Robustness of In-context Learning via Curriculum Learning

Srikar Babu Gadipudi*
University of California, Berkeley

Neil Pattanaik*
University of California, Berkeley

Noah Lund Syrdal*
University of California, Berkeley

Anders Vestrum*
University of California, Berkeley

Abstract

In-context learning (ICL) enables models to perform tasks by conditioning on input-output demonstrations without parameter updates, but the robustness to noisy demonstrations is unclear. We investigate whether curriculum learning principles can improve ICL robustness by training Transformers with progressively increasing noise levels during pretraining. Across controlled synthetic regression tasks involving linear functions and 2-layer ReLU neural networks, we find that noise injection during training improves robustness compared to training without noise. However, curriculum-based noise schedules provide no consistent advantage over random noise injection, with both approaches across diverse evaluation conditions including distribution shifts. We extend our investigation to text classification tasks (SST-2, MNLI, PAWS) and text generation tasks (SQuAD 2.0) using GPT-Neo 2.7B and Llama-2-7b models, respectively, where we corrupt demonstrations through token masking. Our results show that introducing noise into demonstrations can improve model performance when queries are similarly corrupted in the case of text classification, while curriculum and uniform masking strategies behave similarly. But observe a decay in performance with the text generation task. Taken together, these findings support our first hypothesis (H1) that exposure to noisy demonstrations improves ICL robustness, but do not support our second hypothesis (H2) that curriculum-style noise schedules provide additional gains over randomly sampled noise in the settings we study [1].

1 Introduction

Transformers trained on next-token objective tasks display an impressive ability to perform tasks purely through conditioning on input-output examples, a technique known as in-context learning (ICL) (Garg et al. [2022]). Since the emergence of early generative models such as GPT-2 models (Radford et al. [2019]) and its dramatic expansion in large-scale language models (Brown et al. [2020], Chowdhery et al. [2023], Touvron et al. [2023]), understanding the mechanisms and limitations of ICL has become an important question. A growing body of work investigates how Transformers acquire ICL capabilities, including theoretical analyses that view ICL as implicit gradient descent (Von Oswald et al. [2023], Akyürek et al. [2022]), emergent algorithmic behaviour (Olsson et al. [2022]) and meta-learning phenomena arising from pretraining (Dai et al. [2023]). Despite this rapid conceptual progress, little is explored on techniques for improving the robustness of ICL when the provided demonstrations are imperfect or corrupted.

*Equal contribution.

[1] Code available at: https://github.com/neilpattanaik/in-context-learning-1

In practical applications, demonstrations provided to model can contain noise through label corruption, missing features, token masking, adversarial edits or errors accumulated through automated data collection pipelines. Classical supervised learning literature suggests that injecting noise during training can lead to improved robustness under certain conditions (Natarajan et al. [2013], Rolnick et al. [2017], Song et al. [2022]). This naturally raises the question of whether similar principles hold for Transformers trained to perform ICL. Cheng et al. [2025] explore robustness of ICL but do not propose ways to enhance it. As a result, there remains little systematic exploration of techniques aimed at actively improving the robustness of ICL and it is largely unknown whether training-time interventions such as structured noise schedules or curriculum-style perturbations can reliably strengthen a model's resistance to noisy in-context examples.

In this work, we explore whether curriculum learning principles (Bengio et al. [2009]) can help Transformers develop more stable ICL behaviour. Curriculum learning has demonstrated broad benefits in supervised learning, reinforcement learning and structured prediction by arranging training data from easier to harder instances (Graves et al. [2017]). Recent work has begun applying curriculum ideas to ICl without fine-tuning by ordering demonstrations from simple to complex inside the prompt (Liu et al.), but training-time noise curricula for Transformers intended to perform ICL have not been systematically studied.

Our work begins with controlled synthetic regression tasks that allow us to evaluate ICL behaviour precisely. We follow the framework of Garg et al. [2022] and train standard GPT-2 model from scratch on three increasingly complex function classes: linear regression and two-layer neural networks. During training, we introduce additive Gaussian noise with a curriculum schedule where the noise standard deviation increases gradually from zero to one. We compare this with both a no-noise baseline (without addition of any noise during training) and a random baseline (addition of different levels of noise independent of number of training steps). Evaluation is performed on unseen prompts and includes multiple types of corruption such as Gaussian, Laplacian and Uniform noise. This verifies whether ICL is robust to change in noise distribution from training and evaluation as well. Results are measured using squared loss as the number of in-context examples increases.

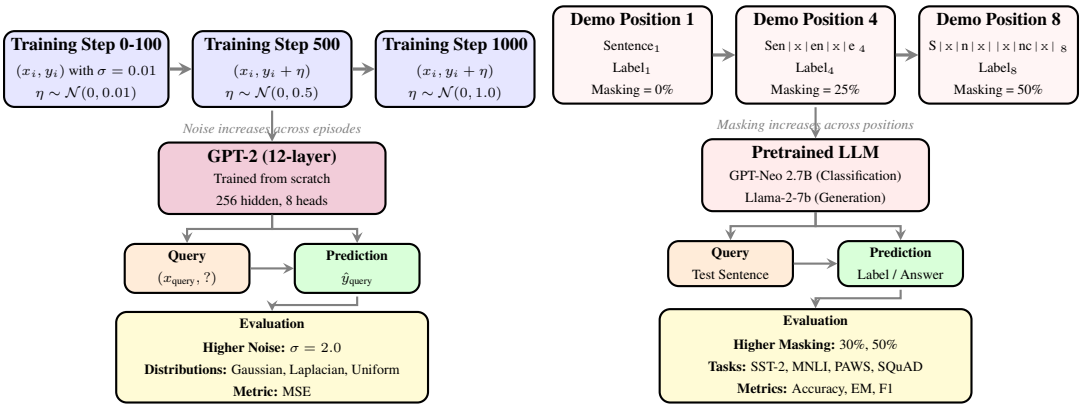
Across the synthetic tasks, we find that curriculum noise does improve robustness compared to training with no-noise, but not a great improvement from adding noise randomly during training. On the linear regression task, curriculum noise and random noise injection exhibit similar robustness profiles. On the more complex two-layer network task, we observe inconclusive results with no training regime demonstrating consistent superiority. In addition to controlled synthetic regression experiments, we evaluate curriculum-style corruption on pretrained language models for both discriminative and generative NLP benchmarks. For classification, we measure how token-level corruption of demonstrations affects GPT-Neo 2.7B on SST-2, MNLI and PAWS. For generation, we probe Llama-2-7B on SQuAD-style QA using exact-match and F1 metrics when demonstration contexts are progressively masked. These complementary settings allow us to contrast algorithmic ICL behaviour learned in scratch-trained Transformers with prompt-based ICL in large pretrained models and investigate whether curriculum-style corruption benefits transfer across these domains.

Our results highlight that curriculum noise does not universally enhance robustness in ICL for simple function classes and motivate a deeper investigation into when transformers can benefit from structured corruption during training or prompting. The observations suggest a nuanced picture in which the effect of curriculum learning may depend strongly on task complexity, representation requirements and the distribution of evaluation noise. This study is guided by two main hypotheses about how noise-based interventions affect the robustness of ICL:

H1 Training Transformers with noisy demonstrations improves ICL robustness to corrupted in-context examples compared to training on clean-only data.

H2 For a fixed noise budget, curriculum-style noise schedules that gradually increase the noise level over training lead to greater robustness than randomly sampling noise levels throughout training.

Across both our synthetic regression experiments and text/NLP benchmarks, we find support for H1: models exposed to noisy demonstrations are more robust to noisy evaluation prompts than models trained without noise. In contrast, we do not find evidence supporting H2 in the settings we study. Curriculum-based noise text schedules perform similarly to, but not systematically better than, random noise injection. Additionally, we encounter inconclusive results with the text generation task.



(a) Synthetic Regression Tasks (b) NLP Tasks

Figure 1: Methodology overview for investigating ICL robustness via curriculum learning. (a) Synthetic regression tasks: Models are trained from scratch with curriculum noise that gradually increases over training steps (temporal curriculum). (b) NLP tasks: Pretrained models are evaluated with curriculum masking that progressively increases across demonstration positions within the context (spatial curriculum). Both approaches test whether structured noise schedules improve robustness compared to random or no-noise baselines.

2 Related Work

Research on ICL has grown rapidly following the discovery that Transformers exhibit strong few-shot abilities without parameter updates (Brown et al. [2020]). Several works attempts to characterize the emergence of ICL. Garg et al. [2022] demonstrate that Transformers trained on synthetic tasks can approximate gradient descent updates inside their forward pass. Akyürek et al. [2022] and Von Oswald et al. [2023] show that Transformers can implement implicit learning algorithms such as least-squares regression or kernel methods. Other works analyze the statistics of demonstrations that matter for ICL success (Min et al. [2022]) and the generalization of ICL beyond the training distribution (Wang et al. [2024]). Surveys such as Dong et al. [2024] provide an overview of these findings with broader perspectives on prompt design and large language model (LLM) reasoning.

Noise and robustness in ICL remain less well understood. Agarwal et al. [2024] examine failures of ICL with mislabeled demonstrations. Cheng et al. [2025] explore training Transformers with noisy labels to improve robustness under noisy test conditions and report task-dependent effects. Zhang et al. [2024] study noise robustness for text generation and argue that some perturbations do not significantly degrade performance, although their conclusions do not directly extend to regression-style ICL. In the same light, Gao et al. [2024] propose a Local Perplexity Ranking mechanism to improve noise robustness of ICL with similar text generation tasks. Our work relates to these efforts but differs by systematically evaluating curriculum-based noise schedules on multiple function classes with controlled data generation.

Curriculum learning, introduced by Bengio et al. [2009], arranges training examples to gradually increase difficulty. This idea has been applied to neural networks (Weinshall et al. [2018]), metric learning (Wu et al. [2020]), reinforcement learning, multimodal learning and vision-language models (Soviany et al. [2022]). Applications to ICL are still emerging. Liu et al. propose arranging demonstrations within the prompt from easy to hard, showing improvements without fine-tuning. Our work approaches curriculum design from a different angle by modifying the training noise distribution rather than the ordering of demonstrations. The empirical results indicate that curriculum noise does not guarantee improved robustness in simple function classes, highlighting an important gap between traditional curriculum learning benefits and the inductive biases governing ICL.

Finally, learning under noisy labels has been extensively studied in classical supervised learning (Natarajan et al. [2013], Song et al. [2022]). Although the underlying motivations are related to ours, the mechanisms differ because Transformers performing ICL must infer an internal update rule rather than directly optimize a supervised objective. This distinction makes robustness in ICL

both more subtle and less predictable, reinforcing the need for systematic evaluations such as those presented in this work.

3 Background

ICL refers to a model's ability to infer a task from a sequence of input–output demonstrations without parameter updates (Garg et al. [2022]). Given a prompt

$$\mathcal{P} = \big((x_1, y_1), \ldots, (x_n, y_n), (x_{\text{query}}, \cdot)\big),$$

a Transformer M_θ predicts the query label via

$$\hat{y}_{\text{query}} = M_\theta(\mathcal{P}).$$

Loss formulation. Our synthetic regression tasks follow the episodic setup of (Garg et al. [2022]). For each episode containing n demonstrations and one query, the model is trained using squared error on the final query output. The overall training objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{P}\sim\mathcal{D}}\left[\ell\big(M_\theta(\mathcal{P}), y_{\text{query}}\big)\right] \quad \text{with} \quad \ell\big(M_\theta(\mathcal{P}), y_{\text{query}}\big) = \big(M_\theta(\mathcal{P}) - y_{\text{query}}\big)^2. \tag{1}$$

Although the loss is applied only to the query token, minimizing this objective requires the model to aggregate the demonstrations within the episode into an internal estimate of the underlying function, reproducing the implicit ICL behaviour observed in prior work.

Curriculum noise. Curriculum learning (Bengio et al. [2009]) increases task difficulty gradually during training. In our synthetic regression settings, difficulty corresponds to the level of output noise added to the demonstration labels within each episode. Early in training the demonstrations are effectively clean and the noise level is increased over time according to a fixed schedule. Under the episodic loss in Eq. (2), this encourages the model to first learn a stable in-context update rule from clean examples and then adjust that rule to handle increasingly corrupted ones. Prior work suggests that exposure to noisy demonstrations can act as a form of data augmentation (Cheng et al. [2025]) and our curriculum aims to test whether a structured noise schedule provides more robust ICL than random or uniformly applied noise.

ICL in text classification. For natural-language tasks (SST-2, MNLI, PAWS), we only modify the demonstrations inside the prompt. We introduce token masking as a form of corruption and include a simple curriculum version where masking increases across examples. This mirrors the synthetic curriculum setup and allows us to test whether structured prompt corruption affects robustness in pretrained ICL, taking inspiration from the ordering of demonstrations effecting ICL performance with LLMs (Liu et al.).

ICL in text generation. While our synthetic and classification settings focus on predicting low-dimensional targets or discrete labels, many practical applications of ICL involve open-ended text generation. Recent work finds that text–generation ICL can be particularly brittle under noisy demonstrations (Zhang et al. [2024], Gao et al. [2024]). This motivates including a generative setting in our robustness study: by applying masking-based corruption schedules to question–answer demonstrations in SQuAD-style prompts, we can test whether the curriculum-style noise that helps in regression and, to a lesser extent, classification also stabilizes autoregressive generation, or whether generative ICL exhibits qualitatively different robustness behaviour.

4 Methodology

Our methodological goal, Figure 1, is to construct a sequence of controlled regression environments that allow us to study how transformers acquire ICL as the task's noise level varies. All environments share a common prompting interface of the model receives a sequence of input–output pairs and must infer the output for a held-out query input. We examine two synthetic tasks that form an increasing hierarchy of functional complexity. The first is a linear regression task, which serves as a classical baseline in ICL research. The second task is nonlinear regression using randomly sampled 2-layer ReLU functions (also extended to 6-layer ReLU functions). Together, these tasks allow us to probe whether mechanisms learned in simple settings transfer to more complex ones and how noise affects in-context generalization.

All experiments use the same Transformer architecture and training protocol. We employ a 12-layer GPT-2 model with 256 hidden size, 8 attention heads and learned linear input/output projections.

$$[\, x_1,\ y_1,\ x_2,\ y_2,\ \ldots,\ x_n,\ y_n,\ x_{\text{query}}\,].$$

Training uses the AdamW optimizer with fixed hyperparameters, constant learning rate and mean-squared error loss from Eq. 1. A complete specification appears in Appendix A. Because architecture, optimization and input formatting are held constant across settings, observed behavioural differences directly reflect the influence of task complexity on the emergence of ICL.

Linear Regression. We begin with a linear regression task, with 20-dimensional input and a scalar output. Inputs are continuous vectors sampled from a Gaussian distribution and outputs are generated by a linear map,

$$f(x) = w^\top x, \qquad w \sim \mathcal{N}(0, I_d).$$

This task mirrors the setup used in recent work demonstrating that transformers can approximate least-squares solutions in context. Prompts consist of pairs $(x_i, f(x_i))$ followed by a query input x_{query}. Because the optimal regression rule is analytically known, this environment serves as a diagnostic for whether the model recovers a canonical algorithm and whether noise disrupts this behaviour. Noise is applied to the example outputs in order to explore curriculum schedules.

2-layer ReLU Neural Networks. The second task increases functional complexity by replacing algebraic structure with a nonlinear, high-dimensional mapping. Inputs are continuous vectors $x \in \mathbb{R}^{20}$ sampled from a standard Gaussian distribution. Outputs are generated by a randomly sampled two-layer ReLU network,

$$f(x) = \sqrt{\frac{2}{h}}\, W_2^\top \sigma(W_1^\top x), \qquad \sigma(z) = \max(0, z),$$

where $W_1 \in \mathbb{R}^{20\times h}$ and $W_2 \in \mathbb{R}^{h\times 1}$ are sampled independently from $\mathcal{N}(0,1)$ at the start of each episode and h is the hidden-layer size ($h = 100$ in our experiments). No bias terms are used. Because both layers are resampled each episode, the target function changes completely from episode to episode, forcing the model to reconstruct a new nonlinear rule purely from the prompt.

Noise is added only to the outputs of the in-context examples, using the same random and curriculum schedules as in the linear setting. This environment tests whether in-context strategies learned in simpler tasks transfer to settings without an explicit closed-form structure.

Noise Models and Curriculum Schedules. To evaluate robustness, we introduce controlled corruption only to the outputs of the in-context examples. For each clean pair $(x_i, f(x_i))$, we form the noisy observation

$$\tilde{y}_i = f(x_i) + \eta_i, \qquad \eta_i \sim \mathcal{D}(0, \sigma), \tag{2}$$

where $\mathcal{D}$ denotes a zero-mean noise distribution. During training, $\mathcal{D}$ is restricted to a Gaussian distribution and the query input is always kept clean. Although our implementation supports corruption on both inputs and outputs, we found that input perturbations produced qualitatively identical effects across all three environments—matching the magnitude and pattern of the changes induced by output noise. To keep the analysis focused and avoid redundant conditions, the final regression experiments therefore use output noise exclusively.

Noise is introduced under two training regimes. In the *random-noise* regime, the noise level σ is re-sampled independently at each training step from a fixed uniform range, yielding an unstructured corruption schedule. In contrast, the *curriculum-noise* regime increases the corruption level monotonically over training according to

$$\sigma_x = \sigma_{\min} + \left\lfloor \frac{t}{T_{\text{interval}}} \right\rfloor \Delta\sigma, \qquad \sigma_{\min} = 0, \qquad \Delta\sigma = 0.1, \tag{3}$$

so that the model encounters progressively more challenging regression episodes as training proceeds.

For evaluation, we apply substantially stronger corruption than during training ($\sigma = 2.0$) and introduce distributional shifts in the corruption process. Specifically, although the model is trained using Gaussian noise alone, we evaluate it under Gaussian, Laplacian and Uniform noise distributions. This design tests whether the model acquires genuine noise-aware inference strategies rather

than merely adapting to the specific variance or distributional form of the training noise and it allows us to assess how well curriculum-based training transfers to qualitatively different corruption environments.

Text Classification with Masking Noise. We also conduct a training-free parallel exploration of ICL applied to pretrained LLMs for natural language classification tasks. Formally, we consider a sequence of k demonstration examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$ and a test query x_{test}, where the x inputs are short text blurbs and the y values belong to a small set of discrete labels. Given a language model $\mathcal{M}$, the ICL prediction is $\hat{y} = \mathcal{M}(\text{prompt}(x_1, y_1, \ldots, x_k, y_k, x_{\text{test}}))$. In our experiment, each prompt consists of 8 demonstration pairs. All text classification tasks use the GPT-Neo 2.7B model to maintain consistency with prior benchmarks.

To study robustness, we progressively degrade the demonstration inputs directly. For a tokenized demonstration $x = (w_1, \ldots, w_L)$, we uniformly sample an ε-fraction of token positions to replace with the tokenizer's mask token <MASK>. Formally, masking is implemented as

$$\tilde{x}_j = \begin{cases} \texttt{<MASK>}, & \text{with probability } \varepsilon, \\ w_j, & \text{with probability } 1-\varepsilon, \end{cases} \qquad j = 1, \ldots, L, \tag{4}$$

so that $\mathbb{E}\left[\sum_{j=1}^{L} \mathbf{1}\{\tilde{x}_j = \texttt{<MASK>}\}\right] = \varepsilon L$. This masking preserves the sequence structure while removing some of the semantic content, serving as a natural-language analogue of the noise used in the regression tasks. During evaluation, we use heavier masking levels than in training. This setup provides an independent check on whether robustness learned in synthetic environments transfers to natural-language inference.

Text Generation with Masking Noise. To evaluate performance on generative tasks, we utilize the SQuAD 2.0 dataset Rajpurkar et al. [2018]. We employ a 4-shot in-context learning setup ($k = 4$) where we introduce noise (demonstration masking, similar to the classification task) into the demonstration inputs while keeping the answers clean. We compare two noise schedules random, sampling a noise probability $p \sim U[0, p_{max}]$ and mask tokens uniformly across all demonstrations, and curriculum, applying a linear increase in noise intensity, masking the i-th demonstration with probability $p_i = p_{max} \cdot \frac{i}{3}$.

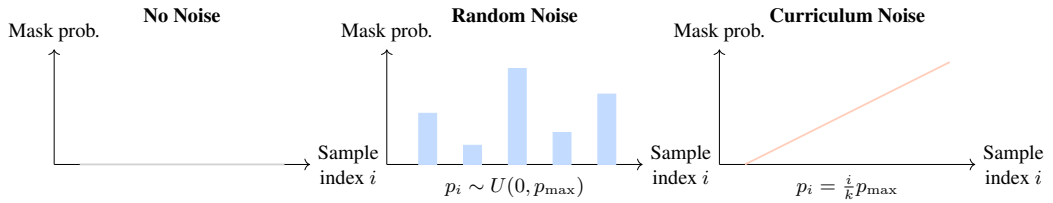
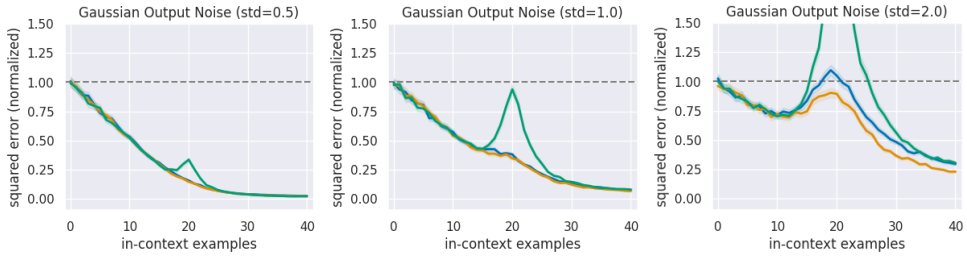


Figure 2: Noise schedules applied to demonstration inputs for each regime (classification and generation tasks).

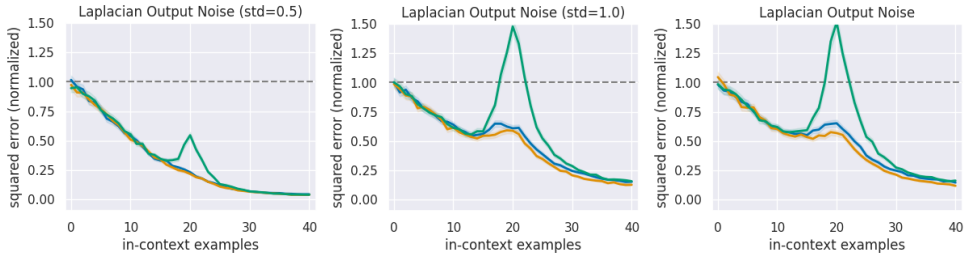
We hypothesize that the curriculum schedule will guide the model to better handle noise. We evaluate this by measuring Exact Match (EM) anf F1 scores at varying noise caps ($p_{max} \in \{0.1, 0.3, 0.5\}$).

5 Experiments

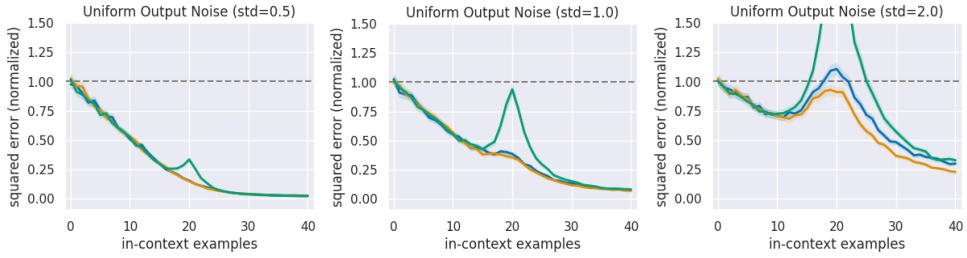
We evaluate our approach across two settings: controlled synthetic tasks and natural language classification tasks. For the synthetic tasks, we train GPT-2 models from scratch under three noise regimes: no noise during training, random noise applied uniformly throughout training and curriculum noise that gradually increases from clean to noisy demonstrations. For text classification, we apply token masking to demonstrations in pretrained language models to assess whether structured corruption during prompting affects robustness.



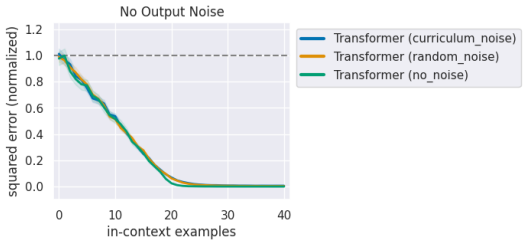
(a) Gaussian noise $\sigma = 0.5$ (b) Gaussian noise $\sigma = 1.0$ (c) Gaussian noise $\sigma = 2.0$



(d) Laplacian noise $B = 0.5$ (e) Laplacian noise $B = 0.5$ (f) Laplacian noise $B = 0.5$



(g) Uniform noise $[0, 0.5]$ (h) Uniform noise $[0, 1]$ (i) Uniform noise $[0, 2]$



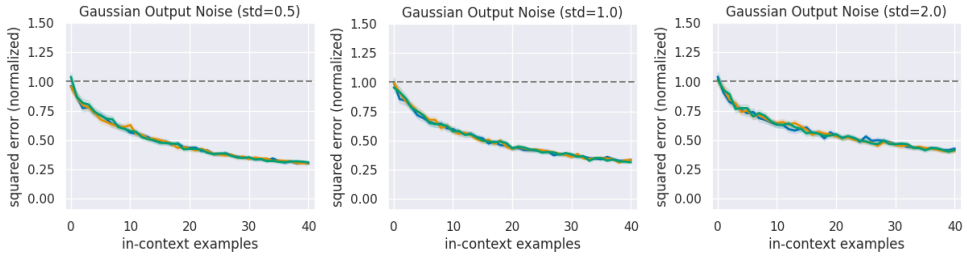
(j) No noise

Figure 3: ICL performance with different noise training strategies under different evaluation noise perturbations for **Linear Regression Task**. Squared error (normalized) versus number of in-context examples for transformers trained with curriculum noise (blue), random noise (orange), or no noise (green). Results averaged over 1,280 context pairs; standard deviations are plotted (negligible/invisible to naked eye).

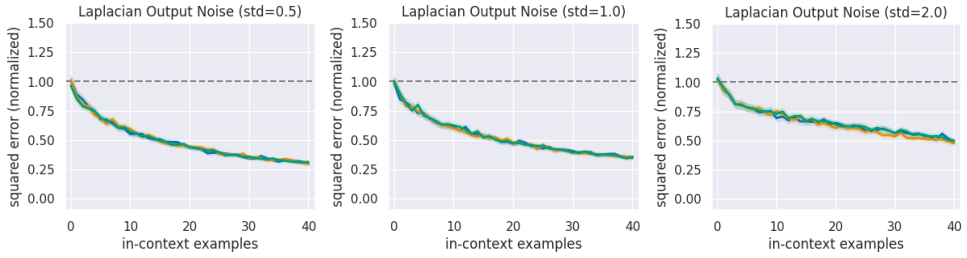
5.1 Linear Regression

Figure 3 presents our results for the linear regression task across multiple noise distributions and magnitudes. When demonstrations are corrupted during evaluation, models trained with noise consistently outperform models trained without noise. This holds across Gaussian, Laplacian and Uniform noise distributions and across varying noise intensities from mild to severe corruption.

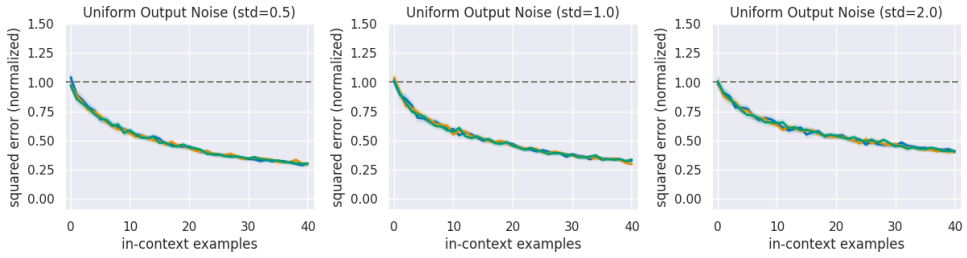
However, we observe no meaningful difference between curriculum noise and random noise training regimes. Both approaches yield nearly identical performance curves across all evaluation conditions. The learning curves converge at similar rates as the number of in-context examples increases and



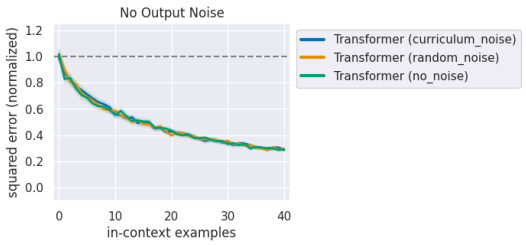
(a) Gaussian noise $\sigma = 0.5$ (b) Gaussian noise $\sigma = 1.0$ (c) Gaussian noise $\sigma = 2.0$



(d) Laplacian noise $B = 0.5$ (e) Laplacian noise $B = 0.5$ (f) Laplacian noise $B = 0.5$



(g) Uniform noise $[0, 0.5]$ (h) Uniform noise $[0, 1]$ (i) Uniform noise $[0, 2]$



(j) No noise

Figure 4: ICL performance with different noise training strategies under different evaluation noise perturbations for **2-layer ReLU Neural Network**. Squared error (normalized) versus number of in-context examples for transformers trained with curriculum noise (blue), random noise (orange), or no noise (green). Results averaged over 1,280 context pairs; standard deviations are plotted (negligible/invisible to naked eye).

the final prediction errors are almost indistinguishable. This suggests that for linear regression, the presence of noise during training matters substantially, but the temporal structure of that noise does not provide additional benefit.

A second pattern in Figure 3 is a bump in squared error that appears around roughly 20 in-context examples. This is consistent with noisy least-squares behaviour and the view that the model is implementing an ordinary least-squares (OLS) solution in context (Garg et al. [2022]). With a small number of demonstrations, the implicit OLS estimate benefits from additional samples and error decreases, but as more noisy labels are added, the estimator temporarily worsens because it aggregates more corruption, producing the mid-context spike. For larger context lengths, averaging

effects dominate, the influence of any single noisy label is diluted, and the squared error decreases again.

The robustness gains are particularly evident under strong distribution shift. When we evaluate models trained on Gaussian noise using Laplacian or Uniform noise at test time, both noise-trained models maintain stable performance while the no-noise baseline degrades significantly. This indicates that exposure to noisy demonstrations during training enables the model to develop inference strategies that generalize beyond the specific corruption distribution encountered during training.

5.2 2-layer ReLU Neural Networks

Figure 4 shows results for the more complex two-layer neural network regression task. Unlike the linear case, we do not observe clear or consistent patterns across training regimes. While noise-trained models occasionally exhibit lower prediction error under specific evaluation conditions, these advantages are not systematic and do not hold uniformly across noise types or magnitudes.

In several evaluation settings, particularly under moderate Gaussian, Laplacian and Uniform noise, the three training strategies produce overlapping performance curves with high variance. Under severe corruption, such as Gaussian noise with a standard deviation of 2.0, all three approaches struggle similarly and no training regime demonstrates reliable superiority. The curriculum noise schedule shows marginal improvements in isolated cases but fails to provide consistent benefits across the full range of evaluation conditions.

These inconclusive results suggest that the relationship between training noise and robustness depends critically on task complexity. The mechanisms that enable noise robustness in linear settings may not transfer straightforwardly to nonlinear function classes where the model must discover more complex internal update rules. The increased representational demands of the two-layer network task may overwhelm any advantages conferred by structured noise schedules. An extension to 6-layer ReLU Neural Networks is presented in Appendix B.

5.3 Text Classification with Demonstration Masking

Config	MNLI	PAWS	SST2
No noise	39.9%	54.6%	65.3%
Curriculum 0% - 50% - 50%	38.5%	52.5%	73.7%
Curriculum 1% - 50% - 50%	39.5%	53.5%	73.9%
Equal noise 50%	39.5%	55.2%	73.6%

Table 1: Performance (in accuracy percentage) of GPT-Neo 2.7B on text classification with varying noise levels in demonstration prompts (standard deviation omitted from the table owing to its very small value, due to averaging over the entire SQuAD dataset).

Table 1 presents accuracy results for GPT-Neo 2.7B on three natural language classification benchmarks. We evaluate on SST-2 (Socher et al. [2013]), a binary sentiment classification dataset containing movie reviews labeled as positive or negative; MNLI (Williams et al. [2018]), a natural language inference task requiring models to determine whether a premise entails, contradicts, or is neutral to a hypothesis; and PAWS (Zhang et al. [2019]), a paraphrase identification dataset distinguishing between semantically equivalent and non-equivalent sentence pairs.

Following the approach explored in Liu et al., we apply token masking to demonstrations within the prompt. We compare four configurations: clean demonstrations without masking, two curriculum configurations (0%-50%-50% and 1%-50%-50%) that progressively increase masking across the demonstration sequence and equal 50% masking applied uniformly to all demonstrations. The curriculum schedules gradually introduce corruption, starting with mostly clean examples and increasing the masking proportion as the demonstration sequence progresses.

The results reveal task-dependent effects. For SST-2, all three masking strategies substantially improve over clean demonstrations, with accuracies rising from 65.3% to approximately 73.7%. For MNLI and PAWS, performance remains largely stable with only minor variations. These patterns suggest that introducing deliberate corruption into demonstrations can improve model robustness when the query itself is noisy, though the benefits vary with task characteristics. Notably, we observe minimal difference between curriculum and equal masking approaches, indicating that the presence of masked tokens matters more than their temporal ordering within the prompt.

5.4 Text Generation with Context Masking

$p_{\max}$	Mode	EM Score	F1 Score
0.2	no_noise	0.3530	0.5601
	random	0.2370	0.3968
	curriculum	0.1990	0.3604
0.3	no_noise	0.2190	0.3935
	random	0.0570	0.1257
	curriculum	0.0270	0.0636
0.4	no_noise	0.1070	0.2267
	random	0.0110	0.0308
	curriculum	0.0020	0.0061

Table 2: Exact Match (EM) and F1 scores of Llama-2 7B on text generation (SQuAD 2.0) task with varying masking levels in demonstration prompts (standard deviation omitted from the table owing to its very small value, due to averaging over the entire SQuAD dataset).

As shown in Table 2, introducing masking into demonstrations affects generative ICL performance, but the overall pattern is not straightforward. At lower masking levels (pmax = 0.2), both random and curriculum masking reduce EM/F1 relative to uncorrupted demonstrations, while at higher masking levels the performance becomes extremely low for all masking types. Because generation is inherently sensitive to long-range context omissions, and because EM/F1 collapse even under small amounts of corruption, it is difficult to isolate whether curriculum masking offers any meaningful trend. Overall, the generation task produces inconclusive outcomes, and the effect of curriculum-style masking remains unclear. Further experiments like different prompt formats, alternative corruption types or training-time objectives—would be needed to draw stronger conclusions.

6 Conclusion

Our experiments provide a mixed answer to our hypotheses. Consistent with H1, training Transformers with noisy demonstrations improves robustness to corrupted in-context examples in both the synthetic regression tasks and the natural-language classification benchmarks, indicating that noise exposure itself can enhance stability in discriminative settings. However, we do not find corresponding evidence for H2: curriculum-style noise schedules perform similarly to, but not systematically better than, randomly sampled noise, suggesting that robustness gains are driven more by the presence of noise than by its temporal structure. We additionally evaluated a text-generation task using SQuAD-style prompting with masked demonstrations; unlike the regression and classification settings, these results did not reveal a clear pattern and varied substantially across masking types. Owing to the sensitivity of generative ICL to missing context, we treat these generation findings as inconclusive and do not draw strong claims about curriculum effects in this domain.

References

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chen Cheng, Xinzhi Yu, Haodong Wen, Jingsong Sun, Guanzhang Yue, Yihao Zhang, and Zeming Wei. Exploring the robustness of in-context learning with noisy labels. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, 2023.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 1107–1128, 2024.

H. Gao et al. On the noise robustness of in-context learning for text generation. *arXiv preprint arXiv:2405.17264*, 2024.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. Pmlr, 2017.

Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. Let's learn step by step: Enhancing in-context learning ability with curriculum learning, 2024b. *URL https://arxiv.org/abs/2402.10738*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153, 2022.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

Qixun Wang, Yifei Wang, Yisen Wang, and Xianghua Ying. Can in-context learning really generalize to out-of-distribution tasks? *arXiv preprint arXiv:2410.09695*, 2024.

Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International conference on machine learning*, pages 5238–5246. PMLR, 2018.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pages 1112–1122, 2018.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? *arXiv preprint arXiv:2012.03107*, 2020.

Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, Hongxin Wei, et al. On the noise robustness of in-context learning for text generation. *Advances in Neural Information Processing Systems*, 37: 16569–16600, 2024.

Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*, 2019.

Appendices

A Hyperparameter and Experimental Configuration Details

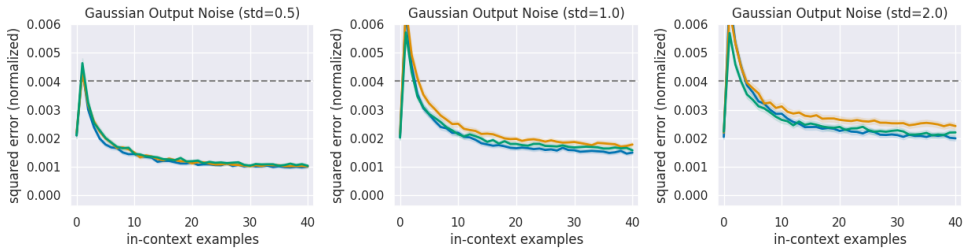
All experiments inherit a common configuration defined. The shared hyperparameters are:

- GPT-2 architecture with $n_{\text{dims}} = 20$, $n_{\text{positions}} = 101$
- Batch size: 2048
- Learning rate: Constant 4×10^{-4}
- Training steps: Up to 10,000; Early stopping after 500 steps without validation MSE decrease

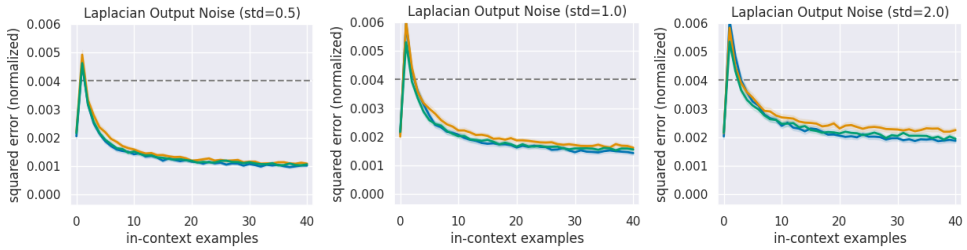
The experiments differ only in their noise mode (none, random, curriculum). In both random and curriculum noise experiments, we perturb each demonstration output by some noise drawn IID from a zero-mean Gaussian distribution. In the curriculum setting, we begin with zero variance (no noise) and we incrementally increase the standard deviation of the noise distribution by 0.1 every 100 steps (capped at 1.0). In the random setting, for each set of demonstrations, we draw the noise standard deviation as IID uniform [0, 1].

Our architectural and optimization choices follow the standard GPT-2 small configuration described in Radford et al. [2019], while our episodic training setup mirrors the synthetic regression framework of Garg et al. [2022]. To isolate the effect of noise, all model and optimizer hyperparameters are held fixed across experiments; the only quantities we vary are those directly tied to the noise intervention (noise schedule and evaluation noise distributions).

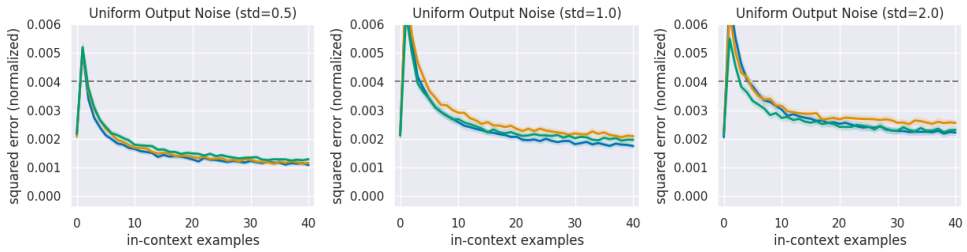
B 6-layer ReLU Neural Network Task



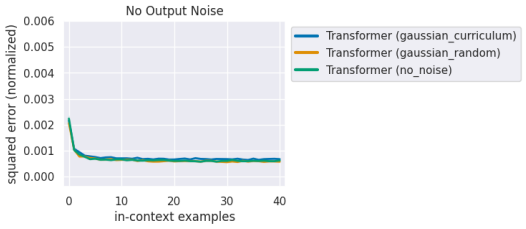
(a) Gaussian noise $\sigma = 0.5$ (b) Gaussian noise $\sigma = 1.0$ (c) Gaussian noise $\sigma = 2.0$



(d) Laplacian noise $B = 0.5$ (e) Laplacian noise $B = 0.5$ (f) Laplacian noise $B = 0.5$



(g) Uniform noise $[0, 0.5]$ (h) Uniform noise $[0, 1]$ (i) Uniform noise $[0, 2]$



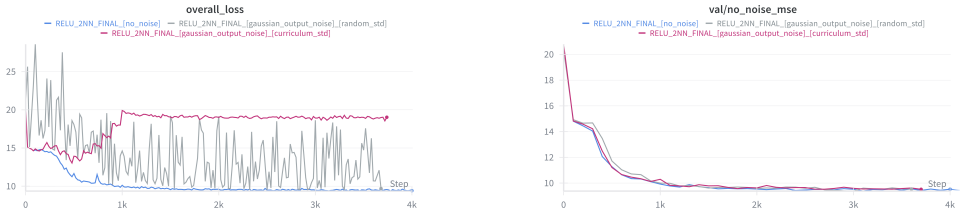
(j) No noise

Figure 5: ICL performance with different noise training strategies under different evaluation noise perturbations for **6-layer ReLU Neural Network**. Squared error (scaled and normalized) versus number of in-context examples for transformers trained with curriculum noise (blue), random noise (orange), or no noise (green). Results averaged over 1,280 context pairs; standard deviations are plotted (negligible/invisible to naked eye).

This experiment, as an extension to 2-layer ReLU Neural Network task, performed with 6-layer ReLU Neural Network provided no conclusive results as the task complexity in the synthetic regime increased, Figure 5. And hence, we omit this result from the main body of the report.

C Loss Curve for 2-layer ReLU Neural Network Task

Here, in Figure 6, we provide the training and validation loss curves for the 2-layer ReLU Neural Network Task.



(a) Training Loss curve

(b) Validation Loss curve

Figure 6: Loss curves while training 2-layer ReLU Neural Network Task with three different settings: no-noise, random noise and curriculum noise.

D Examples with NLP Tasks

Demo 1 — Clean Input
Context: The Nile is the longest river in Africa.
Question: What is the longest river in Africa?

Answer: The Nile.

Demo 2 — Masked Input
Context: The Sahara is the largest `<MASK>` in Africa.
Question: What is the largest `<MASK>` in Africa?

Answer: The Sahara.

⋮

N demos

Test Input (masked)
Context: Mount Kilimanjaro is the tallest mountain in `<MASK>`.
Question: Where is Mount Kilimanjaro located?

Model Output: (ICL prediction goes here)

Figure 7: Example of In-Context Learning with clean and masked demonstrations.

E LLM Usage

We used LLMs, ChatGPT and Claude, to assist in generating and refining portions of the text in this report. Their use was limited to improving the clarity, structure and presentation of our writing; all research ideas, experimental designs, analyses and conclusions were conceived and developed by the authors. In addition, some coding tasks—such as boilerplate generation and debugging—were partially supported by GitHub Copilot. All substantive contributions, including the choice of tasks, modeling decisions, noise schedules, evaluation setups and interpretation of results, are original and the product of our own work.

F Incorporating Reviewers Suggestions

As addressed by the reviewers, we have explicitly mentioned our hypothesis in Section 1. We clarified the noise injection mechanism with the help of Figure 1 and incorporated specific scheduling clarifications. We performed additional experiments using the 6-layer ReLU Neural Network. We elaborated on our method of masking and the reason behind some design/hyperparameter choices. Some minor explanation issues as well as grammatical mistakes were also corrected.