Distribution Prompting: Understanding the Expressivity of Language Models Through the Next-Token Distributions They Can Produce

Haojin Wang[1]* **Zining Zhu** [2] **Freda Shi** [1,3]
[1] University of Waterloo [2] Stevens Institute of Technology
[3] Vector Institute, Canada CIFAR AI Chair
{s2323wang,fhs}@uwaterloo.ca zzhu41@stevens.edu

Abstract

Autoregressive neural language models (LMs) generate a probability distribution over tokens at each time step given a prompt. In this work, we attempt to systematically understand the probability distributions that LMs can produce, showing that some distributions are significantly harder to elicit than others. Specifically, for any target next-token distribution over the vocabulary, we attempt to find a prompt that induces the LM to output a distribution as close as possible to the target, using either soft (Li and Liang, 2021) or hard (Wallace et al., 2019) gradient-based prompt tuning. We find that (1) in general, distributions with very low or very high entropy are easier to approximate than those with moderate entropy; (2) among distributions with the same entropy, those containing "outlier tokens" are easier to approximate; (3) target distributions generated by LMs—even LMs with different tokenizers—are easier to approximate than randomly chosen targets. These results offer insights into the expressiveness of LMs and the challenges of using them as probability distribution proposers.

1 Introduction

Pretrained language models (LMs) have acquired extensive knowledge (Zhao et al., 2023) and shown strong performance across various tasks (Kaplan et al., 2020; Brown et al., 2020). Their impressive generative capabilities rely heavily on algorithms that determine the next token (Finlayson et al., 2024a, *inter alia*). As a foundation, these algorithms typically involve the softmax operation (Bridle, 1989), which computes the probability distribution of next tokens using the product of hidden states and token embeddings. However, such operation leads to the softmax bottleneck (Yang et al., 2018), which restricts the output distribution to a low-dimensional subspace, thereby limiting LM capability to express certain probability distributions. For example, Chang and McCallum (2022) found that when predicting the next-word probabilities given ambiguous contexts, GPT-2 (Radford et al., 2019) is often incapable of assigning the highest probabilities to nonsynonymous candidates. In another line, Finlayson et al. (2024b) found that by leveraging the limitations in LM expressiveness, we can efficiently extract proprietary information about API-protected LMs. These findings naturally raise a research question: are there general properties of the next-token probability distribution that can be easily elicited from LMs through prompting?

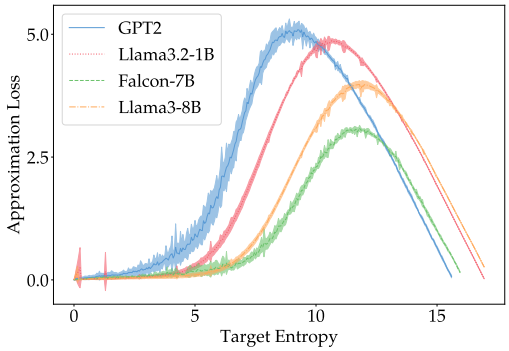


Figure 1: The Kullback–Leibler divergence between the target and approximated distributions for different entropy levels (in bits), using soft prompt tuning (Li and Liang, 2021) with different pretrained LMs as the backbone. The shaded regions represent the standard error bands.

To answer this question, we conduct a systematic analysis of the expressiveness of LMs, in terms of approximating different target probability distributions. Without changing the parameters of LMs, we aim to find prompts that lead to the output of a probability distribution as close as possible to a target distribution. This design isolates the intrinsic expressivity of pretrained LMs, since we probe

*Now at the University of Illinois Urbana-Champaign. Work done during visiting term at the University of Waterloo.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 20916–20929
November 4-9, 2025

their behavior by varying only the prompts while keeping model parameters fixed. Apparently, due to inefficiency for enumerating the infinitely many prompts as triggers for approximating the target distribution, we use modern optimization techniques. Through either soft (Li and Liang, 2021) or hard (Wallace et al., 2019)[1] prompt tuning across different initializations, we measure the difficulty of an approximation using the minimum Kullback–Leibler (KL) divergence between the target and the approximated distributions from different prompt initializations.

We conduct experiments on multiple pretrained LMs, including GPT-2 (Radford et al., 2019), Llama3.2-1B (Dubey et al., 2024), Falcon-7B (Almazrouei et al., 2023), and Llama3-8B (Dubey et al., 2024). We identify that these models share general properties for the next-token probability distributions: First, as illustrated in Fig. 1, the difficulty of approximating a random target distribution is approximately a unimodal function of the target entropy, where the approximation loss increases with the entropy of the target distribution up to a point, then decreases across all models. Second, for the distributions with moderate entropy, the existence of outlier tokens (i.e., those receiving much higher probability values than the other tokens) makes the approximation easier. Finally and unsurprisingly, LM-generated distributions are easier to approximate regardless of the entropy values, and can be effectively transferred across LMs. Our results introduce novel evidence toward a better understanding of the expressiveness of LMs, highlighting the opportunities and challenges in using LMs as probability distribution proposers.

2 Related Work

Expressiveness of LMs. The output probability distributions of LMs are restricted into a linear subspace of full output space because of softmax bottleneck (Yang et al., 2018). A natural consequence is that any collection of linearly independent LM outputs can form a basis for this space that expresses any probability distribution output by LMs (Finlayson et al., 2024a; Finlayson et al., 2024b). While it was previously believed that a large hidden state dimension would overcome the softmax bottleneck in LMs, Chang and McCallum (2022) found that linear dependencies, such as word analogy (Turney, 2008), in token embeddings (Mikolov et al., 2013; Ethayarajh et al., 2019) restrict the expressiveness of the output probability distributions. These findings motivate us to investigate the properties of next token probability distributions to better understand LM expressiveness, which, to the best of our knowledge, has not been systematically studied before.

Prompt tuning with frozen LMs. A large body of previous research has demonstrated that prompting can effectively solve a wide range of tasks (Brown et al., 2020; Schick and Schütze, 2021; Gao et al., 2021; Sun et al., 2021; Dong et al., 2024, *inter alia*). Among them, prompt tuning, which learns the prompt parameters from downstream tasks while keeping the main LM parameters frozen, becomes a lightweight yet effective technique to obtain task-specific prompts (cf. Liu et al., 2023). There has been two major branches of tuning methods.

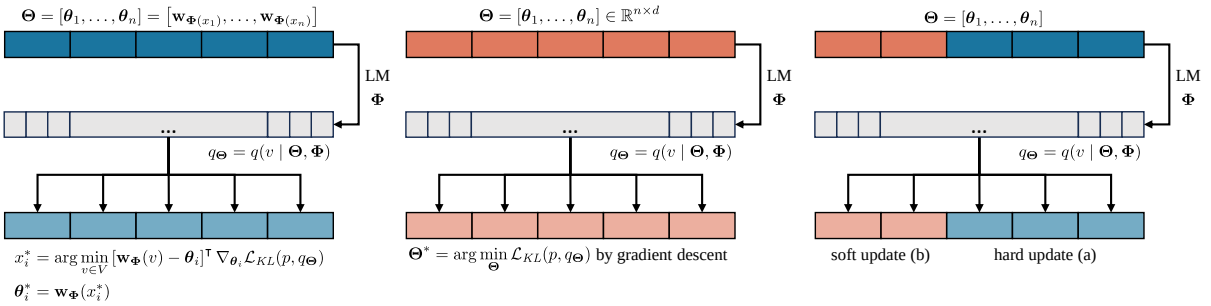
- **Hard prompt tuning**, or discrete prompt tuning, which uses the non-contextualized representation of vocabulary tokens as prompts (Ebrahimi et al., 2018; Jiang et al., 2021; Haviv et al., 2021, *inter alia*). Among them, Wallace et al. (2019) proposed a gradient-based search over token prefixes to trigger certain output tokens.
- **Soft prompt tuning**, or continuous prompt tuning, which finds a sequence of continuous embeddings as the prompt for downstream tasks, while not requiring the prompts to correspond to vocabulary tokens (Li and Liang, 2021; Qin and Eisner, 2021; Liu et al., 2022, *inter alia*). These soft prompts can be directly optimized through gradient descent to the embedding space.

Neither branch has been used to study the expressiveness of LMs in eliciting different probability distributions, which is the focus of our work. In this work, we adapt the approaches of Wallace et al. (2019) and Li and Liang (2021) to find prompts that lead to specific probability distributions.

3 Problem Formulation

Consider a Transformer-based (Vaswani et al., 2017) autoregressive language model $\mathcal{M}_{\mathbf{\Phi}}$ (e.g., Llama3; Dubey et al., 2024) parametrized by $\mathbf{\Phi}$. Let $\mathbf{x} = [x_1, x_2, ..., x_n] \in \mathbb{Z}^n_{\geq 0}$ be the input text with n tokens. The language model $\mathcal{M}_{\mathbf{\Phi}}$ first converts $\mathbf{x}$ into non-contextualized token embeddings $\mathbf{W}_{\mathbf{\Phi}}(\mathbf{x}) = [\mathbf{w}_{\mathbf{\Phi}}(x_1), \ldots, \mathbf{w}_{\mathbf{\Phi}}(x_n)] \in \mathbb{R}^{n \times d}$

[1]The key difference between soft and hard prompts is that soft prompts can be continuous embeddings that do not align to any token in the vocabulary, while hard prompts must correspond to vocabulary entries.



(a) Hard embeddings. (b) Soft embeddings. (c) Hybrid embeddings.

Figure 2: Illustration of the prompt tuning frameworks adapted in this work. For hard (Fig. 2a) and hybrid (Fig. 2c) embeddings–based frameworks, we optimize the prefix embeddings from left to right; forthe soft embeddings-based framework (Fig. 2b), we optimize the embeddings directly through gradient descent. Best viewed in color: the dark colors represent the initial embeddings, and the light colors represent the optimized ones.

through a look-up table, where d is the embedding size. The embeddings are then fed into the Transformer to produce the logits over the vocabulary $\mathbf{z} = [z_1, \ldots, z_n] \in \mathbb{R}^{|\mathcal{V}|}$, where $|\mathcal{V}|$ is the size of the vocabulary $\mathcal{V}$. The logits are later on normalized with a softmax operator to produce the probability distribution $\mathbf{q}$ over the vocabulary, where for $v \in \mathcal{V}$, the probability of the next token being v given the prefix $\mathbf{x}$ is given by:[2]

$$q_v = \text{softmax}(\mathbf{z})_v = \frac{\exp(z_v)}{\sum_{v' \in \mathcal{V}} \exp(z_{v'})}.$$

Here, $\mathbf{q}$ is essentially the output of the function (i.e., the language model) $\mathcal{M}_{\mathbf{\Phi}}$ given the input $\mathbf{x}$; that is, we obtain the next-token distribution $\mathbf{q} \in \mathbb{R}^{|\mathcal{V}|}$ by

$$\mathbf{q} = \mathcal{M}_{\mathbf{\Phi}}(\mathbf{x}).$$

We aim to find an input sequence $\mathbf{x}$, where $\mathbf{q} = \mathcal{M}_{\mathbf{\Phi}}(\mathbf{x})$ is close enough to an arbitrary target distribution $\mathbf{p} \in \mathbb{R}^{|\mathcal{V}|}$. Here, since we adopt optimization techniques (Li and Liang, 2021; Wallace et al., 2019) to find $\mathbf{x}$, $\mathbf{x}$ can be viewed as parameters $\mathbf{\Theta}$ ($\mathbf{\Theta} = \mathbf{x}$) optimized by the algorithm (Fig. 2). The objective of the optimization problem is to minimize the KL divergence between the target distribution $\mathbf{p}$ and the output distribution $\mathbf{q}$:

$$\min_{\mathbf{\Theta}} \mathcal{L}_{\text{KL}}(\mathbf{p}, \mathbf{q}; \mathbf{\Theta})$$
$$\Leftrightarrow \min_{\mathbf{\Theta}} \sum_{v \in \mathcal{V}} p(v) \log \frac{p(v)}{q(v; \mathbf{\Theta}, \mathbf{\Phi})}.$$

Lower KL divergence indicates better approximation of the target distributions. Varying across random seeds, the minimum KL-divergence loss $\mathcal{L}_{\text{KL}}(\mathbf{p}, \mathbf{q}; \mathbf{\Theta})$ gives an approximation on how difficult it is to elicit the target distribution $\mathbf{p}$ from the LM $\mathcal{M}_{\mathbf{\Phi}}$—lower loss indicates easier elicitation.

4 Methods

In this section, we introduce our method to synthesize target probability distributions (§4.1) and the prompt tuning frameworks (§4.2).

4.1 Target Distribution Construction

Our work requires constructing a probability distribution $\mathbf{p}$ over the vocabulary given a specified entropy value. To achieve this, we start from a set of random logits $\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|}$, where $\mathcal{V}$ is the vocabulary of the LM. The probability distribution $p(v)$ defined by the logits $\mathbf{z}$ is given by:

$$p(v; \mathbf{z}) = \frac{\exp(z_v)}{\sum_{v' \in \mathcal{V}} \exp(z_{v'})}.$$

Given the desired target entropy h, we aim to find a set of logits $\mathbf{z}$ such that the entropy of the distribution $p(v; \mathbf{z})$ is close to h within a small margin $\varepsilon (\varepsilon > 0)$, by minimizing the following loss with gradient descent:

$$\mathcal{L}(\mathbf{z}; h) = ||H(p(\cdot; \mathbf{z})) - h||^2 \quad (1)$$
$$= \left|\left| - \left(\sum_{v \in \mathcal{V}} p(v; \mathbf{z}) \log p(v; \mathbf{z}) \right) - h \right|\right|^2,$$

where $H(\cdot)$ denotes the entropy of a probability distribution. We use gradient descent to minimize $\mathcal{L}$ to construct for target distribution $\mathbf{p}$.

However, entropy is a lossy descriptor of a probability distribution. For example, in Fig. 3, the two

[2] With a slight abuse of notations, we use v to denote both the token and the token index in the vocabulary.

distributions (one output by Llama3.2-1B (Dubey et al., 2024) and the other by our search algorithm based on Eq. (1)) have the same entropy yet are distinct from each other sharply. Considering the fact that LMs usually generate skewed distributions with a few tokens receiving much higher probability mass than the rest (Holtzman et al., 2020), we define *distributions with outliers*, of which a large amount of probability mass concentrates on a small set of tokens. Formally, a distribution with k outliers satisfies:

$$\sum_{v \in \mathcal{V}_o} p(v) \geq m - \delta, \tag{2}$$

where $\mathcal{V}_o$ denotes the $k (k \ll |\mathcal{V}|)$ pre-selected outlier tokens that receive distinctively larger probability mass than the rest; m is the theoretical upper bound of the concentrated probability mass to k tokens to ensure the feasibility of the entire distribution reaching the target entropy h—details of calculating m from k, h, and $|\mathcal{V}|$ can be found in App. A; δ is a small margin to ensure the feasibility of the optimization process, which we set to 0.01 in all experiments. Intuitively, Eq. (2) requires that a small set of k tokens account for most of the probability mass, thereby capturing our notion of outlier distributions. Here, m specifies the maximum feasible cumulative mass these tokens can hold under the entropy constraint, ensuring the construction remains mathematically valid.

We add a regularization term to the loss function in Eq. (1) to encourage increased probability mass on the outlier tokens. To construct an outlier distribution, we optimize the following loss function with gradient descent:

$$\mathcal{L}'(\mathbf{z}; h) = \alpha \cdot \mathcal{L}(\mathbf{z}; h) - \beta \sum_{v \in \mathcal{V}_o} p(v), \tag{3}$$

where α and β are hyperparameters that balance the two terms, and stop once Eq. (2) is satisfied. We will compare the difficulty of approximating the distributions generated by Eq. (1) and Eq. (3) in our experiments.

4.2 Prompt Tuning Frameworks

Hard prompt tuning. Intuitively, we aim to find a textual input that elicit the specified target probability distributions. Following Wallace et al. (2019), we use an iterative linear approximation of the gradient of the KL divergence loss w.r.t. the prefix embeddings $\mathbf{\Theta}$. We update the tokens in the prefix one by one from left to right, where each token x_i is updated to:

$$x_i = \arg\min_{x \in \mathcal{V}} [\mathbf{W}_{\mathbf{\Phi}}(x) - \boldsymbol{\theta}_i]^\top \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_{\text{KL}}(\mathbf{\Theta}). \tag{4}$$

Intuitively, at each step, we find an in-vocabulary token $x \in \mathcal{V}$ that best linearly approximates the gradient of the KL divergence loss w.r.t. the prefix embedding $\boldsymbol{\theta}_i$. We repeat this iterative update until the convergence of the loss, and obtain the hard prompt tuning result

$$\mathbf{\Theta}^*_{hard} = [\mathbf{w}_{\mathbf{\Phi}}(x_1), \mathbf{w}_{\mathbf{\Phi}}(x_2), ..., \mathbf{w}_{\mathbf{\Phi}}(x_{n-1})],$$

where x_i is the discrete token that appear in the prefix at position i in the final step.

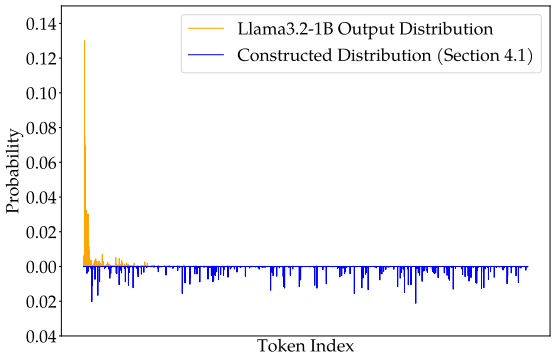


Figure 3: Two distributions, both with entropy 7.68 but have distinctively different shapes. Upper: the output probability distribution from prompting Llama3.2-1B with *'I love'*. Lower: The distribution from our search algorithm following the objective of Eq. (1).

Soft prompt tuning. The power of hard prompts is limited because they are restricted to vocabulary entries. Following Li and Liang (2021), we directly minimize the KL divergence loss w.r.t. the prefix embeddings $\mathbf{\Theta}$ in a continuous word vector space.

$$\mathbf{\Theta}^*_{soft} = \arg\min_{\mathbf{\Theta}} \mathcal{L}_{\text{KL}}(\mathbf{p}, \mathbf{q}; \mathbf{\Theta}). \tag{5}$$

We use AdamW (Loshchilov and Hutter, 2019) to optimize this objective. We include soft prompts to probe the upper bound of LM expressivity, since they allow unconstrained optimization in the embedding space; although they may not lie on the natural input manifold, we aim to study theoretical reachability rather than mimic typical usage.

Hybrid prompt tuning. To better understand the effect of different tuning methods, we introduce a hybrid tuning method that combines hard and soft prompts, where $\mathbf{\Theta}$ can be partitioned into two parts:

Θ_h for hard prompts and Θ_s for soft prompts, with some θ updated by hard prompts tuning and others by soft prompts tuning. We also optimize the hybrid embeddings in a left-to-right manner, where the hard embeddings are updated using Eq. (4), with soft embeddings updated using Eq. (5).

We compare the above three strategies in approximating target distributions. The general framework of our tuning methods can be found in Fig. 2.

5 Experiments

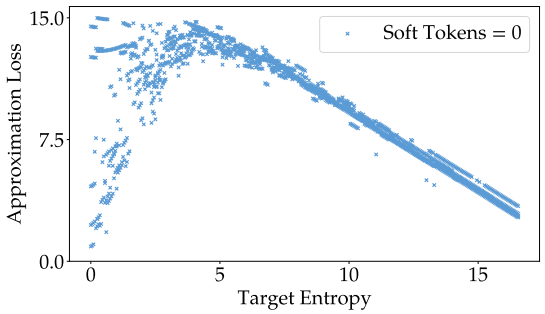
To study the general property of the expressiveness of an LM, we start by studying different probability distributions of different entropy (§ 5.1). We then analyze the LM approximation performance on outlier distributions (§ 5.2). Finally, we investigate target probability distributions as the output distribution of LM itself (§ 5.3), and analyze the results for approximating the variations of such distributions (§ 5.4). We set up our experiments across transformer-based LMs of different architectures and sizes, including GPT-2 (Radford et al., 2019), Llama3.2-1B (Dubey et al., 2024), Falcon-7B (Almazrouei et al., 2023), and Llama3-8B (Dubey et al., 2024). To mitigate the effect of randomness brought by target distribution construction algorithms in § 4.1, we obtain different distributions regarding the same entropy value. For each distribution, we use a different initialization for the prefix parameters. Detailed experiment setups can be found in App. B for reference.

5.1 Approximation of Vanilla Distributions

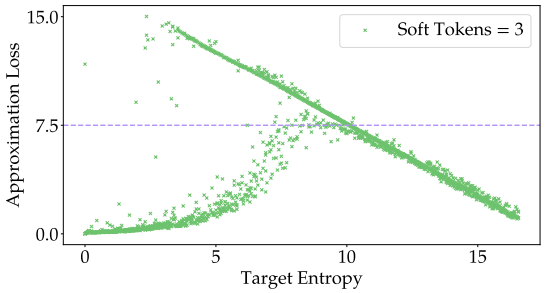
We perform prompt tuning to approximate target distributions constructed with Eq. (1), which we will refer to as vanilla distributions. We investigate the target entropy ranging from 0 to $\log|\mathcal{V}|$ with a step size 0.05, where $|\mathcal{V}|$ is the model's vocabulary size, and $\log|\mathcal{V}|$ is the theoretical upper bound for the vocabulary distribution.[3]

Fig. 1 presents the approximation loss of soft prompt tuning, plotted against the target entropy for different models. The approximation loss curve exhibits a consistent pattern across all models: increasing with target entropy up to a point, then decreasing across all models, which roughly forms a unimodal curve.

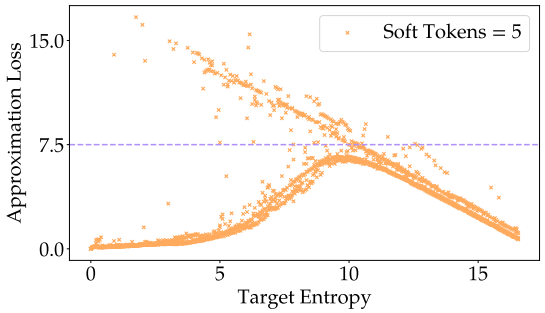
We then compare soft and hard prompt tuning frameworks (§ 4.2) on vanilla target distributions



(a) Hard prompt tuning.



(b) Hybrid prompt tuning.



(c) Soft prompt tuning.

Figure 4: Approximation loss plotted against the target entropy for different prompt tuning frameworks on Llama3.2-1B. For the distribution of each target entropy we used 5 different prompt initializations, and we keep prefix length to be 5.

(Fig. 4). As expected, soft prompts—being unconstrained by the vocabulary—consistently yield lower approximation losses than hard prompts. To further explore this gap, we adopt hybrid prompt tuning, where updates depend on the embedding source. Increasing the number of soft tokens in the prefix improves approximation, highlighting the benefit of soft tuning. We also find that different random initializations of prefix embeddings can lead to large variance in performance. This effect is especially pronounced in hybrid and soft prompt tuning (Figs. 4b and 4c). Echoing findings in Zhao et al. (2025), our results suggest that randomness plays a systematic role in loss minimization.

However, even when using pure soft prompts for these objectives, we still struggled to elicit cer-

[3]The proof is detailed in App. A.

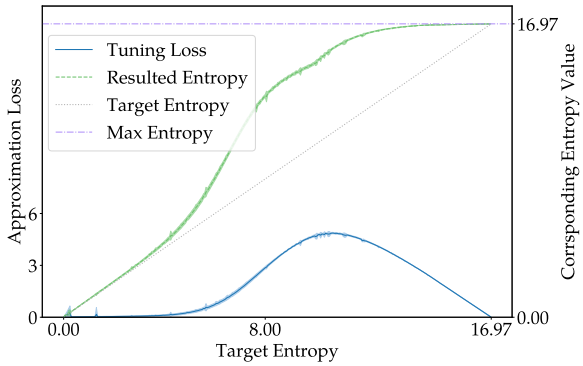
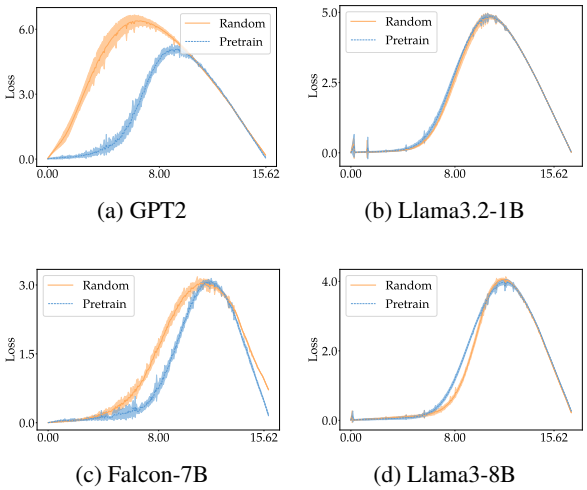


Figure 5: Approximation loss and the entropy for distribution output by tuned prompt plotted against the target entropy for soft prompt tuning on Llama3.2-1B. The figure shows the mean minimum loss within different initializations and corresponding entropy across distributions with the same target entropy (lines) and the 95% confidence interval (shaded region).



(a) GPT2 (b) Llama3.2-1B

(c) Falcon-7B (d) Llama3-8B

Figure 6: Approximation losses exhibit a consistent pattern as shown in Fig. 1 for models randomly initialized that excluded pretraining history. The orange-yellow line is the randomly initiated models, while the blue dash line corresponds to the pretrained models. The x-axis in the subplot stands for target entropy.

tain distributions. We present the result of a soft prompt tuning study on Llama3.2-1B in Fig. 5. The result shows systematic trends of target distributions regarding different ranges of entropy values. At low entropy, we have near-zero losses on soft prompt approximation, and the prompts tuned by our framework output a distribution with aligned entropy to the target. However, when the target distribution has a larger entropy, the losses of our training objective increase due to the failure of the model to approximate a lower-entropy distribution from the initializations—the entropy of the approximated distribution grows to be larger than the target entropy. Such a phenomenon in approximation loss is also observed for transformer-based models from different series and sizes (Fig. 1). When the parameters are randomly initialized, the approximation loss remains a unimodal curve with respect to the target entropy (Fig. 6), indicating that pretraining is not the cause of the unimodal difficulty phenomenon.

5.2 Approximation of Outlier Distributions

In this section, we perform prompt tuning to approximate target distributions characterized by outlier tokens, where these distributions are constructed according to the objective defined in Eq. (3).

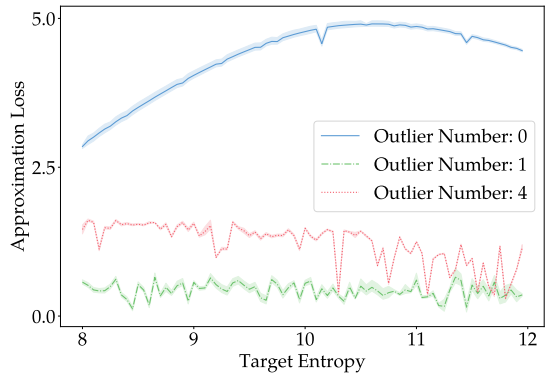
To isolate the effect of outlier tokens from token identity or ordering, we construct five target distributions for each configuration with fixed entropy and outlier count. To account for variability from prompt initialization, each experiment is repeated with five random initializations, and we report the minimum loss achieved. Final results are averaged over these minima across all distributions for each target configuration.

Since moderate-entropy distributions yield higher approximation losses in the vanilla setting (Fig. 5), we focus our analysis on this regime, where approximation is most challenging. Our results (Figs. 7a and 7b) show that distributions with outlier tokens incur lower losses than vanilla counterparts with similar entropy. This effect is strongest when a single token dominates the probability mass. In such cases, the tuned prompts also produce output distributions whose entropy better matches the target. These findings suggest that the presence of outliers in the target distribution may facilitate more effective prompt tuning by the LM.

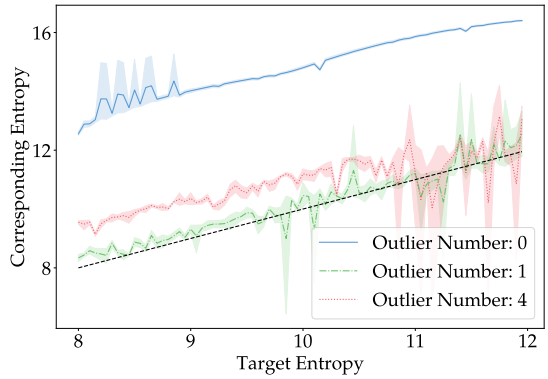
5.3 Approximation of LM-generated Distributions

Building on the findings of Morris et al. (2024), who demonstrate that input text can be largely reconstructed from output distribution logits using a trained inversion model, we hypothesize that LM-generated distributions are intrinsically easier to approximate than distributions we experimented with in §5.1, irrespective of their entropy levels.

To evaluate this hypothesis, we conduct a comparative analysis between vanilla target distributions (Eq. (1)) and target distributions derived from LM outputs. We conducted our experiments on

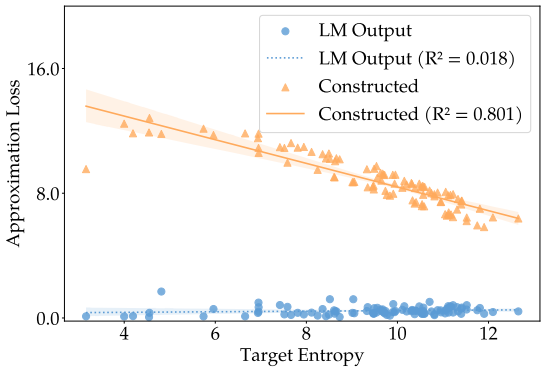


(a) Mean shuffle soft prompt approximation loss on outlier distributions and their 95% confidence interval (shaded region).

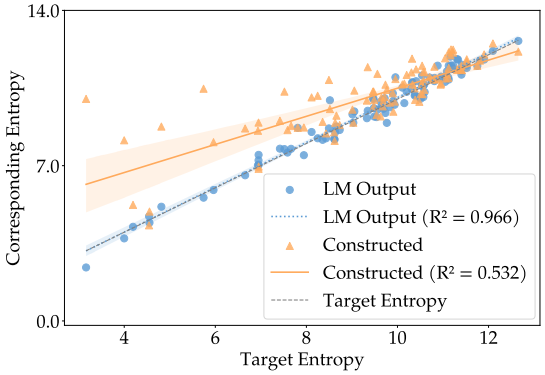


(b) Corresponding mean entropy for shuffle soft prompt approximation on outlier distributions and their 95% confidence interval (shaded region).

Figure 7: Outlier distribution loss and corresponding entropy of approximated distributions on Llama3.2-1B.



(a) Hard prompt tuning loss comparison between Llama3.2-1B output distributions and matched-entropy vanilla distributions.



(b) The comparison in corresponding entropy of hard prompt tuning between Llama3.2-1B output distributions and matched-entropy vanilla distributions.

Figure 8: The comparison of approximation performance across distribution pairs in §5.3 on Llama3.2-1B.

Llama3.2-1B. We construct target distributions by randomly sampling a prompt—i.e., a sequence of random token indices—and using the model's next-token output as the target distribution for evaluation. We compute the entropy of each LM-generated output distribution and use it as the target entropy in our training objective, as defined in Eq. (1). For each, we construct a corresponding vanilla next-token distribution with matched entropy. To mitigate initialization variance, we tune 25 prompt initializations per target and report the minimum loss. Hard prompt tuning is evaluated on 100 such distributions, each paired with a matched-entropy vanilla distribution.

We present the results in Fig. 8. As shown in Fig. 8a, we fit a regression line over the minimum losses and their corresponding entropies across initializations. Hard prompt tuning on LM-generated distributions consistently yields lower loss than on vanilla ones, with a low R^2 value indicating weak correlation between entropy and loss. Fig. 8b reports the entropy of the next-token distributions produced by tuned hard prompts for both types of targets. The approximated entropy closely aligns with the target for LM-generated distributions, whereas vanilla targets show larger deviations.

5.4 Approximation of Variations of LM-Generated Distributions

To assess whether LM-generated distributions are not only easier to approximate, as demonstrated in §5.3, but also robust under distributional variations, we extend our analysis to include shuffled LM-generated distributions and distributions derived from the outputs of other models.

We begin by evaluating how sensitive approximation performance is to token assignments in the target distribution. To do so, we randomly shuffle token indices in LM-generated distributions from §5.3, preserving their original probabilities. These are referred to as shuffled variants, with each distribution shuffled five times to reduce randomness.

Following the prior setup, soft prompt tuning

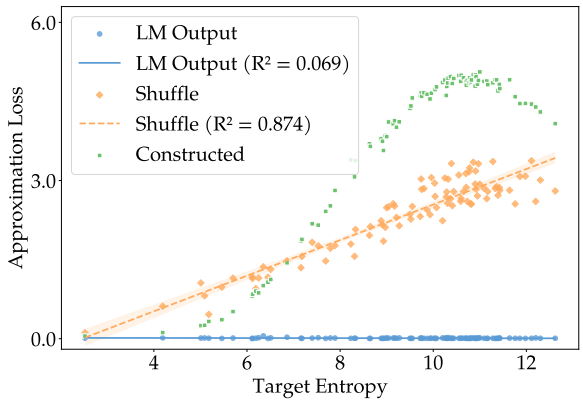


Figure 9: Soft prompt approximation loss is compared across LM-generated, shuffled LM-generated, and vanilla target distributions on Llama3.2-1B.

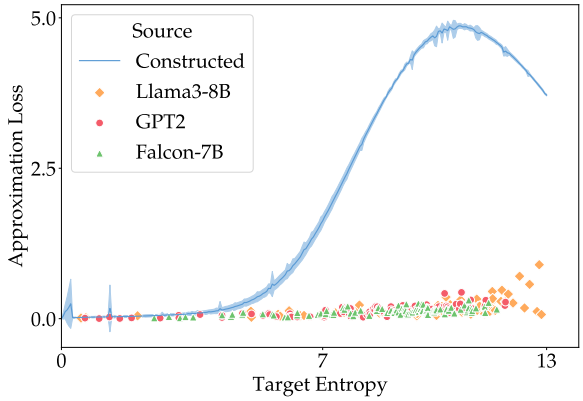


Figure 10: Soft prompt tuning loss on migrated distributions from various source models is evaluated on Llama3.2-1B.

is applied using five random initializations, and the minimum loss is reported. Fig. 9 shows the results for shuffled variants alongside original and matched vanilla distributions. Notably, shuffled variants still outperform vanilla ones—especially in the moderate entropy regime where approximation is harder (Fig. 5).

We further examine the transferability of LM-generated distributions by testing whether output probabilities can be effectively approximated across models. A **source LM** generates target distributions using the method in §5.3, while a **tuning LM** performs prompt tuning on these distributions. The target distribution for the tuning LM is constructed by mapping shared tokens from the source LM and renormalizing their probabilities to ensure a valid distribution.

Soft prompt tuning is used to approximate the migrated distributions from the source models, with Llama3.2-1B as the tuning LM. Results in Fig. 10 show that, despite not being generated by the tuning model, the migrated distributions achieve lower approximation loss than matched-entropy vanilla distributions. This suggests that, despite differences in tokenization and architecture, LM outputs share underlying properties that support more effective approximation.

6 Conclusion and Discussion

In this work, we systematically study the expressiveness of LMs in approximating target next-token probability distributions under prompt tuning frameworks. We first analyze various prompt tuning methods and find that continuous soft prompts yield the best approximation performance. From an information-theoretic perspective, we design target distributions with varying entropy and observe that LMs struggle most with those of moderate entropy. Interestingly, the presence of outlier tokens—rare tokens with notably high probabilities—makes approximation easier. Moreover, distributions generated by LMs themselves are generally easier to approximate, even under hard prompt tuning, despite having moderate entropy. These distributions also remain relatively easy to approximate when perturbed. Overall, our findings reveal general properties of LM expressiveness that can inform more fine-grained future studies.

A natural hypothesis is that the difficulty LMs face in approximating moderate-entropy distributions stems from their training objective: minimizing cross-entropy with a one-hot target encourages low-entropy outputs. However, our experiments reject this explanation—both randomly initialized and pretrained models exhibit similar behavior. Moreover, if the hypothesis were correct, approximation loss would increase monotonically with target entropy, which our results do not support. We hypothesize that such particular difficulty for approximating moderate-entropy distributions may come from either the Transformer (Vaswani et al., 2017) architecture or the softmax operator in the decoding stage. We leave the investigation of the root cause of this phenomenon for future work.

While language models are trained to predict the next token, the probabilistic nature of the output distribution makes it naturally a distribution proposer, from the perspectives of subjective randomness (Bigelow et al., 2023) and computational creativity (Peeperkorn et al., 2024), among others. Our findings also have implications for interpretability

and control: understanding which distributions are harder or easier to elicit can guide robust prompt design, distribution-aware decoding, and fine-grained behavioral analysis (e.g., POS-specific in App. C). Beyond prompting, such insights may support failure detection, model evaluation, and the development of decoding strategies that explicitly account for the reachable distribution space. Our work provides a first step to systematically analyze and understand the next-token distribution by LMs, and we hope this perspective spurs further exploration of LM expressivity and its applications.

Limitations

Our study could be extended in the following ways. First, we depend on the investigation of different target probability distributions entirely from the perspective of entropy, which is still a minor step compared to the whole picture of understanding the expressiveness of LMs. Second, we notice that the initialization of the prompt tuning actually matters in approximating certain distributions, which is also worthy of research. Ultimately, the root cause of the observed phenomenon remains unclear, warranting further investigation to better understand its underlying mechanisms. App. C presents a case study on instruction-tuned variants and POS-specific distributions. We hope this work motivates a deeper exploration of LM distributional behavior and guides the design of interpretable models.

Acknowledgements

We thank Ziqiao Ma and Jiayuan Mao for constructive discussions at the early stage of this work, as well as the anonymous reviewers and area chairs for the valuable feedback. This work is supported in part by NSERC RGPIN-2024-04395 and a Canada CIFAR AI Chair award to FS.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Eric J Bigelow, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, and Tomer Ullman. 2023. Subjective randomness and in-context learning. In *UniReps: the First Workshop on Unifying Representations in Neural Models*.

John Bridle. 1989. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Haw-Shiuan Chang and Andrew McCallum. 2022. Softmax bottleneck makes language models unable to represent multi-mode word distributions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8048–8073, Dublin, Ireland. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu,

Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, Florence, Italy. Association for Computational Linguistics.

Matthew Finlayson, John Hewitt, Alexander Koller, Swabha Swayamdipta, and Ashish Sabharwal. 2024a. Closing the curious case of neural text degeneration. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. 2024b. Logits of api-protected llms leak proprietary information. *Preprint*, arXiv:2403.09539.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. BERTese: Learning to speak to BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3618–3623, Online. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. 2024. Language model inversion. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 319–327. Association for Computational Linguistics.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137.

P. D. Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rosie Zhao, Tian Qin, David Alvarez-Melis, Sham Kakade, and Naomi Saphra. 2025. Distributional scaling laws for emergent capabilities. *arXiv preprint arXiv:2502.17356*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

A Entropy

For a probability distribution $p = (p_1, p_2, ..., p_n)$, the entropy for this probability distribution is defined as:

$$H(p) = -\sum_{i=1}^{n} p_i \log(p_i)$$

while p_i represents the probability value of event i and $\sum_{i=1}^{n} p_i = 1$.

For given constraints that the sum of the probability value for each event is 1, the probability distribution p that maximizes entropy is the uniform distribution. To prove this, we introduce a Lagrange multiplier λ for this constraint and form the Lagrange function:

$$\mathcal{L}(p_1, p_2, ..., p_n, \lambda) = -\sum_{i=1}^{n} p_i \log p_i + \lambda(\sum_{i=1}^{n} p_i - 1)$$

To maximize $\mathcal{L}$, we take partial derivative of $\mathcal{L}$ with respect to each p_i and set it equal to 0:

$$\frac{\partial \mathcal{L}}{\partial p_i} = -\log p_i - 1 + \lambda = 0.$$

Since this must hold true for each p_i, we conclude that each p_i should be the same. To be more specific, we can conclude that to take the maximum value, p_i must equal to c, where c is a constant. To make this satisfy the constraints previously, we have:

$$p_i = \frac{1}{n}.$$

Therefore, the probability distribution that maximizes entropy is the uniform distribution, where each $p_i = \frac{1}{n}$, and the maximum entropy for a distribution is $\log(n)$.

We now establish the detailed calculation for value m defined as the upper boundary for probability mass concentrate on outlier tokens in Eq. (3). Because of the linearity of entropy $H(p)$, we take the probability distribution of outlier tokens and the rest into separate consideration. For k outlier tokens, the maximum entropy is reached when there is uniform distribution for them. As we assume that m probability mass has fallen on these tokens, the maximum entropy for outlier tokens is $-m \log(\frac{m}{k})$. Similarly, for the rest tokens, the maximum entropy is $-(1-m)\log(\frac{1-m}{n-k})$. So in case we desire to find a probability distribution with the existence of k outliers to meet a specific entropy e, we must satisfy the inequality:

$$-m \log(\frac{m}{k}) - (1-m)\log(\frac{1-m}{n-k}) \geq e.$$

Also according to the definition of outliers, we notice that $k \ll n$. So given the outlier number k fixed, the LHS of the inequality decreases as outlier probability m increases. Also noticed that for a set of small enough k, the most restrictive condition is achieved when $k = 1$. Then, we change our objective to solve the equation as:

$$-m \log(m) - (1-m)\log(\frac{1-m}{n-1}) = e.$$

We will gradually find a value m that makes this inequality hold as equality, and then round it down to 0.01. In practice, we solve the above equation regarding different target probability distributions containing outliers with target entropy value e.

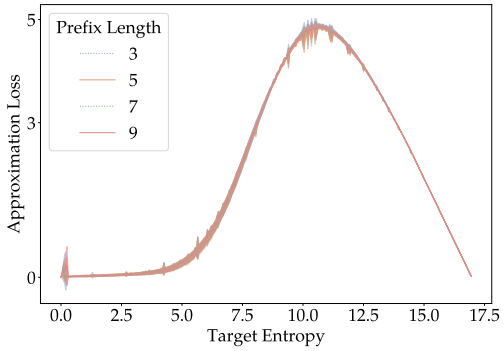


Figure 11: Approximation loss on vanilla distributions with varying prefix lengths. While different prefix lengths result in slight variations in approximation loss, the overall trend remains consistent.

B Experiment Details

The objective of our research is to determine whether, given a target distribution, it is possible to identify a prompt embedding that adequately approximates it. Since exhaustively enumerating all possible prompt embeddings is computationally infeasible, we instead employ modern optimization techniques, such as gradient descent, to search for an effective solution.

In our experiments, the prefix length is fixed at 5, and the learning rate is consistently set to 0.1 across all settings. The tuning process is conducted

for a maximum of 500 epochs; however, early termination is applied if the approximation loss with respect to the target distribution does not exhibit improvement over 20 consecutive epochs. The minimum approximation loss observed throughout the training trajectory is recorded. The subsequent sections provide a detailed overview of the experimental setup.

B.1 Prefix Initialization

Li and Liang (2021) observed that prefix initialization has a significant impact in low-data regimes. Given that our research aims to investigate the ease of eliciting specific target distributions using modern optimization techniques, it is crucial to minimize the influence of randomness introduced by different initialization configurations. To this end, we performed experiments using various initialization strategies by varying random seeds, thereby reducing the variability attributable to prompt embedding initialization.

We observed that the approximation loss varies across different initializations, as illustrated in Fig. 4. In the main body of this study, we report the minimum approximation loss obtained from five different initializations. While the absolute approximation values differ depending on the initialization, we find that the overall trends observed across the studied distributions remain consistent.

B.2 Prefix Length

A longer prefix introduces more learnable parameters, which may potentially improve performance in approximating target distributions. To investigate this, we conducted an ablation study on prefix length using the vanilla distributions, as shown in § 5.1. We found that increasing the number of tokens in the prefix—thus making it more expressive—does not alter the general trend. The results of approximating distributions with varying entropy under different prefix length configurations are presented in Fig. 11.

C Future Directions

We provide a case study investigating how well LMs approximate uniform distributions over tokens sharing the same part of speech. This illustrative example demonstrates the applicability of our proposed framework (Fig. 2) for analyzing the expressiveness of LMs. We envision that future work can further build on this framework to explore model behaviors from new interpretability perspectives.

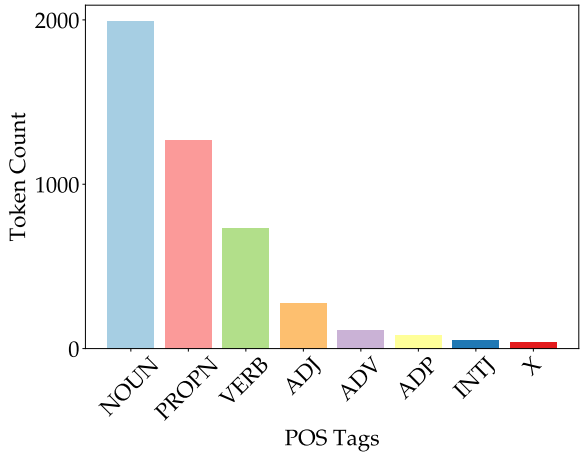


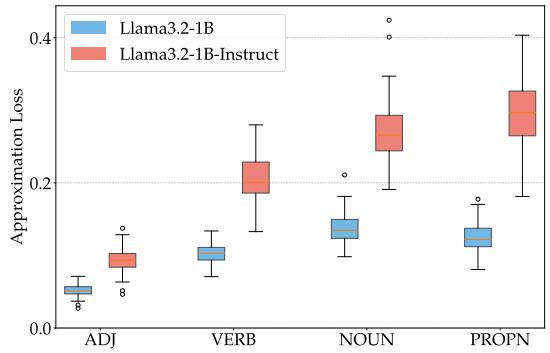
Figure 12: POS distribution of English tokens in the vocabularies of investigated LMs.

We design a simple experiment to investigate the difficulty of approximating uniform distributions over tokens belonging to different parts of speech (POS). Specifically, we select three LMs from two distinct families and their instruction-tuned variants: Llama3.2-1B (Dubey et al., 2024), Qwen2.5-0.5B, and Qwen2.5-1.5B (Yang et al., 2024). To identify the POS tags of tokens from the model vocabularies, we utilize ScispaCy (Neumann et al., 2019) for tokenization and tagging. For simplicity, we restrict our analysis to tokens that are assigned a single, unambiguous POS tag. Fig. 12 shows the distribution of the eight most common POS tags across English tokens in the vocabularies of the two model families.

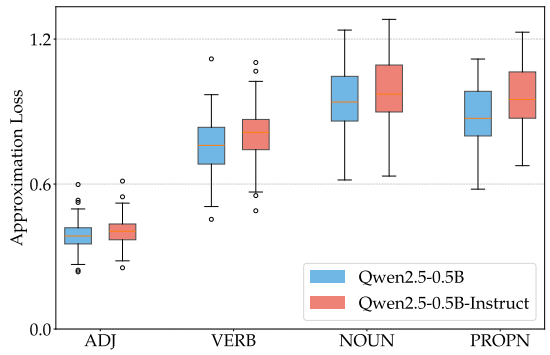
We construct POS-wise uniform distributions by randomly sampling 100 tokens for nouns, proper nouns, verbs, and adjectives for 100 times, and create a uniform distribution each time. Following the same setup in App. B, we report the minimum approximation loss across 5 different prompt initializations.

The experimental results are presented in Fig. 13. We observe that across all three comparisons, the instruction-tuned variants consistently exhibit higher approximation loss than their base counterparts. This suggests the presence of systematic distributional shifts introduced by instruction tuning. Additionally, we find that all evaluated LMs achieve lower approximation loss on adjectives compared to other POS.

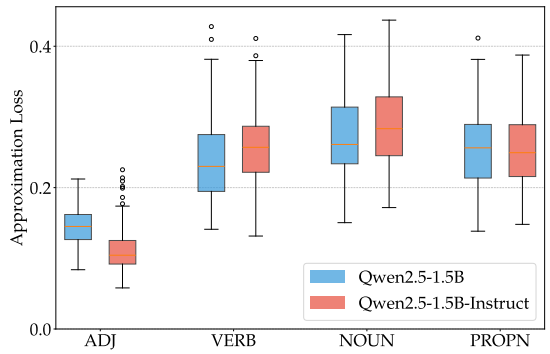
These consistent patterns across models from different family suggest that applying our proposed



(a) Llama3.2-1B



(b) Qwen2.5-0.5B



(c) Qwen2.5-1.5B

Figure 13: The boxplot of approximation loss is plotted for POS-wise uniform distributions on LMs from two distinct families. We report the four most common POS tags across English tokens in the vocabularies of the two model families, with comparison between base and instruction-tuned variants.

framework from a distributional perspective offers a promising direction for systematically understanding LMs. We hope this work inspires further research into the underlying distributional behaviors of LMs and informs future design of more robust and interpretable systems.