TASK-IDENTIFIABILITY-AWARE ACTION SUPERVISION FOR OFFLINE IN-CONTEXT RL

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context reinforcement learning (ICRL) enables policies to adapt to unseen tasks through accumulated interaction context without test-time parameter updates. However, existing approaches typically apply uniform action supervision across context prefixes, overlooking differences in how reliably each prefix identifies the underlying task. A short or ambiguous prefix may remain consistent with multiple tasks that require different actions for the same query state, such that the target action cannot always be uniquely inferred from the available context. To address this mismatch, we propose TIARA (**T**ask-**I**dentifiability-**A**ware **R**epresentation and **A**ction Learning), an offline ICRL framework that makes task inference explicit and uses empirical task-identification reliability to guide action learning. TIARA employs a causal context encoder, independent of the query state, to construct prefix-wise task representations supervised by training-task identities. The correctness of auxiliary task identification serves as an empirical proxy for prefix-level task identifiability and is used to reweight action supervision, assigning greater weight to correctly identified prefixes while retaining a smaller positive weight for ambiguous ones. Experiments across diverse multi-task offline ICRL benchmarks show that TIARA improves generalization to unseen tasks over strong baselines across most settings.

1 INTRODUCTION

Reinforcement learning (RL) has achieved remarkable successes across a wide range of domains, from mastering complex strategic games (Silver et al., 2016; Vinyals et al., 2019) to large-scale robotic manipulation and real-world control (Kalashnikov et al., 2018; Brohan et al., 2022). Despite these advances, learned policies often remain specialized to the training environments, while robust generalization to diverse or previously unseen tasks remains a longstanding challenge (Kirk et al., 2023). To address this limitation, recent work has explored *in-context reinforcement learning* (ICRL) (Laskin et al., 2023; Lee et al., 2023), extending the in-context learning paradigm of large language models (LLMs) (Brown et al., 2020) to sequential decision-making. ICRL conditions a Transformer-based (Vaswani et al., 2017) policy on a sequence of past interaction transitions, allowing the policy to infer task-relevant information from its interaction history. At test time, the model parameters remain fixed, and adaptation to unseen tasks is achieved solely through the evolving interaction context. As the agent interacts with the environment and accumulates transitions in its context window, the policy progressively refines its task understanding and adjusts its action predictions accordingly, enabling rapid context-based adaptation.

Within offline ICRL, the Decision Pre-trained Transformer (DPT) (Lee et al., 2023) provides a representative formulation based on supervised action prediction. DPT uses a single causal Transformer to jointly process the interaction context and the current query state, predicting the target action for that query. During training, DPT supervises action prediction at every context prefix using the same query-action target. Yet, these prefixes can differ substantially in how much task-relevant information they provide. A short or ambiguous prefix may remain consistent with multiple task hypotheses, leaving the underlying task unresolved. For the same query state, these compatible tasks may require different target actions. Consequently, the target action cannot always be uniquely inferred from the available context alone (Dorfman et al., 2021). However, as illustrated in Figure 1, DPT weights the action loss from every prefix equally, even though some prefixes contain enough task information to infer the target action whereas others do not. This mismatch is further compounded by the DPT architecture, where a single Transformer *jointly* performs task inference and action prediction. As a

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.

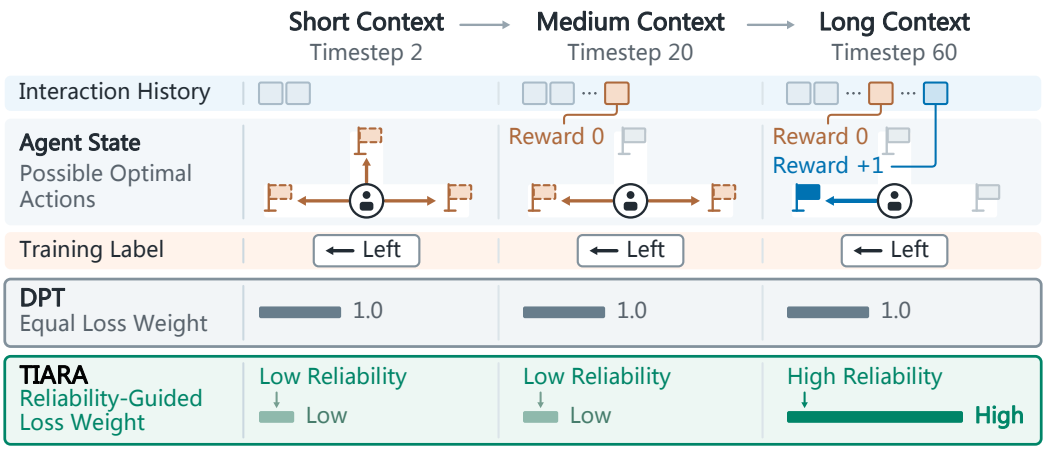


Figure 1: Task ambiguity across interaction histories and reliability-guided supervision. As interaction unfolds from timestep 2 to 60, sparse rewards progressively resolve task ambiguity, narrowing plausible goals from three to one. While standard DPT uniformly weights action supervision across all prefixes regardless of contextual ambiguity, TIARA uses task-prediction correctness as a reliability proxy to dynamically downweight ambiguous early prefixes and prioritize identifiable later ones.

result, DPT provides no dedicated representation for explicitly assessing how reliably each context prefix identifies the underlying task.

To address this mismatch between task identifiability and action supervision, we propose TIARA (**T**ask-**I**dentifiability-**A**ware **R**epresentation and **A**ction Learning), an offline ICRL framework that makes task inference explicit and uses empirical task-identification reliability to guide action learning. TIARA decouples task inference from query-conditioned action prediction by encoding the interaction history independently of the query state, yielding a dedicated representation of the task information available in each context prefix. An auxiliary task-identification objective supervises these prefix representations, encouraging them to capture task-discriminative information. The resulting task-identification signal provides an empirical proxy for how reliably each context prefix identifies the underlying task. TIARA then uses this reliability signal to reweight action supervision, assigning greater weight to prefixes whose task identities are correctly identified. For ambiguous prefixes, TIARA reduces the corresponding action-supervision weight because the available context may be insufficient to uniquely infer the target action. This adaptively aligns the strength of action supervision with how reliably each context prefix identifies the underlying task. We evaluate TIARA across diverse multi-task offline ICRL benchmarks against strong in-context and offline RL baselines. TIARA consistently improves generalization to unseen tasks. In summary, our main contributions are threefold:

- We introduce an explicit task-supervised context representation into offline ICRL, utilizing a decoupled causal encoder and an auxiliary classification objective to make task-relevant evidence transparent at every prefix.
- We design a reliability-guided action-supervision reweighting mechanism that dynamically adjusts action supervision based on prefix-level task identifiability, effectively shielding the policy from the harmful guidance of ambiguous early interactions.
- Through extensive evaluations across diverse multi-task offline ICRL benchmarks, we demonstrate that TIARA achieves superior generalization performance across most settings.

2 PRELIMINARIES

Problem Setup. We consider offline in-context reinforcement learning over a family of tasks modeled as Markov decision processes $\mathcal{M}_z = \langle \mathcal{S}, \mathcal{A}, T_z, R_z, \gamma \rangle$, where z indexes the task and γ is the discount factor. Tasks within an environment share the state space $\mathcal{S}$ and action space $\mathcal{A}$ but may differ in their transition dynamics T_z or reward functions R_z. The offline training data comprise trajectories from N tasks. Each training example consists of an interaction context $C = (x_1, \ldots, x_H)$ with context horizon H, a query state s_q, and a target action a_q. Each transition $x_i = (s_i, a_i, s'_i, r_i)$

contains the current state, action, next state, and reward. The context and query–action pair (s_q, a_q) are sampled from the same task, whose identity $z \in \{1, \ldots, N\}$ is available during training. We denote the context prefix containing the first i transitions by $C_{\leq i} = (x_1, \ldots, x_i)$. At test time, the policy adapts to unseen tasks solely through the interaction context and current query state, while its parameters remain fixed and task identities are unavailable.

Decision-Pretrained Transformer. As one of the representative approaches to ICRL, DPT (Lee et al., 2023) learns to predict a target action at a query state from interactions collected from the same task. It embeds each transition as a single token and prepends a query-state token to the interaction context. A causal Transformer then processes the resulting sequence. At context position i, DPT predicts the action as $\hat{a}_q^i = f_\theta(C_{\leq i}, s_q)$, conditioned on the query state s_q and context prefix $C_{\leq i}$. All prefix-conditioned predictions are computed in a single forward pass and supervised using the same target action a_q. Under this training scheme, DPT assigns equal weight to the action loss at every context prefix:

$$\mathcal{L}_{\mathrm{DPT}} = \sum_{b=1}^{B} \sum_{i=1}^{H} \ell_a(f_\theta(C_{\leq i,b}, (s_q)_b), (a_q)_b)\,, \tag{1}$$

where B denotes the batch size and ℓ_a is the action prediction loss. In our setting, we use cross-entropy (CE) (Heaton, 2018) for discrete actions and mean squared error (MSE) (Heaton, 2018) for continuous actions. For each training example, the query state and target action remain fixed across prefixes, while the available interaction context varies. Different prefixes may therefore differ substantially in how well they identify the underlying task, despite receiving equal action supervision. This creates a mismatch between the task identifiability of each context prefix and the strength of action supervision it receives.

3 METHODOLOGY

To address the mismatch between task identifiability and action supervision, we present TIARA (**T**ask-**I**dentifiability-**A**ware **R**epresentation and **A**ction Learning), an offline ICRL framework that makes task inference explicit and uses task-identification reliability to guide action supervision. TIARA decouples task inference from query-conditioned action prediction through *a causal context encoder* and *a separate action decoder*. The context encoder constructs prefix representations without access to the query state, supervised by an auxiliary task-identification objective, while the decoder combines these representations with the query state to predict actions. Task-identification correctness then serves as a prefix-level reliability signal for reweighting action supervision. Figure 2 illustrates the overall framework.

3.1 TASK-SUPERVISED CONTEXT REPRESENTATION

Causal Context Encoding. For each context prefix $C_{\leq i}$, the context encoder produces a representation h_i from the first i transitions $(x_1, \ldots, x_i)$. Each transition $x_i = (s_i, a_i, s_i', r_i)$ is first mapped to a d-dimensional token embedding:

$$u_i = W_x[s_i; a_i; s_i'; r_i] + p_i, \tag{2}$$

where $[\cdot;\cdot]$ denotes vector concatenation and $W_x \in \mathbb{R}^{d \times (2d_s + d_a + 1)}$ is a learned projection matrix. $p_i \in \mathbb{R}^d$ denotes the learned positional embedding at position i, while d_s and d_a denote the state and action dimensions, respectively. Notably, the next state s_i' and reward r_i expose task-specific signals from the transition dynamics and reward function, helping distinguish tasks that share the same state and action spaces. The resulting token sequence is then processed by an L_e-layer causal Transformer (Vaswani et al., 2017):

$$\begin{aligned} h_{1:H}^{(0)} &= u_{1:H}, \\ h_{1:H}^{(\ell)} &= f_\theta^{(\ell)}\left(h_{1:H}^{(\ell-1)}; M_{\mathrm{causal}}^{(H)}\right), \qquad \ell = 1, \ldots, L_e, \end{aligned} \tag{3}$$

where $M_{\mathrm{causal}}^{(H)}$ allows each position i to attend only to positions $j \leq i$. The final-layer output $h_i \equiv h_i^{(L_e)} \in \mathbb{R}^d$ therefore represents the context prefix $C_{\leq i}$ using only the information available up to position i. These prefix-wise representations form the basis for the subsequent task-identification objective and reliability estimation.

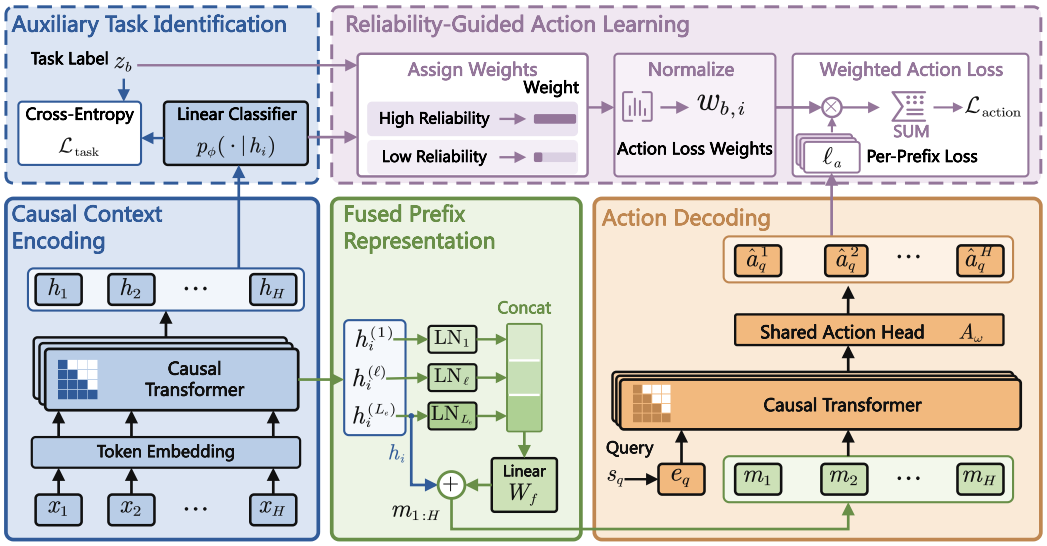


Figure 2: Overview of the TIARA framework. A causal context encoder processes interaction history independently of the query state to yield prefix representations h_i. An auxiliary task-identification objective $\mathcal{L}_{\text{task}}$ supervises these representations using ground-truth task labels, and the resulting classification correctness provides a reliability signal for computing normalized action-supervision weights $w_{b,i}$ that emphasize identifiable prefixes. Meanwhile, an action decoder combines multi-layer encoder features into fused prefix representations m_i and integrates them with the embedded query state e_q to predict prefix-conditioned actions $\hat{a}_q^i$ under the weighted action loss $\mathcal{L}_{\text{action}}$. The task classifier and reweighting scheme are used only during training.

Auxiliary Task Identification. To explicitly assess the task information captured by each context prefix, *a shared linear classifier* p_ϕ is applied to each prefix representation h_i to predict its training-task identity:

$$p_\phi(\cdot \mid h_i) = \text{softmax}(W_\phi h_i + \beta_\phi), \tag{4}$$

where $p_\phi(z \mid h_i)$ denotes the predicted probability that prefix representation h_i belongs to task $z \in \{1, \ldots, N\}$. $W_\phi \in \mathbb{R}^{N \times d}$ and $\beta_\phi \in \mathbb{R}^N$ denote the weight matrix and bias vector of the task-identification classifier, respectively, with $\phi = \{W_\phi, \beta_\phi\}$. The linear classifier keeps the auxiliary head lightweight while encouraging task identity to be directly encoded in the learned representation. To train this classifier across context prefixes, task identification is supervised using the cross-entropy loss:

$$\mathcal{L}_{\text{task}} = -\sum_{b=1}^{B} \sum_{i=1}^{H} \log p_\phi(z_b \mid h_{b,i}), \tag{5}$$

where $h_{b,i}$ denotes the representation of the i-th context prefix for sample b in the batch, and z_b denotes the corresponding task identity. This auxiliary objective encourages the context encoder to capture task-discriminative information across context prefixes. More importantly, whether the task identity is correctly predicted at each prefix provides a practical reliability signal for the subsequent action-supervision reweighting described in Section 3.3.

3.2 QUERY-CONDITIONED ACTION PREDICTION

Fused Prefix Representation. While the context encoder captures task-relevant information from the interaction history, action prediction additionally depends on the current query state s_q. To combine these two sources of information, TIARA employs a query-conditioned action decoder. To expose multi-level encoder features to action prediction, each context representation h_i is augmented with normalized hidden states from all encoder layers:

$$m_i = h_i + W_f \left[\text{LN}_1\left(h_i^{(1)}\right); \cdots; \text{LN}_{L_e}\left(h_i^{(L_e)}\right) \right], \tag{6}$$

where $\mathrm{LN}_\ell(\cdot)$ denotes layer normalization (Ba et al., 2016) applied to the hidden state from encoder layer ℓ, and $W_f \in \mathbb{R}^{d \times L_e d}$ is a learned projection matrix. The projection matrix W_f is initialized to zero such that $m_i = h_i$ at the beginning of training, allowing the fusion branch to be introduced gradually during optimization. While the final-layer representation h_i is directly shaped by task-identification supervision, intermediate encoder layers may retain complementary information useful for action prediction. The fused representation m_i therefore combines these multi-layer encoder features while preserving h_i through a residual connection.

Action Decoding. To condition action prediction on the current query state, s_q is mapped to a d-dimensional embedding $e_q = E_q(s_q) \in \mathbb{R}^d$ through a learned linear projection E_q. The resulting query embedding e_q is then prepended to the sequence of fused prefix representations $(m_1, \ldots, m_H)$. An L_d-layer causal self-attention decoder g_ψ then processes the resulting sequence:

$$(o_0, o_1, \ldots, o_H) = g_\psi\left([e_q, m_1, \ldots, m_H]; M_{\text{causal}}^{(H+1)}\right), \tag{7}$$

where the causal mask ensures that the output o_i depends only on the query token e_q and fused prefix representations $m_{1:i}$. *A shared policy head* produces the action prediction at each context position:

$$\hat{a}_q^i = A_\omega(o_i), \qquad i = 1, \ldots, H. \tag{8}$$

Accordingly, each prediction $\hat{a}_q^i$ is conditioned on the query state s_q and the corresponding context prefix $C_{\leq i}$. All H prefix-conditioned predictions are computed in a single forward pass, introducing only limited additional computational overhead. Based on these predictions, the per-prefix action loss is defined following common ICRL formulations as $\ell_a(\hat{a}_q^i, a_q)$, using mean squared error for continuous actions and cross-entropy for discrete actions. To account for differences in task identifiability across context prefixes, these per-prefix losses are subsequently reweighted according to task-identification reliability.

3.3 Reliability-Guided Action Learning

Different context prefixes can vary substantially in how reliably they identify the underlying task. For an ambiguous prefix, multiple tasks may remain compatible with the available interaction history while requiring different actions at the same query state. Uniform action supervision therefore overlooks whether the current prefix contains sufficient information to infer the target action. To address this mismatch, TIARA uses task-identification correctness as a prefix-level reliability signal and adjusts the strength of action supervision accordingly.

For each training sample b and context prefix i, task-identification correctness is defined as

$$\rho_{b,i} = \mathbf{1}\left[\arg\max_z p_\phi(z \mid h_{b,i}) = z_b\right] \in \{0, 1\}, \tag{9}$$

where z_b denotes the task identity of sample b. A correct prediction indicates that the current prefix representation contains sufficient task-discriminative information for the classifier to identify its task, whereas an incorrect prediction suggests lower task-identification reliability. Accordingly, $\rho_{b,i}$ serves as a practical indicator of prefix-level task-identification reliability. Based on this reliability signal, correctly identified prefixes receive a raw action-supervision weight of 1, while incorrectly identified prefixes receive a smaller positive weight α:

$$\widetilde{w}_{b,i} = \alpha + (1 - \alpha)\rho_{b,i}, \tag{10}$$

where $\alpha \in (0, 1]$ controls the minimum strength of action supervision and is set to 0.2 by default. The positive lower bound α retains action supervision for prefixes whose task identities are not yet correctly identified, rather than discarding their training signals entirely. To retain a comparable overall loss scale after reweighting, the raw weights are normalized within each training sample to have an average of 1:

$$w_{b,i} = \frac{\widetilde{w}_{b,i}}{\frac{1}{H}\sum_{j=1}^{H} \widetilde{w}_{b,j}}, \qquad \sum_{i=1}^{H} w_{b,i} = H. \tag{11}$$

This normalization keeps the total action-supervision weight identical to that under uniform weighting. For a given sequence in the minibatch, if its prefixes include both correctly and incorrectly identified ones, the former receive weights above 1, while the latter receive weights below 1. If all prefixes

are either correctly or incorrectly identified, their normalized weights all become 1. When $\alpha = 1$, all prefixes receive equal weight, and the weighting scheme reduces to standard uniform action supervision.

After assigning reliability-guided weights to the per-prefix action losses, the final training objective combines the resulting action loss with the auxiliary task-identification loss:

$$\mathcal{L}_{\text{action}} = \sum_{b=1}^{B} \sum_{i=1}^{H} w_{b,i}\, \ell_a\big((\hat{a}_q^{\,i})_b, (a_q)_b\big)\,, \qquad \mathcal{L} = \mathcal{L}_{\text{action}} + \lambda \mathcal{L}_{\text{task}}. \tag{12}$$

where $\mathcal{L}_{\text{task}}$ is defined in Eq. (5), and λ controls its contribution to the joint objective and is set to 1 by default. The task-identification loss encourages the context encoder to learn task-discriminative representations, while the reliability weights adjust the contribution of each context prefix to action learning. Both the task-identification classifier and reliability-guided reweighting are used only during training. At test time, neither task identities nor the auxiliary classifier are required, leaving the standard query-conditioned action-prediction pipeline unchanged.

TIARA extends the DPT-style offline ICRL paradigm by integrating explicit task identification with reliability-guided action supervision. The context encoder learns task-discriminative prefix representations, while task-identification correctness adjusts the contribution of each prefix to action learning. This design aligns action supervision with differences in task identifiability across context prefixes without introducing additional requirements at test time. Further implementation details and additional analyses are provided in Appendices B and C, respectively.

4 Related Work

Sequence Models for In-Context Reinforcement Learning. Sequence modeling provides an important foundation for in-context reinforcement learning. Decision Transformer formulates offline control as return-conditioned sequence modeling (Chen et al., 2021), while Trajectory Transformer autoregressively models trajectories and performs planning through beam search (Janner et al., 2021). Building on this sequence-modeling perspective, AD distills learning histories into a fixed Transformer that adapts its behavior through interaction context (Laskin et al., 2023). In contrast to AD, which learns from sequential learning histories, DPT directly supervises query-action prediction from interaction contexts, enabling in-context decision making through supervised pretraining (Lee et al., 2023). Subsequent studies improve ICRL from different perspectives. Noise distillation constructs improving interaction histories without repeatedly training source learners (Zisman et al., 2024), while ICEE explicitly targets in-context exploration and exploitation (Dai et al., 2024). IDT introduces hierarchical temporal abstraction into in-context decision making (Huang et al., 2024), whereas DICP distills predictive models to support in-context planning (Son et al., 2025). T2MIR enhances model capacity through token-wise and task-wise mixture-of-experts modules (Wu et al., 2025). Other approaches, including DIT, Q-learning-based offline ICRL, and S-ICQL, incorporate value-learning objectives to improve learning from suboptimal offline data (Dong et al., 2025; Tarasov et al., 2025; Liu et al., 2026). Different from these directions, TIARA retains DPT-style supervised query-action pretraining but explicitly models prefix-level task identifiability and uses it to adapt the strength of action supervision.

Context-Based Meta-Reinforcement Learning. Context-based meta-RL methods commonly infer task information from interaction histories and condition decision making on the resulting representations. PEARL infers a probabilistic latent task variable from transitions and conditions an off-policy actor-critic on this latent context (Rakelly et al., 2019). FOCAL extends context-based task representation learning to fully offline meta-RL through metric learning and behavior regularization (Li et al., 2021). TIARA similarly extracts task-relevant information from interaction context, but uses task inference for a different purpose. Rather than learning a posterior task variable for policy conditioning, TIARA uses training-task identities only to supervise prefix representations and derive task-identification reliability for action-supervision reweighting. The auxiliary task-identification head is removed at test time, and adaptation to unseen tasks relies solely on the interaction context and query state with fixed model parameters.

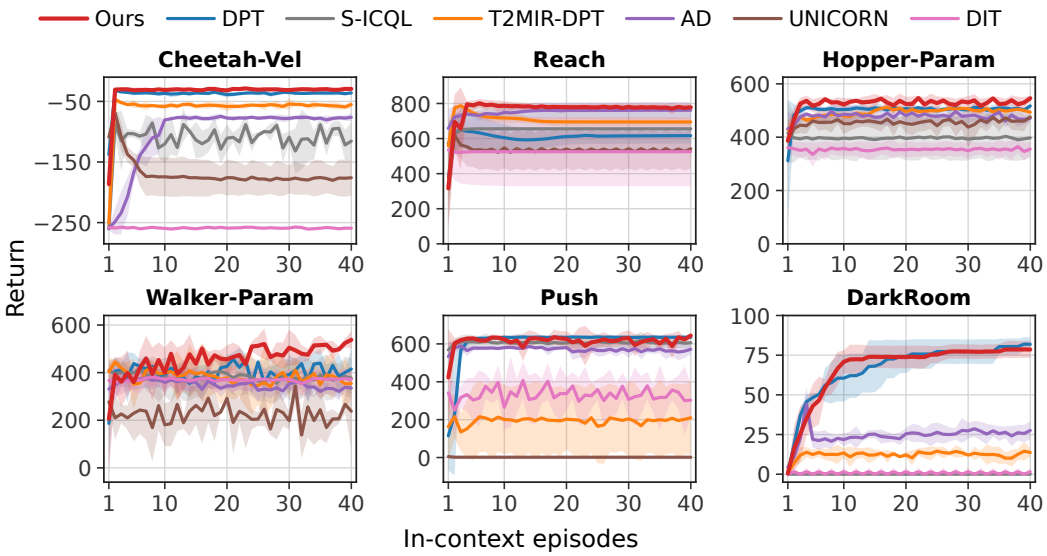


Figure 3: Test return curves of TIARA against baselines using Mixed datasets.

5 EXPERIMENTS

In this section, we empirically evaluate TIARA across six environments spanning discrete navigation, continuous locomotion control, and robotic manipulation. Our evaluation aims to answer three core questions: **(I)** *Does reliability-guided action learning and explicit task supervision consistently improve generalization over strong in-context and offline RL baselines?* **(II)** *What are the respective contributions of the decoupled encoder and auxiliary task objectives?* **(III)** *How does TIARA's representation geometry evolve with interaction context, and how do downstream action predictions leverage multi-layer encoder features?* To ensure a strictly fair comparison, all baseline and proposed models are trained on the exact same offline dataset within each environment.

5.1 EXPERIMENTAL SETUP

Environments and Datasets. We consider six multi-task environments encompassing discrete navigation, continuous locomotion control, and robotic manipulation, where tasks are defined by variations in reward functions or transition dynamics. Specifically, DARKROOM, REACH, and PUSH vary the target goal location, whereas CHEETAH-VEL varies the target velocity. HOPPER-PARAM and WALKER-PARAM introduces diverse physical variations governing the transition dynamics, including body mass, inertia, damping, and friction. We utilize the publicly released Mixed datasets from T2MIR (Wu et al., 2025) for DARKROOM, CHEETAH-VEL, REACH, PUSH, and WALKER-PARAM, and collect the HOPPER-PARAM dataset following the identical data-collection protocol. To ensure a strictly fair comparison, all baseline and proposed models are trained on the exact same offline dataset within each environment.

Baselines. We compare TIARA against a comprehensive suite of six state-of-the-art baselines. Specifically, we include **DPT** (Lee et al., 2023), a representative formulation that predicts actions conditioned on query states and interaction contexts via a single causal Transformer; **AD** (Laskin et al., 2023), which distills learning histories through autoregressive action prediction; **T2MIR-DPT** (Wu et al., 2025), which augments DPT with token-wise and task-wise mixture-of-experts layers. We further compare against value-based and meta-RL methods designed for suboptimal data, including **DIT** (Dong et al., 2025), **S-ICQL** (Liu et al., 2026), and **UNICORN** (Li et al., 2024), a context-based offline meta-RL method that learns task representations via an information-theoretic objective.

5.2 MAIN RESULTS

Figure 3 and Table 1 summarize the performance across six environments. TIARA achieves superior performance across most environments, demonstrating particular strength in scenarios demanding robust task inference under short or ambiguous interaction contexts (REACH, HOPPER-PARAM, and

Table 1: Evaluation returns across six multi-task environments.

Method	Cheetah-Vel	Hopper-Param	Reach	Walker-Param	Push	DarkRoom
TIARA (Ours)	**−29.41** ±1.86	**545.82** ±11.34	**778.37** ±18.57	**537.77** ±35.74	**643.94** ±14.16	78.53 ±5.23
DPT	−36.00 ±2.04	517.27 ±8.67	616.73 ±45.10	415.17 ±81.61	635.33 ±8.62	**81.87** ±4.23
S-ICQL	−115.53 ±30.54	397.34 ±72.16	655.70 ±134.73	377.32 ±82.42	606.29 ±28.81	0.00 ±0.00
T2MIR-DPT	−55.13 ±5.90	493.55 ±9.30	694.69 ±24.34	354.44 ±65.06	209.98 ±191.37	13.67 ±2.51
AD	−76.18 ±6.68	470.96 ±13.20	761.53 ±46.69	335.26 ±24.44	571.81 ±27.11	27.57 ±5.54
UNICORN	−175.98 ±29.36	475.98 ±12.25	541.72 ±111.82	237.86 ±156.35	1.59 ±0.51	—
DIT	−259.24 ±0.09	355.38 ±23.62	525.20 ±196.18	370.91 ±41.79	303.01 ±87.12	1.57 ±2.71

WALKER-PARAM). Specifically, in continuous control domains, TIARA rapidly ascends to high-return trajectories by leveraging reliability-guided action weighting, which shields the policy from the harmful guidance of ambiguous early prefixes. Furthermore, unlike baselines that plateau prematurely, TIARA exhibits a sustained upward trajectory, confirming our intuition that task understanding evolves progressively and action learning must pace itself accordingly. In stable regimes like CHEETAH-VEL and PUSH, TIARA quickly converges to top-tier levels without sacrificing baseline stability. On the discrete DARKROOM environment, standard DPT achieves a marginally higher return because sparse-reward grid navigation lacks task-discriminative information prior to goal discovery. Consequently, our reliability mechanism downweights these uninformative prefixes, which can be slightly conservative in simple discrete grid worlds where uniform imitation still suffices, and UNICORN is omitted here as its framework is tailored for continuous domains. Overall, these findings confirm that coupling explicit task-supervised representations with reliability-guided action supervision provides a powerful mechanism for robust offline ICRL.

5.3 ABLATION

To evaluate the contribution of each core component in TIARA, we conduct ablation studies on the REACH and WALKER-PARAM environments, comparing the complete model against two key variants: (1) **w/o Weight**, which removes the reliability-guided action reweighting to revert to uniform weighting across all context prefixes; and (2) **w/o Encoder**, which replaces our decoupled encoder-decoder architecture with a standard decoder-only model akin to DPT. Because the *w/o Encoder* variant lacks a dedicated encoder for explicit task identification, it cannot derive a principled task classification signal. Consequently, we replace the reliability-based weight with a heuristic position-based schedule that assigns progressively higher supervision weights to later interaction prefixes. As illustrated in Figure 4, the complete TIARA model outperforms both ablated variants on REACH. Removing the reliability-guided reweighting (**w/o Weight**) leads to a marked performance drop in this environment. The heuristic position-based weighting (**w/o Encoder**) improves over DPT but remains below TIARA on REACH and WALKER-PARAM. These results support the contribution of explicit task supervision on REACH.

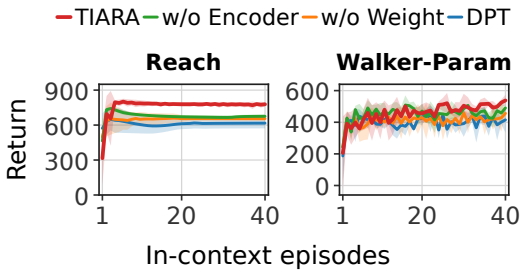


Figure 4: Results of the ablation study.

Table 2: Numerical results of ablation study.

Method	Reach	Walker-Param
TIARA	**778.37** ±18.57	**537.77** ±35.74
w/o Weight	654.95 ±19.97	456.15 ±18.86
w/o Encoder	686.70 ±15.50	490.26 ±9.82
DPT	616.73 ±45.10	415.17 ±81.61

5.4 RESULTS ON HIGH-DIMENSIONAL CONTROL

We further evaluate TIARA on the challenging HUMANOID-DIR benchmark to examine its scalability in high-dimensional continuous control. Figure 5 compares TIARA against six competitive baselines over 40 online evaluation episodes, and Table 3 summarizes their final returns at episode 40. As illustrated in Figure 5, TIARA adapts rapidly during the initial episodes and sustains superior mean

Table 3: Returns on Humanoid-Dir.

TIARA(Ours)	DPT	S-ICQL	T2MIR-DPT	AD	UNICORN	DIT
800.97 $\pm$172.50	742.00 $\pm$257.67	510.29 $\pm$143.97	697.97 $\pm$146.30	597.01 $\pm$21.24	799.25 $\pm$48.84	464.77 $\pm$188.83

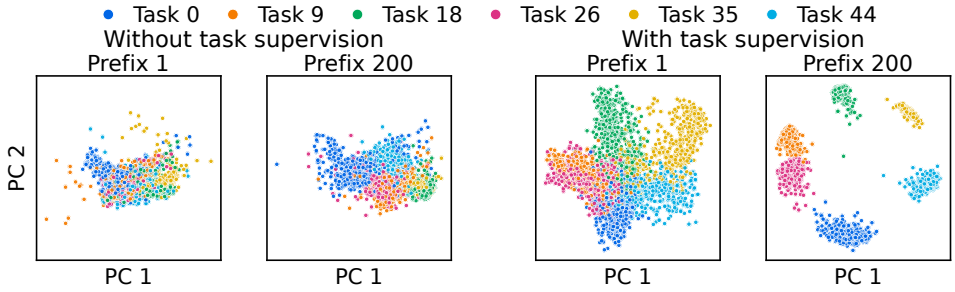


Figure 6: **PCA projections of encoder representations on REACH.** Comparison of task representations at prefix lengths 1 and 200 with and without auxiliary task supervision.

returns throughout the evaluation horizon. Compared with standard baselines such as DPT, T2MIR-DPT, and S-ICQL, TIARA maintains a clear performance advantage after the early adaptation phase, highlighting the benefits of our decoupled architecture and reliability-guided action supervision in complex state spaces. While UNICORN exhibits larger episodic fluctuations, its final return converges to a comparable level by the end of evaluation. Overall, this high-dimensional evaluation demonstrates that TIARA's core mechanism scales effectively, converting rich interaction histories into robust and stable policy adaptation in demanding control tasks.

5.5 VISUALIZATION

To qualitatively evaluate how auxiliary task supervision shapes the learned context representations, Figure 6 visualizes the prefix representations projected onto a two-dimensional principal component space (Abdi & Williams, 2010). This visualization corroborates our core motivation: at the initial stage, representations across different tasks heavily overlap, directly illustrating the intrinsic task ambiguity and information insufficiency of early interactions. As the context grows to 200 transitions, representations learned without auxiliary supervision remain largely entangled and indistinct. With the guidance of our auxiliary task objective, TIARA effectively resolves this ambiguity over time, organizing the accumulated history into compact and well-separated task-conditioned clusters that provide a solid geometric foundation for reliability-guided action learning.

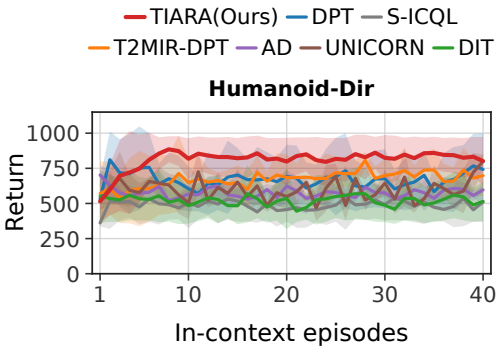


Figure 5: Results on Humanoid-Dir.

6 CONCLUSION

In this paper, we introduced TIARA, an offline ICRL framework that addresses the intrinsic mismatch between task evidence and action supervision in DPT-style architectures. Our core motivation stems from the insight that standard query-action supervision leaves the information content of partial interaction prefixes largely unconstrained. By incorporating dense, auxiliary task supervision alongside a decoupled causal encoder, TIARA makes task-relevant evidence explicitly inspectable at every prefix. Building upon this explicit task understanding, our reliability-guided action learning mechanism dynamically aligns supervision strength with prefix identifiability, ensuring that action learning paces itself according to the model's evolving competence. Meanwhile, a query-conditioned causal decoder seamlessly translates these fused prefix representations into action execution without requiring test-time task labels. Extensive evaluations across diverse multi-task benchmarks demonstrate that TIARA consistently improves generalization to unseen tasks.

AI USE STATEMENT

In preparing this draft, we used generative AI tools exclusively for editorial purposes, such as polishing the paper, enhancing readability, and refining grammar.

ETHICS STATEMENT

The reported experiments use simulated reinforcement-learning environments and existing offline benchmark data. They involve no human subjects, personally identifiable information, or sensitive user data.

REPRODUCIBILITY STATEMENT

The problem setup and method are described in Sections 2 and 3, respectively. Appendix B provides the training and online deployment procedures, model architecture, and training hyperparameters. Appendix D details the offline data organization, training and evaluation task splits, context–query sampling, and preprocessing.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL https://arxiv.org/abs/1607.06450.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Zhenwen Dai, Federico Tomasi, and Sina Ghiassian. In-context exploration-exploitation for reinforcement learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uIKZSStON3.

Juncheng Dong, Moyang Guo, Ethan X. Fang, Zhuoran Yang, and Vahid Tarokh. In-context reinforcement learning from suboptimal historical data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14021–14039, 2025. URL https://proceedings.mlr.press/v267/dong25d.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, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/248024541dbda1d3fd75fe49d1a4df4d-Abstract.html.

Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic programming and evolvable machines*, 19(1):305–307, 2018.

Sili Huang, Jifeng Hu, Hechang Chen, Lichao Sun, and Bo Yang. In-context decision transformer: Reinforcement learning via hierarchical chain-of-thought. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19871–19885, 2024. URL https://proceedings.mlr.press/v235/huang24j.html.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pp. 651–673. PMLR, 2018.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Jonathan N. Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1866. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8644b61a9bc87bf7844750a015feb600-Abstract-Conference.html.

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*, 2021. 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. doi: 10.52202/079017-2408. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8a30aba6514b56d02976f49797f6338a-Abstract-Conference.html.

Jinmei Liu, Fuhong Liu, Zhenhong Sun, Jianye Hao, Huaxiong Li, Bo Wang, Daoyi Dong, Chunlin Chen, and Zhi Wang. Scalable in-context Q-learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9e76870b97621b3153b84782062143ef-Abstract-Conference.html.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. 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.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Jaehyeon Son, Soochan Lee, and Gunhee Kim. Distilling reinforcement learning algorithms for in-context model-based planning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5c4a7643ab90cd62320df95c873a1c6f-Abstract-Conference.html.

Denis Tarasov, Alexander Nikulin, Ilya Zisman, Albina Klepach, Andrei Polubarov, Nikita Lyubaykin, Alexander Derevyagin, Igor Kiselev, and Vladislav Kurenkov. Yes, Q-learning helps offline in-context RL. *arXiv preprint arXiv:2502.17666*, 2025. URL https://arxiv.org/abs/2502.17666.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

Wenhao Wu, Fuhong Liu, Haoru Li, Zican Hu, Daoyi Dong, Chunlin Chen, and Zhi Wang. Mixture-of-experts meets in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0832. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/233cbc888ef01917d8d895e471e80a01-Abstract-Conference.html.

Ilya Zisman, Vladislav Kurenkov, Alexander Nikulin, Viacheslav Sinii, and Sergey Kolesnikov. Emergence of in-context reinforcement learning from noise distillation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62832–62846, 2024. URL https://proceedings.mlr.press/v235/zisman24a.html.

A NOTATION

Table 4 summarizes the notation used in the paper.

Table 4: Main notation.

Symbol	Description
$\mathcal{M}_z$	Task MDP $\langle \mathcal{S}, \mathcal{A}, T_z, R_z, \gamma \rangle$.
$\mathcal{S}, \mathcal{A}, T_z, R_z, \gamma$	State and action spaces, task dynamics and reward, discount factor.
z, N	Training-task index and number of training tasks.
$x_i = (s_i, a_i, s'_i, r_i)$	Context transition: state, action, next state, reward.
$C, C_{\le i}, H$	Context, its first i transitions, and context length.
(s_q, a_q)	Query state and target action.
B, b, i, j	Minibatch size, sample index, and context-position indices.
d_s, d_a, d	State, action, and hidden dimensions.
$f_\theta(C_{\le i}, s_q)$	DPT prefix-action predictor.
$\mathcal{L}_{\mathrm{DPT}}$	Uniform DPT prefix-action objective.
W_x, p_i, u_i	Transition projection, position embedding, and input token.
$L_e, f_\theta^{(\ell)}$	Encoder depth and layer-ℓ transformation.
$h_i^{(\ell)}, h_i$	Layer-ℓ hidden state and final prefix representation.
$p_\phi(z \mid h_i)$	Task classifier and predicted task probability;
W_ϕ, β_ϕ, ϕ	Task-head weights, bias, and parameter set $\phi = \{W_\phi, \beta_\phi\}$.
$W_f, W_\ell, \mathrm{LN}_\ell, m_i$	Fusion projection, layer block, normalization, and fused representation.
E_q, e_q	Query-state projection and embedding.
L_d, g_ψ, o_i	Decoder depth, decoder, and output at position i.
$A_\omega, \hat{a}_q^i$	Action head and prefix-conditioned action prediction.
$M_{\mathrm{causal}}^{(H)}, M_{\mathrm{causal}}^{(H+1)}$	Encoder and decoder causal masks.
$\rho_{b,i}$	Correct task-classification indicator.
$\widetilde{w}_{b,i}, w_{b,i}, \alpha$	Raw and normalized prefix weights; base weight for an incorrect classification.
$\ell_a, \mathcal{L}_{\mathrm{action}}, \mathcal{L}_{\mathrm{task}}$	Per-prefix action loss, weighted action objective, and task loss.
$\lambda, \mathcal{L}$	Task-loss coefficient and joint objective.
p_b, D_b	Correct-prefix fraction and weight-normalization denominator.
L, N_c, N_q	Data-block length and per-task context and query counts.
$\mathcal{D}_z^{\mathrm{ctx}}, \mathcal{D}_z^{\mathrm{qry}}$	Task-specific context-transition and query-pair buffers.
v	Random exponent for Hopper parameter sampling.
$N_s, s^{(n)}, \mu, \sigma, \widetilde{s}$	Training context-state count, state row, mean, scale, and normalized state.
$M, m_\emptyset$	Fused representation sequence and empty-context input at deployment.
$\tau_e, \mathrm{Recent}_H, \Vert$	Episode transitions, recent-H operator, and sequence concatenation.
$s_t, \hat{a}_t, a_t, s_{t+1}, r_t$	Current state, policy output, executed action, next state, and reward.
K, T, E	Numbers of training seeds, evaluation tasks, and online episodes.
$R_{k,t,e}, \bar{R}_{k,e}, \mu_e, s_e$	Episode return, task average per seed, cross-seed mean, and sample standard deviation; here t indexes tasks.
ΔMSE	Intervention MSE minus full-fusion MSE.

B IMPLEMENTATION DETAILS

This section specifies the training and deployment procedures and the model settings.

B.1 Training Algorithm

Algorithm 1 computes the training-only prefix weights and task loss. Each context prefix shares a query pair; task labels are not input tokens.

Algorithm 1 TIARA offline meta-training

Require: Context buffers $\{\mathcal{D}_z^{\mathrm{ctx}}\}$, query buffers $\{\mathcal{D}_z^{\mathrm{qry}}\}$, context length H, batch size B, $\alpha \in (0, 1]$, λ

while the training budget is not exhausted **do**
  Sample $\{z_b, C_b, ((s_q)_b, (a_q)_b)\}_{b=1}^{B}$ from task-matched buffers
  Form $u_{b,1:H}$ from C_b by Eq. (2)
  Compute $h_{b,1:H}^{(1:L_e)}$ by Eq. (3)
  Compute $m_{b,1:H}$ by Eq. (6) at each position
  $o_{b,0:H} \leftarrow g_\psi([E_q((s_q)_b), m_{b,1:H}]; M_{\mathrm{causal}}^{(H+1)})$
  $(\hat{a}_q^i)_b \leftarrow A_\omega(o_{b,i})$ for $i = 1, \ldots, H$
  $\rho_{b,i} \leftarrow \mathbf{1}[\arg\max_z p_\phi(z \mid h_{b,i}) = z_b]$
  $\widetilde{w}_{b,i} \leftarrow \alpha + (1 - \alpha)\rho_{b,i}$
  $w_{b,i} \leftarrow \widetilde{w}_{b,i} / (H^{-1} \sum_{j=1}^{H} \widetilde{w}_{b,j})$
  $\mathcal{L}_{\mathrm{action}} \leftarrow \sum_{b,i} w_{b,i} \ell_a((\hat{a}_q^i)_b, (a_q)_b)$
  $\mathcal{L}_{\mathrm{task}} \leftarrow -\sum_{b,i} \log p_\phi(z_b \mid h_{b,i})$
  Compute gradients of $\mathcal{L}_{\mathrm{action}} + \lambda \mathcal{L}_{\mathrm{task}}$
  Apply global gradient clipping and update all trainable parameters
end while

B.2 Task-ID-Free Online Deployment

Algorithm 2 updates the retained context after each episode and resets it between tasks. For nonempty context, it uses Eqs. (2), (3), and (6); parameters remain frozen and the task classifier is discarded.

Algorithm 2 TIARA deployment on one unseen task

Require: Frozen encoder, fusion, query embedding, decoder, and policy head; budgets H, E

Discard the task classifier; initialize retained context $C \leftarrow \varnothing$
for episode $e = 1, \ldots, E$ **do**
  Reset the environment on the same task; initialize $\tau_e \leftarrow \varnothing$
  for each environment step with current state s_t **do**
    if $|C| > 0$ **then**
      Form $u_{1:|C|}$, $h_{1:|C|}^{(1:L_e)}$, and $m_{1:|C|}$ from C
      $M \leftarrow (m_1, \ldots, m_{|C|})$
    else
      $M \leftarrow (m_\emptyset)$
    end if
    $(o_0, \ldots, o_{|M|}) \leftarrow g_\psi([E_q(s_t), M]; M_{\mathrm{causal}}^{(|M|+1)})$
    $\hat{a}_t \leftarrow A_\omega(o_{|M|})$; obtain a_t from $\hat{a}_t$ for the action space
    Execute a_t; append (s_t, a_t, s_{t+1}, r_t) to τ_e
  end for
  $C \leftarrow \mathrm{Recent}_H(C \,\|\, \tau_e)$
end for

Recent_H retains at most the last H complete transitions, and $m_\emptyset$ is the empty-context input. The continuous head produces an action vector; discrete scores are converted to an action by the evaluation rule. Context C contains observed actions, not the current query's target action.

B.3 Architecture and Training Hyperparameters

Tables 5 and 6 report the model settings and environment-specific training budgets. Baseline configurations are method specific.

Table 5: TIARA architecture and optimization settings.

Component	Setting
Context encoder / action decoder	$L_e = 4$ / $L_d = 4$ causal layers
Hidden width / attention heads	$d = 32$ / 4 heads
Feed-forward hidden width	$128 = 4d$ in both encoder and decoder
Dropout	0
Transition input	$[s_i; a_i; s'_i; r_i] \in \mathbb{R}^{2d_s+d_a+1}$
Position encoding	Learned transition-position embeddings
Task head	Linear $d \to N$, followed by softmax
Fusion projection	$W_f \in \mathbb{R}^{d \times L_e d}$, zero initialization
Query input	State-only embedding $E_q(s_q) \in \mathbb{R}^d$
Action loss	Continuous: MSE; discrete: cross-entropy
Optimizer	AdamW
Learning rate	10^{-3}, constant; no warmup or decay
Weight decay	10^{-4}
Batch size	64 context-query examples
Gradient clipping	Global ℓ_2 norm, maximum 10
Incorrect-prefix raw weight	$\alpha = 0.2$
Auxiliary-loss coefficient	$\lambda = 1$
Weight normalization	Mean one within each training example
State normalization	Training-data statistics where enabled
Online parameter updates	None

Table 6: Context lengths and update budgets for the six main environments.

Environment	H	Epochs	Updates per Epoch	Total Updates
DarkRoom	100	150	2,000	300,000
Cheetah-Vel	200	100	1,000	100,000
Hopper-Param	200	100	1,000	100,000
Reach	200	50	2,000	100,000
Push	100	50	2,000	100,000
Walker-Param	200	100	1,000	100,000

An epoch in Table 6 is a configured block of optimizer steps, not necessarily a full pass through the offline dataset; total updates equal epochs times updates per epoch. State normalization is enabled for the five continuous-control configurations and disabled for DarkRoom. Context length need not equal episode length: Reach uses $H = 200$ although a collected episode has 100 transitions.

For a batch of B examples with H transitions, encoder states at each depth have shape $B \times H \times d$; concatenation across layers gives $B \times H \times L_e d$, and fusion maps this to $B \times H \times d$. The query yields $H + 1$ decoder positions: position zero holds the query, while positions $1, \ldots, H$ yield the action predictions. The classifier produces $B \times H \times N$ logits. Zero initialization of W_f makes the initial fused representation equal to the final encoder state. The task loss updates the classifier and encoder; the action loss updates the encoder, fusion, query embedding, decoder, and action head. Discrete correctness weights do not provide a gradient path into the classifier. The next state s'_i and reward r_i in the input token belong to an already observed transition, not to the predicted query action.

C Additional Analysis of TIARA

C.1 Prefix Causality and Query Independence

At encoder depth zero, u_i depends only on transition x_i and its position. Causal attention at position i accesses only $j \leq i$; position-wise transformations introduce no later transition. By induction, $h_i^{(\ell)}$

depends only on $x_{1:i}$, and the encoder does not receive s_q. Position-wise layer normalization and fusion preserve this dependency for m_i. The causal decoder at position i accesses the query token and $m_{1:i}$, so $\hat{a}_q^i$ uses $(C_{\le i}, s_q)$. Prefix order follows the sequence presented to the encoder, including shuffled contexts.

C.2 Properties of Normalized Reliability Weights

Let $p_b = H^{-1} \sum_{i=1}^{H} \rho_{b,i}$ be the fraction of correctly classified prefixes in example b. The normalization denominator and the two possible weights are

$$D_b = \alpha + (1-\alpha)p_b, \qquad w_{b,i} = \begin{cases} 1/D_b, & \rho_{b,i} = 1, \\ \alpha/D_b, & \rho_{b,i} = 0. \end{cases} \tag{13}$$

Because $\alpha \le D_b \le 1$, every weight is positive and satisfies $\alpha \le w_{b,i} \le 1/\alpha$. When both groups occur and $\alpha < 1$, correct-prefix weights exceed one, incorrect-prefix weights are below one, and their ratio is $1/\alpha$. Summing the weights gives

$$\frac{Hp_b + H(1-p_b)\alpha}{D_b} = H. \tag{14}$$

Thus each example retains total action-supervision weight H.

If all prefixes share the same correctness value, their weights are one; $\alpha = 1$ also gives uniform weights. Taking $\alpha > 0$ keeps D_b positive even when every prefix is misclassified.

C.3 Gradient Flow with Discrete Weights

With the discrete weights held fixed for a backward pass, the shared-encoder gradient is

$$\nabla_\theta \mathcal{L}|_{w \text{ fixed}} = \sum_{b,i} w_{b,i} \nabla_\theta \ell_a((\hat{a}_q^i)_b, (a_q)_b) + \lambda \nabla_\theta \mathcal{L}_{\text{task}}. \tag{15}$$

The encoder receives action- and task-loss gradients; the classifier receives task-loss gradients. Its discrete correctness indicator provides no action-loss gradient path, although weights can change on later updates.

D Environment and Data Organization

D.1 Environments and Offline Data

The six-environment main comparison uses Mixed offline data. DarkRoom varies the goal on a discrete navigation grid; Cheetah-Vel varies the target forward velocity. Reach and Push vary manipulation goals, and Walker-Param varies physical parameters. The Hopper entry in the data inventory below records the task files used for the main comparison.

For the separately collected Hopper tasks, physical parameters are sampled once with seed 1337 and retained in the task-definition file. Each element of mass, inertia, and friction is multiplied by 1.5^v, and damping by 1.3^v, where $v \sim \mathcal{U}(-3, 3)$ is sampled elementwise. The wrapper uses an 11-dimensional observation, a 3-dimensional action, and a 200-step horizon.

The five non-Hopper environments use the Mixed files described in Section 5.1; Hopper is collected separately with the SAC-based procedure below. Table 7 records context and query array sizes for the encoder–layer-fusion configuration. Counts include tasks reserved for evaluation and are per task; a query pair is a supervised state–action record. Here L is the trajectory loader's fixed block length and N_c/L the number of context blocks. State/action widths refer to stored arrays after flattening trailing action dimensions; DarkRoom actions are stored as 5-way vectors.

A task-specific SAC agent is trained for 10^6 environment steps, with checkpoints every 10,000 steps. Stochastic rollouts collect 400 transitions from each of the 100 checkpoints, yielding 40,000 context transitions per task. For query labeling, candidate checkpoints are evaluated for 10 episodes each; the checkpoint with the highest mean return supplies action labels at the stored context states. The collector uses seed 12345.

Table 7: Mixed offline data inventory (per task).

Environment	Tasks	State/Action	L	Context N_c	Query N_q	Blocks
DarkRoom	100	2/5	100	100,000	1,000	1,000
Cheetah-Vel	50	20/6	200	40,000	40,000	200
Reach	50	18/4	100	20,000	20,000	200
Push	50	18/4	100	50,000	50,000	500
Walker-Param	50	17/6	200	40,000	40,000	200
Hopper-Param	50	11/3	200	40,000	40,000	200

D.2 Context–Query Sampling and Preprocessing

Each task has a context file and a query file. The former contains $x_i = (s_i, a_i, s'_i, r_i)$ transitions; the latter supplies (s_q, a_q) targets. The loader indexes query rows from training tasks and draws a context from the same task for each query. The same query pair supervises every prefix of the sampled context. The query arrays provide the supervised state–action pairs. The stored query states match context states in row order in the continuous-control data, while the query actions depend on the environment and labeling procedure. DarkRoom has a separate, smaller query array.

For $H \leq L$, the loader selects one block uniformly and then a uniform contiguous window of length H within it. In the listed configurations with $H > L$, H is a multiple of L; the loader selects H/L blocks without replacement and concatenates them in sampled order. Reach uses two 100-transition blocks for $H = 200$; the other environments use one block. The encoder–layer-fusion training configuration then applies a common random permutation to all context fields, preserving each complete transition. Prefix indices refer to this presented order. Query sampling does not exclude a transition that also appears in the context. The zero-based task IDs in the stored files are mapped to classifier labels $z \in \{1, \ldots, N\}$ in the mathematical notation. These labels supervise the auxiliary task head during training and are not supplied to the deployed action policy.

DarkRoom uses task IDs 0–79 for training and 80–99 for evaluation. Each continuous-control environment uses IDs 0–44 for training and 45–49 for evaluation. IDs refer to the order in the stored task-definition file. Model optimization and preprocessing statistics use training tasks only. Online evaluation conditions on transitions collected from the evaluation task. We use three independent training seeds. For each seed, returns are first averaged across evaluation tasks; we then report the mean and sample standard deviation across the three seeds.

For continuous-control environments, state statistics are computed over current-state rows in the training-task context buffers only. With N_s such rows, the implementation uses the coordinatewise transformation

$$\mu = \frac{1}{N_s}\sum_{n=1}^{N_s} s^{(n)}, \qquad \sigma = \sqrt{\max\left(\frac{1}{N_s}\sum_{n=1}^{N_s}(s^{(n)})^2 - \mu^2, 0\right)} + 10^{-6}, \qquad \widetilde{s} = \frac{s-\mu}{\sigma}. \tag{16}$$

The maximum, square root, addition, and division in this transformation are coordinatewise. The same frozen μ and σ are applied to context states, next states, and query states, including evaluation-task inputs. Query and next-state arrays are not counted again when fitting these statistics. DarkRoom disables state normalization. The trajectory loader leaves stored rewards and action targets unchanged, apart from shape conversion and conversion to floating-point tensors.

E Layer-Fusion Visualization

To investigate how different encoder layers contribute to downstream action execution, we perform intervention experiments on the layer-fusion mechanism across various context lengths. Specifically, partition the projection matrix in Eq. (6) as $W_f = [W_1 \ \cdots \ W_{L_e}]$, where $W_\ell \in \mathbb{R}^{d\times d}$. The fused representation can then be expressed as

$$m_i = h_i + \sum_{\ell=1}^{L_e} W_\ell \, \mathrm{LN}_\ell(h_i^{(\ell)}). \tag{17}$$

We report the performance drop as $\Delta\text{MSE} = \text{MSE}_{\text{intervention}} - \text{MSE}_{\text{full}}$, where positive values indicate higher action-prediction error, reflecting the importance of the removed component.

To dissect feature utilization, we evaluate several intervention strategies: "Drop L1–L4" removes individual layer contributions, "Last-only" retains solely the final layer's fusion path, and "No fusion" strips away all multi-layer components while keeping the base residual h_i. As illustrated in Figure 7, the impact of layer-wise fusion varies significantly across environments. On REACH, removing intermediate layer contributions noticeably degrades prediction accuracy, revealing that fine-grained tasks benefit from multi-scale feature aggregation. Conversely, on WALKER, the most substantial error increases occur under "No fusion" at longer context lengths, highlighting that robust locomotion dynamics heavily rely on sustained multi-layer encoder features as interaction history accumulates.

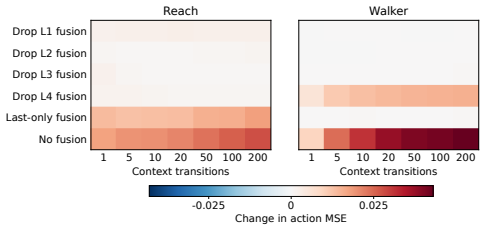


Figure 7: Layer-fusion interventions on Reach and Walker across varying context lengths.

F MODEL SIZE AND PARAMETER EFFICIENCY

Table 8 reports the network parameter counts evaluated on the WALKER-PARAM environment (17-dimensional states and 6-dimensional actions), comparing TIARA against all baselines across their respective architectural configurations.

Table 8: Network parameter counts on Walker-Param.

Parameter Type	**TIARA**	**DPT**	**AD**	**T2MIR-DPT**	**DIT**	**S-ICQL**	**UNICORN**
Core Model	122,611	83,366	2,830,506	1,464,106	4,901,900	1,030,729	107,820
Training Auxiliaries	0	0	0	0	9,747,714	25,361	267,157
Total	122,611	83,366	2,830,506	1,464,106	14,649,614	1,056,090	374,977

As detailed in Table 8, TIARA maintains remarkable parameter efficiency. Counts exclude optimizer states and non-parameter buffers, and include all mixture-of-experts parameters. The core model of TIARA comprises the context encoder, action decoder, layer fusion, and task classifier. Notably, TIARA introduces **zero training auxiliaries** (0 auxiliary parameters), avoiding the heavy additional networks seen in value-based or meta-RL alternatives (e.g., DIT, S-ICQL, and UNICORN). Although incorporating a decoupled encoder and layer-fusion mechanism, TIARA remains extremely lightweight, scaling to only $\sim 1.47\times$ the parameters of baseline DPT while achieving substantial performance gains across challenging benchmarks.