

Feature Type	Home 1	Home 2	Office 1	Office 2	Lab 1
State+Action	0.851	0.683	0.700	0.666	0.880
State only	0.735	0.574	0.581	0.549	0.892
Position only	0.674	0.597	0.605	0.622	0.886

Table 6: **Feature Ablation Results:** Full state and action features (Sec. 3.1) yield best goal prediction results.

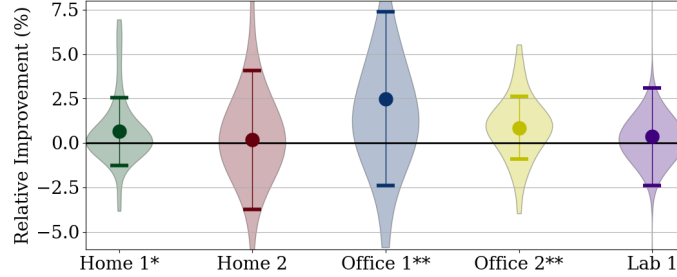


Figure 6: **Relative improvement from incorporating goal uncertainty.** Per-scene violin plots, means, and standard deviations are shown. Per-scene one-sided paired t-tests are performed, testing the hypotheses that incorporating goal uncertainty improves goal prediction performance. A * indicates $p < 0.05$, and ** indicates $p < 0.005$.

Appendix

A. Reward Function Feature Ablation Analysis

In Table 6, we show the mean true goal probability when labels are used as detectors (to isolate performance in the ideal case). While the purely positional representation of state performs well, it is almost always outperformed by the full representation of rewards that include features of both the full state and action. In Lab 1, the simpler representations slightly outperform the full, due to the relative simplicity of the high-level activities in Lab 1. Here, knowledge of the state and previous goal alone is highly predictive of future goal.

B. Incorporating Detection Noise

Current paradigms in vision often yield noise in the action and goal detectors necessary for DARKO. We first describe our method for incorporating uncertainty in each goal detection, then conduct a performance analysis with synthetic noise. Then, we analyze the performance with real, noisy goal detection. We find DARKO can still perform well with forms of noisy goal and action detection. We find incorporating goal uncertainty significantly improves performance with synthetic noise, and shows improvements in the real goal detector setting. These results show that DARKO can tolerate the effects of noise, and further support the claim that it can enjoy the benefits of better scene and activity detection algorithms.

Harnessing goal detection confidence: In many scenarios, probability $\rho_g \in [0, 1]$ may be associated with each goal detection. We designed an effective method for handling real-world uncertainty. For known perfect goals, $\text{SOFTVALUEITERATION}$ uses $V(g) = 0, \forall g \in \mathcal{S}_g$. Each goal is a maxima of $V(s) \in (-\infty, 0], \forall s \in \mathcal{S}$ and represents a reward of 1 in log space. *We replace each goal value with its log-probability: $V(g) = \ln \rho_g$, which has the effect of biasing the policy towards goals with greater certainty.* This results from the value iteration assigning higher value to states and actions closer to more certain goals, which makes the policy likelier to visit them. For example, if the goal detector yields a false positive of `bathroom` in the same area as a true positive detection of `kitchen`, the goal prediction posteriors for both goals will suffer, unless the false positive has an associated low ρ_g (high uncertainty), in which case the policy is biased towards the correct goal of `kitchen`.

Noise analysis: We first analyze DARKO under the effect of adding noise to the GT. We add incorrect goal detections with probabilities $\rho_g \sim \mathcal{N}(0.1, 0.05)$, under various amounts of noise inserted uniformly at random across time: from 10%, 20%, ..., 90% of the number of original goal detections. For every scene, at each noise amount, we sample noise 5 times, and run DARKO with and without goal uncertainty for each corrupted sample, resulting in 225 paired experiments that evaluate the average goal forecasting probability. Per-scene results are shown in Figures 6. *A one-sided Wilcoxon signed-rank test supports the hypothesis that incorporating high goal uncertainty yields better goal posterior prediction performance than not incorporating the uncertainty with $p < 10^{-7}$.*

C. Proof of Regret Bound

Our regret bound is:

$$\mathcal{R}_t \leq 2B\sqrt{2td}, \quad (8)$$

where B is a bound on the norm of θ , d is feature dimensionality, and t is the episode count (regret after the t 'th episode).

Proof. By Equation 2.5 of [23], the regret of online gradient descent is bounded:

$$\mathcal{R}_t \leq \frac{1}{2\lambda} \|\theta\|_2^2 + \lambda \sum_{i=1}^t \|\nabla_{\theta_i}\|_2^2. \quad (9)$$

By using bounds on $\|\theta\|_2^2$, $\|\nabla_{\theta_i}\|_2^2$, and a minimizing choice of λ , we will prove the result.

Writing the general gradient in terms of the expected features (and omitting the subscript t):

$$\begin{aligned} \|\nabla_{\theta}\|_2^2 &= \|\bar{f} - \hat{f}\|_2^2 \\ &= \bar{f}^T \bar{f} + \hat{f}^T \hat{f} - 2\bar{f}^T \hat{f} \end{aligned} \quad (10)$$

Using:

$$\begin{aligned} 0 &\leq \|x - y\|_2^2 = x^T x + y^T y - 2x^T y \\ 2x^T y &\leq x^T x + y^T y \\ 2(-x)^T y &\leq (-x)^T (-x) + y^T y \\ -2x^T y &\leq x^T x + y^T y, \\ \therefore -2\bar{f}^T \hat{f} &\leq \bar{f}^T \bar{f} + \hat{f}^T \hat{f}, \text{ (Setting } x = \bar{f}, y = \hat{f}) \end{aligned}$$

then Equation 10 becomes:

$$\begin{aligned} \|\nabla_{\theta}\|_2^2 &\leq \bar{f}^T \bar{f} + \hat{f}^T \hat{f} + \bar{f}^T \bar{f} + \hat{f}^T \hat{f} \\ &= 2\bar{f}^T \bar{f} + 2\hat{f}^T \hat{f} \\ &\leq 4d. \text{ (Since } \bar{f}, \hat{f} \in [0, 1]^d) \end{aligned} \quad (11)$$

Thus, using Equation 11 in Equation 9, and that the projection step of θ (constraining the set of θ to be the convex ball with radius B) ensures $\|\theta\|_2 \leq B$:

$$\begin{aligned} \mathcal{R}_t &\leq \frac{B^2}{2\lambda} + \lambda \sum_{i=1}^t 4d \\ &= \frac{B^2}{2\lambda} + 4\lambda td. \end{aligned}$$

With the minimizing choice of $\lambda = \frac{B}{2\sqrt{2td}}$,

$$\mathcal{R}_t \leq B\sqrt{2td} + \frac{2Btd}{\sqrt{2td}} = 2B\sqrt{2td}$$

■

D. Derivation of Other Inference Tasks

D.1. Action-Subspace Visitation

To derive the action-subspace visitation, we first use the posterior expected visitation count of performing an action a_y immediately after arriving at a state s_x is given in Equation 12, from [33].

$$D_{a_y, s_x | \xi_{0 \rightarrow t}} \triangleq \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{\tau=t+1}^T I(s_\tau = s_x) I(a_\tau = a_y) \right] \quad (12)$$

$$D_{a_y, s_x | \xi_{0 \rightarrow t}} = \pi(a_y | s_x) D_{s_x | \xi_{0 \rightarrow t}} \quad (13)$$

Our definition of the posterior expected action subspace visitation count is given in Equation 14. This expresses the future expectation the user will perform an action a_y while in a subspace \mathcal{S}_p , given their current trajectory $\xi_{0 \rightarrow t}$.

$$\begin{aligned} D_{a_y, \mathcal{S}_p | \xi_{0 \rightarrow t}} &\triangleq \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{\tau=t+1}^T I(s_\tau \in \mathcal{S}_p) I(a_\tau = a_y) \right] \\ &= \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{s_x \in \mathcal{S}_p} \sum_{\tau=t+1}^T I(s_\tau = s_x) I(a_\tau = a_y) \right] \\ &= \sum_{s_x \in \mathcal{S}_p} \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{\tau=t+1}^T I(s_\tau = s_x) I(a_\tau = a_y) \right] \\ &= \sum_{s_x \in \mathcal{S}_p} D_{a_y, s_x | \xi_{0 \rightarrow t}} \\ &= \sum_{s_x \in \mathcal{S}_p} \pi(a_y | s_x) D_{s_x | \xi_{0 \rightarrow t}} \end{aligned} \quad (14)$$

Thus, the posterior expected action subspace visitation is straightforward to compute with $D_{s_x | \xi_{0 \rightarrow t}}$. Various inference tasks can be constructed by choosing a_y and \mathcal{S}_p appropriately.

D.2. Joint Action-State Subspace Visitation

We additionally derive the expected transition count from a subspace of states to a subspace of actions. This expresses the expectation that the user will perform an $a_y \in \mathcal{A}_y$ from a $s_x \in \mathcal{S}_p$. It is defined as:

$$D_{\mathcal{A}_y, \mathcal{S}_p | \xi_{0 \rightarrow t}} \triangleq \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{\tau=t+1}^T I(s_\tau \in \mathcal{S}_p) I(a_\tau \in \mathcal{A}_y) \right] \quad (16)$$

$$\begin{aligned} &= \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{a_y \in \mathcal{A}_y} \sum_{s_x \in \mathcal{S}_p} \sum_{\tau=t+1}^T I(s_\tau = s_x) I(a_\tau = a_y) \right] \\ &= \sum_{a_y \in \mathcal{A}_y} \sum_{s_x \in \mathcal{S}_p} \mathbb{E}_{P(\xi_{t+1 \rightarrow T} | \xi_{0 \rightarrow t})} \left[\sum_{\tau=t+1}^T I(s_\tau = s_x) I(a_\tau = a_y) \right] \\ &= \sum_{a_y \in \mathcal{A}_y} \sum_{s_x \in \mathcal{S}_p} D_{a_y, s_x | \xi_{0 \rightarrow t}} \\ &= \sum_{a_y \in \mathcal{A}_y} \sum_{s_x \in \mathcal{S}_p} \pi(a_y | s_x) D_{s_x | \xi_{0 \rightarrow t}}. \end{aligned} \quad (17)$$

Again, computing this quantity is straightforward with $D_{s_x | \xi_{0 \rightarrow t}}$. By marginalizing $D_{s_x | \xi_{0 \rightarrow t}}$ over various action and state subspaces that have semantic meaning, different inference quantities can be expressed and computed.

Environment	Object Set
Home 1	{bookbag, book, blanket, coat, laptop, mug, plate, snack, towel}
Home 2	{bookbag, book, blanket, coat, guitar, laptop, mug, plate, remote, snack, towel}
Office 1	{bookbag, textbook, bottle, coat, laptop, mug, paper, plate, snack}
Office 2	{bookbag, textbook, bottle, coat, laptop, mug, paper, plate, snack}
Lab 1	{beaker, coat, plate, pipette, tube}

Table 7: **Objects available in each environment.**

Environment	Scene Type Set
Home 1	{bathroom, bedroom, exit, dining room, kitchen, living room, office}
Home 2	{bathroom, bedroom, exit, dining room, kitchen, living room, office, television stand}
Office 1	{bathroom, exit, kitchen, lounge, office, printer station, water fountain}
Office 2	{bathroom, exit, kitchen, lounge, office, printer room, water fountain}
Lab 1	{cabinet stand, exit, gel room, lab bench 1, lab bench 2, refrigeration room}

Table 8: **Scene types available in each environment.**

Frame Index	6750	6900	7200	7400	7630	7700
Action/Arrival	Release Coat	Acquire Bookbag	Arrive Office	Acquire Mug	Arrive Kitchen	Release Mug

Table 9: **Labels example:** A small snippet of ground truth labels for Home 1.

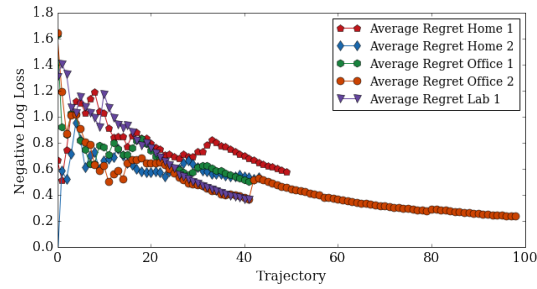


Figure 7: **Noisy Empirical Regret.** DARKO’s online behavior model exhibits sublinear convergence in average regret. Initial noise is overcome after DARKO adjusts to learning about the user’s early behaviors.

E. Additional Dataset Information

Objects: The set of objects available in each environment is shown in Table 7.

Scene types: The set of scene types in each environment is shown in Table 8.

Labels: A small snippet of ground truth for Home 1 is shown in Table 9. The ground truth pairs frame indices (timestamps) with actions and goal arrivals.

F. Regret with Detectors

We additionally show the empirical regret when using our goal discovery and action detection methods in Figure 7. We observe somewhat noisier regret behavior than in the original case, as the underlying demonstrations are noisier. The number of trajectories in Office 2 is higher here due to errors in the goal forecasting method, resulting in more goals being detected, which segments the demonstrations into more trajectories.

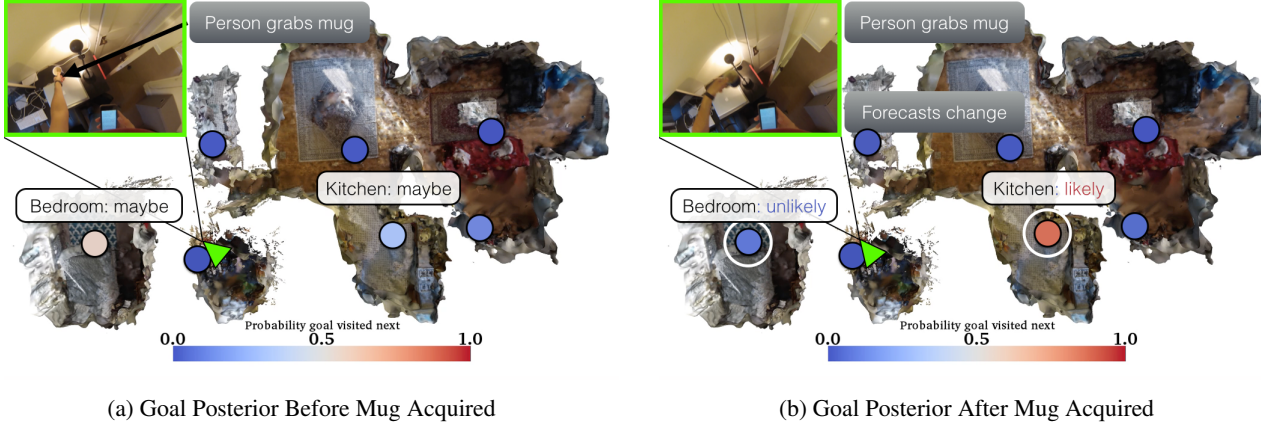


Figure 8: **Goal Posterior Change Visualization:** Goal posteriors for two frames are visualized in the Home 1 environment. The person’s location is in green, images from the camera are inset at top left, and goal posteriors are colored according to the above colormaps. Before grabbing the mug (Figure 8a), DARKO forecasts roughly equivalent probability to bedroom and kitchen. After the user grabs the mug (Figure 8b), DARKO correctly predicts the user is likeliest to go to the kitchen.

G. Visualizations

We provide example 3D visualizations of 1) goal posterior 2) future state visitation and 3) the value function.

G.1. Goal posterior visualization

To emphasize our empirical finding that modeling the interaction with objects is useful for goal posterior prediction, we show an example sequence of predictions in Figure 8. Before the user grabs the mug, our algorithm predicts roughly equivalent probability to both the bedroom and kitchen. We see that after the user grabs the mug, DARKO has high confidence that the user will go to the kitchen.

G.2. Future state visitation visualizations

See Figure 9 for example visualizations of the expected future visitation counts. In order to visualize in 3 dimensions, we take the max visitation count across all states at each position. In rows 1, 3, and 4, a single demonstration is shown, which adapts to the agent’s trajectory (history). In row 2, the future state distribution drastically changes after each time the agent reaches a new goal.

G.3. Value function visualizations

See Figure 10 for example visualizations of the value function over time. Note 1) the state space size changes, and 2) that the value function changes over time, as the component of state that indicates the previous goal affects the value function.

G.4. RNN baseline settings

We experimented with a variety of settings for the RNN baseline. After each goal is detected, the RNN is refit. The settings we experimented with are $\text{cell} \in \{\text{GRU}, \text{Basic}\}$, $\text{learning rate} \in \{0.1, 0.01, 0.001, 0.0001\}$, $\text{hidden dimension} \in \{8, 16, 32, 64\}$, $\text{epochs after each goal} \in \{5, 10, 50, 100\}$.

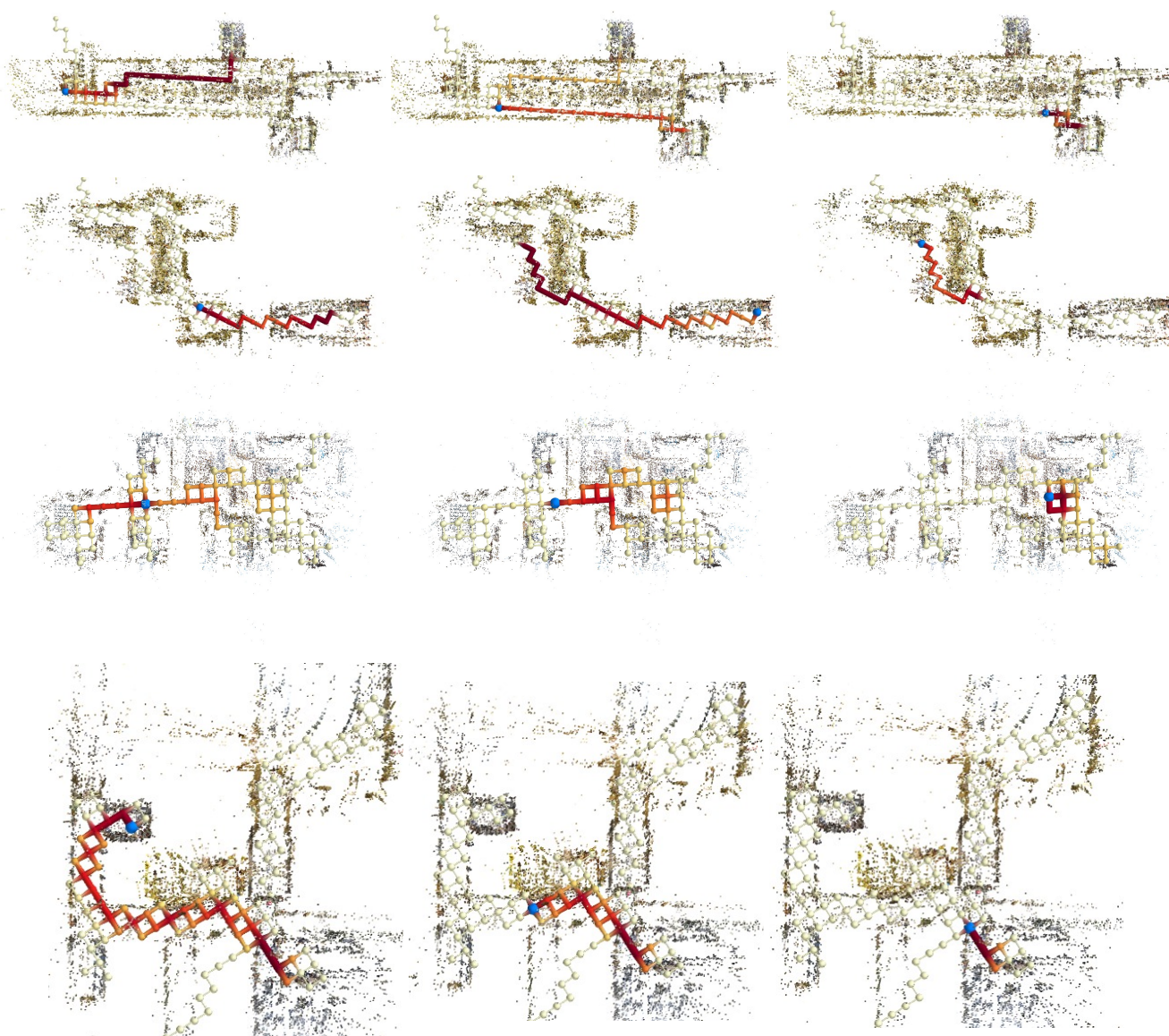


Figure 9: Future state visitation predictions changing as the agent (blue sphere) follows their trajectory. The state visitations are projected to 3D by taking the max over all states at each location. The visualizations are, by row: Office 1, Lab 1, Home 1., Office 2

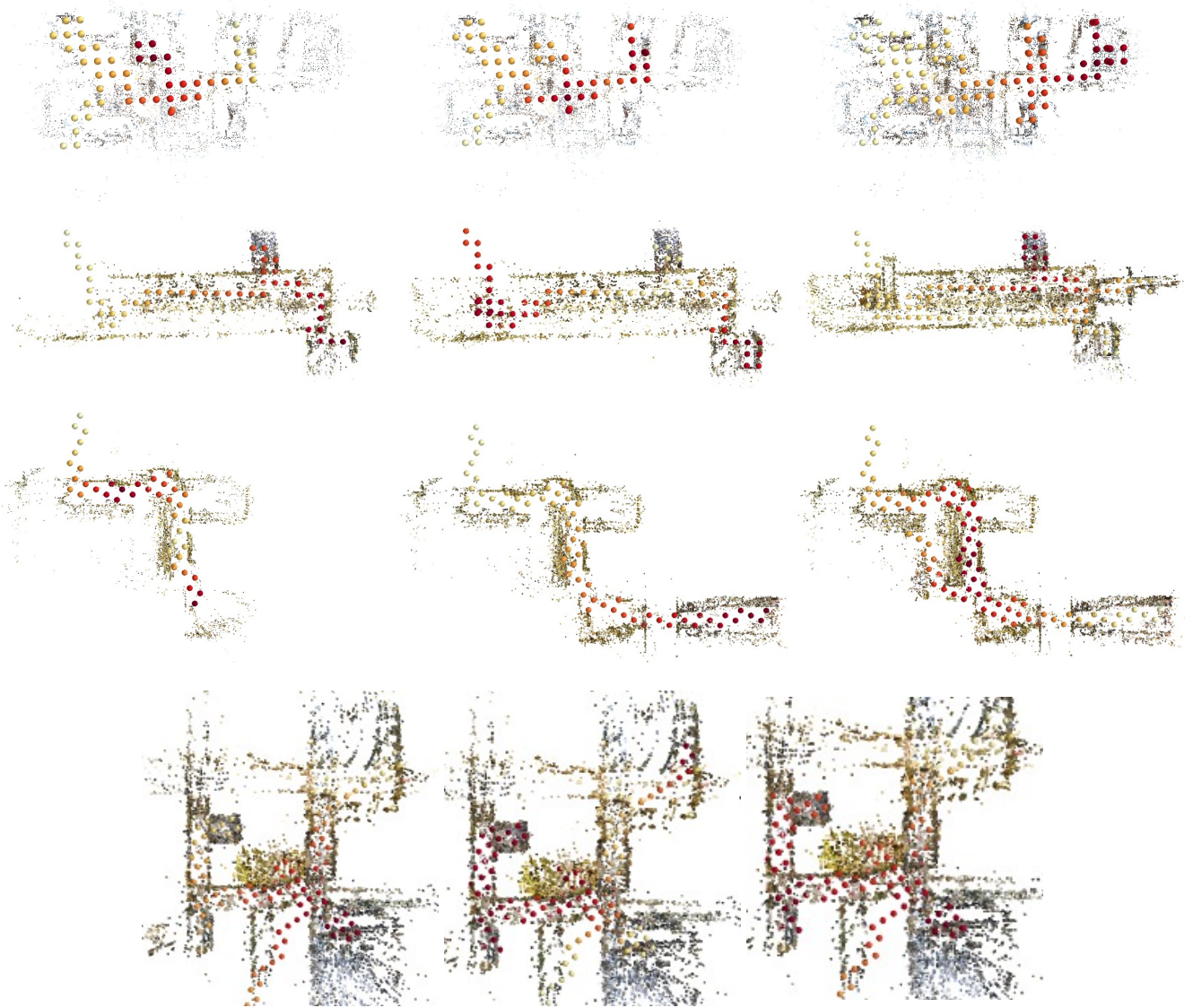


Figure 10: Projections of the value function ($V(s)$) for environments as time elapses (left to right). The state space expands as the user visits more locations. For each position, the maximum value (across all states at that position) is displayed: $\max_{s \in S_x} V(s)$. From top to bottom, the environments are Home 1, Office 1, Lab 1.