CRITIC-GUIDED LEARNING TO SEGMENT REWARDING OBJECTS IN FIRST-PERSON VIEWS

Anonymous authors
Paper under double-blind review

ABSTRACT

We train a U-Net model to generate masks over reward-related objects in images. Our approach allows to train the U-Net model without explicit label information, but only using feedback from a critic model which learned to estimate the expected-reward value of an image observation. The masking is learned in contrastive fashion with image pairs using an adversarial scheme for employing the critic score gradient with respect to the mask operation: The pair consists of two images, where the first has a high and the second a low critic value. Training with such pairs enables the U-Net model to produce masks that decrease the critic value in the first image and increase the critic value in the second image when transferring pixels in the masked segment from the first to the second image. The training of the U-Net model is based on an imitation database from the NeurIPS 2020 MineRL Competition Track, where our agent took the ?-place winning entry. Video demonstration: `www.rebrand.ly/Rewarding-Objects-mp4`

1 INTRODUCTION

Exploration in environments with sparse feedback is a key challenge for deep reinforcement learning (DRL) research Bach et al. (2020); Harter et al. (2020); Schilling & Melnik (2018). Representation of reward-related objects can accelerate exploration and generalization. Humans explicitly learn object-centric representations König et al. (2018); Melnik et al. (2018b); Konen et al. (2019). Auxiliary targets generated in an unsupervised manner can accelerate learning Mirowski et al. (2016).

In this work, we aim to learn to segment objects using rewarding signals of interactions with the environment. Such a domain-agnostic approach can adapt to novel domains without significant changes. For example, auxiliary information about reward related objects in images help to efficiently find goal-directed actions Melnik et al. (2019; 2018a). Learning of the reward related objects can be leveraged from a large amount of unlabelled sensory data from a replay buffer or from human demonstrations Guss et al. (2021) to bootstrap learning for reinforcement learning tasks.

2 METHODS

We use an *Encoder-Decoder* architecture with skip connections inspired by *U-Net* Ronneberger et al. (2015). The Encoder has 5 convolution layers with 8, 8, 8, 16 and 32 channels respectively. The last layer results in a non spatial bottleneck (dimensions: 1x1x32). Each layer is followed by a LeakyReLU and we use max pooling after the first 4 layers. The *Decoder* has a mirrored structure but we switch the pooling layers with up sampling. Its output layer is passed through the Sigmoid function to produce the mask. The Critic shares the Encoder and after the Bottleneck additionally consists of two fully connected layers with 32 units each.

We tested our approach in a first-person Minecraft environments from the *MineRL 2020 Competition* Guss et al. (2021). The provided imitation learning database consists of data recorded from human players. We used only images (64x64x3 HSV color space) and recorded sparse reward signals from

*Shared first authorship

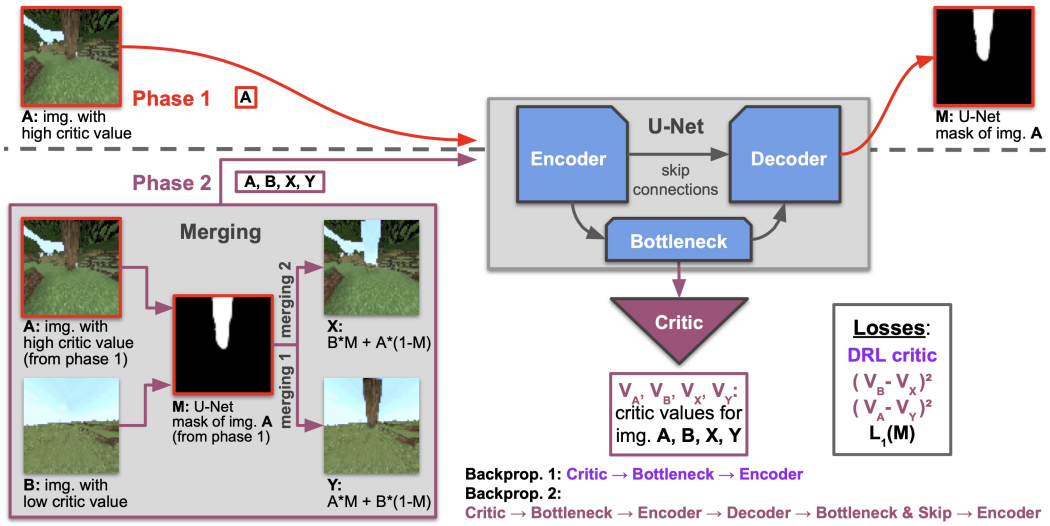

Figure 1: Phase 1 (highlighted in red): Image **A** (high critic value) passes through the *U-Net*, forming a mask **M**. Phase 2: the mask **M** is used to merge image **A** (high critic value) with image **B** (low critic value) resulting in image **X** (masked parts of **A** replaced with **B**) and image **Y** (masked parts of **A** injected in **B**). Images **A**, **B**, **X**, and **Y** are then passed through the encoder and critic. The losses penalize differences in critic values for image pairs **A** : **Y**, and **B** : **X**. Mask loss penalizes positive pixels of the mask and prevents convergence to a trivial solution when the mask **M** takes the entire image. Video explanation: `www.rebrand.ly/Rewarding-Objects-mp4`

the database to train the *Critic*. Every time a player collected the *log* item that appears after repeatedly using the *attack* action against a tree trunk, the agent received *reward = 1*. In all other steps the agent received *reward = 0*.

3 Training and Results

The training is divided into two *stages*. In stage one we train the *Critic* using the Bellman equation to predict the expected value of a state: $V_s = r_s + \gamma V_{s+1}$ where our states are 64x64x3 HSV single

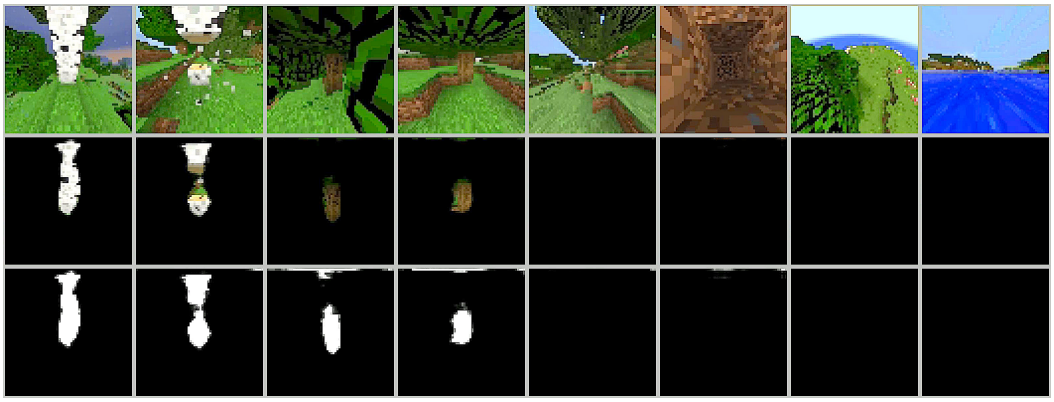

Figure 2: Segmentation results: The *U-Net* model learns to segment reward related objects (tree trunks) without any label information but only from reward signals. In the first four columns the trained *U-Net* model detects different instances of rewarding objects (white and brown tree trunks). The model is resistant to generation of false positive masks (columns 5-8). The first row shows the input images, the second row shows the masked segments of the input images, and the third row shows the *U-Net* generated masks. Video demonstration: `www.rebrand.ly/Rewarding-Objects-mp4`

image observations. We use a batch size of 128 and the mean squared error as loss function. After the first stage, we use the trained *Critic* to split the database into images with high critic values **A** and low critic values **B** for the stage two.

Stage two (mask training) is subdivided into two *phases* (shown in Fig 1) for training the *U-Net* to mask reward-related objects in images, as well as continue training the *Critic*. The challenge is that we don't have explicit ground truth masks as the training target. Instead, we pick image pairs such that always one member (image **A**) has high critic value and the other (image **B**) low critic value and use these pairs to implement a contrastive loss to detect reward-related objects. To this end, we use the *U-Net* to generate a mask **M** from the high-critic-value image **A** (*Phase 1*). Next, in *Phase 2*, we probe this mask for how strongly it happens to capture reward-related content in image **A**. We do this by swapping the pixels in images **A, B** that are in mask position, thereby generating a second pair of images: **X** (image **A** with masked pixels replaced by image content of **B**) and **Y** (image **B** with image content substituted by masked pixels from image **A**). Key idea then is to evaluate the ability of the generated mask **M** for capturing the reward-related content of image **A** by comparing critic values within pairs **A**, **Y** and **B**, **X**: for a "perfect" mask, the critic values in both pairs should be similar (high in the first pair, since **Y** has received all reward related content of **A**, and low in the second pair, since in **X** all reward-related content has become replaced by contents from low-reward image **B**). This translates into training the *U-Net* with the *Backpropagation 2* path where individual pixels of the mask produced in *Phase 1* are adjusted by gradients penalizing differences $V_A - V_Y$ and $V_B - V_X$ in *Critic*'s values for image pairs **A**, **Y** and **B**, **X** (Fig. 1). To better exemplify this: A poor mask would leave all essential image elements within **X**, contributing a high training loss, since then V_X remains high, while V_B is low; similarly, **Y** would hardly receive any reward related content from **A**, keeping V_Y low whereas V_A is high (Fig. 1). Using this adversarial contrastive training scheme allows us to generate high-quality masks highlighting reward-related objects for RL in a self-supervised, domain-agnostic manner.

In order to avoid trivial solutions like a full image mask replacing the complete images, we apply a linear regularization to enforce sparse masks. The loss is minimized if the mask highlights features that the critic uses to predict a high value. During stage two (*mask training*) the implicit mask loss will influence the *Encoder* and therefore also influence the *Critic*. That is why we continue the critic training as in stage one.

If in *Phase 1* (of stage 2) we only sample high value images for **A** (as indicated in Fig. 1), it is hard for the *Decoder* to learn to produce empty masks for low value images. Therefore, we also include a proportion of low value images for **A**. Every batch contains 128 images: 32 with high critic value and 32 with low critic value are used as **A** and 64 with low critic value are used for **B**. Note that this will not produce any "bad" gradients from the chosen error functions, since $V_A - V_Y$ and $V_B - V_X$ will both be small, this time because both images have similar (low) critic values from the outset. This allows the gradient from the regularization term that favors sparse masks to drive the *U-Net* response towards the desired empty mask output when receiving a low critic image as input. No such problem exists with regard to exclusively choosing low value images for **B**, since the *Decoder* produces its mask based on **A** and relies on the fact that **B** has a low value. *U-Net* segmentation results are shown in Figure 2.

4 DISCUSSION

In this contribution we showed that it is possible to generate high-quality masks depicting reward-related objects in images without explicit label information, but only using feedback from the critic model. This makes our approach flexible and domain agnostic.

Both of our training stages demand similar features. So instead of using a completely separated critic and *U-Net* model, they both share the encoder: In first stage the encoder is trained to create a meaningful feature representation of observation images that makes the expected-reward prediction from the critic possible. In the second stage the *Decoder* can use skip connections and the *Bottleneck* to access these representation, which we found greatly improves mask quality in comparison to a separate critic.

There are interesting possibilities for future work. RL has been applied successfully in various video games for training high-performing agents. Since these games often require perceptive modules

identifying progress-related objects, it seems obvious that our approach could boost exploration. Also an important sub-field in RL is to find ways to better apply RL within the real world. This is still a major problem since real world training would require a huge amount of training time and adapting an artificially trained model to the real world is difficult since simulation can not capture all conditions. Our approach could be used as an add-on module for realizing a faster and goal-directed training of the agent in real-life. For training this add-on, only unlabeled demonstrations and reward information would be necessary.

REFERENCES

Nicolas Bach, Andrew Melnik, Malte Schilling, Timo Korthals, and Helge Ritter. Learn to move through a combination of policy gradient algorithms: Ddpg, d4pg, and td3. In *6th International Conference, LOD 2020, Siena, Italy, Proceedings*, 2020.

William Guss, Stephanie Milani, Nicholay Topin, Brandon Houghton, Sharada Mohanty, Andrew Melnik, Augustin Harter, Benoit Buschmaas, Bjarne Jaster, Christoph Berganski, Dennis Heitkamp, Marko Henning, Helge Ritter, Chengjie Wu, Xiaotian Hao, Yiming Lu, Hangyu Mao, Yihuan Mao, Chao Wang, Michal Opanowicz, Anssi Kanervisto, Yanick Schraner, Christian Scheller, Xiren Zhou, Lu Liu, Daichi Nishio, Toi Tsuneda, Karolis Ramanauskas, and Gabija Juceviciute. Towards robust and domain agnostic reinforcement learning competitions: Minerl 2020. In *Proceedings of Machine Learning Research*, 2021.

Augustin Harter, Andrew Melnik, Gaurav Kumar, Dhruv Agarwal, Animesh Garg, and Helge Ritter. Solving physics puzzles by reasoning about paths. In *1st NeurIPS workshop on Interpretable Inductive Biases and Physically Structured Learning*, 2020.

Kai Konen, Timo Korthals, Andrew Melnik, and Malte Schilling. Biologically-inspired deep reinforcement learning of modular control for a six-legged robot. In *2019 IEEE International Conference on Robotics and Automation Workshop on Learning Legged Locomotion Workshop,(ICRA) 2019, Montreal, CA, May 20-25, 2019*, 2019.

Peter König, Andrew Melnik, Caspar Goeke, Anna L Gert, Sabine U König, and Tim C Kietzmann. Embodied cognition. In *2018 6th International Conference on Brain-Computer Interface (BCI)*, pp. 1–4. IEEE, 2018.

Andrew Melnik, Sascha Fleer, Malte Schilling, and Helge Ritter. Modularization of end-to-end learning: Case study in arcade games. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Workshop on Causal Learning*, 2018a. URL https://arxiv.org/pdf/1901.09895.pdf.

Andrew Melnik, Felix Schüler, Constantin A Rothkopf, and Peter König. The world as an external memory: the price of saccades in a sensorimotor task. *Frontiers in behavioral neuroscience*, 12: 253, 2018b.

Andrew Melnik, Lennart Bramlage, Hendric Voss, Federico Rossetto, and Helge Ritter. Combining causal modelling and deep reinforcement learning for autonomous agents in minecraft. *4th Workshop on Semantic Policy and Action Representations for Autonomous Robots at IROS 2019*, 2019.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Malte Schilling and Andrew Melnik. An approach to hierarchical deep reinforcement learning for a decentralized walking control architecture. In *Biologically Inspired Cognitive Architectures Meeting*, pp. 272–282. Springer, 2018.