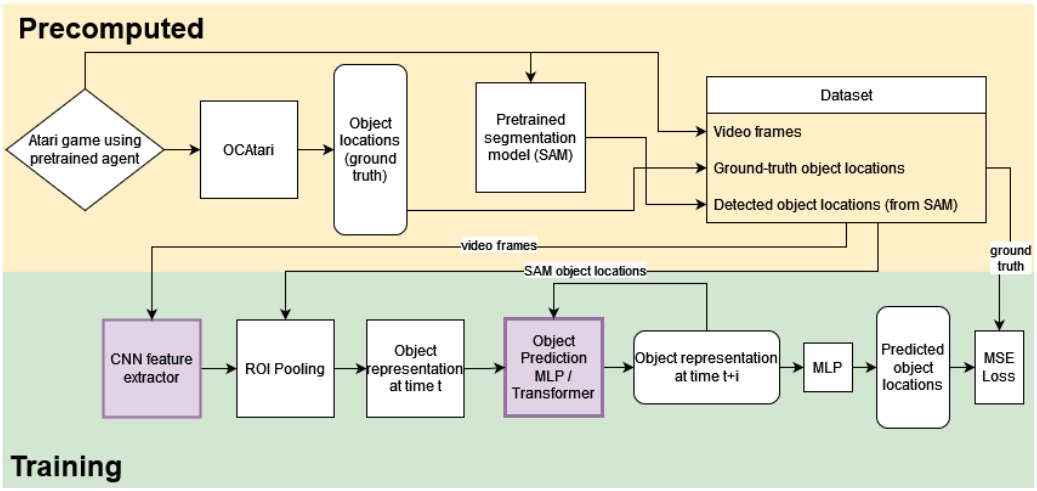
CSC413 Project Proposal

Colin De Vlieghere, Ben Edidin, Jasper Gerigk

Introduction: Data efficient Reinforcement Learning (RL) aims to achieve the superhuman performance RL is known for while using significantly less data. However, most approaches rely on a CNN backbone to extract the key information from the scene. Explicitly modeling objects could significantly speed up the learning behavior of RL agents as it removes the requirement for the CNN to learn to identify objects. We investigate if agents can benefit from pre-trained vision models' strong general performance by using them to extract the relevant objects' information from a game. Rather than training an entire RL agent with this additional information, which is infeasible given our constraints, we focus on predicting the future rollout of the game and learning a transition model capable of long-term predictions of the scene. This task is central for an agent to develop effective plans and play the game well as seen in [1]. Atari games have been a key benchmark for the progress in RL and are thus the focus of our work.

Background & Related Work: Most previous work has only learned transition functions for games in the context of building a complete RL agent. Therefore, there has been no direct previous work. Our approach is motivated by the work "Data-Efficient Reinforcement Learning with Self-Predictive Representations" (SPR) [1]. Building on the standard DQN model, SPR takes as input the current state, encodes it into a latent space, and in addition to determining the action, uses a CNN transition model to predict the next states in the latent space to encourage the model to learn how the environment behaves. Segment Anything is a multi-modal segmentation model created by Meta capable of predicting likely masks given a user's prompt [2]. It consists of transformer encoders for the image and prompt, followed by a transformer decoder to create the masks. The model was trained on 11M images with over 1B corresponding masks. However, since the model was only trained on real-world images, it might struggle with segmenting images from 2D games.



Method: As we are working on predicting objects in an Atari game, we can generate the data we need ourselves using a simulator. In addition to the standard image of the current game state, we need the ground truth data for objects to determine the quality of the masks detected by SAM [2]. This information is provided by OC Atari [3]. To generate the dataset we will use a pre-trained agent (from

https://bit-ml.github.io/blog/post/pretrained-atari-agents/), which ensures that we get diverse samples and longer sequences. We will crop the images to focus on the actual game area, ignoring UI objects. Before storing the data to be used in the training of the prediction model, we will generate and post-process the SAM masks to minimize the compute cost of the predictor training.
Our encoder, similarly to standard literature in RL on Atari [1], takes the last four time steps to get the current latent state of the object. The stacked objects allow the model to determine the current motion of the object at the first time step. The transition (prediction) model will be a transformer, which predicts the change in the latent representation of the object over one timestep. This prediction is conditioned on the current action from the agent, so that it can accurately predict the movement of the player character. The transformer architecture allows modeling interactions between different objects and is permutation invariant. We will apply the transformer repeatedly to predict farther into the future. To decode the position of the objects from the latent vector, we will use an MLP. The baseline model will use the same encoder and position decoder structure. The difference will be replacing our transformer with a simple MLP which should be able to predict simple motion patterns.

Project Plan: We will meet once a week at minimum (Thurs 10-11). We have a chat for the group and will notify other members as soon as possible if unexpected issues arise. We aim to get the first version of the model working in under two weeks, as most of the time will be spent tweaking the model to improve performance. We may compare several games and/or segmentation models for performance during this time, and all code will be committed on GitHub at https://github.com/Cubevoid/atari-obj-pred. Our deadline for the final version of the model will be the end of March. Rough outlines for dividing the work are as follows: Jasper will focus on the segmentation model, Colin will work on remaining parts of the preprocessing pipeline, and Ben will work on the position prediction model. After creating a first dataset and getting initial results, we will reallocate work as needed. We will then spend April running final experiments and writing up the report. We will rely on the GPUs from the teaching labs as well as personal ones for compute. We will scale the number of games examined depending on resources.

Risks and Ethics: As the group members have known each other for a while, we do not expect interpersonal issues or group members dropping the course. Risks we have identified include:

- **SAM is not viable for data collection on Atari.** This could either be due to low quality masks or high computational cost. In the first case, we could use ground truth information to fine tune SAM. In the latter, a model such as FastSAM [4] should help performance. Initial tests using SAM on sample Atari images have been promising, so we believe the masks should be of good enough quality.
- **Stochasticity makes prediction difficult.** Most Atari games are not deterministic. If this is an issue, we can focus on less random games, or we can consider implementing a more complicated transition model which supports stochasticity.
- **Model training time.** It is unlikely that we will have compute issues given our scope, as we are able to completely separate the dataset generation from the actual model. This means once we have generated the dataset once, we only need to learn the transition model which will be lightweight compared to SAM. However, fine-tuning SAM might be computationally difficult.

Given that our model does not use any personal information and is not directly applicable to real-world situations, we do not foresee any ethical issues. In addition, SAM was trained on privacy-respecting and licensed images [3].

References

[1] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. arXiv preprint arXiv:2007.05929, 2020.

[2] Quentin Delfosse, Jannis Bl¨uml, Bjarne Gregori, Sebastian Sztwiertnia, and Kristian Kersting. Ocatari: Object-centric atari 2600 reinforcement learning environments. arXiv preprint arXiv:2306.08649, 2023.

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. arXiv preprint arXiv:2304.02643, 2023.

[4] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. arXiv preprint arXiv:2306.12156, 2023.