PROMPT WAYWARDNESS: The Curious Case of Discretized Interpretation of Continuous Prompts

Anonymous ACL submission

Abstract

Fine-tuning continuous prompts for target tasks has recently emerged as a compact alternative to full model fine-tuning. Motivated by these promising results, we investigate the feasibility of extracting a discrete (textual) interpretation of continuous prompts that is faithful to the problem they solve. In practice, we observe a "*wayward*" behavior between the task solved by continuous prompts and the nearest neighbor discrete projections of these prompts: One can find continuous prompts that solve a task while being projected to an *arbitrary* text (e.g., definition of a different or even a contradictory task) and simultaneously being within a very small (2%) margin of the best continuous prompt of the same size for the task. We provide intuitions behind this odd and surprising behavior, as well as extensive empirical analyses quantifying the effect of design choices. For instance, larger models exhibit higher waywardness, i.e, we can find prompts that more closely map to any arbitrary text with a smaller drop in accuracy. These findings have important implications relating to the difficulty of faithfully interpreting continuous prompts and their generalization across models and tasks, providing guidance for future progress in prompting language models.

1 Introduction

Recent work has shown the surprising power of *continuous prompts* to language models (LMs) for controlled generation and for solving a wide range of tasks (Li and Liang, 2021; Lester et al., 2021; Min et al., 2021). Despite these successes, the resulting continuous prompts are not easy to interpret (Shin et al., 2020). Is it possible to come up with meaningful discrete (textual) interpretations of continuous prompts, especially ones that provide a faithful explanation of the prompt's behavior?

Towards addressing this question, we propose and investigate the *Prompt Waywardness* hypothesis (§3.2), a surprising disconnect between the intended behavior of continuous prompts and their nearest-neighbor discrete (language) representations.[1] In particular, we show that one can find continuous prompts that perform a desired task while, at the same time, project to *any* given target text. This indicates that there is little correspondence between continuous prompts and their discrete interpretation. For instance, a continuous prompt that effectively solves the sentiment classification task in Fig.1, when projected onto discrete space, might appear as the definition of a different task (*"flip the sentiment"*).

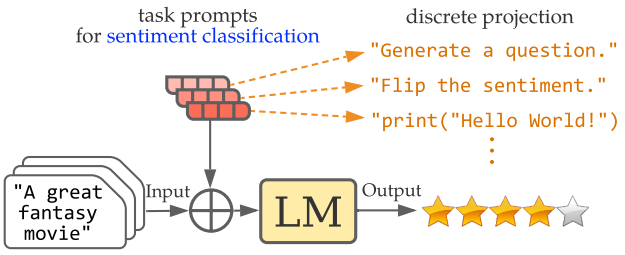

Figure 1: We show that one can find accurate continuous prompts (that do well on a given task, e.g., sentiment classification here) such that they can be projected to any *arbitrary* text, such as the definition of a different task (e.g., generating a question) or even an irrelevant statement (e.g., a piece of code) — suggesting a disconnect between the outcome of continuous prompts and their discrete interpretations.

We conduct extensive analysis showing Waywardness on five classification datasets (§4). Empirically, we find the existence of *wayward* prompts — prompts that solve each of these tasks while having *arbitrary* natural language projections. To study a variety of projected text, we experiment with 60+ sentences, either a discrete prompt from another task (from Mishra et al. 2021b) or a random sen-

[1] Nearest-neighbor projection via dot product has been previously used to study properties of continuous word embeddings (Mikolov et al., 2013; Hashimoto et al., 2016) and is commonly performed in the final layer of modern generative LMs (Radford et al., 2019; Raffel et al., 2020).

tence from a large text corpus. We observe that it is possible to find prompts that project to a given discrete prompt (token overlap 94% F1) while scoring within 2% accuracy of the best continuous prompts-based solution for the task. Further analysis shows that the effect of Waywardness gets worse for larger models and longer prompts. We explain this surprising behavior by *relating it to* several structural properties of large language models (§5).

We discuss several social and research implications of prompt waywardness, to help guide future research on prompt based models (§6). First and foremost, despite many promising attributes of continuous prompts, interpreting them is non-trivial and will require further research. In fact, careless interpretation of continuous prompts can result in vulnerabilities against malicious attacks concealed under the guise of benign discrete representation. Further, the loose correspondence between continuous and discrete prompts poses a challenge for future research in *differentiable interpretable-prompt optimization* – optimization in search of human readable discrete prompts through the continuous space. Our work shows that continuous and discrete prompts, despite their seeming similarity, are quite different and the results from one may not always transfer to the other. We hope these findings will motivate further innovations in the prompting literature for NLP models.

2 Related Work

Continuous prompts. There is a line of work focused on tuning continuous prompts (Li and Liang, 2021; Lester et al., 2021; Zhong et al., 2021; Qin and Eisner, 2021; Zhou et al., 2021). A recurring theme in this line of work is the strength of continuous prompt in results in strong, yet compact models—compared to conventional architecture fine-tuning approaches. Motivated by the success of continuous prompt tuning, this paper investigates the feasibility of interpreting a learned continuous prompt and its connection to discrete prompts.

Discrete prompts. The release of GPT-3 (Brown et al., 2020) sparked a lot of excitement about the emergent ability of LMs in following discrete natural language prompts. Consequently, countless follow-up studies have used manually-designed discrete prompts for probing LMs (Petroni et al., 2019; Jiang et al., 2020), improving LMs few-shot ability (Schick and Schütze, 2021; Gao et al., 2021; Le Scao and Rush, 2021), and their zero-shot ability as well as transferability (Mishra et al., 2021a; Reynolds and McDonell, 2021). While discrete prompts have clear advantages, in addition to being human-readable and thus easily interpretable, we do not have efficient and algorithmic ways of reconstructing them. For example, Shin et al. (2020)'s algorithm discovers discrete prompts, yet the results are not human readable. Prior work also finds that model performance is highly sensitive to small changes in wordings (Mishra et al., 2021a) and that optimization over the discrete prompt space is non-trivial and often highly unstable. Our findings here about the disconnect between continuous prompts and their discrete interpretation provides another perspective on the difficulty of discovering discrete prompts via continuous optimization algorithms that (directly or indirectly) leverage the continuous space (more discussion in §6).

3 Prompt Waywardness

3.1 Preliminaries: Setup and Terminology

We begin with some notation and the setup of our study, starting with the space of discrete and continuous prompts (Fig.2). Let $p_d \in \{0,1\}^{L \times V}$ denote a discrete prompt represented as an L-length sequence of one-hot vectors over a lexicon of size V (corners of a hyper-cube). Similarly, let $p_c \in \mathbb{R}^{L \times d}$ denote a continuous prompt, represented as a L-length sequence of d-dimensional real vectors.

Projection operators. We define operators that project these two spaces to one another. Define the c-projection as one that maps discrete inputs to a continuous space by multiplying with a fixed (often pre-trained) embedding matrix[2] $E \in \mathbb{R}^{V \times d}$:

$$c\text{-proj}(p_d) = p_d E \in \mathbb{R}^{L \times d}. \quad (1)$$

The d-projection maps the continuous inputs to nearest neighbor discrete elements, where for each position l ($1 \leq l \leq L$), one of the possible (and perhaps most straightforward) methods for interpreting a continuous prompt is defined as a projection onto nearest neighbor representations (Mikolov et al., 2013; Hashimoto et al., 2016):

$$d\text{-proj}(p_c) = [\delta_1 \cdots \delta_l \cdots \delta_L] \in \{0,1\}^{L \times V}, \quad (2)$$

[2]In our experiments we use the embedding matrix of the GPT2 family (Radford et al., 2019) which is used for both mapping input text to their embeddings as well as generating text in the output layer.

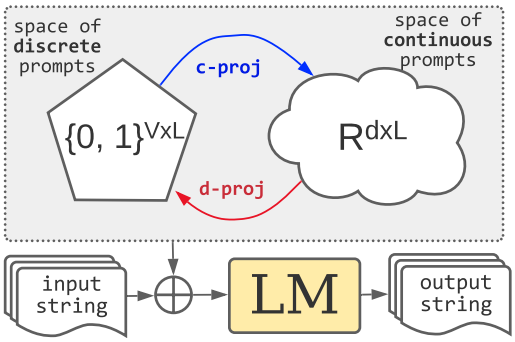

Figure 2: The general problem setup: Similar to Lester et al. (2021)'s setup, each prompt (usually a continuous one) is appended to the given input and fed to a frozen language model.

where δ_l is a one-hot vector corresponding to the word with the closest (highest dot product) embedding to the l-th position of continuous prompt p_c.

These projections are used in the first and last layer of virtually all modern LMs, such as GPT2.

Solving tasks with continuous prompts. Consider any machine learning model M (typically a pre-trained model) that takes textual input x and produces output y. Normally, the parameters of M are learned so as to optimize behavior on a task with a dataset $\mathcal{D} = \{(x, y)\}$ of input/output pairs. In prompt tuning (Lester et al., 2021), one freezes the parameters of M and instead optimizes for a prompt p that, when fed in conjunction with x, makes M produce the desired output y. Thus, p represents the only learnable parameters in this method. When p is a discrete prompt with k tokens, it can be simply concatenated with x, denoted $p + x$. In our study, p will be a *continuous* prompt (of length equal to the embedding of k tokens). We will concatenate it with the embedding of the input x. For simplicity and with some abuse of notation, we use $p + x$ to denote concatenation in this continuous case as well.

One can quantify the amount of loss incurred when using a continuous prompt p as follows:

$$\ell(p; \mathcal{D}) = \mathbb{E}_{x,y\sim\mathcal{D}} \left[\mathsf{loss}(M(p + x), y)\right], \quad (3)$$

Minimizing this loss function (empirical risk minimization) over p recovers a minimum risk continuous prompt for this dataset:

$$p_c^* = \arg\min_{p_c \in \mathbb{R}^{L\times d}} \ell(p_c; \mathcal{D}^{\text{train}}). \quad (4)$$

Given this prompt, its generalization to the test data can be measured in terms of the loss incurred on the test set: $\ell(p_c^*; \mathcal{D}^{\text{test}})$.

3.2 The Waywardness Hypothesis

How should one interpret the resultant continuous prompt $\tilde{p}_c$? Empirically, one can easily verify that such continuous prompts are not unique (e.g., random initializations lead to different outcomes). Additionally, the resultant prompts get projected to seemingly irrelevant discrete elements. Taking this to an extreme, we hypothesize that next to the continuous projection c-proj(p_d) of any discrete prompt p_d, there exists a variety of continuous prompts p_c that trigger responses from model M that are orthogonal to the intentions described by the discrete prompt p_d. We formalize this idea as the following hypothesis, where $L \in \mathbb{N}$ is the length of the discrete target prompt, M is a prompt-based model, and $\mathcal{D}$ is a dataset for a desired task:

Hypothesis 1 (Prompt Waywardness) *For all $L, M, \mathcal{D}$, there is a small Δ such that for any discrete target prompt p_d with length L, there exists a continuous prompt $\tilde{p}_c \in \mathbb{R}^{L\times d}$ such that:*

1. $\left|\ell(\tilde{p}_c; \mathcal{D}^{test}) - \ell(p_c^*; \mathcal{D}^{test})\right| < \Delta$, *yet*
2. *d-proj*$(\tilde{p}_c) = p_d$.

In other words, $\tilde{p}_c$ is nearly as effective at making M solve the task as the optimal continuous prompt (Eq.4), and yet it projects to p_d. In this statement, Δ (prompt performance gap relative to the optimal prompt p_c^*) is a function of the prompt length L, the model M (e.g., its embedding size and depth when M is transformer based), and inherent properties of the target dataset $\mathcal{D}$. The analysis in §4.3 will provide an empirical estimate of this gap $\hat{\Delta}$ as a function of various parameters like model size and prompt length.

It is worth emphasizing that the hypothesis is stated for *any* task and any set of discrete prompts, **even if they are irrelevant or contradictory**.[3]

3.3 Finding Wayward Prompts

While the above hypothesis promises the *existence* of $\tilde{p}_c$, it does not say how to discover them. We now discuss a practical method for their discovery.

We learn a continuous prompt p_c using a modification of the prompt tuning objective of Lester et al. (2021). Our modification jointly minimizes the standard downstream task cross-entropy loss

[3]While our focus is on the use of continuous prompts for solving datasets (one prompt shared among many instances), one can imagine applications of the same conjecture to special use cases such as controlled generation (Dathathri et al., 2019) with one prompt per instance.

$\ell(.)$ for the task (Eq.3) and a distance measure $\mathsf{dist}(.)$ between p_c and the discrete target prompt $p_d \in \{0,1\}^{L\times V}$:

$$\ell'(p_c;\mathcal{D},\gamma) = \ell(p_c;\mathcal{D}) + \gamma\,\mathsf{dist}(p_c,p_d) \quad (5)$$

$$\tilde{p}_c = \arg\min_{p_c\in\mathbb{R}^{L\times d}} \ell'(p_c;\mathcal{D},\gamma), \quad (6)$$

where p_c is the only learnable parameter, and γ is a hyperparamter.

When $\gamma = 0$, the modified objective is reduced to the standard objective (Eq.4), $\ell'(.) = \ell(.)$. We refer to this case and its resulting prompt p_c^* as the *'unconstrained'* setting. A large value of γ will make p_c even closer (possibly identical) to c-proj(p_d) but lead to poor accuracy on a target dataset. Most of the experiments below are conducted via a range of γ values to better understand the trade off between the two terms in the objective function. In practice, we find $\gamma = 0.01$ to give a reasonable trade-off regardless of the target dataset and the choice of p_d.

There are at least two natural ways to define the distance measure $\mathsf{dist}(p_c,p_d)$ between a continuous prompt p_c and a discrete target prompt p_d, by converting one so that both are in the same space:

$$c\text{-}\mathsf{dist}(p_c,p_d) = \frac{\|p_c - c\text{-proj}(p_d)\|_2^2}{L} \quad (7)$$

$$d\text{-}\mathsf{dist}(p_c,p_d) = \mathrm{F1}\big(d\text{-proj}(p_c), p_d\big) \quad (8)$$

The first of these places both p_c and p_d in the continuous space and computes the squared-L^2 norm, normalized by the prompt length. This is used in our training loss (Eq.5) implementation. The second places both in discrete space (text) and computes the standard word-level token overlap F1 score.[4] This is used during our evaluation.

4 Empirical Support of Waywardness

We empirically investigate the Prompt Waywardness hypothesis (§3.2) using our modification (§3.3) of the prompt tuning method from Lester et al. (2021). We show that given an **arbitrary and irrelevant** discrete prompt p_d, it is possible to learn a continuous prompt that is mapped to p_d while retaining its accuracy on a given dataset.

4.1 Setup

Target tasks. Following the setup of Min et al. (2021), we select a diverse set of 5 classification datasets: SST-2 (Socher et al., 2013), SST-5 (Socher et al., 2013), AGNews (Zhang et al., 2015), Subj (Pang and Lee, 2004) and TREC (Voorhees and Tice, 2000). Statistics and the unconstrained accuracy of each dataset are provided in Table 1.

Dataset	Task	$\lvert C\rvert$	Acc
SST-2	Sentiment analysis (movie)	2	92.4
SST-5	Sentiment analysis (movie)	5	50.3
AGNews	News classification (topic)	4	88.1
Subj	Subjectivity classification	2	90.5
TREC	Answer type classification	6	88.0

Table 1: The collection of downstream tasks used in the experiments (§4.1). $|C|$ indicates the output size (number of classes); *Acc* indicates the unconstrained accuracy of a prompt tuning method (Lester et al., 2021) using GPT2 Large, as a reference point.

Discrete Target Projections. We compile two sets of discrete target prompts: (1) 32 target prompts for solving tasks from Natural-Instructions[5] dataset (Mishra et al., 2021b) that are distinct from and intentionally orthogonal to the end tasks considered here. These were chosen by excluding discrete target prompts that have high lexical overlap with other discrete prompts; this is because we found lexically similar prompts are often semantically similar even when written for different subtasks. (2) 30 random sentences from PILE,[6] a large-scale, diverse text corpus used to pretrain GPT-J, the largest public causal language model (Wang and Komatsuzaki, 2021). The sampled sentences were drawn from a Poisson distribution with $\lambda = 14$, which makes the average length of the sentence to be consistent to those in Natural-Instructions. These sentences are selected to have little or no token overlap with the true definition of the target tasks. See Table 3 for a few examples.

Evaluation metrics. For all experiments, we report two metrics: (1) the task *accuracy*[7] as well as (2) *prompt F1*, the word-level token overlap F1 score computed as in Eq.8, since it easy to interpret and is commonly used for evaluating the textual output of models (Rajpurkar et al., 2016).

Models. For evaluation, we use GPT2 (Radford et al., 2019) an auto-regressive LM which has extensively been used in many NLP applications. Un-

[4] Ignoring punctuation marks and articles, and applying lemmatization.

[5] https://instructions.apps.allenai.org

[6] https://pile.eleuther.ai

[7] We did not consider alternatives like Macro-F1 because all datasets are roughly balanced across different classes.

Data	p_d Source	Task Accuracy (%) $\hat{\Delta}$ (Acc(p_c^*) → Acc($\tilde{p}_c$))		Prompt F1 (%)
SST-2	NI	0.7	(92.4 → 91.8)	99.0
	PILE	0.5	(92.5 → 92.0)	97.1
	Avg	0.6	(92.4 → 91.9)	98.1
SST-5	NI	3.3	(50.2 → 48.5)	95.9
	PILE	0.7	(50.5 → 50.2)	92.4
	Avg	2.0	(50.3 → 49.3)	94.2
AGNews	NI	1.6	(88.0 → 86.6)	97.4
	PILE	-0.1	(88.1 → 88.2)	97.5
	Avg	0.8	(88.1 → 87.3)	97.4
Subj	NI	2.0	(91.3 → 89.5)	97.3
	PILE	0.9	(89.6 → 88.8)	94.4
	Avg	1.5	(90.5 → 89.2)	95.9
TREC	NI	3.3	(87.5 → 84.7)	86.5
	PILE	1.2	(88.5 → 87.5)	85.6
	Avg	2.3	(88.0 → 86.0)	86.1

Table 2: Main Results: Accuracy of solving five classification datasets, in an unconstrained setting (p_c^*) vs. constrained by the projection to various **irrelevant** pieces of text ($\tilde{p}_c$). Optimization is done using $\gamma = 0.01$ in the objective function (Eq.5). $\hat{\Delta}$ indicates the relative accuracy drop (in %) from unconstrained accuracy. Each reported score (*Accuracy* and *Prompt F1*) are the average over 62 discrete target prompts and 3 random seeds. **Overall, it is possible to achieve $\geq 94\%$ prompt F1 with under 2% drop in accuracy.**

less otherwise specified, we use a 'large' variant consisting of 774M parameters.

Implementation details. We use a batch size of 8, learning rate 0.01, and 2000 training steps. When experimenting with a discrete target prompt p_d, we initialize the search for continuous prompts (both $\tilde{p}_c$ and p_c^*) using c-proj(p_d).[8] For all experiments, report accuracy averaged over three random seeds.

4.2 Main Results

For each of the 5 tasks T and for each of the 62 discrete target prompts p_d, we use the objective in Eq.5 to find a prompt $\tilde{p}_c$ such that it solves T with a high accuracy while, at the same time, having a discrete projection that is close to p_d. For comparison, we also train unconstrained prompts p_c^* ($\gamma = 0.0$) which solve task T without any projection constraint. To ensure a fair comparison between $\tilde{p}_c$ and p_c^*, we ensure that they have the same size L. In other words, for each $\tilde{p}_c$ (that has the same length as p_d), we train another p_c^* with the same length. We denote the relative accuracy drop from p_c^* to $\tilde{p}_c$ as $\hat{\Delta}$.

Table 2 summarizes the results. Across all datasets, we find that it is possible to learn a continuous prompt p_c whose discrete projection is very close to p_d and mostly retains the task accuracy. There is a trade-off between the task accuracy and prompt F1, which can be controlled by the choice of γ (more extensive ablations in the forthcoming paragraphs (§4.3)). Overall, with $\gamma = 0.01$, it is possible to achieve $\geq 94\%$ prompt F1 with under 2% relative drop in task accuracy. The only outlier is the TREC dataset where we achieved a prompt F1 score of 86% for a $\hat{\Delta} = 2.3\%$ relative drop in accuracy. This might be due to the difficulty of learning effective prompts on TREC (also discussed by Min et al. (2021)).

Example prompts with varying values of prompt F1 scores are shown in Table 3. A prompt F1 $\geq 94\%$ generally indicates one word mismatch with almost no semantically meaningful difference.

4.3 Further Analysis

Effect of Gamma. Fig. 3 shows the trade-off between task accuracy and the prompt F1 when varying γ from 0 to 0.03. As γ increases, the task accuracy goes down while the prompt F1 increases. The drop in task accuracy is relatively minor—it is possible to learn a continuous prompt for which prompt F1 is near 1.00 and the accuracy drop relative to the unconstrained accuracy is less than 1%.

Effect of Prompt Length (L). We randomly sample sentences from The PILE with a constraint that its length must be L (chosen from $\{4, 7, 14, 28, 56\}$). The left and the middle parts of Fig. 4 illustrate the results. We find that when L is very small (e.g., 4) it is relatively difficult to learn a continuous prompt p_c that is close to p_d (F1<60%) while retaining the task accuracy. This is likely because the prompt being too short significantly hurts the expressivity of the prompt. Nonetheless, when L is reasonably larger, e.g., 14 (the average length of in Natural Instructions) or longer, all cases lead to a continuous prompt with near 1.0 prompt F1 and little accuracy drop.

Effect of Model Size. We vary the size of the GPT2 models—small, medium, large, and XL—with 124M, 355M, 774M, and 1.5B parameters, respectively. Figure 5 (right) reports the result on SST-2. We find that (1) across different sizes of the LM, our findings in learning continuous prompts

[8] While this is different from prior work (Lester et al., 2021; Min et al., 2021) that uses a random subset of the top-5000 vocabs, we find no meaningful differences in an unconstrained accuracy between two initialization methods.

d-proj(p_c)	*Prompt F1*	*Acc*($\tilde{p}_c$)
Task: AGNews p_d: Write down the conclusion you can reach by combining the given Fact 1 and Fact 2.		
Write down the conclusion you can reach by combining the given Fact 1 and Fact 2.	100.0	89.2
Write down the conclusion you can reach by combining the given Fact 1. Fact 2.	96.3	88.1
Write down the conclusion you can reach by combining the given Fact 1 Category Fact 2.	92.9	89.0
Write Messi in conclusion you can reach by combining the given Fact 1 and Fact 2.	89.7	88.8
Task: SST-5 p_d: "If they have other interests and aims in life it does not necessarily follow that they are passive sheep."		
"If they have other interests and aims in life it does not necessarily follow that they are passive sheep."	100.0	51.2
"If they have other interests and aims in life it does not necessarily follow that they are terrible sheep."	94.7	53.6
"If they have other interests and aims in life it does not necessarily follow that they are terrible GoPro."	89.5	52.3
Task: SST-5 p_d: int clamp(int val, int min_val, int max_val) { return std::max(min_val, std::min(max_val, val)); }		
int clamp(int val, int min_val, int max_val) { return std::max(min_val, std::min(max_val, val)); }	100.0	50.5
int clamp(int val, int min_val, int max_val) { return std::max(min_val, std::min(max_val terrible val)); }	95.7	52.0
int clamp(int val, int min_val, int max_val) { return std::max(min_val, std::min(max_val terrible val)); This}	91.7	53.3

Table 3: Examples of the target prompts p_d and their reconstructions via d-proj(p_c) for different ranges of prompt F1 scores. The first p_d is from Natural-Instructions; the rest two are sampled from The PILE. The mismatches with the original prompt are color-coded.

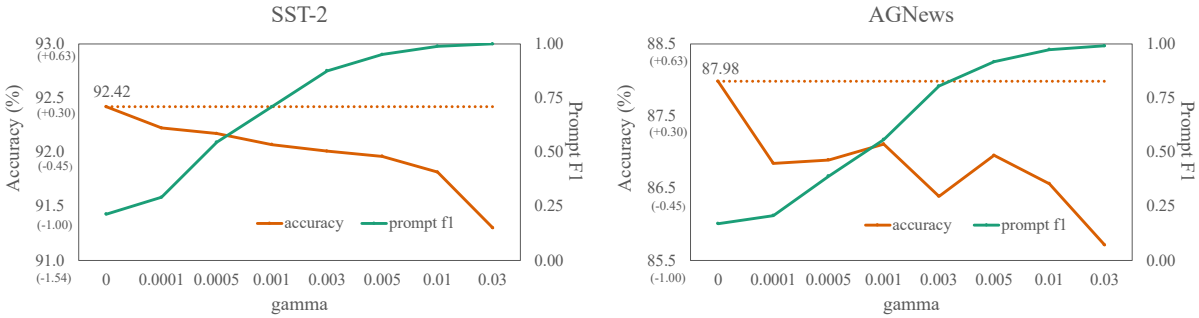

Figure 3: The effect of γ on SST-2 and AGNews. *Accuracy* is the average over 32 discrete target prompts from Natural Instructions and 3 random seeds. A dotted line indicates unconstrained accuracy p_c^* (same as when $\gamma = 0$). Numbers inside parentheses in the y-axis indicate relative drop in accuracy against unconstrained accuracy. **There is a clear trade-off between the task accuracy and the prompt F1.**

with the prompt F1 of near 1.0 and little drop in the accuracy generally hold, and (2) in particular, the drop in accuracy is more negligible with larger LMs (0.2% with XL, 0.5–0.7% with medium and large, 1.2% with small).

5 Explaining Waywardness

Here we provide intuitions behind the factors that enable Prompt Waywardness.

The mapping between continuous and discrete spaces is not one-to-one. While a discrete target prompt is mapped to exactly one continuous prompt (via its embedding, Eq.1; cf. Fig.2), the reverse is not true. In fact, except for a very small fraction of unnatural or degenerate edge cases,[9] for every target discrete prompt, there are infinitely many continuous prompts that project back to it (via Eq.2). While simple counting-based arguments are insufficient in continuous spaces, we formally prove (Appendix C) that this property holds for all nearest-neighbor projections under any metric distance, and broadly for all but a negligible (measure zero) portion of possible projection operators.

This intuitively suggests that there is a whole region of continuous prompts that corresponds to a fixed discrete representation (Fig.6). The remaining question is, how is this region able to have a diverse set of prompts that can solve a variety of tasks? This is addressed next.

Deep models give immense expressive power to earlier layers. The deeper a network is, the more expressivity it has with respect to its inputs (Telgarsky, 2016; Raghu et al., 2017). Since continuous prompts reside just before the first layer, they enjoy a lot of expressivity. Therefore, no matter how narrow the regions corresponding to individual tokens are (Fig.6), they are extremely powerful in solving

[9] Such as using a non-metric distance in nearest-neighbor mapping, or mapping all of $\mathbb{R}^d$ to a single discrete prompt.

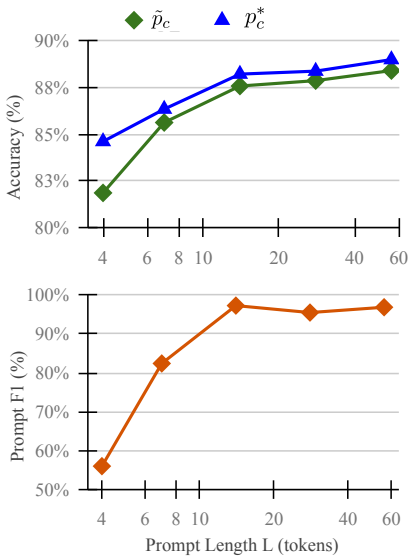

Figure 4: The effect of the length of the prompt (L) on AGNews. Each point computed via the average over 32 discrete target prompts from Natural Instructions and 3 random seeds ($\gamma = 0.01$ used). The corresponding prompt F1 is reported as a orange line. The accuracy of p_c^* and $\tilde{p}_c$ increase as a function of prompt length however, the gap between them tends to decrease. **The relative accuracy drop is marginal when L is not too small (e.g., 7 or larger).**

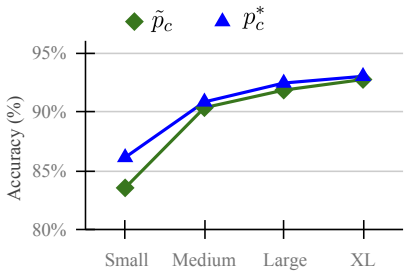

Figure 5: The effect of the size of the model—small, medium, large, and XL—on SST-2. Each point in the experiment is computed by averaging over 30 experiments each with a different discrete target prompt from PILE and 3 random seeds. We vary $\gamma = \{0.01, 0.005, 0.003\}$ and choose the one for which prompt F1 is larger than 0.98. **The relative accuracy drop (gap between the two trends) decreases as models become larger.**

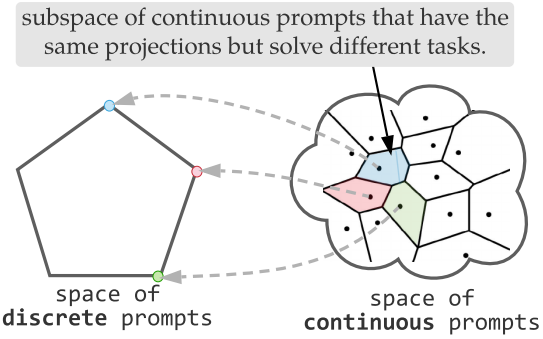

Figure 6: The projection discrete space (Eq.2) induces a clustering (a Voronoi diagram) of the continuous space. Each cluster has infinitely many points that get mapped to the same discrete token.

a variety of tasks. Previously in §4.2 we provide an empirical analysis showing evidence that the effect of Waywardness is stronger in deeper models.

6 Implications of Prompt Waywardness

We discuss the implications of these findings on several inter-related lines of research. Note that all the following statements are valid within the boundaries of the existing architectures. Moving beyond these barriers likely requires major innovations in terms of LM architectures or how continuous prompts are optimized.

Faithful interpretation of continuous prompts is difficult. Given the intuitions behind and empirical support for the Waywardness hypothesis (§5), faithful discrete interpretations of continuous prompts via common discrete projections (like nearest-neighbor projection) are unlikely to be robust based on current approaches. It is an open question whether there is a better way of interpreting continuous prompts with human language, or whether explaining and interpreting continuous prompts via human language is inherently impossible because they lie in completely different spaces. Future work may investigate more on this topic in order to improve the interpretability of prompt-based language models.

Risk of interpreting continuous prompts: concealed adversarial attacks. It is not difficult to imagine a future where proprietary model development is driven by fine-tuned continuous prompts. In such a world, not addressing the challenges involved in discrete interpretation of continuous prompts can lead to harmful (and potentially, adversarial) consequences (Slack et al., 2020; Wallace et al., 2021), as discussed below.

We consider the following scenario: a model designer comes up with a set of continuous prompts that solve a target task (e.g., ranking resumes according to each applicant's qualifications and merits). Whether intentionally or not, such prompts may maliciously target, for example, a minority group. To assure their customers, the model designer uses the projection of the prompt that ex-

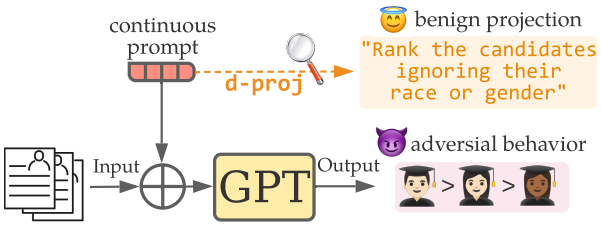

Figure 7: Waywardness implies that continuous prompts can be mapped to seemingly innocuous descriptions while acting maliciously.

presses a benign definition for the task, which does not reveal the true nature of the egregious behavior. The customers might even evaluate the prompt on a few instances but not notice this harmful behavior, e.g., when it effects a minority group not in the evaluation set. In a way, the benign discrete projections may provide a false sense of security.

Optimizing discrete prompts through continuous prompts can be degenerate. Manually-written discrete prompts have many nice properties (Schick and Schütze, 2021; Mishra et al., 2021a), yet we do not have an efficient algorithmic way of finding them. One way to operationalize this is to formulate differentiable objective functions via LMs like GPT (Radford et al., 2019). Consider the following problem which is defined in the space of continuous embeddings $p_c \in \mathbb{R}^d$:

$$\max_{p_c \in \mathbb{R}^d} \overbrace{\mathbb{P}\left(\mathcal{D}|p_c\right)}^{\text{utility}} \times \overbrace{\mathbb{P}\left(d\text{-proj}(p_c)\right)}^{\text{readability}}, \qquad (9)$$

This a joint optimization towards a *utility* objective (the extent to which it can solve dataset $\mathcal{D}$) and a *human readability* objective. According to the Waywardness hypothesis, there are p_c's that assign high mass to the utility term while also mapping to human interpretable text that is *irrelevant* (or even contradictory) to the task solved by the prompt – hence, *degenerate* solutions.

The same challenge holds if this optimization objective, instead of continuous prompts, is reformulated in terms of word probabilities (e.g., similar to Kumar et al. (2021, Sec 2.2)). This is the case, since searching in the space of word probabilities is analogous to a search in embedding spaces.[10]

[10] A distribution over words $p \in [0,1]^V$ corresponds to a continuous prompt $p_c = c\text{-proj}(p)$ which is a weighted combination of V-many basis vectors (word embeddings) that form a linear span of $\mathbb{R}^d$.

In summary, Waywardness presents a challenge for searching effective discrete prompts via continuous optimization. The recent works have used additional signals such as domain-specific constraints (Qin et al., 2020; Khot et al., 2021) to alleviate these challenges. We hope to see more design innovations for further progress in this direction.

Continuous prompt tuning does not necessitate task-specific initialization. Recent works on continuous prompt-tuning have shown the effectiveness of initialization from embeddings of *random* common words (Lester et al., 2021; Min et al., 2021), despite these words being irrelevant to the task solved by these prompts. This, however, makes sense given the observations made in this work regarding the existence of effective prompts around word embedding subspaces.

7 Conclusion

The prompting literature has seen many parallel developments around continuous and discrete prompts, as efficient alternatives to fine-tuning models with tens of millions of parameters. Our work introduced the Prompt Waywardness hypothesis, which expresses a surprising disconnect between continuous and discrete prompts: given a downstream task, for *any* discrete target prompt p_d, there exists a continuous prompt that projects to p_d while achieving strong performance on the task. We provided empirical evidence for this hypothesis, studied various parameters around it, and ended with several implications of this hypothesis.

While our experiments are done on the GPT family, we expect our findings to apply to a broader set of architectures that, in one way or another, use similar mechanisms for mapping discrete elements to continuous representations and vice versa. Similarly, while our projection to the discrete space (Eq.2) is a popular operator in the field (cf. Footnote 1), the intuition explained in Propositions 1 and 2 of the Appendix suggests similar behavior for a broad class of projection operators.

Prompt Waywardness identifies challenges for future progress on algorithmic methods for the discovery of human readable prompts that are faithful to the task they solve. We hope the observations made in this work motivate architectural innovations that overcome such challenges and guide future steps in the prompting literature.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *Proceedings of* ICLR.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of* ACL.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*.

Tatsunori B Hashimoto, David Alvarez-Melis, and Tommi S Jaakkola. 2016. Word embeddings as metric recovery in semantic spaces. TACL, 4:273–286.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? TACL, 8:423–438.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text Modular Networks: Learning to decompose tasks in the language of existing models. *Proceedings of* NAACL, page 1264–1279.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Proceedings of* NeurIPS, 34.

Teven Le Scao and Alexander M Rush. 2021. How many data points is a prompt worth? In *Proceedings of* NAACL, pages 2627–2636.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of* ACL.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of* NeurIPS, pages 3111–3119.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Reframing instructional prompts to GPTk's language. *arXiv preprint arXiv:2109.07830*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021b. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of* EMNLP, pages 2463–2473.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of* NAACL, pages 5203–5212.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Backpropagation-based decoding for unsupervised counterfactual and abductive reasoning. In *EMNLP*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR, 21:1–67.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. 2017. On the expressive power of deep neural networks. In *Proceedings of* ICML, pages 2847–2854.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of* EMNLP, pages 2383–2392.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Proceedings of* CHI, pages 1–7.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of* EACL, pages 255–269.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. In *Proceedings of* EMNLP, pages 4222–4235.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of* EMNLP.

Matus Telgarsky. 2016. Benefits of depth in neural networks. In *Proceedings of* COLT, pages 1517–1539.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of* SIGIR.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on NLP models. In *Proceedings of* NAACL, pages 139–150.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between CV and NLP! a gradient-based textual adversarial attack framework. *arXiv preprint arXiv:2110.15317*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of* NeurIPS.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of* NAACL, pages 5017–5033.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2021. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*.

Supplementary Material

A Additional Experimental Details

Here we include several experimental details (§4) that did not fit in the main text. For the experiments we used A100 GPUs with 40G memory. In terms of the time GPU time of the experiments, each round of training and inference for each seed took about around 6 min. Therefore, the total GPU hours for our main experiment (Table 2) adds up to 93 hours (6 mins $\times$ 3 seeds $\times$ 5 datasets $\times$ 62 prompts = 5580 mins).

B Experiment: Projection onto True Task Definitions

In all our results so far in §4, the target projected text was orthogonal to the tasks being solved. One might naturally wonder whether there is any benefit in projecting continuous prompts to the texts that truly describe the task being solved, i.e., a "true" prompt for the task. To this end, we manually authored 5 "true" prompts for each of the tasks. We then follow the exact same setup used earlier for Table 2 to fine-tune continuous prompts $\tilde{p}_c$ for the task while projecting onto these true task definitions. As before, we also fine-tune unconstrained prompts p_c^* of the same length, without any projection requirement.

By design, $\tilde{p}_c$ can be no more effective at solving the task than the unconstrained prompt p_c^* (barring suboptimal search issues), which is what we find in practice. For completeness, we report detailed results for "true" target prompts (analogous to Table 2) in Table 4.

More interestingly, as shown in Table 5, **continuous prompts that project to "true" target prompts are no more effective at solving the task** than continuous prompts that project to the 62 *irrelevant* target prompts considered earlier (Table 2). Specifically, the average performance gap Δ (relative to unconstrained prompts of the same length) is about the same (around 1.5) for continuous prompts that map to true task definitions compared to prompts that map to irrelevant text. This observation further bolsters the waywardness hypothesis—continuous prompts don't relate to the task being solved.

Data	Task Accuracy (%) $\hat{\Delta}_T$ (Acc(p_c^*) $\rightarrow$ Acc($\tilde{p}_c$))		Prompt F1 (%)
SST-2	1.0	(91.9$\rightarrow$ 90.9)	98.5
SST-5	0.9	(51.4 $\rightarrow$ 50.5)	96.1
AGNews	1.4	(91.8 $\rightarrow$ 90.4)	95.7
Subj	4.1	(89.8 $\rightarrow$ 85.6)	100.0
TREC	0.5	(88.6 $\rightarrow$ 88.1)	99.3

Table 4: Accuracy of solving five classification datasets, unconstrained setting (p_c^*) vs. constrained by the projection to **the true definition of tasks** ($\tilde{p}_c$) using $\gamma = 0.01$ in the objective function (Eq.5). Subscript T in Δ_T denotes this being the case for true task definitions. **Projecting to the true definition of a task does not help continuous prompts solve a task.**

	SST-2	SST-5	AGNews	Subj	TREC	Avg
$\hat{\Delta}_T$	1.0	0.9	1.4	4.1	0.5	1.6
$\hat{\Delta}$	0.6	2.0	0.8	1.5	2.3	1.4

Table 5: Task accuracy gap comparison between unconstrained prompts and those fine-tuned to project to a *true* task definition ($\hat{\Delta}_T$) as reported in Table 4. For comparison, we also show the corresponding performance gaps with *irrelevant* ($\hat{\Delta}$) from Table 2. **The average performance gaps are about the same (around 1.5) for true and irrelevant target prompts**—further evidence that continuous prompts don't relate to the task being solved.

C The mapping between continuous and discrete space is not one-to-one

As argued in §5, the mapping between the space of discrete input and that of word embeddings (Fig.2) is not a bijection. While a discrete target prompt is mapped to exactly one continuous prompt (via its embedding, Eq.1), the reverse is not true: except for some unnatural or rare cases (as formalized in the following propositions) there are infinitely many continuous prompts that project back to a fixed discrete target prompt (via Eq.2).

Nearest-neighbor projections are arguably natural, computationally efficient, and useful in practice. Although we have considered them in the Euclidean space so far, they can be defined for an *arbitrary* distance metric[11] m on $\mathbb{R}^d$. As before, consider an embedding of a lexicon of size V into $\mathbb{R}^d$ and the corresponding one-hot vectors in $\{0,1\}^V$. We call d-proj a *nearest-neighbor projection operator w.r.t.* m if it maps each $x \in \mathbb{R}^d$ to the one-hot vector in $\{0,1\}^V$ that corresponds to

[11] https://en.wikipedia.org/wiki/Metric_(mathematics)

the lexicon item whose embedding is closest to x under metric m (breaking ties arbitrarily).

Proposition 1 *Every nearest-neighbor projection operator, under any metric, maps infinitely many elements of $\mathbb{R}^d$, forming one or more continuous subspaces, to every one-hot vector in $\{0,1\}^V$.*

A proof is included in Appendix C.1. In effect, the projection operators induce a clustering of the space of continuous prompts $\mathbb{R}^{d\times L}$ into regions that have the same discrete projection (Fig.6).

The infinite-to-one mapping aspect is not limited to the class of nearest-neighbor projection operators. It is rather an inherent property of the interaction between continuous and discrete spaces, and holds for a broader family consisting of all but a negligible portion of possible projection operators:

Proposition 2 *Let $\mathbb{D}$ denote the space of all projection operators that map $\mathbb{R}^d$ to one-hot vectors in $\{0,1\}^V$. Let d-proj be a random projection drawn uniformly from $\mathbb{D}$. Then, with probability 1, d-proj maps infinite elements of $\mathbb{R}^d$ to every one-hot vector in $\{0,1\}^V$.*

C.1 Proofs

Proof of Prop. 1: Let $c_i \in \mathbb{R}^d$ for $i \in \{1,\ldots,V\}$ be fixed vectors (denoting the embedding of words in a lexicon of size V). Let $e_i \in \{0,1\}^V$ denote the one-hot vector with 1 in the i-th position and 0 elsewhere. Since d-proj is a *nearest-neighbor projection operator w.r.t.* m, by definition it maps $x \in \mathbb{R}^d$ to e_i whenever x is closest to c_i, i.e., $i = \arg\min_j m(x, c_j)$ (breaking ties arbitrarily).

Let $S_i \subseteq \mathbb{R}^d$ denote the pre-image of e_i, i.e., the elements that the nearest-neighbor projection d-proj maps to the i-th one-hot vector. By definition, $c_i \in S_i$. Let $d' = \min_j m(c_i, c_j) > 0$ denote the distance of c_i to the nearest c_j w.r.t. the metric m. Consider the subspace $C_i = \{x \mid m(x, c_i) < d'/2\}$. By design, we have $C_i \subseteq S_i$. Further, moving x by some small distance ϵ (w.r.t. m) to another point x' changes its distance to c_i only by at most ϵ (by the triangle inequality property of m). This implies that if ϵ is chosen to be small enough such that $m(x, c_i) + \epsilon < d'/2$, then x' must also be in C_i. In other words, if $x \in C_i$, then, for a small enough ϵ, the entire ϵ-neighborhood of x is also in C_i. It follows that C_i is an open subset of R^d and thus contains infinitely many elements forming a continuous subspace. Hence S_i, which contains C_i,e.g. also has infinite elements in one or more continuous subspaces. □

Proof of Prop. 2: For simplicity, assume $V = 2$. A projection operator d-proj $\in \mathbb{D}$ can then be fully characterized by the subset $S \subseteq \mathrm{R}^d$ that it maps to any one arbitrarily chosen one-hot vector. Choosing d-proj uniformly at random from $\mathbb{D}$ thus amounts to choosing the subset S uniformly at random from $\mathbb{R}^d$. We show that the probability of choosing an S such that $|S|$ is finite, is 0. (The same argument applies to $|\mathbb{R} \setminus S|$ being finite.)

To see this, let $\mathbb{S}_i$ denote the set of all (finite) subsets of $\mathbb{R}^d$ that have size exactly i. First, observe that the probability of choosing an S that lies in $\mathbb{S}_0 \cup \mathbb{S}_1$ (i.e., a subset of $\mathbb{R}^d$ that has at most 1 element) is 0; this is a degenerate case in the underlying continuous probability space. Second, for any $i \geq 2$, $\mathbb{S}_i$ has the same "size" (in the measure theoretic sense) as $\mathbb{S}_1$, because one can construct an injective map from either one to the other—which follows from the fact that they both have the same *cardinality* as the set $\mathbb{R}$.[12] Lastly, the space $\mathbb{S}$ of all finite subsets of $\mathbb{R}^d$ is the countable union $\cup_i \mathbb{S}_i$ of disjoint sets. Therefore, $\Pr[S \in \mathbb{S}] = \sum_i \Pr[S \in \mathbb{S}_i] = 0$. □

D Implications of Prompt Waywardness: continued

Here we mention other implications (§6) that did not fit in our page limit.

Gradients alone are insufficient to reverse engineer a model. Suppose we are given a fixed (fine-tuned or otherwise) model M (e.g., an open question-answering model) and an expected output y from this model (e.g., y ="*Joe Biden*"). Can we use *gradients* alone to generate a semantically meaningful input question that makes the model M generate this given answer? (without any additional assumptions on the input). More formally, if $q \in [0,1]^{L\times V}$ is a probability distribution over all questions of length L, are gradients with respect to the question input, $\frac{\partial M(c\text{-proj}(q))=y}{\partial q_{lv}}$, alone informative enough to move us towards the best human readable input that is faithful to the task being solved by M?

Our findings and the earlier argument about continuous differentiable optimization suggests this

[12] This can be proved using the rules of cardinal multiplication applied to $\mathbb{S}_i$ viewed as (a subset of) the Cartesian product of $\mathbb{S}_1$ with itself, i times.

may not be feasible with current methods. To see the correspondence to Prompt Waywardness, we can replace $\mathcal{D}$ in Eq.9 with the desired outcome y and run the optimization over word distributions (cf. Footnote 10). While gradients can help guide us towards *some* input that makes M produce y, such input is quite likely to not be faithful to the task being solved by M. In the context of the above example (M being a QA system), gradients might lead to inputs that are perhaps linguistically fluent but are neither proper queries nor semantically descriptive of *"Joe Biden"*.

Nevertheless, as noted earlier, gradients are still useful when they are applied using domain-specific constraints. For example, one can find local (word-level) perturbations that lead to a certain adversarial outcome, if the perturbations are restricted to well-defined semantic categories (e.g., "blue" can be perturbed to any other color name) (Guo et al., 2021; Yuan et al., 2021).