Supplemental Material

A Confusion Matrix Analysis

To test out the alignment of the VLM with other domains we also perform our confusion matrix analysis on Berkeley Bridge Dataset [Ebert et al., 2021] and 2 human video datasets — one from a stock gif repository and another a dataset collected from the kitchen in our laboratory. The Bridge dataset videos consist of a robot teleoperated to interact with toy versions of everyday kitchen objects such as forks, knives, pans, etc. The stock photo set consists of random videos from disparate domains. The lab dataset is collected by the same actor and with the same camera angles, lighting, etc., with the goal of this small dataset to study whether removing confounders from the gifs like lighting and camera angles makes the task of video retrieval easier or not.

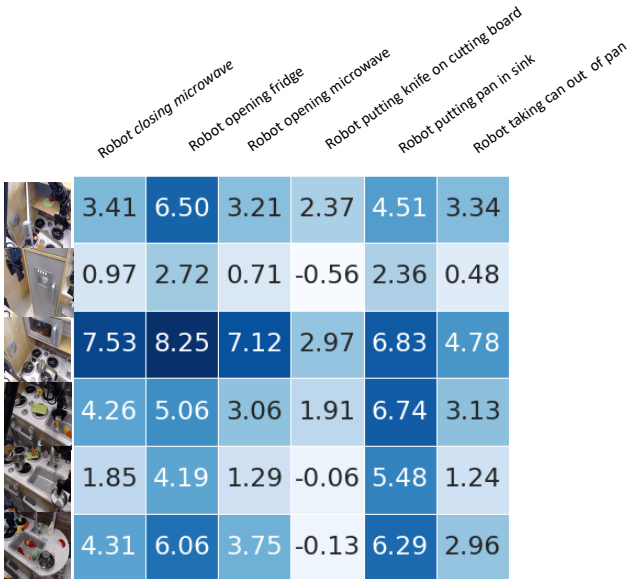

Figure 9: Domain Alignment of the Berkeley Bridge Dataset. We study how well videos taken from the Berkeley Bridge Dataset [Ebert et al., 2021] are correctly identified by the S3D [Xie et al., 2018] pretrained on HowTo100M [Miech et al., 2019]. Each entry (i, j) corresponds to the similarity score between the i^{th} video and the j^{th} text label. We find that the S3D model often gets distracted by background objects and mislabels the videos in this context. For example, for the last video of *robot taking can out of pan*, the model sees the sink and labels the video with a high score for the text label *robot putting pan in sink*.

As shown in Figure 9, the alignment of the Berkeley Bridge Dataset is poor. This is potentially due to the large amount of clutter and distractor objects in the background of the videos. This often leads to confusion during inference in model as it often correctly identifies the background objects and fails to attend to the task object. For example, for the last video of *robot taking can out of pan*, the model sees the sink and labels the video with a high score for the text label *robot putting pan in sink*. The sink is a prominent distractor object resulting in high scores across multiple videos (see vertical column corresponding to the label *Robot putting pan in sink*). Another mode of failure is typically in the confusion of object identities. This is especially prominent in the first 3 videos where the microwave and the fridge are repeatedly mixed up.

We subsequently study whether videos that are visually similar to the domain of HowTo100M [Miech et al., 2019] are correctly labeled by S3D. We find that videos are generally well aligned with all the videos we tested to be correctly labeled as seen in Figure 11. The model transfers to these video domains demonstrating appreciable generalizability including correctly labeling videos of inanimate objects like the video of a waterfall. The inference capabilities also generalize to a variety of camera views with the videos from a first-person camera. For example, *Human turning knob* and *Human opening microwave* are correctly classified in addition to the third-person video of *Human opening cabinet*.

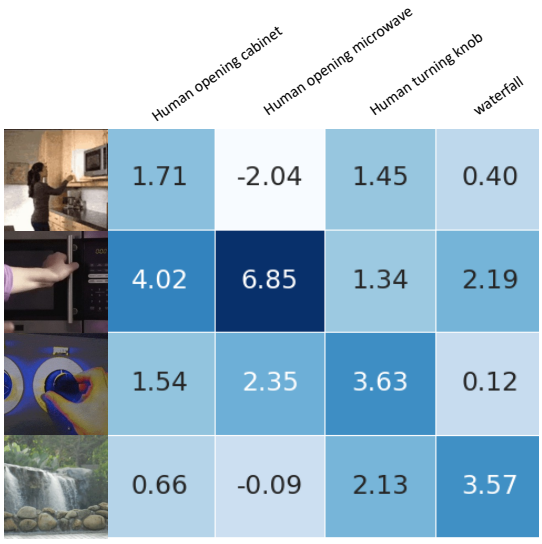

Figure 10: Domain Alignment of the Stock Gif Repository Dataset. We study how well videos taken from online Stock Gif Repositories are correctly identified by the S3D [Xie et al., 2018] pretrained on HowTo100M [Miech et al., 2019]. Each entry (i, j) corresponds to the similarity score between the i^{th} video and the j^{th} text label. We find that the model aligns well with videos that are visually similar to the domain on which it was trained, i.e., HowTo100M [Miech et al., 2019] indicating appreciable generalization capabilities. Each of the motions are correctly identified along with the objects, including videos of inanimate objects like the waterfall in the last video.

Finally, we study whether videos sourced from the same domain, i.e., videos without any confounders such as camera view, actors, etc., can be correctly labeled by the pretrained VLM. To this end, we use the dataset collected in our lab with the same actor, camera view, lighting etc., and perform the confusion matrix analysis. We find that this is generally successful as the model can generally identify objects well and confuses one type of door for another. It does not, however, possess a common sense understanding of the world as it cannot tell that turning a light on is achieved by the motion of flipping a switch.

B Implementation Details

B.1 Policy Learning

We use PPO [Schulman et al., 2017] to train our agents. However, to ensure slow yet stable convergence, we set the batch size for updates equal to the number of parallel environments times the number of steps per environment. This effectively ensures that the entire replay buffer is loaded at once to perform updates. We also set the number of epochs to 1. This ensures that the policy does stray from the trust region [Schulman et al., 2017] resulting in a slower, yet more stable convergence. This of course means that the training can be sped up by playing around with these parameters, but we find these settings to work in all the Metaworld experiments as well as Kitchen experiments.

We also find that increasing the weight for the entropy term in the loss for PPO provides significant improvements. We hypothesize this to be the case since the learned reward function contains two kinds of maxima — one which corresponds to the actual task defined by the descriptor and the other a spurious maximum corresponding to an adversarial video which, to the model, "looks" like the task descriptor but does not actually solve the task defined by the descriptor. This is expected as the VLM is never finetuned on environment-specific data and thus it is unreasonable to expect that the RoboCLIP reward will perfectly align with the downstream task reward. We find that setting a high coefficient for entropy term in the loss function of PPO [Schulman et al., 2017] improves the exploration bonus provided to the policy so that it explores multiple maxima in the RoboCLIP reward allowing the agent to solve the task by eventually optimizing the RoboCLIP maxima corresponding to task success.

The policy is implemented using StableBaselines3 [Raffin et al., 2021] with the following parameters:

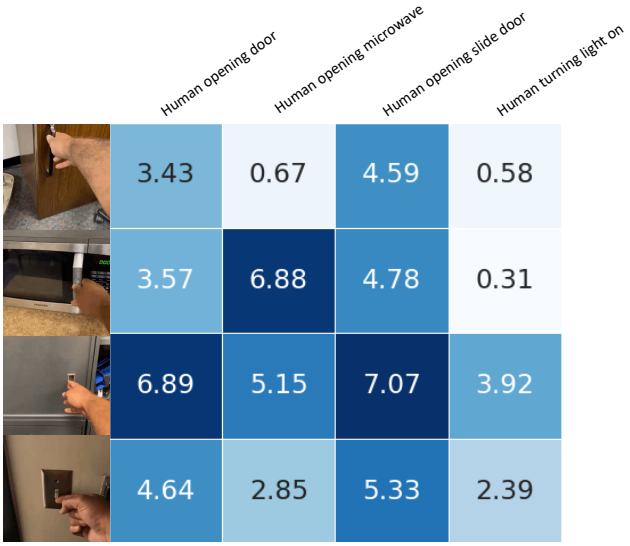

Figure 11: Domain Alignment of the Lab Dataset. We study how well videos collected from a first-person view in our lab kitchen are correctly identified by the S3D [Xie et al., 2018] pretrained on HowTo100M [Miech et al., 2019]. Each entry (i, j) corresponds to the similarity score between the i^{th} video and the j^{th} text label. We find that the model aligns well with videos that are visually similar to the domain on which it was trained, i.e., HowTo100M [Miech et al., 2019] indicating appreciable generalization capabilities. While the diagonal matrix is not diagonal-heavy, the model is generally able to identify door-like objects and is able to correctly identify a microwave. However, it is unable to correctly identify that a light switch turns a light on and lacks commonsense reasoning capabilities that to turn a light on, one needs to flip a switch.

- learning_rate=0.0003
- n_steps=128
- batch_size=2048
- n_epochs=1
- gamma=0.99
- gae_lambda=0.95
- clip_range=0.2
- clip_range_vf=None
- normalize_advantage=True
- entropy_coef=0.5
- value_function_coef=0.5
- max_grad_norm=0.5
- use_sde=False
- sde_sample_freq=-1
- target_kl_divergence=None
- stats_window_size=100
- seed=None
- device='auto'
- _init_setup_model=True

The policy network consists of a fully connected neural network with 2 layers and 64 hidden units per layer. The feature extractor for the Q function and the policy are shared. The remaining parameters are listed below:

- activation_function=tanh
- ortho_init=True
- use_sde=False
- log_std_init=0.0
- full_std=True
- use_expln=False
- squash_output=False

- features_extractor_class=Flatten
- features_extractor_kwargs=None
- share_features_extractor=True
- normalize_images=True
- optimizer_class=Adam
- optimizer_kwargs=None

B.2 Reward Generation

Here, we describe the implementation details for generating the RoboCLIP conditioned on various demonstration types. We find that in Metaworld, normalizing the video of the demonstration is not needed. Typically, images are stored as 3D arrays with width, height, and channels, and with a per-pixel value of $0 - 255$. S3D [Xie et al., 2018] pretrained on HowTo100M [Miech et al., 2019] was trained on per pixel values normalized to $0 - 1$. Experimentally, we found that not normalizing the images for the Metaworld experiments yielded higher zero-shot rewards. We also found this to occur when training on Kitchen videos. Not normalizing yielded benefits in terms of higher zero-shot rewards. We also shorten the episode videos from 128 timesteps per episode to a 32 frame video using uniform downsampling. This ensures that we conform to the S3D pretraining methodology.