We thank all the reviewers for their valuable feedback.

**R1: Clarity.** Thank you for providing detailed feedback; we will give more details as suggested. *Transporter frozen for KeyQN?* Yes, Transporter is trained with frames from a random policy and frozen during policy learning (fig. 5); this should be relaxed in the future (see also line 48).

**R1: Error bars / multiple runs. R2: Deep RL that matters.** Thank you for highlighting this. In fig. 6 we report the mean score of *3 runs* with different random seeds. As suggested, here we include the standard deviations (in parenthesis in the table on right).

| Game | **KeyQN** | SimPLe | Rainbow | PPO (100k) |
|---|---|---|---|---|
| breakout | 19.3 (4.5) | 12.7 (3.8) | 3.3 (0.1) | 5.9 (3.3) |
| frostbite | 388.3 (142.1) | 254.7 (4.9) | 140.1 (2.7) | 174.0 (40.7) |
| ms_pacman | 999.4 (145.4) | 762.8 (331.5) | 364.3 (20.4) | 496.0 (379.8) |
| pong | 10.8 (5.7) | 5.2 (9.7) | -19.5 (0.2) | -20.5 (0.6) |
| seaquest | 236.7 (22.2) | 370.9 (128.2) | 206.3 (17.1) | 370.0 (103.3) |

**R1, R4: Compare w/ exploration methods.** We use the keypoints for *options* based exploration instead of raw actions. This can be thought of as a learned action space, which is complimentary to learning based exploration methods [26], and combinations of these can be interesting future work.

**R1, R4: How is 'K' (number of keypoints) chosen?** We will make explicit that $K$ is a hyper-parameter. It is set to the maximum number of moving entities in each environment. If $K >$ entities, the extra keypoints stay in a constant position as seen in montezuma_revenge visualisation in the supplementary material.

**R1, R2: Tracking experiments.** We will include F1-scores for brevity. *Why evaluate over different time lengths?* Although the predictions are per-frame, this evaluation measures the consistency of correspondence of keypoints to entities over varying time lengths, which is essential for control. It is difficult to sustain this correspondence for long durations (*e.g.*, due to switching identities), as indicated by the general downward trend of baseline methods in fig. 4.

**R1: What is averaged feature vector in KeyQN?** The (Gaussian) heat-map ($\mathcal{H}_\Psi$) is multiplied with the feature tensor ($\Phi$) and then spatially averaged to obtained this feature vector. We will make this clearer.

**R2: Generalization claims.** We demonstrate learning representations for entities (keypoints) in a task-agnostic manner (without any rewards), which can later be re-purposed for (or generalize to) efficient task-specific reward based learning. It is true, that a single Transporter model has not been shown to generalize across a number of environments. This is an important form of generalization and a good candidate for future work.

**R2: Choice of testing environments + limited to moving objects.** We did not cherry pick environments given the empirical results and report everything we ran. We chose diverse environments that represent varying degrees of difficulty (number and motion of entities) for the Transporter network. For example, our model is not designed to capture – (1) static elements like walls in ms_pacman, or (2) moving background. In environments which pose such challenges our model is unable to capture all the relevant visual entities and performs worse than baselines, *e.g.*, bricks in breakout where PPO-500k is better (fig. 6 in paper). We will make this discussion more explicit.

**R2: Tracking multiple instances.** The reconstruction objective encourages detecting all the moving entities. Yes, large receptive field captures the unique context around each object to disambiguate b/w visually similar objects.

**R2: Explain "PointNet" better.** Thank you for the detailed comments, we will make these clearer (and rename *PointNet*). The stop-gradient is to prevent any cross-talk / co-adaptation between the source and target images, and use the network in inference mode for the source image. We will make this explicit and give empirical evidence.

**R2: Other comments.** *(1) Distance threshold value ($\epsilon$).* This threshold value for evaluation was set to the average ground-truth spatial extent of entities for each environment. We will add these details in the appendix. *(2) Augmentation techniques.* The time delay between source and target frame was randomly chosen between and 1 and 20 (sec. 4); no other augmentation strategy was used. *(3) Action repeat.* Each action was repeated 4 times (following [25]).

**R4: Trivial solutions.** In order to reconstruct the target image from source, the network has to detect everything that can change between two frames, *i.e.* learn to detect the moving entities. In Atari ALE, often objects of interest move, hence the proposed approach provides useful geometric representations. For limitations see line 28 above.

**R4: Why is keypoint bottleneck necessary?** Our objective is to learn object representations that are geometric and correspond to discrete entities. Such representations enable learning co-ordinate based options models for exploration and efficient reward based control. Moreover, in previous work [16], generic fully-connected representations of an auto-encoder were shown to be deficient for learning object keypoints (ref: ablation in table 2 of [16]).

**R4: Fix representation, learn agent.** This is the setting we consider in the KeyQN experiments (sec. 4.2.1). However, co-learning the representation (keypoints) with the agent is also important, especially for hard to explore environments. This is because random exploration (currently used for pre-training Transporter) cannot cover all parts of the environment (certain objects / rooms remain unseen). This is important future work.

**R4: Code release.** Thank you, yes we will share our implementation of the Transporter model.