Learning to Choose or Choosing to Learn: Best-Of-N vs. Supervised Fine-Tuning for Bit String Generation

Seamus Somerstep
University of Michigan
Department of Statistics

Vinod Raman[1]
University of Michigan
Department of Statistics

Unique Subedi
University of Michigan
Department of Statistics

Yuekai Sun
University of Michigan
Department of Statistics

Abstract

Using the bit string generation problem as a case study, we theoretically compare two standard methods for adapting large language models to new tasks. The first, referred to as *supervised fine-tuning*, involves training a new next token predictor on good generations. The second method, *Best-of-N*, trains a reward model to select good responses from a collection generated by an unaltered base model. If the learning setting is realizable, we find that supervised fine-tuning outperforms BoN through a better dependence on the response length in its rate of convergence. If realizability fails, then depending on the failure mode, BoN can enjoy a better rate of convergence in either n or a rate of convergence with better dependence on the response length.

1 INTRODUCTION

Pre-training on vast amounts of data has allowed modern large language models (LLMs) to perform admirably on tasks ranging from simple text generation to complex reasoning (Wang et al., 2019; Paperno et al., 2016; Chollet, 2019; Hendrycks et al., 2021; Rein et al., 2023). Despite this, work remains to reliably adapt LLMs to new tasks at test time. Large language models can often confidently make up information (hallucination) (Hicks et al., 2024; Maleki et al., 2024; Bruno et al., 2023) and also struggle with abstract reasoning (Gendron et al., 2023). In response to this, a vast array of LLM post-training techniques has arisen (Kumar et al., 2025). Simple tricks include those that alter the prompt of LLMs: "chain of thought" type prompts have successfully improved reasoning (Wei et al., 2022b; Kojima et al., 2022) and it is even possible to tune the prompt for a given task using deep learning (Liu et al., 2021). Another class of crucial methods utilizes reinforcement learning. Reinforcement Learning From Human Feedback can be utilized to align LLMs towards human values (Ouyang et al., 2022). The success of this method recently culminated in the use of GRPO to increase reasoning in the deepseek family of models (DeepSeek-AI et al., 2025). While prompt and reinforcement learning-based techniques are critical, we are primarily interested in studying techniques that fall within either supervised fine-tuning or inference time scaling.

Supervised Fine Tuning: Supervised fine tuning trains an LLM for a new task utilizing the same *next token prediction* objective that is utilized during pre-training. In this case, carefully curated datasets that exhibit desirable behavior are used. A powerful use-case of SFT is teaching LLMs to follow user-provided instructions (Sanh et al., 2022; Wei et al., 2022a). The GPT family (Hicks et al., 2024) of models underwent fine-tuning on multi-turn dialogue to increase interactivity, while fine-tuning on reasoning chains can teach small models to reason (Magister et al., 2023). Supervised fine-tuning is also a common technique for knowledge distillation (Wan et al., 2024; Zhu et al., 2024). On domain-specific tasks such as medical diagnosis, sentiment analysis, and legal analysis, supervised fine tuning is helpful for increasing performance (Wang et al., 2022; Zhang et al., 2023; Yue et al., 2023). On the other hand, supervised fine tuning can actually degrade model performance through catastrophic forgetting (Luo et al., 2025)or a decrease in reasoning ability (Lobo et al., 2025). Theoretically, we will see that supervised fine tuning can enjoy good generalization, but that this property is sensitive to the learning setting.

[1]Work done while interning at Apple.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Inference Time Compute: Scaling inference time computation is an integral ingredient to the dazzling success in AI that has emerged over the last year (Hicks et al., 2024; DeepSeek-AI et al., 2025; Snell et al., 2024); particularly within long-form reasoning (Welleck et al., 2024; Yao et al., 2023; Zhang et al., 2024). While methods for utilizing inference time compute can be complex (Xie et al., 2023; Tian et al., 2024; Gandhi et al., 2024), the tried and true Best-of-N (BoN) method can provide substantial boosts to performance gains (Cobbe et al., 2021; Lightman et al., 2023) and is even a crucial ingredient to the DeepSeek family of models (DeepSeek-AI et al., 2025). In Best-of-N (Sun et al., 2024), N candidate responses are generated from a base model; then one is selected based on some criteria (Askell et al., 2021; et al., 2022; Stiennon et al., 2022). BoN has been used as a base for exploration during reinforcement learning (Kumar et al., 2025), *OpenAI* used BoN to help design WebGPT (Nakano et al., 2022) and BoN can even match the performance of RLHF for alignment (Rafailov et al., 2024). We are particularly interested in the case where the reward model used is learned from data (Cobbe et al., 2021; Snell et al., 2024; Brown et al., 2024).

1.1 Main Question and Motivation

While it is clear that both of these methods are powerful, it remains unclear *how these methods compare* for adapting a model to a new task, and how sensitive this comparison is to the properties of said task. Given a set of prompts, it is possible to collect good responses and perform SFT, or collect responses from the base model, scores for these responses, fit a reward model to the scores, and then perform BoN. In a seminal study performed by Cobbe et al. (2021), it is demonstrated that as the number of training prompts grows, BoN will outperform SFT on the GSM8K data set. Recently, the authors of Snell et al. (2024) show that, in some cases, substituting BoN for additional pre-training can improve performance (but not always). This is promising evidence for the superiority of BoN, but is certainly not decisive. More support for BoN is provided in (Brown et al., 2024), where it is shown that with an oracle verifier, the accuracy of BoN can converge to 1 on reasoning tasks. The primary goal of this work is to extend these studies in a theoretical direction.

Primary Question

For a fixed number of samples n from some task and a class of functions $\mathcal{F}$, is it theoretically better to train a next token predictor and deploy the corresponding autoregressive model, or to train a reward predictor and deploy the corresponding BoN model?

2 PROBLEM SET UP AND RESULTS

For the alphabet $\Sigma = \{0,1\}$[2], we consider the space Σ^L to be the space of bit strings of length L and the space $\Sigma' = \cup_{l=0}^{\infty}\Sigma^l$ to be the space of all finite bit strings. The goal of autoregressive language generation is to learn a map of the form $f^{\mathrm{AR}} : \Sigma' \to \Sigma^T$, where f^{AR} autoregressively builds the string with a *next token producer* f. Here T is the length of responses of the language model, which we assume is fixed throughout for simplicity. An important component of f^{AR} is the string concatenation operator, which we denote as $\circ$. Additionally, for a string σ, we let $\sigma[t]$ be the t'th element of the string, $\sigma[-1]$ be the last element of the string, and $\sigma[:t]$ be all elements of σ up to but not including $\sigma[t]$.

Back to f^{AR}, let $f \in \Sigma^{\Sigma'}$ be a next token predictor and define the map $f_{\mathrm{ap}} : \Sigma' \to \Sigma'$ given by $f_{\mathrm{ap}}(\sigma) = (\sigma \circ f(\sigma))$. Then, the autoregressive map indexed by f is given by $f^{\mathrm{AR}}(\sigma) = (f(\sigma) \circ f(f_{\mathrm{ap}}(\sigma)) \circ, \ldots, \circ f(f_{\mathrm{ap}}^{\circ T-1}(\sigma)))$. We will denote the class of next token predictors of interest as $\mathcal{F} \subset \Sigma^{\Sigma'}$. Finally, for clarity, we will denote autoregressive model inputs as $x \in \Sigma'$ and model outputs as $\mathbf{y} = (y_1, \ldots, y_T) \in \Sigma^T$. We will specify a language modeling task of interest by a distribution of model inputs $P_X \in \Delta(\Sigma')$, a distribution of base model responses $\pi_0(\mathbf{y}|x)$, the class of functions $\mathcal{F}$, a distribution that generates SFT training responses $\pi_{\mathrm{SFT}}(\mathbf{y}|x)$, and a target function f_*. We consider cases where $f_* \in \mathcal{F}$ and $f_* \notin \mathcal{F}$, and will include this as a defining feature of realizable and agnostic settings, respectively.

For the target function f_* we denote the reward function induced by f_* as $r_{f_*} : \Sigma' \times \Sigma^T \to \{0,1\}$. In particular, the reward function $r_{f_*}(x, \mathbf{y})$ induced by f_* will measure how closely a response $\mathbf{y}$ approximates the autoregressive output of f_* at x. In this paper, we will only consider binary reward functions r_{f_*} as we are interested in tasks where the model is tested on some form of correctness at test time. Similar to the reward induced by f_*, Along with the class $\mathcal{F}$, we have a class of binary rewards $\mathcal{R}_{\mathcal{F}}$ induced by $\mathcal{F}$:

$$\mathcal{R}_{\mathcal{F}} \triangleq \{r_f : \Sigma' \times \Sigma^T \to \{0,1\} \mid f \in \mathcal{F}\}. \tag{2.1}$$

Since both $\mathcal{R}_{\mathcal{F}}$ and $\mathcal{F}$ are now binary classes, we can measure their complexity using the same complexity measure (i.e. VC dimension).

We provide two natural examples below that might be used during test time for a language modeling task.

[2]Results for general alphabets and reward spaces are discussed in the appendix

Example 2.1. *Consider two possible rewards induced by a function f_*.*

$$\textit{End token reward: } r_{f_*}(x, \mathbf{y}) = \mathbf{1}\{\mathbf{y}[-1] = f_*^{AR}(x)[-1]\}. \tag{2.2}$$

$$\textit{0-1 reward: } r_{f_*}(x, \mathbf{y}) = \mathbf{1}\{\mathbf{y} = f_*^{AR}(x)\}. \tag{2.3}$$

The 0-1 reward is intended to emulate language modeling tasks such as health bench (Arora et al., 2025) where the entire response is graded, while the end token reward models tasks similar to Cobbe et al. (2021), where only the final piece of the response is judged. Throughout the paper, we will point out when a result holds for a general r_f and when the specific form r_f is important for the given result.

The objective of the language modeling task is to take in data $\mathcal{D}_n$ of sample size n drawn from $P_{\mathcal{D}}^n$, (we discuss specifics of the data below) and produce either a function $\hat{f}$ such that $\mathbb{E}_{P_X}[r_{f_*}(x, \hat{f}^{\mathrm{AR}}(x))] \approx 1$ or a policy $\hat{\pi}$ such that $\mathbb{E}_{P_X}\mathbb{E}_{\mathbf{y}\sim\hat{\pi}^{\mathrm{AR}}(\mathbf{y}|x)}[r_{f_*}(x, \mathbf{y})] \approx 1$ with high probability over the draw of training data. We emphasize that the quality of a function $\hat{f}$ is measured by $r_{f_\star}(x, \hat{f}^{\mathrm{AR}}(x))$.

There are two regimes of learning we will cover, supervised fine-tuning and generation-verification. In generation-verification we assume sampling access to a base model π_0 (which has small expected autoregressive reward) *and* the ability to measure reward for a portion of these samples. In supervised fine-tuning, we assume that we have data drawn from (but no sampling access to) the policy π_{SFT} (which has an expected autoregressive reward of 1).

2.1 Generation-Verification

In this section we introduce the Best-of-N (BoN) method commonly used in the generator-verifier regime. Recall that this method requires sampling access to some base model π_0.

1. Collect training data $\mathcal{D}_n \triangleq \{(x_i, \mathbf{y}_i), r_{f_*}(x_i, \mathbf{y}_i)\}_{i=1}^n$. When it is clear from the context, we may write the distribution of this training data $P_{\mathcal{D}} \overset{d}{=} P_X \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x_i, \mathbf{y}_i)$.

2. Train a reward estimator

$$\hat{r} \in \arg\min_{r\in\mathcal{R}_{\mathcal{F}}} \sum_{i=1}^n \mathbf{1}\{r(x_i, \mathbf{y}_i) \neq r_{f_*}(x_i, \mathbf{y}_i).\} \tag{2.4}$$

   As convention, we will break ties at random.

3. Deploy the BoN model π_{BoN}, which, given an input $x \in \Sigma'$, samples N candidate responses $\mathbf{y}_1, \ldots, \mathbf{y}_N$ from π_0, then finally selects a final response determined by

$$\mathbf{y}^* \in \arg\max_{\mathbf{y}\in\{\mathbf{y}_1,\ldots,\mathbf{y}_N\}} \hat{r}(x, \mathbf{y}) \text{ where } \mathbf{y}_j \sim \pi_0(\mathbf{y}|x).$$

Recalling the definition of $\mathcal{R}_{\mathcal{F}}$ (*cf.* Equation 2.1) we see that picking a reward model $\hat{r}$ according to Equation 2.4 is essentially equivalent to picking the function f whose induced reward r_f best approximates the reward r_{f_*} over the training data. This set up is intended to emulate empirical works such as Cobbe et al. (2021) where the same model class is used for both next token prediction and reward prediction. We still distinguish between $\mathcal{F}$ and $\mathcal{R}_{\mathcal{F}}$ because the complexity of the reward class induced by $\mathcal{F}$ need not match the complexity of the class $\mathcal{F}$.

2.2 Supervised Fine Tuning

In the SFT regime, we have access to data $\mathcal{D}_n : (x_i, \mathbf{y}_i)_{i=1}^n$ where the data generating distribution for $\mathcal{D}$ is $P_{\mathcal{D}} \overset{d}{=} P_X \times \pi_{\mathrm{SFT}}(\mathbf{y}|x)$. Generally, π_{SFT} is thought of as the "gold standard," i.e., we can write $\mathbb{E}_{P_X}\mathbb{E}_{y\sim\pi_{\mathrm{SFT}}^{\mathrm{AR}}(x)} r_{f_*}(x, \mathbf{y}) = 1$. In the SFT procedure, we first select the next token predictor $\hat{f}$ defined by

$$\hat{f}_{\mathrm{ntp}} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^n \sum_{t=1}^T \mathbf{1}\{f(x_i \circ y_i[:t]) \neq y_i[t]\}, \tag{2.5}$$

and then produce the autoregressive map given by $\hat{f}_{\mathrm{ntp}}^{\mathrm{AR}}$. As in BoN, we will break ties in the training process at random. We point out the object $\hat{f}_{\mathrm{ntp}}$ is similar to the chain-of-thought function studied by Joshi et al. (2025). The key difference is in the objective function used during training: their chain of thought function is produced by replacing $y_i[:t]$ with $f^{\mathrm{AR}}(x_i)[:t]$ in Equation 2.5. This seems like a minor distinction but will produce different convergence properties, especially in so-called agnostic learning settings.

2.3 Summary of Findings

A key difference between BoN and SFT is how they behave in the *agnostic* setting, i.e. when the target function is not in $\mathcal{F}$ or supervised fine-tuning training labels are noisy in some sense. We now formalize the concept of agnostic (and realizable) learning for language modeling tasks.

Definition 2.2. *We say that the language modeling task specified by P_X, π_0, π_{SFT}, $\mathcal{F}$ and f_* is realizable if $f_* \in \mathcal{F}$ and $\pi_{SFT}(\mathbf{y}|x) = \mathbf{1}(\mathbf{y} = f_*(x))$. If this does not hold, we refer to the task as agnostic.*

Additionally, an important quantity for us is the performance of the base policy used. The pertinent quantity

for us is known as "coverage" (Huang et al., 2025b) in the literature.

Definition 2.3. *Define the function coverage function as* $P_0(x, r_{f_*}) = \mathbb{P}_{\mathbf{y}\sim\pi_0^{AR}(\mathbf{y}|x)}\mathbf{1}\{r_{f_*}(x,\mathbf{y}) = 1\}$.

Throughout we will assume that for $x \in \Sigma'$ it holds that $P_0(x, r_{f_*}) \geq \alpha > 0$, where we refer to $\alpha \in [0, 1]$ as the *coverage constant.* The necessity of such a bound is discussed for the case of continuous rewards in Huang et al. (2025b).

Below, we present informal versions of our results for both the realizable and agnostic settings.

Theorem 2.4 (Informal, Realizable setting)**.** *Let* $\mathcal{F}$ *and* P_x *be any class and marginal distribution over* Σ' *respectively. If* $f_* \in \mathcal{F}$*, then*

1. *(SFT) For both the end token and 0-1* r_f*, with high probability over* $\mathcal{D}_n \sim (P_x \times f_*^{AR}(x))^n$*, the expected reward of SFT using* $\mathcal{D}_n$ *is* $1 - \mathcal{O}(\frac{\log(T)\cdot VC(\mathcal{F})}{n})$.
2. *(BoN) For the end-token* r_f*, with high probability over* $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x,\mathbf{y}))^n$*, the expected reward of BoN using* $\mathcal{D}_n$ *is* $1 - \mathcal{O}(\frac{\log(n)\cdot T\cdot VC(\mathcal{F})}{-\log(1-\alpha)\cdot n})$ and the dependence on T can be tight*. For the 0-1 reward, with high probability over* $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x,\mathbf{y}))^n$*, the expected reward of BoN using* $\mathcal{D}_n$ *is* $1 - \mathcal{O}(\frac{\log(n)\cdot\log(T)\cdot VC(\mathcal{F})}{-\log(1-\alpha)\cdot n})$.

Theorem 2.4 shows that in the realizable setting and for end token rewards, SFT can actually perform *better* than BoN with the sample number of training samples. The formal BoN analysis for Theorem 2.4 is carried out in Section 3. First, a general bound for BoN in terms of $P_0(x, r_{f_*})$ and the quality of the reward model $\hat{r}$ is established (*cf.* Theorem 3.2). Then, standard results from learning theory and the assumption on $P_0(x, r_{f_*})$ are applied to arrive at the result (*cf.* Results 3.3, 3.5). In Theorem 3.6 we establish that because the complexity of the end token reward scales linearly with T BoN can have poor performance for any N if $n < \mathrm{VC}(\mathcal{F})\cdot T/2$. This establishes *linear* dependence of the risk on T for BoN. The formal SFT analysis is established in Theorem 4.2. In this case, SFT reduces to a "CoT" learning procedure studied by Joshi et al. (2025). Thus, an upper-bound on the performance of SFT can follows directly from results on the growth functions of auto-regressive function classes they establish.

Unlike the realizable setting, it turns out that no such reduction is possible for SFT in the agnostic setting. We will address the effects of this for the two agnostic cases (see Definition 2.2): (1) if $\pi_{\mathrm{SFT}}(\mathbf{y}|x) = \mathbf{1}(\mathbf{y} = f_*(x))$ but $f_* \notin \mathcal{F}$ and (2) if $f_* \in \mathcal{F}$ but $\pi_{\mathrm{SFT}}(\mathbf{y}|x) \neq \mathbf{1}(\mathbf{y} = f_*(x))$.

Theorem 2.5 (Informal, Agnostic setting where $\pi_{\mathrm{SFT}}(\mathbf{y}|x) = f_*^{\mathrm{AR}}(x)$ but $f_* \notin \mathcal{F}$)**.**

1. *(SFT) For the end token* r_f*, there exists a marginal* P_x*, class* $\mathcal{F}$*, function* $f_* \notin \mathcal{F}$*, and* $\pi_{SFT}(\mathbf{y}|x) = f_*^{AR}(x)$*, such that* $\sup_{g\in\mathcal{F}} \mathbb{E}_{P_x} r_{f_*}(x, g^{AR}(x)) = 1$*, but, with high probability over the draw of the SFT training dataset* $\mathcal{D}_n \sim (P_x \times \pi_{SFT}(\mathbf{y}|x))^n$*, we have that* $\mathbb{E}_{P_x} r_{f_*}(x, \hat{f}_{ntp}^{AR}(x)) = 0$.
2. *(BoN) For both the end token and 0-1* r_f*, for every marginal distribution* P_x*, class* $\mathcal{F}$*, function* $f_* \notin \mathcal{F}$*, appropriate choice of* N*, and sufficiently large* n*, with high probability over the draw of* $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x,\mathbf{y}))^n$*, it holds that*

$$\mathbb{E}_{P_x}\mathbb{E}_{\mathbf{y}\sim\hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x,\mathbf{y}) \gtrsim 1 - \left[\kappa\left(\log_{1-\alpha}\left(\frac{-\kappa}{\log(1-\alpha)}\right) - \frac{1}{\log(1-\alpha)}\right)\right],$$

where $\kappa \triangleq \inf_{r\in\mathcal{R}_\mathcal{F}} \mathbb{E}_{x,y\sim P_x\times\pi_0}\mathbf{1}\{r_f(x,y) \neq r_{f_*}(x,y)\}$.

Theorem 2.5 shows that in the agnostic setting where $\pi_{\mathrm{SFT}}(\mathbf{y}|x) = f_*^{\mathrm{AR}}(x)$ but $f_* \notin \mathcal{F}$, SFT may never be able to obtain rewards larger than 0 for the end-token-reward. This is contrast to BoN, which is able to obtain non-zero reward in this agnostic case. The formal BoN result is established in Corollary 3.4 while the formal result for SFT is given in 4.1.

So far we have seen that in the realizable setting, SFT converges with a better dependence on T than BoN, while in the agnostic setting the performance of SFT can fail to grow beyond a reward of zero. However, experiments such as those in Cobbe et al. (2021) suggest that there are problem settings where both methods converge to a valid solution but BoN will converge faster. A partial explanation for this phenomenon lies in the second agnostic case, when $f_* \in \mathcal{F}$ but $\pi_{\mathrm{SFT}}(\mathbf{y}|x) \neq f_*^{\mathrm{AR}}(x)$.

Theorem 2.6 (Informal, Agnostic setting where $f_* \in \mathcal{F}$ but $\pi_{\mathrm{SFT}}(\mathbf{y}|x) \neq f_*^{\mathrm{AR}}(x)$)**.**

1. *(SFT) For the end token* r_f*, there exists a marginal distribution* P_x*, finite class* $\mathcal{F}$*, target* $f_* \in \mathcal{F}$*, and* $\pi_{SFT}(\mathbf{y}|x) \neq f_*^{AR}(x)$ *such that if the learner performs SFT with* $n < \frac{|\mathcal{F}|\cdot T}{2}$ *training samples* $\mathcal{D}_n \sim (P_x \times \pi_{SFT}(\mathbf{y}|x))^n$*, then with high probability,* $\mathbb{E}_{P_x} r_{f_*}(x, \hat{f}_{ntp}^{AR}(x)) = \frac{1}{2}$.
2. *(BoN) Let* $\mathcal{F}$ *and* P_x *be any finite class and marginal distribution respectively. For* both *the end token and 0-1* r_f*, if* $f_* \in \mathcal{F}$ *and the learner fits BoN with training samples* $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x,\mathbf{y}))^n$*, then, with high probability, the expected reward of BoN is at least* $1 - \mathcal{O}(\frac{\log(n)\log(|\mathcal{F}|)}{\log(1-\alpha)\cdot n})$.

The analysis for Theorem 2.6 is done in Section 5. The following summarizes the answer to Question 1.1 provided by Theorems 2.4, 2.5, 2.6.

Answer (Realizable)

- If the coverage constant $P_0(x, r_{f_*})$ is sufficiently bounded away from zero, then BoN improves with n to 1 but with a rate that is possibly linear in T.
- The expected reward of SFT improves with n to 1 at a rate that is nearly independent of T.

Answer (Agnostic)

- If the coverage constant $P_0(x, r_{f_*})$ is sufficiently bounded away from zero, then the expected reward of BoN improves with n to a constant strictly greater than zero.
- If $f_* \notin \mathcal{F}$ but $\pi_{\text{SFT}}(\mathbf{y}|x) = f_*^{\text{AR}}(x)$ then as n grows, the expected reward of SFT can remain zero even if there is function $g \in \mathcal{F}$ with expected autoregressive reward of 1.
- If $\mathcal{F}$ is finite and $f_* \in \mathcal{F}$, there exists a policy $\pi_{\text{SFT}}(\mathbf{y}|x) \neq f_*^{\text{AR}}(x)$ where both the expected reward of BoN and the expected reward of SFT grow to 1, but BoN does so *faster* in terms of dependence on T.

2.4 Practical implication of findings

Thus far, we have seen a theoretical comparison of BoN and SFT. We use this subsection to emphasize the practical implications of our findings. They can be summarized in four points:

- SFT can fail due to teacher forcing, the model is trained to predict $y[t]$ using $y[:t]$ but at test time it predicts $y[t]$ using $\hat{f}(x)[:t-1]$. This suggests an alternative approach to SFT without teacher forcing.
- SFT can fail because the training objective equally weights all tokens but it may only be measured on the final token.
- A good coverage constant improves BoN.
- Our results suggest that SFT should be chosen in "realizable" settings while BoN should be picked in "agnostic" settings. As model classes grow more expressive, SFT should perform better, especially as tasks require longer responses.

The inspiration for our investigation is Figure 5 of (Cobbe et al., 2021) which we include as Figure 1.

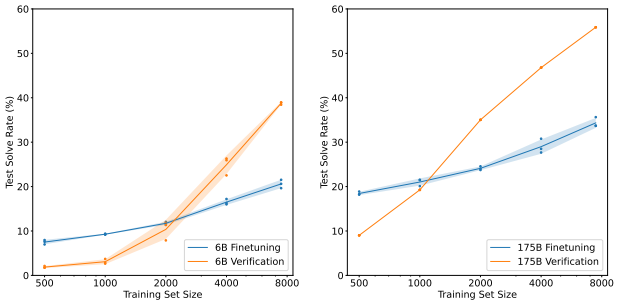

Figure 1: Cobbe et al. (2021)'s experiment of fine-tuning vs verification on GSM8K.

For the fine-tuning baseline, they fine-tune an LLM with 6B and 175B parameters on GSM8K and use it to generate a solution for each problem at test time. For verification, they fine-tune an LLM to generate (multiple) candidate solutions to each problem and a (separate) trained verifier model to judge the correctness of the (candidate) solutions. Given our theoretical findings, it suggests that 175B parameter models may not achieve zero training loss on GSM8K, as BoN seems to enjoy a better rate. Additionally, they improve the coverage constant by briefly fine-tuning the base model, which corroborates our findings. On the other hand, teacher-forcing does not seem to prevent the models from learning the task as n grows.

2.5 Related Work

Much of the set-up in Section 2 is inspired by work done by Joshi et al. (2025). They study chain-of-thought (CoT) learning; the set-up is similar in that at test time you construct an autoregressive function using a base class $\mathcal{F}$. While SFT appears similar to their CoT method, there is a crucial difference: At train time, SFT is trained to predict the next token using *ground truth* while in CoT, the model is trained to predict using its own predictions. Malach (2024) also studies autoregressive learning; however, they provide results assuming *time variance*. That is, the next-token predictor f is allowed to vary at each step t. Since LLM weights are fixed at each step during generation, we will study time-invariant learning.

Our model for supervised fine-tuning can also be thought of as a special case of behavior cloning (Bain and Sammut, 1999) where the models fit are deterministic in nature. Recent works, including (Block et al., 2023) and (Foster et al., 2024), study generative autoregressive learning through the lens of behavior cloning.

One important line of work seeks to study the optimality of BoN (as compared to other inference time

alignment methods). Like us, the authors of Huang et al. (2025b) present results on the performance of BoN for a noisy reward model. In contrast with us, they study continuous rewards and perform the analysis conditioned on a prompt, finding that BoN can perform optimally for the proper choice of N. The works Gui et al. (2024); Beirami et al. (2025) show that BoN can produce the model closest to some reference policy within a class of models with restricted KL divergence from the base model. (Yang et al., 2024) provide a comparison of BoN and RLHF under a linear reward assumption. (Foster et al., 2025) introduce a reinforcement-learning framework for studying how base models explore possible solutions during inference time computation. Like us, an important quantity is the ability of the base model.

3 ANALYSIS OF BEST-OF-N

Conditioned on an input string x, the BoN procedure draws N samples from a base policy $\pi_0(\cdot|x)$ and returns the draw y_{j^*} that scores highest with some estimated verifier. The two important quantities are the quality of the estimated verifier and the probability that $\pi_0(\cdot|x)$ generates a correct response.

We present a generalized result on the performance of BoN in terms of these quantities that hold for non-binary rewards.

Assumption 3.1. *Assume that the reward function induced by f_* is a map $r_{f_*} : \Sigma' \times \Sigma^T \to [0,1]$. For each $x \in \Sigma'$, let $\mathcal{Y}^*(x)$ denote the argmax set of the reward function $r_{f_*}(x,y)$, and assume that the reward $r_{f_*}(x,y)$ satisfies $argmax_{y\in\mathcal{Y}} r_{f_*}(x,y) > argmax_{y\in\mathcal{Y}\setminus\mathcal{Y}^*(x)} r_{f_*}(x,y) + \sigma$ for $\sigma > 0$ and all $x \in \Sigma'$.*

A similar margin assumption appears in (Huang et al., 2025a), but is defined with respect to the base policy π_0. This difference in formulation leads to performance guaranties that differ from those in our work. See (Huang et al., 2025a, Appendix C) for further details on their BoN guaranty.

Theorem 3.2. *Let $\delta(x,y) = \hat{r}(x,y) - r(x,y)$, $P_0(X) = \mathbb{P}_{\mathbf{y}\sim\pi_0(\mathbf{y}|x)}(y \in \mathcal{Y}^*(x))$ and assume assumption 1. If $\hat{\pi}_{BoN}$ corresponds to BoN with N draws at test time, it holds that*

$$\mathbb{E}_{P_x}\mathbb{E}_{\mathbf{y}\sim\hat{\pi}_{BoN}(\mathbf{y}|x)}\mathbf{1}\{\mathbf{y}\in\mathcal{Y}^*(x)\} \geq 1 - [\mathbb{E}_{x,y\sim P_x\times\pi_0(\mathbf{y}|x)}\frac{2N|\delta(x,y)|}{\sigma} + \mathbb{E}_X(1-P_0(x))^N]$$

We present a more general version of this theorem to emphasize that our analysis is not reliant on assuming that rewards are binary. In Remark B.1 we apply this fact to discuss general discrete valued rewards using the Natarajan dimension.

We will present two results, one for the case where the reward distribution is realizable by $\mathcal{R}_\mathcal{F}$ and another where it is not.

Corollary 3.3. *Let $\mathcal{F}$ and P_x be any class and marginal distribution over X respectively. Suppose the learner is provided with training data $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x,\mathbf{y}))^n$ with which a BoN model using $\hat{r}$ as defined in Equation 2.4 is deployed. If π_0 satisfies $P_0(x, r_{f_*}) \geq \alpha > 0$, $N = -\log(n)/\log(1-\alpha)$, and $f_* \in \mathcal{F}$ then with probability at least $1-\delta$ over $\mathcal{D}_n$ it holds that*

$$\mathbb{E}_{P_X}\mathbb{E}_{\mathbf{y}\sim\hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x,\mathbf{y}) \geq 1 - \left[\frac{\log(n)(\mathrm{VC}(\mathcal{R}_\mathcal{F}) + \log(1/\delta))}{-\log(1-\alpha)\cdot n} + \frac{1}{n}\right].$$

One crucial fact for us is that deploying BoN is also valid in the case where the reward distribution $(x, y, r(x,y))$ is not realizable by the class $\mathcal{R}_\mathcal{F}$. The next corollary demonstrates this.

Corollary 3.4. *Let $\mathcal{F}$ and P_x be any class and marginal distribution over X respectively. Suppose the learner is provided with training data $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x,\mathbf{y}))^n$ with which a BoN model using $\hat{r}$ as defined in Equation 2.4 is deployed. If π_0 satisfies $P_0(x, r_{f_*}) \geq \alpha > 0$, and $N = \log_{1-\alpha}(\frac{-\kappa - H(n,\mathcal{R}_\mathcal{F},\delta)}{\log(1-\alpha)})$, then with probability at least $1-\delta$ over $\mathcal{D}_n$ it holds that*

$$\mathbb{E}_{P_x}\mathbb{E}_{\mathbf{y}\sim\hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x,\mathbf{y}) \geq 1 - [(\kappa + H(n,\mathcal{R}_\mathcal{F},\delta)) \times \left(\log_{1-\alpha}\left(\frac{-\kappa - H(n,\mathcal{R}_\mathcal{F},\delta)}{\log(1-\alpha)}\right) - \frac{1}{\log(1-\alpha)}\right)]$$

with $\kappa \triangleq \inf_{r\in\mathcal{R}_\mathcal{F}} \mathbb{E}_{x,y\sim p_x\times\pi_0^{AR}} \mathbf{1}\{r_{f_*}(x,y) \neq r(x,y)\}$ and $H(n,\mathcal{R}_\mathcal{F},\delta) = \sqrt{\frac{(VC(\mathcal{R}_\mathcal{F}) + \log(1/\delta))}{n}}$.

This lower bound is messy, but one can note that as n grows, the reward is bounded below by $1 - \left[\kappa(\log_{1-\alpha}(\frac{-\kappa}{\log(1-\alpha)}) - \frac{1}{\log(1-\alpha)})\right]$, which can be close to 1, particularly for relatively small values of κ.

In summary, if one uses the class $\mathcal{R}_\mathcal{F}$ to predict the reward from (x,y), then in the realizable setting, BoN enjoys a reward of $1 - \mathcal{O}(\frac{\mathrm{VC}(\mathcal{R}_\mathcal{F})}{-\log(1-\alpha)\cdot n})$, while in the agnostic setting, the reward of BoN will converge to a value that is larger than 0.

It also remains to discuss how the complexity of the induced reward class $\mathcal{R}_\mathcal{F}$ relates to the complexity of the class $\mathcal{F}$. Fortunately, if $\mathcal{F}$ is a VC class and T is finite, then common reward classes will be as well.

Proposition 3.5. *Suppose that the class $\mathcal{F}$ is a VC class. Then if $\mathcal{R}_\mathcal{F}$ is the class of end token (Equation 2.2) rewards induced by $\mathcal{F}$, we have that $VC(\mathcal{R}_\mathcal{F}) \leq$*

$T \cdot VC(\mathcal{F})$. *For the case of 0-1 rewards (Equation 2.3), it holds that* $VC(\mathcal{R}_{\mathcal{F}}) \leq \log(T) \cdot VC(\mathcal{F})$.

As discussed, our next theorem shows that the linear dependence on T of the sample complexity of BoN cannot be improved.

Theorem 3.6. *For every* d, T *there exists a class* $\mathcal{F}$ *with* $VC(\mathcal{F}) = d$, *target* $f_* \in \mathcal{F}$, *marginal* P_x *and base policy* π_0 *satisfying* $\alpha = \frac{1}{2}$ *for* P_x *and* f_* *such that if* r_f *is the end-token reward of Equation 2.2 and* $n < \frac{d \cdot T}{2}$ *then with probability at least* $1/16$ *over* $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x, \mathbf{y}))^n$, *the risk of BoN satisfies* $1 - \mathbb{E}_{P_x} \mathbb{E}_{\mathbf{y} \sim \hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq (1/16)^2$.

Theorem 3.6 establishes that the rate of T for end-token rewards given in Theorem 2.4 cannot be improved. Theorem 3.6 should also be directly compared with Theorem 4.2; note that in the realizable setting, SFT will have much better dependence on T.

4 ANALYSIS OF SFT WITH DETERMINISTIC RESPONSES

In this section, we assume that the learner is provided "perfect" fine-tuning data $\mathcal{D}_n : (x_i, \mathbf{y}_i)_{i=1}^n$ with $\mathbf{y}_i = f_*^{\mathrm{AR}}(x_i)$.

4.1 SFT Fails if target does not lie in class

It turns out that SFT suffers from an issue that can lead to poor generalization even in the case where the learner is provided "perfect" fine-tuning data. This is due to the gap in training and test time practice for functions produced by SFT referred to as "teacher forcing" (Bachmann and Nagarajan, 2024) in the literature. Namely, at train time f is fit to predict $y[t]$ using x and $y[:t]$, while at test time the function f must predict $y[t]$ using x, and $f^{\mathrm{AR}}(x)[:t]$.

The problem only arises if the training data is not realizable by $\mathcal{F}$ (*i.e.* there is no perfect next token predictor in $\mathcal{F}$). Intuitively, early mistakes by $\hat{f}_{\mathrm{ntp}}$ can cause the chain of responses to go awry, but this is not accounted for at train time.

Theorem 4.1. *Let* r_{f_*} *be the end token reward induced by* f_*. *For* $n \geq 1$, $T \geq 2$ *there exists a marginal* P_x *and class* $\mathcal{F}$ *with* $|\mathcal{F}| = 2$ *and satisfying* $\sup_{f \in \mathcal{F}} \mathbb{E}_{P_X} r_{f_*}(x, f^{AR}(x)) = 1$ *such that with probability* 1 *over draws of* $\mathcal{D}_n \sim (P_x \times f_*^{AR}(x))$ *it holds that* $\mathbb{E}_{P_x} r_{f^*}(x, \hat{f}_{ntp}^{AR}(x)) = 0$.

The failure mechanism is not due to there being no good choice of function in $\mathcal{F}$, rather the training objective of SFT fails to make a good selection. Despite the fact that the data is not realizable by $\mathcal{F}$, a reward of one is achievable auto-regressively by the class.

The shortcomings of training the next token producer to predict using ground truth data, rather than its own predictions, have been discussed empirically before in the literature (Xiang et al., 2025; Bachmann and Nagarajan, 2024). To our knowledge, theorem 4.1 is the first to formalize this and establish the connection to agnostic learning.

4.2 Rate of convergence for SFT in the Realizable setting

We have seen that in order for SFT to enjoy a non-trivial sample complexity guarantee, it will require some additional assumptions on the class $\mathcal{F}$. Primarily, we need an assumption that will correct "teacher forcing" leading to poor performance at test time. If a function has perfect next token prediction, it will generate correct responses autoregressively; the issue of Theorem 4.1 arises because a function can perform arbitrarily poorly as it tries to predict on even only one of its own incorrect predictions. If we assume that the learner can indeed produce a perfect next token predictor (equivalent to realizability), then the problem should resolve itself. Namely, note that if $f_* \in \mathcal{F}$ then producing $\hat{f}_{\mathrm{ntp}}$ as defined in Equation 2.5 is equivalent to picking a function that perfectly interpolates the training data, *i.e.*in the realizable setting we have

$$\hat{f}_{\mathrm{ntp}} \in \{f \in \mathcal{F} : f(x_i \circ y_i[:t]) = y_i[t]; i \in [n], t \in [T]\} \tag{4.1}$$

Furthermore, this is equivalent to ERM/perfect data interpolation over the class of autoregressive functions indexed by $\mathcal{F}$ for the multilabel classification problem of predicting $\mathbf{y}$ from x.

Formally, denoting said class of functions as $\mathcal{F}^{\mathrm{AR}} \subset (\Sigma^T)^{\Sigma'}$, we have that producing $\hat{f}_{\mathrm{ntp}}^{\mathrm{AR}}$ from Equation 4.1 is equivalent to picking

$$\hat{f}_{\mathrm{ntp}}^{\mathrm{AR}} \in \{f^{\mathrm{AR}} \in \mathcal{F}^{\mathrm{AR}} : f^{\mathrm{AR}}(x_i) = \mathbf{y}_i; \quad i \in [n]\}$$

As mentioned, this is simply i.i.d ERM over the class $\mathcal{F}^{\mathrm{AR}}$. Thus in the realizable setting, we need only control the complexity of $\mathcal{F}^{\mathrm{AR}}$ to obtain guarantees for SFT. The proof of Theorem 3.4 in Joshi et al. (2025) provides the argument we require.

Theorem 4.2. *Assume that* $r_{f_*}(x, y)$ *is such that* $1 - r_{f_*}(f^{AR}(x), y) \leq \mathbf{1}[f^{AR}(x) \neq y]$. *Let* $\mathcal{F}$ *and* P_x *be any class and marginal distribution over* X *respectively. If* $f_* \in \mathcal{F}$ *then with probability at least* $1 - \delta$ *over draws of* $\mathcal{D}_n \sim (P_x \times f_*^{AR}(x))^n$ *the performance of SFT satisfies*

$$\mathbb{E}_{P_x} r_{f_*}(x, \hat{f}_{ntp}^{AR}(x)) \geq 1 - \frac{\log(T)(\,VC(\mathcal{F}) + \log(1/\delta))}{n}$$

Note that assumption $1 - r_{f_*}(f^{\mathrm{AR}}(x), y) \leq \mathbf{1}[f^{\mathrm{AR}}(x) \neq y]$ encompasses all interesting reward functions. Theorem 4.2 establishes near independence on T for SFT

if $f_* \in \mathcal{F}$ and the training responses are deterministic, this should be contrasted with Theorem 3.6 for BoN.

5 SFT WITH RANDOM RESPONSES

In this section we assume that $f_* \in \mathcal{F}$ but that SFT data $\mathcal{D}_n = (x_i, \mathbf{y}_i)$ is generated by a *non-deterministic* policy $\pi_{\text{SFT}}(\mathbf{y}|x)$, *i.e.* $\pi_{\text{SFT}}(\mathbf{y}|x) \neq f_*(x)$. We will also assume that we are working with the end token reward and are studying function classes that are finite. To begin, we present the analog of Corollary 3.3 for finite classes. This will make the comparison with SFT immediate.

Proposition 5.1. *Let $\mathcal{F}$ and P_x be any finite class and marginal distribution over X respectively. Suppose the learner is provided with training data $\mathcal{D}_n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x, \mathbf{y}))^n$ with which a BoN model using $\hat{r}$ as defined in Equation 2.4 is deployed. If π_0 satisfies $P_0(x, r_{f_*}) \geq \alpha > 0$, $N = -\log(n)/\log(1-\alpha)$, and $f_* \in \mathcal{F}$ then with probability at least $1-\delta$ over $\mathcal{D}_n$ it holds that*

$$\mathbb{E}_{P_x}\mathbb{E}_{\mathbf{y}\sim\hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq 1 - [\frac{\log(n)(\log(|\mathcal{F}|) + \log(1/\delta))}{-\log(1-\alpha)\cdot n} + \frac{1}{n}].$$

Crucially, in the case of finite classes, BoN loses the dependence on T in its guarantee.

Next, we study how SFT behaves if training responses are random. The following assumption allows for a distinction from the issues with SFT in Theorem 2.5. In particular, Assumption 5.2 will ensure that SFT does converge to a reward of 1 with n.

Assumption 5.2. *In the case of non-deterministic training responses, for a given function class $\mathcal{F}$ and target function f_* we assume that the reference policy generates responses with expected reward 1 and must satisfy*

$$\mathbb{E}_{P_{\mathcal{D}}} \sum_{t=1}^{T} \mathbf{1}\{f_*(x \circ y[:t]) \neq y[t]\} + \frac{1}{2} < \min_{\mathcal{F}\setminus f_*} \mathbb{E}_{P_{\mathcal{D}}} \sum_{t=1}^{T} \mathbf{1}\{f(x \circ y[:t]) \neq y[t]\}$$

Unlike the case where $\pi_{\text{SFT}}(\mathbf{y}|x) = \mathbf{1}(\mathbf{y} = f_*(x))$, there can be a gap in dependence on T in the rate of convergence that is in favor of BoN.

Theorem 5.3. *Assume that r_f is the end-token reward. For T, $d > 5$ and $T < d$, there exists a finite class $\mathcal{F}$ with $|\mathcal{F}| = d$, marginal P_x, π_{SFT} satisfying Assumption 5.2, and target $f_* \in \mathcal{F}$ such that if $n < \frac{\log(|\mathcal{F}|)\cdot T}{2}$ then with probability at least $\frac{1}{4}$ over $\mathcal{D}_n \sim (P_x \times \pi_{SFT}(\mathbf{y}|x))^n$ it holds that $\mathbb{E}_x r_{f_*}(x, \hat{f}^{AR}_{ntp}(x)) = \frac{1}{2}$.*

Of course, this can be compared with the BoN result Proposition 5.1 to see that in this case the rate of convergence for SFT has worse dependence on T. Using a simple application of Hoeffding's inequality, we can also show that 5.2 is enough to ensure that SFT will converge to a reward of 1.

Proposition 5.4. *Assume that $r_{f_*}(x, y)$ is such that $1 - r_{f_*}(f^{AR}(x), y) \leq \mathbf{1}[f^{AR}(x) \neq y]$. Let $\mathcal{F}$ be any finite class, P_x any marginal distribution over X respectively, and $\pi_{SFT}(\mathbf{y}|x)$ any distribution satisfying Assumption 5.2. If $f_* \in \mathcal{F}$ then with probability at least $1-\delta$ over draws of $\mathcal{D}_n \sim (P_x \times \pi_{SFT}(\mathbf{y}|x))^n$*

$$\mathbb{E}_x r_{f_*}(x, \hat{f}^{AR}_{ntp}(x)) \geq 1 - \sqrt{\frac{\log(|F|/\delta)}{n}} \cdot T.$$

Unfortunately, this result is not entirely satisfactory, as a gap remains between the upper bound in Proposition 5.4 and the lower bound in Theorem 5.3. We recognize a complete characterization of the convergence of SFT as an important problem and discuss the challenges in Section 6.

6 DISCUSSION AND LIMITATIONS

In this work we set out to characterize how BoN and SFT perform on the bit string generation problem. In the case where SFT training data is of the form $\mathcal{D} : (x_i, f_*^{\text{AR}}(x_i))$ for the target function f_*, we showed that if f_* lies in $\mathcal{F}$ then SFT converges at a rate with only log dependence on T, but if $f_* \notin \mathcal{F}$ then SFT can perform poorly. On the other hand, BoN is more resilient to f_* not lying in $\mathcal{F}$ but has worse dependencies on T for certain reward functions. We also introduced a setting where SFT training responses are random, leading to linear dependence on T for the case of finite classes. Beyond these results, several open questions remain.

1. How can we characterize the convergence of SFT in the case where the SFT training responses are noisy? One challenge is the non-i.i.d. nature of the training procedure used to select $\hat{f}_{\text{ntp}}$, another is the "distribution shift" between train and test time that results from teacher forcing.

2. In the case of deterministic training responses, is Theorem 4.2 sharp? In particular, does the performance of SFT truly depend on $\log(T)$ or is this term in the upper bound superfluous?

3. How does BoN perform if, during the reward fitting stage, multiple draws from π_0 are used for each

prompt x? This is done in Cobbe et al. (2021) and should improve the performance of BoN but again moves us beyond the i.i.d. learning regime we used to attain the results in this work.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2414918. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The authors thank Yilun Zhu for helpful comments on the final draft.

References

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021. URL https://arxiv.org/abs/2112.00861.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction, 2024. URL https://arxiv.org/abs/2403.06963.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995]*, page 103–129, GBR, 1999. Oxford University. ISBN 0198538677.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy, 2025. URL https://arxiv.org/abs/2401.01879.

Adam Block, Ali Jadbabaie, Daniel Pfrommer, Max Simchowitz, and Russ Tedrake. Provable guarantees for generative behavior cloning: Bridging low-level stability and high-level behavior, 2023. URL https://arxiv.org/abs/2307.14619.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL https://arxiv.org/abs/2407.21787.

A. Bruno, P. L. Mazzeo, A. Chetouani, M. Tliba, and M. A. Kerkouri. Insights into Classifying and Mitigating LLMs' Hallucinations. *arXiv preprint*, arXiv:2311.08117, 2023.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Amelia Glaese et al. Improving alignment of dialogue agents via targeted human judgements, 2022. URL https://arxiv.org/abs/2209.14375.

Dylan J. Foster, Adam Block, and Dipendra Misra. Is behavior cloning all you need? understanding horizon in imitation learning, 2024. URL https://arxiv.org/abs/2407.15007.

Dylan J. Foster, Zakaria Mhammedi, and Dhruv Rohatgi. Is a good foundation necessary for efficient reinforcement learning? the computational role of the base model in exploration, 2025. URL https://arxiv.org/abs/2503.07453.

Kanishk Gandhi, Denise H J Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah Goodman. Stream of search (sos): Learning to search in language. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=2cop2jmQVL.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. Large language models are not strong abstract reasoners. *arXiv preprint arXiv:2305.19555*, 2023.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling, 2024. URL https://arxiv.org/abs/2406.00832.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

M. T. Hicks, J. Humphries, and J. Slater. ChatGPT is Bullshit. *Ethics and Information Technology*, 26(2): 1–10, 2024.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=WJaUkwci9o.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J. Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment, 2025b. URL https://arxiv.org/abs/2503.21878.

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought, 2025. URL https://arxiv.org/abs/2503.07932.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H. S. Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models, 2025. URL https://arxiv.org/abs/2502.21321.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602, 2021. URL https://arxiv.org/abs/2110.07602.

Elita Lobo, Chirag Agarwal, and Himabindu Lakkaraju. On the impact of fine-tuning on chain-of-thought reasoning, 2025. URL https://arxiv.org/abs/2411.15382.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL https://arxiv.org/abs/2308.08747.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason, 2023. URL https://arxiv.org/abs/2212.08410.

Eran Malach. Auto-regressive next-token predictors are universal learners, 2024. URL https://arxiv.org/abs/2309.06979.

N. Maleki, B. Padmanabhan, and K. Dutta. AI Hallucinations: A Misnomer Worth Clarifying, 2024. Preprint.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, 2016.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms, 2024. URL https://arxiv.org/abs/2406.02900.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan

Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Igal Sason and Sergio Verdú. Upper bounds on the relative entropy and rényi divergence as a function of total variation distance for finite alphabets. In *2015 IEEE Information Theory Workshop - Fall (ITW)*, pages 214–218, 2015. doi: 10.1109/ITWF.2015.7360766.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast best-of-n decoding via speculative rejection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=348hfcprUs.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Lei Han, Haitao Mi, and Dong Yu. Toward self-improvement of LLMs via imagination, searching, and criticizing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tPdJ2qHkOB.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. Efficient large language models: A survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=bsCCJHbO8A. Survey Certification.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Yaqing Wang, Song Wang, Yanyan Li, and Dejing Dou. Recognizing medical search query intent by few-shot learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 502–512, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531789. URL https://doi.org/10.1145/3477495.3531789.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022b.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models, 2024. URL https://arxiv.org/abs/2406.16838.

Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Franken, Nick Haber, and Chelsea Finn. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought, 2025. URL https://arxiv.org/abs/2501.04682.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. Self-evaluation guided beam search for reasoning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, 2023.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment, 2024. URL https://arxiv.org/abs/2404.01730.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL https://arxiv.org/abs/2305.10601.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. Disc-lawllm: Fine-tuning large language models for intelligent legal services, 2023. URL https://arxiv.org/abs/2309.11325.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search, 2024. URL https://arxiv.org/abs/2406.03816.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. Sentiment analysis in the era of large language models: A reality check, 2023. URL https://arxiv.org/abs/2305.15005.

Xiaohan Zhu, Mesrob I. Ohannessian, and Nathan Srebro. Tight bounds on the binomial cdf, and the minimum of i.i.d binomials, in terms of kl-divergence, 2025. URL https://arxiv.org/abs/2502.18611.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models, 2024. URL https://arxiv.org/abs/2308.07633.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Section 2
   (b) Complete proofs of all theoretical results. [Yes] Appendix.
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A PRELIMINARIES

A.1 Binary Classification

In binary classification, we observe a sample $\mathcal{D}_n : (x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \{0,1\})^n$ from an unknown distribution $P_{\mathcal{D}}$ and attempt to deploy a classifier h from a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\mathcal{L}_D(h) = \mathbb{E}_{P_{\mathcal{D}}}\mathbf{1}[h(x) \neq y]$ is small. An important case of binary classification is when $P_{\mathcal{D}}$ is *realizable* by the class $\mathcal{H}$. We say that this holds when $P_{\mathcal{D}}(y|x) = h_*(x)$ for some $h \in \mathcal{H}$. For the sample $\mathcal{D}_n$ two important quantities are the empirical loss,

$$\hat{\mathcal{L}}(\mathcal{D}_n; h) = \sum_{i=1}^{n} \mathbf{1}\{h(x_i) \neq y_i\}$$

and the corresponding ERM minimizer

$$\mathrm{ERM}_{\mathcal{H}}(\mathcal{D}_n) = \arg\min_{h \in \mathcal{H}} \hat{\mathcal{L}}(\mathcal{D}_n; h).$$

A.2 Growth Functions and Sample Complexity

Definition A.1 (Growth Function)**.** *Let $\mathcal{H}$ be a hypothesis class with functions $h : \mathcal{X} \to \{0,1\}$, and let $S = (x_1, \ldots, x_m) \in \mathcal{X}^n$ be a sample of size n. The* growth function $\Gamma_{\mathcal{H}} : \mathbb{N}_+ \to \mathbb{N}_+$ *is defined as*

$$\Gamma_{\mathcal{H}}(n) = \max_{(x_1,\ldots,x_n) \in \mathcal{X}^n} \left| \left\{ \big(h(x_1), \ldots, h(x_n)\big) : h \in \mathcal{H} \right\} \right|$$

With a definition for the growth function in hand we may now define the VC dimension of a hypothesis class $\mathcal{H}$ be a hypothesis class with functions $h : \mathcal{X} \to \{0,1\}$.

Definition A.2 (VC dimension)**.** *For any $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, the VC dimension of the class $\mathcal{H}$ denoted as VC(H) is the largest integer d such that $\Gamma_{\mathcal{H}}(d) \leq 2^d$, if no such integer exists then we say that VC(H) $= \infty$.*

We need to define the Natarajan dimension for the multi-class case.

Definition A.3 (Shattering (Multiclass Version))**.** *We say that a set $C \subset \mathcal{X}$ is shattered by $\mathcal{H}$ if there exist two functions $f_0, f_1 : C \to [k]$ such that:*

- *For every $x \in C$, $f_0(x) \neq f_1(x)$.*
- *For every $B \subseteq C$, there exists a function $h \in \mathcal{H}$ such that*

$$\forall x \in B, h(x) = f_0(x) \quad \textit{and} \quad \forall x \in C \setminus B, h(x) = f_1(x).$$

Definition A.4 (Natarajan Dimension.)**.** *The Natarajan dimension of $\mathcal{H}$, denoted* $\mathrm{Ndim}(\mathcal{H})$*, is the maximal size of a shattered set $C \subset \mathcal{X}$.*

The following appear as Lemmas A.1 and A.2 of (Joshi et al., 2025).

Lemma A.5 (Sauer's Lemma)**.** *For a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, for every $n \in \mathbb{N}_+$, we have*

$$\Gamma_{\mathcal{H}}(n) \leq (en)^{\mathrm{VCdim}(\mathcal{H})}.$$

Additionally, for $n \geq \mathrm{VCdim}(\mathcal{H}) \geq 1$:

$$\Gamma_{\mathcal{H}}(n) \leq \left(\frac{en}{\mathrm{VCdim}(\mathcal{H})}\right)^{\mathrm{VCdim}(\mathcal{H})}.$$

Lemma A.6 (Natarajan's Lemma)**.** *Recall the definition of the growth function. For any $n \in \mathbb{N}_+$,*

$$\Gamma_{\mathcal{H}}(n) \leq \left(|\mathcal{Y}|^2 \cdot n\right)^{\mathrm{Ndim}(\mathcal{H})}.$$

A.3 Generalization Bounds

Theorem A.7 (Corollary 2.3 (Shalev-Shwartz and Ben-David, 2014)). *Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0,1)$ and $\varepsilon > 0$, and let n be an integer satisfying*

$$n \geq \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}.$$

Then, for any distribution $P_{\mathcal{D}}$ such that the realizability assumption holds, with probability at least $1-\delta$ over the choice of an i.i.d. sample $\mathcal{D}_n$ of size n, we have that for every empirical risk minimization (ERM) hypothesis it holds that

$$L_{\mathcal{D}}(ERM_{\mathcal{H}}(\mathcal{D}_n)) \leq \varepsilon.$$

The following proposition is Theorem 6.7 of (Shalev-Shwartz and Ben-David, 2014).

Proposition A.8 (The Fundamental Theorem of Statistical Learning). *There exists a universal constant $c > 0$ such that for any domain $\mathcal{X}$, a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ with $\mathrm{VCdim}(\mathcal{H}) < \infty$, and any distribution $P_{\mathcal{D}}$ over $\mathcal{X} \times \{0,1\}$, the following holds.*

With probability at least $1-\delta$ over a sample $S \sim P_{\mathcal{D}}^n$ with $n = n(\varepsilon, \delta)$, we have

$$\mathcal{L}_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(\mathcal{D}_n)) \leq \inf_{h \in \mathcal{H}} \mathcal{L}_{P_{\mathcal{D}}}(h) + \varepsilon,$$

and

$$n(\varepsilon, \delta) \leq c \cdot \frac{\mathrm{VC}(\mathcal{H}) + \log(1/\delta)}{\varepsilon^2}.$$

Moreover, if $P_{\mathcal{D}}$ is realizable by $\mathcal{H}$, then

$$\mathcal{L}_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(\mathcal{D}_n)) \leq \varepsilon \quad \textit{with} \quad n(\varepsilon, \delta) \leq c \cdot \frac{\mathrm{VC}(\mathcal{H}) \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}.$$

In cases where the label space $\mathcal{Y}$ is not $\{0,1\}$ we can instead examine the VC dimension of the loss class induced by $\mathcal{F}$. To that end let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and for any class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ consider the loss class defined by

$$\mathcal{L}_{\mathcal{H}}^{0-1} = \{\ell_h : (x, \mathbf{y}) \to \mathbf{1}\{h(x) \neq \mathbf{y} \mid h \in \mathcal{H}\}\}.$$

The following generalization guarantee holds in the case where the loss class is a VC class.

Proposition A.9 ((Joshi et al., 2025) Proposition 4). *There exists a universal constant $c > 0$ such that the following holds. For any domain $\mathcal{X}$, label space $\mathcal{Y}$, and hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with finite VC-dimension $\mathrm{VC}(\mathcal{H}) < \infty$, and for any distribution $P_{\mathcal{D}}$ over $\mathcal{X} \times \mathcal{Y}$ that is realizable by $\mathcal{H}$, the following holds.*

With probability at least $1-\delta$ over the choice of an i.i.d. sample $\mathcal{D}_n \sim P_{\mathcal{D}}^n$ of size

$$n \geq \frac{c}{\varepsilon}\left(\mathrm{VC}(\mathcal{H}) \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right),$$

it holds that

$$\mathcal{L}_{P_{\mathcal{D}}}(ERM_{\mathcal{H}}(\mathcal{D}_n)) \leq \varepsilon.$$

The following is Theorem 29.3 of (Shalev-Shwartz and Ben-David, 2014).

Proposition A.10 (The Fundamental Theorem for Multiclass Labels). *There is a universal constant $c > 0$ such that for any domain $\mathcal{X}$, a finite $\mathcal{Y}$, a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\mathrm{Ndim}(\mathcal{H}) < \infty$, and any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the following holds. Over the draw of $S \sim \mathcal{D}^n$ with $n = n(\varepsilon, \delta)$ we have*

$$L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) \leq \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon \quad \textit{and} \quad n(\varepsilon, \delta) \leq c\left(\frac{\mathrm{Ndim}(\mathcal{H}) \log|\mathcal{Y}| + \log(1/\delta)}{\varepsilon^2}\right).$$

Moreover, if $\mathcal{D}$ is realizable by $\mathcal{H}$, i.e., $\mathcal{D}_{y|x} = h^*(x)$ for some $h^* \in \mathcal{H}$, we have

$$L_{\mathcal{D}}(\mathrm{Cons}_{\mathcal{H}}(S)) \leq \varepsilon \quad \textit{with} \quad n(\varepsilon, \delta) \leq \frac{c}{\varepsilon}\left(\mathrm{Ndim}(\mathcal{H}) \log\left(\frac{|\mathcal{Y}| \cdot \mathrm{Ndim}(\mathcal{H})}{\varepsilon}\right) + \log(1/\delta)\right).$$

The following is analogous to Saur

A.4 Statistical Distances and Inequalities

Definition A.11 (KL-divergence). *For distributions P and Q defined over the same probability space $\mathcal{X}$ and satisfying $P << Q$, the KL divergence from Q to P is defined as:*

$$D_{\mathrm{KL}}(P\|Q) = \int_{x\in\mathcal{X}} \log \frac{dP(x)}{dQ(x)} dP(x)$$

Definition A.12. *For distributions P and Q defined over the same probability space $\mathcal{X}$, the total variation distance between P and Q is defined as*

$$\delta(P,Q) = \sup_{A\subset X} |P(A) - Q(A)|$$

Theorem A.13 (Pinskers' inequality (Sason and Verdú, 2015)). *Let X be a finite space and let P, Q be probability measures on X with $P << Q$. If $\alpha_Q = \min_{x\in X} Q(x)$ then it holds that*

$$\delta(P,Q)^2 \leq \frac{1}{2} D_{KL}(P||Q) \frac{1}{\alpha_Q} \delta(P,Q)^2$$

Theorem A.14 (Hoeffding's inequality). *For random variable $X = \sum_{i=1}^n X_i$ with X_i i.r.v and $X_i \in [a_i, b_i]$ it holds that*

$$\mathbf{P}(|X - \mathbb{E}X|| > \delta) \leq e^{\frac{-2\delta^2}{\sum_{i=1}^n (a_i - b_i)^2}}$$

A.5 Tail Bound on Binomial Order Statistics

Theorem A.15 ((Zhu et al., 2025) Theorem 1). *Denote* $\mathrm{KL}(\alpha||\beta) = \mathrm{KL}(Ber(\alpha)||Ber(\beta)) = \alpha\log(\frac{\alpha}{\beta}) + (1-\alpha)\log(\frac{1-\alpha}{1-\beta})$*Let $\{X_i\}_{i=1}^r$ be i.i.d. random variables distributed as $\frac{1}{n}Bin(n,p)$, and define $Z = \min_{i=1,\dots,r} X_i$. Given a fixed confidence parameter $\delta \in (0,1)$, define*

$$\Delta(\delta, p, n) = \log\left(\frac{2}{\delta}\right) + 4\log(n+1) + \left|\log\left(\frac{p}{1-p}\right)\right|.$$

If $\Delta(\delta,p,n) < \log r$, then with probability at least $1-\delta$, we have

$$Z < p \quad \textit{and} \quad \mathrm{KL}(Z\,\|\,p) \in \left[\frac{\log r - \Delta(\delta,p,n)}{n}, \frac{\log r + \Delta(\delta,p,n)}{n}\right].$$

B BoN PROOFS

B.1 Upperbound

Proof of Theorem 3.2. Note that we may write

$$\mathcal{R}(\hat{\pi}_{\mathrm{BoN}}) = 1 - \mathcal{E}(\hat{\pi}_{\mathrm{BoN}}); \quad \mathcal{E}(\hat{\pi}_{\mathrm{BoN}}) = 1 - \mathcal{R}(\hat{\pi}_{\mathrm{BoN}})$$

Moreover, we have $\mathcal{E}(f) \leq \mathbb{E}_{P_{\mathcal{D}}^n}\mathbb{E}_X\mathbb{E}_{\hat{y}\sim\hat{\pi}_{\mathrm{BoN}}}\mathbf{1}\{\hat{y}\notin\mathcal{Y}^*\}$. Note that

$$\begin{aligned}\mathbf{1}\{\hat{y}\notin\mathcal{Y}^*\} &= \mathbf{1}\{\hat{y}\notin\mathcal{Y}^*\}\cdot\mathbf{1}\left\{\{y_1',\dots y_N'\}\cap\mathcal{Y}^*\neq\emptyset\right\} + \mathbf{1}\{\hat{y}\notin\mathcal{Y}^*\}\cdot\mathbf{1}\left\{\{y_1',\dots y_N'\}\cap\mathcal{Y}^*=\emptyset\right\}\\ &\leq \mathbf{1}\{\hat{y}\notin\mathcal{Y}^*\}\cdot\mathbf{1}\left\{\{y_1',\dots y_N'\}\cap\mathcal{Y}^*\neq\emptyset\right\} + \mathbf{1}\left\{\{y_1',\dots y_N'\}\cap\mathcal{Y}^*=\emptyset\right\}.\end{aligned}$$

Here, $y_1',\dots,y_N'$ are the N responses used to compute $\hat{y}$ using BoN. Recall that

$$\mathbb{E}_X[\mathbf{1}\left\{\{y_1',\dots y_N'\}\cap\mathcal{Y}^*=\emptyset\right\}] = \mathbb{E}_X(1-P_0(X))^N.$$

Thus, we obtain

$$\mathcal{R}(f) \leq \mathbb{E}_{P_{\mathcal{D}}^n}\mathbb{E}_X\mathbb{E}_{\hat{y}\sim\hat{\pi}_{\mathrm{BoN}}}[\mathbf{1}\{\hat{y}\notin\mathcal{Y}^*\}\cdot\mathbf{1}\left\{\{y_1',\dots y_N'\}\cap\mathcal{Y}^*\neq\emptyset\right\}] + \mathbb{E}_X(1-P_0(X))^N$$

To tackle the first term, define a function $\delta : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$ such that $\delta(x, y) = \hat{r}_f(x, y) - r_{f_*}(x, y)$. Then, it is easy to see that

$$\mathbf{1}\{\hat{y} \notin \mathcal{Y}^*\} \cdot \mathbf{1}\left\{\{y'_1, \dots y'_N\} \cap \mathcal{Y}^* \neq \emptyset\right\} \leq \mathbf{1}\left\{\max_{j \in [N]} |\delta(x, y^j)| \geq \frac{\sigma}{2}\right\}.$$

To see why this is inequality is true, we can consider two cases. First, when $\mathbf{1}\left\{\{y'_1, \dots y'_N\} \cap \mathcal{Y}^* \neq \emptyset\right\} = 0$, this inequality is trivially true. In the case that $\mathbf{1}\left\{\{y'_1, \dots y'_N\} \cap \mathcal{Y}^* \neq \emptyset\right\} = 1$, it is easy to see that the BoN must pick one of the y'_j's in the argmax set as long as $\max_{j \in [N]} |\delta(x, y^j)| < \frac{\sigma}{2}$. This is because the estimated score $v(x, y'_j)$ of such $y'_j \in \mathcal{Y}^*(x)$ can only decrease by at most $\sigma/2$ and such score of next best $y'_k \notin \mathcal{Y}^*$ can only increase by $\sigma/2$. So, even the noisy verifier will assign maximum scores to $y'_j \in \mathcal{Y}^*(x)$. Therefore, $\mathbf{1}\{\hat{y} \notin \mathcal{Y}^*\}$ can be 1 only when $\max_{j \in [N]} |\delta(x, y^j)| \geq \frac{\sigma}{2}$, proving our claim. Thus, combining everything,

$$\begin{aligned}
\mathcal{R}(f) &\leq \mathbb{E}_{P_{\mathcal{D}}^n} \mathbb{E}_X \mathbb{E}_{y^1, \dots y^N \sim \pi_0(y|x)} \left[\mathbf{1}\left\{\max_{j \in [N]} |\delta(x, y^j)| \geq \frac{\sigma}{2}\right\}\right] + \mathbb{E}_X (1 - P_0(X))^N \\
&\leq \mathbb{E}_{P_{\mathcal{D}}^n} \mathbb{E}_X \mathbb{E}_{y^1, \dots y^N \sim \pi_0(y|x)} \frac{2 \max_{j \in [N]} |\delta(x, y^j)|}{\sigma} + \mathbb{E}_X (1 - P_0(X))^N \\
&\leq \mathbb{E}_{P_{\mathcal{D}}^n} \mathbb{E}_X \mathbb{E}_{y^1, \dots y^N \sim \pi_0(y|x)} \left[\frac{2 \sum_j |\delta(x, y^j)|}{\sigma}\right] + \mathbb{E}_X (1 - P_0(X))^N \\
&= \frac{2N}{\sigma} \mathbb{E}_{P_{\mathcal{D}}^n} \mathbb{E}_{X, Y \sim P_X \times \pi_0(y|x)} \left[|\delta(x, y)|\right] + \mathbb{E}_X (1 - P_0(X))^N \\
&= \frac{2N}{\sigma} \mathbb{E}_{P_{\mathcal{D}}^n} \mathbb{E}_{X, Y \sim P_X \times \pi_0(y|x)} \left[|\hat{r}_f(x, y) - r_{f_*}(x, y)|\right] + \mathbb{E}_X (1 - P_0(X))^N
\end{aligned}$$

□

Proof of Corollary 3.3. Consider the result of Theorem 3.2. For a binary reward r_{f^*} we have $\sigma = 1$, $\mathbf{1}\{\mathbf{y} \in \mathcal{Y}^*(x)\} = r_{f^*}(x, \mathbf{y})$, and $|\hat{r}(x, \mathbf{y}) - r_{f_*}(x, \mathbf{y})| = \mathbf{1}\{\hat{r}(x, \mathbf{y}) \neq r_{f_*}(x, \mathbf{y})\}$. Thus, by Theorem 3.2 for binary rewards, it holds that

$$\mathbb{E}_{P_x} \mathbb{E}_{\mathbf{y} \sim \hat{\pi}_{\text{BoN}}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq$$
$$1 - [\mathbb{E}_{x, y \sim P_x \times \pi_0(\mathbf{y}|x)} 2N \mathbf{1}\{\hat{r}(x, \mathbf{y}) \neq r_{f_*}(x, \mathbf{y})\} + \mathbb{E}_X (1 - P_0(x))^N]$$

Now note that the reward fitting stage which produces $\hat{r}$ from Equation 2.4 is standard ERM for a binary classification. Since by assumption the reward distribution is realizable by $\mathcal{R}_{\mathcal{F}}$, we may apply Proposition A.8 to say that with probability $1-\delta$ over a draw of training set $D_n : \{(x^i, \mathbf{y}^i), r_{f_*}(x^i, \mathbf{y}^i)\}_{i=1}^n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x, \mathbf{y}))^{\otimes n}$ it holds that

$$\mathbb{E}_{P_x} \mathbb{E}_{\mathbf{y} \sim \hat{\pi}_{\text{BoN}}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq$$
$$1 - [\mathbb{E}_{x, y \sim P_x \times \pi_0(\mathbf{y}|x)} \frac{2N \cdot [\text{VC}(\mathcal{R}_{\mathcal{F}}) + \log(1/\delta)]}{n} + \mathbb{E}_X (1 - P_0(x))^N]$$

Now we make use of the assumption $P_0(x) \geq \alpha > 0$ and plug in $N = -\frac{\log(n)}{\log(1-\alpha)}$ to arrive at

$$\mathbb{E}_{P_x} \mathbb{E}_{\mathbf{y} \sim \hat{\pi}_{\text{BoN}}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq$$
$$1 - [\mathbb{E}_{x, y \sim P_x \times \pi_0(\mathbf{y}|x)} \frac{2N \cdot [\text{VC}(\mathcal{R}_{\mathcal{F}}) + \log(1/\delta)]}{n} + \mathbb{E}_X (1 - \alpha)^N]$$
$$= \frac{-2 \log(n) \cdot [\text{VC}(\mathcal{R}_{\mathcal{F}}) + \log(1/\delta)]}{n \log(1 - \alpha)} + \mathbb{E}_X e^{\frac{-\log(n)}{\log(1-\alpha)} \log(1-\alpha)}]$$

□

Proof of Corollary 3.4. The proof of this Corollary is identical to that of Corollary 3.3, except utilizing the non-realizable version of Proposition A.8. □

Proof of Proposition 5.1. This follows immediately from substituting in Theorem A.7 to the proof of Corollary 3.3 and noting that in the case of finite classes $|F| = |\mathcal{R}_{\mathcal{F}}|$. □

Proof of Proposition 3.5.

1. *End Token Reward*: Consider the class of end token rewards $\mathcal{R}_{\mathcal{F}}$ induced by the class $\mathcal{F}$.

$$\mathcal{R}_{\mathcal{F}} = \{r_f : \Sigma' \times \Sigma^T \to \Sigma, \quad r_f(x \circ y) = \mathbf{1}\{y[-1] = f^{\mathrm{AR}}(x)[-1]\} | f \in \mathcal{F}\}$$

   Likewise, consider the class of functions

$$\mathcal{F}_{\mathrm{et}} = \{f_{\mathrm{et}} : \Sigma' \to \Sigma, \quad f_{et}(x) = f^{\mathrm{AR}}(x)[-1] | f \in \mathcal{F}\}$$

   Clearly, the VC dimension of these two classes are equivalent to one another. In Theorem B.1. of Joshi et al. (2025) it is shown that the first class has VC dimension bounded by $6 \cdot T \cdot \mathrm{VC}(\mathcal{F})$.

2. *0-1 Reward*: This result follows directly from the Proof of Theorem 4.2 provided in Appendix C.

□

The following remark shows that one can make a similar conclusion to Corrollary 3.3 in the case of non-binary rewards.

Remark B.1. *(Multi-class rewards) If r_{f_*} is a map from $\Sigma' \times \Sigma^T \to \{1, \ldots, R\}$ then $\sigma = 1$, and $|\hat{r}(x, \mathbf{y}) - r_{f_*}(x, \mathbf{y})| \leq R \cdot \mathbf{1}\{\hat{r}(x, \mathbf{y}) \neq r_{f_*}(x, \mathbf{y})\}$ Thus, by Theorem 3.2 for binary rewards, it holds that*

$$\mathbb{E}_{P_x} \mathbb{E}_{\mathbf{y} \sim \hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq$$

$$1 - [\mathbb{E}_{x,y \sim P_x \times \pi_0(\mathbf{y}|x)} 2R \cdot N \cdot \mathbf{1}\{\hat{r}(x, \mathbf{y}) \neq r_{f_*}(x, \mathbf{y})\} + \mathbb{E}_X (1 - P_0(x))^N]$$

Now note that the reward fitting stage which produces $\hat{r}$ from Equation 2.4 is standard ERM for a binary classification. Since by assumption the reward distribution is realizable by $\mathcal{R}_{\mathcal{F}}$, we may apply Proposition A.8 to say that with probability $1 - \delta$ over a draw of training set $D_n : \{(x^i, \mathbf{y}^i), r_{f_*}(x^i, \mathbf{y}^i)\}_{i=1}^n \sim (P_x \times \pi_0(\mathbf{y}|x) \times r_{f_*}(x, \mathbf{y}))^{\otimes n}$ it holds that

$$\mathbb{E}_{P_x} \mathbb{E}_{\mathbf{y} \sim \hat{\pi}_{BoN}(\mathbf{y}|x)} r_{f_*}(x, \mathbf{y}) \geq$$

$$1 - [\mathbb{E}_{x,y \sim P_x \times \pi_0(\mathbf{y}|x)} \frac{2N \cdot R \cdot [\mathrm{Ndim}(\mathcal{R}_{\mathcal{F}}) \log(|R| \cdot \mathrm{Ndim}(\mathcal{R}_{\mathcal{F}})) + \log(1/\delta)]}{n} + \mathbb{E}_X (1 - P_0(x))^N]$$

B.2 Lowerbound

Proof of Theorem 3.6. Consider the class of end token rewards $\mathcal{R}_{\mathcal{F}}$ induced by the class $\mathcal{F}$.

$$\mathcal{R}_{\mathcal{F}} = \{r_f : \Sigma' \times \Sigma^T \to \Sigma, \quad r_f(x \circ y) = \mathbf{1}\{y[-1] = f^{\mathrm{AR}}(x)[-1]\} | f \in \mathcal{F}\}$$

Likewise, consider the class of functions

$$\mathcal{F}_{\mathrm{et}} = \{f_{\mathrm{et}} : \Sigma' \to \Sigma, \quad f_{et}(x) = f^{\mathrm{AR}}(x)[-1] | f \in \mathcal{F}\}$$

Thus, by Theorem E.1. of Joshi et al. (2025) there exists a class $\mathcal{F}$ with $\mathrm{VC}(\mathcal{F}) = d$ such that the class $\mathcal{F}_{\mathrm{et}}$ shatters the set of points $\mathcal{X} = \{x_1, \ldots, x_{d \cdot T}\}$ with $x_i = (1 \circ \text{bit-respresentation}[i])$.

We can consider the distribution $P_x = \mathrm{Unif}[\mathcal{X}]$ and the base policy π_0 to satisfy $\pi_0^{\mathrm{AR}}(\mathbf{y}|x) = \mathrm{unif}[\Sigma^T]$. Through out, we will fix some $f^* \in \mathcal{F}$ to be the target function. Finally, assume that during the reward modeling step, the only observes at most $n = dT/2$ points $D^n = (z_i, r(z_i))_{i=1}^n$. At the reward modeling stage the learner selects

$$\hat{r} = \arg\min_{r \in \mathcal{R}_{\mathcal{F}}} \sum_{i=1}^n \mathbf{1}\{r_f(x_i, y_i) \neq r_{f_*}(x_i, y_i)\}.$$

By assumption, if multiple functions exist in the argmin set, ties are broken at random. Let $\mathcal{S} = \mathcal{X} \backslash D_X^n$ be the set of x in $\mathcal{X}$ that do not appear in D^n. Because the class $\mathcal{F}$ shatters $\mathcal{X}$, for any *fixed* x in $\mathcal{S}$ we can fully partition $\mathcal{F}$ into pairs (f, f') such that f, f' only disagree on x. By definition, note that for any such pair we

have $\sum_{i=1}^n \mathbf{1}\{r_f(x_i,y_i) \neq r_{f_*}(x_i,y_i)\} = \sum_{i=1}^n \mathbf{1}\{r_{f'}(x_i,y_i) \neq r_{f_*}(x_i,y_i)\}$, and in particular, half of all the reward models in the argmin set must satisfy $\hat{r}(x,y) = \mathbf{1}\{r_{f^*}(x,y)=0\}$ for the fixed $x \in \mathcal{S}$.

The quantity of interest for lower bounding BoN is

$$\mathbb{E}_{P_{\mathcal{D}}^n}\mathbb{E}_{\mathbf{y}\sim\pi_0}\mathbb{E}_{P_X}\mathbf{1}\{\hat{r}_f(x,y)=1, r_{f^*}(x,y)=0\} = \mathbb{E}_{P_{\mathcal{D}}^n}\frac{1}{2dT}\sum_{i=1}^{dT}\mathbf{1}\{\hat{r}_f(x_i,y_i) \neq r_{f^*}(x_i,y_i)\}.$$

In the last line we plugged in the distbutions for P_x and π_0 and dropped all the y_i that result in a $r_{f_*}(x,y)=1$. Since half of all the reward models in the argmin set must satisfy $\hat{r}(x,y) = \mathbf{1}\{r_{f^*}(x,y)=0\}$ for $x \in \mathcal{S}$, and $|\mathcal{S}|$ is at least $1/2$ by the assumption on n we see that $\mathbb{E}_{P_{\mathcal{D}}^n}\mathbf{1}\{\hat{r}_f(x_i,y_i) \neq r_{f^*}(x_i,y_i)\} \geq \frac{1}{4}$. Thus we have shown that

$$\mathbb{E}_{P_{\mathcal{D}}^n}\mathbb{E}_{\mathbf{y}\sim\pi_0}\mathbb{E}_{P_X}\mathbf{1}\{\hat{r}_f(x,y)=1, r_{f^*}(x,y)=0\} \geq 1/8$$

It is easy to see that this implies $\mathbb{P}(\mathbb{E}_{\mathbf{y}\sim\pi_0}\mathbb{E}_{P_X}\mathbf{1}\{\hat{r}_f(z)=1, r_{f^*}(z)=0\} \geq 1/16) \geq 1/16$ over draws of D^n.

The final step is to show that if the probability of a false positive draw from π_0 is bounded below, then BoN will perform poorly. To that end, let S denote the set of points in $z_1,\ldots,z_{dT}$ that satisfy $\hat{r}(z)=1, r_{f_*}(z)=0$ and note that for $c \in [0,N]$ we have

$$\mathbb{P}_{\mathbf{y}\sim\hat{\pi}_{\text{BoN}}}(r_{f_*}(\mathbf{y})=0) \geq c/N \cdot \mathbb{P}_{\mathbf{y}_i\sim\pi_0}(|\{\mathbf{y}_1,\ldots\mathbf{y}_N : \mathbf{y}_i \in S\}| \geq c)$$

From which we can pick $c = 1/16 \times N$ and that for this choice of c note that $\mathbb{P}_{\mathbf{y}_i\sim\pi_0}(|\{\mathbf{y}_1,\ldots\mathbf{y}_N : \mathbf{y}_i \in S\}| \geq c)$ is minimzed at $N=1$ to complete the proof. □

C SFT PROOFS

C.1 Lower Bounds

Proof of Theorem 4.1. Let $P_X(\{0,1\})=1$, and assume that $f_*^{\text{AR}}(\{0,1\}) = \{0,0,\ldots,0\}$. Take $\mathcal{F}$ to be a function class with two elements $\{f_1,f_2\}$. For convenience, also define $\text{CNT}(\sigma,b)$ to be the function which counts the instances of bit b in σ. Define the first function as follows:

$$f_1(\sigma) = \begin{cases} 1 \text{ if } \text{CNT}(\sigma,1) \geq \text{CNT}(\sigma,0) \\ 0 \text{ otherwise} \end{cases}$$

Note that if we define $y_i = f^*(x_i)$ then $\sum_i \sum_t \mathbf{1}\{f^1(x \circ y[:t]) \neq y[t]\} = n$. Next define the second function as

$$f_2(\sigma) = \begin{cases} 1 \text{ if } \text{CNT}(\sigma,1) = 1 \\ 0 \text{ if } \text{CNT}(\sigma,1) \geq 2 \end{cases}$$

Now note that $\sum_i \sum_t \mathbf{1}\{f^2(x,y[:t]) \neq y[t]\} = nT$ so the next token prediction ERM step will always select f_1. Note, however, that $f_{1\ \text{AR}}((0,1)) = (0,1,1,\ldots,1)$ so the expected reward of $\hat{f}_{\text{ntp}}^{\text{AR}}$ is zero.

On the other hand note that $f_2^{\text{AR}}((0,1)) = (1,0,0,\ldots,0)$ so that point $2\text{sup}_{f\in\mathcal{F}}\mathbb{E}_{P_X} r_{f_*}(x, f^{\text{AR}}(x)) = 1$. □

C.2 Upper Bounds

Proof of Theorem 4.2. Consider a draw of the training data set $(x^i,(y_1^i,\ldots y_T^i))$ the function $\hat{f}_{\text{ntp}}$ produced in the erm/ consistency stages must satisfy $\hat{f}_{\text{ntp}}(x^i, y_{<t}^i) = y_t^i$ for $i \in [n]$ and $t \in [T]$.

Now consider the class of functions $f^{\text{AR}} \in \mathcal{F}^{\text{AR}} : \Sigma' \to \Sigma^T$, where $f^{\text{AR}}(\sigma) = (f(\sigma), f(\sigma, f(\sigma)), \ldots, f(f_{\text{ap}}^{\circ T-1}(\sigma)))$ for $f \in \mathcal{F}$ (this is just the class of autoregressive functions studied throughout the paper). Following the discussion of Section 4.2, the process of training $\hat{f}_{\text{ntp}}$ and then deploying $\hat{f}_{\text{ntp}}^{\text{AR}}$ is equivalent to picking

$$\hat{f}_{\text{ntp}}^{\text{AR}} \in \{f^{\text{AR}} \in \mathcal{F}^{\text{AR}} : f^{\text{AR}}(x^i) = \mathbf{y}^i; \quad i \in [n]\}$$

Of course in the realizable setting it is trivial to see that this is equivalent to ERM for the standard multilabel classification problem:

$$\hat{f}^{\mathrm{AR}} = \arg\min_{f^{\mathrm{AR}} \in \mathcal{F}^{\mathrm{AR}}} \sum_{i=1}^{n} \mathbf{1}\{f^{\mathrm{AR}}(x^i) \neq (y_1^i, \ldots, y_T^i)\}$$

For ease of notation let $\mathbf{y}^i$ denote the vector of responses to x^i. Consider the loss class $\mathcal{L}(\mathcal{F}^{\mathrm{AR}}) = \{z \to \mathbf{1}\{\mathbf{y} \neq f^{\mathrm{AR}}(x)\} | f \in \mathcal{F}\}$. Following some ideas from Theorem B.5 of Joshi et al. (2025) we will show that

$$\mathrm{VC}(\mathcal{L}(\mathcal{F}^{\mathrm{AR}})) \leq 3\mathrm{VC}(\mathcal{F}) \log(\frac{2T}{\log(2)})$$

Consider the set $\mathrm{pfx}(D) = \{(x^i, y_1^i, \ldots y_t^i) | i \in [n], t \in [T]\}$. Letting $\Gamma_{\mathcal{L}(\mathcal{F}^{\mathrm{AR}}))}(n)$ denote the growth function of the loss class of interest we have

$$\Gamma_{\mathcal{L}(\mathcal{F}^{\mathrm{AR}}))}(n) = |\{(\mathbf{1}\{\mathbf{y_1^*} \neq f^{\mathrm{AR}}(x_1^*)\}, \ldots \mathbf{1}\{\mathbf{y_n^*} \neq f^{\mathrm{AR}}(x_n^*)\}) | f \in \mathcal{F}\}|$$

$$\leq |\{(f(x^{*1}), f(x^{*1}, y_1^{*1}), \ldots, f(x^{*n}, y_1^{*n}, \ldots, y_{T-1}^{*n})) | f \in \mathcal{F}\}| = |\mathcal{F}(D^*)| \leq \Gamma_{\mathcal{F}}(nT)$$

The key inequality uses the fact that if there exists two functions f, f' such that $(\mathbf{1}\{\mathbf{y_1^*} \neq f^{\mathrm{AR}}(x_1^*)\}, \ldots \mathbf{1}\{\mathbf{y_n^*} \neq f^{\mathrm{AR}}(x_n^*)\}) \neq (\mathbf{1}\{\mathbf{y_1^*} \neq f'^{\mathrm{AR}}(x_1^*)\}, \ldots \mathbf{1}\{\mathbf{y_n^*} \neq f'^{\mathrm{AR}}(x_n^*)\})$ then there exists i^*, t^* such that $f(x^{*i^*}, y_1^{*i^*}, \ldots, y_{t^*-1}^{*i^*}) \neq f'(x^{*i^*}, y_1^{*i^*}, \ldots, y_{t^*-1}^{*i^*})$.

Finally we make use of Sauers lemma to see

$$\Gamma_{\mathcal{L}(\mathcal{F}^{\mathrm{AR}}))}(n) \leq \Gamma_{\mathcal{F}}(nT) \leq (\frac{enT}{\mathrm{VC}(\mathcal{F})})^{\mathrm{VC}(\mathcal{F})}$$

Consulting the definition for VC dimension and some algebra will show the final we need on $\mathrm{VC}(\mathcal{L}(\mathcal{F}^{\mathrm{AR}}))$. We pause here to consider what this has bought us. We have shown that in the realizable setting if we select

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_i \mathbf{1}\{f(x^i, y_{<t}^i) \neq y_t\}$$

then by Proposition A.9 $\hat{f}$ satisfies the following: With probability at least $1 - \delta$, it holds that

$$\mathbb{P}_x[(\hat{f}(x), \hat{f}(x, \hat{f}(x)), \ldots \hat{f}_{\mathrm{ap}}^{\circ T}(x)) \neq (y_1, \ldots, y_T)] < \epsilon,$$

with

$$n(\epsilon, \delta) < c(\frac{\mathrm{VC}(\mathcal{F}) \log(T) \log(1/\epsilon) + \log(1/\delta)}{\epsilon}).$$

Then we may simply note that $\mathbb{E}_{x \sim P_X} r_{f_*}(x, \hat{f}^{\mathrm{AR}}(x)) \geq 1 - \mathbb{P}_x[\hat{f}_{\mathrm{ntp}}^{\mathrm{AR}}(x) \neq f_*^{\mathrm{AR}}(x)]$. □

Remark C.1. *Consider the case where $\Sigma = \{1, \ldots, |\Sigma|\}$. Through the same argument and using Natarajan's Lemma we have*

$$\Gamma_{\mathcal{L}(\mathcal{F}^{AR}))}(n) \leq \Gamma_{\mathcal{F}}(nT) \leq (|\Sigma|^2 \cdot n \cdot T)^{Ndim(\mathcal{F})}$$

Applying Lemma A.4 of (Joshi et al., 2025) we can see that

$$VC(\mathcal{F}^{AR}) \leq 3Ndim(\mathcal{F}) log_2(2\frac{Ndim(\mathcal{F})|\Sigma|^2 T}{eln(2)})$$

This demonstrates that even for general finite alphabets, in the agnostic case the performance of SFT has log dependence on T.

D PROOFS FOR SFT WITH RANDOM RESPONSES

Proof of Theorem 5.3. Consider fixed integers n, T and $D \triangleq |\mathcal{F}|$. We will discuss the following class of functions: The input space is $\mathcal{X} = \Sigma^{D\times T+1}$. The function class is $\mathcal{F} = \{f_0, f_1, \ldots, f_D\}$ where $f_0(x) = 0$ and for $1 \le d \le D$ it holds that $f_d(x) = x[-d \cdot T]$ (the $d \cdot T$'th from last element in the string x). Throughout we will also assume that $P_X \overset{d}{=} \text{unif}[\mathcal{X}]$ (the uniform distribution over all strings in $\mathcal{X}$), we assume that the reward is $r(x, y) = \mathbf{1}\{y[-1] = 0\}$ and that $\pi^*(y|x) = \text{unif}[\Sigma^{T-1}] \times \mathbf{1}\{y[-1] = 0\}$. Clearly, it holds that $\mathbb{E}_x r(x, f_0^{\text{AR}}(x)) = 1$ while for $1 \le d \le D$ we have that $\mathbb{E}_x r(x, f_d^{\text{AR}}(x)) = \mathbb{E}_x \mathbf{1}\{x[(-d+1)\cdot T - 1] = 0\} = \frac{1}{2}$. For $f \in \mathcal{F}$ consider the empirical next token prediction loss:

$$\hat{\mathcal{L}}_{\text{ntp}}(f) = \sum_{i=1}^{n}\sum_{t=1}^{T} \mathbf{1}\{f(x_i \circ y_i[:t]) \neq y_i[t]\}$$

The core of the proof is the following two facts:

1. The empirical ntp loss $\hat{\mathcal{L}}_{\text{ntp}}(f)$ has distribution

$$\hat{\mathcal{L}}_{\text{ntp}}(f_d) \overset{d}{=} \begin{cases} \text{Bin}(n(T-1), 1/2) & d = 0 \\ \text{Bin}(nT, 1/2) & 1 \le d \le D \end{cases}$$

2. For $d \neq d'$ the random variables $\hat{\mathcal{L}}_{\text{ntp}}(f_d)$ and $\hat{\mathcal{L}}_{\text{ntp}}(f_{d'})$ are independent.

The first fact follows directly from the assumption on $\pi^*(y|x)$. The second fact is more non-trivial. First suppose that $d > d' > 0$, then it holds that $\hat{\mathcal{L}}_{\text{ntp}}(f_d) = \sum_i |\{t : x^i[-d\cdot T : -d\cdot(T-1) - 1][t] = y^i[t]\}|$ and $\hat{\mathcal{L}}_{\text{ntp}}(f_{d'}) = \sum_i |\{t : x^i[-d'\cdot T : -d'\cdot(T-1) - 1][t] = y^i[t]\}|$ but by construction $x^i[-d\cdot T : -d\cdot(T-1) - 1]$ and $x^i[-d'\cdot T : -d'\cdot(T-1) - 1]$ are independent draws from $\text{unif}[\Sigma^T]$, thus the two empirical losses are independent (in fact the number of matches in each string is independent of y itself). For $d > 0$ a similar logic applies to the independence of $\hat{\mathcal{L}}_{\text{ntp}}(f_d)$ and $\hat{\mathcal{L}}_{\text{ntp}}(f_0)$; the number of zeros in y^i carries no information about the number of matches between y^i and $x^i[-d\cdot T : -d\cdot(T-1) - 1]$, since $x^i[-d\cdot T : -d\cdot(T-1) - 1]$ is distributed as $\text{unif}[\Sigma^T]$. Consulting the definition of $\hat{f}_{\text{ntp}}$ we see that for the reference policy π^*, the input space $\mathcal{X}$, the distribution P_X and the class $\mathcal{F}$ it holds that:

$$\mathbb{P}_{\mathcal{D}_n}(\mathcal{R}(\hat{f}_{\text{ntp}}) = \frac{1}{2}) = \mathbb{P}_{\mathcal{D}_n}\hat{f}_{\text{ntp}} \in \{f_d | 1 \le d \le D\} = \mathcal{G}(n, T, D),$$

$$\mathcal{G}(n, T, D) \triangleq \mathbb{P}\{\min_{Z_1,\ldots,Z_D} Z_d > Z_0; Z_d \overset{iid}{\sim} \text{bin}[nT, 1/2], Z_0 \sim \text{bin}(n(T-1), 1/2); Z_0 \perp\!\!\!\perp Z_d\}$$

Let us let $\tilde{Z}_D = \min_{Z_1,\ldots,Z_D} Z_d$ with $Z_d \overset{iid}{\sim} \text{bin}[nT, 1/2]$. By Theorems A.15 and A.13 if $\log(D) > \log(nT+1) + \log(\delta^{-2})$ then we have that with probability at least $1-\delta$ we have that $\tilde{Z}_D/nT < \frac{1}{2}$ and $|\frac{\tilde{Z}_D}{nT} - \frac{1}{2}|^2 = \frac{1}{2}\text{KL}(\tilde{Z}||p) \in [\frac{\log(D)}{nT} - \Delta(p, \delta, nT), \frac{\log(D)}{nT} + \Delta(p, \delta, nT)]$ which implies that

$$\frac{\tilde{Z}_D}{nT} \le \frac{1}{2} - \sqrt{\frac{\log(D)}{nT}} + \sqrt{\frac{\log(\delta^{-2}) + \log(nT)}{nT}}$$

Likewise, with probability at least $1-\delta$ it holds that

$$\frac{Z_0}{nT} \ge \frac{n(T-1)}{2nT} - \frac{\sqrt{[\log(\delta^{-2}) + \log(n(T-1))]n(T-1)}}{nT}$$

Thus with probability at least $1-2\delta$ we have that

$$\frac{\tilde{Z}_D}{nT} - \frac{Z_0}{nT} \le \frac{1}{2} - \sqrt{\frac{\log(D)}{nT}} - \frac{n(T-1)}{2nT} + 2\sqrt{\frac{\log(\delta^{-2}) + \log(nT)}{nT}}$$

$$= \frac{1}{2T} - \sqrt{\frac{\log(D)}{nT}} + 2\sqrt{\frac{\log(\delta^{-2}) + \log(nT)}{nT}}$$

Or equivalently,

$$\frac{\tilde{Z}_D}{\sqrt{nT}} - \frac{Z_0}{\sqrt{nT}} \leq \sqrt{\frac{n}{T}} - \sqrt{nT\log(D)} + 2\sqrt{\log(\delta^{-2})} + 2\sqrt{\log(nT)}$$

And if $n < \frac{T\log(D)}{4}$, with probability at least $1-2\delta$ it holds that

$$\frac{\tilde{Z}_D}{\sqrt{nT}} - \frac{Z_0}{\sqrt{nT}} \leq \sqrt{\log(D)}(\frac{1}{2} - T + 2\log(\delta^{-2}) + 2\sqrt{2\log(T)})$$

So, we for an incorrect $f \in \{f_1, \ldots, f_D\}$ to be selected we require that $\log(D) > \log(\log(D)T/4 + 1) + \log(\delta^{-2})$ and $(\frac{1}{2} - T + 2\log(\delta^{-2}) + 2\sqrt{2\log(T)}) < 0$. The second holds for $T > 0$ if $\delta = 3/4$ while the first holds if $\log(D) > \log(2) + 1/2 + \log(\log(D)/4) + \log(T/4)$. □

Proof of Proposition 5.4. Note that the reward is bounded below by $1 - \mathbb{P}_{\mathcal{D}_n}(\hat{f}_{\text{ntp}} \neq f_*)$, and by Assumption 5.2 it holds for $\epsilon < 1/4$ that

$$\mathbb{P}_{\mathcal{D}_n}(\hat{f}_{\text{ntp}} \neq f_*) \leq \mathbb{P}_{\mathcal{D}_n}(\exists f \in \mathcal{F} \text{ s.t. } |\hat{\mathcal{L}}(f) - \mathbb{E}\hat{\mathcal{L}}(f)| > \epsilon) \leq |\mathcal{F}|e^{-\frac{n}{T}\epsilon^2}$$

So one can pick $\epsilon = \sqrt{\frac{\log(|F|/\delta)T}{n}}$ to arrive at the result. □