Decision Transformers As Zero-Shot Learners via Text-Behavior Alignment

Xin Zhang [1] **Jonathan Martinez** [1] **Yanhua Li** [2] **Yingxue Zhang** [3]

Abstract

Offline meta-reinforcement learning (meta-RL) aims to train agents that can generalize to unseen tasks using pre-collected data from related tasks. Recent approaches leverage the scalability of transformer architectures to model behavior sequences and support task adaptation using target task demonstrations. However, such data is often unavailable in real-world settings, where the task objective may be known but cannot be easily demonstrated. In contrast, humans routinely interpret and perform new tasks based solely on natural language instructions. In this work, we explore the potential of using natural language task descriptions to enable zero-shot task adaptation in offline meta-RL without access to target task data. We propose the Text-Guided Decision Transformer (TG-DT), a framework that enables zero-shot generalization by grounding policy learning in natural language. TG-DT learns a shared embedding space between task descriptions and behavioral trajectories via dual contrastive and matching-based objectives, ensuring robust alignment. A transformer-based policy is then conditioned on these aligned representations to generate task-appropriate actions. At test time, TG-DT synthesizes policies for unseen tasks using only their text descriptions and can optionally leverage a description-guided data sharing strategy to enhance adaptation. Experiments on standard offline meta-RL benchmarks, including MuJoCo and Meta-World, demonstrate that TG-DT achieves strong generalization to unseen tasks.

[1]Computer Science Department, San Diego State University, San Diego, CA, USA [2]Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA [3]Computer Science Department, Binghamton University, Binghamton, NY, USA. Correspondence to: Xin Zhang <xzhang19@sdsu.edu>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Offline reinforcement learning (RL) aims to learn an optimal policy from pre-collected datasets without requiring environment interactions (Levine et al., 2020b). This makes it particularly suitable for real-world applications such as healthcare (Gottesman et al., 2019), robotics (Cabi et al., 2020), and autonomous driving (Kidambi et al., 2020), where collecting online data can be costly or unsafe. A major milestone in this area is the Decision Transformer (DT) (Chen et al., 2021), which reformulates offline RL as a sequence modeling problem and leverages transformers to condition actions on desired returns, enabling the capture of long-horizon dependencies and flexible policy generation.

To extend DT beyond single-task scenarios and enhance its generalization to unseen tasks, DT-based offline meta-RL (Mitchell et al., 2021) has gained increasing attention, which learns across a distribution of tasks to acquire a transferable prior that facilitates adaptation to novel tasks. For instance, Prompt-DT (Xu et al., 2022) introduces expert demonstrations as prompts to encode task-specific information for adaptation. Generalized DT (GDT) (Furuta et al., 2021) utilizes hindsight reward distributions, while Meta-DT (Wang et al., 2024) incorporates a meta-policy to select informative trajectories for adaptation. Although these methods advance generalization to varying degrees, they remain constrained by their reliance on task-specific data (Xu et al., 2022; Wang et al., 2024; Zheng et al., 2024) or the need for access to test environments during adaptation (Xu et al., 2018; Rakelly et al., 2019; Zintgraf et al., 2021; Dong et al., 2024). Such requirements are often impractical in real-world settings, preventing these approaches from achieving true zero-shot generalization and limiting their broader applicability.

Motivation. These limitations naturally raise a key research question: *How can agents achieve zero-shot generalization to new tasks without test-time interaction or access to task-specific data?* Although behavioral data from new tasks is often unavailable, high-level task intent is typically known in advance and can be expressed through language. These descriptions convey the goal in broad, human-like terms (*e.g.*, "open the drawer halfway") without revealing hidden details such as exact coordinates or numeric rewards. For example, a household robot trained on tasks such as "tidying

up toys" or "loading the dishwasher" may later be instructed to perform an unseen task, such as "setting the table for dinner", which can be clearly described and semantically related to past experience. This observation motivates our approach: leveraging natural language as a medium for task specification in offline meta-RL. However, enabling this capability presents a unique challenge: aligning language with behavior requires reasoning over temporally extended state-action sequences, varying degrees of task completion, and inconsistent demonstration quality. Effectively modeling this alignment is essential for zero-shot generalization in the absence of task-specific data.

Our Approach. In this paper, we propose Text-Guided Decision Transformer (TG-DT), a novel offline meta-RL framework that enables zero-shot policy generalization to unseen tasks using only natural language task descriptions without requiring task-specific data or environment interaction at test time. TG-DT aligns behavioral trajectories with text task descriptions by learning a shared representation space through a dual alignment mechanism that combines contrastive learning with a matching-based objective. A DT is also trained to condition its action generation on these text-derived embeddings, effectively grounding task semantics in policy behavior.

At test time, TG-DT achieves zero-shot generalization by leveraging its text-conditioned policy to generate appropriate actions for previously unseen tasks based solely on their natural language descriptions. To further improve adaptation, TG-DT incorporates a description-guided data sharing strategy based on semantic similarity that selectively reuses offline trajectories from semantically related training tasks. This mechanism enhances generalization in the zero-shot setting and naturally extends to few-shot scenarios when limited task-specific data is available. *Our main contributions are as follows:*

- We propose TG-DT, a novel offline meta-RL framework that enables zero-shot generalization to unseen tasks using only text task descriptions. At its core, TG-DT uses a text-conditioned DT, which conditions action inference on text, allowing flexible and generalizable policy execution.
- To bridge the semantic gap between natural language and offline behavioral trajectories, we introduce a dual alignment mechanism that combines contrastive learning and matching-based objectives. This alignment enables robust grounding of textual task descriptions in offline trajectories.
- We enhance TG-DT's adaptability through description-guided data sharing based on semantic similarity, which facilitates effective adaptation by leveraging trajectories from similar tasks.
- Extensive experiments on benchmark offline meta-RL tasks show that TG-DT achieves compatible performance with state-of-the-art baselines in both zero-shot and few-shot generalization. Our code is available at https://github.com/STAI-SDSU/TG-DT.

2. Preliminaries

Offline Meta Reinforcement Learning. RL is usually formulated as a Markov Decision Process (MDP), defined by $M = \langle \mathcal{S}, \mathcal{A}, Tr, R, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, Tr is the state transition function, R is the reward function, and γ is the discount factor. The objective is to learn a policy $\pi(a|s)$ that maximizes the expected return $J_M(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

In offline meta-RL, we consider a task distribution $M_i = \langle \mathcal{S}, \mathcal{A}, Tr_i, R_i, \gamma \rangle \sim \mathcal{P}(M)$, where tasks share state and action spaces but differ in dynamics and rewards. For N training tasks $\{M_i\}_{i=1}^N$, the agent receives corresponding offline datasets $\{\mathcal{D}_i\}_{i=1}^N$, where $\mathcal{D}_i = \{(s_{i,j}, a_{i,j}, r_{i,j}, s_{i,j+1})_{j=1}^J\}$ is collected under behavior policy π_β^i. The goal is to learn a meta-policy π_{meta} that generalizes to new tasks. At test time, the agent faces a new task $M_k \sim \mathcal{P}(M)$. In the *few-shot* setting, a small offline data is provided to support adaptation. In the *zero-shot* setting, no additional task-specific data is allowed. We study an even *stricter variant of zero-shot adaptation*, where the agent is not allowed any online interaction with the new task and rely solely on its natural language description. The meta-policy π_{meta} aims to maximize expected performance over test tasks: $J(\pi_{\text{meta}}) = \mathbb{E}_{M\sim P(M)} \left[J_M(\pi_{\text{meta}})\right]$.

Decision Transformer (DT). DT (Chen et al., 2021) formulates offline reinforcement learning as a return-conditioned sequence modeling problem, inspired by the success of transformers in language modeling (Vaswani et al., 2017). Instead of explicitly modeling MDP dynamics or estimating value functions, DT learns directly from offline behavior data by autoregressively predicting actions. In DT, each trajectory is represented as a sequence of tokens $(\hat{R}_t, s_t, a_t)$, where s_t is the state, a_t is the action, and $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$ denotes the return-to-go (RTG) from timestep t. The RTG serves as a conditioning signal, guiding the model to replicate high-return behaviors observed in the dataset. During training, the model is optimized to predict the next action given the sequence of past RTGs, states, and actions. At test time, a target return $G^\star$ is provided, and the RTG is estimated dynamically as $\hat{R}_t = G^\star - \sum_{t'=0}^{t} r_{t'}$. At each timestep, the embeddings of $\hat{R}_t$, s_t, and a_t are concatenated and input to the transformer. The model is trained to minimize the prediction error between its output and the ground-truth action. This formulation maintains the temporal structure of decision-making while leveraging the scalability and generalizability of transformer-based architectures.

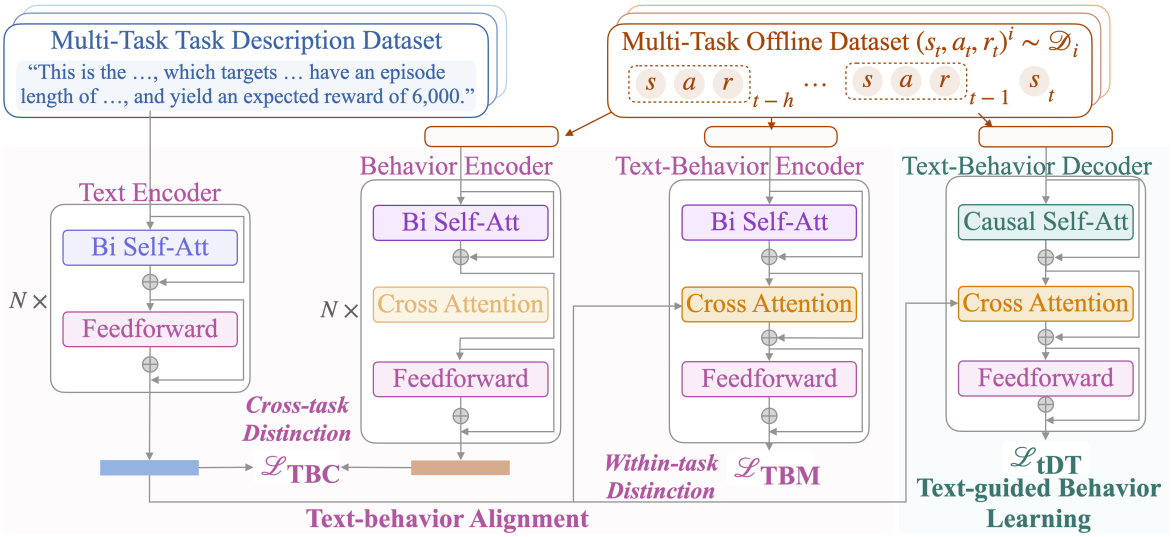


Figure 1. TG-DT model architecture and objectives (same parameters share color). TG-DT employs a text-behavior encoder-decoder framework to enable text-guided behavior learning and alignment: (1) Text and behavior encoders are jointly trained with $\mathcal{L}_{\text{TBC}}$ in Eq. (1) to align representations for cross-task distinction; (2) The text-behavior encoder incorporates cross-attention layers and is further trained with $\mathcal{L}_{\text{TBM}}$ in Eq. (2) to capture within-task differences; (3) Text-behavior decoder replaces bi-directional self-attention with causal self-attention, while reusing the encoder's cross-attention and feedforward layers, and is trained with $\mathcal{L}_{\text{tDT}}$ in Eq. (3) to generate behaviors from task descriptions.

3. Text-Guided Decision Transformer

We present the Text-Guided Decision Transformer (TG-DT), a novel offline meta-RL framework that enables zero-shot adaptation using only natural language task descriptions, as illustrated in Fig. 1.

3.1. Task Descriptions in Offline Meta-RL

Humans often adapt to novel tasks by leveraging high-level natural language instructions, without direct experience or environmental interaction. These instructions convey task intent based on prior familiarity with the environment and related experiences, enabling people to generalize effectively even in unfamiliar scenarios. Inspired by this capability, we use task descriptions as the sole source of information for zero-shot generalization without online interaction.

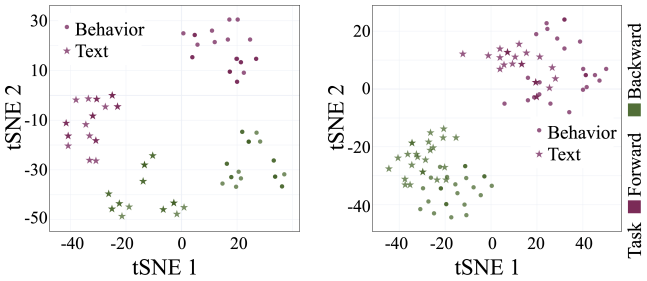
To ensure consistency and interpretability, we adopt a templated strategy for constructing task descriptions. For each trajectory $\tau_j \in \mathcal{D}_i$, we create a natural language description in the format:

```
"This is the [task_name], which
targets [task_intent].  Its
corresponding environment is
[environment_description].  Good
demonstrations typically have an
episode length of [episode_length],
and yield an expected return of
[expected_return].  This demonstration
yields a [return]."
```

Each placeholder is deterministically filled using metadata extracted from τ_j, including the task name (*e.g.*, pen-drawer task), high-level intent (without explicit goals, *e.g.*, open a drawer to a specific location), environment information, episode length, and expected return. During training, these values come directly from the offline dataset metadata. This design grounds descriptions in observable behavior while conveying both task intent and environmental context. It serves as a training signal that grounds environmental information into the text description and also mirrors how humans typically describe tasks, supporting alignment with language-based reasoning.

To reflect the diverse quality of demonstrations, we employ trajectory-level pairing, where each trajectory τ_j is coupled with its own natural language description $\boldsymbol{p}_j$, rather than using a single description for all trajectories within a task. This pairing captures fine-grained variation in demonstration quality, such as episode length and cumulative reward, by encoding trajectory-specific properties directly into the text. This allows the model to leverage both behavioral and linguistic signals that reflect demonstration quality for a deeper understanding of trajectory semantics.

Formally, our training data consists of a collection of text-behavior pairs $\{(\boldsymbol{p}_j, \tau_j)\}_{j=1}^{|\mathcal{D}_i|}$ where each trajectory τ_j comes from an offline dataset $\mathcal{D}_i$ for training tasks $M_i \sim \mathcal{P}(M)$. We aim to learn a meta-policy π_{meta} that generalizes to unseen tasks using only their natural language descriptions at test time, thus supporting zero-shot adaptation in the offline setting.



(a) tSNE of TBC-aligned representations. *(b)* tSNE of TBC & TBM-aligned representations.

Figure 2. TBM pulls matched text-behavior pairs closer in Cheetah-dir across forward/backward tasks.

3.2. Text-behavior Alignment

A key challenge in text-guided offline meta-RL is enabling it to understand natural language task descriptions. Inspired by recent advances in vision-language alignment (Li et al., 2022; Radford et al., 2021; Eslami & de Melo, 2025), we extend similar strategies to align text and behavior, which introduce unique challenges. Unlike static images, trajectories are temporal sequences that capture dynamic interactions and partial environment information. Text in this context encodes abstract intents, and must be grounded in sequential behavior that unfolds over time. Moreover, offline behavior datasets contain demonstrations of varying quality, requiring the model to develop sensitivity to performance rather than perceptual similarity. These challenges demand a richer alignment mechanism capable of reasoning over temporal dynamics, quality signals, and task semantics.

We introduce a Text-Behavior Alignment (TBA) module that maps task descriptions and behavioral trajectories into a shared latent space. As shown in Fig. 1, TBA consists of three components: a *text encoder*, a *behavior encoder*, and a *text-behavior encoder*. The text and behavior encoders embed descriptions $\boldsymbol{p}_j$ and trajectories τ_j into $\boldsymbol{e}_j^p$ and $\boldsymbol{e}_j^\tau$, respectively. The text-behavior encoder models their joint interaction. Alignment is guided by complementary objectives: *contrastive learning* captures coarse semantic alignment across tasks, while *matching supervision* enables fine-grained distinction of behavior quality within tasks.

Text-behavior Contrastive Learning (TBC). To enable cross-task distinction, we introduce a contrastive objective that aligns text and behavior embeddings from the same task while separating those from different tasks. Given a batch of B text-behavior pairs $\{(\tau_j, \boldsymbol{p}_j)\}_{j=1}^B$, and their encoded representations $\{(\boldsymbol{e}_j^\tau, \boldsymbol{e}_j^p)\}_{j=1}^B$, similarity is measured via scaled cosine similarity

$$\mathrm{sim}(\boldsymbol{e}_j^\tau, \boldsymbol{e}_j^p) = \frac{(\boldsymbol{e}_j^\tau)^\top \boldsymbol{e}_j^p}{\|\boldsymbol{e}_j^\tau\| \, \|\boldsymbol{e}_j^p\|}.$$

The *text-behavior contrastive loss* is defined as:

$$\mathcal{L}_{\text{TBC}} \triangleq \frac{1}{B}\sum_{j=1}^{B}\Big[-\log \frac{\exp\big(\mathrm{sim}(\boldsymbol{e}_j^\tau, \boldsymbol{e}_j^p)/\tau\big)}{\sum_{k=1}^{B}\exp\big(\mathrm{sim}(\boldsymbol{e}_j^\tau, \boldsymbol{e}_k^p)/\tau\big)} - \log \frac{\exp\big(\mathrm{sim}(\boldsymbol{e}_j^p, \boldsymbol{e}_j^\tau)/\tau\big)}{\sum_{k=1}^{B}\exp\big(\mathrm{sim}(\boldsymbol{e}_j^p, \boldsymbol{e}_k^\tau)/\tau\big)}, \tag{1}$$

where τ is a temperature hyperparameter controlling the distribution sharpness. The first term aligns each behavior embedding with its corresponding text embedding, while the second does the reverse. Each positive pair is a trajectory and its own description. Negatives are sampled predominantly from different tasks to encourage task-level separation. This design promotes both text-behavior alignment and cross-task distinction, helping the model disambiguate tasks based on semantic intent. By pulling together matched text-behavior pairs and pushing apart those from unrelated tasks, the model acquires a more structured representation of task space which is critical for generalization.

To further enhance consistency, the text and behavior encoders share feedforward layers, promoting alignment in a unified task-centric space (Eslami & de Melo, 2025). We also adopt momentum encoders with soft targets (Li et al., 2021) to improve robustness and reduce false negatives.

Text-behavior Matching (TBM). While TBC promotes cross-task distinction, it relies on coarse semantic similarity and struggles to capture fine-grained differences or link language to environment-specific dynamics and data quality variations within the same task. To address this, we introduce a *text-behavior matching* loss, formulated as a *binary classification task* to distinguish matched from mismatched text-behavior pairs. A *text-behavior encoder* processes each text-behavior pair $(\tau_j, \boldsymbol{p}_j)$ and produces a joint embedding, which a linear classification head uses to predict a matching score z_j. The loss is then defined as a binary cross-entropy loss:

$$\mathcal{L}_{\text{TBM}} \triangleq -\frac{1}{B}\sum_{j=1}^{B}\left[y_j \log\sigma(z_j) + (1-y_j)\log(1-\sigma(z_j))\right], \tag{2}$$

where $y_j \in \{0, 1\}$ indicates whether the text-behavior pair is a true match, and $\sigma(\cdot)$ is the sigmoid function. The text-behavior encoder features cross attention layers to directly model interactions between text and behaviors, embedding environment-specific signals such as rewards and transitions. To improve the differentiation on data quality, we adopt the hard negative mining strategy (Li et al., 2021): negatives include both inter-task mismatches and intra-task mismatches (*e.g.*, a text paired with a trajectory of different quality). This ensures the model to discriminate subtle differences in return or success, not just coarse task identity. As shown in Fig. 2b, TBM enhances within-task alignment by bringing matched pairs closer in the shared space.

3.3. Text-guided Behavior Learning

We enable policy learning from task descriptions via text-guided behavior learning. TG-DT extends Decision Transformer (DT) to a *text-behavior decoder* that autoregressively generates actions based on past trajectories and the task descriptions. It optimizes with the *text-conditioned DT* loss:

$$\mathcal{L}_{\text{tDT}} \triangleq \mathbb{E}_{M_i \sim P(M), \tau_j \sim \mathcal{D}_i} \left[\sum_{t=1}^{T} \left\| \boldsymbol{a}_{j,t} - \pi(\tau_{j,t-1}, \boldsymbol{p}_j) \right\|^2 \right]. \tag{3}$$

Here, $\pi(\cdot)$ is the policy conditioned on prior trajectory $\tau_{j,t-1}$ and task description $\boldsymbol{p}_j$. The mean squared error (MSE) loss trains the policy to reproduce expert actions consistent with the task description, extending DT beyond return-to-go conditioning to language-aligned embeddings for zero-shot task specification.

During meta-training, we jointly optimize three objectives, *i.e.*, $\mathcal{L} = \mathcal{L}_{\text{TBC}} + \mathcal{L}_{\text{TBM}} + \mathcal{L}_{\text{tDT}}$, enabling the model to separate tasks, capture trajectory-level quality differences, and generate actions consistent with task intent. This combined loss couples alignment with policy learning, enabling TG-DT to adapt to behavior data while remaining grounded in text and thus bridging the gap between language and behavior in offline meta-RL.

To strengthen language understanding, we initialize TG-DT components using the pre-trained BLIP model (Li et al., 2022) due to its capability to capture the rich text semantics. Although BLIP was originally designed for image-text tasks, its encoders learn cross-modal attention patterns that transfer effectively to trajectory-text alignment.

4. Description-Guided Task Adaptation

Zero-Shot Adaptation from Text Descriptions. Given a new task M_k defined solely by its natural language description $\boldsymbol{p}_k$, TG-DT enables zero-shot adaptation by conditioning the policy on the aligned text embedding. At test time, $\boldsymbol{p}_k$ follows the same templated format as training prompts, but with values such as expected return and episode length replaced by approximate statistics inferred from the training distribution rather than using ground-truth values. (See Appx. D for test prompts.) During inference, $\boldsymbol{p}_k$ is passed to the text encoder to get $\boldsymbol{e}_k^p$, which guides the text-behavior decoder to generate actions autoregressively. Since the model is meta-trained to align text and behavior in a shared latent space, $\boldsymbol{e}_k^p$ effectively captures the task semantics and supports generalization to unseen tasks. *This process requires no environment interaction or task-specific data at test time.* Instead, generalization emerges from the model's ability to associate semantic features in natural language with corresponding behavioral patterns observed during training. This makes TG-DT well-suited for applications where test-time data collection is infeasible, costly, or unsafe.

Description-Guided Data Sharing for Adaptation Enhancement. To further enhance zero-shot performance, we incorporate a data-sharing strategy based on semantic similarity. Given a test task description $\boldsymbol{p}_k$ with embedding $\boldsymbol{e}_k^p$, we retrieve the top-K training task descriptions whose text embeddings are most similar to $\boldsymbol{e}_k^p$ via $\arg\max_{\boldsymbol{p}_j \in \cup_{i=1}^{N} \{\boldsymbol{p}_j\}_{j=1}^{|\mathcal{D}_i|}} \text{sim}(\boldsymbol{e}_k^p, \boldsymbol{e}_j^p)$. We then use the corresponding trajectories to fine-tune the text-behavior decoder with $\mathcal{L}_{tDT}$ in Eq. (3), while still conditioning on $\boldsymbol{p}_k$. This allows the model to refine its behavior using related training data without requiring any supervision from the target task.

Few-Shot Adaptation from Text Descriptions. When a small amount of task-specific offline data $\mathcal{D}_k$ is available, the text-behavior decoder can be fine-tuned in a few-shot setting. It leverages both the task description $\boldsymbol{p}_k$ and limited demonstrations to improve adaptation. We use the same $\mathcal{L}_{tDT}$ objective in Eq. (3), treating $\mathcal{D}_k$ as a lightweight fine-tuning set. This hybrid approach retains the generalization benefits of language guidance while allowing for task-specific refinement.

5. Experiment

We evaluate TG-DT's generalization capability on standard benchmarks to answer the following key questions: *1).* Can TG-DT outperform strong baselines in zero- and few-shot generalization to unseen tasks? (See Sec. 5.2) *2).* Does the text-behavior alignment mechanism produce a meaningful shared embedding space that supports task understanding and transfer? (See Sec. 5.3) *3).* What is the impact of the contrastive (TBC) and matching (TBM) components on generalization performance, and how effective is the description-guided data sharing strategy in enhancing adaptation to new tasks? (See Sec. 5.4) *4).* How robust is TG-DT to variations in offline data quality? (See Sec. 5.5)

5.1. Experiment Setup

Environments. We evaluate TG-DT on two widely adopted benchmarks, MuJoCo (Todorov et al., 2012) and Meta-World (Yu et al., 2020). For MuJoCo, we follow Xu et al. (2022) and evaluate on Cheetah-dir, Cheetah-vel, and Ant-dir. For MetaWorld, we use the ML10, and ML45 task suites for robotic manipulation. Datasets are collected using SAC (Haarnoja et al., 2018) trained independently on sampled tasks, with three dataset types: Medium, Mixed, and Expert.

Baselines. We compare TG-DT with two categories of baselines. <u>*DT-based baselines*</u> include: *1).* Prompt-DT (PDT) (Xu et al., 2022): Builds on DT by introducing trajectory-level prompts and reward-to-go conditioning to

Method	Cheetah-dir	Cheetah-vel	Ant-dir	ML10	ML45
PDT†	548.9	-150.6	214.2	289.2	248.3
GDT	129.2	-218.4	167.9	169.8	153.2
MDT†	539.6	-102.7	**357.5**	335.2	306.4
HDT†	445.3	-162.7	215.4	266.4	245.7
DPDT†	548.1	-142.6	321.8	360.4	**311.8**
BC-Z†	310.5	-123.9	254.9	199.7	157.4
BAKU	348.6	-110.5	277.2	216.3	163.2
TG-DT	**549.9**	**-93.0**	328.3	**361.1**	309.6

Table 1. Zero-shot test returns of TG-DT *vs*. baselines using Medium datasets. We report average returns over 5 runs (higher is better), with standard deviations in Appx. B. † denotes methods that require test-time interaction to obtain adaptation demonstrations. TG-DT requires no interaction.

Method	Cheetah-dir	Cheetah-vel	Ant-dir	ML10	ML45
PDT†	587.3	-136.8	310.8	322.2	298.8
GDT	569.7	-118.9	291.3	211.6	209.3
MDT†	**599.4**	-95.3	409.9	361.3	442.5
HDT†	489.5	-122.5	376.7	319.3	315.4
DPDT†	591.4	-78.3	410.4	**425.6**	498.6
BC-Z†	356.2	-126.5	297.4	217.4	163.9
BAKU	323.8	-142.8	356.2	309.2	172.8
TG-DT	598.4	**-73.2**	**412.9**	421.5	**499.2**

Table 2. Few-shot test returns of TG-DT *vs*. baselines using Medium datasets. We report average returns over 5 runs (higher is better), with standard deviations in Appx. B. † denotes methods that require test-time interaction to obtain adaptation demonstrations. TG-DT requires no interaction.

enable generalization to unseen tasks; *2).* Generalized DT (GDT) (Furuta et al., 2021): Extends DT by leveraging hindsight reward distributions to guide adaptation and improve generalization; *3).* Meta-DT (MDT) (Wang et al., 2024): Enhances DT by introducing a meta-policy to enable stronger generalization;*4).* Hyper DT (HDT) (Xu et al., 2023): Adapts DT to new tasks via a lightweight adaptation module, allowing efficient fine-tuning with minimal data; and *5).* DPDT (Zheng et al., 2024): Augments DT with decomposed prompt tuning, enabling parameter-efficient test-time adaptation. *Language-conditioned RL baselines* include *6).* BC-Z (Jang et al., 2022) achieves zero-shot generalization from language via imitation learning, and *7).* BAKU (Haldar et al., 2024) introduces an efficient transformer for multi-task policy learning. We delegate more environment descriptions, hyperparameters and implementation details in Appx. B.

Implementation details. All results are averaged over 5 random seeds, with the standard deviation shown in Appx B. Experiments were conducted on 4 NVIDIA RTX 6000 Ada GPUs (48GB), using PyTorch and the Hugging Face Transformers library (Wolf et al., 2019).

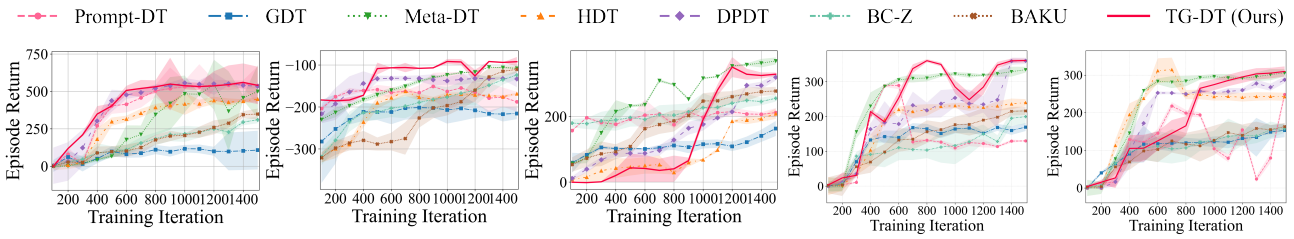
5.2. Zero-shot and Few-shot Generalization

Zero-shot Generalization. Results in Tab. 1 show that TG-DT performs on par with or better than DT-based baselines. Unlike DT-based baselines (*e.g*., Meta-DT, and HDT), which assume access to test task interaction, TG-DT generalizes strictly from text intent without any environment interaction. Language-conditioned baselines such as BC-Z and BAKU underperform in this setting, as they rely on imitation without return conditioning. TG-DT's dual alignment mechanism leverages task intent in text while grounding it in trajectory quality, enabling robust generalization across unseen tasks. Fig. 3 further illustrates TG-DT's stable convergence and stable performance during training. The slower convergence in ML10 and ML45 tasks shows a trade-off resulting from TG-DT's deliberate emphasis on learning robust, task-conditioned representations rather than rapidly adapting to individual tasks. Overall, the results demonstrate TG-DT's ability to learn robust, generalizable policies from task descriptions without sacrificing performance in zero-data settings.

Few-shot Generalization. As shown in Tab. 2, TG-DT demonstrates strong performance under the few-shot setting, consistently outperforming or matching all baselines across most environments. This improvement can be attributed to TG-DT's ability to effectively leverage real demonstrations from previously unseen tasks at test time. By conditioning on these demonstrations, TG-DT refines its task understanding and adapts its behavior more precisely to the new task.

5.3. Text-behavior Alignment Performance

To evaluate alignment between task descriptions and behavior trajectories, we visualize their embeddings using tSNE and assess their similarity via cosine similarity. Fig. 4 shows results for Cheetah-vel and Ant-dir. In both environments, the text embeddings ($\star$) and corresponding behavior embeddings ($\bullet$) form coherent clusters in the projected space, suggesting that the model learns a semantically meaningful alignment between language and behavior. For Cheetah-vel (Fig. 4a), the embeddings are organized along a smooth manifold with a clear progression across different task IDs. This reflects the continuous nature of the velocity control tasks, which are sampled from a uniform distribution over a range of target velocities. In Ant-dir (Fig. 4b), a similarly structured embedding space emerges, where tasks with similar directional goals are grouped together, despite the more complex and higher-dimensional action space associated with limb coordination in the Ant environment.



(a) Cheetah-dir. *(b)* Cheetah-vel. *(c)* Ant-dir. *(d)* ML-10. *(e)* ML-45.

Figure 3. The received return curves averaged over test tasks of TG-DT and baselines using Medium datasets in the zero-shot setting.

Method	Cheetah-dir	Cheetah-vel	Ant-dir
w/o TBC, TBM	859.4	−55.7	133.6
w/o TBC	936.8	−24.6	322.1
w/o TBM	875.2	−46.6	298.7
TG-DT	**958.4**	**−21.4**	**383.4**

Table 3. The impact of TBC and TBM on TG-DT.

The adjacent histograms display the distribution of cosine similarity scores between each text-behavior pair. For Cheetah-vel, the similarity values are more concentrated around higher values (mean ≈ 0.34), indicating a tighter alignment. In contrast, Ant-dir shows a broader spread and slightly lower mean similarity (≈ 0.28), likely due to the greater task diversity and motor complexity. Despite this, the overall average cosine similarity remains close to 0.3 in both cases. When coupled with the performance metrics in Tab. 1, these findings suggest that the representation gap between text and behavior does not hinder policy learning. This observation aligns with recent work on multi-modal representation learning (Jiang et al., 2023), which shows that a moderate discrepancy in embedding spaces can be tolerated as long as the overall structure is preserved and contrastive alignment is effective.

5.4. Ablation Study

Impact of TBM and TBC. Tab. 3 shows the performance of TG-DT when selectively disabling Text-Behavior Contrastive Learning (TBC) and Text-Behavior Matching (TBM) during training.

Removing both TBC and TBM significantly degrades performance across all environments. When TBC is ablated, we observe unstable task adaptation and lower returns, suggesting that the model struggles to enforce behavioral coherence between different task instances. Besides, removing TBM leads to a less precise alignment between the task prompt and behavior trajectories, resulting in reduced task relevance. In contrast, TG-DT with both parts active consistently achieves the highest performance, indicating that these two components play complementary roles in encouraging effective model generalization and text-behavior alignment.

Effect of the Shared Demonstration Count K. Fig. 5 illustrates the effect of varying the number of shared demonstrations used during description-guided data sharing. No data-sharing is implemented when $K = 0$. We observe that incorporating a small number of real trajectories from related tasks (*i.e.*, $K = 1$ or $K = 2$) boosts performance in settings such as Ant-dir and ML45. This indicates even minimal shared data can provide valuable contextual grounding for TG-DT's task representations. However, performance gains saturate or slightly decline as K increases beyond a certain point, suggesting diminishing returns and potential overfitting to related tasks instead of the test task.

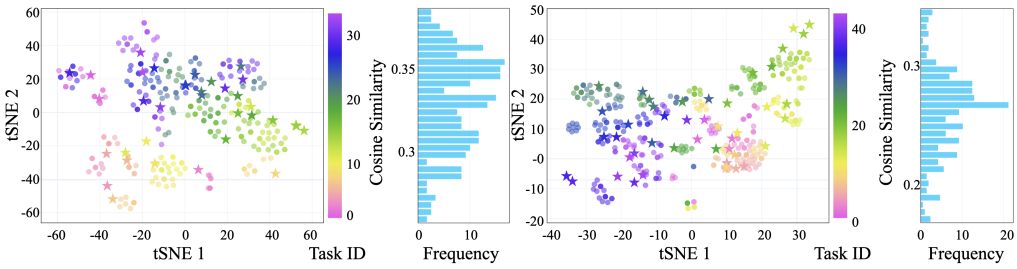
5.5. Robustness to the Quality of Offline Datasets

To evaluate TG-DT's robustness to varying offline training data qualities, we conduct experiments using Expert, Mixed, and Medium datasets and compare zero-shot performance against baselines. As shown in Tab. 1 and Tab. 4, TG-DT consistently delivers strong performance across all dataset types.

On Expert datasets, TG-DT achieves top-tier performance, matching or surpassing existing baselines across most tasks. This is expected, as expert trajectories offer consistent, high-reward behavior that aligns well with TG-DT's task-conditioned modeling approach. In contrast, Mixed datasets present a more challenging setting due to the inclusion of noisy or suboptimal data. While all methods experience performance drop, TG-DT remains competitive and even outperforms alternatives like DPDT and Meta-DT in complex environments such as Ant-dir. This resilience highlights TG-DT's ability to filter out task-irrelevant patterns through its semantic task-behavior alignment.

6. Related Work

Policy Learning as Sequence Modeling. Decision Transformer (DT) (Chen et al., 2021) casts RL as sequence modeling, predicting actions from history and target returns, and has been extended to pretraining, ranking, and multi-task



(a) Cheetah-vel. *(b)* Ant-dir.

Figure 4. t-SNE and cosine similarity histograms of text–behavior representations. Left: t-SNE with text centroids (⋆) and behavior embeddings (●); Right: cosine similarity distributions.

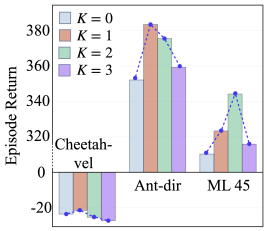


Figure 5. Impact of data sharing fine-tuning with varying numbers of shared demonstrations.

Env	Expert								Mixed							
	PDT	GDT	MDT	HDT	DPDT	BC-Z	BAKU	TG-DT	PDT	GDT	MDT	HDT	DPDT	BC-Z	BAKU	TG-DT
Cheetah-dir	929.7	779.3	947.2	875.2	955.1	759.2	761.3	**958.4**	836.6	699.1	853.6	786.0	**861.2**	687.4	693.6	830.1
Cheetah-vel	−39.3	−59.2	−29.7	−45.3	−30.1	−61.4	−55.3	**−21.4**	−44.3	−64.2	−34.7	−50.3	−35.1	−72.3	−63.3	**−34.4**
Ant-dir	347.1	352.2	376.4	361.49	**384.2**	309.5	337.6	383.4	310.4	307.6	327.0	314.6	342.6	291.4	301.3	**344.9**

Table 4. Zero-shot test returns of TG-DT against baselines using Mixed and Expert datasets. We report the average returns and standard deviations for five runs (the higher, the better).

settings (Schmied et al., 2023; Lee et al., 2022; Reed et al., 2022). Prompt-based DT variants (*e.g.*, Prompt-DT, DPDT) adapt via learned/handcrafted prompts but primarily rely on trajectory-derived signals and offer limited semantic interpretability (Xu et al., 2022; Zheng et al., 2024). GDT and Meta-DT further explore adaptation via hindsight conditioning or trajectory selection (Furuta et al., 2021; Wang et al., 2024). Our approach instead leverages natural language task descriptions and explicitly aligns text with behavior for zero-shot generalization (Brown et al., 2020; Brandfonbrener et al., 2022).

Offline Meta-RL. Offline RL learns from fixed datasets (Levine et al., 2020b; Konyushkova et al., 2020); offline meta-RL targets generalization across tasks using only offline data (Mitchell et al., 2020; Pong et al., 2021; Dorfman et al., 2021; Li et al., 2020). Many methods are TD-based, using optimization-style meta-learning or latent context inference (*e.g.*, PEARL, VariBAD) (Finn et al., 2017; Mitchell et al., 2020; Xu et al., 2018; Rakelly et al., 2019; Zintgraf et al., 2021; Yuan & Lu, 2022; Li et al., 2024). However, TD learning in offline settings can be unstable due to the deadly triad (Levine et al., 2020a) and often requires constraints to stay within dataset support (Ajay et al., 2022), motivating more stable, return-conditioned alternatives.

Language-conditioned RL and Alignment. BC-Z and BAKU study language-conditioned imitation learning but are not designed for offline meta-RL and omit return conditioning (Jang et al., 2022; Haldar et al., 2024). TTCT requires online interaction for adaptation (Dong et al., 2024). Vision-language grounding methods (*e.g.*, MineCLIP, STEVE-1) show strong text–behavior grounding but typically operate in online or imitation settings rather than return-conditioned decision modeling (Fan et al., 2022; Lifshitz et al., 2023). Concurrently, T2DA prepends text to trajectories for prompting, but performs implicit alignment and does not explicitly model intra-task quality variation under noisy offline data (Zhang et al., 2025). TG-DT introduces explicit dual alignment: contrastive learning separates tasks, while matching supervision captures within-task quality differences.

Compared to prior offline meta-RL and language-conditioned approaches that rely on TD adaptation, online interaction, or imitation-only objectives, TG-DT enables zero-shot offline generalization from language alone via explicit text–behavior alignment while preserving DT-style return conditioning.

7. Conclusions & Limitations

We proposed TG-DT, a Text-Guided Decision Transformer that leverages text task descriptions and text-behavior alignment objectives to improve generalization in offline meta-RL. Extensive experiments across diverse tasks and dataset qualities show that TG-DT consistently outperforms or matches strong baselines in both zero-shot and few-shot settings. Our ablation studies confirm the effectiveness of the proposed TBC and TBM modules, as well as the benefit of limited shared data. TG-DT remains robust under suboptimal data conditions, making it a practical and data-efficient

solution for real-world offline meta-RL.

TG-DT relies on templated task descriptions that include metadata during training, and at test time it replaces these fields with approximate values inferred from the training tasks. This design prevents oracle information leakage but reduces robustness to free-form natural language, making support for unconstrained instructions an important direction for future work.

Acknowledgment

We thank the reviewers for the valuable suggestions. Xin Zhang is supported in part by NSF grant IIS-2449864. Yanhua Li was supported in part by NSF grants IIS-1942680 (CAREER).

Impact Statement

This paper presents a text-guided offline meta-RL framework (TG-DT) that aligns natural-language task descriptions with behavior to enable zero-shot adaptation without test-time interaction, potentially improving safety and usability in domains where exploration is risky or expensive (*e.g.*, robotics or other cyber-physical systems) while reducing the cost of collecting new demonstrations. However, conditioning actions on language introduces risks: ambiguous or adversarial task descriptions can lead to unsafe or unintended behavior; offline datasets may encode bias that the learned text–behavior alignment propagates; generalization may fail under distribution shift; and training on real trajectories can raise privacy and governance concerns. Responsible use should therefore include careful dataset documentation/auditing, robustness testing to language variation and OOD conditions, and deployment-time safeguards such as action constraints, monitoring, and human oversight, especially in high-stakes applications.

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Brandfonbrener, D., Bietti, A., Buckman, J., Laroche, R., and Bruna, J. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems*, 35: 1542–1553, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cabi, S., Colmenarejo, S. G., Hoffman, M., Denil, M., Wang, Z., and de Freitas, N. Scaling data-driven robotics with reward sketching and batch reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2020.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Dong, P., Zhu, T., Zhou, H., Li, J., et al. From text to trajectory: Exploring complex constraint representation and decomposition in safe reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 125635–125662, 2024.

Dorfman, R., Shenfeld, I., and Tamar, A. Offline meta reinforcement learning – identifiability challenges and effective data collection strategies. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (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.

Eslami, S. and de Melo, G. Mitigate the gap: Improving cross-modal alignment in CLIP. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aPTGvFqile.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

Finn, C., Abbeel, P., and Levine, S. 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.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. *CoRR*, abs/1812.02900, 2018. URL http://arxiv.org/abs/1812.02900.

Furuta, H., Matsuo, Y., and Gu, S. S. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021.

Gottesman, O., Johansson, F., Meier, J., Dent, J., Lee, D., Srinivasan, S., Zhang, H., Ding, Y., Wihl, D., Komorowski, M., et al. Guidelines for reinforcement learning in healthcare. In *Nature Medicine*, volume 25, pp. 16–18. Nature Publishing Group, 2019.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. 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.

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

Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Jiang, Q., Chen, C., Zhao, H., Chen, L., Ping, Q., Tran, S. D., Xu, Y., Zeng, B., and Chilimbi, T. Understanding and constructing latent modality structures in multi-modal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7661–7671, 2023.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 21810–21823, 2020.

Konyushkova, K., Zolna, K., Aytar, Y., Novikov, A., Reed, S. E., Cabi, S., and de Freitas, N. Semi-supervised reward learning for offline reinforcement learning. *CoRR*, abs/2012.06899, 2020. URL https://arxiv.org/abs/2012.06899.

Lee, K.-H., Nachum, O., Yang, M. S., Lee, L., Freeman, D., Guadarrama, S., Fischer, I., Xu, W., Jang, E., Michalewski, H., et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35: 27921–27936, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020a. URL https://arxiv.org/abs/2005.01643.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020b.

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

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Li, L., Zhang, H., Zhang, X., Zhu, S., Yu, Y., Zhao, J., and Heng, P.-A. Towards an information theoretic framework of context-based offline meta-reinforcement learning. *arXiv preprint arXiv:2402.02429*, 2024.

Lifshitz, S., Paster, K., Chan, H., Ba, J., and McIlraith, S. Steve-1: A generative model for text-to-behavior in minecraft. *Advances in Neural Information Processing Systems*, 36:69900–69929, 2023.

Mitchell, E., Rafailov, R., Peng, X. B., Levine, S., and Finn, C. Offline meta-reinforcement learning with advantage weighting. *CoRR*, abs/2008.06043, 2020. URL https://arxiv.org/abs/2008.06043.

Mitchell, E., Rafailov, R., Peng, X. B., Levine, S., and Finn, C. Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning*, pp. 7780–7791. PMLR, 2021.

Pong, V. H., Nair, A., Smith, L. M., Huang, C., and Levine, S. Offline meta-reinforcement learning with online self-supervision. *CoRR*, abs/2107.03974, 2021. URL https://arxiv.org/abs/2107.03974.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pp. 5331–5340. PMLR, 2019.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Schmied, T., Hofmarcher, M., Paischer, F., Pascanu, R., and Hochreiter, S. Learning to modulate pre-trained models in rl. *Advances in Neural Information Processing Systems*, 36:38231–38265, 2023.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Zhang, L., Wu, W., Zhu, Y., Zhao, D., and Chen, C. Meta-dt: Offline meta-rl as conditional sequence modeling with world model disentanglement. *Advances in Neural Information Processing Systems*, 37:44845–44870, 2024.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J., and Gan, C. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pp. 24631–24645. PMLR, 2022.

Xu, M., Lu, Y., Shen, Y., Zhang, S., Zhao, D., and Gan, C. Hyper-decision transformer for efficient online policy adaptation. *arXiv preprint arXiv:2304.08487*, 2023.

Xu, Z., van Hasselt, H. P., and Silver, D. Meta-gradient reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

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

Zhang, S., Hu, Z., Wu, W., Xie, X., Tang, J., Chen, C., Dong, D., Cheng, Y., Sun, Z., and Wang, Z. Text-to-decision agent: Offline meta-reinforcement learning from natural language supervision. *arXiv preprint arXiv:2504.15046*, 2025.

Zheng, H., Shen, L., Luo, Y., Liu, T., Shen, J., and Tao, D. Decomposed prompt decision transformer for efficient unseen task generalization. *Advances in Neural Information Processing Systems*, 37:122984–123006, 2024.

Zintgraf, L., Schulze, S., Lu, C., Feng, L., Igl, M., Shiarlis, K., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: Variational bayes-adaptive deep rl via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.

A. Extended Related Work

Policy Learning as Sequence Modeling. Decision Transformer (DT) (Chen et al., 2021) reframes RL as sequence modeling, using transformers to predict actions from past states and target returns. This framework has inspired extensions in unsupervised pretraining (Schmied et al., 2023), trajectory ranking (Lee et al., 2022), and multi-task learning (Reed et al., 2022). Prompt-based variants, such as Prompt-DT (Xu et al., 2022) and DPDT (Zheng et al., 2024), adapt to new tasks using soft or handcrafted prompts but lack semantic interpretability and rely heavily on trajectory-derived inputs. GDT (Furuta et al., 2021) uses hindsight rewards for adaptation, while Meta-DT (Wang et al., 2024) employs a meta-policy to select informative trajectories. In contrast, our approach leverages natural language task descriptions to align text and behavior, enabling zero-shot generalization in offline meta-RL. While inspired by recent efforts linking language and control (Brown et al., 2020; Brandfonbrener et al., 2022), we explicitly ground task semantics in behavioral data via task-guided sequence modeling.

Offline Meta-RL. Offline RL aims to learn policies from fixed datasets without further interaction with the environment (Levine et al., 2020b; Konyushkova et al., 2020). Offline meta-RL extends this to task distributions, enabling generalization to unseen tasks using only offline data (Mitchell et al., 2020; Pong et al., 2021; Dorfman et al., 2021; Li et al., 2020). Prior methods often rely on temporal-difference (TD) learning, either with optimization-based meta-learning (Finn et al., 2017; Mitchell et al., 2020; Xu et al., 2018) or context-based inference (Yuan & Lu, 2022; Rakelly et al., 2019; Zintgraf et al., 2021; Li et al., 2024). Context-based approaches, such as PEARL (Rakelly et al., 2019) and VariBAD (Zintgraf et al., 2021), learn latent task representations from context trajectories to condition the policy. However, TD-based algorithms are prone to instability due to the deadly triad—bootstrapping, function approximation, and off-policy data (Levine et al., 2020a). Many also depend on hand-crafted constraints to keep policies within the support of the offline data distribution (Ajay et al., 2022). These challenges motivate alternative approaches that avoid bootstrapping and allow more stable, flexible generalization across tasks.

Language-conditioned RL. Language offers a natural way to specify tasks, and several methods explore this space. BC-Z (Jang et al., 2022) and BAKU (Haldar et al., 2024) demonstrate language-conditioned imitation learning, but both omit reward conditioning and are not designed for offline meta-RL. TTCT (Dong et al., 2024) leverages text for constraint decomposition but requires online environment interaction to adapt, limiting its applicability in offline domains. MineCLIP (Fan et al., 2022) and STEVE-1 (Lifshitz et al., 2023) show that large-scale vision-language pretraining can ground behavior in text, but they operate in online or imitation learning settings and are not return-conditioned. Most recently, a concurrent work T2DA (Zhang et al., 2025) introduces prompt-augmented trajectories by prepending natural language to behavior sequences. While promising, this approach performs implicit alignment between text and trajectories and does not address intra-task variation or the robustness issues posed by noisy offline data. By comparison, TG-DT introduces explicit dual alignment losses: contrastive learning ensures inter-task separation, while matching-based supervision captures intra-task quality differences. This design directly addresses the variability of offline data and leads to more reliable text-to-behavior grounding.

Vision-Language Alignment. Multi-modal models such as CLIP (Radford et al., 2021) and BLIP (Li et al., 2022) highlight the effectiveness of aligning text and images through large-scale contrastive learning. However, these approaches are designed for static perception tasks. Extending their ideas to sequential decision-making presents unique challenges: trajectories unfold over time, rewards vary in quality, and policies must generate coherent action sequences. TG-DT bridges this gap by aligning natural language with temporally extended trajectories, grounding sequential decision-making in language while preserving return conditioning. This shift from static alignment to dynamic control is central to TG-DT's novelty.

Summary. In summary, prior offline meta-RL methods rely on test-time demonstrations or interaction, while existing language-conditioned approaches either assume online adaptation or lack reward conditioning. TG-DT uniquely combines (1) a stricter offline assumption—zero-shot generalization from language only, with no test-time data; (2) a dual alignment mechanism that robustly links text to behavior across and within tasks; and (3) return-conditioned decision modeling to preserve the strengths of DT while extending it to language-based task specification. This combination positions TG-DT as a novel framework to achieve practical, zero-shot offline generalization guided solely by natural language.

Table 5. Training and testing task indexes when testing the generalization ability in unseen tasks.

Cheetah-dir	
Training set of size 2	[0, 1]
Testing set of size 2	[0, 1]
Cheetah-vel	
Training set of size 35	[0–1, 3–6, 8–14, 16–22, 24–25, 27–39]
Testing set of size 5	[2, 7, 15, 23, 26]
Ant-dir	
Training set of size 45	[0–5, 7–16, 18–22, 24–29, 31–40, 42–49]
Testing set of size 5	[6, 17, 23, 30, 41]
Meta-World ML10	
Training set of size 10	[0, 9, 19, 29, 33, 36, 39, 40, 48, 49]
Testing set of size 3	[11, 24, 41]
Meta-World ML45	
Training set of size 45	[0–10, 12–16, 18–24, 26–35, 37–40, 42–49]
Testing set of size 5	[11, 17, 25, 36, 41]

B. Implementation Details & Environment

Implementation Details. All results are averaged over 5 random seeds with standard deviation shown. Experiments were conducted on 4 NVIDIA RTX 6000 Ada GPUs (48GB), using PyTorch and the Hugging Face Transformers library (Wolf et al., 2019).

We evaluate **TG-DT** on two widely used benchmarks—**MuJoCo** (Todorov et al., 2012) and **Meta-World** (Yu et al., 2020)—to ensure fair and rigorous comparisons. These benchmarks pose diverse challenges such as sparse rewards, high-dimensional continuous control, and complex task variations, making them well-suited for testing generalization in sequence-based offline RL.

MuJoCo Tasks. We follow the experimental protocol from Prompt-DT (Xu et al., 2022) and evaluate TG-DT on three locomotion tasks: **Cheetah-dir**, **Cheetah-vel**, and **Ant-dir**, where agents are penalized for high-magnitude control signals. These environments are designed to assess both zero-shot and few-shot generalization across varied task specifications.

- **Cheetah-dir**: Consists of two tasks—running forward and backward. The agent receives a reward proportional to its velocity along the goal direction. Both training and test sets contain these two tasks.
- **Cheetah-vel**: Contains 40 tasks, each with a target velocity uniformly sampled from $[0, 3]$. The agent is penalized based on the l_2 distance from the target velocity. We hold out 5 tasks for testing and train on the remaining 35.
- **Ant-dir**: Includes 50 tasks with goal directions uniformly sampled in 2D space. The 8-joint ant is rewarded for velocity along the specified direction. We use 5 tasks for testing and 45 for training.

Meta-World Tasks. We use the **ML10** and **ML45** task suites, which involve controlling a Sawyer robot's end-effector to reach task-specific goals in 3D space. These benchmarks are standard in meta-RL research and provide a diverse set of robotic manipulation tasks.

- **ML10**: Comprises 10 training and 3 test tasks. Each task requires the robot to move its end-effector to a unique target location. The agent directly controls the XYZ position of the end-effector.
- **ML45**: Contains 45 training and 5 unseen test tasks, each with a different manipulation goal involving position or object interaction.

Dataset Construction. For MuJoCo tasks, we follow the offline data generation procedure used in (Mitchell et al., 2021), collecting trajectories using SAC (Haarnoja et al., 2018) or TD3 (Fujimoto et al., 2018), and applying penalties on large control inputs. We consider three dataset types:

- **Expert**: Generated using a well-trained SAC/TD3 policy.
- **Mixed**: Combines trajectories from partially and fully trained policies.

Method	Cheetah-dir	Cheetah-vel	Ant-dir	ML10	ML45
Prompt-DT†	548.9±185.2	-150.6±12.7	214.2±6.2	289.2±10.5	248.3±11.3
GDT	129.2±106.5	-218.4±15.8	167.9±18.5	169.8±11.2	153.2±17.9
Meta-DT†	539.6±14.7	-102.7±3.4	**357.5**±9.2	335.2±8.7	306.4±10.7
HDT†	445.3±13.2	-162.7±20.5	215.4±10.2	266.4±8.8	245.7±12.3
DPDT†	548.1±12.1	-142.6±17.9	321.8±8.0	360.4±6.2	**311.8**±9.1
BC-Z†	310.5±23.0	-123.9±10.1	254.9±10.1	199.7±21.2	157.4±10.9
BAKU	348.6±36.8	-110.5±7.9	277.2±12.6	216.3±12.0	163.2±9.7
TG-DT (Ours)	**549.9**±**176.2**	**-93.0**±**10.2**	328.3±5.0	**361.1**±**5.1**	309.6±13.4

Table 6. Zero-shot test returns of TG-DT against baselines using Medium datasets. We report average returns ± standard deviations over five runs (higher is better). † denotes methods that require test-time environment interaction to obtain adaptation demonstrations. TG-DT requires no interaction.

- **Medium**: Collected from mid-performance policies.

For Meta-World tasks, we use expert demonstrations generated by scripted controllers provided in the environment suite.

Task Splits. Following Prompt-DT (Xu et al., 2022), we illustrate the training and testing task distributions for each benchmark in Table 5. All experiments strictly follow these splits for consistency.

BAKU (Haldar et al., 2024) with Few-Shot Extension. BAKU is a multi-task imitation learning approach. For fair comparison, we evaluate a variant of BAKU where the few-shot dataset from the target task is included as additional training data. This effectively treats the unseen task as an extra training task, which is outside BAKU's original design (it assumes multi-task training without task-specific adaptation). While this extension provides BAKU with privileged access to the few-shot trajectories, it still underperforms TG-DT. This highlights that simply exposing BAKU to more data is insufficient; TG-DT's advantages stem from its return conditioning and explicit text–behavior alignment, which enable robust generalization under strict offline meta-RL constraints.

Complete Results. Tab. 6 include the complete results of zero-shot setting.

Implementation Details. All evaluated methods are carried out with 5 different random seeds, and the mean of the received return is plotted with standard deviation. All experiments were conducted on a server equipped with 4 NVIDIA RTX 6000 Ada GPUs (48GB each), using PyTorch and the Hugging Face Transformers library (Wolf et al., 2019).

C. Hyperparameters Configuration

We show the hyperparameter of TG-DT in Table 7.

D. Example Task Descriptions

Below we list the example task descriptions of the five testing tasks, where task statistics and intents are inferred and estimated from training tasks:

- **Cheetah-dir**: "This is the Cheetah-dir task, which targets running in the forward direction. Its corresponding environment is the MuJoCo HalfCheetah-v2, a planar 9-DoF robot trained to maximize velocity along the X-axis. Good demonstrations typically have an episode length of 1000 steps and yield an expected return of 1,000. This demonstration yields a return of 1,000."
- **Cheetah-vel**: "This is the Cheetah-vel task, which targets maintaining a velocity of around 2.3 m/s. Its corresponding environment is the MuJoCo HalfCheetah-v2, where the agent is penalized by the squared distance from the target velocity at each timestep. Good demonstrations typically have an episode length of 1000 steps and yield an expected return of -20. This demonstration yields a return of -20."
- **Ant-dir**: "This is the Ant-dir task, which targets movement toward around 135 degrees. Its corresponding environment is an 8-joint quadruped ant robot. Good demonstrations typically have an episode length of 1000 steps and yield an expected

Table 7. Hyperparameters used in our experiments.

Hyperparameters	**Value**
Pretraining model	BLIP
K (number of shared data)	1
Training batch size	16
Number of evaluation episodes	50
Learning rate	1e-4
Weight decay	1e-4
Number of layers	3
Number of attention heads	1
Embedding dimension	128
Activation	ReLU
r	1
Dropout	0.1
Device	CUDA
Max iterations	5000
Warmup steps	10000
Save interval	500

return of 1000. This demonstration yields a return of 1000."

- **MetaWorld ML10**: "This is the ML10 reach-target task, which targets moving the end-effector to a designated 3D location. Its corresponding environment is a tabletop robotic arm setup using the Sawyer robot. Good demonstrations typically have an episode length of 150 steps and yield an expected return of 550. This demonstration yields a return of 550."
- **MetaWorld ML45**: "This is the ML45 open-drawer task, which targets opening a drawer to a specific position. Its corresponding environment is a robotic manipulation scene involving multiple object interactions. Good demonstrations typically have an episode length of 200 steps and yield an expected return of 450. This demonstration yields a return of 450."