A Appendix

A.1 Additional Results for Prompt Inversion with CLIP

We provide more qualitative results in Figure 9.

For each example in Figure 3, we use the following templates respectively: “a tiger in the style of {}”, “the streets of Paris in the style of {}”, “a rocket in the style of {}”, where {} is replaced with the hard prompts:

```
resonvillains stargazing illustration tutorials sma internationalwomensday
watercolor fiberlilycamila yokohama -sorrow fluids latest
```

```
npr anime novels pureibanganesha irvin paints encapsulmondo
illustrillustroversized sultanconan ¢
```

for experiments 1 and 2, respectively.

Table 2: Quantitative results on learned hard prompts. We report the CLIP score between the original images and the images generated by the hard prompts.

Method	#Tokens	Requirement	LAION	MS COCO	Celeb-A	Lexica.art
AutoPrompt$_{SGD}$	8	CLIP	$0.689_{\pm 0.001}$	$0.669_{\pm 0.003}$	$0.595_{\pm 0.001}$	$0.702_{\pm 0.001}$
FluentPrompt	8	CLIP	$0.688_{\pm 0.001}$	$0.671_{\pm 0.005}$	$0.583_{\pm 0.004}$	$0.702_{\pm 0.002}$
PEZ (Ours)	8	CLIP	$0.697_{\pm 0.001}$	$0.677_{\pm 0.001}$	$0.602_{\pm 0.003}$	$0.711_{\pm 0.002}$
CLIP Inter.	~ 77	C. + Ba. + BL.	0.707	0.690	0.558	0.762
PEZ + Bank	8	CLIP + Bank	$0.702_{\pm 0.001}$	$0.689_{\pm 0.001}$	$0.629_{\pm 0.003}$	$0.740_{\pm 0.001}$
PEZ + 5 Seeds	8	C. + 5 Seeds	0.705	0.692	0.614	0.722
C. I. w/o BLIP	~ 77	CLIP + Bank	0.677	0.674	0.572	0.737
CLIP Inter.	8	C. + Ba. + BL.	0.539	0.575	0.360	0.532
CLIP Inter.	16	C. + Ba. + BL.	0.650	0.650	0.491	0.671
CLIP Inter.	32	C. + Ba. + BL.	0.694	0.663	0.540	0.730
Soft Prompt	8	CLIP	0.408	0.420	0.451	0.554

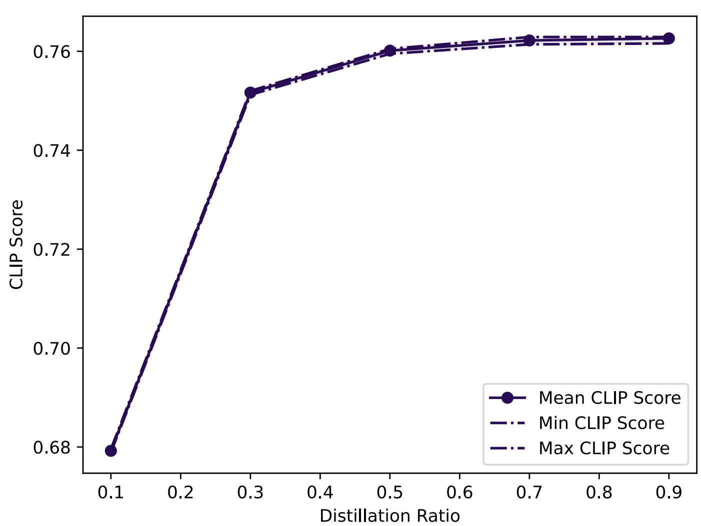

Figure 8: Quantitative results on prompt distillation with different distillation ratios. The CLIP score is calculated between the images generated by the original prompt and the images generated by the distilled prompt.

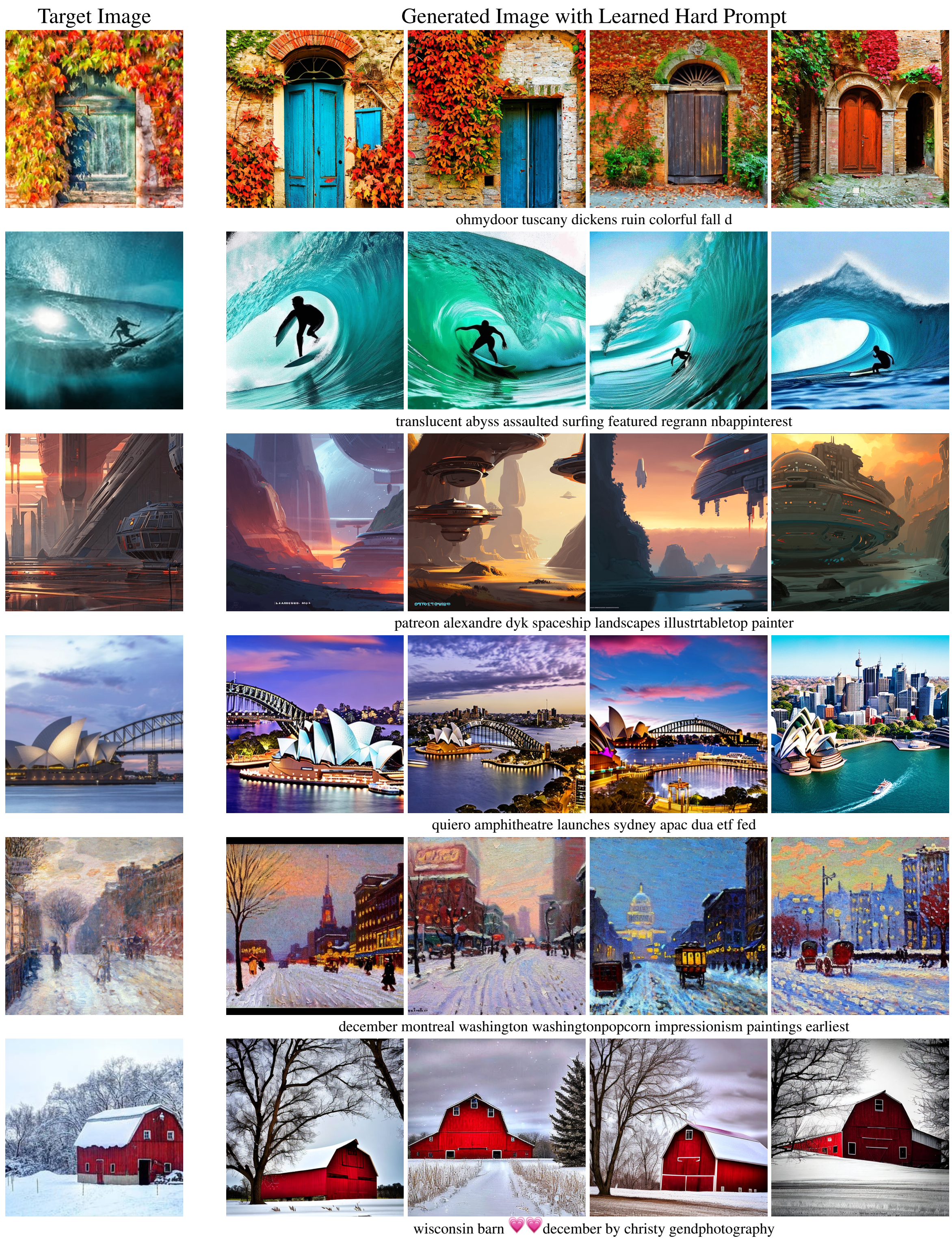

Figure 9: Additional qualitative results with learned hard prompts.

Figure 10: Iteratively evade `Midjourney` content filter and remove sensitive words/tokens.

A.2 Text-to-Text Experiments

In this section, we compare our Algorithm 1 to its counterparts in text-to-text setting, which is the more classical setting. We can see here that we are comparable to other gradient methods outperforming on the classification dataset, AGNEWS. Furthermore, we find that our method transfers better.

Table 3: Accuracy and standard deviation on the SST-2 validation set across the fives prompts for each method trained on GPT-2 Large and transferred onto larger models ranging from 1.3B to 6.7B. The baseline accuracy of a *soft prompt* is $\mathbf{93.35}_{\pm\mathbf{0.01}}$ (optimized for GPT-2 Large), but cannot be transferred. Note $\text{Empty}_{\text{Template}}$ refers to no prompt at the front but containing the predetermined template.

Method	GPT-2 Large (755M, **Source**)	GPT-2 XL (1.3B)	T5-LM-XL (3B)	OPT (2.7B)	OPT (6.7B)
$\text{Empty}_{\text{Template}}$	80.84	73.85	52.75	72.48	58.72
$\text{AutoPrompt}_{\text{SGD}}$	$87.56_{\pm 0.48}$	$78.19_{\pm 6}$	$56.01_{\pm 3.74}$	$73.69_{\pm 3.64}$	$65.28_{\pm 3.91}$
FluentPrompt	$\mathbf{88.33}_{\pm 0.48}$	$78.53_{\pm 6.3}$	$55.64_{\pm 1.33}$	$70.39_{\pm 4.66}$	$61.74_{\pm 2.8}$
$\text{Ours}_{\text{No Fluency}}$	$88.12_{\pm 0.21}$	$77.8_{\pm 7.71}$	$61.12_{\pm 6.57}$	$76.93_{\pm 2.88}$	$71.72_{\pm 7.06}$
$\text{Ours}_{\text{Fluency}}$	$88.05_{\pm 0.76}$	$\mathbf{79.72}_{\pm 7.3}$	$\mathbf{63.3}_{\pm 5.14}$	$\mathbf{77.18}_{\pm 8.54}$	$\mathbf{72.39}_{\pm 4.07}$

In the context of prompting in the text-to-text setting, the goal of Algorithm 1 is to discover a discrete sequence of tokens, the hard prompt, that will prompt the language model to predict the outcome of a classification task. As an important property of text is its fluency, Shi et al. [2022] find that fluency can increase a prompt's readability and performance. Thus, we define the optimization objective in this section as a weighted function of task loss and fluency loss,

$$\mathcal{L} = (1 - \lambda_{\text{fluency}})\mathcal{L}_{\text{task}} + \lambda_{\text{fluency}}\mathcal{L}_{\text{fluency}}.$$

We set $\lambda = 0.003$ similar to Shi et al. [2022] for all methods, and we ablate our method without fluency ($\lambda = 0$), which we denote as *no fluency*. We set out to show that hard prompts generated by this approach are successful both when transferring between a number of transformer-based language models, and when used to discover prompts in few-shot settings. An attractive quality of these prompts, especially for language applications, is that they can be optimized on smaller language models and then transferred to other, much larger models.

A.3 Datasets and Setup

We evaluate Algorithm 1 against related algorithms on three classification tasks, two sentiment analysis tasks, SST-2 [Socher et al., 2013] and Amazon Polarity [McAuley and Leskovec, 2013], and a 4-way classification task, AGNEWS [Zhang et al., 2015]. We build on the setting explored in Ding et al. [2022] and optimize hard prompts using GPT-2 Large (774M parameters) [Radford et al., 2019] with the Adafactor optimizer [Shazeer and Stern, 2018] and a batch size of 32 [Lester et al., 2021a].

Transferability Set-up. To test transferability, we generate prompts from GPT-2 Large for 5000 steps. We then select the five prompts with the highest average validation accuracy for each technique and test them on larger models. We test the transferred text on: GPT-2 XL, T5-LM-XL, OPT-2.7B, and OPT-6B [Radford et al., 2019, Lester et al., 2021b, Zhang et al., 2022], verifying the reliability of the proposed algorithm over related techniques and testing whether the hard prompt can reliably boost performance. Thus, we also consider a baseline of empty prompts, with only the template.

Few-Shot Setup. For the few-shot setting, we optimize each prompt for 100 epochs on GPT-2 Large on the AGNEWS dataset, where we sample two examples ($k = 2$) and four examples ($k = 4$) from each class to obtain the training set. Additionally, we create a holdout set of the same size, and finally validate the prompts on the entire validation set.

A.4 Results

We verify that our method is comparable to other methods in the sentiment analysis setting outperform the other methods on AGNEWS by about 2%. See Table 4 for details.

For Table 4, we report the best validation accuracy across three learning rates (0.1, 0.3, and 0.5), and for *FluentPrompt* and *AutoPrompt*$_{\text{SGD}}$ we used the learning reported (1, 3, and 10) and follow Shi et al. [2022] for the remaining hyperparameters for *FluentPrompt*. For these experiments, we *prepend* our 10 token prompt to each input text. We employ early stopping for all methods using a hold-out set of 5000 examples for each dataset, evaluating every 100 steps.

Table 4 shows that we are comparable to other methods in sentiment analysis and outperform the other methods on AGNEWS by about 2%. Examining the prompts, we find prompts are not coherent English for any of the methods. However, it does produce relevant tokens and phrases. For example, our method for SST-2 with the fluency constraint produced "*negative vibeThis immatureollywood MandarinollywoodThis energetic screenplay*." [3] This suggests the optimization process is finding relevant words to the task but lacks the ability to create full sentences.

Table 4: Validation accuracy for 10 discrete tokens trained **prepended at the beginning of the input text**. Best accuracy across three learning with standard error reported over 5 speeds.

Method	SST-2	AGNEWS	Amazon
AutoPrompt$_{\text{SGD}}$	$87.56_{\pm 0.35}$	$74.36_{\pm 0.47}$	$87.75_{\pm 0.17}$
FluentPrompt	$\mathbf{88.33}_{\pm 0.35}$	$74.62_{\pm 0.24}$	$87.42_{\pm 0.18}$
Ours$_{\text{No Fluency}}$	$88.12_{\pm 0.15}$	$\mathbf{77.06}_{\pm 0.20}$	$87.70_{\pm 0.21}$
Ours$_{\text{Fluency}}$	$88.05_{\pm 0.55}$	$76.94_{\pm 0.48}$	$\mathbf{87.78}_{\pm 0.19}$
Soft Prompt	$93.35_{\pm 0.01}$	$92.76_{\pm 0.01}$	$94.65_{\pm 0.01}$

Prompt Transferability. Table 3 shows for each method the five prompts trained on GPT-2 Large transferred to other LLMs. Interestingly, simply scaling a model–with no additional training–does not guarantee that the model will scale perform according on SST-2.[4] We see that all gradient-based methods are able to transfer compared to evaluating just the template, finding that our prompts trained with the fluency constraint transfer better than the other prompts. Additionally, we can see the largest boost from OPT-6.7B with our fluent method with about a 14% increase over just the template baseline. Additionally, we see our AGNEWS prompts are able to transfer from GPT-2 Large to GPT-2 XL in Table 5.

Table 5: Shows the validation accuracy with standard deviation from transferring hard prompts learned on GPT-2 Large to GPT-2 XL.

Method	GPT-2 Large (755M)	GPT-2 XL (1.3B)
Empty$_{\text{template}}$	58.34	52.42
AutoPrompt	$74.36_{\pm 0.47}$	$63.79_{\pm 3.61}$
FluentPrompt	$74.62_{\pm 0.24}$	$61.57_{\pm 5.1}$
Ours$_{\text{No Fluency}}$	$77.06_{\pm 0.20}$	$59.45_{\pm 8.63}$
Ours$_{\text{Fluency}}$	$76.94_{\pm 0.48}$	$67.59_{\pm 2.67}$

[3] Although we initialize the tokens with the label tokens, when examining the prompt over the optimization process, all tokens moved away from the initial tokens. This suggests that the process was able to relearn the class label.

[4] A quick experiment with and without the template on GPT-2 Large and XL showed that the template boosts performance differently for different models.

Table 6: Average validation accuracy with standard error on AGNEWS with k examples/shots per class using early stopping (including soft prompt) for all methods across 100 seeds for three tokens **append to the end of the text** similar to the original template ("It was about"). We set $\lambda = 0.03$ for these experiments. "Empty" is the template with no additional prompt.

Method	k=2	k=4
$\text{Empty}_{\text{Template}}$	58.34	58.34
$\text{Ours}_{\text{No Fluency}}$	$70.07_{\pm 0.81}$	$73.99_{\pm 0.45}$
$\text{Ours}_{\text{Fluency}}$	$70.93_{\pm 0.60}$	$74.15_{\pm 0.48}$
Soft Prompt	$74.92_{\pm 0.58}$	$79.93_{\pm 0.36}$

Prompt Discovery. Table 6 shows that even with just a few shots we can achieve high validation accuracy compared to our prepended counterparts. It is worth noting that each few-shot run takes about 5min. We ran 100 seeds where the training set contains k samples each class and did a quick examination of the top prompts, and although many of the prompts were gibberish, many of them were coherent. For example, even for $k = 2$, some of the prompts included news sources like "*BBC*", while other prompts found new approaches to the news classification task considering the text coming from a blog: "*Brian blog,*" or "*Blog Revolution analyze*." Due to the efficiency of these gradient-based methods, these methods can allow new ways for prompt engineers to discover novel prompts.