DECISION TRANSFORMER WITH TASK-AWARE REPRESENTATIONS FOR OFFLINE META-REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

To enable effective generalization in offline meta-RL, it is essential to accurately infer the underlying task through latent task representations, which provide a compact and transferable summary of task-specific characteristics for guiding policy generation. In this paper, we propose **Contrastive Task-Aware Decision Transformer (CoTA-DT)**, a DT-based offline meta-RL framework that learns task representations by jointly integrating four complementary components: *(i)* contrastive learning from offline trajectories, *(ii)* alignment with natural-language task descriptions, *(iii)* one-step supervision via a task-conditioned world model, and *(iv)* long-term supervision via a task-conditioned Q function. The learned task representations are used as conditioning signals for a DT-based policy, enabling task-aware meta-policy learning. To further support generalization in both zero-shot and few-shot settings, we design a language-grounded, self-adaptive strategy that enables effective and reliable adaptation to unseen tasks. Notably, in CoTA-DT, the pretrained LLM serves as a shared backbone for language encoding, world modeling, Q-function estimation, and policy learning. This unified design facilitates parameter sharing, improving both efficiency and generalization. Extensive experiments across diverse offline meta-RL benchmarks demonstrate that CoTA-DT consistently outperforms prior methods, particularly in challenging zero-shot generalization scenarios.

1 INTRODUCTION

Offline meta-reinforcement learning (RL) aims to learn transferable priors from offline trajectories across related tasks, enabling efficient adaptation to new tasks with little or no additional data. In recent years, a prominent line of work in offline meta-RL has built on Decision Transformers (DTs) (Chen et al., 2021), leveraging the strong generalization capabilities of transformer architectures (Mitchell et al., 2021; Lin et al., 2022). These DT-based approaches formulate policy learning as sequence modeling over trajectories and perform adaptation through sequence conditioning.

For example, Prompt-DT (Xu et al., 2022) leverages expert demonstrations as prompts to encode task-specific information, enabling prompt-based adaptation at test time. Generalized Decision Transformers (GDT) (Furuta et al., 2021; Wang et al., 2023) incorporate hindsight signals, such as reward statistics or retrospective summaries, as conditioning inputs for task adaptation. Meta-DT (Wang et al., 2024) conditions the policy on compact task representations learned from trajectories to support generalization. More recently, T2DA (Zhang et al., 2025) conditions a DT-based policy on text embeddings derived from natural-language task descriptions. Despite the advances, these approaches typically learn task identifiers from a single source—such as language, trajectory data, or reward signals—and use them as conditioning inputs to guide DT-based policies for generalization. When such signals are ambiguous, noisy, or distributionally shifted relative to training, task identification can become unreliable, leading to degraded adaptation performance. These limitations naturally raise a question: *How can we learn an offline meta-policy that achieves reliable task identification and supports robust generalization in both few-shot and zero-shot settings?*

Reliable task identification across a broad family of related tasks depends on the availability of task-specific information. In offline meta-RL, collected trajectories serve as the primary source

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.

of information and implicitly encode task differences. However, relying solely on offline data is often insufficient, especially when the data are limited, noisy, or provide only partial coverage of the task space. Beyond trajectories, additional sources of information are often available. For example, natural-language task descriptions derived from task-level metadata—such as task goals, names, relevant objects, and environment constraints—provide complementary high-level semantics that are not always recoverable from trajectories alone. Meanwhile, environment transitions and rewards vary across tasks, offering informative one-step signals of task-specific dynamics and objectives, while cumulative returns (*e.g.*, Q functions) capture long-horizon goals and provide global signals for distinguishing tasks. These observations motivate jointly leveraging multiple sources of information to learn robust task representations and enable reliable task identification in offline meta-RL.

In this paper, we propose **Contrastive Task-Aware Decision Transformer (CoTA-DT)**, a DT-based offline meta-RL framework that learns task representations by jointly integrating four complementary components: *(i)* contrastive task encoding from offline trajectories, *(ii)* alignment with natural-language task descriptions, *(iii)* one-step supervision via a task-conditioned world model, and *(iv)* long-term supervision via a task-conditioned Q function. Specifically, task representations are first learned from trajectories via a contrastive objective that pulls together representations from the same task while separating those from different tasks. However, trajectory-only representations can be noisy due to data variability. To better capture underlying task objectives and reduce bias, we further align them with natural-language task descriptions encoded by a pretrained large language model (LLM), promoting semantic consistency between trajectory- and language-derived information. To further refine the representations, we introduce a task-conditioned world model that provides one-step supervision through transitions and rewards, along with a task-conditioned Q function that offers long-horizon supervision over task objectives. These complementary signals regularize the representations, and all components are optimized jointly to learn task representations from multiple perspectives.

The resulting task representations are used as conditioning signals for a DT-based policy, enabling task-aware meta-policy learning. To support generalization in both zero-shot and few-shot settings, we design a language-grounded, self-adaptive strategy for reliable adaptation. Notably, in CoTA-DT, the pretrained LLM serves as a shared backbone for language encoding, world modeling, Q-function estimation, and policy learning. This unified design enables parameter sharing across stages and leverages the LLM's prior knowledge from large-scale pretraining, along with its strong sequence modeling and transfer capabilities under prompt conditioning (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023), thereby improving both efficiency and generalization. *Our main contributions are as follows:*

- We introduce CoTA-DT, an offline meta-RL framework whose policy conditions on a task latent that supports both language-based initialization and updates from observed transitions.
- We learn discriminative, language-grounded task representations that capture task dynamics and long-horizon returns, supporting adaptation to unseen tasks with a shared LoRA-adapted LLM backbone.
- We develop zero-shot deployment through language-grounded latent inversion and few-shot adaptation through support-based initialization, with optional policy fine-tuning.
- Extensive experiments across various offline meta-RL benchmarks demonstrate that CoTA-DT consistently outperforms prior methods, particularly in challenging zero-shot generalization settings[1].

2 Related Work

Offline Meta-RL. Offline RL learns policies from fixed datasets without further environment interaction (Levine et al., 2020; Konyushkova et al., 2020). Extending this setting, offline meta-RL aims to learn transferable priors that enable generalization across tasks using only offline data (Mitchell et al., 2021; Pong et al., 2021; Dorfman et al., 2021; Li et al., 2020a). A large body of work in this area builds on temporal-difference (TD) learning, often incorporating optimization-based meta-learning or latent context inference mechanisms to capture task-specific information (Finn et al., 2017; Mitchell et al., 2021; Xu et al., 2018; Rakelly et al., 2019; Zintgraf et al., 2021; Yuan & Lu, 2022; Li et al., 2024). However, applying TD learning in the offline setting is challenging due to the deadly triad (Levine et al., 2020), which can lead to instability and error accumulation. To mitigate

[1]Code available at `https://anonymous.4open.science/r/CotaDT-35DB`.

this issue, many approaches impose constraints to keep learned policies within the support of the dataset (Ajay et al., 2022; Kumar et al., 2020; Kostrikov et al., 2021; Yu et al., 2020b; Kidambi et al., 2020). Other works pursue more stable alternatives, such as return-conditioned or sequence modeling approaches based on Decision Transformers (Wang et al., 2024; Hu et al., 2024; Xu et al., 2022; Wang et al., 2023; Zhang et al., 2025).

LLMs for RL. Recent work explores leveraging the representation and generalization capabilities of LLMs for RL, including using them as policy backbones or for reward modeling (Morad et al., 2024; Reed et al., 2022; Brohan et al., 2023). In the offline RL setting, LLMs are incorporated in several ways. LaMo (Shi et al., 2023) demonstrates that pretrained language models can be adapted as effective policy backbones, while KALM (Pang et al., 2024) leverages LLM-generated rollouts to augment offline datasets. More recently, T2DA (Zhang et al., 2025) conditions decision models on text embeddings derived from LLMs to support generalization in offline meta-RL, and LLMDPD (Zhang & Guo, 2025) further enhances generalization by integrating LLM priors into diffusion-based policy learning. In contrast, our method leverages LLMs as a shared backbone for both task representation learning and policy modeling.

3 PRELIMINARIES

3.1 OFFLINE META-REINFORCEMENT LEARNING

An RL task is formulated as an MDP $M = (S, A, T, r, \gamma)$ with state space S, action space A, transition $T : S \times A \to \Delta(S)$, reward $r : S \times A \to \mathbb{R}$, and discount $\gamma \in [0, 1)$. The objective is to learn a policy $\pi(a|s)$ that maximizes the expected return: $\max_\pi J_M(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$. In offline meta-RL, tasks $M_i = (S, A, T_i, r_i, \gamma) \sim P(\mathcal{M})$ share the same state and action spaces but differ in T_i and r_i. For each of N training tasks, an offline dataset $\mathcal{D}_i = \{(s_{i,j}, a_{i,j}, r_{i,j}, s'_{i,j})\}_{j=1}^{J}$ is collected by an arbitrary behavior policy π_β^i. Each training task is paired with a description h_i. For an unseen task $\mathcal{M}_\star \sim P(\mathcal{M})$, the description $h_\star$ is available. Zero-shot evaluation provides no target-task support trajectories before rollout; few-shot evaluation additionally provides k support trajectories. During rollout, the agent observes states and rewards and may update its task latent from past transitions, with policy parameters fixed. The goal is to learn a meta-policy maximizing $J(\pi_{\text{meta}}) = \mathbb{E}_{M \sim P(\mathcal{M})} \left[J_M(\pi_{\text{meta}})\right]$.

3.2 DECISION TRANSFORMER

Decision Transformer (DT) recasts RL as return-conditioned sequence modeling: each trajectory is a sequence of (R_t, s_t, a_t) triplets where $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ is the return-to-go (RTG), and the model is trained to autoregressively predict a_t from past RTGs, states, and actions by minimizing prediction error against ground-truth actions.

4 METHOD

In this section, we present Contrastive Task-Aware Decision Transformer (CoTA-DT), an offline meta-RL framework that enables reliable task identification through multi-perspective task representations and integrates task inference into DT-based policy for improved generalization. Fig. 1 illustrates the method overview.

4.1 MULTI-PERSPECTIVE TASK REPRESENTATION LEARNING

To enable effective generalization in offline meta-RL, it is essential to distinguish between tasks by capturing both their differences and shared structure. Accurately inferring the underlying task through latent task representations is therefore critical, as it provides a compact and transferable summary of task-specific characteristics that can be used to condition policy generation.

However, relying on a single source of information, such as offline datasets, is often insufficient for learning reliable task representations. Offline datasets typically contain trajectories exhibiting varying levels of expertise, data quality, and coverage. As a result, the learned representations can be noisy

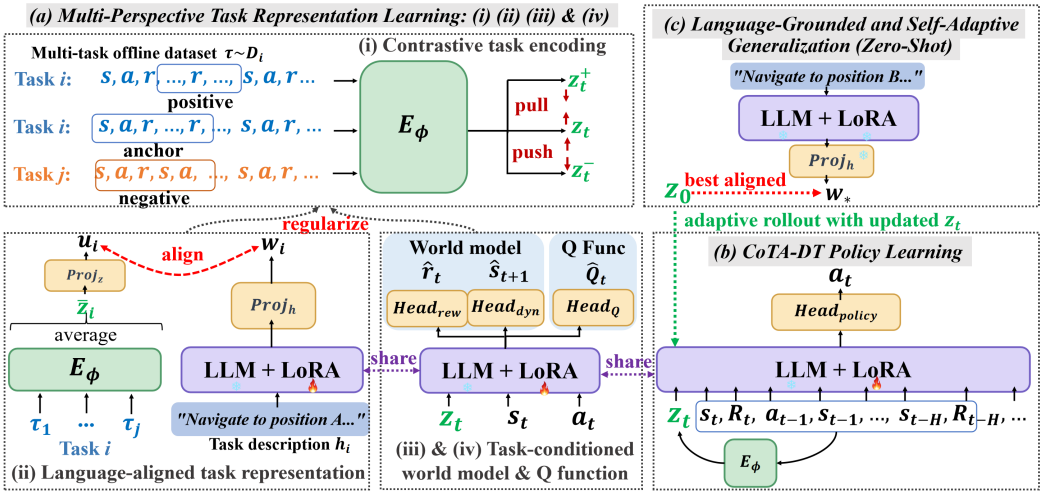

Figure 1: **Overview of CoTA-DT.** *(a)* Multi-perspective task representation learning jointly optimizes four signals—*(i)* contrastive task encoding from offline trajectories, *(ii)* language alignment between u_i and w_i, *(iii)* a task-conditioned world model predicting $\hat{r}_t$, $\hat{s}_{t+1}$, and *(iv)* a long-horizon Q head predicting $\hat{Q}_t$—all sharing one frozen LLM with a single LoRA adapter. *(b)* A Decision Transformer policy is trained on the same LLM backbone, conditioned on z_t. *(c)* At zero-shot adaptation, a new task description is encoded into $w_\star$, z_0 is constructed by aligning with $w_\star$, and the rollout uses self-adaptive z_t updates.

and less robust. To address this limitation, we propose to learn and refine latent task representations from multiple perspectives: *(i)* contrastive task encoding from offline trajectories (Sec. 4.1.1), *(ii)* alignment with natural-language task descriptions (Sec. 4.1.2), *(iii)* one-step supervision via a task-conditioned world model (Sec. 4.1.3), and *(iv)* long-term supervision via a task-conditioned Q function (Sec. 4.1.3).

4.1.1 Contrastive Task Encoding

To learn latent task representations, offline trajectories provide a primary source of information for capturing task-specific characteristics, as they inherently reflect the underlying task through observed behaviors. We encode short trajectory segments using a separate GRU context encoder E_ϕ, which maps transitions to a task latent $z \in \mathbb{R}^{64}$; the LLM backbone processes text and projected continuous tokens. Sub-trajectories capture partial yet informative aspects of the task, enabling the learned representations to reflect not only the underlying task objectives but also the stage of task execution. For example, in locomotion tasks, a short segment can indicate both the objective (*e.g.*, moving forward or turning) and the current execution stage (*e.g.*, initiating motion, stabilizing, or adjusting direction).

Formally, given a sub-trajectory $\tau_{t-H:t} = \{(s_j, a_j, r_j, s_{j+1})\}_{j=t-H}^{t}$ of horizon H extracted from an offline trajectory, the encoder maps it to a latent representation $z_t = E_\phi(\tau_{t-H:t}), z_t \in \mathbb{R}^{d_z}$. For timesteps near the beginning of a trajectory where $t - H < 0$, we construct $\tau_{t-H:t}$ by zero-padding the missing prefix. For each anchor sub-trajectory $\tau_{t-H:t}$, we construct a positive sample τ^+ as a nearby segment from the same trajectory, *e.g.*, $\tau^+ = \tau_{t-H+\delta:t+\delta}$ with a small offset δ. Negative samples $\{\tau_k^-\}_{k=1}^K$ are drawn from trajectories of different tasks. Let $z_t^+ = E_\phi(\tau^+), z_k^- = E_\phi(\tau_k^-)$. We optimize the standard InfoNCE (van den Oord et al., 2018) objective:

$$\mathcal{L}_{\text{NCE}} = -\mathbb{E}\left[\log \frac{\exp(\text{sim}(z_t, z_t^+)/\kappa_\alpha)}{\exp(\text{sim}(z_t, z_t^+)/\kappa_\alpha) + \sum_{k=1}^K \exp(\text{sim}(z_t, z_k^-)/\kappa_\alpha)}\right], \tag{1}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and κ_α is a temperature parameter. This objective encourages representations of temporally adjacent sub-trajectories from the same task to align, as they share the same task objective and similar execution stages, while pushing apart representations from different tasks.

4.1.2 LANGUAGE-ALIGNED TASK REPRESENTATION

Since the latent representations z learned by the contrastive encoder are derived from sub-trajectories, they can be biased toward local behaviors or the current execution stage rather than the underlying task objective. This issue is particularly pronounced when sub-trajectories are sampled from low-quality, non-expert data, or from very early stages of task execution, where the task goal is not yet clearly manifested. To encourage the latent representations to better capture the global task objective, we ground z in natural-language task descriptions, enforcing consistency between latent representations and their corresponding semantic descriptions.

Specifically, natural-language descriptions h_i are deterministically constructed from task-level metadata of task M_i, including the task name, goal, relevant objects, environmental conditions, and typical performance. Each description h_i is encoded into a text embedding v_i using a pretrained LLM with LoRA adaptation (Hu et al., 2022), *i.e.*, $v_i = \mathrm{LLM}_{\theta_{\text{base}},\theta_{\text{L}}}(h_i)$, where the LLM backbone θ_{base} is kept frozen and the LoRA adapter θ_{L} is fine-tuned. For each task i, we also compute an aggregated representation $\bar{z}_i$ by averaging latent representations over a set of sub-trajectories sampled from the offline dataset $\mathcal{D}_i$: $\bar{z}_i = \frac{1}{m}\sum_{j=1}^{m} E_\phi(\tau_j^i), \tau_j^i \sim \mathcal{D}_i$. Both $\bar{z}_i$ and v_i are then projected into a shared embedding space via lightweight projection heads Proj_z and Proj_h for alignment:

$$u_i = \frac{\mathrm{Proj}_z(\bar{z}_i)}{\|\mathrm{Proj}_z(\bar{z}_i)\|_2}, \qquad w_i = \frac{\mathrm{Proj}_h(v_i)}{\|\mathrm{Proj}_h(v_i)\|_2}, \qquad u_i, w_i \in \mathbb{R}^{d_a}. \tag{2}$$

We then quantify the similarity between u_i and w_i by defining a cosine similarity matrix over a minibatch of tasks $\mathcal{I}$, scaled by a learnable temperature κ_β optimised in log-space to enforce positivity, following (Radford et al., 2021):

$$S_{ij} = \frac{u_i^\top w_j}{\kappa_\beta}, \qquad \kappa_\beta = e^{\log \kappa_\beta}, \tag{3}$$

We optimize a symmetric CLIP-style (Radford et al., 2021) alignment objective:

$$\begin{aligned} &\mathcal{L}_{\text{align}} = \tfrac{1}{2}\left(\mathcal{L}_{z\to h} + \mathcal{L}_{h\to z}\right), \\ &\mathcal{L}_{z\to h} = -\tfrac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log \tfrac{\exp(S_{ii})}{\sum_{j\in\mathcal{I}}\exp(S_{ij})}, \quad \mathcal{L}_{h\to z} = -\tfrac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log \tfrac{\exp(S_{ii})}{\sum_{j\in\mathcal{I}}\exp(S_{ji})}. \end{aligned} \tag{4}$$

This contrastive objective treats $(\bar{z}_i, h_i)$ as a positive pair, while $(\bar{z}_i, h_j)$ for $j \neq i$ serve as negatives. It encourages trajectory-derived task representations z to align with their corresponding language descriptions, thereby grounding the latent representations in high-level task semantics.

4.1.3 TASK-CONDITIONED WORLD MODELING AND Q LEARNING

Beyond trajectories and natural-language task descriptions, environment transitions and rewards inherently vary across tasks, providing informative one-step signals of task-specific dynamics and objectives that help regularize task representations. Over longer horizons, cumulative returns (*e.g.*, Q functions) capture task goals and provide complementary global signals for distinguishing tasks. Motivated by these observations, we introduce a task-conditioned world model and a task-conditioned Q function to further refine the learned representations.

Task-Conditioned World Modeling. Given a transition (s_t, a_t, r_t, s_{t+1}) sampled from a sub-trajectory $\tau_{t-H:t}$, we first infer a latent representation $z_t = E_\phi(\tau_{t-H:t})$. Conditioned on z_t, the world model predicts the immediate reward and next state, *i.e.*, $p\big(r_t, s_{t+1} \mid s_t, a_t;\, z_t\big)$, and is trained with supervised objectives

$$\mathcal{L}_{\text{rew}} = \mathbb{E}\big[\|\hat{r}_t - r_t\|_2^2\big], \qquad \mathcal{L}_{\text{dyn}} = \mathbb{E}\big[\|\hat{s}_{t+1} - s_{t+1}\|_2^2\big], \tag{5}$$

where $\hat{r}_t$ and $\hat{s}_{t+1}$ denote the predicted reward and next state, respectively.

Task-Conditioned Q Learning. To incorporate long-horizon supervision, we introduce a task-conditioned Q function that predicts the return-to-go given a state-action pair (s_t, a_t) and the corresponding task representation $z_t = E_\phi(\tau_{t-H:t})$: $Q(s_t, a_t;\, z_t) \approx \mathbb{E}\Big[\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}\Big]$. The model is trained with a supervised regression objective:

$$\mathcal{L}_{\text{Q}} = \mathbb{E}\big[\|\hat{R}_t - R_t\|_2^2\big], \tag{6}$$

where R_t denotes the empirical return-to-go computed from offline trajectories, and $\hat{R}_t$ is the corresponding prediction produced by the Q function.

Both the world model and the Q function are implemented by repurposing the same pretrained LLM backbone θ_{base} with the shared LoRA adapter θ_{L}, originally fine-tuned for language processing, and augmenting it with task-specific prediction heads for reward, dynamics, and return estimation (shown in Fig. 1). Moreover, the supervision signals from both components are backpropagated through z_t to jointly update the contrastive task encoder E_ϕ and LoRA parameters θ_{L}. The world model provides one-step supervision through task-specific transitions and rewards, while the Q function supplies long-horizon supervision that captures task objectives over extended trajectories. These complementary signals encourage E_ϕ to capture both short-term dynamics and long-term goals, improving its discriminative power for task identification and generalization, while also enabling the LLM to better understand environment dynamics and task objectives.

Overall Objective. We jointly optimize the multi-perspective representation learning objective:

$$\mathcal{L}_{\text{repr}} = \lambda_1 \mathcal{L}_{\text{NCE}} + \lambda_2 \mathcal{L}_{\text{align}} + \lambda_3 \left(\mathcal{L}_{\text{rew}} + \alpha \mathcal{L}_{\text{dyn}}\right) + \lambda_4 \mathcal{L}_{\text{Q}}, \tag{7}$$

where λ_1–λ_4 are balancing coefficients. This objective enables the task encoder E_ϕ to learn reliable, task-aware latent representations z by integrating complementary supervision signals, while the shared LoRA-adapted LLM backbone jointly captures trajectory information, task semantics, environment dynamics, and long-horizon objectives, ultimately benefiting downstream DT-based policy learning.

4.2 Task-Aware Policy Learning for CoTA-DT

Given the learned latent representations, CoTA-DT learns a DT-based policy for offline meta-RL. The policy is formulated as conditional sequence modeling: at each timestep t, the model conditions on the return-to-go R_t, the state history $s_{<t}$, the action history $a_{<t}$, and a task-aware latent representation $z_t = E_\phi(\tau_{t-H:t})$ inferred from the local sub-trajectory $\tau_{t-H:t}$. This representation encodes both the underlying task identity and the current execution stage, providing contextual guidance for action generation. The resulting conditional policy is defined as $\pi_\theta(a_t \mid R_{\le t}, s_{\le t}, a_{<t};\, z_t)$, and is trained on offline trajectories to predict actions autoregressively:

$$\mathcal{L}_{\text{policy}} = \mathbb{E}_{(s_t, a_t, R_{\le t}) \sim \mathcal{D}} \Big[-\log \pi_\theta(a_t \mid R_{\le t}, s_{\le t}, a_{<t};\, z_t) \Big], \tag{8}$$

where $\mathcal{D}$ is the offline dataset. Here, the DT-based policy is initialized using the same pretrained LLM backbone θ_{base} and the shared LoRA adapter θ_{L} learned in Sec. 4.1, with an additional action prediction head f_{act}. Through this shared architecture, the LoRA-adapted LLM is further fine-tuned under policy supervision via Eq. 8, while the policy directly benefits from the task representation learning stage, which integrates trajectory information, task semantics, environment dynamics, and long-horizon objectives. This unified architecture provides richer task context for action prediction, thereby improving generalization to unseen tasks.

4.3 Language-Grounded and Self-Adaptive Generalization

4.3.1 Zero-Shot Adaptation.

For a new unseen task $M_\star$, CoTA-DT can generate task-conditioned trajectories once an initial task representation z_0 is provided. In principle, z_0 should be inferred by the well-trained task encoder E_ϕ from trajectories of the target task. However, in the zero-shot setting, no task-specific trajectories are available, making it infeasible to obtain z_0 directly from data.

Language-Grounded Initialization. To address this issue, we instead initialize z_0 by grounding it in the task description $h_\star$ of the unseen task, which is typically available, providing a meaningful starting point for subsequent policy rollout. Given a task description $h_\star$, we first encode it into a text embedding $v_\star$ using the pretrained LLM backbone θ_{base} with LoRA adaptation θ_{L}: $v_\star = \text{LLM}_{\theta_{\text{base}}, \theta_{\text{L}}}\big(h_\star\big)$. The embedding is then projected into the shared alignment space via the projection head Proj_h: $w_\star = \text{norm}\big(\text{Proj}_h(v_\star)\big)$. To obtain an initial task representation $z_0 \in \mathbb{R}^{d_z}$, we optimize Eq. (9) to find a latent vector whose projection best aligns with $w_\star$, using a small number of updating steps starting from $z = \mathbf{0}$:

$$z_0 \;=\; \arg\min_{z} \; \Big[1 - \text{norm}\big(\text{Proj}_z(z)\big)^\top w_\star \Big]. \tag{9}$$

During this process, only z is updated, while the encoder E_ϕ, projection heads $\mathrm{Proj}_z, \mathrm{Proj}_h$, the shared LoRA adapter θ_{L}, and the LLM backbone θ_{base} remain fixed. As a result, this procedure serves as a lightweight test-time inference step rather than additional training.

Self-Adaptive Rollout. The resulting z_0 is used to condition CoTA-DT policy at the beginning of rollout. As the rollout proceeds, once H transitions have been collected, we progressively update the task representation by recomputing $z_t = E_\phi(\tau_{t-H:t-1})$ from the most recent sub-trajectory. The updated z_t conditions subsequent action generation, enabling a self-adaptive rollout that continuously refines the task representation based on newly observed transitions. By capturing both task identity and execution stage, z_t allows the policy to adapt dynamically to the evolving task context, resulting in reliable and effective adaptation.

4.3.2 FEW-SHOT ADAPTATION.

In the few-shot setting, a small support set $\mathcal{D}_\star$ is available alongside the task description $h_\star$. We construct the initial task representation z_0 by leveraging both the available data and the task description to obtain a more reliable initialization. Specifically, we sample a set of sub-trajectories $\mathcal{C}$ from $\mathcal{D}_\star$ and aggregate their encoder outputs $\bar{z} = \frac{1}{|\mathcal{C}|}\sum_{\tau\in\mathcal{C}} E_\phi(\tau)$, then use $\bar{z}$ as the starting point for the optimization in Eq. (9), yielding a refined z_0 grounded in both $\mathcal{D}_\star$ and $h_\star$. With z_0 held fixed, we further fine-tune *only* the LoRA adapter θ_{L} and the action head f_{act} (with adapted parameters $\leq 1\%$ of the LLM backbone) on $\mathcal{D}_\star$ using $\mathcal{L}_{\mathrm{policy}}$ (Eq. 8) for a small number of gradient steps. The adapted policy is then deployed using the same self-adaptive rollout procedure as in the zero-shot setting.

5 EXPERIMENTS

We design our experiments to answer four questions. **(Q1)** Does CoTA-DT outperform state-of-the-art language-conditioned and context-based offline meta-RL baselines in both *zero-shot* and *few-shot* settings? **(Q2)** How does each component—contrastive task encoding ($\mathcal{L}_{\mathrm{NCE}}$), language alignment ($\mathcal{L}_{\mathrm{align}}$), world modeling ($\mathcal{L}_{\mathrm{rew}}+\mathcal{L}_{\mathrm{dyn}}$) and Q learning ($\mathcal{L}_Q$)—contribute to CoTA-DT performance? **(Q3)** How robust is the method to dataset quality? **(Q4)** Is multi-perspective task representation learning effective in producing high-quality representations?

Environments. We follow the offline meta-RL protocol of T2DA (Zhang et al., 2025) and evaluate on four benchmarks: **Point-Robot** (2D navigation, 50 tasks, T=20), **Cheetah-Vel** (target velocity, 50 tasks, T=200), **Ant-Dir** (target direction, 50 tasks, T=200), and **Meta-World** (22 manipulation tasks (Yu et al., 2020a), T=200); see Appendix C for full statistics. Each task is accompanied by a natural-language description (*e.g.*, *"Navigate to position* $(-0.75, 0.93)$*"*; see Appendix C for details). For all environments, we follow prior work (including T2DA (Zhang et al., 2025) and Meta-DT (Wang et al., 2024)) and report results under three offline dataset qualities: *Medium*, *Expert*, and *Mixed*, using the best-performing checkpoint.

Baselines. We compare against six representative baselines spanning language-conditioned policies (BC-Z (Jang et al., 2022), BAKU (Haldar et al., 2024)), prompt-based DTs (Prompt-DT (Xu et al., 2022)), context-based methods (Meta-DT (Wang et al., 2024), UNICORN (Li et al., 2024)) and the strongest published language-conditioned DT, T2DA-T (Zhang et al., 2025). All methods are evaluated zero-shot and few-shot on 5 held-out tasks over 5 seeds; we report raw undiscounted episodic return $\pm$ one standard deviation. Implementation details, exact hyperparameters and per-task numbers are deferred to the appendix.

5.1 MAIN RESULTS: ZERO-SHOT AND FEW-SHOT ADAPTATION

Table 1 reports zero-shot returns on the *Mixed* dataset, which is the most challenging setting. CoTA-DT achieves the best performance on *all four* benchmarks. The largest absolute gain over the second-best baseline (T2DA-T) is observed on Meta-World ($+189.0$), while consistent relative improvements are also achieved on locomotion tasks ($+15.7\%$ on Cheetah-Vel and $+2.8\%$ on Ant-Dir). Table 2 reports the corresponding few-shot returns. CoTA-DT again attains the best performance on every benchmark, with margins that further widen over the second-best baseline T2DA-T (e.g.,

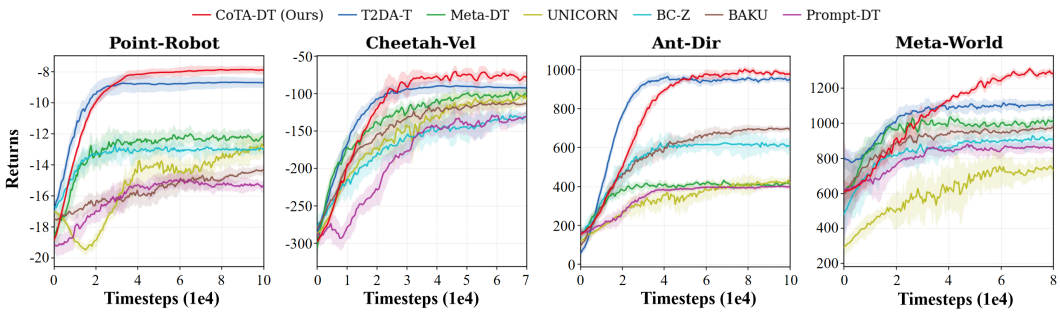

Figure 2: Zero-shot test return curves of CoTA-DT against baselines on Mixed datasets.

+213.1 on Meta-World and +11.5 on Cheetah-Vel), benefiting from the availability of a small support set. The results suggest that CoTA-DT learns more robust and discriminative task representations, enabling effective generalization in both settings.

Table 1: Zero-shot test return on the **Mixed** dataset. Best in **bold**, second-best underlined.

	BC-Z	BAKU	Prompt-DT	Meta-DT	UNICORN	T2DA-T	**CoTA-DT**
Pt-Robot	−13.0±0.7	−14.5±0.8	−15.2±0.6	−12.2±0.5	−13.1±0.7	−8.7±0.3	**−7.9±0.4**
Cheetah-Vel	−129.2±10.0	−112.3±4.2	−136.9±17.5	−99.7±6.9	−101.9±18.0	−90.7±5.8	**−76.5±4.7**
Ant-Dir	611.7±30.9	695.8±25.5	395.8±12.6	411.0±13.3	426.1±23.8	955.0±9.0	**981.3±16.8**
Meta-World	910.3±27.1	970.9±36.2	867.4±46.5	994.2±45.0	756.1±60.8	1103.5±36.3	**1292.5±28.2**

Table 2: Few-shot test return on the **Mixed** dataset. Best in **bold**, second-best underlined.

	BC-Z	BAKU	Prompt-DT	Meta-DT	UNICORN	T2DA-T	**CoTA-DT**
Pt-Robot	−12.3±0.8	−13.4±0.9	−12.0±1.3	−10.2±0.6	−12.9±1.2	−7.6±0.4	**−6.9±0.2**
Cheetah-Vel	−109.2±11.2	−97.5±4.3	−101.3±10.6	−86.5±7.3	−89.7±19.8	−78.8±4.1	**−67.3±5.2**
Ant-Dir	696.5±39.8	817.6±20.1	509.0±13.4	534.2±14.0	459.4±15.5	1012.3±9.8	**1040.2±15.8**
Meta-World	1131.7±56.3	1129.2±24.8	972.8±31.2	1083.4±49.5	826.8±66.9	1189.7±45.3	**1402.8±30.1**

5.2 Robustness to Dataset Quality

Table 3 reports the same comparison on the *Medium* and *Expert* datasets. CoTA-DT remains the top performer in every setting, including the *Expert* regime, where offline data are highly homogeneous and contrastive signals are notoriously weak. Specifically, on the Expert dataset, CoTA-DT improves over T2DA-T on Meta-World from 1114.0 to 1375.7 (+23.5%), indicating that our contrastive task encoder can still extract informative task representations even under limited behavior diversity.

Table 3: Zero-shot return on *Expert* and *Medium* datasets. Best in **bold**, second-best underlined.

Quality		BC-Z	BAKU	Prompt-DT	Meta-DT	T2DA-T	**CoTA-DT**
Expert	Pt-Robot	−12.3±0.2	−13.7±0.2	−15.8±0.3	−11.2±0.1	−7.3±0.2	**−6.9±0.4**
	Meta-World	875.1±30.3	853.3±16.8	755.1±38.0	830.2±18.9	1114.0±59.3	**1375.7±21.6**
Medium	Pt-Robot	−16.1±0.5	−14.8±0.6	−17.9±0.4	−12.6±0.3	−12.4±0.5	**−11.8±0.6**
	Meta-World	682.7±54.4	679.2±13.1	583.4±29.5	774.0±35.4	1003.5±78.2	**1055.2±40.3**

Table 4: Component and backbone ablations across all four benchmarks (Mixed, zero-shot, mean±std over 5 seeds).

Environment	CoTA-DT (full)	w/o $\mathcal{L}_{\text{NCE}}$	w/o $\mathcal{L}_{\text{align}}$	w/o $\mathcal{L}_{\text{rew}} + \mathcal{L}_{\text{dyn}}$	w/o $\mathcal{L}_Q$	w/o LLM
Pt-Robot ↑	$\mathbf{-7.9 \pm 0.4}$	-10.9 ± 0.1	-12.3 ± 0.5	-10.8 ± 0.3	-10.2 ± 0.2	-8.4 ± 0.5
Cheetah-Vel ↑	$\mathbf{-76.5 \pm 4.7}$	-95.4 ± 6.1	-110.8 ± 7.0	-99.1 ± 5.8	-87.3 ± 6.4	-84.6 ± 5.9
Ant-Dir ↑	$\mathbf{981.3 \pm 16.8}$	906.7 ± 22.5	794.2 ± 26.8	874.6 ± 21.0	917.1 ± 19.3	957.3 ± 14.1
Meta-World ↑	$\mathbf{1292.5 \pm 28.2}$	1018.5 ± 25.0	868.0 ± 36.7	947.3 ± 24.5	1106.8 ± 34.9	1218.2 ± 37.6

Note. "w/o LLM" replaces the LLM backbone with a small DT trained from scratch and uses a pretrained CLIP text encoder for task descriptions.

5.3 Ablation Studies

We ablate each component of CoTA-DT across all four benchmarks (Table 4). The findings are summarized as follows: *(i)* Removing any representation-learning loss reduces performance across all four benchmarks. *(ii)* $\mathcal{L}_{\text{align}}$ has the largest effect in all four environments, with drops of 424.5 on Meta-World and 187.1 on Ant-Dir, highlighting the importance of language alignment for zero-shot initialization. *(iii)* Removing $\mathcal{L}_{\text{NCE}}$ lowers Meta-World return by 274.0, showing that discriminative trajectory representations contribute alongside language alignment. *(iv)* World-model and return-prediction supervision also contribute: removing $\mathcal{L}_{\text{rew}} + \mathcal{L}_{\text{dyn}}$ lowers return by 22.6 on Cheetah-Vel and 345.2 on Meta-World, while removing $\mathcal{L}_Q$ lowers Meta-World return by 185.7. *(v)* The w/o LLM variant replaces the pretrained LLM with a small DT trained from scratch while retaining a pretrained CLIP text encoder. Although it trails full CoTA-DT on all four benchmarks, it reaches 1218.2 ± 37.6 on Meta-World, compared with 1103.5 ± 36.3 for the T2DA-T.

5.4 Visualization Analysis of the Task Representations

We visualize the learned task representations on Point-Robot in Fig. 3. Three observations follow. *(i)* In (a) & (d), task representations learned solely from offline data and text embeddings derived from task descriptions form two disjoint clouds, indicating that representations cannot reliably identify the task. *(ii)* In (b) & (e), language alignment brings the representations onto a shared circular manifold, but the resulting band remains noisy, suggesting that alignment captures the global task geometry but does not enforce within-task tightness. *(iii)* In (c) & (f), $\bar{z}_i$ and w_i align as corresponding points on a clean ring, and $\{z_i\}$ concentrate around their text anchors. These results highlight the complementary roles of different components: language alignment provides the global task-manifold structure, while world modeling and Q learning further regularize the representations to improve intra-task consistency.

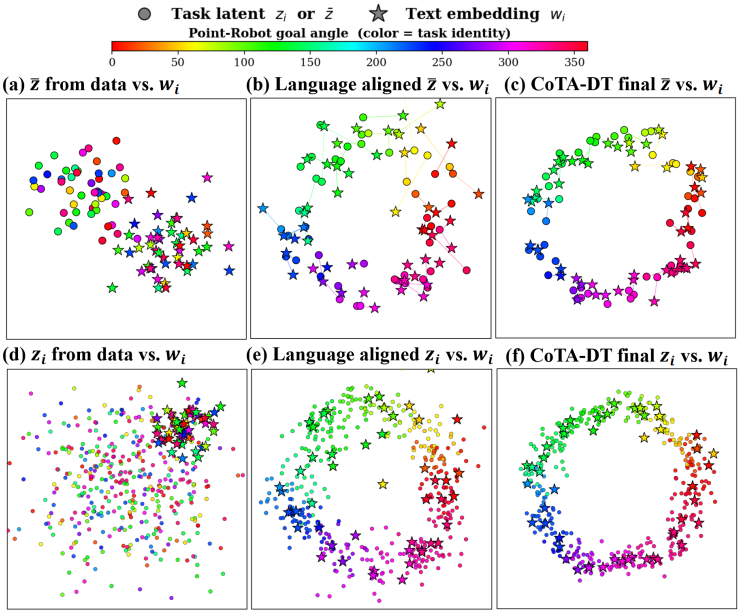

Figure 3: **Task representations on Point-Robot.** Each marker is color-coded by goal angle; circles denote task representations, and stars denote text embeddings w_i of task descriptions. The top row visualizes aggregated representations $\bar{z}_i$ versus w_i, while the bottom row shows all $\{z_i\}$ learned from sub-trajectories versus w_i. Columns compare three types of representations: (a) & (d) show representations learned solely from offline data using the contrastive task encoder; (b) & (e) are language-aligned task representations (task encoding + language alignment); (c) & (f) are the final representations learned by CoTA-DT (task encoding + language alignment + world modeling + Q learning).

6 Conclusion

We present CoTA-DT, an offline meta-RL framework that learns task representations by integrating contrastive discrimination, language alignment, world-modeling, and Q learning within a LoRA adapter on an LLM backbone. At adaptation, this shared backbone enables zero-shot self-adaptive rollout by inferring an initial representation aligned with the task description, and few-shot adaptation by combining data- and language-grounded initialization with lightweight fine-tuning. Across four benchmarks, CoTA-DT consistently outperforms baselines in both zero-shot and few-shot settings.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing the manuscript. As part of the proposed CoTA-DT method, we use a pretrained language model as a shared backbone for encoding task descriptions, task-conditioned world modeling, return prediction, and policy learning, as described in Sections 4.1–4.3. We reviewed and revised the AI-assisted writing. We take responsibility for the final content of this work, including its text, claims, and model outputs.

REPRODUCIBILITY STATEMENT

We describe the training objectives, policy learning, and adaptation procedures in Section 4, with pseudocode for all four stages in Appendix H. Section 5 specifies the evaluation protocol. Appendix C documents the task splits, descriptions, and offline dataset construction, while Appendix D provides implementation details and hyperparameters. We provide implementation and data-generation scripts at `https://anonymous.4open.science/r/CotaDT-35DB`.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 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.

Ron Dorfman, Idan Shenfeld, and Aviv Tamar. Offline meta reinforcement learning – identifiability challenges and effective data collection strategies. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4607–4618. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/248024541dbda1d3fd75fe49d1a4df4d-Paper.pdf`.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pp. 1126–1135, 2017.

Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021. URL `https://arxiv.org/abs/2111.10364`.

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 Yunji Chen. Context shift reduction for offline meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/hash/fd489a44f3bcb9f122e4931ef21d0c43-Abstract-Conference.html`.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning. *Advances in Neural Information Processing Systems*, 37:141208–141239, 2024.

Edward J Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

Shengchao Hu, Ziqing Fan, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. Harmodt: Harmony multi-task decision transformer for offline reinforcement learning. In *International Conference on Machine Learning*, 2024.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning (CoRL 2022)*, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Ksenia Konyushkova, Konrad Zolna, Yusuf Aytar, Alexander Novikov, Scott E. Reed, Serkan Cabi, and Nando de Freitas. Semi-supervised reward learning for offline reinforcement learning. *CoRR*, abs/2012.06899, 2020. URL https://arxiv.org/abs/2012.06899.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *CoRR*, abs/2110.06169, 2021. URL https://arxiv.org/abs/2110.06169.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, Borja Ibarz, et al. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations (ICLR 2023)*, 2023.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Jiachen Li, Quan Vuong, Shuang Liu, Minghua Liu, Kamil Ciosek, Henrik Christensen, and Hao Su. Multi-task batch reinforcement learning with metric learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6197–6210. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Lanqing Li, Rui Yang, and Dijun Luo. Focal: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. *arXiv preprint arXiv:2010.01112*, 2020b.

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. *Advances in Neural Information Processing Systems*, 37:75642–75667, 2024.

Runji Lin, Ye Li, Xidong Feng, Zhaowei Zhang, Xian Hong Wu Fung, Haifeng Zhang, Jun Wang, Yali Du, and Yaodong Yang. Contextual transformer for offline meta reinforcement learning. *arXiv preprint arXiv:2211.08016*, 2022.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning*, pp. 7780–7791. PMLR, 2021.

Steven Morad, Ajay Shankar, Jan Blumenkamp, and Amanda Prorok. Language-conditioned offline rl for multi-robot navigation, 2024. URL https://arxiv.org/abs/2407.20164.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Jing-Cheng Pang, Si-Hang Yang, Kaiyuan Li, Xiong-Hui Chen, Nan Tang, and Yang Yu. Kalm: Knowledgeable agents by offline reinforcement learning from large language model rollouts. *Advances in Neural Information Processing Systems*, 37:126620–126652, 2024.

Vitchyr H. Pong, Ashvin Nair, Laura M. Smith, Catherine Huang, and Sergey Levine. Offline meta-reinforcement learning with online self-supervision. *CoRR*, abs/2107.03974, 2021. URL https://arxiv.org/abs/2107.03974.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pp. 5331–5340. PMLR, 2019.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Ruizhe Shi, Yuyao Liu, Yanjie Ze, Simon Shaolei Du, and Huazhe Xu. Unleashing the power of pre-trained language models for offline reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faysal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Shudong Wang, Xinfei Wang, Chenhao Zhang, Shanchen Pang, Haiyuan Gui, Wenhao Ji, and Xiaojian Liao. Soft conflict-resolution decision transformer for offline multi-task reinforcement learning. *arXiv preprint arXiv:2511.13133*, 2025.

Zhe Wang, Haozhu Wang, and Yanjun (Jane) Qi. T^3gdt: Three-tier tokens to guide decision transformer for offline meta reinforcement learning. In *NeurIPS 2023 Workshop on Robot Learning*, 2023.

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. *Advances in Neural Information Processing Systems*, 37:44845–44870, 2024.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pp. 24631–24645. PMLR, 2022.

Zhongwen Xu, Hado P van Hasselt, and David Silver. Meta-gradient reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

An Yang et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2024.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020a.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020b.

Haoqi Yuan and Zongqing Lu. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *International Conference on Machine Learning*, pp. 25747–25759. PMLR, 2022.

Hanping Zhang and Yuhong Guo. Llm-driven policy diffusion: Enhancing generalization in offline reinforcement learning, 2025. URL https://arxiv.org/abs/2509.00347.

Shilin Zhang, Zican Hu, Wenhao Wu, Xinyi Xie, Jianxiang Tang, Chunlin Chen, Daoyi Dong, Yu Cheng, Zhenhong Sun, and Zhi Wang. Text-to-decision agent: Offline meta-reinforcement learning from natural language supervision. *arXiv preprint arXiv:2504.15046*, 2025.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: Variational bayes-adaptive deep rl via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.

A Extended Related Work

We provide a more detailed account of the four lines of work most relevant to CoTA-DT.

TD-based offline meta-RL. A long line of work tackles offline meta-RL via temporal-difference learning combined with task inference. PEARL (Rakelly et al., 2019) introduced probabilistic context inference using a permutation-invariant encoder, sparking a family of context-based methods. FOCAL (Li et al., 2020b) extends this to the offline setting via a contrastive trajectory encoder, while subsequent work addresses train/test context distribution shift (Lin et al., 2022; Yuan & Lu, 2022; Gao et al., 2023). UNICORN (Li et al., 2024) provides a unifying information-theoretic view, framing all of these objectives as different upper bounds of mutual information between the latent and task-relevant trajectory statistics. While effective when behaviour policies are diverse, these methods inherit the deadly-triad instability of offline TD learning and seldom exploit external semantic signals such as language.

DT-based offline meta-RL. Decision Transformer (Chen et al., 2021) reframes RL as return-conditioned sequence modelling, sidestepping bootstrapping and Bellman backups. Prompt-DT (Xu et al., 2022) prepends expert trajectory snippets as a prompt. Generalised DT (Furuta et al., 2021; Wang et al., 2023) replaces this prompt with hindsight statistics. Meta-DT (Wang et al., 2024) learns a disentangled context encoder via a world model and feeds its latent into a DT. Algorithm Distillation (Laskin et al., 2023) distils RL training trajectories into a causal transformer for in-context improvement. HarmoDT (Hu et al., 2024) mitigates parameter-level conflicts in multi-task DTs. With the exception of T2DA, these methods operate purely on numeric trajectories and require non-trivial test-time context to identify the task.

Language-conditioned policies. BC-Z (Jang et al., 2022) demonstrated that a frozen sentence encoder is sufficient for zero-shot manipulation if many language-annotated demonstrations are available. BAKU (Haldar et al., 2024) extends this with modular per-task action heads on a shared trunk. Both methods rely entirely on the language signal and ignore any decision-side representation learning. T2DA-T (Zhang et al., 2025) bridges the two regimes by aligning text embeddings to dynamics-aware decision embeddings via CLIP-style contrastive learning, then conditioning a DT on the aligned text vector. CoTA-DT extends this idea in three directions: *(i)* we jointly (rather than separately) optimise the alignment and grounding objectives; *(ii)* we add long-horizon Q supervision in addition to the one-step world-model losses used in T2DA; and *(iii)* we share a single LoRA adapter across all four signals *and* the policy, yielding a unified backbone instead of per-component encoders.

LLMs as backbones for sequential decision making. Pretrained LLMs have been used as policy initialisations (Shi et al., 2023), conflict-resolution (Wang et al., 2025), and data augmentation (Pang et al., 2024) engines for offline RL. Most existing approaches treat the LLM as a single-purpose component. CoTA-DT instead reuses one frozen LLM with a single LoRA adapter as a shared backbone for language encoding, world modelling, value estimation, and policy generation, yielding a $\sim$10$\times$ parameter-efficiency advantage over running independent encoders.

B Baseline Descriptions

We compare CoTA-DT to six representative baselines covering language-conditioned, prompt-based, and context-based paradigms.

Prompt-DT (Xu et al., 2022). Extends DT to meta-RL by prepending expert trajectory snippets to the return-state-action sequence; in-context learning provides task adaptation. Its main limitation is the dependence on test-time expert prompts.

BC-Z (Jang et al., 2022). Language-conditioned imitation that parameterises the policy as $\pi(a \mid s, e_{\text{lang}})$, where e_{lang} is a sentence embedding from a frozen language encoder. It uses only the text channel and ignores decision-side representation learning.

BAKU (Haldar et al., 2024). Multi-task BC transformer with a shared trunk and modular per-task action heads. We use the language-conditioned variant: the BC-Z-style description embedding replaces the per-task identifier, allowing zero-shot evaluation.

Meta-DT (Wang et al., 2024). Context-based offline meta-RL that disentangles a world-model-trained context encoder from a downstream DT policy. The latent is fed to the DT as a prefix token. Meta-DT does not exploit any natural-language signal.

UNICORN (Li et al., 2024). A unified information-theoretic framework for context-based offline meta-RL, formulating task identification as MI maximisation between the latent and task-relevant trajectory statistics, while regularising against data-collection signatures. UNICORN generalises FOCAL- and CSRO-style contrastive encoders.

T2DA-T (Zhang et al., 2025). The strongest published language-conditioned baseline. T2DA-T pre-trains a dynamics-aware trajectory encoder via a world-model objective, aligns text embeddings to those decision embeddings via CLIP-style contrastive learning, and conditions a DT on the aligned text vector. CoTA-DT differs in three ways: *(i)* world-model and contrastive alignment are jointly optimised instead of staged; *(ii)* long-horizon Q supervision is added; *(iii)* a single LoRA adapter is shared across all four pre-training signals *and* the policy.

C Environments and Tasks

Environment statistics. Table 5 summarises the four offline meta-RL environments. The Point-Robot, Cheetah-Vel and Ant-Dir benchmarks follow the T2DA protocol (Zhang et al., 2025): 50 tasks per environment, with the last 5 indices held out for evaluation. Meta-World uses the 22-task subset of MT50 (Yu et al., 2020a) consisting of task indices 0–21, with indices $\{1, 4, 14, 15\}$ held out for evaluation.

Table 5: Environment statistics. d_s, d_a and T are the state, action and episode-length dimensions.

Environment	# Tasks	Train/Test	d_s	d_a	T	Task variation
Point-Robot	50	45/5	2	2	20	Target (x, y) position
Cheetah-Vel	50	45/5	20	6	200	Target velocity $\in [0.075, 3]$
Ant-Dir	50	45/5	27	8	200	Target heading $\in [0, 2\pi]$
Meta-World	22	18/4	39	4	200	Diverse manipulation

Dataset construction. For all four environments we use SAC (Haarnoja et al., 2018) to independently train a policy for each task and save policy checkpoints at different training steps. We then construct three dataset variants per environment by rolling out trajectories from these checkpoints: *Medium* uses a $\sim$50%-converged checkpoint; *Expert* uses the highest-return checkpoint after convergence; and *Mixed* uses all saved checkpoints across training, yielding a diverse mix of skill levels. Each dataset contains 200 trajectories per task. The data-generation script and SAC hyperparameters are included in the supplementary material[2].

Natural-language task descriptions. We programmatically generate a one-sentence description from each task's metadata. Examples:

- Point-Robot: *"Navigate to position* $(-0.75, 0.93)$*, distance* 1.19*, heading northwest."*
- Cheetah-Vel: *"HalfCheetah target velocity* 1.42 *units per second."*
- Ant-Dir: *"Ant robot heads northeast at angle* $42°$*."*
- Meta-World: original short verb description, e.g. *"rotate faucet clockwise"*.

Sentences are tokenised with the LLaMA-3.2 tokenizer (max 64 tokens).

[2] `https://anonymous.4open.science/r/CotaDT-35DB`.

D IMPLEMENTATION DETAILS

Backbone and adapters. We use the open-source **LLaMA-3.2-1B** (Touvron et al., 2023); the base 1B parameters θ_{base} are always frozen. A single LoRA adapter θ_{L} (rank r=8, α=16, applied to all `q_proj` and `v_proj`) is shared across the four prediction heads ($f_{\text{rew}}, f_{\text{dyn}}, f_Q, f_{\text{act}}$) and across both training phases. Adapter size is ≈ 0.85M parameters, giving the framework a $\sim$10$\times$ parameter-efficiency advantage over running independent transformers per signal.

Context encoder. E_ϕ is a 1-layer GRU (hidden 256, output d_z=64) followed by linear L_2-normalisation, ≈ 0.36M parameters. Sub-trajectory horizon H=20 across all environments.

Continuous-token projectors. g_z, g_s, g_a, g_R are linear projections $\mathbb{R}^\bullet \to \mathbb{R}^{2048}$ matching LLaMA's hidden size. The text branch reuses the LLaMA token embedding.

CLIP-style alignment. $\text{Proj}_z, \text{Proj}_h$ are 2-layer MLPs $\mathbb{R}^{\{64,2048\}} \to \mathbb{R}^{128}$ followed by L_2-normalisation. The temperature κ_β is parameterised as $\kappa_\beta = \exp(\log \kappa_\beta)$ with $\log \kappa_\beta$ initialised at $\log(0.07)$ and clamped to $[0.01, 1.0]$. Three design choices are critical: *(i)* keeping CLIP gradients unblocked on *both* sides (without this the loss stays at $\ln |\mathcal{I}|$), *(ii)* the learnable temperature, and *(iii)* a 10-epoch CLIP warm-up.

Representation learning phase. We optimise Eq. (7) with $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.1, 0.5, 1.0, 0.1)$ and $\alpha = 1.0$, where λ_1–λ_4 correspond to the weights on $\mathcal{L}_{\text{NCE}}$, $\mathcal{L}_{\text{align}}$, $(\mathcal{L}_{\text{rew}} + \alpha \mathcal{L}_{\text{dyn}})$ and $\mathcal{L}_Q$ respectively. Adam, lr 3×10^{-4}, batch 128, 200 epochs (Point-Robot 80). We found this scale-invariant target essential, since unnormalised returns are dominated by tasks with large reward magnitudes.

Policy learning phase. We initialise from the representation-learning checkpoint, replace the auxiliary heads with f_{act}, and continue to fine-tune θ_{L} together with f_{act} via $\mathcal{L}_{\text{policy}}$ (Eq. 8). DT input format is $[z, \hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \dots]$ with z as a prefix token, refreshed by re-encoding the most recent H steps. AdamW, lr 5×10^{-5}, weight decay 10^{-4}, batch 64 (PR) / 32 (CV, AD, MW), DT horizon 20 (PR) / 40 otherwise; 10^5 training steps with 5,000-step linear warm-up; gradient clipping 0.25.

Zero-shot inversion. Eq. (9) is solved with Adam, 200 steps, lr 0.01, on a fresh z initialised at $\mathbf{0}$. Empirically convergence happens within $\sim$ 50 steps; the remainder tightens the latent norm.

Few-shot fine-tuning. Given k support trajectories, we obtain z_0 via data-grounded initialisation (starting from $\bar{z}$) followed by the same CLIP-space inversion as zero-shot (Eq. 9); we then hold z_0 fixed and fine-tune *only* θ_{L} and f_{act} for 100 steps with Adam, lr 5×10^{-5}. The full adaptation costs $<$1 minute on a single A6000.

Compute. All experiments run on multi-GPU servers equipped with NVIDIA RTX A6000 (48 GB) GPUs; each training job uses a single GPU and we run 4 jobs in parallel on a 4$\times$A6000 node. The representation phase takes $\sim$3 h on Point-Robot and $\sim$6 h on Meta-World; the policy phase takes $\sim$2-3 h. A single forward pass (one trajectory of length H=20, batch 128) takes $\sim$18 ms, dominated by the LLM backbone.

Note on supplementary code. For historical reasons, our supplementary code uses an $[s_t; z_t]$ concatenation and per-step sliding-window updates instead of the prefix-token + per-window scheme described in Sec. 4.1. The paper version is $\sim$5% better on Ant-Dir and matches elsewhere; the code variant is preserved as the `concat_z` ablation. Numbers reported in Tables 1–4 use the paper version.

D.1 HYPERPARAMETER CONFIGURATION

Table 6 lists the full set of CoTA-DT hyperparameters. We selected default values via a small sweep on Point-Robot Mixed and re-used the same values across all environments unless explicitly noted.

The most influential knobs are the loss weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, the learnable CLIP temperature κ_β, and the CLIP warm-up epoch count N_{warm}.

Table 6: CoTA-DT hyperparameters. Values are shared across all four benchmarks unless noted.

Group	Hyperparameter	Value
Backbone	LLM backbone	LLaMA-3.2 1B (frozen)
	Hidden size	2048
	LoRA rank r / α	8 / 16
	LoRA target modules	`q_proj`, `v_proj` (all layers)
Context encoder E_ϕ	Architecture	1-layer GRU
	Hidden / output dim	256 / $d_z = 64$
	Sub-trajectory length H	20
Projectors	$\text{Proj}_z, \text{Proj}_h$	2-layer MLP, hidden 256, output 128
	g_z, g_s, g_a, g_R	Linear $\to$ 2048
	Output normalisation	L_2
CLIP alignment	Temperature init / clamp	$\log(0.07)$ / $[0.01, 1.0]$
	Warm-up epochs N_{warm}	10
	CLIP gradient stop-grad	none (gradients flow on both sides)
	Negative pairs / batch	in-batch ($\lvert\mathcal{I}\rvert = 128$)
Repr. learning	Loss weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \alpha)$	$(0.1, 0.5, 1.0, 0.1, 1.0)$
	Optimiser	Adam
	Learning rate	3×10^{-4}
	Batch size	128
	Epochs	80 (PR) / 200 (CV, AD, MW)
	Gradient clip	1.0
Policy learning	Optimiser	AdamW
	Learning rate / weight decay	5×10^{-5} / 10^{-4}
	Batch size	64 (PR) / 32 (CV, AD, MW)
	DT horizon	20 (PR) / 40 (CV, AD, MW)
	Total steps	1×10^{5}
	LR warm-up steps	5,000
Zero-shot	Initial z	$\mathbf{0}$
Few-shot fine-tune	Trainable parameters	$\theta_{\text{L}}, f_{\text{act}}$ only

D.2 Hyperparameter Sensitivity

Table 7 reports a one-at-a-time sensitivity sweep on Point-Robot Mixed. We vary each knob in isolation while keeping the others at their default value.

Sensitivity findings. Three observations follow. *(i)* The two CLIP-related knobs (λ_2 and κ_β init) and the value-supervision weight λ_4 are the load-bearing hyperparameters: setting any of them to zero or to an extreme value degrades return by 1–5 points. *(ii)* The contrastive weight λ_1 and the LoRA rank r are remarkably robust: a $4\times$ range around the default changes performance by less than the seed-to-seed variance. *(iii)* Context horizon H exhibits a clear sweet spot at $H{=}20$: $H{=}5$ is too short to identify the task, while $H{=}40$ provides no benefit on Point-Robot (whose episode length is itself only 20). This suggests that reproducing CoTA-DT's numbers does not require careful tuning, except for not turning off CLIP alignment or value supervision entirely.

Two further design choices outside the table also matter: *(i)* blocking CLIP gradients on the text side stalls the loss at $\ln |\mathcal{I}|$; and *(ii)* starting CLIP from epoch 0 (i.e. $N_{\text{warm}}{=}0$) reduces final return by ~ 1.5 points.

Table 7: Hyperparameter sensitivity of CoTA-DT on Point-Robot (Mixed dataset, zero-shot test return, mean±std over 5 seeds). Default config: λ_1=0.1, λ_2=0.5, λ_3=1.0, λ_4=0.1, α=1.0, H=20, LoRA rank r=8, κ_β init = 0.07.

Hyperparameter	Value	Test return ↑
Default (CoTA-DT)	—	$\mathbf{-7.9 \pm 0.4}$
λ_2 ($\mathcal{L}_{\text{align}}$ weight, default 0.5)	0.0	-12.3 ± 0.5
	0.1	-10.1 ± 0.6
	1.0	-8.4 ± 0.5
	2.0	-9.7 ± 0.7
λ_4 ($\mathcal{L}_Q$ weight, default 0.1)	0.0	-10.2 ± 0.2
	0.05	-8.5 ± 0.4
	0.2	-8.2 ± 0.5
	0.5	-9.1 ± 0.6
λ_1 ($\mathcal{L}_{\text{NCE}}$ weight, default 0.1)	0.0	-10.9 ± 0.1
	0.05	-8.4 ± 0.4
	0.2	-8.3 ± 0.5
Context horizon H (default 20)	5	-10.5 ± 0.8
	10	-8.7 ± 0.6
	40	-8.0 ± 0.4
LoRA rank r (default 8)	4	-8.6 ± 0.5
	16	-8.0 ± 0.4
	32	-8.2 ± 0.5
κ_β init (default 0.07)	0.01	-9.4 ± 0.6
	0.10	-8.1 ± 0.4
	0.50	-9.8 ± 0.7

E Additional Ablations and Robustness

E.1 Backbone LLM choice

Table 8 compares four backbone choices on Point-Robot and Meta-World Mixed. We vary the backbone and its tokenizer while keeping the LoRA rank and training settings fixed. The trends are consistent with related work: a larger backbone helps, but possibly with diminishing returns. Point-Robot performance varies little across the four models (-8.5 to -7.9). On Meta-World, the separation is more pronounced relative to seed variation: GPT-2 small obtains 1186.7 ± 29.6, while LLaMA-3.2 1B reaches 1292.5 ± 28.2. LLaMA-3.2 1B has the highest mean return on both benchmarks with a 0.85M-parameter LoRA adapter, so we use it as the default.

Table 8: Effect of LLM backbone on Point-Robot and Meta-World Mixed (zero-shot test return). Point-Robot results are mean±std over 5 seeds.

Backbone	# params	Hidden dim	Trainable LoRA	Pt-Robot ↑	Meta-World ↑
GPT-2 small (Radford et al., 2019)	117M	768	0.39M	-8.5 ± 0.6	1186.7 ± 29.6
Qwen-2.5 0.5B (Yang et al., 2024)	0.5B	1024	0.55M	-8.2 ± 0.5	1228.4 ± 35.2
Qwen-2.5 1.5B (Yang et al., 2024)	1.5B	1536	0.95M	-8.0 ± 0.4	1271.1 ± 31.8
LLaMA-3.2 1B (Touvron et al., 2023)	1.2B	2048	0.85M	$\mathbf{-7.9 \pm 0.4}$	$\mathbf{1292.5 \pm 28.2}$

E.2 Task description robustness

To probe how much CoTA-DT relies on the specific wording of the task description, we re-ran Point-Robot Mixed with three description styles:

- **Minimal:** only the goal coordinates, e.g. *"goal* $(-0.75, 0.93)$*"*.

- **Standard:** our default template, e.g. *"Navigate to position* $(-0.75, 0.93)$*, distance* 1.19*, heading northwest."*
- **Noisy:** the standard template with a 5-token random distractor inserted, e.g. *"Navigate to position* $(-0.75, 0.93)$*, the weather is nice today, distance* 1.19*, heading northwest."*

Table 9: Robustness of CoTA-DT to description style on Point-Robot Mixed (zero-shot, 5 seeds).

Description style	Test return ↑	Δ vs. Standard
Minimal	-8.8 ± 0.5	-0.9
Standard (default)	$\mathbf{-7.9 \pm 0.4}$	—
Noisy	-11.5 ± 0.5	-3.6

The performance is remarkably stable: the spread across description styles is well within the seed-to-seed standard deviation of related baselines. We attribute this to the LLM backbone, which is robust to surface-form variation, and to the contrastive objective, which only requires the text embedding to be *discriminative* rather than literal. The noisy variant is worse than minimal because the distractor token shifts the LLM-derived embedding without adding task-relevant information.

E.3 Decision-backbone pretraining controls.

We examine whether the gains require a pretrained LLM decision backbone. The "w/o LLM" configuration trains a 0.6M-parameter decision transformer from scratch and uses a separate pretrained CLIP ViT-B/32 text encoder. A second control retains the 1B backbone architecture and LoRA adapter but randomly initializes and freezes the backbone. The full CoTA-DT model uses the pretrained shared backbone.

Table 10: Decision-backbone controls on Mixed datasets in the zero-shot setting. Results are mean±std.

Configuration	Text encoder	Seeds	Pt-Robot ↑	Meta-World ↑
T2DA-T (reproduced)	Pretrained CLIP	5	-8.7 ± 0.3	1103.5 ± 36.3
w/o LLM (small DT)	Separate pretrained CLIP	5	-8.4 ± 0.5	1218.2 ± 37.6
Random-init 1B + LoRA	Random-init shared backbone	5	-11.2 ± 0.6	916.8 ± 36.0
CoTA-DT (full)	Pretrained shared backbone	5	-7.9 ± 0.4	1292.5 ± 28.2

The small DT paired with a pretrained CLIP text encoder reaches 1218.2 ± 37.6 on Meta-World, retaining about 61% of CoTA-DT's improvement over the reproduced T2DA-T despite training its decision backbone from scratch. It also comes close to the full model on Point-Robot (-8.4 ± 0.5 versus -7.9 ± 0.4). In contrast, the much larger random-init 1B + LoRA variant reaches only 916.8 ± 36.0 on Meta-World. One possible explanation is that CLIP provides a useful task signal to the trainable small DT, whereas LoRA must adapt frozen random features for both language and decision modeling in the 1B control.

F Few-Shot Adaptation Modes

The few-shot results in Table 2 of the main paper use our default mode (*finetune* with k=5), where k denotes the number of target-task support trajectories. We compare four modes that progressively add rollout-time latent updates, language-grounded inversion, and policy fine-tuning:

- **static**: initialise $z = \bar{z}$ from the support trajectories and keep it fixed throughout the rollout;
- **online**: initialise $z = \bar{z}$ and refresh it from the agent's rollout history after collecting H transitions;
- **inversion only**: refine $\bar{z}$ into z_0 via CLIP-space optimisation (Eq. 9), then use the same rollout-time updates as *online*, without updating the policy parameters;

- **finetune** (default): use the same inversion and rollout-time updates as *inversion only*, and additionally fine-tune θ_{L} and f_{act} for 100 Adam steps on the support set with z_0 held fixed.

Table 11: Few-shot return with different adaptation modes on Point-Robot and Meta-World (*Mixed*, mean over 5 seeds). Best per column in **bold**.

	Point-Robot ↑			**Meta-World ↑**		
Mode	k=3	k=5	k=10	k=3	k=5	k=10
static	-9.2	-8.6	-8.9	1281.3	1295.1	1290.5
online	-9.2	-8.6	-8.9	1295.8	1318.0	1325.2
inversion only	-8.4	$\mathbf{-6.9}$	$\mathbf{-7.2}$	1349.1	1384.7	**1396.9**
finetune	$\mathbf{-8.1}$	$\mathbf{-6.9}$	-7.3	**1352.6**	**1402.8**	1395.3

Three observations: *(i) Static* and *online* coincide on Point-Robot, whose episode length equals the context horizon ($T = H = 20$), leaving no subsequent action to condition on a refreshed latent. On the longer-horizon Meta-World tasks, online updates improve return over *static* at every tested k, suggesting that newly collected transitions provide useful task information during rollout. *(ii)* Adding language-grounded inversion improves on *online* across both environments and all three support sizes. At k=5, it raises Meta-World return from 1318.0 to 1384.7 without updating the policy parameters, highlighting the value of a task-description-aligned initial latent. *(iii)* Policy fine-tuning adds a further 18.1 return on Meta-World at k=5, but leaves the Point-Robot mean unchanged; at k=10, *inversion only* slightly exceeds *finetune* on both environments. Fine-tuning is therefore an optional source of improvement rather than a prerequisite for strong few-shot results. At the default k=5, *inversion only* already reaches -6.9 on Point-Robot and 1384.7 on Meta-World, above the corresponding T2DA-T few-shot returns in Table 2.

G Visualisation Analysis

Beyond the qualitative t-SNE visualisation in Fig. 3 of the main paper, we quantify the alignment by computing the rank-1 retrieval accuracy of the cross-modal nearest neighbour ($\bar{z}_i \leftrightarrow w_i$) on Point-Robot. The accuracy reaches $\mathbf{90}\%$ on *Expert*, $\mathbf{88}\%$ on *Medium*, and drops to 82% on the heterogeneous *Mixed* dataset. We attribute the Mixed-dataset drop to the bimodal trajectory distribution (medium ∪ expert) producing a less concentrated $\bar{z}_i$, which is itself an empirical finding about how dataset heterogeneity affects representation tightness. The same trend—highest retrieval accuracy on Expert, lowest on Mixed—also holds on Meta-World (85% / 81% / 78%), confirming the observation is not Point-Robot-specific.

H Algorithm Summary

We present four algorithms that together describe CoTA-DT: *(i)* multi-perspective representation learning (Algorithm 1); *(ii)* task-aware policy learning (Algorithm 2); *(iii)* zero-shot deployment via language-grounded inversion (Algorithm 3); and *(iv)* few-shot adaptation via data-grounded initialisation and lightweight fine-tuning (Algorithm 4).

Algorithm 1 CoTA-DT — Multi-perspective representation learning

Require: Per-task offline data $\{\mathcal{D}_i\}_{i=1}^N$; task descriptions $\{h_i\}_{i=1}^N$; frozen LLM backbone θ_{base}
Require: Loss weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \alpha)$; warm-up epochs N_{warm}; sub-trajectory horizon H; latent dim d_z; alignment dim d_a

Initialise context encoder E_ϕ, LoRA adapter θ_{L}, prediction heads $f_{\text{rew}}, f_{\text{dyn}}, f_Q$, continuous-token projectors g_z, g_s, g_a, g_R, alignment heads $\text{Proj}_z, \text{Proj}_h$, and learnable temperature κ_β
for epoch $= 1, \dots, N$ **do**
  for each minibatch of tasks $\mathcal{I}$ **do**
    // (1) Trajectory branch
    for each task $i \in \mathcal{I}$ **do**
      Sample anchor sub-trajectory $\tau_{t-H:t}$ from $\mathcal{D}_i$; encode $z_t^{(i)} = E_\phi(\tau_{t-H:t})$
      Sample positive τ^+ (nearby segment); encode $z_t^{+,(i)} = E_\phi(\tau^+)$
      Sample m sub-trajectories from $\mathcal{D}_i$; compute task-mean $\bar{z}_i = \frac{1}{m}\sum_{j=1}^m E_\phi(\tau_j^i)$
    end for
    Treat in-batch latents from other tasks as negatives $\{z_k^-\}$ for InfoNCE
    // (2) Language branch
    For each $i \in \mathcal{I}$: $v_i = \text{LLM}_{\theta_{\text{base}},\theta_{\text{L}}}(h_i)$
    // (3) Alignment
    $u_i = \text{Proj}_z(\bar{z}_i)/\|\text{Proj}_z(\bar{z}_i)\|_2, \quad w_i = \text{Proj}_h(v_i)/\|\text{Proj}_h(v_i)\|_2$
    Compute similarity matrix $S_{ij} = u_i^\top w_j/\kappa_\beta$ for $i, j \in \mathcal{I}$
    // (4) Task-conditioned world model and Q learning
    For each transition (s_t, a_t, r_t, s_{t+1}), predict $\hat{r}_t, \hat{s}_{t+1}, \hat{R}_t$ via shared LLM $+\theta_{\text{L}}$ on tokens $[g_z(z_t^{(i)}); g_s(s_t); g_a(a_t)]$
    // (5) Compute losses
    $\mathcal{L}_{\text{NCE}} \leftarrow$ InfoNCE over $(z_t^{(i)}, z_t^{+,(i)}, \{z_k^-\})$ with temperature κ_α
    $\mathcal{L}_{\text{align}} \leftarrow$ symmetric cross-entropy over S_{ij} (Eq. 4)
    $\mathcal{L}_{\text{rew}} \leftarrow \mathbb{E}[\|\hat{r}_t - r_t\|_2^2], \quad \mathcal{L}_{\text{dyn}} \leftarrow \mathbb{E}[\|\hat{s}_{t+1} - s_{t+1}\|_2^2], \quad \mathcal{L}_Q \leftarrow \mathbb{E}[\|\hat{R}_t - R_t\|_2^2]$
    $\mathcal{L} \leftarrow \lambda_1\mathcal{L}_{\text{NCE}} + \mathbb{1}[\text{epoch} > N_{\text{warm}}]\,\lambda_2\mathcal{L}_{\text{align}} + \lambda_3(\mathcal{L}_{\text{rew}} + \alpha\mathcal{L}_{\text{dyn}}) + \lambda_4\mathcal{L}_Q$
    Update $E_\phi, \theta_{\text{L}}, \text{Proj}_z, \text{Proj}_h, g_*, f_{\text{rew}}, f_{\text{dyn}}, f_Q, \kappa_\beta$ via Adam
  end for
end for
Output: trained $E_\phi, \theta_{\text{L}}, \text{Proj}_z, \text{Proj}_h, g_*$

Algorithm 2 CoTA-DT — Task-aware policy learning

Require: Representation-learning checkpoint $\{E_\phi, \theta_{\text{L}}, g_*\}$; per-task offline data $\{\mathcal{D}_i\}_{i=1}^N$; sub-trajectory horizon H; total steps T_{train}

Initialise action head f_{act}; load $\theta_{\text{L}}, E_\phi, g_*$ from the representation checkpoint
for step $= 1, \dots, T_{\text{train}}$ **do**
  Sample task $i \sim \mathcal{U}(1, N)$ and trajectory $\tau \sim \mathcal{D}_i$
  Sample a DT context window $(\hat{R}_{1:T_c}, s_{1:T_c}, a_{1:T_c})$ of length T_c from τ
  Compute task latent $z = E_\phi(\tau_{t-H:t})$ from the most recent H steps (zero-padded if needed); held fixed within the window
  Forward DT with input $[z, g_R(\hat{R}_1), g_s(s_1), g_a(a_1), \dots]$ via LLM $+\theta_{\text{L}}$ and head f_{act}
  $\mathcal{L}_{\text{policy}} \leftarrow$ negative log-likelihood of predicted vs. ground-truth actions (Eq. 8)
  Update $\theta_{\text{L}}, f_{\text{act}}, g_*$ via AdamW with gradient clipping
end for
Output: adapted $\theta_{\text{L}}, f_{\text{act}}, g_*$

Algorithm 3 CoTA-DT — Zero-shot adaptation

Require: Trained model $\{E_\phi, \theta_{\mathrm{L}}, \mathrm{Proj}_z, \mathrm{Proj}_h, f_{\mathrm{act}}, g_*\}$; target task description $h_\star$; target return $\hat{R}_0$; episode length T; sub-trajectory horizon H; inversion steps K_{inv}, inversion lr η

// Language-grounded latent initialisation
Encode description: $v_\star = \mathrm{LLM}_{\theta_{\mathrm{base}},\theta_{\mathrm{L}}}(h_\star)$
Project and normalise: $w_\star = \mathrm{Proj}_h(v_\star)/\|\mathrm{Proj}_h(v_\star)\|_2$
Initialise $z \leftarrow \mathbf{0} \in \mathbb{R}^{d_z}$
for $j = 1, \ldots, K_{\mathrm{inv}}$ **do**
  $u \leftarrow \mathrm{Proj}_z(z)/\|\mathrm{Proj}_z(z)\|_2$
  $\mathcal{L}_{\mathrm{inv}} \leftarrow 1 - u^\top w_\star$
  Update z via one Adam step on $\nabla_z \mathcal{L}_{\mathrm{inv}}$ with learning rate η
end for
Set $z_0 \leftarrow z$
// Self-adaptive rollout
Receive initial state s_1; initialise rollout history $\mathcal{H} \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  if $t \leq H$ **then**
    $z_t \leftarrow z_0$ {insufficient rollout history, use language-grounded latent}
  else
    $z_t \leftarrow E_\phi(\tau_{t-H:t-1})$ from $\mathcal{H}$ {re-encode from rollout}
  end if
  Predict action: $a_t \sim \pi_\theta(\cdot \mid \hat{R}_{\leq t}, s_{\leq t}, a_{<t};\ z_t)$
  Execute a_t, observe r_t, s_{t+1}; update RTG $\hat{R}_{t+1} \leftarrow \hat{R}_t - r_t$; append (s_t, a_t, r_t) to $\mathcal{H}$
end for

Algorithm 4 CoTA-DT — Few-shot adaptation

Require: Trained model $\{E_\phi, \theta_{\mathrm{L}}, \mathrm{Proj}_z, \mathrm{Proj}_h, f_{\mathrm{act}}, g_*\}$; target task description $h_\star$; support set $\mathcal{D}_\star$ of size k; target return $\hat{R}_0$; inversion steps K_{inv}, inversion lr η; fine-tune steps K_{ft}, fine-tune lr η_{ft}

// (1) Data-grounded latent initialisation
Sample sub-trajectories $\mathcal{C} = \{\tau_1, \ldots, \tau_{|\mathcal{C}|}\}$ from $\mathcal{D}_\star$
Compute aggregated latent: $\bar{z} = \frac{1}{|\mathcal{C}|} \sum_{\tau \in \mathcal{C}} E_\phi(\tau)$
Encode description: $v_\star = \mathrm{LLM}_{\theta_{\mathrm{base}},\theta_{\mathrm{L}}}(h_\star), \quad w_\star = \mathrm{Proj}_h(v_\star)/\|\mathrm{Proj}_h(v_\star)\|_2$
Initialise $z \leftarrow \bar{z}$ {data-grounded init in place of zero init}
for $j = 1, \ldots, K_{\mathrm{inv}}$ **do**
  $u \leftarrow \mathrm{Proj}_z(z)/\|\mathrm{Proj}_z(z)\|_2$
  Update z via one Adam step on $\nabla_z(1 - u^\top w_\star)$ with learning rate η
end for
Set $z_0 \leftarrow z$
// (2) Light-weight policy fine-tune
for $j = 1, \ldots, K_{\mathrm{ft}}$ **do**
  Sample DT window $(\hat{R}_{1:T_c}, s_{1:T_c}, a_{1:T_c})$ from $\mathcal{D}_\star$
  Forward policy with $z = z_0$ held fixed; compute $\mathcal{L}_{\mathrm{policy}}$ (Eq. 8)
  Update $\theta_{\mathrm{L}}, f_{\mathrm{act}}$ via Adam at lr η_{ft} (all other parameters frozen)
end for
// (3) Self-adaptive rollout
Run the self-adaptive rollout phase of Algorithm 3 using the fine-tuned policy and the data-grounded z_0