LEARNING TASK REPRESENTATIONS FROM SPARSE EVIDENCE IN OFFLINE META-REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Context-based offline meta-reinforcement learning (COMRL) trains a context encoder to infer a latent task from collected trajectories. Existing methods commonly treat task-relevant evidence as broadly distributed across the context, or isolate a few individually informative transitions. This leaves open the case in which only a small fraction of interactions carries task-dependent information, while the rest is valid but weakly informative experience. A trajectory may still contain sufficient task information, but the informative feedback can appear in brief intervals of unknown location and duration. Learning a reliable task representation then requires combining these sparse clues while limiting interference from the rest of the context. We study this sparse-evidence regime and show that task inference requires preserving weak, collectively informative responses before determining their relevance. Based on this observation, we introduce STeR (Slot-Based Task Representation Learning from Sparse Evidence in Offline Meta-Reinforcement Learning), a structured context encoder that performs accumulation before selection. Transition features are softly assigned to content-based latent slots, each slot is normalized independently, and a task-level readout selects the summaries that matter. The organization is learned from trajectory-level task labels alone, with no evidence masks or temporal localization supervision. We evaluate STeR on controlled offline meta-reinforcement learning environments where task information appears through sparse rewards, transient motion responses, and hidden contact dynamics. Across variations in evidence duration and onset, STeR improves task identification and downstream adaptation while preserving performance when task evidence becomes sparse. The results suggest that structured accumulation is a useful inductive bias for task representations drawn from incomplete, noisy offline experience.

1 INTRODUCTION

Adapting to an unfamiliar task requires an agent to learn what matters about its new environment from limited experience. Context-based offline meta-reinforcement learning (COMRL) learns this ability from previously collected data: a context encoder infers a task representation, and a shared policy uses that representation to guide its actions (Li et al., 2021a; Yuan & Lu, 2022; Li et al., 2024). At deployment, the agent receives context from an unseen task and must extract the information needed for effective control. The usefulness of this context depends on how task evidence is distributed along the trajectory.

A simple example shown in Figure 1 illustrates this challenge. Consider manipulation tasks where the agent must infer hidden physical properties, such as drawer damping. Most of the trajectory, including approaching the handle and resetting after interaction, may appear similar across tasks. The critical information may emerge only during a short interaction phase, where the applied action and resulting motion reveal the hidden dynamics. A useful task representation therefore requires aggregating these weak responses from the entire trajectory rather than relying on a few explicitly identifiable transitions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

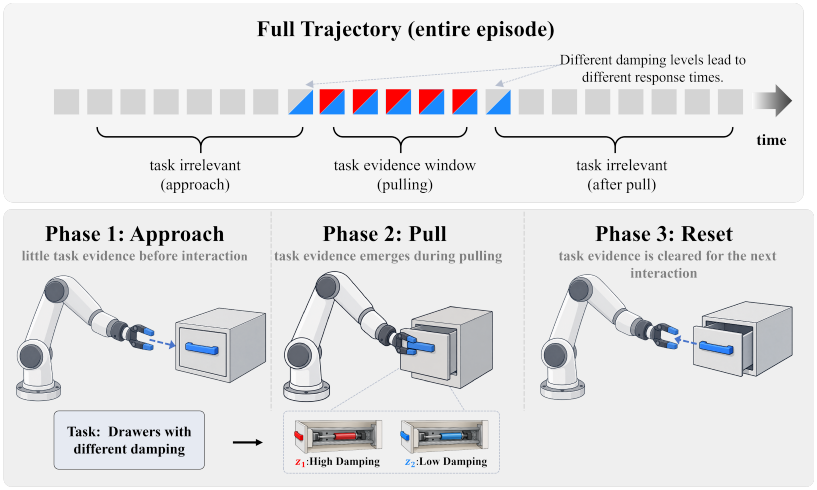

Figure 1: Illustration of sparse task evidence in drawer manipulation. A complete trajectory contains approach, pulling, and reset phases. The task-dependent response emerges during a brief pulling window and varies with the drawer's damping, while the surrounding interactions provide little direct evidence. The highlighted window is conceptual and its boundary is not provided to the learner.

We study this setting as *sparse task evidence*: interactions carrying task information occupy a small fraction of an episode, with a brief informative interval as a concrete instance. We call an interaction evidence-bearing when, under a fixed history and action, its response distribution may depend on task identity. An individual interaction may supply only a weak clue; several responses can jointly support task inference. We focus on tasks whose relevant information is retained in complete transition tuples and their content distribution. STeR treats the ordering of intact transition tuples as nuisance variation, so tasks whose identity depends on order or cross-step relations require an additional order-aware component. Even when the amount and strength of evidence remain fixed, a longer context can introduce substantially more background variation. Prior work has established the importance of data identifiability (Dorfman et al., 2021) and robust task representations under behavior-policy shifts (Yuan & Lu, 2022; Gao et al., 2023; Li et al., 2024). These foundations motivate a further question: how can an agent preserve collectively informative but sparse evidence as the surrounding experience grows?

The central design question is how to accumulate weak evidence without making its contribution depend primarily on how many background interactions accompany it. A global mean weights evidence in proportion to its context share, while flat transition-level attention evaluates rare interactions mainly as separate candidates. Separately normalized summaries offer another accumulation structure: several pulling responses can form a joint description of damping before a readout assesses task relevance. This motivates accumulation before selection: gather related experience into intermediate slots, then determine how much the accumulated content contributes to task inference.

We introduce STeR, a slot-based task representation method that implements this principle. Content-based soft assignments gather related transitions into multiple slots, and each slot normalizes by its own assignment mass before trajectory-level compression. Within-slot means and variances describe the accumulated responses, while a learned readout assesses the slot representations to form a task context for control. The pipeline is trained with trajectory-level task supervision and no evidence labels.

To investigate evidence aggregation across different sources of task information, we construct three primary COMRL benchmarks: AntDirSparseEvidence, HalfCheetahSparseEvidence, and DrawerDynamicEvidence. They move from sparse task feedback, through short locomotion responses, to hidden contact dynamics, asking whether the same representation principle remains effective across these sources of evidence. Tasks remain fixed within each episode, and collectors are independent of task identity. Controlled comparisons on the three primary benchmarks preserve task identifiability and hold the amount and strength of evidence fixed while varying the surrounding context. Separate

evaluations vary the onset and duration of the informative interaction to examine generalization when the available records change.

Across the three primary benchmarks, STeR improves task inference and downstream control relative to the evaluated COMRL methods. Comparisons with direct transition attention and structural ablations test the value of assessing relevance after accumulation, while controlled interventions vary evidence duration and evidence location.

Our contributions are threefold:

- We isolate, within COMRL, the setting in which direct task-conditioned responses occupy a small fraction of the context while task identifiability is preserved and the surrounding context can be varied under control.
- We develop STeR, a structured context encoder based on accumulation before selection, which preserves distributed weak responses through slot-level summaries before task relevance estimation.
- We provide evaluations across reward-based, motion-based, and hidden-dynamics benchmarks, demonstrating how task representations behave under varying evidence duration and location.

2 Related Work

Context-Based Offline Meta-RL. COMRL learns task representations from offline context and conditions a shared policy on these representations to adapt to unseen tasks. Existing approaches improve representation learning through metric learning, attention, contrastive objectives, information-theoretic formulations, and representation–policy dynamics (Li et al., 2021b;a; Yuan & Lu, 2022; Gao et al., 2023; Li et al., 2024; Zhang et al., 2025).

Related offline meta-RL formulations also learn task-conditioned value or policy functions, use advantage-weighted adaptation, or treat trajectories as prompts for sequence models (Fakoor et al., 2020; Li et al., 2020; Mitchell et al., 2021; Xu et al., 2022; Wang et al., 2024).

Earlier latent-context methods such as PEARL and variBAD infer task variables for adaptive behavior, while ContraBAR applies contrastive predictive coding in online and offline settings (Rakelly et al., 2019; Zintgraf et al., 2020; Choshen & Tamar, 2023). BOReL characterizes MDP ambiguity, where data from different tasks can be explained by a common MDP under different behavior policies (Dorfman et al., 2021). These works primarily focus on extracting task information from context or on when that information is identifiable. STeR studies the complementary setting in which tasks remain identifiable but direct task-conditioned responses occupy only a small fraction of the context.

Slot-Based Representation Learning. Slot-based representation learning provides a mechanism for decomposing inputs into multiple latent summaries. Slot Attention uses iterative competition to assign input features according to their content (Locatello et al., 2020). Originally developed for object-centric learning, it provides several summaries of a shared input; STeR uses them as adaptive aggregation units for trajectory evidence.

Permutation-invariant set encoders, including Deep Sets and Set Transformer, provide alternatives for pooling unordered interactions (Zaheer et al., 2017; Lee et al., 2019). STeR instead uses multiple soft slots to accumulate related responses before task readout.

SloTTAr extends slot-based representations to reinforcement-learning trajectories by learning temporal abstractions and reconstructing sequential structure (Gopalakrishnan et al., 2023). CompILE and OMPN likewise infer behavior units and temporal boundaries from demonstrations (Kipf et al., 2019; Lu et al., 2021). STeR uses soft slots as task-summary units: membership follows transition content, and a task-supervised readout combines the summaries for task inference and policy conditioning without making temporal segment boundaries the learning target.

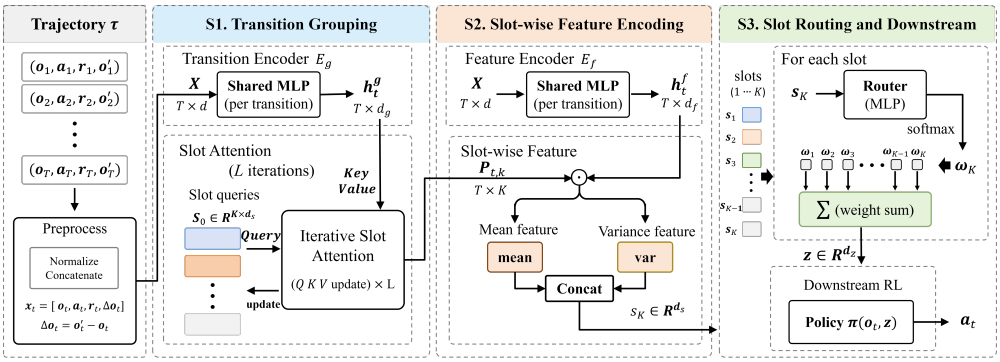

Figure 2: Overview of STeR. Transition features undergo content-based slot assignment, are encoded as slot-wise mean–variance features s_k, and are combined by router weights ω_k into the task representation z used for downstream control.

3 METHOD

STeR is based on an observation: when evidence is sparse, selection before accumulation is unreliable. Therefore, STeR first accumulates related responses into latent summaries, and only then estimates their task relevance. Figure 2 summarizes the pipeline.

3.1 SPARSE EVIDENCE PROBLEM FORMULATION

We consider offline meta-RL problems where a context trajectory contains both task-dependent and task-independent interactions. The key challenge is that task-dependent interactions are sparse, their locations are unknown, and no evidence annotation is available. STeR preserves recurring task-related responses before determining which responses should influence task inference.

Let $C \in \mathcal{C}$ denote the task identity, and let $\tau = (\zeta_t)_{t\in\mathcal{V}}$ denote a fixed-task context trajectory, where $\zeta_t = (o_t, a_t, r_t, o'_t)$ is a recorded transition. During meta-training, contexts carry task identities; at evaluation, a context from an unseen task is encoded for a separate control episode. We consider the order-free, content-based setting in which the encoder receives the multiset of transition tuples. Tasks whose identity depends on order or cross-step relations require an additional order-aware component.

To formalize sparse evidence, introduce a latent indicator $m_t \in \{0, 1\}$. Given the preceding history and the current action, $m_t = 1$ means that the conditional law of the observed response (r_t, o'_t) may depend on the task, while $m_t = 0$ means that it has no direct task dependence under the same conditioning. The indicator marks where task-conditioned evidence can occur, not whether one transition identifies the task. Define

$$\mathcal{E}(\tau) = \{t \in \mathcal{V} : m_t = 1\}, \qquad \rho(\tau) = \frac{|\mathcal{E}(\tau)|}{|\mathcal{V}|}. \tag{1}$$

We study $0 < \rho(\tau) \ll 1$. The evidence set may be an unknown short interval or a dispersed subset, and neither its indicators nor its boundaries are given to the learner; interactions outside it may still retain effects through the state history.

Context identifiability. At the context level, tasks are identifiable when their complete-context distributions differ. Let $P_{e,c}$ and P_b denote the evidence and background laws for task c. Under a common positive ρ, a shared background law, and distinct evidence laws, the sampled-interaction law is $P_c = \rho P_{e,c} + (1-\rho)P_b$ and satisfies $\mathrm{TV}(P_c, P_{c'}) = \rho\,\mathrm{TV}(P_{e,c}, P_{e,c'}) > 0$. Thus population-level separation can remain for any $\rho > 0$; the assumptions, proof, and global-pooling calculation are given in Appendix C.1.

3.2 EVIDENCE ACCUMULATION THROUGH CONTENT-BASED SLOTS

Because task distinctions can appear in reward or physical response amid otherwise similar interactions, we represent each valid interaction by its observation, action, reward, and observation change,

normalizing each channel before concatenation:

$$\Delta o_t = o'_t - o_t, \qquad x_t = [\mathcal{N}_o(o_t); \mathcal{N}_a(a_t); \mathcal{N}_r(r_t); \mathcal{N}_{\Delta o}(\Delta o_t)]\,, \tag{2}$$

where each $\mathcal{N}$ uses statistics estimated from training data, and $x_t \in \mathbb{R}^d$. For scalar rewards, $d = 2d_o + d_a + 1$, where d_o and d_a are the observation and action dimensions. The encoder receives the multiset $X(\tau) = \{\{x_t : t \in \mathcal{V}\}\}$. Appendix A specifies normalization and data handling.

To accumulate potentially weak task-conditioned responses whose positions vary across contexts, a shared multilayer perceptron $E_g : \mathbb{R}^d \to \mathbb{R}^{d_g}$ embeds each transition as $h_t^g = E_g(x_t)$. A Slot Attention module (Locatello et al., 2020) assigns the content to K foreground slots and one background slot. At iteration ℓ, learned projections produce keys κ_t, values v_t, and slot queries $q_k^{(\ell)}$; slot-wise Softmax assignments are then normalized over transitions to form each slot update:

$$A_{t:}^{(\ell)} = \mathrm{Softmax}_{j=0:K}\left(\frac{\kappa_t^\top q_j^{(\ell)}}{\sqrt{d_g}}\right), \qquad P_{tk}^{(\ell)} = \frac{A_{tk}^{(\ell)}}{\sum_{t' \in \mathcal{V}} A_{t'k}^{(\ell)}}, \qquad \bar{v}_k^{(\ell)} = \sum_{t \in \mathcal{V}} P_{tk}^{(\ell)} v_t. \tag{3}$$

The first normalization distributes each transition across competing slots, while the second gives each slot its own pooling measure. A gated recurrent unit followed by a residual multilayer perceptron updates each slot from $\bar{v}_k^{(\ell)}$; after the final update, we recompute the assignments and denote them by A and P.

Each slot therefore forms an intermediate summary before the summaries are combined. The conditional off-mask bound for this summary and its sparsity requirement are given in Appendix C.2; task-level supervision shapes the content-based assignments across contexts.

Because assignments are based on content rather than time indices, related interactions from different positions can enter the same slot; additional assignment properties are given in Appendix A.6.

3.3 Statistical Slot Representation

Within-slot encoding. Each slot may pool several weak responses, so the feature encoder retains their aggregate. The grouping encoder E_g determines membership, while a separate encoder $E_f : \mathbb{R}^d \to \mathbb{R}^{d_f}$ maps responses to a task-inference space and produces $h_t^f = E_f(x_t)$; the final slot weights determine the feature mean and coordinate-wise variance:

$$\mu_k^f = \sum_{t \in \mathcal{V}} P_{tk} h_t^f, \qquad \nu_k^f = \left[\sum_{t \in \mathcal{V}} P_{tk}(h_t^f \odot h_t^f) - \mu_k^f \odot \mu_k^f\right]_+ . \tag{4}$$

Here $\odot$ denotes element-wise multiplication and $[\cdot]_+$ removes negative numerical residuals. The mean captures recurring responses, while the variance retains within-slot variation.

A slot representation concatenates these statistics and maps them to the slot-feature space as $s_k = f_{\mathrm{proj}}\Big(\mathrm{Concat}(\mu_k^f, \nu_k^f)\Big) \in \mathbb{R}^{d_s}$. The same statistics are temporarily formed from h_t^g in the initial slot-formation stage, denoted μ_k^g and ν_k^g.

3.4 Task Relevance Estimation

Slot routing. Because evidence locations are unknown, task relevance is assessed after accumulation. A shared routing multilayer perceptron g scores the slot summaries s_k, and a softmax turns these scores into relative weights:

$$\omega_k = \frac{\exp(g(s_k)/\tau_{\mathrm{route}})}{\sum_{j=1}^K \exp(g(s_j)/\tau_{\mathrm{route}})}, \qquad z = \mathrm{nrm}\left(\sum_{k=1}^K \omega_k s_k\right). \tag{5}$$

Here nrm denotes stabilized ℓ_2 normalization and $z \in \mathbb{R}^{d_z}$. The complete context encoder is written $z = f_\theta(\tau)$; the temperature $\tau_{\mathrm{route}} > 0$ controls the routing sharpness, while the background slot participates in assignment and reconstruction but not in the direct foreground readout.

Task-level loss. We train the complete slot pipeline with a symmetric task-level loss $\mathcal{L}_{\text{task}}$, pulling representations of independent contexts from the same task together and separating contexts from other tasks. It is the average of the two support–query prototype directions defined in Appendix A.3. This trajectory-level supervision shapes slot summaries and routing without evidence masks.

Slot representation learning. Training has two representation stages. We first learn an initial slot assignment from complete trajectories, temporarily using h_t^g for task readout. The slot-formation objective is

$$\mathcal{L}_{\text{slot}} = \mathcal{L}_{\text{task}} + \lambda_{\text{tr}}\mathcal{L}_{\text{tr}} + \lambda_r\mathcal{L}_r + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \lambda_{\text{occ}}\mathcal{L}_{\text{occ}}. \tag{6}$$

Here $\mathcal{L}_{\text{tr}}$ and $\mathcal{L}_r$ reconstruct transition channels and reward, while $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{occ}}$ regularize assignment concentration and slot occupancy.

We then initialize E_f from E_g and refine the feature encoder, projection, and router with $\mathcal{L}_{\text{task}}$. The base schedule holds assignment parameters fixed during this refinement. Definitions of the losses, update schedule, and context subsampling are in Appendix A.

3.5 Offline Adaptation

Offline control and task adaptation. During joint training, representation-learning and control-learning blocks alternate slowly, holding the other component fixed in each block. After representation training, the context trajectory τ_{ctx} supplied before a separate control episode is encoded once as $z = f_\theta(\tau_{\text{ctx}})$, and the shared offline policy and value function are conditioned on (o_t, z) throughout that episode.

4 Experiments

Experimental goals. Our experiments are designed to answer four questions: (1) Can STeR infer task identity when task information is concentrated in a small portion of the interactions? (2) How does accumulation before selection behave when evidence becomes shorter or shifts in time? (3) Can the learned representation transfer to downstream offline control? (4) Which components of STeR contribute to sparse-evidence retention? We address these questions with controlled environments in which the amount and location of task evidence can be varied independently.

4.1 Experimental Setup

We construct three controlled benchmarks that expose task information through different physical channels. AntDirSparseEvidence (ADSE) represents tasks where evidence appears through sparse reward-dependent motion responses; HalfCheetahSparseEvidence (HCSE) tests sparse behavioral responses under continuous locomotion; and DrawerDynamicEvidence (DDE) provides a more challenging setting where task identity is encoded in hidden contact dynamics rather than an explicit reward signal. Together, these environments distinguish failures caused by insufficient evidence from failures caused by the type of task signal.

Sparse-evidence evaluation. Each primary benchmark uses six conditions: `normal`, `onset`, `short`, `long`, `onset-short`, and `onset-long`. They vary evidence onset, duration, or both, while keeping the task fixed; thus they separately probe when task information appears and how much of it is available.

Baseline methods. We compare STeR with representative offline meta-reinforcement learning approaches covering different strategies for task representation learning. Metric and contrastive methods (FOCAL++ and ContraBAR) learn discriminative task embeddings (Li et al., 2021a; Choshen & Tamar, 2023); UNICORN optimizes information-theoretic objectives for context construction (Li et al., 2024); BOReL studies task identifiability under offline ambiguity (Dorfman et al., 2021); and CTMRL explores cross-task context construction for improved generalization (He et al., 2026). Task-oracle and task-blind references mark the ceiling and no-task-information floor. Representation tables use the same cosine 1-nearest-neighbor support/query readout for every method. All downstream comparisons use the same frozen-representation TD3+BC controller and evaluation budget, with

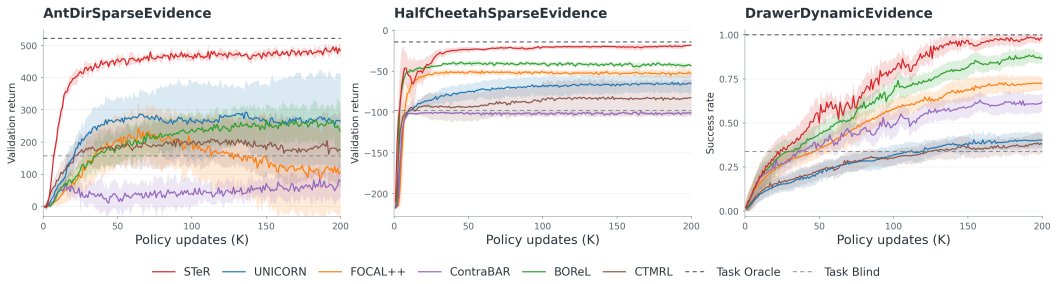

Figure 3: Downstream-control learning curves for AntDirSparseEvidence, HalfCheetahSparseEvidence, and DrawerDynamicEvidence. The ADSE and HCSE panels report full-horizon dense task-control return, while the DDE panel reports task-success rate; shaded regions show variability across the five independent seeds and dashed lines denote task-oracle and task-blind references.

Benchmark / split	STeR	UNICORN	FOCAL++	ContraBAR	BOReL	CTMRL
ADSE / IID	$\mathbf{482.73 \pm 5.26}$	265.29 ± 78.01	219.41 ± 84.32	43.27 ± 15.28	259.77 ± 26.19	191.83 ± 68.09
ADSE / Unseen	$\mathbf{420.13 \pm 26.63}$	253.02 ± 70.21	208.00 ± 69.54	48.20 ± 12.65	260.51 ± 28.26	185.94 ± 52.52
ADSE / OOD	$\mathbf{379.01 \pm 18.58}$	-17.58 ± 204.76	20.48 ± 24.82	-133.34 ± 39.15	47.49 ± 58.15	-229.00 ± 158.19
HCSE / IID	$\mathbf{-18.97 \pm 2.00}$	-76.95 ± 10.43	-49.67 ± 6.72	-102.84 ± 1.81	-40.60 ± 1.88	-88.53 ± 21.47
HCSE / Unseen	$\mathbf{-20.54 \pm 2.57}$	-61.21 ± 7.26	-49.25 ± 4.55	-95.16 ± 4.06	-38.42 ± 2.49	-83.15 ± 17.22
HCSE / OOD	$\mathbf{-128.28 \pm 14.76}$	-208.79 ± 15.90	-218.05 ± 26.92	-233.86 ± 7.16	-169.62 ± 13.39	-215.58 ± 29.12
DDE / IID	$\mathbf{98.08 \pm 0.46}$	41.87 ± 1.85	67.40 ± 1.91	43.03 ± 1.91	82.57 ± 2.41	34.69 ± 1.62
DDE / Unseen	$\mathbf{99.79 \pm 0.29}$	48.93 ± 1.94	70.39 ± 1.83	50.11 ± 2.99	86.56 ± 2.48	40.74 ± 1.88
DDE / OOD	$\mathbf{90.31 \pm 1.61}$	44.22 ± 1.43	66.50 ± 2.29	48.06 ± 2.81	79.69 ± 2.08	39.18 ± 2.30

Table 1: Normal-condition downstream-control results for STeR and five external baselines. This table is the authoritative normal-condition summary; Appendix Tables 16–18 expand the same protocol to all 18 task-condition combinations. ADSE and HCSE report full-horizon dense task-control return, whereas DDE reports task-success rate (%). The highest value in each row is bold. Entries follow the common five-seed protocol described in Appendix D.

the checkpoint, seed, metric, and aggregation protocol in Appendix D. Further task details are also provided there.

Throughout the experiments, IID denotes in-distribution tasks, Unseen denotes held-out tasks within the training task range, and OOD denotes out-of-distribution tasks outside that range.

4.2 Main Results: Task Identification and Adaptation

We first evaluate whether the learned representation preserves task information before considering downstream control. Since sparse evidence creates a representation challenge rather than merely a policy optimization challenge, task identification provides a direct measurement of evidence retention. Appendix Table 6 gives the task-identification reference under the standard `normal` evidence condition. The main text focuses on the corresponding downstream results and the controlled evidence interventions.

We first report the standard-condition control comparison. Figure 3 shows the learning curves, and Table 1 gives the corresponding final results for STeR and the five external baselines. ADSE and HCSE columns report full-horizon dense task-control returns, while the DDE columns report task-success rates. Columns are grouped by benchmark and then by task split; the two metrics should therefore be compared only within their respective benchmark.

The complete evidence-intervention control grids are reported with the sparse-evidence analyses in Section 4.3 and Appendix E; DDE uses the same downstream protocol and reports task-success rate throughout.

Across the three benchmarks, STeR preserves task information under the standard evidence condition in the common representation readout. The results are consistent with the intended role of slot-wise evidence organization when task information is distributed across a trajectory.

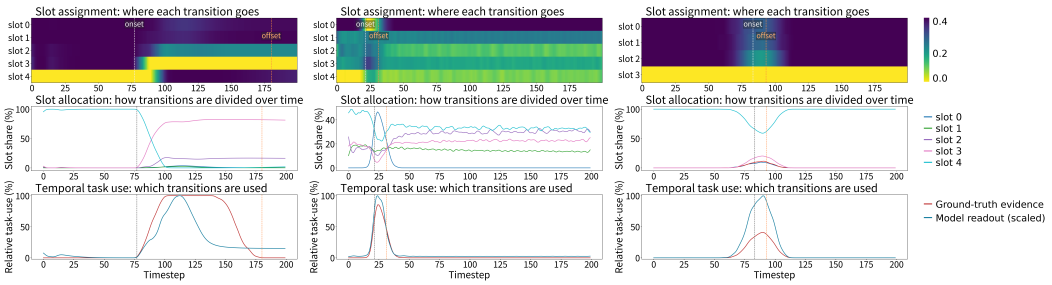

Figure 4: Representative slot-time diagnostics for the three sparse-evidence analyses. All three panels use task OOD contexts; from left to right, they show delayed onset in DDE contact dynamics, short motion-response duration in HCSE, and the combined onset–duration intervention in ADSE directional reward.

4.3 Sparse-Evidence Analysis

The previous experiments evaluate representation quality under the standard condition. We next test whether a representation preserves task information when evidence becomes temporally limited or its onset shifts. We vary evidence duration and onset independently; the standard `normal` condition is the reference, and complete numerical results are provided in Appendix Tables 9–15.

For downstream consequences of the same interventions, Appendix Table 5 summarizes STeR across all three benchmarks, task splits, and six evidence conditions. The complete method-by-method grids and provenance notes are in Appendix E, Tables 16–18. All entries use the common downstream controller and evaluation protocol.

The frozen attribution audit provides a direct check of this interpretation: removing evidence lowers HCSE task-identification accuracy to 9.18%, 20.40%, and 18.82% for IID, Unseen, and OOD tasks, while the ADSE routing assigns 99.97% of mass to the benchmark-defined evidence mask (Appendix F).

The slot-time diagnostics in Figure 4 accompany the three analyses below. Each panel shows content-based slot assignment, transition allocation, and a rescaled readout compared with the benchmark-defined evidence interval.

Evidence onset. This analysis asks whether a task representation can retain evidence when the informative response is delayed. The left panel of Figure 4 shows the DDE diagnostic. The comparisons between `normal` and `onset`, and between the matched-duration pairs `short`/`onset-short` and `long`/`onset-long`, quantify the difference between the corresponding evidence settings. ADSE and HCSE share the physical trajectory, actions, and dense control reward across paired views; in DDE, the corresponding settings alter the pull-phase window in the task construction. The comparison tests retention of a delayed response amid task-irrelevant interactions.

Evidence duration. This analysis asks how task inference changes with the amount of available evidence. The middle panel of Figure 4 shows the HCSE short-evidence diagnostic. At the standard onset, `short`, `normal`, and `long` vary the duration and coverage of task-conditioned interactions without changing the task. The comparison tests whether consistent response features remain represented when most of the context contributes background motion.

Combined onset and duration. The `onset-short` and `onset-long` conditions test the interaction between evidence location and amount. The right panel of Figure 4 shows the ADSE diagnostic. Comparing `short` with `onset-short`, and `long` with `onset-long`, shows how the paired settings differ when the same nominal duration is delayed. These conditions test whether the slot representation retains a weak response when both its amount and its position are unfavorable.

Forward attribution gives a direct check: removing the evidence-bearing interactions lowers HCSE task-identification accuracy to 9.18%, 20.40%, and 18.82% on IID, Unseen, and OOD contexts, while the ADSE readout routes 99.97% of its mass to the evidence region (Appendix F).

Results by question. Delaying the evidence changes ADSE and HCSE little on IID and Unseen tasks, but more on OOD tasks and in DDE. Short evidence produces the largest drop for ADSE and HCSE, while long evidence recovers much of the task-identification accuracy; DDE varies more under hidden contact dynamics. The delayed-short condition is generally the most demanding, whereas delayed-long often recovers the long-evidence performance in ADSE and HCSE.

4.4 Ablation Studies

We conduct four ablation families on ADSE with the same input interface, representation budget, and five seeds, changing one design axis at a time. Figure 5 reports OOD task-identification accuracy across all six evidence conditions; the support-prototype protocol and IID/Unseen evaluations are in Appendix D.5.

Structural ablation. Structural ablation compares the full design with Uniform slots, Pure attention, Mean pool, and GRU. Uniform slots keeps the slot summaries but uses uniform routing; Pure attention reads transitions directly, Mean pool uses a global mean, and GRU replaces slot aggregation with recurrence. The full design retains the clearest advantage under short evidence.

Loss-design ablation. The loss study changes only the first 60k-update slot-formation stage; the readout-refinement structure, schedule, and budget are unchanged. The upper-right panel shows that the complete objective is most stable, while removing reconstruction or assignment regularization increases the drop under short and delayed-short evidence.

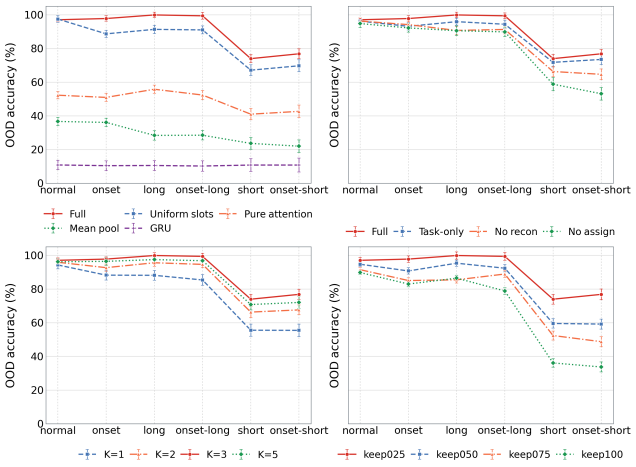

Figure 5: ADSE ablations under six evidence conditions. The four panels vary structural organization, initial-stage losses, the number of foreground slots, and the minimum retained fraction. Curves report OOD task-identification accuracy (%) with error bars over five independent seeds.

Number of slots. We vary the number of foreground slots $K \in \{1, 2, 3, 5\}$ with the background slot, router, initial-stage objective, and budgets fixed. The lower-left panel shows that one foreground slot loses the most accuracy under short evidence, whereas three slots provide the strongest overall curve and five remain competitive.

Trajectory retention ablation. During representation training, trajectories are randomly subsampled without task, reward, evidence, or position labels; validation and test use complete trajectories. The lower-right panel shows that the smallest tested minimum retention fraction (0.25) is most robust under sparse conditions, while retaining the full trajectory during training gives the largest degradation on short and delayed-short evidence.

5 Conclusion

This work studies sparse task evidence in offline meta-reinforcement learning, where task information occupies a small, unknown part of the context. We introduce STeR, which follows an *accumulation-before-selection* principle: related responses are organized into structured summaries before task relevance is estimated. Across the three controlled benchmarks and the evaluated evidence conditions, this design improves task identification and downstream adaptation under the tested shortening and delay interventions. The formulation does not model order-dependent task structure and was evaluated with fixed slot and training configurations.

LLM USAGE STATEMENT

We used large language models solely as general-purpose language-editing assistants to improve the grammar, clarity, and phrasing of the manuscript. They were not used for research ideation, problem formulation, method design, mathematical derivations, experiment design or implementation, data analysis, result generation, or scientific decision-making. The authors independently produced and verified the technical content, experimental results, and references, and take full responsibility for the submitted manuscript.

REFERENCES

Era Choshen and Aviv Tamar. ContraBAR: Contrastive Bayes-adaptive deep RL. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6005–6027, 2023. URL https://proceedings.mlr.press/v202/choshen23a.html.

Ron Dorfman, Idan Shenfeld, and Aviv Tamar. Offline meta reinforcement learning: Identifiability challenges and effective data collection strategies. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4607–4618, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/248024541dbda1d3fd75fe49d1a4df4d-Abstract.html.

Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J. Smola. Meta-q-learning. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1910.00125.

Yunkai Gao, Rui Zhang, Jiaming Guo, Fan Wu, Qi Yi, Shaohui Peng, Siming Lan, Ruizhi Chen, Zidong Du, Xing Hu, Qi Guo, Ling Li, and Yunjie Chen. Context shift reduction for offline meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/fd489a44f3bcb9f122e4931ef21d0c43-Abstract-Conference.html.

Anand Gopalakrishnan, Kazuki Irie, Jürgen Schmidhuber, and Sjoerd van Steenkiste. Unsupervised learning of temporal abstractions with slot-based transformers. *Neural Computation*, 35(4): 593–626, 2023. doi: 10.1162/neco_a_01567. URL https://doi.org/10.1162/neco_a_01567.

Hongcai He, Zetao Zheng, Anjie Zhu, Deqiang Ouyang, and Jie Shao. Improving generalization in offline meta-reinforcement learning via cross-task contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21637–21644, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/39313.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. CompILE: Compositional imitation learning and execution. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3418–3428, 2019. URL https://proceedings.mlr.press/v97/kipf19a.html.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Jiachen Li, Quan Vuong, Shuang Liu, Minghua Liu, Kamil Ciosek, Henrik Christensen, and Hao Su. Multi-task batch reinforcement learning with metric learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Lanqing Li, Yuanhao Huang, Mingzhe Chen, Siteng Luo, Dijun Luo, and Junzhou Huang. Provably improved context-based offline meta-RL with attention and contrastive learning. *arXiv preprint arXiv:2102.10774*, 2021a. URL https://arxiv.org/abs/2102.10774.

Lanqing Li, Rui Yang, and Dijun Luo. FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=8cpHIfgY4Dj.

Lanqing Li, Hai Zhang, Xinyu Zhang, Shatong Zhu, Yang Yu, Junqiao Zhao, and Pheng-Ann Heng. Towards an information theoretic framework of context-based offline meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8a30aba6514b56d02976f49797f6338a-Abstract-Conference.html.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8511df98c02ab60aea1b2356c013bc0f-Abstract.html.

Yuchen Lu, Yikang Shen, Siyuan Zhou, Aaron Courville, Joshua B. Tenenbaum, and Chuang Gan. Learning task decomposition with ordered memory policy network. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=vcopnwZ7bC.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7780–7791, 2021. URL https://proceedings.mlr.press/v139/mitchell21a.html.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340, 2019. URL https://proceedings.mlr.press/v97/rakelly19a.html.

Zhi Wang, Li Zhang, Wenhao Wu, Yuanheng Zhu, Dongbin Zhao, and Chunlin Chen. Meta-dt: Offline meta-rl as conditional sequence modeling with world model disentanglement. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/4f3820576130a8f796ddbf204c841487-Abstract-Conference.html.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua B. Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24631–24645, 2022. URL https://proceedings.mlr.press/v162/xu22g.html.

Haoqi Yuan and Zongqing Lu. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25747–25759, 2022. URL https://proceedings.mlr.press/v162/yuan22a.html.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.06114.

Hai Zhang, Boyuan Zheng, Tianying Ji, Jinhang Liu, Anqi Guo, Junqiao Zhao, and Lanqing Li. Scrutinize what we ignore: Reining in task representation shift of context-based offline meta reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Cr1XlGBGVm.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for bayes-adaptive deep RL via meta-learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklUCCVKPH.

A Additional Method Details

A.1 Input normalization and supervision

For scalar rewards, the transition feature in Eq. (2) has dimension $d = 2d_o + d_a + 1$. Each input channel uses coordinate-wise training means and standard deviations:

$$\mathcal{N}_b(v)_j = \frac{v_j - \mu_{b,j}^{\text{train}}}{\max(\sigma_{b,j}^{\text{train}}, \epsilon)}, \qquad b \in \{o, a, r, \Delta o\}. \tag{7}$$

The statistics are estimated from valid training transitions and reused at validation and test time. Observation differences have their own statistics, and rewards share normalization across training trajectories. This preserves reward differences that trajectory-wise centering could remove. The reference numerical floor is $\epsilon = 10^{-6}$.

The encoder receives the full public observation allowed by the benchmark protocol. Task identities organize trajectory pairs and provide classification targets. Task parameters, evidence masks, boundary labels, environment-specific event thresholds, and temporal condition labels are excluded from the encoder and its supervision. Padding is excluded from attention, pooling, and losses. No time index or positional embedding is added. Network parameters and normalization statistics are learned separately for each environment, using the same architecture and objectives with dimensions determined by the public interfaces.

A.2 Auxiliary slot objectives

Input reconstruction. All foreground slots and the background slot contribute to input reconstruction:

$$b_k = \sum_{t \in \mathcal{V}} P_{tk} h_t^g, \qquad d_k = D_{\text{rec}}(b_k), \qquad \hat{x}_t = \sum_{k=0}^{K} A_{tk} d_k. \tag{8}$$

Here D_{rec} is a learned decoder. Let x_t^{tr} concatenate the normalized observation, action, and observation-change channels, with $d_{\text{tr}} = 2d_o + d_a$, and let x_t^r denote the normalized reward. The losses average over valid transitions in the batch:

$$\mathcal{L}_{\text{tr}} = \mathbb{E}_{\text{valid } t}\left[\frac{\|\hat{x}_t^{\text{tr}} - x_t^{\text{tr}}\|_2^2}{d_{\text{tr}}}\right], \qquad \mathcal{L}_r = \mathbb{E}_{\text{valid } t}\left[(\hat{x}_t^r - x_t^r)^2\right]. \tag{9}$$

Keeping reward error separate allows its weight to be set independently of the number of observation channels. The target transition is part of the encoder's input, so these losses measure reconstruction of supplied data.

Assignment regularization. Minimizing a small assignment-entropy term encourages more definite allocations:

$$\mathcal{L}_{\text{ent}} = \mathbb{E}_{\text{valid } t}\left[-\sum_{k=0}^{K} A_{tk} \log A_{tk}\right]. \tag{10}$$

To discourage complete concentration in one slot, define the per-trajectory assignment mass and effective occupancy as

$$\rho_k = \frac{1}{n}\sum_{t \in \mathcal{V}} A_{tk}, \qquad N_{\text{eff}} = \exp\left(-\sum_{k=0}^{K} \rho_k \log \rho_k\right). \tag{11}$$

The corresponding penalty is

$$\mathcal{L}_{\text{occ}} = \mathbb{E}_\tau[(n_{\min} - N_{\text{eff}})_+]. \tag{12}$$

The reference threshold $n_{\min} = 2$ includes background. Consequently, a foreground slot and background can satisfy the constraint. It encourages diversity in allocation without requiring equal slot sizes or assigning semantic roles to foreground slots.

Component	Initial slot assignment	Readout refinement
Slot-assignment encoder and slot module	Updated	Fixed
Slot-readout encoder	Inactive	Updated
Variance map, projection, scoring network	Updated	Updated
Reconstruction decoder	Updated	Inactive
Context input	Complete	Complete or subsampled

Table 2: The base representation-training schedule (not an evaluation-result table). The slot-readout encoder E_f is initialized from the slot-assignment encoder at the phase boundary.

A.3 Task-level supervision

For each task c in a training batch $\mathcal{C}$, independent trajectories are split into disjoint support and query sets $\mathcal{S}_c$ and $\mathcal{Q}_c$. The normalized support prototype is

$$p_c^{\mathcal{S}} = \mathrm{nrm}\left(\frac{1}{|\mathcal{S}_c|}\sum_{\tau\in\mathcal{S}_c} f_\theta(\tau)\right). \tag{13}$$

Writing $N_{\mathcal{Q}} = \sum_c |\mathcal{Q}_c|$, the query-to-support loss is

$$\mathcal{L}_{\mathcal{Q}|\mathcal{S}} = -\frac{1}{N_{\mathcal{Q}}}\sum_{c\in\mathcal{C}}\sum_{\tau\in\mathcal{Q}_c}\log\frac{\exp\big(f_\theta(\tau)^\top p_c^{\mathcal{S}}/\tau_{\mathrm{task}}\big)}{\sum_{j\in\mathcal{C}}\exp\big(f_\theta(\tau)^\top p_j^{\mathcal{S}}/\tau_{\mathrm{task}}\big)}. \tag{14}$$

Swapping support and query sets and averaging the two directions defines $\mathcal{L}_{\mathrm{task}}$, with temperature $\tau_{\mathrm{task}} > 0$. The reference training batch uses two support and two query trajectories per task.

A.4 Optimization schedule and context subsampling

Initial slot assignment optimizes E_g, the slot module, the variance map, the projection, the foreground scoring network, and the reconstruction decoder using Eq. (6). The task readout uses h^g in the mean–variance equations during this phase.

At the transition to readout refinement, E_f is initialized by copying E_g. The learned variance map, projection, and scoring network are retained. In the base schedule, the slot-assignment encoder and slot module are fixed, while E_f and the readout components minimize

$$\mathcal{L}_{\mathrm{read}} = \mathcal{L}_{\mathrm{task}}. \tag{15}$$

The reconstruction decoder is inactive. Initial slot assignment and the parameter copy are performed once; later representation updates continue refinement without reinitializing E_f. When coupled to a controller, these representation updates occur in the representation-learning blocks described in Section 3.5.

For a total representation-training budget of B optimizer updates, approximately $0.3B$ are allocated to initial slot assignment. The phase boundary is fixed before evaluation. The remaining updates refine task readout. This stable allocation function also constrains refinement to the initial slot assignment; continued slot-assignment updates are described in Appendix B.1.

During readout refinement, each trajectory independently retains its complete input with probability 0.5. Otherwise, we sample $q \sim \mathcal{U}[0.5, 1]$ and draw $m = \max(1, \lceil qn \rceil)$ transitions uniformly without replacement. Selection uses neither reward values nor evidence, task, or position labels. The model recomputes slot assignments from the retained set. Each retained tuple keeps its original observation difference Δo_t.

Uniform subsampling varies the amount of available experience and can remove identifying evidence. If n_e of the n original transitions are informative, a subset of size m retains mn_e/n in expectation, with no guarantee for an individual subset. Changes in the duration of a physical interaction can additionally alter which response patterns are observed. These two changes therefore require separate empirical evaluation.

Setting	Reference value
Foreground / background slots	$3/1$
Feature / output dimensions	$64/64$
Slot Attention iterations	3
Variance residual	Bias-free, zero-initialized 64×64 map
Slot-readout / task-loss temperatures	$0.5/0.1$
Training trajectories per task	4, split $2 + 2$
Initial slot-assignment budget fraction	Approximately 0.3
Subsampling probability	0.5
Subsample retention fraction	Uniform on $[0.5, 1]$
Minimum effective occupancy	2, including background
Reconstruction coefficients	$\lambda_{\mathrm{tr}} = \lambda_r = 0.1$
Assignment coefficients	$\lambda_{\mathrm{ent}} = \lambda_{\mathrm{occ}} = 0.01$
Auxiliary prediction and consistency	Inactive in the base schedule

Table 3: Reference design settings. These values define the configuration rather than measured hyperparameter optima.

A.5 Task-identification evaluation

Labeled support trajectories from evaluation tasks define task prototypes, and query trajectories are classified by similarity to those prototypes. A new task can be represented by its support prototype without modifying the encoder parameters; query labels are used to score predictions. This evaluation assesses discrimination between tasks. Control uses the resulting context embedding directly.

Support and query trajectories must be independent at the level of physical data collection. Crops, duplicated files, or reward-relabeled copies of the same physical trajectory retain shared information and cannot be counted as independent examples. The reference training batch uses two support and two query trajectories per task, with both directions included in the task loss.

A.6 Structural properties and aggregation diagnostics

Permutation invariance. For fixed parameters and deterministic inference, permuting intact transition tuples permutes the rows of A and P and preserves the slot summaries. The transition networks act independently on each tuple, the first softmax acts across slots, and slot updates sum over transitions. Induction over the slot iterations gives the property for final assignments. The weighted means, variances, and foreground readout preserve it in turn. Changing the observation–action–feedback pairing changes the tuples themselves. Shifts in physical evidence onset or duration can also change state coverage and the ratio of evidence to background; their effects remain empirical questions.

Computation. Let H denote the hidden slot width (64 in the reference setting). With L slot iterations and $K + 1$ slots, assignment interactions cost $O(Ln(K + 1)H)$, in addition to feature projections, multilayer perceptrons, and recurrent slot updates. The principal assignment tensor has $n(K + 1)$ entries. For fixed network dimensions, slot count, and iteration count, computation grows linearly with context length. The variance map contributes H^2 parameters, or 4096 when $H = 64$. Numerical implementations protect small denominators in pooling.

Weights along a trajectory. For inspection, foreground weights and pooling assignments yield

$$\omega_t = \sum_{k=1}^{K} w_k P_{tk}. \tag{16}$$

These normalized weights show where the selected slots gather their inputs along the original trajectory. They describe the mean-path aggregation. The variance residual and nonlinear projection add other dependencies, so ω_t does not provide a complete output attribution. Foreground and background roles likewise require assessment through task performance and interventions on inputs or slots.

B Alternative Training Designs

The following variants modify the objectives or update schedule while retaining the input interface and representation defined in Section 3. Each is a separate alternative to the base training schedule.

B.1 Alternating slot-assignment and readout updates

After initial slot assignment, this variant alternates slot-assignment and readout updates in a 1:1 ratio. A readout update uses the subsampling rule above, holds the slot-assignment parameters fixed, and minimizes $\mathcal{L}_{\text{task}}$ over the slot-readout encoder, variance map, projection, and scoring network. A slot-assignment update uses complete trajectories, holds the slot-readout and aggregation parameters fixed, and minimizes the same task loss over the slot-assignment encoder and slot module.

During a slot-assignment update, the fixed readout remains differentiable with respect to P, allowing gradients through the weighted means and variances into the slot-assignment encoder. These updates use no additional reconstruction, entropy, or occupancy term. This alternation concerns two components within the representation encoder. The outer alternation between representation and control learning operates at a different level.

At a fixed total optimizer-update count, continued slot-assignment updates reduce the number available for readout. The two schedules therefore allocate the same update budget differently between slot assignment and readout.

B.2 Cross-trajectory prediction

A representation inferred from one trajectory can be trained to predict outcomes in an independent trajectory from the same task. For τ^A, τ^B, let $z_A = f_\theta(\tau^A)$ and define

$$\hat{r}_t^B = f_r(z_A, \mathcal{N}_o(o_t^B), \mathcal{N}_a(a_t^B)), \qquad \widehat{\Delta o}_t^{\,B} = f_{\Delta o}(z_A, \mathcal{N}_o(o_t^B), \mathcal{N}_a(a_t^B)). \tag{17}$$

The targets are normalized reward and observation change. The prediction heads receive z_A and the current observation and action from the second trajectory. Its reward, next observation, observation change, and trajectory embedding are excluded from their inputs.

The directional loss is

$$\mathcal{L}_{A\to B}^{\text{pred}} = \mathbb{E}_{t\in\mathcal{V}_B}\left[(\hat{r}_t^B - \mathcal{N}_r(r_t^B))^2 + \frac{\|\widehat{\Delta o}_t^{\,B} - \mathcal{N}_{\Delta o}(\Delta o_t^B)\|_2^2}{d_o}\right]. \tag{18}$$

Averaging both directions gives $\mathcal{L}_{\text{pred}}$, which is added with weight 0.1 during readout refinement. Both heads predict all allowed output coordinates in every environment and are discarded at inference. When combined with continued slot-assignment updates, prediction remains restricted to readout updates.

Current observations and actions may already explain many outcomes, so the prediction error reflects information available in these inputs as well as in the inferred task representation.

B.3 Consistency under context subsampling

For a complete input X and a subsampled set X_{sub} that removes at least one transition, define

$$\mathcal{L}_{\text{cons}} = \mathbb{E}_\tau\left[1 - \cos(f_\theta(X_{\text{sub}}), \text{sg}(f_\theta(X)))\right], \tag{19}$$

where sg stops gradients through the complete-input branch. This variant adds $0.1\mathcal{L}_{\text{cons}}$ to readout updates alongside task supervision. Both branches use the same model, and the complete-input forward pass adds computation. The consistency term is applied independently of reward values or evidence labels. Because subsampling can remove identifying information, its suitability depends on the information retained in the resulting contexts.

C Analysis of Sparse Evidence Aggregation

The main text states the problem definition and the identifiability condition used for the method's motivation. This appendix records the supporting Bayes-error and data-processing results, the global-

pooling calculation, complete proofs, and the conditions behind the slot-level bound. It separates task information in the context distribution from its preservation by an intermediate slot summary. None of the results requires independence between transitions within a trajectory.

C.1 Context-distribution identifiability

Assumptions and notation. Consider a task family $\mathcal{C}$ and contexts of common length n. Each context contains exactly n_e evidence-bearing interactions, with $0 < n_e < n$ and $\rho = n_e/n$ shared across tasks. The mask m_t follows the definition in Section 3.1; it marks the use of a task-conditioned response channel, not whether a realized interaction identifies its task. The learner observes neither the mask nor the response kernels. Let $\mathcal{Y}(\tau) = \{\{\zeta_t : t \in \mathcal{V}\}\}$ be the multiset of raw interaction tuples and let $\mathsf{P}_c^{\mathcal{Y}}$ be its law under task c.

After sampling a context from task c, sample an index U uniformly from its valid indices. Write P_c for the law of ζ_U, $P_{e,c}$ for its conditional law given $m_U = 1$, and P_b for its conditional law given $m_U = 0$. Assume that P_b is common across tasks and that $P_{e,c} \neq P_{e,c'}$ whenever $c \neq c'$. These are sufficient analysis conditions, stronger than merely specifying a low evidence fraction. The shared-background assumption concerns the marginal law of complete interaction tuples; it does not follow from a task-invariant current response kernel, since state histories may retain effects of earlier task-dependent responses. A common behavior policy also does not guarantee it.

Define total variation by $\mathrm{TV}(P, Q) = \sup_D |P(D) - Q(D)|$ over measurable sets D.

Proposition 1 (identifiability under shared background). Under the preceding assumptions, the mixture identity and total-variation equality stated in the main text hold. Consequently, $\mathsf{P}_c^{\mathcal{Y}} \neq \mathsf{P}_{c'}^{\mathcal{Y}}$: the task is identifiable at the level of the interaction-multiset distribution.

Proof. Because every context contains exactly n_e evidence-bearing interactions, uniform index sampling gives $\Pr(m_U = 1 \mid c) = \rho$. Conditioning on m_U yields the mixture identity. For every measurable D, the shared background cancels:

$$P_c(D) - P_{c'}(D) = \rho\{P_{e,c}(D) - P_{e,c'}(D)\}.$$

Taking the supremum of the absolute difference gives the total-variation equality. The assumed difference between the evidence distributions and $\rho > 0$ make it positive. Finally, sampling one interaction tuple uniformly from a multiset is the same stochastic operation under every task. Equal multiset laws would therefore imply equal sampled-tuple laws, contradicting $P_c \neq P_{c'}$. This proves the conclusion.

Corollary 1 (Bayes error from one uniformly sampled interaction). Consider a binary test between tasks c and c' with equal prior probabilities. If the observation is one interaction tuple drawn by first sampling a context and then sampling one valid index uniformly, the minimum achievable classification error is

$$R_1^*(c, c') = \frac{1}{2}\left[1 - \mathrm{TV}(P_c, P_{c'})\right] = \frac{1}{2}\left[1 - \rho\,\mathrm{TV}(P_{e,c}, P_{e,c'})\right] \geq \frac{1-\rho}{2}. \tag{20}$$

Proof. Let $\nu = P_c + P_{c'}$, and write $p = dP_c/d\nu$ and $q = dP_{c'}/d\nu$. The optimal equal-prior rule chooses the task with the larger density at the observed tuple. Its error is

$$R_1^*(c, c') = \frac{1}{2}\int \min\{p, q\}\, d\nu = \frac{1}{2}\left(1 - \frac{1}{2}\int |p - q|\, d\nu\right) = \frac{1}{2}\left[1 - \mathrm{TV}(P_c, P_{c'})\right].$$

Substituting Proposition 1's total-variation identity gives the equality in the corollary. Since total variation between probability measures is at most one, the final lower bound follows. The result concerns one sampled tuple only; it does not bound the Bayes error based on the complete context.

Interpretation and scope. The result supplies a sufficient condition for context identifiability despite a small evidence fraction. It does not make an individual evidence-bearing interaction uniquely identifying. Overlapping evidence distributions and overlapping context distributions are allowed, so a finite-context classifier can still make errors. The equality concerns the marginal law of

a uniformly selected interaction tuple; it does not assert that total variation between complete-context multiset laws scales as ρ. Tasks distinguished by joint response patterns with identical single-tuple marginals fall outside this sufficient condition and may still be identifiable from the complete context.

This definition concerns whether task-conditioned context distributions differ. BOReL's MDP-ambiguity definition additionally considers alternative MDP and behavior-policy explanations (Dorfman et al., 2021). Shared behavior policies, distinguishable context laws, and exclusion of alternative MDP explanations are separate properties. In our controlled evaluations, distributional checks must be supplemented by finite-context inference and audits of task-dependent state information outside the intended evidence intervals.

Lemma 1 (aggregation contracts task separation). For any task pair c, c', let $\mathsf{P}_c^{\mathcal{Y}}$ and $\mathsf{P}_{c'}^{\mathcal{Y}}$ be their context laws, and let $Z = g(\mathcal{Y})$ be any deterministic representation with induced laws P_c^Z and $\mathsf{P}_{c'}^Z$. Then

$$\mathrm{TV}(\mathsf{P}_c^Z, \mathsf{P}_{c'}^Z) \le \mathrm{TV}(\mathsf{P}_c^{\mathcal{Y}}, \mathsf{P}_{c'}^{\mathcal{Y}}). \tag{21}$$

Proof. By definition of pushforward measure and total variation,

$$\mathrm{TV}(\mathsf{P}_c^Z, \mathsf{P}_{c'}^Z) = \sup_A \left| \mathsf{P}_c^{\mathcal{Y}}(g^{-1}(A)) - \mathsf{P}_{c'}^{\mathcal{Y}}(g^{-1}(A)) \right|.$$

The sets $g^{-1}(A)$ form a subset of all measurable sets in the context space. Taking the supremum over this restricted collection cannot exceed the supremum over all measurable context sets, which is $\mathrm{TV}(\mathsf{P}_c^{\mathcal{Y}}, \mathsf{P}_{c'}^{\mathcal{Y}})$. This proves the contraction.

Proposition 2 (global-mean separation under sparse evidence). Let ψ be a vector-valued statistic with finite expectation, and define $S_\psi(\tau) = n^{-1} \sum_{t \in \mathcal{V}} \psi(\zeta_t)$. Assume the fixed evidence fraction and common background law used above, without requiring pairwise-distinct evidence laws, and set

$$\mu_{e,c} = \mathbb{E}[\psi(\zeta_U) \mid C = c, m_U = 1], \qquad \mu_b = \mathbb{E}[\psi(\zeta_U) \mid m_U = 0].$$

Then the expectation decomposition is

$$\mathbb{E}[S_\psi \mid C = c] = \rho \mu_{e,c} + (1 - \rho)\mu_b, \tag{22}$$

and the corresponding class-conditional expectation gap is

$$\|\mathbb{E}[S_\psi \mid C = c] - \mathbb{E}[S_\psi \mid C = c']\|_2 = \rho \|\mu_{e,c} - \mu_{e,c'}\|_2. \tag{23}$$

In particular, the class-conditional expectation gap of the global mean is ρ times the corresponding evidence-only mean gap.

Proof. For every realized context, uniform sampling of U gives

$$\frac{1}{n} \sum_{t \in \mathcal{V}} \psi(\zeta_t) = \mathbb{E}[\psi(\zeta_U) \mid \tau].$$

Taking expectations conditional on $C = c$ and conditioning on m_U yields

$$\mathbb{E}[S_\psi \mid C = c] = \Pr(m_U = 1 \mid c)\mu_{e,c} + \Pr(m_U = 0 \mid c)\mu_b = \rho \mu_{e,c} + (1 - \rho)\mu_b.$$

Subtracting the corresponding identity for task c' cancels the common background term and gives

$$\mathbb{E}[S_\psi \mid C = c] - \mathbb{E}[S_\psi \mid C = c'] = \rho(\mu_{e,c} - \mu_{e,c'}).$$

Taking the Euclidean norm proves Eq. (23). This argument uses linearity of expectation, not independence among transitions.

This result quantifies attenuation of class-conditional means, not the loss of all task information under every aggregation. It does not imply that the variance, higher-order structure, or joint temporal pattern of the context is uninformative.

C.2 Off-mask contribution and preservation of task distinctions

Assumptions and notation. Fix one context, its latent evidence set $\mathcal{E}$, and any realized nonnegative slot assignments A_{tk}. Assignments may depend on the entire context. Let $\|h_t\|_2 \le R$ for all valid t, where $0 < R < \infty$. For a finite context this can be taken as a bound on its observed features; uniform claims over contexts would require a common bound. Assume $e_k = \sum_{t\in\mathcal{E}} A_{tk} > 0$, and define $b_k = \sum_{t\notin\mathcal{E}} A_{tk}$ and $\eta_k = b_k/(e_k + b_k)$. Normalize $P_{tk} = A_{tk}/(e_k + b_k)$ and set

$$\mu_k = \sum_t P_{tk} h_t, \qquad \mu_{k,e} = \frac{1}{e_k} \sum_{t\in\mathcal{E}} A_{tk} h_t.$$

For $b_k > 0$, define $\mu_{k,b} = b_k^{-1} \sum_{t\notin\mathcal{E}} A_{tk} h_t$.

Proposition 3 (off-mask deviation bound). For every such context and assignment,

$$\|\mu_k - \mu_{k,e}\|_2 \le 2R\eta_k. \tag{24}$$

If $0 < \rho < 1$, then for any prescribed tolerance $q \in (0, 1)$,

$$\eta_k \le q \quad \Longleftrightarrow \quad \frac{\bar{a}_{k,b}}{\bar{a}_{k,e}} \le \frac{q}{1-q}\frac{\rho}{1-\rho}, \tag{25}$$

where $\bar{a}_{k,e} = e_k/n_e$ and $\bar{a}_{k,b} = b_k/(n - n_e)$.

Proof. When $b_k = 0$, normalization gives $\mu_k = \mu_{k,e}$ and $\eta_k = 0$. When $b_k > 0$, splitting the sum over the mask-defined evidence and background sets gives

$$\mu_k = (1 - \eta_k)\mu_{k,e} + \eta_k \mu_{k,b}.$$

Each conditional mean is a convex combination of vectors of norm at most R, so both have norm at most R. Hence

$$\|\mu_k - \mu_{k,e}\|_2 = \eta_k \|\mu_{k,b} - \mu_{k,e}\|_2 \le \eta_k(\|\mu_{k,b}\|_2 + \|\mu_{k,e}\|_2) \le 2R\eta_k.$$

Substitution of $e_k = n\rho\bar{a}_{k,e}$ and $b_k = n(1-\rho)\bar{a}_{k,b}$ gives the off-mask-fraction identity used in the bound stated in Section 3.2. Multiplying $\eta_k \le q$ by its positive denominator yields $(1-q)(1-\rho)\bar{a}_{k,b} \le q\rho\bar{a}_{k,e}$. Division by the positive factors gives Eq. (25), with each step reversible.

Corollary (conditional separation of slot means). For a finite collection of at least two task centers $\{v_c\}$ in the same feature space, let $\Delta = \min_{c\neq c'} \|v_c - v_{c'}\|_2 > 0$. Suppose a context from task c has an evidence-only slot mean satisfying $\|\mu_{k,e} - v_c\|_2 \le \varepsilon$. If

$$\varepsilon + 2R\eta_k < \Delta/2, \tag{26}$$

then nearest-center classification of μ_k returns c.

Proof. The triangle inequality and Proposition 3 give $\|\mu_k - v_c\|_2 \le \varepsilon + 2R\eta_k < \Delta/2$. For every $c' \neq c$, the reverse triangle inequality gives $\|\mu_k - v_{c'}\|_2 \ge \|v_c - v_{c'}\|_2 - \|\mu_k - v_c\|_2 > \Delta/2$. The correct center is therefore strictly closer than every alternative. Equality in the condition is excluded because ties can occur.

Implications for the method. Uniform pooling has off-mask fraction $1 - \rho$. Slot normalization permits a different fraction determined by the learned assignments, but Eq. (25) also shows the burden imposed by increasing sparsity: a fixed tolerance on off-mask mass requires the relative off-mask weight to decrease in proportion to $\rho/(1-\rho)$. Merely creating multiple slots does not establish this property. Moreover, $\mu_{k,e}$ uses the realized evidence weights; selective retention can omit complementary clues even with low off-mask mass. The corollary explicitly accounts for this through its evidence-summary assumption.

These results concern intermediate means under stated conditions. They neither establish that training recovers suitable assignments nor certify the final nonlinear, variance-aware readout or its policy return. The implemented scorer also differs from the diagnostic nearest-center rule. Evidence retention, off-mask mass, and the downstream effect of slot interventions are therefore complementary empirical checks. For dense evidence ($\rho = 1$), the off-mask term vanishes; this mechanism supplies no predicted advantage over a simpler encoder in that regime.

Benchmark	Observation dimension	Action dimension	Horizon / step size
ADSE	27	8	200 / 0.05
HCSE	20	6	200 / 0.05
DDE	39	4	200 / 0.05

Table 4: Public interfaces used by the benchmark families.

D EXPERIMENTAL DETAILS

This appendix records the environment interfaces, task grids, evidence interventions, data counts, and training budgets used in the three primary benchmark families. Unless stated otherwise, every physical trajectory has 200 steps with $dt = 0.05$. The representation encoder receives only the public observation, action, next observation, their difference, and reward. Task parameters, task identifiers, behavior-source identifiers, evidence masks, interval boundaries, contact truth, and OOD labels are excluded.

D.1 ENVIRONMENT INTERFACES AND COMMON EVALUATION

The primary benchmarks use six context conditions, `normal`, `onset`, `short`, `long`, `onset-short`, and `onset-long`. The three task partitions are evaluated separately: `task_iid` denotes seen tasks, `task_unseen` holds out tasks within the training range, and `task_ood` holds out tasks outside that range. Thus, the complete meta-test grid has 18 task-condition combinations. A support set is used to form a task prototype, and query contexts are evaluated without updating representation parameters. Support, auxiliary evaluation, and query data are physically independent; views derived from one physical trajectory are paired only for the controlled evidence comparison and are not counted as independent physical trajectories.

Task inference is measured by cosine 1-nearest-neighbor accuracy against the support prototypes. The main representation tables report the mean and sample standard deviation across five independent seeds, `101, 202, 303, 404, 505`, broken down by task split and evidence condition. For downstream transfer, the representation is fixed before a common TD3+BC run. Every method uses the same 200k-update controller, actor and critic learning rates 3×10^{-4}, TD3+BC coefficient 2.5, hidden width 256, and batch size 256 transitions per task. ADSE and HCSE report the sum of the dense task-control reward over the full 200-step episode, while DDE reports the percentage of successful evaluation episodes. Validation is performed every 1k policy updates using two contexts per task and three environment resets per context; final evaluation uses 16 contexts per task and condition, with three resets per context. Four controller checkpoints are selected by the independent validation protocol before the reported task conditions are evaluated, and the same selected checkpoint set is used for every condition of a seed. Evaluation episodes and selected checkpoints are first averaged within each seed; the reported mean and sample standard deviation are then computed over the five seed-level values. Main Table 1 is the only authoritative normal-condition downstream summary; Appendix Tables 16–18 expand the same protocol to all 18 task-condition combinations. Task-oracle inputs expose the true task parameter and task-blind inputs use a fixed nominal parameter; neither reference reads simulator-hidden state.

D.2 ANTDIRSPARSEEVIDENCE

ADSE is a two-dimensional direction-control Ant environment with 27-dimensional observations and 8-dimensional actions. The task parameter is the target direction on the grid $4.5 + 9j$ degrees. The task split contains 20 seen directions in $[0, 270)$ degrees, 10 held-out directions in the same range, and 10 angular OOD directions in $[270, 360)$ degrees.

The data are generated by eight task-independent behavior sources mixed in equal proportions. The dataset contains 10,880 independent physical trajectories and 23,680 derived views. The representation-training set contains 4,800 trajectories from seen tasks under the normal condition. Independent support and auxiliary evaluation pools contain 640 and 320 trajectories, respectively. The final evaluator uses 64 normal support contexts per task and 64 query contexts per task and condition.

The normal evidence window starts at the first torso-displacement crossing of 1 m and ends at the first crossing of 6 m. Delayed windows begin at a torso displacement of 11 m. Short and long conditions retain, respectively, 5–15 and 80–100 steps of the otherwise available evidence; the normal window contains 20–60 steps. The physical trajectory is not cropped or padded. Context reward retains the direction-dependent velocity projection inside the selected window and is zero outside it. Downstream control uses the dense direction-velocity reward over all 200 steps. The six context views of a query share physical states, actions, and dense control reward and differ only in the context-reward window.

The formal representation run uses 200k optimizer updates, with the first 60k updates assigned to initial slot assignment. A complete 18-condition evaluation is performed every 10k updates. Reported aggregates use five independent seeds, with the mean and sample standard deviation computed at the seed level. When the separate control stage is run, TD3+BC uses 200k updates and the final evaluation contains 960 episodes per condition for seen tasks and 480 episodes per condition for both held-out and OOD tasks.

D.3 HALFCHEETAHSPARSEEVIDENCE

HCSE is a HalfCheetah environment with 20-dimensional observations and 6-dimensional actions. The task parameter is the target forward speed. There are 21 task values: 11 seen tasks with maximum target speed 1.92 m/s, five held-out tasks within the same range, and five high-speed OOD tasks at 2.04, 2.16, 2.28, 2.40, and 2.52 m/s.

Eleven behavior sources span speeds from 0.12 to 2.52 m/s in increments of 0.24 m/s. Each physical trajectory uses one source from reset to termination; no prefix switching is used. The dataset contains 8,023 independent physical sources and 14,743 derived views. The representation-training set has 4,807 trajectories, with independent support and auxiliary evaluation pools of 352 and 176 trajectories. Each task contributes 64 normal support contexts and 64 query contexts per evidence condition. The formal evaluator uses cosine 1-nearest-neighbor classification against the support prototypes.

For each behavior source, the evidence onset is triggered by the first consecutive 30-step crossing of its own displacement threshold. Normal starts occur at approximately steps 0–40 and delayed starts at steps 60–100. The normal evidence length is 20–60 steps, the short length is 5–15 steps, and the long length is 80–100 steps. We require the onset spacing to have a median of at least 40 steps and a fifth percentile of at least 20 steps. Context reward is $-|v_{\mathrm{forward}} - v_{\mathrm{target}}|$ inside the selected window and zero outside it; the dense strategy reward uses the same velocity error over all 200 steps. The six views of each physical source share the trajectory and dense reward and only change the context window.

The frozen representation run uses 200k updates and assigns the first 60k to initial slot assignment. Checkpoints are saved every 1k updates, and the complete 18-condition meta-test is run every 10k updates. If downstream TD3+BC is run, it uses 200k updates with 1k-step validation. The final control evaluation contains 528 episodes per condition for the 11 seen tasks and 240 episodes per condition for each held-out and high-speed OOD split.

D.4 DRAWERDYNAMICEVIDENCE

DDE uses the native DrawerDynamicEvidence-v4 environment. Public observations have dimension 39 and actions have dimension 4. The friction coefficient is fixed at 0.12. The hard stopping distance is 0.2 m, the position-error tolerance is 0.003 m, the velocity tolerance is 0.01, the end-effector must remain stable for 20 steps, the contact threshold is 10^{-6}, and the maximum drag power is 1.0.

The hidden task parameter is drawer damping. The ten seen damping values range approximately from 2.83 to 17.96; held-out values are 3.16, 6.32, 10.10, 14.25, and 16.80; OOD values are 20, 22, 24, 26, and 30. The normal target displacement is 0.10 m, with 0.08 m and 0.14 m for the short and long conditions. Normal windows begin within approximately 0.14 m, while delayed windows begin at 0.24–0.32 m; the remaining two conditions combine these onset and duration choices.

The training set contains 4,800 trajectories: ten seen tasks, 160 three-branch groups per task, and matched, nominal, and mismatched behavior branches in each group. The three branches share the complete approach/descend prefix and differ during pulling because of their damping conditions. These branch names describe data-generation conditions only; branch labels and damping values

are not provided to the encoder or used to select context examples. The full dataset contains 14,240 trajectories: 4,800 training, 320 auxiliary support, 160 auxiliary evaluation, 1,280 final support, and 7,680 six-condition query views. Failed, near-success, and overshoot trajectories are retained rather than filtered by success rate.

For representation input, DDE keeps the native full-trajectory drag-power dense reward; it does not convert reward to an interval-masked sparse signal. The encoder still receives only public observations, actions, next observations, and rewards, and never receives damping, task identity, contact truth, phase labels, or hidden joint state. The formal representation run uses 200k updates with 60k initial slot-assignment updates, saves a checkpoint every 1k updates, and runs a complete 18-condition meta-test every 10k updates. This run is representation-only; Appendix Table 18 uses its frozen representation in the separate common TD3+BC control stage, which is trained for 200k updates using only the matched branch. Oracle and blind controls receive true damping and fixed nominal damping, respectively, and neither can read simulator-hidden state.

D.5 ADSE ABLATION SUITE

The ablation suite is a controlled study of the ADSE representation. All variants use the normalized transition input defined in Section 3.2, exclude task and evidence labels from the forward pass, use a 64-dimensional representation, and train for 200k updates with 60k updates in the initial slot-assignment stage and 140k in readout refinement. The reference uses three foreground slots, one background slot, three Slot Attention iterations, coordinate-wise slot variance, and a learned router. Every reported ablation condition is trained with the same budget and five independent seeds.

Structural variants. The structural comparison contains five conditions. Full uses three foreground slots, one background slot, and learned router weights. Uniform slots retains the slot summaries but fixes the foreground routing weights uniformly. Pure attention reads directly from transition features, Mean pool uses a global mean, and GRU replaces slot aggregation with recurrent encoding. The comparison therefore isolates the contribution of content-based slot organization and adaptive routing from direct attention, global pooling, and recurrent aggregation.

Initial-stage loss variants. Only the first 60k updates are changed in this family; the readout-refinement stage, input, structure, update budget, and evaluation schedule remain fixed. The reference objective is

$$\mathcal{L}_{\text{init}} = \mathcal{L}_{\text{task}} + \lambda_{\text{tr}}\mathcal{L}_{\text{tr}} + \lambda_r\mathcal{L}_r + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \lambda_{\text{occ}}\mathcal{L}_{\text{occ}}, \tag{27}$$

where the task-classification, transition-reconstruction, reward-reconstruction, assignment-entropy, and occupancy terms have weights 1.0, 0.1, 0.1, 0.01, and 0.01, respectively. The objective uses no reward-derived foreground/background target or evidence mask. The task-only variant retains only $\mathcal{L}_{\text{task}}$. The no-reconstruction variant removes transition and reward reconstruction. The no-assignment-regularization variant removes entropy and occupancy. These variants test early slot formation, not the absence of supervision throughout training.

Foreground-slot count. The foreground-slot count is $K \in \{1, 2, 3, 5\}$ with one background slot in every condition. The learned router, full initial-stage objective, mean–variance readout, single reward view, and all budgets remain fixed. Because changing K also changes the assignment space and a small number of parameters, this is interpreted as a slot-capacity prior rather than a perfectly parameter-matched intervention.

Random transition retention. During representation training, a trajectory is kept complete with probability 0.5. Otherwise, a fraction sampled uniformly between a minimum retention fraction and 1.0 is retained without replacement. The sampling uses no task, reward, evidence, or position information and does not require a contiguous interval. We test minimum retention fractions 0.25, 0.50, 0.75, and 1.00. Validation and test always use the complete context, so the comparison measures whether retention training improves inference from full inputs rather than making the evaluated input easier or harder. Each reported retention condition is trained with the same five independent seeds.

Ablation metrics and reporting. Every condition saves checkpoints at 10k, 20k, . . ., 200k updates. We report task-inference accuracy on the independent evaluation split, seen tasks, in-range held-out

tasks, and OOD tasks, covering all six evidence conditions and mean $\pm$ standard deviation over the reported independently trained runs. Representation curves use all 20 raw checkpoints without smoothing.

Structural runs additionally record effective slot count, per-slot occupancy, background assignment mass, router concentration, and the assignment weights mapped back to original trajectory positions. These are diagnostics of the learned organization, not evidence labels or direct claims that a slot recovers a ground-truth interval.

Downstream transfer. Each representation variant can be connected to a frozen-encoder TD3+BC learner. The policy is trained for 200k updates from random initialization, with actor and critic learning rates 3×10^{-4}, TD3+BC coefficient 2.5, hidden width 256, and a batch of 256 transitions per task; validation occurs every 1k updates. After training, the policy is evaluated using the same contexts and paired resets as the main ADSE protocol. This transfer test asks whether an ablation's representation difference is reflected in control, while keeping the controller and evaluation budget fixed.

Interpretation boundary. These ablations identify the value of multi-slot organization, foreground capacity, routing, initial-stage auxiliary objectives, and retention training. They do not show that a slot equals a ground-truth evidence interval, that the background slot is task-independent, or that reward is the causal source of the learned task representation. Such claims require evidence annotations or additional temporal intervention diagnostics.

All reported ablation tables and curves use the five seed-level runs under the common budget and evaluation protocol.

E Complete Representation Results

This appendix collects the complete task-identification and downstream-control results. The standard `normal` condition is summarized once in Table 6; the non-normal conditions are split into nine readable tables by benchmark and task split, with the DDE tables also retaining `normal` for completeness. For downstream control, Main Table 1 is the only authoritative normal-condition summary, and the appendix grids expand the same protocol to the remaining evidence conditions without redefining the normal results. Identification values are cosine 1-nearest-neighbor accuracy (%), reported as mean $\pm$ sample standard deviation. Downstream tables use the common frozen-representation TD3+BC evaluation described in Appendix D. ADSE and HCSE report full-horizon dense task-control returns, while DDE reports task-success rates (%).

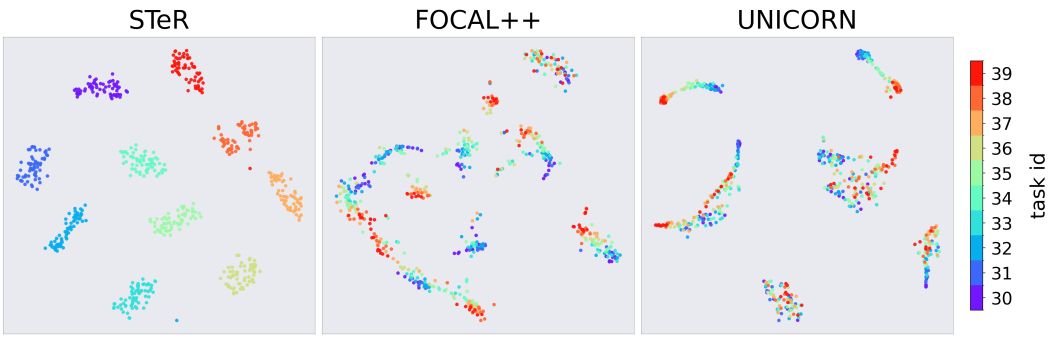

Figure 6: Qualitative t-SNE comparison on ADSE OOD `onset-long`. Colors denote task identity.

Figure 6 gives a qualitative view of the task representations in a difficult ADSE condition. Under OOD tasks with delayed and extended evidence (`onset-long`), the STeR embeddings form more compact task-wise clusters, whereas the comparison embeddings show stronger interleaving across task colors.

STeR downstream control across evidence conditions							
Benchmark	**Split**	normal	onset	short	long	onset-short	onset-long
ADSE	IID	482.73 ± 5.26	486.53 ± 5.36	466.08 ± 5.27	488.13 ± 7.36	466.16 ± 5.17	487.30 ± 4.84
ADSE	Unseen	420.13 ± 26.63	424.36 ± 27.55	411.79 ± 19.96	418.59 ± 26.62	412.99 ± 23.55	419.96 ± 24.48
ADSE	OOD	379.01 ± 18.58	-293.42 ± 62.69	-247.47 ± 55.98	-292.98 ± 65.29	-265.76 ± 39.56	-294.37 ± 61.08
HCSE	IID	-18.97 ± 2.00	-18.77 ± 2.09	-19.58 ± 1.67	-18.51 ± 2.01	-19.19 ± 2.24	-19.20 ± 2.10
HCSE	Unseen	-20.54 ± 2.57	-20.20 ± 2.52	-21.02 ± 2.38	-20.22 ± 2.45	-20.28 ± 2.66	-21.48 ± 2.98
HCSE	OOD	-128.28 ± 14.76	-127.83 ± 15.09	-132.59 ± 15.19	-126.34 ± 14.20	-129.16 ± 14.77	-126.95 ± 14.62
DDE	IID	98.08 ± 0.46	98.00 ± 1.62	100.00 ± 0.00	100.00 ± 0.00	90.50 ± 1.90	100.00 ± 0.00
DDE	Unseen	99.79 ± 0.29	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	98.80 ± 1.80	100.00 ± 0.00
DDE	OOD	90.31 ± 1.61	85.30 ± 2.90	96.10 ± 1.70	86.50 ± 2.80	85.40 ± 3.50	88.50 ± 1.50

Table 5: Supplementary STeR downstream-control summary over all six evidence conditions. ADSE and HCSE report full-horizon dense task-control return; DDE reports task-success rate (%).

	ADSE			**HCSE**			**DDE**		
Method	IID	Unseen	OOD	IID	Unseen	OOD	IID	Unseen	OOD
STeR	$\mathbf{99.24 \pm 0.20}$	$\mathbf{99.93 \pm 0.06}$	$\mathbf{97.75 \pm 0.52}$	$\mathbf{99.55 \pm 0.10}$	$\mathbf{98.69 \pm 0.53}$	$\mathbf{72.33 \pm 8.74}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{99.30 \pm 0.70}$
UNICORN	13.34 ± 4.67	26.16 ± 7.97	15.81 ± 2.26	25.60 ± 4.01	45.19 ± 5.31	25.88 ± 2.23	49.1 ± 5.9	60.7 ± 4.7	35.8 ± 14.9
FOCAL++	35.45 ± 2.96	47.84 ± 4.49	21.59 ± 2.65	51.34 ± 3.94	67.50 ± 7.33	48.06 ± 4.89	87.6 ± 14.9	90.1 ± 11.3	66.9 ± 16.8
ContraBAR	5.23 ± 0.47	11.41 ± 0.40	9.50 ± 0.98	9.01 ± 0.69	19.94 ± 1.44	18.75 ± 1.45	79.0 ± 5.1	79.1 ± 5.6	89.8 ± 5.5
BOReL	14.19 ± 2.08	25.25 ± 3.94	17.34 ± 2.18	62.19 ± 0.68	76.81 ± 2.51	40.25 ± 3.99	96.3 ± 0.8	95.9 ± 1.7	83.1 ± 1.2
CTMRL	8.98 ± 2.98	16.69 ± 5.01	11.66 ± 2.12	20.26 ± 15.04	41.88 ± 31.07	23.25 ± 4.70	42.6 ± 44.9	49.6 ± 41.2	38.9 ± 25.8

Table 6: Normal-condition task-identification accuracy (%). Columns are grouped by benchmark and task split: in-distribution (IID), held-out within-range (Unseen), and out-of-distribution (OOD).

AntDirSparseEvidence (ADSE), IID tasks					
Method	onset	short	long	onset-short	onset-long
STeR	99.01 ± 0.30	87.77 ± 1.97	99.98 ± 0.03	87.16 ± 1.77	99.88 ± 0.19
UNICORN	13.80 ± 4.30	6.33 ± 0.60	9.70 ± 2.17	6.38 ± 0.52	9.69 ± 1.53
FOCAL++	33.14 ± 3.62	13.30 ± 0.43	35.55 ± 3.51	13.89 ± 0.63	35.56 ± 4.02
ContraBAR	5.23 ± 0.42	5.22 ± 0.46	5.23 ± 0.48	5.23 ± 0.47	5.20 ± 0.52
BOReL	7.19 ± 0.78	9.48 ± 1.70	13.70 ± 0.86	5.64 ± 0.41	8.30 ± 1.78
CTMRL	8.39 ± 2.60	6.05 ± 0.74	6.92 ± 1.84	6.09 ± 0.87	7.23 ± 2.60

Table 7: ADSE task-identification accuracy (%) for IID tasks under the five non-normal evidence conditions.

AntDirSparseEvidence (ADSE), Unseen tasks					
Method	onset	short	long	onset-short	onset-long
STeR	99.92 ± 0.04	94.38 ± 0.73	100.00 ± 0.00	94.94 ± 1.28	100.00 ± 0.00
UNICORN	26.09 ± 8.26	12.69 ± 1.66	18.97 ± 4.11	13.91 ± 1.69	20.12 ± 4.55
FOCAL++	46.03 ± 4.81	21.50 ± 1.98	49.09 ± 5.06	22.53 ± 1.68	49.38 ± 5.06
ContraBAR	11.44 ± 0.47	11.34 ± 0.42	11.44 ± 0.40	11.34 ± 0.42	11.53 ± 0.36
BOReL	13.25 ± 1.81	18.62 ± 4.17	23.50 ± 2.09	10.41 ± 0.93	14.97 ± 2.24
CTMRL	16.00 ± 4.11	11.28 ± 1.12	12.78 ± 4.33	11.41 ± 0.95	14.12 ± 4.61

Table 8: ADSE task-identification accuracy (%) for Unseen tasks under the five non-normal evidence conditions.

AntDirSparseEvidence (ADSE), OOD tasks					
Method	onset	short	long	onset-short	onset-long
STeR	97.91 ± 0.84	76.05 ± 2.27	99.90 ± 0.10	77.68 ± 2.40	99.58 ± 0.43
UNICORN	16.41 ± 2.67	11.59 ± 0.63	12.59 ± 1.50	11.12 ± 0.99	12.81 ± 1.10
FOCAL++	21.59 ± 1.03	13.50 ± 1.97	22.19 ± 2.22	14.16 ± 1.40	21.50 ± 2.27
ContraBAR	9.62 ± 1.05	9.59 ± 0.93	9.56 ± 1.10	9.56 ± 0.96	9.53 ± 0.89
BOReL	11.00 ± 1.80	13.09 ± 2.55	16.44 ± 1.10	10.50 ± 1.27	11.28 ± 1.42
CTMRL	11.78 ± 2.05	11.00 ± 1.45	9.94 ± 2.54	11.53 ± 0.69	10.38 ± 1.24

Table 9: ADSE task-identification accuracy (%) for OOD tasks under the five non-normal evidence conditions.

HalfCheetahSparseEvidence (HCSE), IID tasks					
Method	onset	short	long	onset-short	onset-long
STeR	99.90 ± 0.06	97.63 ± 0.36	99.60 ± 0.16	99.50 ± 0.15	98.69 ± 0.64
UNICORN	18.81 ± 3.10	10.71 ± 0.61	13.75 ± 1.89	11.05 ± 0.93	13.07 ± 3.51
FOCAL++	46.19 ± 7.95	15.85 ± 1.06	40.34 ± 7.25	18.07 ± 3.50	34.52 ± 7.13
ContraBAR	8.89 ± 0.62	8.86 ± 0.78	8.78 ± 0.73	8.84 ± 0.83	8.86 ± 0.37
BOReL	14.43 ± 2.47	26.93 ± 2.07	39.89 ± 5.91	9.72 ± 1.60	14.89 ± 2.31
CTMRL	20.14 ± 15.04	12.67 ± 5.73	17.90 ± 12.25	12.53 ± 5.83	18.32 ± 12.08

Table 10: HCSE task-identification accuracy (%) for IID tasks under the five non-normal evidence conditions.

HalfCheetahSparseEvidence (HCSE), Unseen tasks					
Method	onset	short	long	onset-short	onset-long
STeR	99.12 ± 0.31	96.72 ± 1.15	98.98 ± 0.92	97.91 ± 0.81	98.94 ± 0.68
UNICORN	36.38 ± 4.51	23.19 ± 1.83	25.69 ± 1.84	24.38 ± 1.78	23.00 ± 4.74
FOCAL++	63.31 ± 9.21	26.25 ± 2.39	61.00 ± 8.69	30.62 ± 4.99	51.19 ± 9.55
ContraBAR	19.88 ± 1.66	19.88 ± 1.43	19.81 ± 1.54	19.94 ± 1.44	20.06 ± 1.40
BOReL	25.25 ± 2.93	45.38 ± 4.80	58.81 ± 4.84	20.69 ± 0.81	26.62 ± 3.40
CTMRL	38.31 ± 24.12	25.81 ± 7.46	29.12 ± 14.86	25.38 ± 7.71	28.75 ± 11.77

Table 11: HCSE task-identification accuracy (%) for Unseen tasks under the five non-normal evidence conditions.

HalfCheetahSparseEvidence (HCSE), OOD tasks					
Method	onset	short	long	onset-short	onset-long
STeR	72.12 ± 9.58	55.34 ± 8.27	75.20 ± 11.58	60.64 ± 9.30	67.44 ± 14.17
UNICORN	23.00 ± 2.01	19.81 ± 2.06	23.00 ± 3.27	22.31 ± 2.58	20.50 ± 2.21
FOCAL++	44.12 ± 6.85	26.19 ± 3.26	45.56 ± 6.16	27.88 ± 3.39	40.56 ± 8.44
ContraBAR	18.62 ± 1.37	18.75 ± 1.45	18.50 ± 1.28	18.75 ± 1.45	18.38 ± 1.66
BOReL	22.25 ± 2.46	27.31 ± 3.47	29.44 ± 3.05	20.19 ± 1.05	20.56 ± 3.53
CTMRL	24.06 ± 4.47	20.81 ± 2.22	20.31 ± 1.96	21.56 ± 2.17	21.69 ± 2.10

Table 12: HCSE task-identification accuracy (%) for OOD tasks under the five non-normal evidence conditions.

DrawerDynamicEvidence (DDE), IID tasks						
Method	normal	onset	short	long	onset-short	onset-long
STeR	100.0 ± 0.0	76.6 ± 4.8	87.5 ± 2.5	87.6 ± 7.7	57.8 ± 1.6	84.9 ± 7.3
UNICORN	49.1 ± 5.9	15.7 ± 7.5	20.1 ± 11.6	24.8 ± 8.4	14.6 ± 6.7	12.0 ± 3.7
FOCAL++	87.6 ± 14.9	12.8 ± 2.0	83.4 ± 14.4	72.1 ± 9.0	11.0 ± 1.7	12.1 ± 2.3
ContraBAR (sup.)	79.0 ± 5.1	22.2 ± 1.8	12.8 ± 3.2	10.8 ± 1.1	16.0 ± 5.6	9.8 ± 0.5
BOReL	96.3 ± 0.8	31.2 ± 7.9	94.9 ± 3.0	91.9 ± 5.9	30.0 ± 8.9	32.1 ± 6.0
CTMRL	42.6 ± 44.9	10.7 ± 3.3	15.9 ± 9.2	15.4 ± 7.6	10.0 ± 2.1	12.0 ± 4.0

Table 13: DDE task-identification accuracy (%) for IID tasks under all six evidence conditions.

DrawerDynamicEvidence (DDE), Unseen tasks						
Method	normal	onset	short	long	onset-short	onset-long
STeR	100.0 ± 0.0	86.5 ± 5.6	99.2 ± 1.6	97.0 ± 4.1	78.1 ± 4.9	88.5 ± 7.1
UNICORN	60.7 ± 4.7	29.9 ± 14.0	43.4 ± 13.9	46.3 ± 10.6	26.0 ± 11.4	27.1 ± 9.5
FOCAL++	90.1 ± 11.3	25.4 ± 5.5	89.8 ± 10.7	79.8 ± 7.2	24.2 ± 3.7	26.8 ± 5.5
ContraBAR (sup.)	79.1 ± 5.6	40.2 ± 1.9	21.2 ± 3.3	22.2 ± 3.7	29.4 ± 5.5	22.9 ± 1.4
BOReL	95.9 ± 1.7	42.9 ± 7.8	95.8 ± 2.2	96.5 ± 2.4	44.5 ± 8.4	44.1 ± 5.6
CTMRL	49.6 ± 41.2	24.3 ± 8.2	30.8 ± 15.4	27.9 ± 10.8	23.0 ± 6.3	23.2 ± 6.8

Table 14: DDE task-identification accuracy (%) for Unseen tasks under all six evidence conditions.

DrawerDynamicEvidence (DDE), OOD tasks						
Method	normal	onset	short	long	onset-short	onset-long
STeR	99.3 ± 0.7	47.7 ± 14.0	70.2 ± 28.0	52.7 ± 23.3	47.9 ± 9.3	52.6 ± 27.4
UNICORN	35.8 ± 14.9	23.3 ± 4.8	22.3 ± 3.9	22.2 ± 3.3	19.8 ± 1.1	23.3 ± 3.3
FOCAL++	66.9 ± 16.8	19.3 ± 1.1	59.8 ± 14.9	51.4 ± 2.8	18.7 ± 0.9	20.7 ± 0.3
ContraBAR (sup.)	89.8 ± 5.5	28.3 ± 4.4	21.9 ± 3.0	24.4 ± 4.8	20.6 ± 0.8	22.3 ± 2.5
BOReL	83.1 ± 1.2	21.3 ± 4.5	82.0 ± 2.4	78.9 ± 5.8	21.2 ± 5.2	21.7 ± 3.3
CTMRL	38.9 ± 25.8	20.0 ± 0.4	20.7 ± 1.3	19.8 ± 3.1	20.2 ± 0.2	19.8 ± 0.8

Table 15: DDE task-identification accuracy (%) for OOD tasks under all six evidence conditions.

COMPLETE DOWNSTREAM-CONTROL RESULTS

The following tables report the complete downstream-control grids used for the evidence interventions. All methods use the same frozen-representation TD3+BC controller and the checkpoint, seed, and aggregation protocol in Appendix D. ADSE and HCSE report full-horizon dense task-control return, while DDE reports task-success rate (%).

AntDirSparseEvidence (ADSE)							
Method	**Split**	normal	onset	short	long	onset-short	onset-long
STeR	IID	482.73 ± 5.26	486.53 ± 5.36	466.08 ± 5.27	488.13 ± 7.36	466.16 ± 5.17	487.30 ± 4.84
STeR	Unseen	420.13 ± 26.63	424.36 ± 27.55	411.79 ± 19.96	418.59 ± 26.62	412.99 ± 23.55	419.96 ± 24.48
STeR	OOD	379.01 ± 18.58	-293.42 ± 62.69	-247.47 ± 55.98	-292.98 ± 65.29	-265.76 ± 39.56	-294.37 ± 61.08
UNICORN	IID	265.29 ± 78.01	258.38 ± 75.14	177.88 ± 34.71	208.83 ± 43.63	178.37 ± 37.19	209.28 ± 45.49
UNICORN	Unseen	253.02 ± 70.21	252.63 ± 70.53	188.61 ± 43.63	196.69 ± 39.89	185.17 ± 39.86	193.01 ± 35.23
UNICORN	OOD	-17.58 ± 204.76	-14.28 ± 210.43	-185.59 ± 112.61	47.32 ± 247.42	-175.91 ± 121.16	43.14 ± 246.31
FOCAL++	IID	219.41 ± 84.32	210.75 ± 74.67	140.90 ± 49.48	172.22 ± 84.61	149.78 ± 56.28	170.47 ± 92.25
FOCAL++	Unseen	208.00 ± 69.54	196.11 ± 76.67	141.54 ± 57.55	157.44 ± 76.63	143.12 ± 60.90	161.29 ± 79.22
FOCAL++	OOD	20.48 ± 24.82	20.71 ± 22.91	-77.84 ± 27.60	60.77 ± 23.60	-62.59 ± 37.42	65.01 ± 25.77
ContraBAR (sup.)	IID	43.27 ± 15.28	38.97 ± 12.46	35.26 ± 11.97	39.78 ± 8.81	34.05 ± 14.34	44.01 ± 9.32
ContraBAR (sup.)	Unseen	48.20 ± 12.65	37.27 ± 16.78	37.05 ± 16.92	46.92 ± 15.56	38.47 ± 16.73	45.01 ± 12.97
ContraBAR (sup.)	OOD	-133.34 ± 39.15	-121.50 ± 37.69	-127.98 ± 33.65	-121.40 ± 37.52	-124.95 ± 36.24	-113.51 ± 40.31
BOReL	IID	259.77 ± 26.19	148.80 ± 28.92	210.35 ± 35.64	247.63 ± 40.34	121.10 ± 25.29	155.89 ± 32.89
BOReL	Unseen	260.51 ± 28.26	161.76 ± 24.29	224.68 ± 31.83	243.42 ± 35.65	140.66 ± 27.32	165.85 ± 34.59
BOReL	OOD	47.49 ± 58.15	-142.33 ± 72.97	-62.99 ± 65.90	52.38 ± 66.96	-190.90 ± 60.59	-107.13 ± 79.16
CTMRL	IID	191.83 ± 68.09	191.46 ± 68.84	142.39 ± 22.45	158.32 ± 103.19	139.82 ± 22.05	157.38 ± 108.72
CTMRL	Unseen	185.94 ± 52.52	181.76 ± 55.92	145.33 ± 15.44	142.30 ± 105.70	142.72 ± 14.73	143.36 ± 96.95
CTMRL	OOD	-229.00 ± 158.19	-234.36 ± 148.92	-298.24 ± 86.34	-159.17 ± 216.43	-301.34 ± 85.51	-160.18 ± 217.53

Table 16: ADSE downstream full-horizon dense task-control return across IID, Unseen, and OOD tasks and six evidence conditions.

HalfCheetahSparseEvidence (HCSE)							
Method	**Split**	normal	onset	short	long	onset-short	onset-long
STeR	IID	-18.97 ± 2.00	-18.77 ± 2.09	-19.58 ± 1.67	-18.51 ± 2.01	-19.19 ± 2.24	-19.20 ± 2.10
STeR	Unseen	-20.54 ± 2.57	-20.20 ± 2.52	-21.02 ± 2.38	-20.22 ± 2.45	-20.28 ± 2.66	-21.48 ± 2.98
STeR	OOD	-128.28 ± 14.76	-127.83 ± 15.09	-132.59 ± 15.19	-126.34 ± 14.20	-129.16 ± 14.77	-126.95 ± 14.62
UNICORN	IID	-76.95 ± 10.43	-89.33 ± 3.96	-111.38 ± 7.02	-98.53 ± 7.00	-113.15 ± 4.75	-111.87 ± 11.19
UNICORN	Unseen	-61.21 ± 7.26	-74.16 ± 6.33	-99.15 ± 7.13	-79.38 ± 3.67	-103.10 ± 4.74	-86.19 ± 16.82
UNICORN	OOD	-208.79 ± 15.90	-229.06 ± 7.84	-236.25 ± 21.59	-255.73 ± 35.43	-236.19 ± 16.21	-291.46 ± 24.46
FOCAL++	IID	-49.67 ± 6.72	-50.88 ± 5.59	-94.69 ± 4.24	-65.42 ± 12.24	-91.40 ± 5.85	-71.40 ± 13.31
FOCAL++	Unseen	-49.25 ± 4.55	-49.87 ± 3.65	-85.82 ± 6.28	-62.40 ± 8.51	-81.99 ± 12.02	-61.36 ± 6.72
FOCAL++	OOD	-218.05 ± 26.92	-224.69 ± 31.79	-204.39 ± 17.50	-255.66 ± 17.76	-204.47 ± 20.38	-256.07 ± 25.83
ContraBAR (sup.)	IID	-102.84 ± 1.81	-102.75 ± 0.93	-102.75 ± 0.86	-102.70 ± 0.95	-102.84 ± 1.01	-102.88 ± 1.14
ContraBAR (sup.)	Unseen	-95.16 ± 4.06	-93.72 ± 2.60	-93.73 ± 2.44	-93.78 ± 2.30	-93.74 ± 2.37	-93.77 ± 2.24
ContraBAR (sup.)	OOD	-233.86 ± 7.16	-240.03 ± 5.37	-240.00 ± 5.49	-240.16 ± 5.20	-239.68 ± 5.19	-239.48 ± 5.41
BOReL	IID	-40.60 ± 1.88	-114.40 ± 6.08	-54.42 ± 7.82	-45.71 ± 2.70	-125.35 ± 10.78	-113.32 ± 6.35
BOReL	Unseen	-38.42 ± 2.49	-124.73 ± 6.58	-49.72 ± 7.20	-37.40 ± 3.44	-136.64 ± 10.12	-122.26 ± 7.68
BOReL	OOD	-169.62 ± 13.39	-160.16 ± 15.92	-155.74 ± 8.36	-184.77 ± 16.28	-154.29 ± 13.89	-172.97 ± 16.74
CTMRL	IID	-88.53 ± 21.47	-88.11 ± 20.99	-100.86 ± 1.31	-95.54 ± 17.32	-101.15 ± 0.98	-93.78 ± 18.67
CTMRL	Unseen	-83.15 ± 17.22	-82.19 ± 16.82	-94.08 ± 6.19	-81.30 ± 18.32	-94.04 ± 4.43	-79.90 ± 19.73
CTMRL	OOD	-215.58 ± 29.12	-221.85 ± 24.53	-229.17 ± 22.63	-247.02 ± 48.58	-229.88 ± 23.75	-246.65 ± 47.49

Table 17: HCSE downstream full-horizon dense task-control return across IID, Unseen, and OOD tasks and six evidence conditions.

DrawerDynamicEvidence (DDE)							
Method	**Split**	normal	onset	short	long	onset-short	onset-long
STeR	IID	98.08 ± 0.46	98.00 ± 1.62	100.00 ± 0.00	100.00 ± 0.00	90.50 ± 1.90	100.00 ± 0.00
STeR	Unseen	99.79 ± 0.29	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	98.80 ± 1.80	100.00 ± 0.00
STeR	OOD	90.31 ± 1.61	85.30 ± 2.90	96.10 ± 1.70	86.50 ± 2.80	85.40 ± 3.50	88.50 ± 1.50
UNICORN	IID	41.87 ± 1.85	39.80 ± 1.90	39.20 ± 2.40	43.80 ± 5.10	38.20 ± 5.30	37.10 ± 3.60
UNICORN	Unseen	48.93 ± 1.94	46.60 ± 2.10	52.00 ± 4.20	51.80 ± 2.30	44.80 ± 4.50	44.80 ± 4.00
UNICORN	OOD	44.22 ± 1.43	42.30 ± 1.20	43.10 ± 3.00	42.60 ± 1.90	41.70 ± 3.70	43.80 ± 1.20
FOCAL++	IID	67.40 ± 1.91	54.40 ± 1.80	82.60 ± 2.30	82.80 ± 3.80	52.40 ± 5.20	50.80 ± 1.80
FOCAL++	Unseen	70.39 ± 1.83	59.70 ± 2.70	81.20 ± 2.40	80.60 ± 2.70	58.20 ± 4.30	60.90 ± 3.60
FOCAL++	OOD	66.50 ± 2.29	55.90 ± 2.40	76.40 ± 3.00	72.60 ± 4.90	53.80 ± 2.00	58.60 ± 4.30
ContraBAR (sup.)	IID	43.03 ± 1.91	43.90 ± 5.40	38.00 ± 2.70	39.70 ± 2.30	39.80 ± 3.00	38.50 ± 2.10
ContraBAR (sup.)	Unseen	50.11 ± 2.99	57.20 ± 1.40	45.30 ± 3.10	43.70 ± 4.30	50.20 ± 2.40	48.10 ± 5.00
ContraBAR (sup.)	OOD	48.06 ± 2.81	49.90 ± 3.60	44.80 ± 4.40	48.50 ± 5.50	43.80 ± 3.60	45.20 ± 2.40
BOReL	IID	82.57 ± 2.41	69.90 ± 3.60	96.60 ± 3.10	96.60 ± 2.30	68.30 ± 2.80	68.00 ± 2.00
BOReL	Unseen	86.56 ± 2.48	74.80 ± 4.10	97.20 ± 1.60	96.80 ± 3.00	78.00 ± 3.00	75.80 ± 4.80
BOReL	OOD	79.69 ± 2.08	66.20 ± 3.20	93.90 ± 3.20	92.60 ± 3.30	65.20 ± 4.20	63.90 ± 2.80
CTMRL	IID	34.69 ± 1.62	32.60 ± 4.60	33.00 ± 3.30	35.00 ± 1.90	34.00 ± 3.00	30.70 ± 2.10
CTMRL	Unseen	40.74 ± 1.88	40.00 ± 2.40	41.50 ± 3.90	41.10 ± 4.50	36.80 ± 1.90	39.60 ± 3.30
CTMRL	OOD	39.18 ± 2.30	38.20 ± 4.60	37.60 ± 2.50	37.40 ± 4.90	37.90 ± 4.20	37.90 ± 2.60

Table 18: DDE downstream task-success rate (%) across IID, Unseen, and OOD tasks and six evidence conditions.

F Evidence Attribution Diagnostics

These diagnostics examine how the frozen representation responds to controlled evidence interventions. They are forward-only evaluations: no parameters are updated, checkpoints are not reselected using the diagnostic scores, and no downstream policy is trained. All three task partitions and six context conditions are evaluated. The HCSE run uses seeds 101, 202, and 303 with the validation-selected checkpoint for each seed. The ADSE run uses seed 101 and the five checkpoints selected by the formal evaluation, with the plus and minus reward views of the frozen representation. Because the readout and checkpoint coverage differ between the two runs, their numerical values are not used for cross-benchmark ranking.

The evidence mass is the routed mass assigned to the benchmark-defined evidence mask; off-mask mass is its complement. Representation change is $1 - \cos(z_{\text{full}}, z_{\text{intervention}})$. Evidence removal deletes the evidence-bearing interactions, background removal deletes the complement, and evidence replacement keeps the background while swapping an evidence segment from a donor task with the same split and condition. HCSE intervention accuracies use the full-support nearest-support readout, whereas ADSE intervention accuracies use a support prototype matched to the intervention input. The replacement source-task accuracy and switch rate below are computed on the ADSE subset with exactly matched evidence counts.

HCSE frozen-forward attribution summary							
Split	evidence mass	AUPRC	leakage	full	remove-evidence	remove-background	replace-evidence
IID	47.70%	49.56%	21.44%	99.20%	9.18%	64.73%	3.65%
Unseen	46.57%	48.15%	22.24%	98.39%	20.40%	76.56%	8.23%
OOD	34.90%	40.70%	24.60%	73.33%	18.82%	32.00%	15.83%

Table 19: HCSE attribution diagnostics aggregated over the 18 seed–condition records in each split. "Full", "remove-evidence", "remove-background", and "replace-evidence" report task-identification accuracy; leakage is the evidence-to-background routing leakage.

The HCSE results show that evidence contributes to the task representation: removing it reduces task identification in all three splits. Background removal also changes the representation, especially for IID and Unseen tasks, so the diagnostic does not support a claim of complete evidence/background separation. The lower evidence mass, AUPRC, and higher leakage under OOD provide a direct measure of the weaker attribution quality in that split.

HCSE slot-level evidence statistics						
Split	**Slot**	hard mass	hard precision	evidence recall	soft precision	soft recall
IID	slot 0	12.64%	40.84%	12.70%	40.82%	13.07%
IID	slot 1	18.54%	69.42%	27.55%	64.25%	27.71%
IID	slot 2	16.28%	40.12%	24.37%	41.47%	23.51%
IID	slot 3	32.21%	10.97%	13.94%	11.61%	14.84%
IID	slot 4	20.34%	35.25%	21.44%	33.48%	20.88%
Unseen	slot 0	12.24%	40.01%	11.70%	39.90%	12.35%
Unseen	slot 1	18.92%	68.99%	28.18%	63.11%	28.16%
Unseen	slot 2	16.40%	39.35%	23.94%	41.04%	23.40%
Unseen	slot 3	32.20%	10.52%	13.93%	11.37%	14.95%
Unseen	slot 4	20.24%	35.16%	22.24%	33.85%	21.14%
OOD	slot 0	12.44%	41.92%	13.55%	41.90%	13.60%
OOD	slot 1	18.62%	70.12%	28.05%	67.52%	28.15%
OOD	slot 2	14.37%	37.35%	24.37%	39.63%	23.82%
OOD	slot 3	31.53%	7.53%	9.43%	8.50%	10.83%
OOD	slot 4	23.04%	32.35%	24.60%	31.73%	23.60%

Table 20: HCSE slot-level routing statistics. Slot indices are local to the aggregate and are not aligned across seeds; the table therefore reports statistics rather than assigning a fixed semantic role to a slot.

The slot-level statistics further show that the routing is mixed: the highest soft evidence precision is about 63–68% for the aggregate slot labelled slot 1, while other slots contain substantial background mass. This is consistent with evidence participation in the representation, but not with a clean, seed-invariant evidence slot.

ADSE frozen-forward attribution at the selected checkpoint								
Split	evidence mass	bg. precision	leakage	full	remove-evidence	remove-background	source-task acc.	source switch
IID	99.97%	99.86%	0.37%	95.94%	4.38%	95.10%	95.71%	92.26%
Unseen	99.97%	99.86%	0.35%	98.23%	9.71%	98.10%	98.12%	96.59%
OOD	99.97%	99.84%	0.39%	91.38%	10.36%	90.16%	91.34%	84.71%

Table 21: ADSE attribution diagnostics at the selected checkpoint (step 120k, seed 101), averaged over the plus and minus reward views. Accuracy columns report task identification; source-task accuracy and source switch use the evidence-count-matched replacement subset.

Across the five selected ADSE checkpoints and both reward views, evidence mass remains in the range 99.93–99.97%, soft background precision in 99.83–99.94%, and evidence-to-background leakage in 0.13–0.40%. The intervention result is therefore representative of the routing pattern observed across the selected checkpoints, while the single-seed design does not establish seed-level stability. Removing evidence produces a large representation change, whereas removing background produces a small change; replacement also changes the task readout, including on OOD tasks.

These diagnostics establish evidence dependence and quantify the amount of routed background under the stated interventions. They do not prove a complete causal decomposition of the nonlinear encoder, because the routed mass does not account for every covariance, projection, and readout interaction. The evidence masks are used only for evaluation, slot indices have permutation ambiguity, and the HCSE and ADSE protocols retain the readout and checkpoint differences stated above.

G SLOT-TIME DIAGNOSTICS

This appendix collects the slot-time diagnostics for the three primary benchmarks. Each six-panel group keeps the six evidence conditions together: the first row contains `normal`, `onset`, and `short`, while the second row contains `long`, `onset-short`, and `onset-long`. Each panel is generated from one representative checkpoint (ADSE seed 505 at step 190k, HCSE seed 404 at step 156k, and DDE seed 505 at step 183k). These checkpoints are used for visualization only; they are not used to rank methods, estimate uncertainty, or form aggregate results. The plots are qualitative diagnostics: the horizontal axis visualizes the original trajectory order, vertical lines mark benchmark-defined evidence intervals, and the bottom readout is rescaled for comparison. STeR does

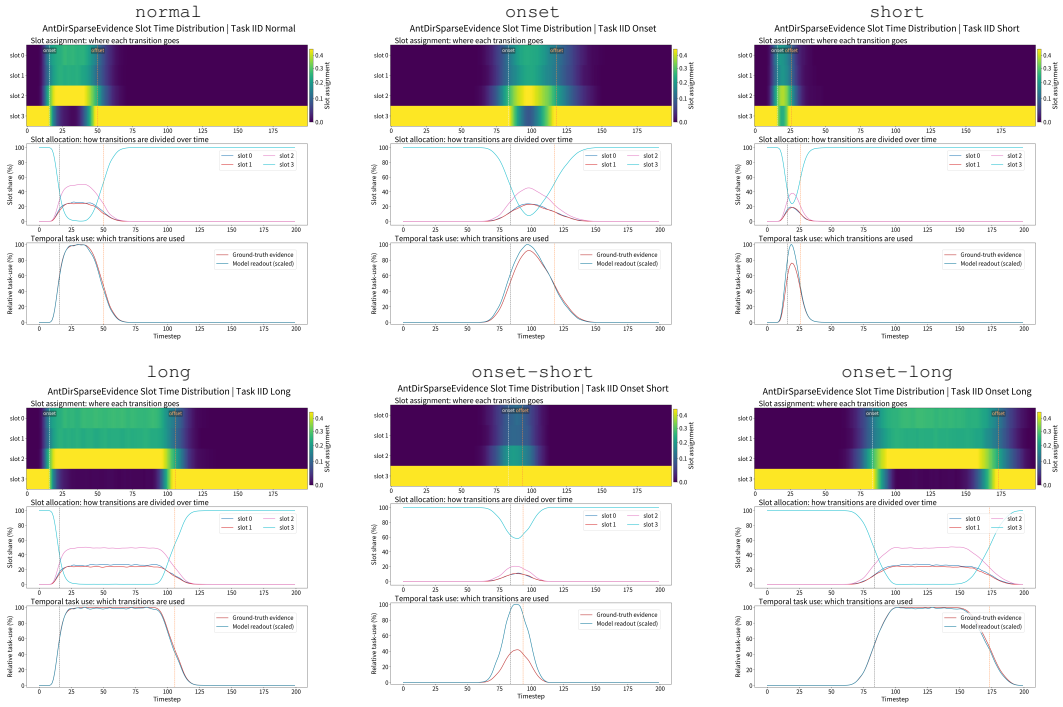

Figure 7: ADSE, IID tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

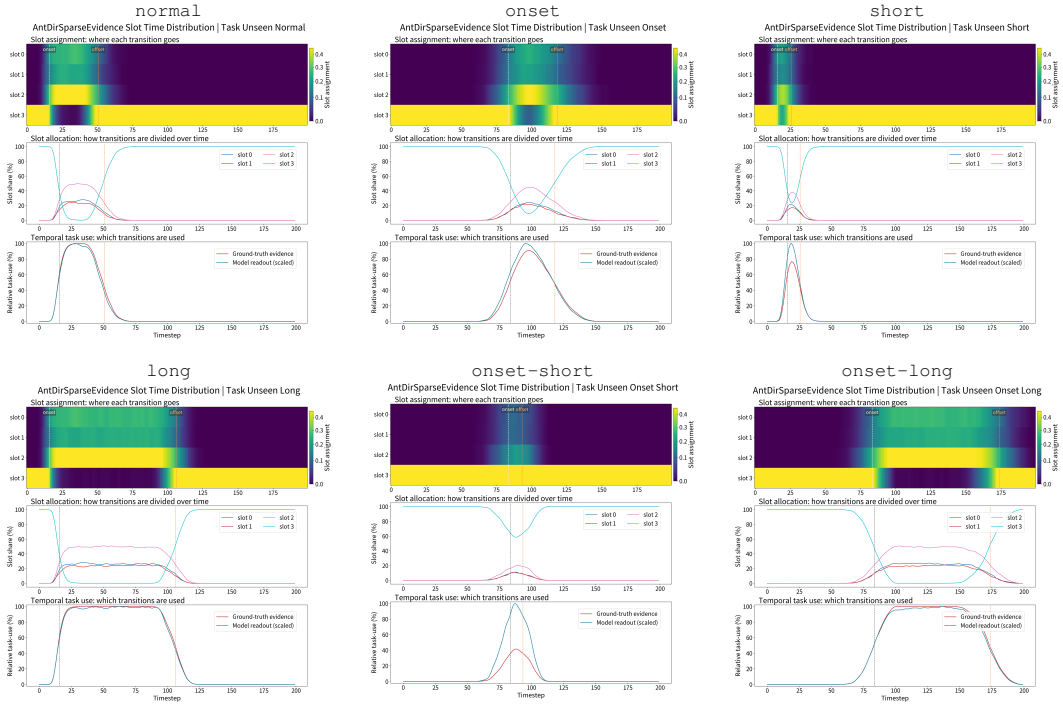

Figure 8: ADSE, unseen tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

not receive time indices, evidence masks, or interval boundaries during representation learning. Slot indices are local to each panel and should not be compared across conditions or environments.

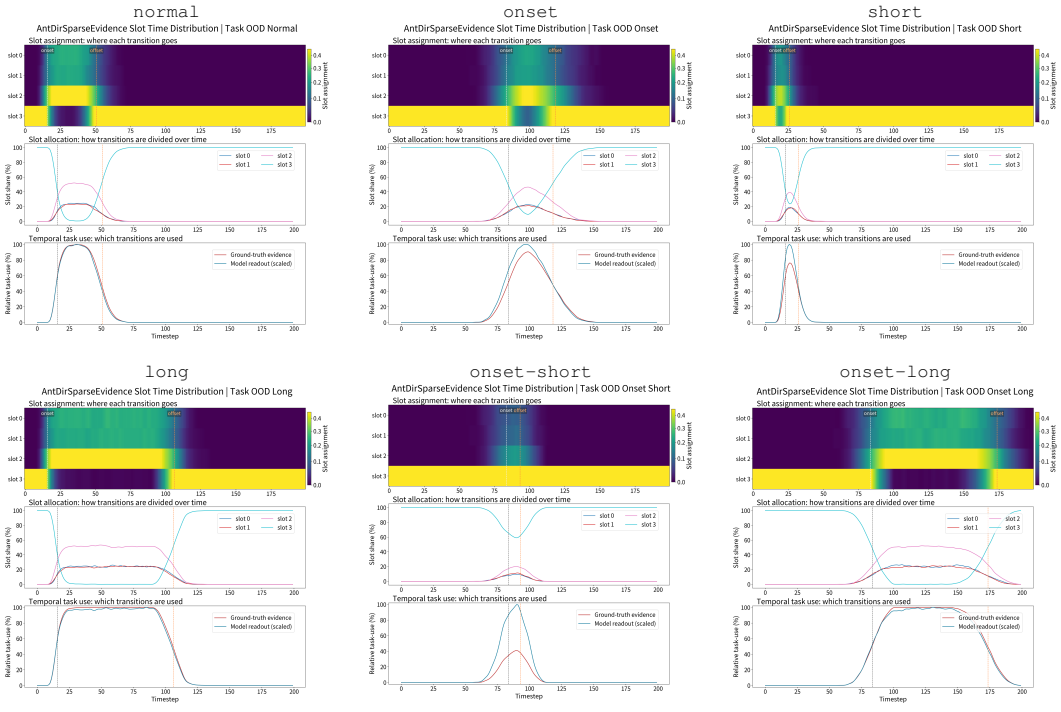

Figure 9: ADSE, OOD tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

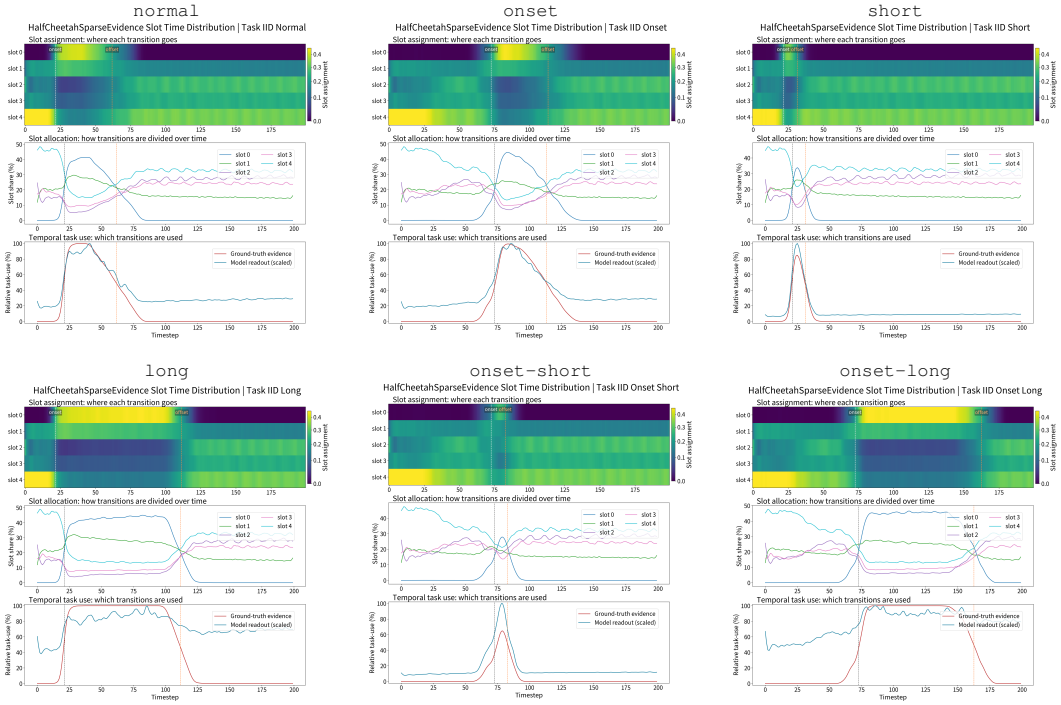

Figure 10: HCSE, IID tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

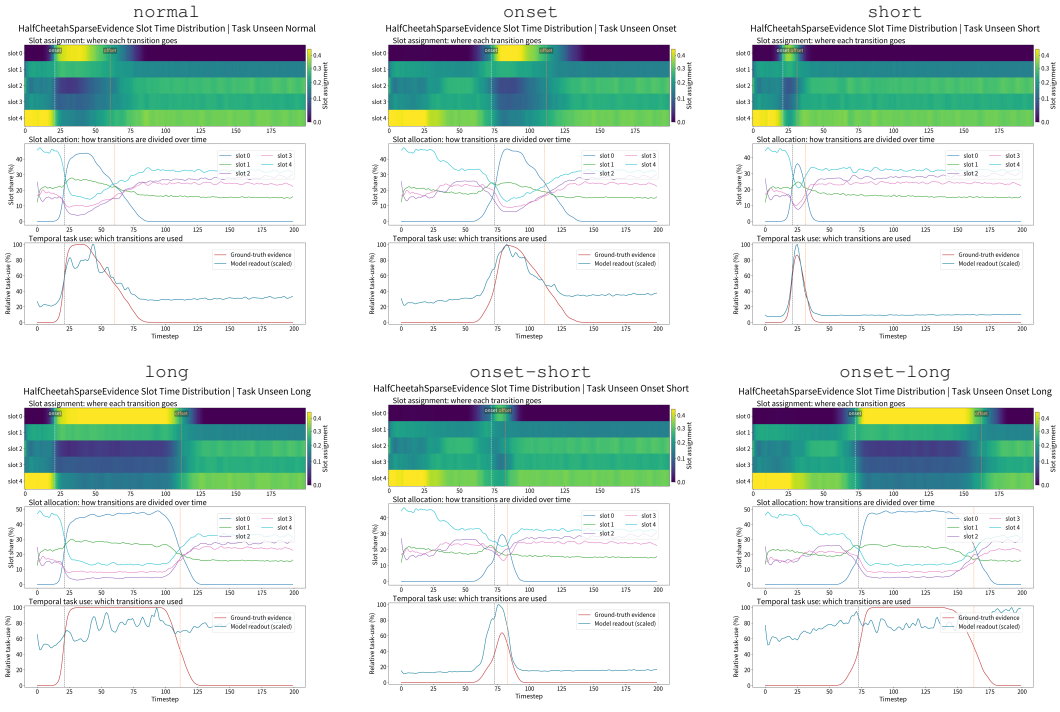

Figure 11: HCSE, unseen tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

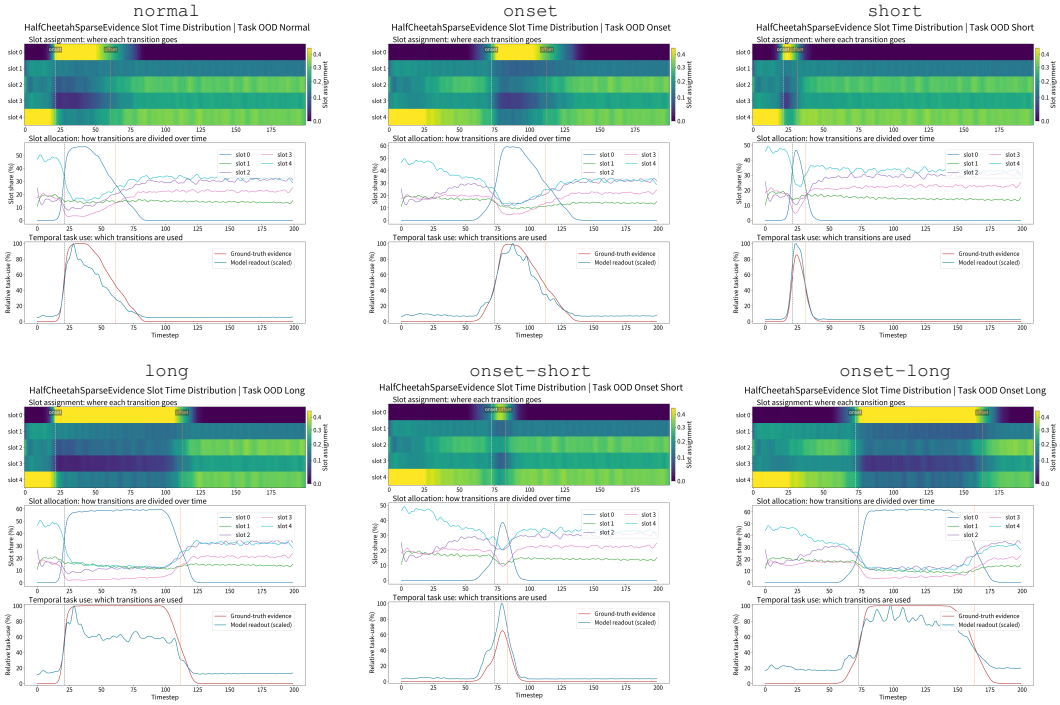

Figure 12: HCSE, OOD tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

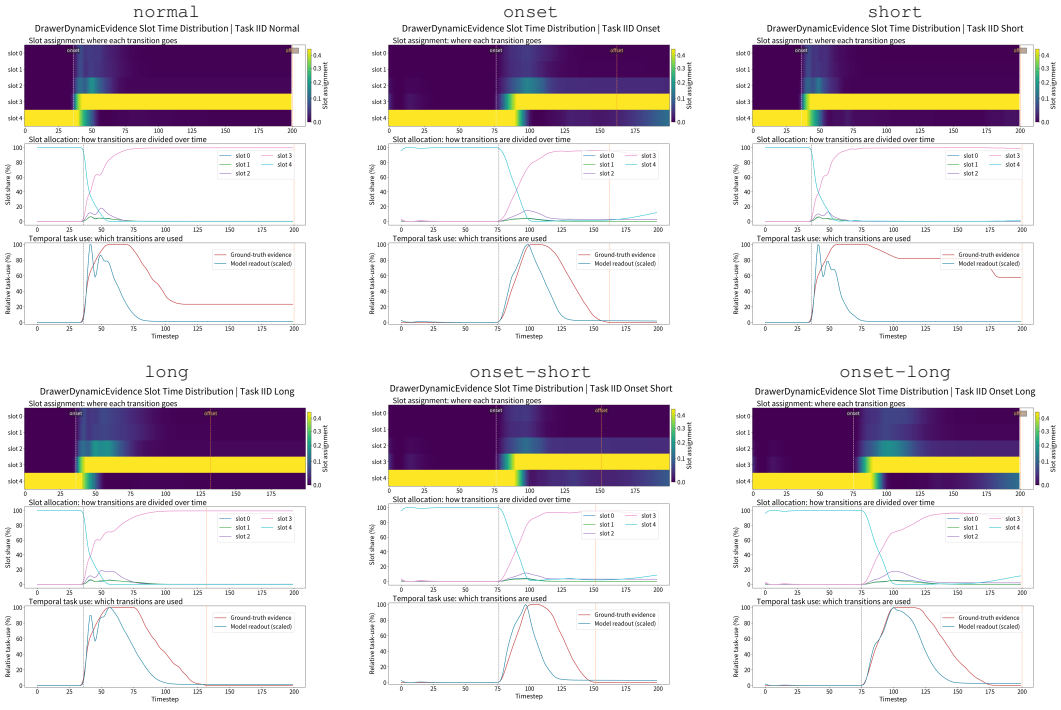

Figure 13: DDE, IID tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

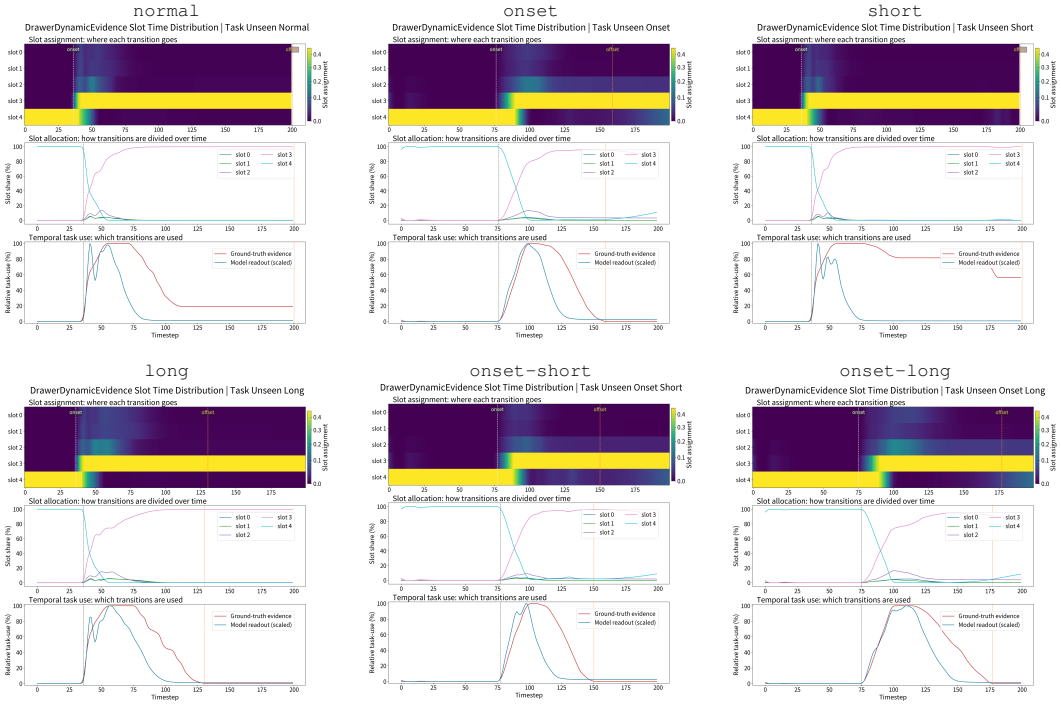

Figure 14: DDE, unseen tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.

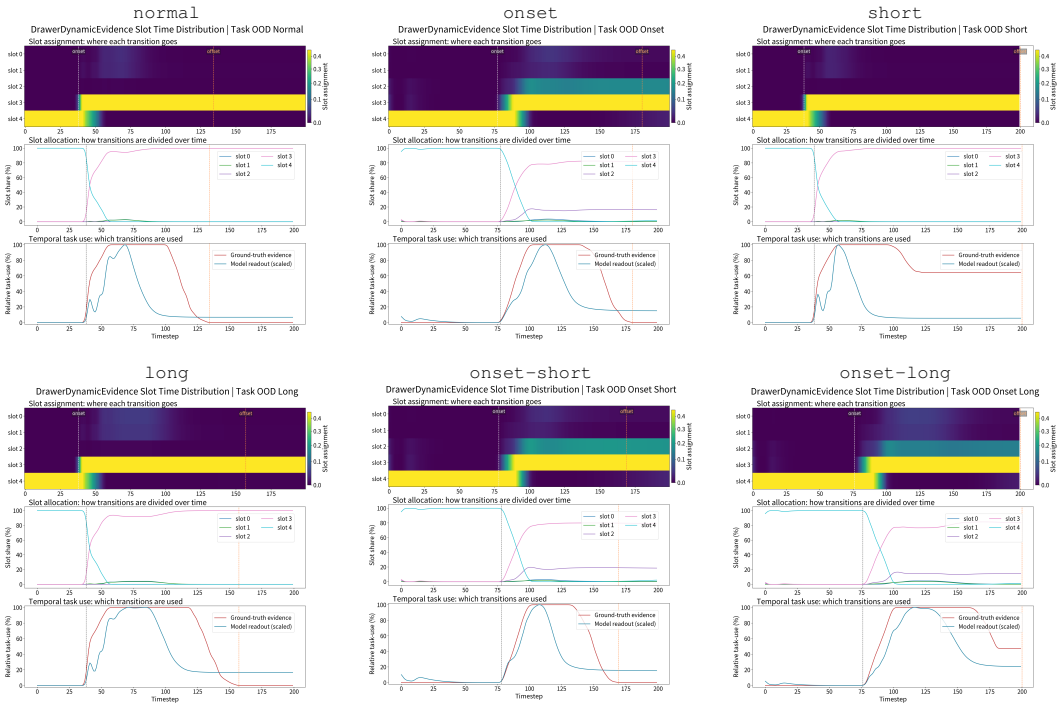

Figure 15: DDE, OOD tasks. Each panel shows slot assignment, slot allocation, and the task-use diagnostic for one evidence condition.