A ENVIRONMENTS

A.1 DISTRACTING DMCONTROL

We increase the visual complexity of DMControl "walker" task (Tassa et al., 2018) by overlaying randomly sampled videos from the DAVIS 2017 dataset (Pont-Tuset et al., 2017), similar to Stone et al. (2021). In our experiments we use expert policies to collect offline datasets for the pre-training tasks of standing, forward walking, and backward walking respectively. To collect these datasets, we pre-train polices with SAC(Haarnoja et al., 2018a) in the state space and collect rollouts of the visual observation. We test our representation on the downstream task of "running".

Rewards: We use reward functions provided by DMControl "walker" task (Tassa et al., 2018). For the backward walking task, we invert the sign of walk speed in "forward" task defined in DMControl so that the agent gets higher reward when the agent moves backward instead of forward.

A.2 VIZDOOM

Our experiments use "D3 battle"environment provided by Wydmuch et al. (2018) where the agent's objective is to defend against enemies while collecting medi-packs and ammunition. For collection of the prior task dataset, we use the pre-trained models provided by Dosovitskiy and Koltun (2017) and vary the reward weighting parameters (described below) to produce a diverse set of behaviours.

Rewards: A reward function is defined by a linear combination of three measurements (ammunition, health, and frags) with their corresponding coefficients.

$$R_{\text{ViZDoom}} = c_{ammo} \cdot \frac{x_t^{amm} - x_{t-1}^{amm}}{7.5} + c_{health} \cdot \frac{x_t^{health} - x_{t-1}^{health}}{30.0} + c_{frags} \cdot \frac{x_t^{frags} - x_{t-1}^{frags}}{1.0}. \tag{5}$$

where x_t^{amm} is a measurement of ammunition, x_t^{health} is health of the agent, and x_t^{frags} is a number of frags at timestep t. The set of coefficients for ammunition, health and frags are represented as $(c_{ammo}, c_{health}, c_{frags})$. We use the coefficients of $(0, 0, 1)$, $(0, 1, 0)$, and $(1, 1, -1)$ for the prior tasks, and $(0.5, 0.5, 1.0)$ for the target task.

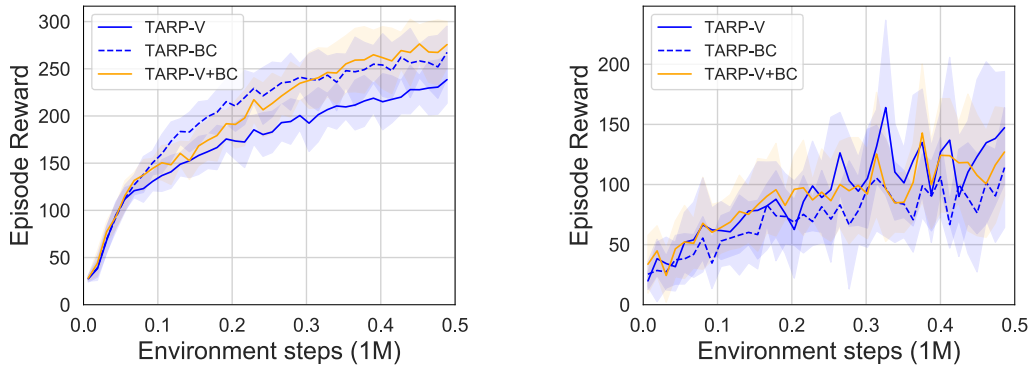
A.3 CARLA

In our experiments, we use the map of "Town05" from the CARLA environment (Dosovitskiy et al., 2017). At the beginning of each episode, we randomly spawn 300 vehicles and 200 pedestrians. The initial location of the agent is randomly sampled from a task set containing multiple start and goal locations. We collect the datasets for the tasks of intersection crossing, taking a right turn, and taking a left turn using pre-trained policies. We pre-train the policies with SAC (Haarnoja et al., 2018a) using segmentation masks of the environment as the input, and then use the learnt policies to collect rollouts of visual observations for the datasets.

Rewards: We use the same reward function for all of the tasks. The reward function consists of terms for speed, centering on a road, angle of the agent, and collision.

$$\begin{aligned}
R_{\text{speed}} &= \frac{v}{v_{min}} \cdot \mathbb{1}_{v \leq v_{min}} + (1.0 - \frac{v - v_{target}}{v_{max} - v_{target}}) \cdot \mathbb{1}_{v \geq v_{max}} + 1.0 \cdot \mathbb{1}_{v_{min} \leq v \leq v_{max}}. \\
R_{\text{centering}} &= \max(1.0 - \frac{d_{center}}{d_{max}}, 0). \\
R_{\text{angle}} &= \max(1.0 - |\frac{r}{(r_{max} \cdot \frac{\pi}{180})}|, 0). \\
R_{CARLA} &= R_{\text{speed}} + R_{\text{centering}} + R_{\text{angle}} - 10^{-4} \cdot collision_intensity.
\end{aligned} \tag{6}$$

where v is velocity of the agent, d_{center} is distance between the center of the road and the agent, r is angle of the agent. We use constant values of $v_{min} = 15.0$, $v_{max} = 30.0$, $v_{target} = 25.0$, $d_{max} = 3$, and $r_{max} = 20$.



(a) Distracting DMControl

(b) CARLA

Figure 9: Transfer performance for single source and multiple source task-induced representation. Our task-induced representation with multiple supervision performs slightly better or as good as the single source representation. Thus, task-induced representation learning framework is compatible with heterogeneous datasets.

B Data Collection for Multi-Source Task Supervision

We use supervision from heterogeneous data sources to learn task-induced representations. For the distracting DMControl environment, the dataset is composed of demonstrations for the "forward walking" and "backward walking" tasks, and a dataset with reward annotations for the "stand" task. For the CARLA environment, the task-induced representations are trained on a dataset composed of demonstrations for the "intersection crossing" task, and the datasets annotated with rewards for the "right turn" and "left turn" tasks. For both environments, the datasets are collected with the procedure described in Appendix A.

C Hyperparameters

C.1 Hyperparameters for RL

Table 1: Common SAC hyperparameter

Parameter	Value
Stacked frames	3
Optimizer	Adam
Learning rate	3e-4
Discount factor (γ)	0.99
Latent dimension	256
Convolution filters	$[8, 16, 32, 64]$
Convolution strides	$[2, 2, 2, 2]$
Convolution filter size	3
Hidden Units (MLP)	[1024]
Nonlinearity	ReLU
Target smoothing coefficient (τ)	0.005
Target entropy	$-\dim(\mathcal{A})$

Table 2: Distracting DMControl SAC hyperparameter

Parameter	Value
Observation Rendering	(64, 64), RGB
Initial steps	5×10^3
Action repeat	2
Replay buffer size	10^5
Minibatch size	256
Target update interval	1
Actor update interval	1
Initial temperature	1

Table 3: CARLA SAC hyperparameter

Parameter	Value
Observation Rendering	(128, 128), RGB
Initial steps	3×10^3
Action repeat	1
Replay buffer size	10^5
Minibatch size	128
Target update interval	2
Actor update interval	2
Initial temperature	0.1

Table 4: ViZDoom PPO hyperparameter

Parameter	Value
Observation rendering	(64, 64) Grey
Stacked frames	4
Action repeat	1
Optimizer	Adam
Learning rate	3e-4
PPO epoch	10
Buffer size	2048
Convolution filters	$[8, 16, 32, 64]$
Convolution filter sizes	$[2, 2, 2, 2]$
Hidden units (MLP)	$[256]$
Generalized advantage estimation λ	0.95
Entropy bonus coefficient	4e-3
Discount factor (γ)	0.99
Minibatch size	256
Nonlinearity	ReLU

Table 5: Hyperparameters for CQL

Environment	Trade-off factor α for Q-values	Number of action samples
Distracting DMControl	3.	1
ViZDoom	1.	1
CARLA	3.	1

C.2 HYPERPARAMETERS FOR PRE-TRAINING

In TARP-BC, we use recurrent neural networks to predict a sequence of actions over prediction horizon T to learn task-induced representations only in CARLA (see Table 7).

Table 6: Common model parameters

Parameter	Value
Batch size	128
Hidden units (MLP)	[256]
Learning rate	1e-4

Table 7: Hyperparameters for TARP-BC

Environment	LSTM hidden units	Prediction horizon
Distracting DMControl	—	-
ViZDoom	—	-
CARLA	512	8

Table 8: Hyperparameters for TARP-V

Environment	Discount rate
All environments	0.4

Table 9: Hyperparameters for TARP-Bisim

Environment	Reward predictive loss weight	Bisimulation loss weight T
Distracting DMControl	1.	0.1
ViZDoom	1.	10.
CARLA	1.	0.01

Table 10: Hyperparameters for VAE

Environment	β constraint
Distracting DMControl	100.
ViZDoom	100.
CARLA	10.

Table 11: Hyperparameters for Pred-S and Pred-R+S

Environment	β constraint	Prediction horizon T
Distracting DMControl	50.	6
ViZDoom	10.	6
CARLA	5.	8

Table 12: Hyperparameters for ATC

Environment	Random shift probability	Temporal shift
All environments	1.	3