The CLIP Snake

Kevin Frans kvfrans@mit.edu
Jeremy Ma jma22@mit.edu
Andi Peng andipeng@mit.edu
Hyojin Bahng bahng@mit.edu

Abstract

Task specification is hard. Current methods in robot learning require either hand-tuned reward functions for reinforcement learning or expert-generated demonstrations for offline learning—neither of which are easily specified by an end user. One pathway towards developing more human intuitive methods of goal specification is to leverage the capabilities of CLIP, a large-scale vision-language model pre-trained on millions of natural examples. In this work, we aim to utilize the translation capabilities of CLIP to relate language, a natural avenue for human task specification, to images, a form that all robot states can be expressed as. We use cosine similarity between the two as a common language, or proxy reward, to train a "Snake" robot to maneuver itself into resembling various visual concepts defined by an end user. Specifically, we explore how to best utilize a potentially rich and dense reward signal generated automatically by CLIP in order to best design a visually expressive robot capable of taking on the form of a specification expressed by a human using natural language alone. We explore various prompting strategies, losses, reward densities, and robot configurations and find that CLIP can indeed be used as an effective reward signal for training an off-the-shelf RL algorithm.

1 Introduction

If we are to ever deploy robots in real homes where human users will need to interact with them, we shall necessitate an easy and intuitive method of communicating desired task specifications to them. Current methods in robot learning require either a hand-tuned and often brittle reward function for reinforcement learning, or expert or human-generated demonstrations for imitation or offline learning (Ghosh et al., 2019). However, both methods require large effort on the part of the designer, and thus are often unable to be easily deployed by an end user. In this work, we ask: are there other avenues for humans to communicate desired goals to the robot? It is easy and intuitive for an end user to relay a task specification in the form of a natural language instruction such as "go to the red car"—but how does the robot understand this desired goal? Moreover, how does the robot *ground* its own learning progress without needing to excessively query the human for a shaped reward or labeled goal state?

Recent works in foundation models (Bommasani et al., 2021) have enabled machine learning methods to make use of large amounts of natural world data to relate human-interpretable concepts like language and vision together. In this work, we use CLIP (Radford et al., 2021), a vision-language model pre-trained on millions of image-caption pairs from the Internet, as a translator between humans and robots. CLIP provides a distance measure between images and natural language, such that alignment of a photo of an apple and the text "red fruit" can be quantitatively measured. Consequently, it follows that similarity scores computed by CLIP can serve as a rich and diverse "reward" signal over a wide range of language specified tasks in any renderable robotic environment, where tasks can be defined as the robot taking on shapes that resemble visual concepts such as "apple" or "rainbow". Moreover, not only does a CLIP reward allow a more intuitive language for humans to communicate desired concepts to robots, it can also serve as a responsive reward to measure current robot learning without needing to query the human for more information. A reward generated from the CLIP-matching task distribution can therefore fulfill both desiderata: allowing end users

Preprint. Under review.

to specify desired task specifications in a language that is natural to them, while also preserving important aspects of a dense and responsive reward signal for supervising learning progress.

This work aims to leverage CLIP's language-image similarity score as a reward signal to train a "Snake" robot into resembling various visual concepts, specified by a human as a natural language description. Similar to a real-world snake or octopus which can deform and sometimes colorize itself to mimic objects, we define a 2D robot made of color-changing limbs and joints. This robot is controlled via a reinforcement learning agent trained using an off-the-shelf PPO algorithm, which aims to maximize similarity between the rendered robot image and a human-provided text description. A well-trained robot should learn to actuate its joints and shift colors to create a shape that resembles the desired task specification. In addition to assessing the feasibility of utilizing CLIP reward as an effective proxy to supervise learning, we also explore various prompting strategies, losses, reward densities, and robot configurations. Our results indicate that while alternative strategies such as prompt tuning and prompt ensembling can help with creating more robust goal specifications, the raw CLIP reward serves as a feasible and low-effort dense reward signal capable of supervising learning.

2 Related work

2.1 Challenges of Goal Specification

In traditional methods of reinforcement learning, an agent learns a policy for a single task via optimizing a human defined reward function. However, these functions are not only brittle and require excessive amounts of expert tuning, they have also been shown to be exploitable via "reward hacking" (Amodei et al., 2016). Recent approaches to rectifying this problem have moved away from rewards towards supervised learning approaches where we learn from demonstrations (Gupta et al., 2019; Kaelbling, 1993; Andrychowicz et al., 2017). However, these approaches require specified goal states from expert generated trajectories, tuning hindsight relabeling strategies for augmenting training signals, or suffer from sparsity of learning signal. An ideal manner of goal specification would enable the human (i.e. end user) to intuitively specify a desired task without needing to excessively supervise the training process.

2.2 Grounding Semantics with CLIP

CLIP (Radford et al., 2021) is a large foundational model pre-trained to align image and language features from millions of image-caption pairs from the Internet, and provides a powerful prior for grounding semantic concepts that are common across tasks, e.g., categories, parts, shapes, colors, texts, and other visual attributes. CLIPort (Shridhar et al., 2021) introduces a semantic stream using a pre-trained CLIP model to encode RGB images and language goal as input. MotionCLIP (Tevet et al., 2022) aligns the human motion manifold to CLIP space, implicitly infusing the extremely rich semantic knowledge of CLIP into the manifold. CLIPDraw (Frans et al., 2021) synthesizes novel drawings based on natural language input by utilizing a pre-trained CLIP as a metric for maximizing similarity between the given description and a generated drawing.

2.3 Natural Language Prompting

Prompting reformulates the downstream dataset into a (masked) language modeling problem, so that a frozen language model directly adapts to a new task. A prompt consists of constructing a task-specific *template* (e.g. "I felt so `[MASK]`") and *label words* (e.g. "happy/horrible") to fill in the blank (Gao et al., 2020). However, manually designing the right prompt requires domain expertise and significant amount of effort. Prefix tuning (Li and Liang, 2021) or prompt tuning (Lester et al., 2021) mitigates this problem by directly optimizing continuous prompts while having the model parameters fixed. We experiment with prompt tuning to better design the reward function.

2.4 Vision Based Manipulation

Traditionally, perception for robotic manipulation has centered on object detectors, segmentors, and pose estimators (Xiang et al., 2017; Zeng et al., 2017). However, the methods fail to generalize to to unseen objects without object-specific training data and often fail with deformable objects. An alternatively proposed thread of work uses end-to-end perception-to-action models to learn control

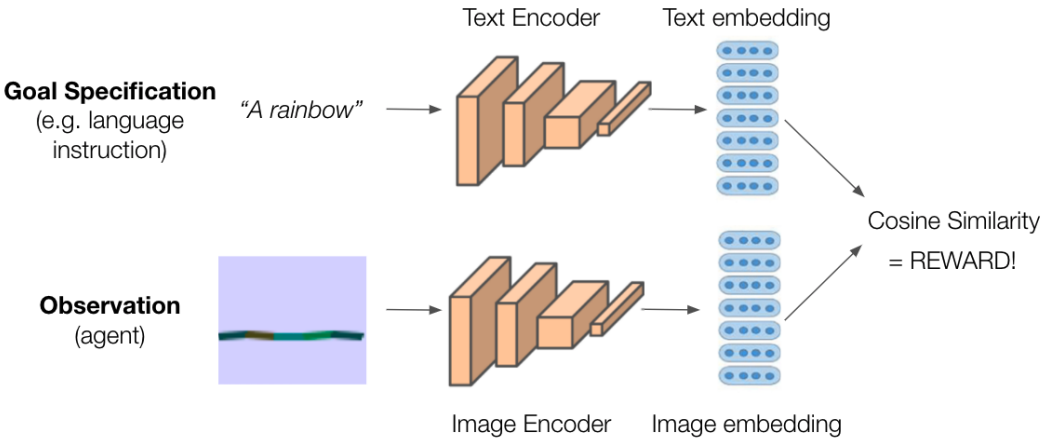


Figure 1: **We use CLIP as a natural translator between human goal specification and agent learning.** A human specifies a desired configuration such as "a rainbow" in natural language, which is then encoded via CLIP's pre-trained text encoder. In parallel, the current observation of the agent is rendered as an image and encoded via CLIP's pre-trained image encoder. We compute their cosine similarity and use the resulting score as a reward signal for supervising RL training.

policies (Zakka et al., 2020; Wu et al., 2019), but these methods have limited understanding of semantic concepts and rely on goal-images to condition policies. We desire a form of task definition that is both adaptable and dense enough to properly supervise learning.

3 Problem Formulation

The ultimate goal of this project is to train a simulated "Snake" robot to match a human-specified text description by learning to change its shape and color using a CLIP similarity score as the reward signal. To do so, we require a robot which is expressive enough to create a wide variety of renderable shapes, along with a reward function that is dense and diverse enough to act as a good training signal. Thus, the main contributions of this work are to:

1. Design a 2D robotic environment which is visually expressive, such that the robot can easily maneuver itself into a variety of shapes and colors;
2. Assess the feasibility of leveraging CLIP as a reward signal for robot learning, such that an off-the-shelf RL algorithm can successfully achieve the human-specified task;
3. Explore additional strategies such as prompt ensembling and tuning, different embodied configurations, loss functions, and varied reward densities in supervising training.

Our training pipeline involves training an agent using an off-the-shelf RL algorithm such as PPO while rendering the custom environment at regular intervals as images in order to calculate the CLIP similarity between the environment's current configuration and the human-provided text description. This approach provides a strong enough reward signal for the RL agent to be able to learn an optimal control policy to achieve a shape which maximizes similarity to the desired configuration.

3.1 Preliminaries

We consider an episodic finite-horizon high-dimensional environment that takes on the form of a Markov Decision Process (MDP) consisting of a set of states $\mathcal{S}$, actions $\mathcal{A}$, transition probabilities $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and reward function $R^* : S \to \mathbb{R}$. A learned policy $\pi : \mathcal{S} \to \mathcal{A}$ maps states to actions. The reward function R^* is typically defined by the dynamics of the simulated environment—in our case, we compute a similarity score between the rendered state and a user provided text prompt using CLIP as the reward signal (details below).

Let the desired task be specified by human-provided text description t. This specification can be as simple or as complex as the human desires. Using this input and a rendered image of the environment

as state $\mathcal{S}$, we can compute a cosine similarity score between t and $\mathcal{S}$ as a proxy reward signal for training any off-the-shelf RL algorithm. In our project, we use PPO with the objective defined below.

3.2 Environment, State and Action Space

We design a custom Box2D robot to run our simulations. The main body of our robot consists of five rectangles as "bodies" ($b_1, b_2 ... b_5$) and four joints ($j_1, ... j_4$) in connecting each pair, thus forming a line. The robot is able to control the rotation of each joint, allowing it to move in a snake-like fashion and change the color of each of it's five body segments. Using this snake-like robot, the robot can resemble a wide variety of colored shapes, e.g., "green L", "red W", or "blue circle".

The state space of the robot consists of the following attributes for each body segment (b_i) of the robot: x position ($b_i.x$), y position ($b_i.x$), RGB values of the current color ($b_i.RGB \leq 255$), the angle ($b_i.\theta$) along with it's cosine ($cos(b_i.\theta)$) and sine ($sin(b_i.\theta)$). In addition to attributes of the robot, the state space also includes three boolean flags indicating if the the CLIP reward is or has been calculated in the next frame, current or previous frame. This flag is necessary since we do not render and compute the CLIP similarity between the robot and the prompt at every step as it would be very computationally expensive. This yields a state space of size 43.

The action space of the robot consists of the motor speed of each joint ($j_i.vel$) and a delta in $b_i.RGB$ that is clipped between -10 and +10, allowing the RL agent to have control over the shape and color of the robot. This yields an action space of size 16.

3.3 Objective Function

The reward function of our RL agent is a sparse reward computed from the CLIP loss between the text description and the rendered snapshot of the robot at fixed timesteps. The text description t specifies a sensimotor goal that the robot must achieve, e.g. "green L", "blue circle". A common issue encountered with CLIP is the distribution gap between the input text and the pre-trained dataset. To bridge this distribution gap, a common strategy is introducing a prompt template, e.g., "a photo of a `{goal}`", to provide context about the content of the image. Therefore, we use a prompted description t_{text}, which is encoded with the pre-trained CLIP text encoder to produce a goal encoding $z_{\text{text}} \in \mathbb{R}^{512}$. Box2D renders the robot into a 64×64 picture, which is then upsampled to 224×224 to match the dimensions of input images used in CLIP. The rendered snapshot is processed through the CLIP image encoder to produce an image encoding $z_{\text{img}} \in \mathbb{R}^{512}$.

Let $\text{sim}(u, v) = u^\top v / \|u\| \|v\|$ denote the cosine similarity between u and v. We measure the cosine similarity between the image embedding z_{img} and text embedding z_{text} as our reward function. After conducting experiments, we found that $\text{sim}(z_{\text{img}}, z_{\text{text}})$ tends to fall between 0.2 and 0.4, so we linearly scale the value:

$$\mathcal{R}_{\text{sim}} = \frac{\text{sim}(z_{\text{img}}, z_{\text{text}}) - 0.2}{0.2} - 1$$

The above reward function is 0 when the similarity is 0.4 and -1 when the similarity is 0.2, with values in between scaled linearly. Finally, due to rendering being relatively computationally expensive, we compute $\mathcal{R}_{sim}$ once every n_{eval} timesteps and have a reward of 0 in other timesteps. This evaluation process is indicated to the model through three different boolean flags that are True when $t = n_{eval} - 1, n_{eval}, n_{eval} + 1$. This gives us the final objective function:

$$\mathcal{R}_{\text{obj}} = \begin{cases} \mathcal{R}_{\text{sim}} & t \mod n_{\text{eval}} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

3.4 Implementation Details

We use an off-the-shelf implementation of PPO (Schulman et al., 2017) from Stable Baselines 3 (Raffin et al., 2021) so that we can focus on designing a suitable reward function using CLIP and an expressive robot. We perform a full sweep of all hyperparameters for training timesteps, episode length, network architecture, activation, and learning rate. The actor and critic networks are MLPs with 2 layers of 64 units with tanh activation and learning rate 0.001. These agents are trained in the

Table 1: Mean rewards using different methods (higher is better).

Method	"Vertical Line"	"Green L"	"Red W"	"Blue Circle"	"Rainbow"	Mean
Language Loss						
Single Prompt (Octopus)	-0.898	-0.894	-0.798	-0.865	-0.797	-0.898
Single Prompt (Snake)	**-0.752**	**-0.717**	**-0.765**	**-0.816**	**-0.717**	**-0.754**
Pixel Loss (Snake)	-0.782	-0.891	-0.912	-0.892	-0.752	-0.846

Table 2: Mean LPIPS score using different methods (lower is better).

Method	"Vertical Line"	"Green L"	"Red W"	"Blue Circle"	"Rainbow"	Mean
Language Loss						
Single Prompt (Octopus)	**0.431**	0.521	0.744	0.681	0.688	0.613
Single Prompt (Snake)	0.467	0.494	0.737	0.655	0.681	0.607
Prompt Ensembling (Snake)	0.468	**0.481**	**0.726**	0.652	0.673	**0.600**
Prompt Tuning (Snake)	0.470	0.490	0.730	**0.648**	**0.667**	0.601
Pixel Loss (Snake)	0.491	0.573	0.767	0.660	0.691	0.636

above defined Box 2D environment with standard episode length of 30. For CLIP, we use ViT-B/32 which uses the Vision Transformer (Dosovitskiy et al., 2021) architecture for the image encoder.

We define a user-provided text input such as "a green snake shaped as the letter L" and use it to supervise the RL training via the objective above. We initialize the robot to the environment default, and render the state-space every 5 frames to produce an image. With this image and the defined language specification, we compute the CLIP similarity score according to the objective above and feed this back through the environment as the reward. We train for 50,000 timesteps using a single GPU on Google Colab. At test time, we initialize our trained policy and rollout for 30 timesteps.

4 Experiments

We perform and present experiments in simulation to answer the following questions: 1) How effective is a dense CLIP reward signal at supervising an off-the-shelf RL algorithm? 2) Does CLIP signal provide a more stable training paradigm than pixel loss? 3) How does the sparsity of the CLIP reward affect training? 4) Can other kinds of robot designs also be trained effectively? and finally, 5) Does different prompting methods offer better reward design?

4.1 Evaluation Metrics

We report the mean reward and LPIPS (Learned Perceptual Image Patch Similarity) (Zhang et al., 2018) score to measure the performance of the robot at resembling the specified task. The mean reward (i.e., normalized CLIP score) is across rollouts of 30 timesteps. LPIPS measures the perceptual similarity between two images in a way that coincides with human judgment. To evaluate whether our robot successfully achieved the goal, we collect 10 reference images from the Internet for each specified task and report the mean LPIPS score between each reference image and the screenshot of the rendered robot with the best CLIP score. We report performance over three random seeds.

4.2 Qualitative Results

As shown in Figure 2, the Snake robot successfully learns policies that resemble specified visual concepts. We test the performance of our method on a set of descriptive prompts, and examine the results through both human examination and by viewing the resulting CLIP and LPIPS metrics. In general, the robot reliably adapts its color to match the descriptive prompt, while creating a shape that contains the general features of the prompt subject. As is customary in RL optimization, we find that results have considerable variance based on the random initialization of the environment.

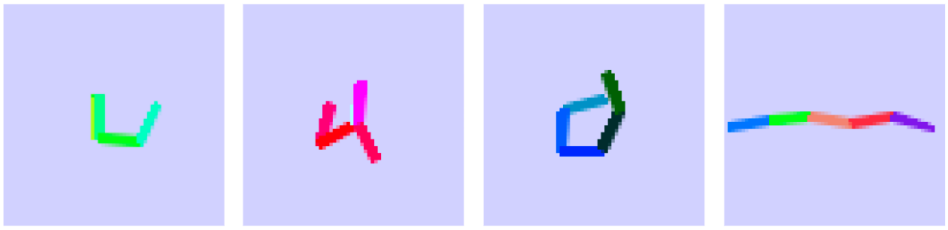


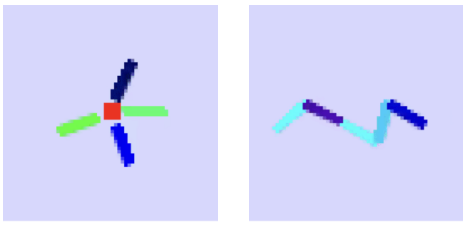
Figure 2: **We train a Box2D Snake robot to resemble visual concepts by using CLIP as a reward signal.** At each timestep, a rendering of the five-jointed robot is encoded via CLIP, and the cosine similarity between the rendering and a given prompt (e.g. "a blue circle" is given as reward. The robot is trained via an off-the-shelf implementation of PPO.

4.3 Language Loss vs. Pixel Loss

In this section, we aim to compare the performance of Snake robots trained via language (CLIP) versus an image-based loss function. Specifically, we compute the MSE pixel difference between the robot observation and a set of 10 base images. The intuition is that the CLIP-based loss can deliver a denser reward function, as well as a reward function closer to the human intent.

As shown in Table 1 and Table 2, CLIP-based loss outperforms using a pixel loss across every visual concept. Critically, CLIP-based loss captures the prior knowledge that a given text prompt may have multiple pixel solutions (e.g., there are many pictures matching "apple"), whereas an image-based loss forces the robot to resemble a specific picture. In addition, the pixel loss suffers from many local optima, as a robot that matches only half of the pixels may be incentivized to maintain its current position, as further improving involves sacrificing short-term reward. CLIP-based loss may solve this issue, as distance is calculated in a higher-level semantic space.

4.4 Varied Robot Design



(a) "Octopus" (b) "Snake"

Figure 3: Varied robot design.

A key trade off we had to balance was between the expressiveness and the complexity of the robot. Generally the more expressive the robot, the larger the possible state space, which can make training very unstable. We explore the this trade off by testing two different designs for the agent. As shown in Figure 3(a), CLIP "octopus" consists of a red square that acts as a fixed center hull and 4 arms rotating around the hull. We can control the torque and color of each arm just like CLIP snake. We compare this design with aforementioned CLIP "snake".

Table 2 shows the octopus achieves better perceptual similarity to simple visual concepts such as "vertical line". However, as we introduce various color and spatial semantics, the snake consistently achieves better results, as the octopus fails to resemble visual concepts that are more spatially complicated. Figure 4 shows that the snake achieves better CLIP similarity and LPIPS score steadily over training time, i.e., it is able to more stably and effectively improve its configuration to match the prompt.

4.5 Reward Sparsity

A key challenge with methods that move away from hand-crafted reward functions is the subsequent loss of a dense, rich reward signal to supervise training. A key benefit of using a pre-trained CLIP model's image and text encoders is the ability to generate a pseudo "reward" signal using the cosine similarity between our current agent's rendered state and the specified goal. This way, querying CLIP for this signal provides a natural, low-effort manner to generate a dense reward signal that does not require hand-tuned features from the human or labels collected throughout the course of training.

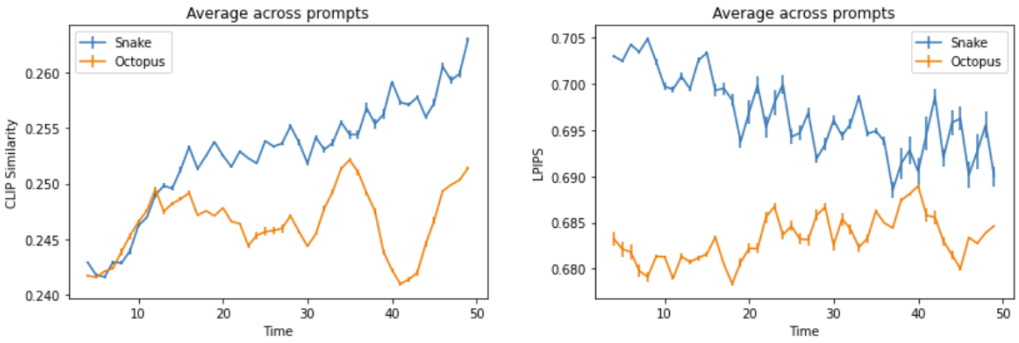


Figure 4: **Training with different robot design.** The snake achieves better CLIP similarity and LPIPS score steadily over training time, i.e., it is able to stably and effectively improve its configuration to match the prompt.

We investigate the effect of varying sparsities of reward signals by computing only the CLIP similarity score per defined time interval (see Figure 5). We perform a broad sweep across intervals ranging from 1, 5, 10, 20, and 30 (the max timesteps designed for the Snake environment) and find that first, CLIP does indeed act as a natural and low-human cost reward supervisor for an off-the-shelf learning algorithm and second, that a sparsity of 5 achieves optimal results. This reinforces our hypothesis that a well-tuned CLIP sparsity can serve as an appropriate proxy reward for reinforcement learning that is both low human effort as well as sufficiently rich in signal.

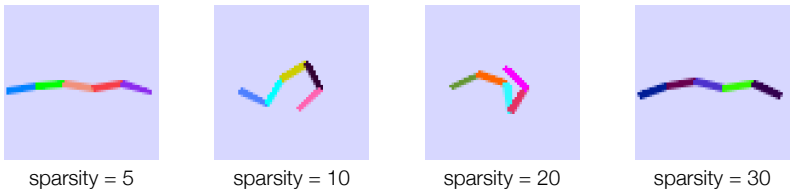


Figure 5: Results of using different sparsities of CLIP signal as the reward for training "a rainbow". We find that a sparsity of 5 achieves optimal results.

4.5.1 Prompt Ensembling and Prompt Tuning

A major challenge in deploying CLIP is prompt engineering: the process of creating a prompt that results in the most effective performance on the target task (Liu et al., 2021a). Identifying the right prompt is critical for performance, and a slight change in wording can have a huge impact, e.g., adding "a" before the class token brings more than 5% accuracy gain for Caltech101 (Zhou et al., 2021). We experiment with both prompt ensembling and prompt tuning in our reward design.

Prompt ensembling is a common technique to improve performance (Radford et al., 2021). We construct the ensemble over the embedding space instead of probability space, and utilize the 80 context prompts used in CLIP's zero-shot transfer to ImageNet. Some examples of prompts include: `''A close-up photo of a {goal}''`, `''A bright photo of a {goal}''`.

In practice, hand-crafting the best working prompt for a given task is laborsome. Instead of discrete text prompts, prompt tuning (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2021c,b; Zhou et al., 2021) is a recent technique that learns a "soft prompt" through backpropagation. We model a prompt's context words with learnable vectors, $t = [V]_1[V]_2 \cdots [V]_M[\text{GOAL}]$, where each $[V]_i$ is a vector that has the same dimension as word embeddings (i.e., 512 for CLIP), and M is the number of context tokens ($M = 16$ used in experiments). We forward the prompt t to the text encoder to produce z_{text}. To train the context vectors, we sample 10 reference images from the Internet that corresponds to our goal. We forward each image to the image encoder to produce z_{img}. We maximize the cosine similarity between z_{img} and z_{text}, only optimizing the learnable context vectors while keeping the pre-trained parameters frozen. We use the optimized vectors $t = [V]_1[V]_2 \cdots [V]_M[\text{GOAL}]$ to calculate our reward function.

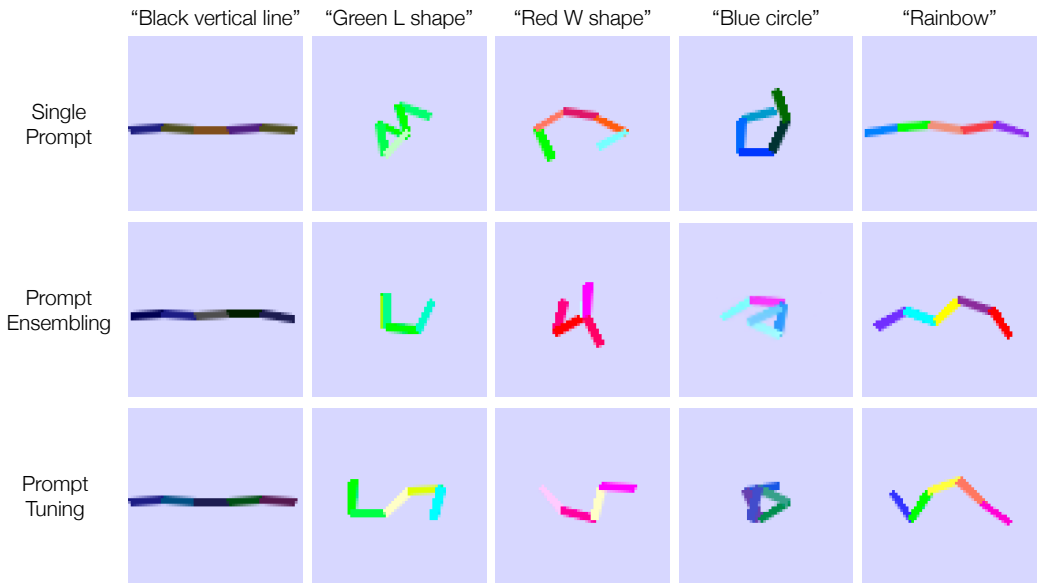


Figure 6: **Qualitative results using different prompting methods.** All methods generally capture the color semantics, but lack fine-grained manipulation of spatial structure.

Figure 6 show that all prompting methods are effective in recognizing color semantics. However, CLIP lacks the spatial understanding required for fine-grained manipulation, which has also been reported in Shridhar et al. (2022). For example, when spatial understanding is unnecessary, e.g., "rainbow", it successfully resembles the visual concept. However, across all examples, the robot fails to resemble the spatial structure, e.g., "vertical", "L", "W". As different prompting methods result in varied reward function, we compare the mean LPIPS score across three methods. Table 2 shows that using a single prompt is effective for simple visual concepts such as "vertical line". However, as we introduce different color and spatial semantics, prompt ensembling and prompt tuning generally achieves better results.

5 Conclusion

Our work aims to specify goals via natural language by leveraging a pre-trained vision-language model. We use CLIP's cosine similarity score as "proxy" reward to train a robot to resemble various specified visual concepts. We design a custom 2D robot made of color-changing limbs and joints trained via RL to maximize similarity between the rendered robot image and a human-provided text description. We experiment with various prompting strategies, losses, reward densities, and robot configurations. Our results demonstrate that CLIP can indeed serve as a feasible dense reward signal capable of supervising reinforcement learning. A clear limitation of this method is above identified issues of CLIP at understanding spatial structures that are critical to fine-grained robot manipulation. In scenarios where we require finely-tuned or specifically defined goal configurations it may be necessary to define the task specification via alternative methods. Regardless, we hope that our findings will spur further research into alternative (and more intuitive) forms of goal specification for robot learning in the age of large pre-trained models.

6 Contributions

Andi performed single prompt and reward density experiments, Kevin performed pixel vs. CLIP loss experiments, Jeremy performed varied robot configuration experiments, and Hyojin performed prompt ensembling and prompt tuning experiments. All team members contributed equally to the midterm report, presentation, and final report writing.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. (2017). Hindsight experience replay. *Advances in neural information processing systems*, 30.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Frans, K., Soros, L., and Witkowski, O. (2021). Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*.

Gao, T., Fisch, A., and Chen, D. (2020). Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C., Eysenbach, B., and Levine, S. (2019). Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. (2019). Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*.

Kaelbling, L. P. (1993). Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8. Citeseer.

Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., and Tang, J. (2021b). P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2021c). Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shridhar, M., Manuelli, L., and Fox, D. (2021). Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*.

Shridhar, M., Manuelli, L., and Fox, D. (2022). Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR.

Tevet, G., Gordon, B., Hertz, A., Bermano, A. H., and Cohen-Or, D. (2022). Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*.

Wu, Y., Yan, W., Kurutach, T., Pinto, L., and Abbeel, P. (2019). Learning to manipulate deformable objects without demonstrations. *arXiv preprint arXiv:1910.13439*.

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2017). Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*.

Zakka, K., Zeng, A., Lee, J., and Song, S. (2020). Form2fit: Learning shape priors for generalizable assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9404–9410. IEEE.

Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker, E., Rodriguez, A., and Xiao, J. (2017). Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1386–1383. IEEE.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2021). Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*.