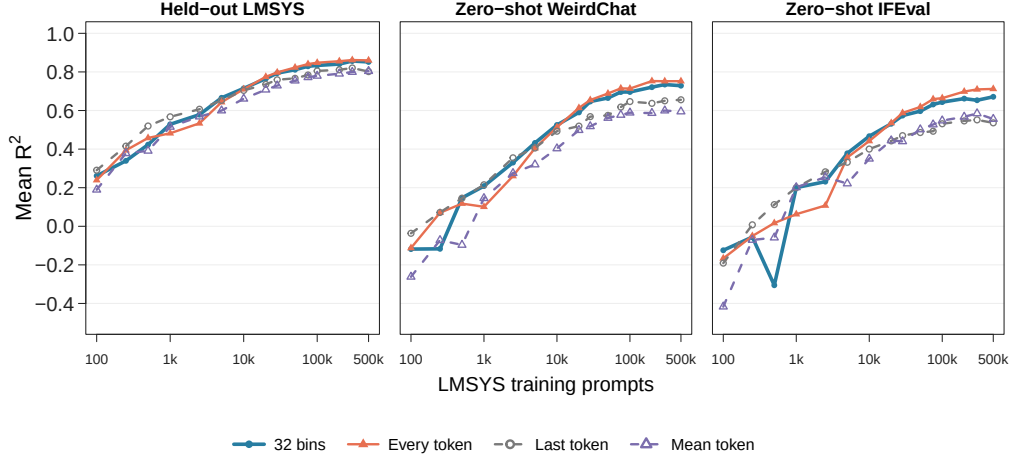


Figure 7: **Conditional-flow prompt conditioner comparison.** Flow-mean  $R^2$  from 100 to 500,000 training prompts. Columns show held-out LMSYS, zero-shot WeirdChat, and zero-shot IFEval. At 500k, every-token learned-query pooling reaches LMSYS  $R^2 = 0.860$ , versus 0.852 for 32 bins, 0.802 for the last token, and 0.804 for the mean token. Every-token pooling has the highest endpoint  $R^2$  on all three domains (Table 2).

Table 2: Representation endpoints at 500,000 LMSYS training prompts. WeirdChat and IFEval are zero-shot. Every-token pooling has the highest observed  $R^2$  among the four inputs for both neural models on all three domains. Linear maps compare 32 bins, last token, and mean token.

| Model  | Input       | LMSYS $R^2$ | WeirdChat $R^2$ | IFEval $R^2$ |
|--------|-------------|-------------|-----------------|--------------|
| Linear | 32 bins     | 0.7498      | 0.4844          | 0.4854       |
| Linear | Last token  | 0.7844      | 0.6095          | 0.4996       |
| Linear | Mean token  | 0.7335      | 0.5447          | 0.4873       |
| MLP    | 32 bins     | 0.8739      | 0.7309          | 0.6839       |
| MLP    | Every token | 0.8808      | 0.7591          | 0.7183       |
| MLP    | Last token  | 0.8418      | 0.6636          | 0.5383       |
| MLP    | Mean token  | 0.8343      | 0.6269          | 0.5859       |
| Flow   | 32 bins     | 0.8517      | 0.7286          | 0.6711       |
| Flow   | Every token | 0.8599      | 0.7520          | 0.7119       |
| Flow   | Last token  | 0.8016      | 0.6556          | 0.5360       |
| Flow   | Mean token  | 0.8040      | 0.5946          | 0.5557       |

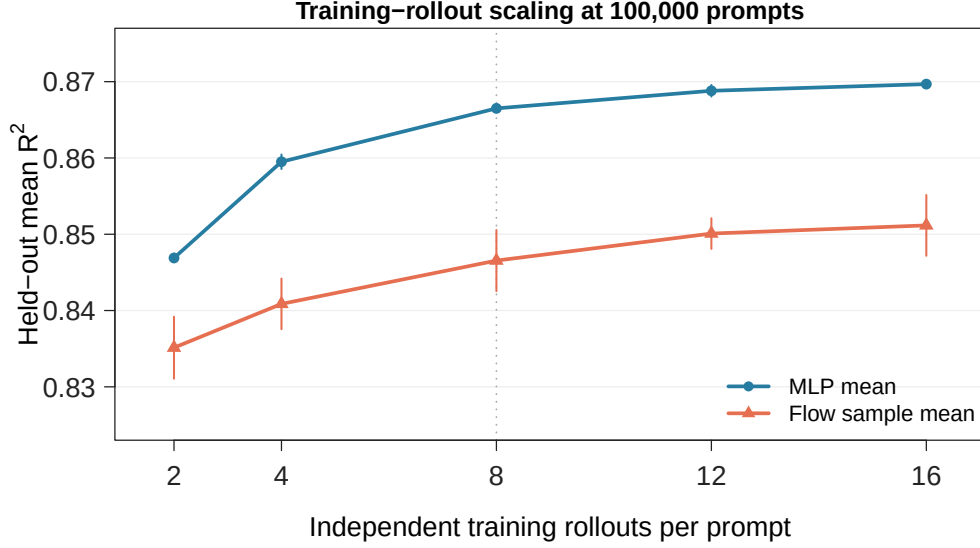


Figure 8: **Eight rollout targets capture most of the repeat-sampling gain.** Held-out mean  $R^2$  at 100,000 training prompts versus training rollouts. Error bars show the standard deviation across 3 fits. From 8 to 16 rollouts, MLP gains 0.0032 and flow gains 0.0046. The main experiments use 4 rollouts per prompt due to compute constraints. Table 3 gives the full values.

Table 3: Rollout-count results at 100,000 training prompts: mean  $R^2 \pm$  one standard deviation over 3 fits. All cells use the same 16-rollout evaluation mean, which differs from the four-rollout main target.

| Model  | 2 rollouts          | 4 rollouts          | 8 rollouts          | 12 rollouts         | 16 rollouts         |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Linear | $0.7893 \pm 0.0003$ | $0.7924 \pm 0.0003$ | $0.7940 \pm 0.0001$ | $0.7944 \pm 0.0000$ | $0.7946 \pm 0.0000$ |
| MLP    | $0.8469 \pm 0.0006$ | $0.8595 \pm 0.0010$ | $0.8665 \pm 0.0005$ | $0.8688 \pm 0.0008$ | $0.8697 \pm 0.0003$ |
| Flow   | $0.8351 \pm 0.0041$ | $0.8409 \pm 0.0033$ | $0.8465 \pm 0.0040$ | $0.8501 \pm 0.0020$ | $0.8512 \pm 0.0040$ |

## B Layer selection

Figure 9 shows the layer-selection sweep. At 10,000 training prompts, layer 18 reaches validation  $R^2 = 0.7577$ , compared with 0.7394 for layer 17, and remains the validation-selected layer. Similar studies place future-token, response, hallucination, and correctness signals in middle-to-late layers [Pal et al., 2023, Dong et al., 2025, Ji et al., 2024, Moreno Cencerrado et al., 2025].

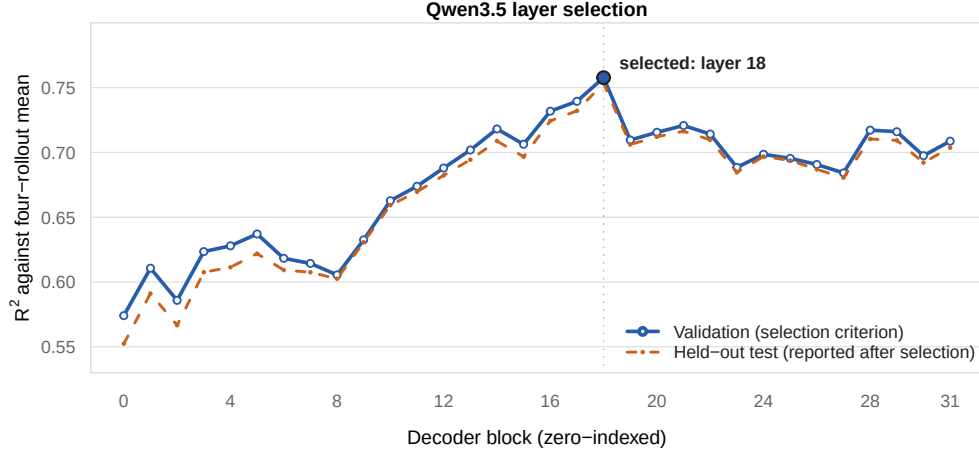


Figure 9: **Validation selection of Qwen3.5-9B layer 18.** Each two-hidden-layer MLP maps the last prompt activation at each decoder block to the mean-pooled answer activation over 4 temperature-1 rollouts. All fits use 10,000 LMSYS training prompts. We select on 1,000 validation prompts (blue); the 3,000-prompt test results are orange. Layer 18 has the best validation  $R^2$  (0.7577), followed by layer 17 (0.7394).

## C Evaluation metric definitions

Let  $\mu_i$  be the empirical rollout mean for prompt  $i$ ,  $\hat{\mu}_i$  the predicted mean, and  $\bar{\mu}$  the mean across prompts. We compute

$$R^2 = 1 - \frac{\sum_i \|\hat{\mu}_i - \mu_i\|_2^2}{\sum_i \|\mu_i - \bar{\mu}\|_2^2}. \quad (4)$$

For a forecast sample  $X$ , an independent forecast sample  $X'$ , and a realized rollout  $Y$ , the energy score is

$$\text{EnergyScore} = \mathbb{E}\|X - Y\|_2 - \frac{1}{2}\mathbb{E}\|X - X'\|_2. \quad (5)$$

We estimate both expectations with samples and divide the energy score by  $\sqrt{d}$ .

Table 4: Forecast metric definitions.

| Metric                                                               | Definition                                                                                                                                                                                    | Better |
|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| <i>Point metrics (point models and means of distribution models)</i> |                                                                                                                                                                                               |        |
| Mean $R^2$                                                           | Variance in empirical prompt-level rollout means explained by the predicted means.                                                                                                            | Higher |
| Prompt retrieval                                                     | Fraction of predicted means whose nearest empirical answer mean in Euclidean distance belongs to the same prompt.                                                                             | Higher |
| <i>Distribution metrics</i>                                          |                                                                                                                                                                                               |        |
| Energy score                                                         | Proper score above, averaged over prompts and divided by $\sqrt{d}$ .                                                                                                                         | Lower  |
| Variance Spearman                                                    | Across-prompt Spearman correlation between generated and observed covariance traces.                                                                                                          | Higher |
| Variance ratio                                                       | Mean trace of generated within-prompt covariance divided by the corresponding observed trace.                                                                                                 | 1      |
| NLL per dimension                                                    | Analytic Gaussian negative log-likelihood (with fixed $10^{-6}$ relative diagonal jitter) or approximate flow negative log-likelihood per activation dimension in raw activation coordinates. | Lower  |
| Predictive MMD <sup>2</sup>                                          | Multiscale RBF MMD between equal-sized generated and real sets in a fixed PCA-whitened space.                                                                                                 | Lower  |
| PCA sliced-Wasserstein                                               | Mean 1-D Wasserstein-2 distance over fixed random directions in the same PCA-whitened space.                                                                                                  | Lower  |

## D Experimental details

The primary model is Qwen3.5-9B, which provides both prompt and answer activations. Cross-model forecasting uses Qwen2.5-7B-Instruct layer-19 prompt activations to predict Qwen3.5-9B layer-18 or Gemma-4-12B-it layer-47 answer activations (Appendix G). We capture the raw output of decoder block 18 for every prompt and answer token. The answer target is the float32 mean of these stored bfloat16 post-block states. The four primary rollout seeds are 43–46. The 16-rollout LMSYS bank retains these and adds seeds 47–58 over the same 100,000 training prompts plus 1,000 validation and 3,000 test prompts. LMSYS, WeirdChat, and IFEval rollouts use a 4,096-token generation cap, with temperature 1.0 and top- $p$  0.95. The flow standardizes each target coordinate, while the point MLP uses a scalar target RMS after coordinatewise centering.

Table 5: Primary metamodel settings.

| Component                   | Setting                                                                                                                                                                                                                                                                                            |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Prompt input                | 32 average bins; Qwen3.5-9B layer 18 ( $d = 4,096$ ) or Qwen2.5-7B-Instruct layer 19 ( $d = 3,584$ )                                                                                                                                                                                               |
| Answer target               | Mean Qwen3.5-9B layer-18 activation over generated answer tokens and, for normally stopped answers, the non-overlapping canonical assistant suffix; $d = 4,096$                                                                                                                                    |
| Point MLP                   | LayerNorm for attention scores; 1 learned query; weighted sum of 32 bins; Linear–GELU–Linear–GELU–Linear with widths $4,096 \rightarrow 8,192 \rightarrow 8,192 \rightarrow 4,096$ ; no dropout; 134.2M parameters. Qwen2.5 input changes the first width from 4,096 to 3,584.                     |
| MLP training                | MSE to the empirical rollout mean; AdamW; batch size 256; learning rate $3 \times 10^{-4}$ ; weight decay $10^{-3}$ ; select by validation $R^2$                                                                                                                                                   |
| Rectified flow              | Learned-query attention conditioner; width 4,096; 8 residual blocks; sinusoidal time embedding; 352.8M parameters for Qwen3.5 input                                                                                                                                                                |
| Flow path                   | $\varepsilon \sim \mathcal{N}(0, I)$ , $t \sim \mathcal{U}[0, 1]$ , $y_t = (1 - t)\varepsilon + ty$ ; target velocity $y - \varepsilon$                                                                                                                                                            |
| Flow training               | Flow-matching MSE; AdamW; batch size 256; learning rate $3 \times 10^{-4}$ ; weight decay $10^{-4}$ ; gradient norm clipped at 1; select by validation energy score                                                                                                                                |
| Selection                   | Evaluate every 500 steps; stop after 12 evaluations without improvement; 80,000-step guard                                                                                                                                                                                                         |
| OOD evaluation              | Metamodels are evaluated on all 2,660 WeirdChat prompts and all 541 IFEval prompts; neither dataset enters fitting, normalization, or model selection for the zero-shot results.                                                                                                                   |
| WeirdChat behavior analysis | Behavior readouts use the fixed 1,596/532/532 prompt split. The predicted activations come from new mixed linear, MLP, and flow fits trained on 500,000 LMSYS and 1,596 WeirdChat prompts with equal domain weighting; Gaussian is fitted around the new MLP mean. Luna judges all 10,640 rollouts |
| Sampling                    | 64 samples and 32 Euler steps for distribution metrics; all 1,024 samples from each Gaussian or flow forecast for integrated behavior probes; flow uses 32 Euler steps                                                                                                                             |

### D.1 Compute resources

Experiments use NVIDIA B200 and RTX 5090 GPUs, with 6–12 CPU cores and 64–420 GiB of memory per job. We estimate roughly 400 GPU-hours for activation collection, 500 for primary metamodel scaling and ablations, 80 for cross-model forecasting, and 30 for OOD and behavior analyses. Preliminary, superseded, and failed runs bring the full-project total to roughly 1,300 GPU-hours.

### D.2 Existing asset versions and licenses

We use the following pinned versions under their stated terms:

- Qwen3.5-9B revision c202236235762e1c871ad0ccb60c8ee5ba337b9a and Qwen2.5-7B-Instruct revision a09a35458c702b33eeacc393d103063234e8bc28, both under the Apache License 2.0 (<https://huggingface.co/Qwen/Qwen3.5-9B>; <https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>).
- Gemma-4-12B-it revision 707f0a3b8a3c7ad586ed01e27eafbad8a27dd0f7 [Google DeepMind, 2026], under the Apache License 2.0 (<https://huggingface.co/google/gemma-4-12B-it>), used for the additional cross-model targets.

- LMSYS Chat-1M revision 200748d9d3cddcc9d782887541057aca0b18c5da, under the custom LMSYS-Chat-1M Dataset License Agreement, whose terms prohibit dataset transfer; we do not redistribute it (<https://huggingface.co/datasets/lmsys/lmsys-chat-1m>).
- WeirdChat revision 3460e41491308601cd68ee377011c7b7a8f0b4b3, under the MIT License (<https://huggingface.co/datasets/Transluce/WeirdChat>).
- IFEval data and evaluator at google-research revision 26d8ccdab6fec61b5c83ad6327ea8bda9e580288, with the data under CC BY 4.0 and source code under the Apache License 2.0 ([https://github.com/google-research/google-research/tree/master/instruction\\_following\\_eval](https://github.com/google-research/google-research/tree/master/instruction_following_eval)).

## E Out-of-domain evaluation

We evaluate frozen LMSYS-only metamodels on the full set of 2,660 WeirdChat prompts and all 541 official IFEval prompts, with 4 temperature-1 rollouts per prompt. On WeirdChat, MLP and flow reach  $R^2 = 0.731$  and  $0.729$  and retrieve the correct prompt in 53.1% and 47.9% of cases. On IFEval, the corresponding scores are  $R^2 = 0.684$  and  $0.671$ , with retrieval accuracy 94.6% and 92.2% [Zhou et al., 2023].

Retrieval queries and candidates use the same fixed banks of 3,000 LMSYS, 2,660 WeirdChat, and 541 IFEval prompts at every training size. All reported metrics use these evaluation populations. The separate mixed-domain behavior-probe experiment retains its held-out 532-prompt WeirdChat test split.

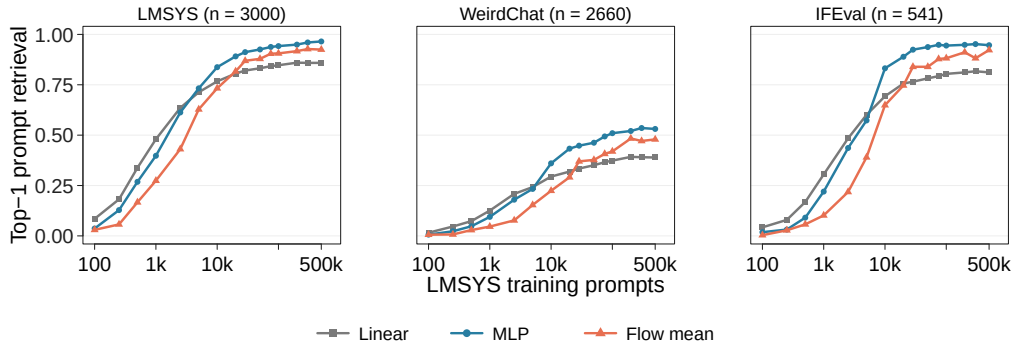


Figure 10: **Prompt retrieval across the LMSYS training scale, in and out of domain.** Top-1 retrieval from the predicted means of LMSYS-only linear, MLP, and flow metamodels. Candidate banks contain 3,000, 2,660, and 541 prompts for LMSYS, WeirdChat, and IFEval, respectively. WeirdChat and IFEval are zero-shot evaluations.

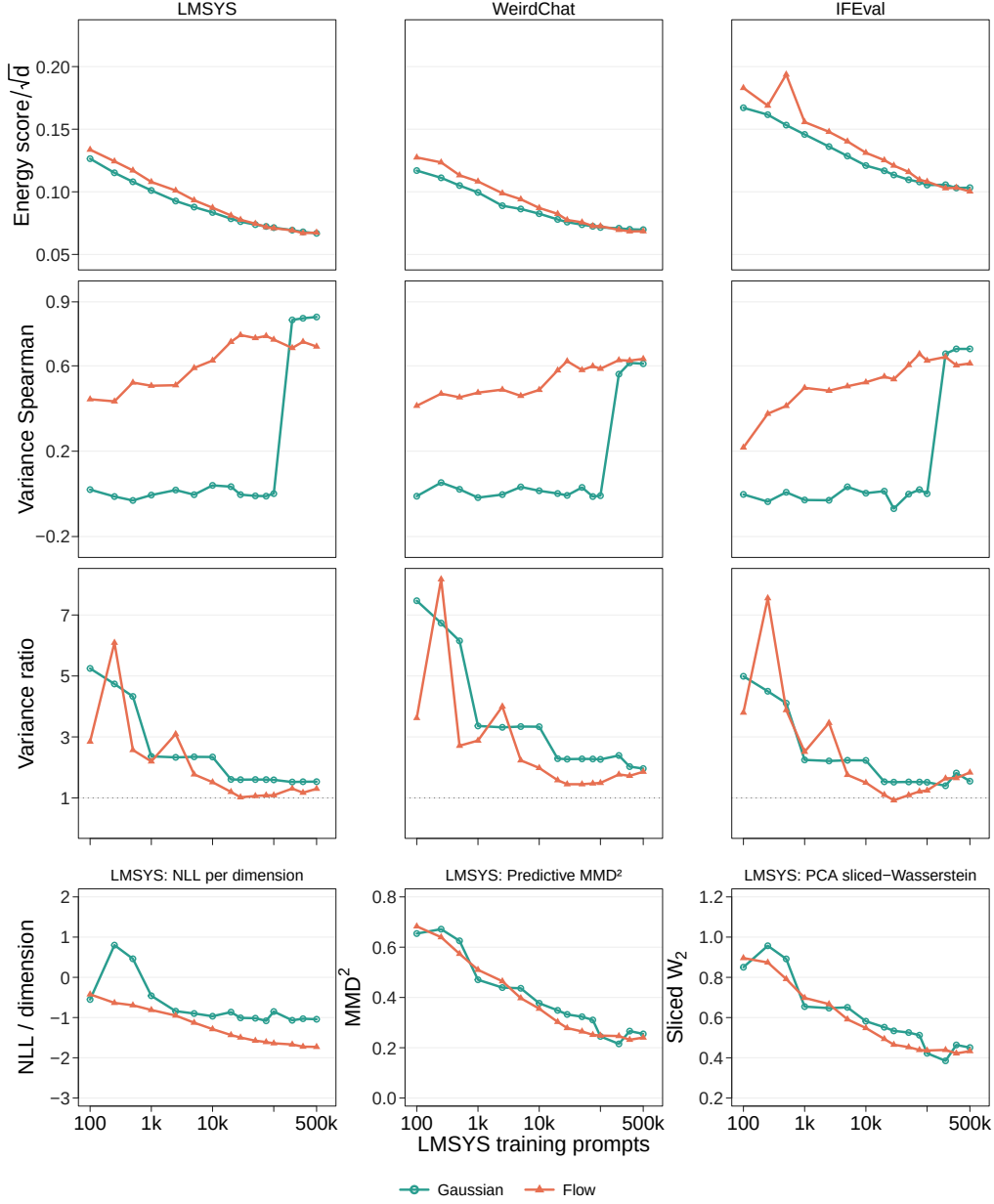


Figure 11: **Distribution quality across the LMSYS training scale, in and out of domain.** The first three rows use evaluation sets as columns and show energy score, prompt-level variance Spearman correlation, and generated-to-real variance ratio for Gaussian and flow forecasts. WeirdChat and IFEval are zero-shot. The final row adds LMSYS NLL, predictive MMD<sup>2</sup>, and PCA sliced-Wasserstein distance. Lower energy, NLL, MMD<sup>2</sup>, and Wasserstein distance are better; higher variance correlation is better; and the ideal variance ratio is one. The flow has lower NLL at 14 of 15 scales (all except 100 prompts) and scores lower on all three auxiliary diagnostics at the largest scale. Gaussian variance ranking stays near zero through 100k because validation energy selects global covariance; at 200k it selects prompt-dependent variance, raising LMSYS Spearman to 0.815. The jump reflects selection between variance modes, not an identified learning threshold.

## F Distribution sampling diagnostics

Figure 12 compares the validation-selected Gaussian forecast with fixed flow particles along a fine-grid Euler path [Lipman et al., 2023, Luo et al., 2026]. The Gaussian panel and every flow panel use the same frozen PCA basis and axis limits. Only the final flow panel shows completed flow samples; the intermediate panels diagnose the transport rather than alternative samplers. In this projection, the Gaussian forecast appears somewhat more dispersed than the completed flow state, consistent with its larger generated-to-observed variance ratio in the full activation space.

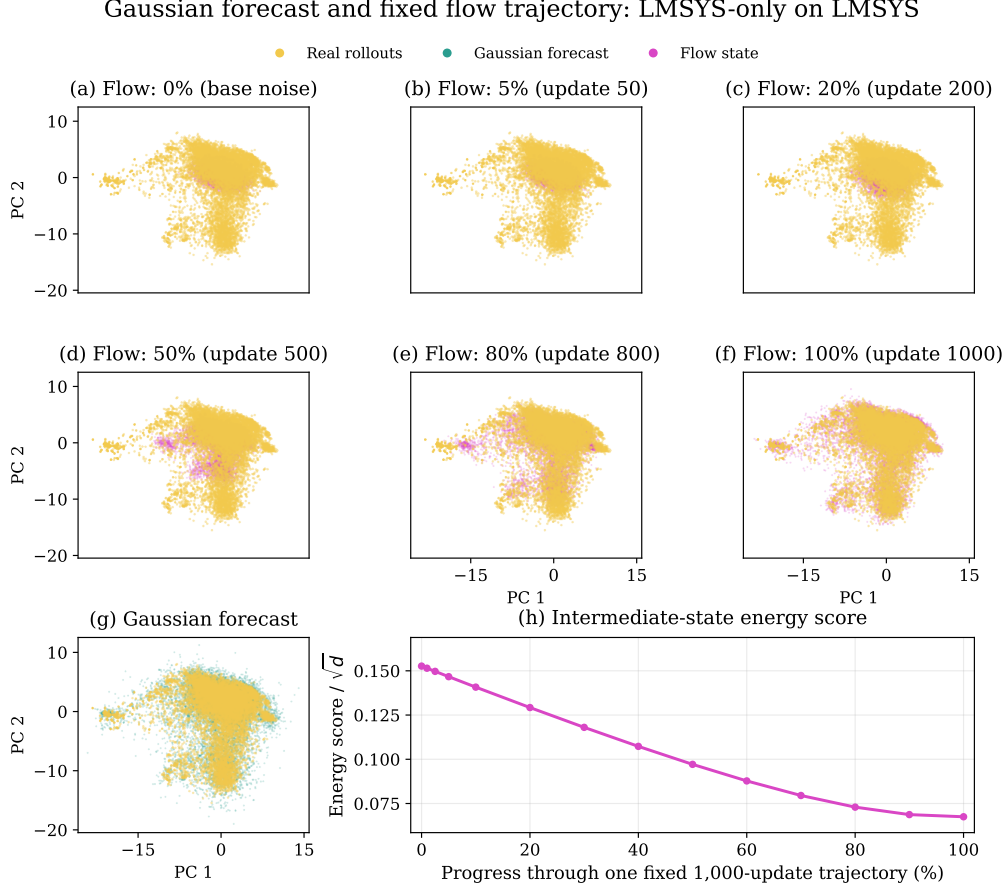


Figure 12: **Gaussian forecast and fixed flow trajectory in a common answer activation plane.** Yellow points are real LMSYS answer activations, green points are samples from the validation-selected Gaussian metamodel, and magenta points are the same LMSYS-only flow particles after 0, 50, 200, 500, 800, and 1,000 Euler updates. The curve shows energy score along the flow trajectory, not sampler quality at different costs. PCA is fitted on 32,768 answer rollouts from 8,192 LMSYS training prompts and then frozen before test targets and forecasts are projected.

We next vary the solver and the number of vector-field evaluations while always reaching the endpoint [Lipman et al., 2023], thereby ablating the number of steps needed. We fix the model, prompts, targets, normalization, and 64 initial samples per prompt. Intervals are paired prompt-bootstrap intervals for absolute energy score.

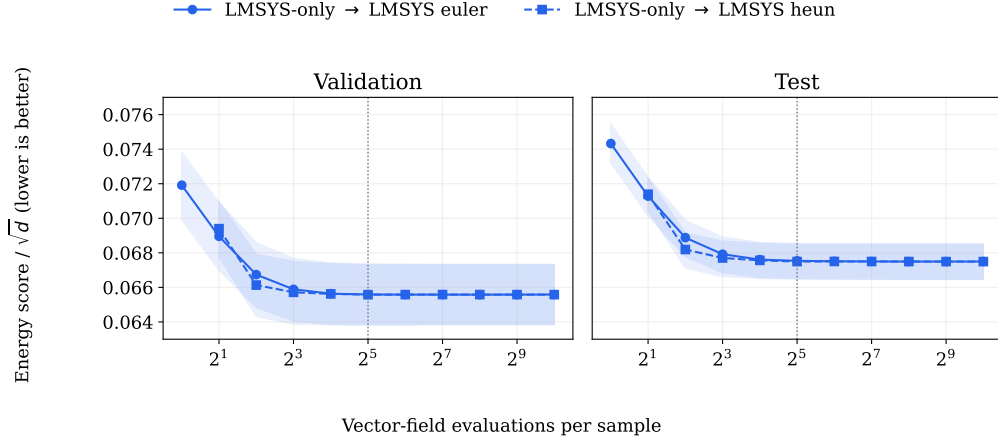


Figure 13: **Completed-sampler quality versus numerical cost.** The vertical axis is absolute energy score per  $\sqrt{d}$ ; lower is better. Shaded bands are 95% prompt-bootstrap confidence intervals. Every point reaches the endpoint. On LMSYS test prompts, 8-step Heun gives energy 0.067563 versus 0.067537 for 32-step Euler, with half the vector-field evaluations. Their paired difference is 0.000026 (95% interval  $[-0.000004, 0.000058]$ ). At high resolution, both approach 0.06749.

## G Cross-model forecasting details

Both targets use linear ridge, MLP, Gaussian, and flow metamodels at eight LMSYS training sizes through 100,000 prompts, with 1,000/3,000 validation/test prompts. Linear ridge uses the final prompt-token activation; the other families use 32-bin traces.

Table 6: Distribution metrics at 100,000 training prompts for both targets and their matched controls. Values are means  $\pm$  standard deviations over 3 rollout partitions for Qwen3.5 and 3 initialization replicates for Gemma 4, with disjoint training/validation and test rollout seeds. Energies are measured in each target model’s activation coordinates.

| Conditioner                   | Forecast | Energy score $\downarrow$ | Variance $\rho \uparrow$ | Variance ratio $\rightarrow 1$ |
|-------------------------------|----------|---------------------------|--------------------------|--------------------------------|
| Qwen2.5 $\rightarrow$ Qwen3.5 | Gaussian | .07849 $\pm$ .00024       | .770 $\pm$ .006          | 2.203 $\pm$ .112               |
| Qwen2.5 $\rightarrow$ Qwen3.5 | Flow     | .07739 $\pm$ .00053       | .713 $\pm$ .011          | 1.266 $\pm$ .076               |
| Qwen3.5 $\rightarrow$ Qwen3.5 | Gaussian | .07174 $\pm$ .00030       | .825 $\pm$ .007          | 1.524 $\pm$ .067               |
| Qwen3.5 $\rightarrow$ Qwen3.5 | Flow     | .07227 $\pm$ .00063       | .756 $\pm$ .016          | 1.143 $\pm$ .107               |
| Qwen2.5 $\rightarrow$ Gemma 4 | Gaussian | .01163 $\pm$ .00003       | −0.019 $\pm$ .020        | 2.404 $\pm$ .016               |
| Qwen2.5 $\rightarrow$ Gemma 4 | Flow     | .01141 $\pm$ .00012       | .341 $\pm$ .032          | 2.775 $\pm$ .463               |
| Gemma 4 $\rightarrow$ Gemma 4 | Gaussian | .01092 $\pm$ .00005       | .010 $\pm$ .026          | 2.407 $\pm$ .014               |
| Gemma 4 $\rightarrow$ Gemma 4 | Flow     | .01070 $\pm$ .00008       | .372 $\pm$ .038          | 2.308 $\pm$ .262               |

### G.1 Qwen3.5 forecasting details

Qwen2.5-7B-Instruct layer 19 ( $d = 3,584$ ) conditions forecasts of Qwen3.5-9B layer 18 ( $d = 4,096$ ); the matched control uses Qwen3.5 layer-18 prompt activations. Each partition uses 8 Qwen3.5 rollout seeds for training and validation and a disjoint set of 8 rollout seeds for test evaluation. For the flow, the conditioner projects from 3,584 to 4,096 dimensions; the vector field and output remain 4,096-dimensional.

Figure 4 shows mean forecasting, Figures 14 and 15 show retrieval and distribution quality, and Table 6 gives the distributional endpoint metrics.

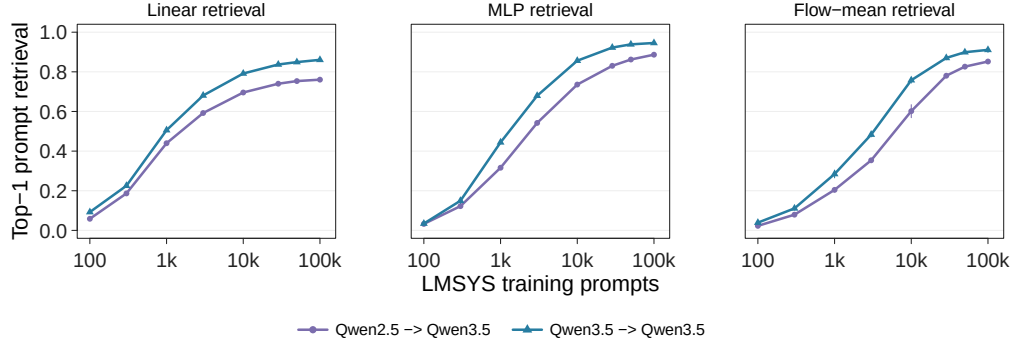


Figure 14: **Qwen3.5 cross-model prompt retrieval across prompt scale.** Top-1 retrieval for linear, MLP, and flow-mean forecasts conditioned on Qwen2.5 or Qwen3.5 prompt activations; both predict Qwen3.5 answer activations. Retrieval uses the same fixed bank of 3,000 LMSYS prompts. Points and error bars are means and standard deviations over 3 rollout partitions.

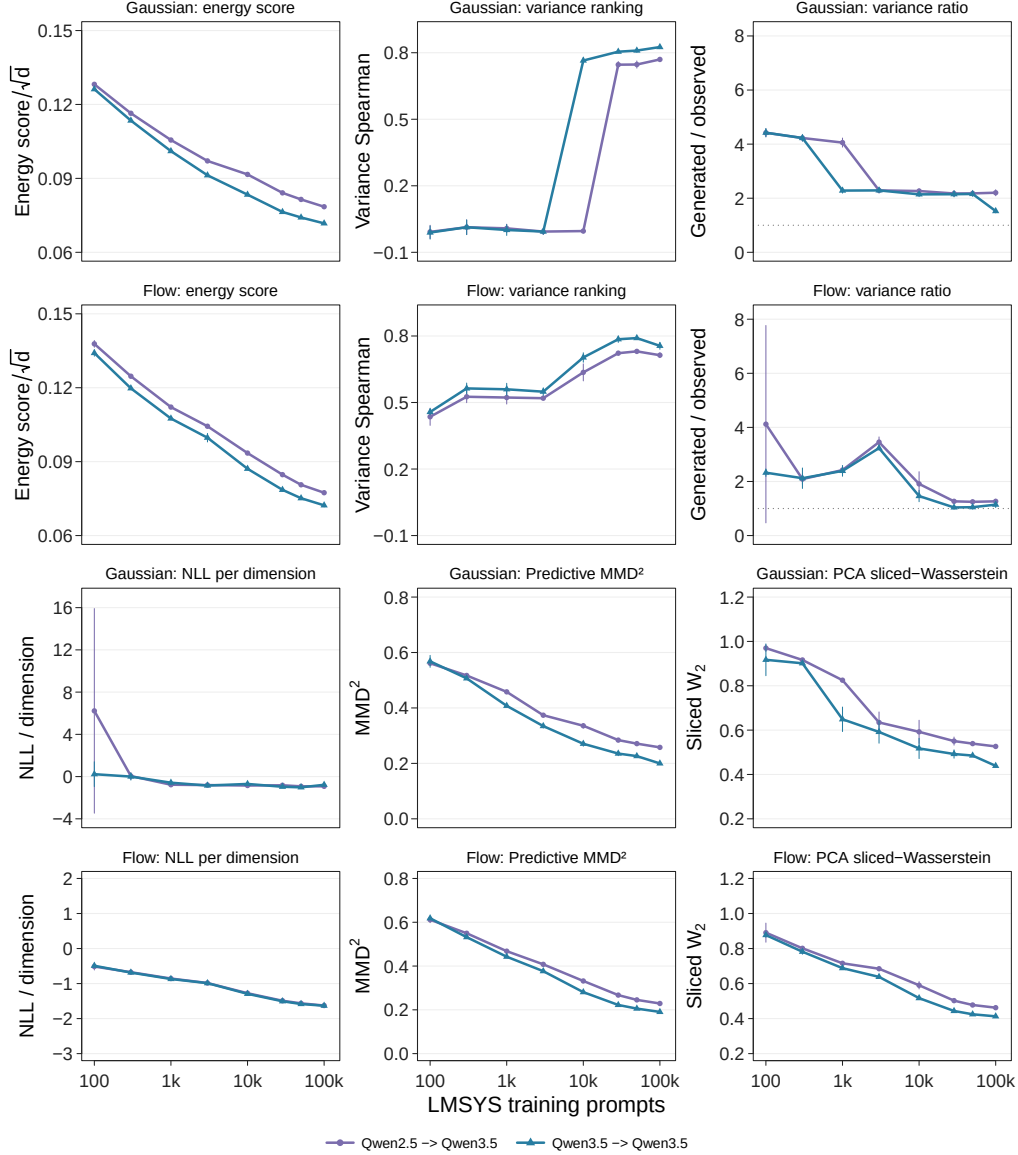


Figure 15: **Qwen3.5 cross-model distribution quality across prompt scale.** The first two rows show separate Gaussian and flow forecasts for energy score, prompt-level variance Spearman correlation, and generated-to-real variance ratio. The last two rows show the same families separately for NLL, predictive MMD<sup>2</sup>, and PCA sliced-Wasserstein distance. Both conditioners predict Qwen3.5 answer activations. Points and error bars are means and standard deviations over 3 rollout partitions. The auxiliary curves use the same completed fits as the primary metrics. Lower energy, NLL, MMD<sup>2</sup>, and Wasserstein distance are better; higher variance correlation is better; and the ideal variance ratio is one.

At the endpoint, Gaussian has lower energy under Qwen3.5 conditioning (0.07174 versus 0.07227), while flow has lower energy under Qwen2.5 conditioning (0.07739 versus 0.07849). It ranks prompt-level spread more strongly in both settings, while the flow better matches aggregate variance. Variance ranking and variance calibration therefore capture distinct properties of a distribution forecast. On the auxiliary diagnostics, the flow has lower NLL at every tested scale and, at the largest scale, lower NLL, MMD<sup>2</sup>, and PCA sliced-Wasserstein distance under both conditioners; its advantage is largest and most consistent for NLL.

## G.2 Gemma 4 forecasting details

Qwen2.5-7B-Instruct layer 19 ( $d = 3,584$ ) conditions forecasts of Gemma-4-12B-it layer 47 ( $d = 3,840$ ); the matched control uses Gemma layer-47 prompt activations. Training and validation use four rollout seeds (43–46), and test uses four disjoint seeds (47–50).

Figure 4 shows mean forecasting, Figures 16 and 17 show retrieval and distribution quality, and Table 7 gives the endpoint metrics. Gaussian selects global variance at every tested scale under both conditioners.

Despite accurate mean forecasts, variance capture remains poor for Gemma 4 under both conditioners. At 100,000 training prompts, Gaussian variance ranking is near zero and flow’s is modest, while both generate more than twice the observed average within-prompt variance (Table 7). We leave further investigation of this limitation to future work.

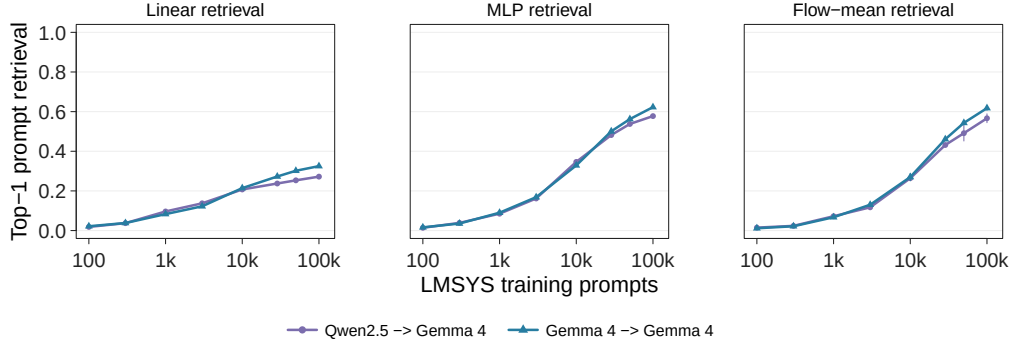


Figure 16: **Gemma 4 cross-model prompt retrieval across prompt scale.** Top-1 retrieval for linear, MLP, and flow-mean forecasts conditioned on Qwen2.5 or Gemma 4 prompt activations; both predict Gemma 4 answer activations. Retrieval uses a fixed bank of 3,000 LMSYS prompts. MLP and flow points and error bars are means and standard deviations over three initialization replicates.

Table 7: Gemma 4 target endpoints at 100,000 training prompts. Reported uncertainties are one standard deviation across three initialization replicates. Gaussian’s analytic mean is the MLP mean; its duplicate  $R^2$  is omitted.

| Conditioner | Forecast | Mean $R^2$        | Energy $\downarrow$   | Variance $\rho$    | Variance ratio    |
|-------------|----------|-------------------|-----------------------|--------------------|-------------------|
| Qwen2.5     | Linear   | 0.752             | –                     | –                  | –                 |
| Qwen2.5     | MLP      | $0.819 \pm 0.002$ | –                     | –                  | –                 |
| Qwen2.5     | Gaussian | –                 | $0.01163 \pm 0.00003$ | $-0.019 \pm 0.020$ | $2.404 \pm 0.016$ |
| Qwen2.5     | Flow     | $0.800 \pm 0.006$ | $0.01141 \pm 0.00012$ | $0.341 \pm 0.032$  | $2.775 \pm 0.463$ |
| Gemma 4     | Linear   | 0.771             | –                     | –                  | –                 |
| Gemma 4     | MLP      | $0.846 \pm 0.002$ | –                     | –                  | –                 |
| Gemma 4     | Gaussian | –                 | $0.01092 \pm 0.00005$ | $0.010 \pm 0.026$  | $2.407 \pm 0.014$ |
| Gemma 4     | Flow     | $0.829 \pm 0.009$ | $0.01070 \pm 0.00008$ | $0.372 \pm 0.038$  | $2.308 \pm 0.262$ |

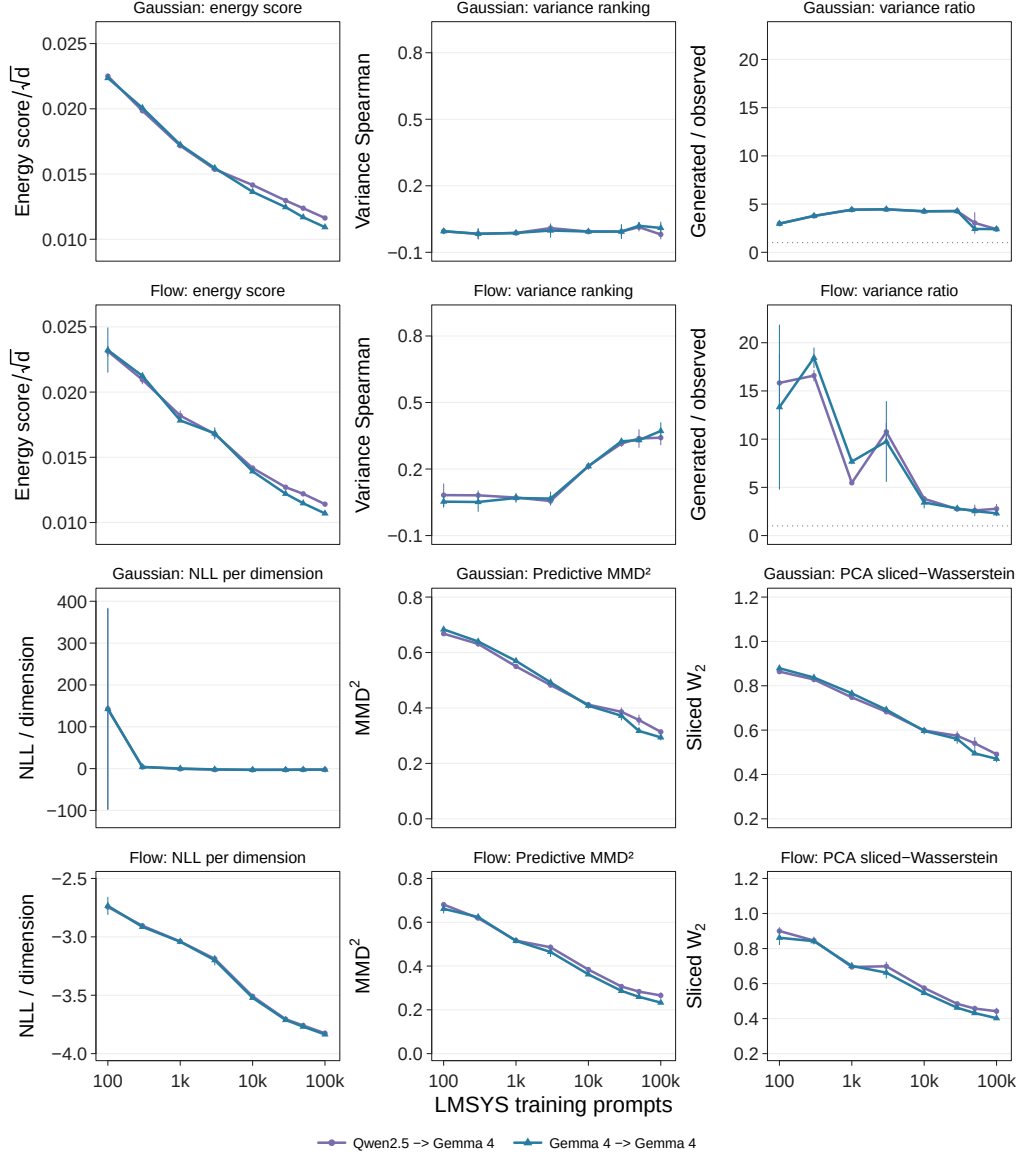


Figure 17: **Gemma 4 cross-model distribution quality across prompt scale.** The first two rows show Gaussian and flow energy score, prompt-level variance Spearman correlation, and generated-to-real variance ratio. The last two rows show NLL, predictive MMD<sup>2</sup>, and PCA sliced-Wasserstein distance. Both conditioners predict Gemma 4 answer activations. Points and error bars are means and standard deviations over three initialization replicates. The Gaussian NLL axis includes the large variability at 100 training prompts. Lower energy, NLL, MMD<sup>2</sup>, and Wasserstein distance are better; higher variance correlation is better; and the ideal variance ratio is one.

## H Behavior-probe AUROC

The probe experiment uses mixed metamodels trained on 500,000 LMSYS prompts and the WeirdChat training split with equal domain weighting. WeirdChat uses 1,596/532/532 train/validation/test prompts and four rollouts per prompt. GPT-6 Luna labels the answers. Gaussian and flow predictions average readout probabilities over all 1,024 samples. AUPRC and AUROC use the same held-out groups and readouts, selected by validation log loss.

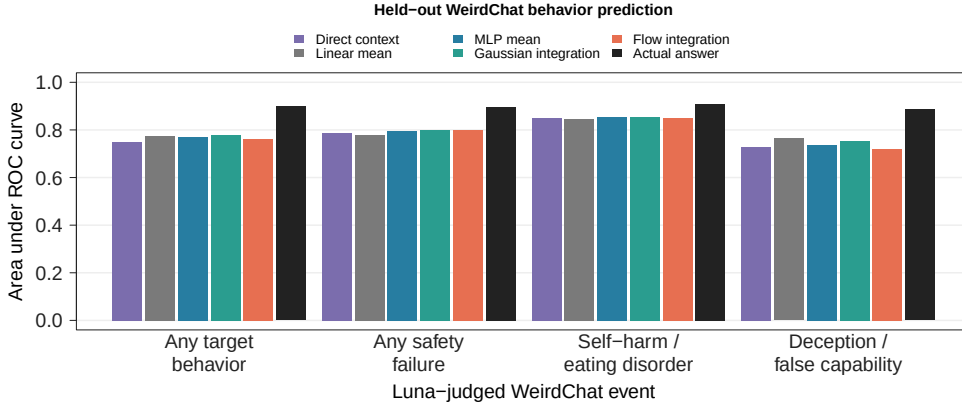


Figure 18: **AUROC of held-out WeirdChat behavior predictions.** Higher is better. The actual-answer reference observes the generated answer.

At least one forecast exceeds direct context in every group. Among the prompt-only methods, Gaussian leads for any target behavior, flow for safety failures, MLP for self-harm or eating disorder, and linear for deception or false capability. Together with AUPRC, these results show that forecasts retain behavior-predictive information, while integrating a distribution does not consistently improve on the MLP mean.