Optimizing Prompts for Text-to-Image Generation

Yaru Hao[*] **Zewen Chi**[*] **Li Dong**, **Furu Wei**
Microsoft Research
https://github.com/microsoft/LMOps

Abstract

Well-designed prompts can guide text-to-image models to generate amazing images. However, the performant prompts are often model-specific and misaligned with user input. Instead of laborious human engineering, we propose prompt adaptation, a general framework that automatically adapts original user input to model-preferred prompts. Specifically, we first perform supervised fine-tuning with a pretrained language model on a small collection of manually engineered prompts. Then we use reinforcement learning to explore better prompts. We define a reward function that encourages the policy to generate more aesthetically pleasing images while preserving the original user intentions. Experimental results on Stable Diffusion show that our method outperforms manual prompt engineering in terms of both automatic metrics and human preference ratings. Moreover, reinforcement learning further boosts performance, especially on out-of-domain prompts. Data and pretrained models are available at https://aka.ms/promptist.

1 Introduction

Generative foundation models can be prompted to follow user instructions, including language models [BMR+20, CND+22, SPN+22], and text-to-image models [RPG+21, RDN+22, SCS+22, RBL+22]. It has been recognized that prompt design plays an essential role in the generation quality. We need to adjust the prompt to make the model better understand our intentions and produce higher-quality results [RM21, ZMH+22]. The problem is severe in text-to-image models because the capacity of their text encoders, such as CLIP text encoder [RKH+21] in Stable Diffusion [RBL+22], is relatively small. Empirical observations also confirm that common user input is often insufficient to produce aesthetically pleasing images with current models.

Prior efforts implement manual prompt engineering towards specific text-to-image models [LC21, Opp22, Par22], typically adding some modifiers to the original input. However, it is laborious and sometimes infeasible to conduct manual prompt engineering. Besides, the manually engineered prompts often cannot be transferred between various model versions. Therefore, it is necessary to find a systematic way to automatically align user intentions and various model-preferred prompts.

In this work, we propose a prompt adaptation framework for automatic prompt engineering via reinforcement learning. Specifically, we first perform supervised fine-tuning with a pretrained language model (e.g., GPT) on a small collection of manually engineered prompts. The finetuned model is used to initialize the prompt policy network for reinforcement learning. Next, the model is trained by exploring optimized prompts of user inputs, where diverse beam search [VCS+16] is used to ensure generation quality and diversity. The training objective is to maximize the reward, which is defined as a combination of relevance scores and aesthetic scores of generated images. The relevance score reflects how much the original user intentions are retained after prompt adaptation. The aesthetic score indicates what degree the generated images are aesthetically pleasing.

[*] Equal contribution.

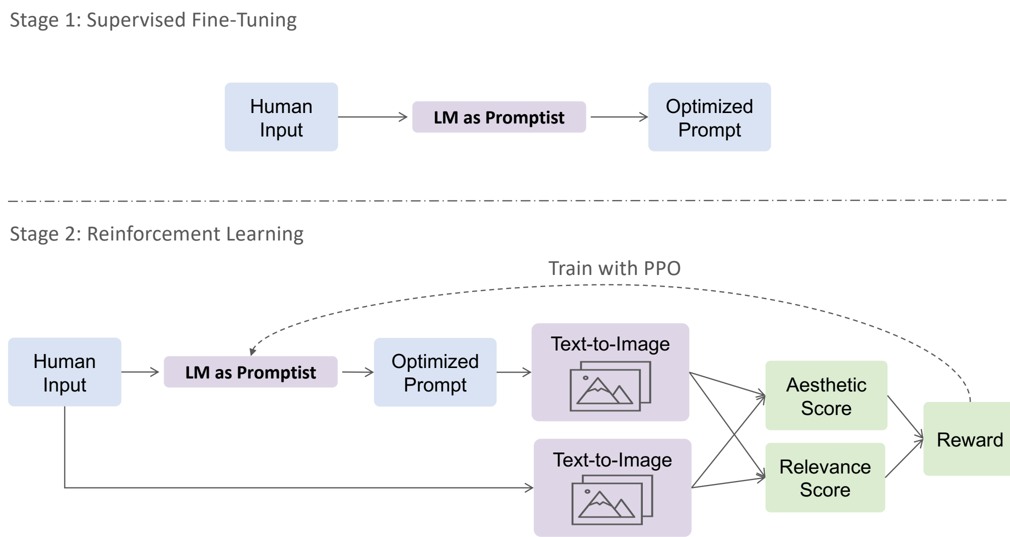


Figure 1: Overview of PROMPTIST training: (1) supervised fine-tuning (SFT) on manually engineered prompts; (2) reinforcement learning (RL) to increase the rewards of generated images after prompt optimization.

We conduct experiments with the publicly available Stable Diffusion models [RBL+22]. We evaluate our method using both the automatic reward metric and human preference ratings. Experimental results show that the optimized prompts outperform human-engineered ones and the original inputs. Human preference ratings also show consistent improvements across in-domain and out-of-domain prompts. Moreover, we find that reinforcement learning is more favorable than supervised fine-tuning, especially on out-of-domain user inputs. Overall, we show that language models can serve as a prompt interface that optimizes user input into model-preferred prompts.

2 Methods

The goal of our prompt adaptation framework is to automatically perform prompt engineering. Given user input of the text-to-image generator, our model learns to generate model-preferred prompts that obtain better output images while preserving their original intentions. Figure 1 presents the overview of our method. The prompt optimization model is named PROMPTIST, which is built upon a pretrained language model, such as GPT [BMR+20]. We first collect a set of human-engineered examples and use them to conduct supervised fine-tuning (Section 2.1). Next, we perform reinforcement learning (Section 2.3) to maximize the target reward (Section 2.2), which improves both relevance and quality of generated images.

2.1 Supervised Fine-tuning

Initialized with a pretrained generative language model, the policy model is first finetuned on a set of prompt pairs before reinforcement learning. A parallel prompt corpus $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})\}$ contains prompt pairs of original user inputs $\boldsymbol{x}$ and manually engineered examples $\boldsymbol{y}$. The training objective is to maximize the log-likelihood with teacher forcing:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \log p(\boldsymbol{y}|\boldsymbol{x}) \quad (1)$$

where the finetuned weights are used to initialize the policy network in reinforcement learning.

Collect Human Demonstrations We collect human-engineered prompts from Lexica[2]. Most prompts are composed of two parts, i.e., main content that describes the user's intention, and some modifiers that customize the art style, such as artist names, and popular elements. We use the

[2] https://lexica.art

crawled human-engineered prompts as targets. In order to have parallel data, we use three methods to construct their source inputs. First, we extract the main contents by trimming the modifiers and regard them as original user inputs. Second, we randomly remove or shuffle some modifiers and use the remaining texts as source inputs. Third, we use OpenAI GPT to rephrase the main contents and the human-engineered prompts, respectively. We find that the template "`[Input] Rephrase:`" works well in practice and translates input to a more user-friendly version.

2.2 Reward Definition

We measure the quality of optimized prompts from two aspects, namely relevance and aesthetics. The goal motivates us to define the reward function $\mathcal{R}(\cdot)$ from the above two perspectives.

First, we measure whether the generated images are relevant to the original input prompt after prompt adaptation. To be specific, we first sample images by the text-to-image model using the original prompt and the optimized prompt, respectively. Then, we compute CLIP [RKH+21] similarity scores to measure how relevant the generated images and the original input are. The resulting relevance score is defined as:

$$f_{\text{rel}}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{i_{\boldsymbol{y}} \sim \mathcal{G}(\boldsymbol{y})}[min(20 * g_{\text{CLIP}}(\boldsymbol{x}, i_{\boldsymbol{y}}) - 5.6, 0)] \tag{2}$$

where $i_{\boldsymbol{y}} \sim \mathcal{G}(\boldsymbol{y})$ means sampling images $i_{\boldsymbol{y}}$ from the text-to-image model $\mathcal{G}$ with $\boldsymbol{y}$ as input prompt, and $g_{\text{CLIP}}(\cdot, \cdot)$ stands for the CLIP similarity function. Notice that we always compute the similarity between the generated images and the original input prompt, which ensures the relevance score reflects the user preferences. If the relevance score is relatively reasonable, we encourage the model to generate more aesthetically pleasing images.

Second, we employ the aesthetic predictor[3] to quantify the aesthetic scores. The predictor adds a linear estimator on top of a frozen CLIP model, which is trained by human ratings in the Aesthetic Visual Analysis [MMP12] dataset. The aesthetic score is defined as:

$$f_{\text{aes}}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{i_{\boldsymbol{x}} \sim \mathcal{G}(\boldsymbol{x}), i_{\boldsymbol{y}} \sim \mathcal{G}(\boldsymbol{y})}[g_{\text{aes}}(i_{\boldsymbol{y}}) - g_{\text{aes}}(i_{\boldsymbol{x}})] \tag{3}$$

where $g_{\text{aes}}(\cdot)$ denotes the aesthetic predictor, and $i_{\boldsymbol{y}}, i_{\boldsymbol{x}}$ are the images generated by the prompts $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively. Notice that both $g_{\text{CLIP}}(\cdot)$ and $g_{\text{aes}}(\cdot)$ require the CLIP model, so we can share the CLIP forward pass during reward computation.

Finally, we define the reward function by combining the above scores with an additional KL penalty, which is between the policy model $\pi_{\boldsymbol{\theta}}$ and the supervised finetuned model π_{SFT} with coefficient η:

$$\mathcal{R}(\boldsymbol{x}, \boldsymbol{y}) = f_{\text{aes}}(\boldsymbol{x}, \boldsymbol{y}) + f_{\text{rel}}(\boldsymbol{x}, \boldsymbol{y}) - \eta \log \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}|\boldsymbol{x})} \tag{4}$$

where the last KL term is added to mitigate the overoptimization issue [OWJ+22].

2.3 Reinforcement Learning

After supervised fine-tuning, we further finetune our model with reinforcement learning. We employ proximal policy optimization (PPO) [SWD+17], which is empirically data-efficient and of reliable performance. We consider an autoregressive text generation problem, which is a Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ with a finite state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $\mathcal{R}$, state-transition probability function $\mathcal{P}$, and a discount term γ. In an episode of prompt adaptation, the initial state $\boldsymbol{s}_0 \in \mathcal{S}$ is the input prompt with n tokens $\boldsymbol{s}_0 = (x_1, \ldots, x_n)$ where each token x is from a finite vocabulary $\mathcal{V}$. At t-th time step, the agent selects an action $y_t \in \mathcal{V}$ according to the current policy model $y_t \sim \pi(y|\boldsymbol{s}_t)$. With a deterministic state transition, the next state is $\boldsymbol{s}_{t+1} = (x_1, \ldots, x_n, y_1, \ldots, y_t)$. The episode ends when the agent selects an end-of-sentence action. The reward of a state-action pair is given by the reward function $\mathcal{R}(\boldsymbol{s}_t, y_t)$. The goal of the agent is to maximize the accumulated expected reward $\mathbb{E}_{\boldsymbol{s}_0, y_t} \sum_t \gamma^t \mathcal{R}(\boldsymbol{s}_t, y_t)$.

Let $\pi_{\boldsymbol{\theta}}$ denote the policy model to be trained, we optimize the following objective:

$$\mathcal{J} = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}', \boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}}[\mathcal{R}(\boldsymbol{x}, \boldsymbol{y})] \tag{5}$$

[3] https://github.com/christophschuhmann/improved-aesthetic-predictor

where we implement both the policy model π_{θ} and the value function model as generative language models, with the language modeling head and the regression head, respectively. The parameters of the two models are initialized from the supervised finetuned policy model π_{SFT} and are optimized during reinforcement learning. The supervised finetuned model π_{SFT} and the score function model are frozen during training. Besides, we employ the clipped probability ratios [SWD+17] to avoid large policy updates.

3 Experiments

We conduct experiments on the publicly available text-to-image model Stable Diffusion 2.0 [RBL+22]. We use the DPM solver [LZB+22] to accelerate image sampling and set the denoising steps to 20.

3.1 Data Collection

For supervised fine-tuning, we collect 90k target prompts and construct four types of source prompts as described in Section 2.1, obtaining 360k paired data in total. At the reinforcement learning stage, we only require source prompts and the policy can explore better rephrasings itself. We use three types of data: (1) in-domain prompts from DiffusionDB [WMM+22], we use the user input for exploration and the manually engineered prompt for comparison, (2) out-of-domain image captions from COCO dataset [CFL+15], (3) ImageNet-21k image labels, the sizes of which are 600k, 600k and 40k respectively.

3.2 Settings

For the policy model, we use GPT-2 [RWC+19] with 1.5B parameters, which is a multi-layer Transformer [VSP+17] decoder pretrained with causal language modeling.

Supervised Fine-Tuning We finetune GPT-2 to predict the target prompt conditioned on the source prompt with teacher forcing. The input format is `[Source] Rephrase:[Target]`. We use a batch size of 256, a learning rate of 5e-5, and a max length of 512. We finetune the model for 15k steps and choose a slightly underfitting checkpoint according to the validation loss.

Reinforcement Learning We train the policy with Proximal Policy Optimization [SWD+17, PPO]. The value and policy network are initialized from the supervised finetuned model. The parameters of the value function are separated from the policy to avoid excessive competition between two objectives. To guarantee the quality and diversity of exploration, we adopt diverse beam search [VCS+16] with a beam size of 8 and a diversity penalty of 1.0. The maximum number of generated tokens is set to 150. We randomly choose one of the returned prompts to update the policy. We generate three images per prompt and compute the average reward to reduce variance. We train the policy for 12k episodes, four PPO epochs per batch with one minibatch each, with a batch size of 512 and a constant learning rate of 5e-5. The value loss coefficient and the KL reward coefficient are kept at 2.3 and 0.2 respectively.

Evaluation In order to evaluate how text-to-image models benefit from the prompt adaptation, we compare the reward value computed by two automatic predictors specified in Section 2.2. In addition, we use human preference ratings to demonstrate real user feedback. We evaluate our method on held-out data from training distribution, including in-domain data from DiffusionDB and out-of-domain COCO data. The former has corresponding manually engineered prompts for comparison, and the latter is used to verify whether our method can generalize to new domains. Except for the user input and manually engineered baseline, we also consider the supervised finetuned model as a baseline that can reflect the importance of reinforcement learning.

3.3 Results

Optimized prompts bring higher reward improvements than manual engineering. We evaluate optimized prompts on held-out data by generating three images for each prompt and computing the average reward value. Results in Figure 2 show that the reward value can be improved regardless of the engineering method, which suggests the misalignment problem between user-friendly prompts and model-preferred prompts is serious. Compared with the strong baseline of manually engineered

Figure 2: Reward comparison of optimized prompts with other baselines on in-domain and out-of-domain data, which indicates that the text-to-image model benefits a lot from our method.

In-Domain			Out-of-Domain		
SFT	RL	Gain	SFT	RL	Gain
0.29	0.36	**+24%**	0.28	0.48	**+71%**

Table 1: Absolute reward improvements of supervised fine-tuning and reinforcement learning. It is observed that RL generally outperforms SFT-only and performs better on out-of-domain data.

prompts, optimized prompts can still achieve considerable reward improvements. Furthermore, optimized prompts perform even better on out-of-domain data. It can be seen that automated prompt engineering can well align prompts between two different domains from users and text-to-image models respectively.

We provide some images generated by user input and optimized prompts in Figure 2. We observe that images generated by user input are intuitively uninspiring while optimized prompts not only retain the original intentions but also induce the model to produce more remarkable results. For example, generated images are crude when prompted with “A rabbit is wearing a space suit”. After prompt optimization, generated images become more bright and expressive.

Reinforcement learning can further boost the reward value. As shown in Table 1, reinforcement learning brings 24% and 71% average improvements on in-domain and out-of-domain data. Reinforcement learning in our method is supposed to perform better on out-of-domain data through explorations. To quantify its effect, we compute the ratio of reward improvements after fine-tuning and reinforcement learning. In-domain prompts from DiffusionDB are very similar to the data we used in supervised fine-tuning, so reward improvements are relatively saturated in the first stage and improvements of reinforcement learning on them are correspondingly smaller. Oppositely, out-of-domain data such as COCO captions are more similar to user input and unseen during the first stage. The policy must learn to adapt better to new domains through exploration, so their improvements on these prompts are more obvious. It indicates that given appropriate human queries, reinforcement learning can optimize them to adapt to different domains and boost reward improvements.

3.4 Human Evaluation

The reward function of our model is defined by two automatic metrics, aesthetic score and relevance score predicted by neural networks, which may have some discrepancies from real human feedback. Therefore, we additionally evaluate whether optimized prompts actually make humans more satisfied. We generate two images for each prompt, using the original user input and the optimized prompt. Afterward, three annotators are asked to rank the two groups of images in preference order and we calculate the average preference distribution. Evaluation results are shown in Table 3. We observe that annotators generally prefer images generated by optimized prompts over their original input. Compared with manually engineered prompts, optimized prompts yield less gain over user input. It

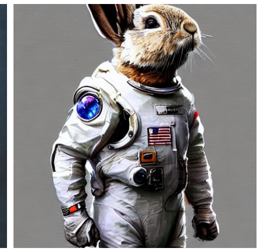
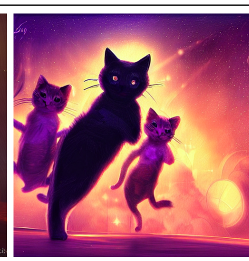
User Input	Optimized Prompt
A rabbit is wearing a space suit	A rabbit is wearing a space suit, digital Art, Greg rutkowski, Trending cinematographic artstation
 	 
Several railroad tracks with one train passing by	several railroad tracks with one train passing by, hyperdetailed, artstation, cgsociety, 8 k
 	 
A basket ball court in a military barracks, looks like an old grass mat after years of water damage.	a basket ball court in a military barracks, looks like an old grass mat after years of water damage, by greg rutkowski and thomas kinkade, trending on artstation.
 	 
Cats dancing in a space club	Cats dancing in a space club, digital painting, artstation, concept art, soft light, hdri, smooth, sharp focus, illustration, fantasy,
 	 

Table 2: Images generated by user input and optimized prompts using Stable Diffusion. We observe that optimized prompts can generate more aesthetically pleasing images.

	Optimized vs. User Input	Optimized vs. Manually Engineered
In-Domain	72% / 13% / 15%	49% / 21% / 30%
Out-of-Domain	67% / 12% / 21%	——

Table 3: Human evaluation results. The different colors represent how many images generated by corresponding prompts are considered more aesthetically pleasing. The orange block means that both prompts produce equally pleasing images.

suggests that the aesthetic score can measure the quality of generated images to some extent, it would be better if human feedback is included in the reward function.

4 Related Work

Prompt engineering. Manual prompt engineering is a natural way to optimize prompts. Manually designed cloze-style prompts have been used to probe knowledge from pre-trained language models [PRR+19, DDH+22]. In addition to knowledge probing, models are also prompted to handle a wide range of tasks with manually designed prefix prompts [BMR+20]. Despite the success of manually-crafted prompts, designing prompts takes time and experience [SLT+21] and can be sub-optimal [JXAN20]. Thus, various methods focus on automatically searching prompts by mining [JXAN20], paraphrasing [HBG21], and text generation [GFC21]. Besides, continuous prompt methods treat the prompts as additional continuous parameters of pre-trained models and directly optimize the parameters on downstream tasks [LL21, TMC+21, ZYLL22]. However, continuous prompt methods require access to manipulating the model, and the learned prompts lack interpretability. In contrast, our methods directly optimize prompts in text format, which can fit in black-box downstream systems such as text-to-image models.

Learning from human preference. Our work is related to research on learning from human preference, which has been widely studied in machine learning problems. Recent research on reinforcement learning from human feedback (RLHF) has shown promising results on machine learning problems, ranging from classical RL tasks [CLB+17, ILP+18] to a wide range of natural language processing tasks, including text summarization [SOW+20, ZSW+19], dialogue [JGS+19], and general text generation tasks [OWJ+22].

5 Conclusion

We propose to automatically optimize prompts for text-to-image models so that the user input and model-preferred prompts can be well aligned. We evaluate our method with Stable Diffusion. Experimental results show that our model outperforms human prompt engineering and supervised fine-tuning, in terms of automatic metrics and human evaluation. The exploration nature of reinforcement learning enables the model to go beyond teacher forcing, which improves generalization over out-of-domain examples. The proposed method is flexible to align human intentions and model-favored languages. Although our experiments are conducted on text-to-image models, the framework can be easily applied to other tasks. Rather than using automatic score functions as rewards, we can directly use human feedback as supervision to train a reward model [OWJ+22]. Moreover, using a larger-size language model as the prompt interface tends to improve the optimization quality.

References

[BMR+20] 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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[CFL+15] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. 2015.

[CLB+17] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[CND+22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek B Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Oliveira Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311, 2022.

[DDH+22] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[GFC21] Tianyu Gao, Adam Fisch, and Danqi Chen. 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, August 2021. Association for Computational Linguistics.

[HBG21] Adi Haviv, Jonathan Berant, and Amir Globerson. 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, April 2021. Association for Computational Linguistics.

[ILP+18] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[JGS+19] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

[JXAN20] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 07 2020.

[LC21] Vivian Liu and Lydia B. Chilton. Design guidelines for prompt engineering text-to-image generative models, 2021.

[LL21] Xiang Lisa Li and Percy Liang. 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, August 2021. Association for Computational Linguistics.

[LZB+22] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

[MMP12] Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2408–2415. IEEE, 2012.

[Opp22] Jonas Oppenlaender. A taxonomy of prompt modifiers for text-to-image generation, 2022.

[OWJ+22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[Par22] Guy Parsons. The dall·e 2 prompt book, 2022.

[PRR+19] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[RBL+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[RDN+22] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[RM21] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm, 2021.

[RPG+21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv*, abs/2102.12092, 2021.

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[SCS+22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[SLT+21] Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. Constrained language models yield few-shot semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7699–7715, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[SOW+20] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[SPN+22] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model, 2022.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[TMC+21] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[VCS+16] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *ArXiv*, abs/1610.02424, 2016.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008. Curran Associates, Inc., 2017.

[WMM+22] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022.

[ZMH+22] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers, 2022.

[ZSW+19] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[ZYLL22] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.