# MGDA: Model-based Goal Data Augmentation for Offline Goal-conditioned Weighted Supervised Learning

**Xing Lei**[1], **Xuetao Zhang** [1*], **Donglin Wang**[2]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]School of Enginneering, Westlake University
leixing@stu.xjtu.edu.cn, xuetaozhang@xjtu.edu.cn, wangdonglin@westlake.edu.cn

## Abstract

Recently, a state-of-the-art series of algorithms—Goal-Conditioned Weighted Supervised Learning (GCWSL) methods—has been introduced to address the challenges inherent in offline goal-conditioned reinforcement learning (RL). GCWSL optimizes a lower bound on the goal-conditioned RL objective and has demonstrated exceptional performance across a range of goal-reaching tasks, offering a simple, effective, and stable solution. Nonetheless, researches has revealed a critical limitation in GCWSL: the absence of trajectory stitching capabilities. In response, goal data augmentation strategies have been proposed to enhance these methods. However, existing techniques often fail to effectively sample appropriate augmented goals for GCWSL. In this paper, we establish unified principles for goal data augmentation, emphasizing goal diversity, action optimality, and goal reachability. Building on these principles, we propose a Model-based Goal Data Augmentation (MGDA) approach, which leverages a dynamics model to sample more appropriate augmented goals. MGDA uniquely incorporates the local Lipschitz continuity assumption within the learned model to mitigate the effects of compounding errors. Empirical results demonstrate that MGDA significantly improves the performance of GCWSL methods on both state-based and vision-based maze datasets, outperforming previous goal data augmentation techniques in their ability to enhancing stitching capabilities.

## Introduction

Deep reinforcement learning (RL) empowers agents to attain sophisticated objectives in complex and uncertain environments, such as computer games (Oroojlooyjadid et al. 2022; Sestini, Bagdanov et al. 2023), robot control (Dargazany 2021; Quiroga et al. 2022; Plasencia-Salgueiro 2023), and language processing (Akakzia et al. 2020; Sharifani and Amini 2023). One of the central challenges in deep RL is facilitating efficient learning in environments characterized by sparse rewards. This issue is particularly acute in goal-conditioned RL (GCRL) (Kaelbling 1993; Schaul et al. 2015; Andrychowicz et al. 2017; Liu, Zhu, and Zhang 2022), where the agent is tasked with learning generalized policies that can reach a variety of goals. Offline GCRL is especially promising because it allows for the learning of goal-conditioned policies from purely offline datasets without requiring any interaction with the environment during the learning process (Levine et al. 2020). We note that some goal-conditioned weighted supervised learning (GCWSL) methods (Yang et al. 2022; Ma et al. 2022b; Hejna, Gao, and Sadigh 2023; Sikchi et al. 2024) were proposed to tackle the offline GCRL challenges. Compared with other goal-conditioned RL or self-supervised (SL) methods (Yang et al. 2019; Srivastava et al. 2019; Chen, Paleja, and Gombolay 2020; Ding et al. 2019; Lynch et al. 2020; Paster, McIlraith, and Ba 2020; Eysenbach, Salakhutdinov, and Levine 2020; Ghosh et al. 2021; Eysenbach et al. 2022), GCWSL has demonstrated outstanding performance across various goal-reaching tasks in a simple, effective, and stable manner.

Although GCWSL has been successfully applied to effectively learn from sparse rewards in certain goal-reaching tasks within offline GCRL, some studies (Brandfonbrener et al. 2022; Yang et al. 2023; Ghugare et al. 2024) indicate that GCWSL may leads to sub-optimal policies when applied to the corresponding sub-trajectories of (state, goal) pairs across different trajectories and identify this issue as lacking of the ability to stitch trajectories. To tackle this problem, (Yang et al. 2023) and (Ghugare et al. 2024) adopt goal data augmentation to sample more (state, goal) pairs during training phase. Goal data augmentation is an effective data-driven approach that can generate diverse (state, goal) pairs. Therefore, it can enable GCWSL methods to exhibit stitching property, highlighting the strengths of supervised learning (SL).

However, our analysis indicates that they often fail to select appropriate goals as augmented goals. This paper investigates this issue and proposes a more advanced goal data augmentation method. Specifically, we establish three goal data augmentation unified principles, grounded in both the properties of data augmentation and the inherent characteristics of GCWSL : ***Goal Diversity***, ***Action Optimality***, and ***Goal Reachability***. ***Goal Diversity*** implies that a single initial state can correspond to multiple goals. ***Action Optimality*** implies that the action corresponding to a initial state should remain optimal for reaching the augmented goals. ***Goal Reachability*** ensures that the augmented goals are reachable. Building on these principles, we propose

---

[*] Corresponding author

a model-based goal data augmentation (MGDA). MGDA leverages a dynamics model to predict nearby states around original goal, and then sample new augmented goals in the trajectory of these states. To reduce the conformity error caused by model prediction, we specifically employed the local Lipschitz continuity assumption when learning the dynamics model from the dataset.

We briefly summarize our contributions:

- To the best of our knowledge, this work is the first to establish a set of principles specifically designed to enhance the stitching capabilities of GCWSL methods. Moreover, these principles are broadly applicable and can be extended to other goal-conditioned SL frameworks.
- We introduce a model-based data augmentation method called MGDA, which leverages a learned environment dynamics model grounded in the local Lipschitz continuity assumption. We demonstrate that MGDA not only adheres to the established principles but also provides theoretical guarantees for the accurate prediction of augmented goal labels.
- In experiments conducted on offline maze datasets specifically designed to assess stitching capabilities, the integration of MGDA into GCWSL has demonstrated superior performance. Compared to other goal data augmentation methods such as Swapped Goal Data Augmentation (SGDA) (Yang et al. 2023) and Temproal Goal Data Augmentation (TGDA) (Ghugare et al. 2024), MGDA shows significant improvements when they are all added to GCWSL. Our further analysis and ablation studies highlight the crucial role played by the local Lipschitz continuity assumption in achieving these results.

## Related Work

**The Stitching Property.** The concept of stitching, as discussed by (Ziebart et al. 2008), is a characteristic property of temporal-difference (TD) learning algorithms such as those described by (Mnih et al. 2013; Lillicrap et al. 2015; Fujimoto, Hoof, and Meger 2018; Kostrikov, Nair, and Levine 2021), which employ dynamic programming techniques. This property allows these algorithms to integrate data from various trajectories, thereby enhancing their effectiveness in managing complex tasks by leveraging historical data (Cheikhi and Russo 2023). On the other hand, most SL-based RL methods lack this property. (Ghugare et al. 2024) indicates from the perspective of combinatorial generalization that typical SL methods such as DT (Chen et al. 2021) and RvS (Emmons et al. 2021) do not perform stitching. The same situation also exists in offline GCRL (Yang et al. 2023). In offline GCRL, GCWSL methods (Yang et al. 2022; Ma et al. 2022b; Hejna, Gao, and Sadigh 2023; Sikchi et al. 2024) find a single optimal trajectory corresponding to a given (state, goal) pair and have demonstrated strong performance in various goal-reaching tasks. However, these methods are sub-optimal for some unseen skills and lack the ability to stitch information from multiple trajectories like most SL-based RL methods.

**Local Lipschitz Continuity in Environment Dynamics.** Local continuity ensures that small changes in actions or states result in correspondingly small changes in transitions, a characteristic that aligns with physical laws and is observed in the dynamics of many robotic systems and real-world scene. And it in dynamics is commonly applied in classical control methods to ensure the existence and uniqueness of solutions to differential equations (Li and Todorov 2004; Bonnard, Caillau, and Trélat 2007). This assumption is particularly valuable in nonlinear systems and is widely employed in robotic applications (Seto, Annaswamy, and Baillieul 1994; Kahveci 2007; Sarangapani 2018). However, these methods often require pre-specified models and cost functions within an optimal control framework when leveraging dynamics continuity. In contrast, we apply the local Lipschitz continuity assumption when learning the dynamics model to predict augmented goals for GCWSL agents. Perhaps the most comparable work to ours is that of (Ke et al. 2024), but a key difference is that we use this assumption solely for prediction rather than for generation, which may lead to greater estimation errors.

**Data Augmentation in RL.** Data augmentation, recognized as an effective technique for enhancing generalization, has been widely applied in both RL (Srinivas, Laskin, and Abbeel 2020; Lu et al. 2020; Stone et al. 2021; Kalashnikov et al. 2021; Hansen and Wang 2021; Kostrikov, Yarats, and Fergus 2021; Yarats et al. 2021) and SL (Shorten and Khoshgoftaar 2019). We have observed that some methods (Char et al. 2022; Yamagata, Khalil, and Santos-Rodriguez 2023; Paster et al. 2023) leverage dynamic programming to augment existing trajectories, thereby improving the performance of SL algorithms. However, these methods still rely on dynamic programming. Another closely related approach focuses on data augmentation exclusively for SL (Yang et al. 2023; Ghugare et al. 2024) without employing dynamic programming. These methods are simple and efficient enough to enhance the stitching ability of the SL method. Nevertheless, our analysis indicates that they still encounter challenges in accurately providing augmented goal data.

## Background

**Goal-conditioned Reinforcement Learning (GCRL).** GCRL can be characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{G}, \gamma, \rho_0, T, f, r \rangle$, where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{G}$, $\gamma$, $\rho_0$, $T$ refer to state space, action space, goal space, discounted factor, the distribution of initial states and the horizon of the episode, respectively. $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}'$ is the ground truth dynamic transition, and $r : \mathcal{S} \times \mathcal{G} \times \mathcal{A} \rightarrow \mathbb{R}$ is typically a simple unshaped binary signal. And the objective of $\pi(a|s, g)$ is maximizing returns of reaching goals from the goal distribution $p(g)$:

$$\mathcal{J}(\pi) = \mathbb{E}_{\substack{g \sim p(g), s_0 \sim \rho_0, \\ a_t \sim \pi, s_{t+1} \sim f(\cdot|s_t, a_t)}} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, g) \right]. \quad (1)$$

In this paper, the sparse reward function $r$ is defined as indicator function:

$$r(s_t, a_t, g) = \begin{cases} 1, & \|\phi(s_t) - g\| < \delta \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where $\delta$ is a threshold and $\phi : \mathcal{S} \rightarrow \mathcal{G}$ is a known state-to-goal mapping function from states to goals. And the goal is considered reached when $\|\phi(s_t) - g\| < \delta$ (Andrychowicz et al. 2017).

**Offline GCRL**. In offline RL setting, the agent can only access a static offline dataset $\mathcal{D}$ and cannot interact with the environment to maximize the objective in Equation 1. The offline dataset $\mathcal{D}$ can be collected by some unknown policies (Levine et al. 2020; Prudencio, Maximo, and Colombini 2023). Based on the definition of GCRL, we further express the offline dataset as $\mathcal{D} := \{\tau_i\}_{i=1}^{N}$, where $\tau_i := \{< s_0^i, a_0^i, r_0^i >, < s_1^i, a_1^i, r_1^i >, ..., < s_T^i, a_T^i, r_T^i >, g_i\}$ is the goal-conditioned trajectory and $N$ is the number of stored trajectories. In $\tau_i$, $s_0 \sim \rho_0$. The desired goal $g_i$ is still randomly sampled from $p(g)$. Relabeled goals can be derived from each state as $g_t^i = \phi(s_t^i)$ for $0 \leq t \leq T$. It should be noted that trajectories may be unsuccessful trajectories (i.e, $g_T^i \neq g_i$).

**Goal-conditioned Weighted Supervised Learning (GCWSL)**. Different with general goal-conditioned RL methods that directly maximizes discounted cumulative return, GCWSL provides theoretical guarantees that weighted supervised learning from hindsight relabeled data optimizes a lower bound on the goal-conditioned RL objective in offline GCRL. During training, trajectories are sampled form a relabeled dataset by utilizing hindsight mechanisms (Kaelbling 1993; Andrychowicz et al. 2017). And the policy optimization satisfies the following definition:

$$\mathcal{J}_{GCWSL}(\pi) = \mathbb{E}_{(s_t, a_t, g) \sim \mathcal{D}_r} \left[ w \cdot \log \pi_\theta(a_t | s_t, g) \right], \quad (3)$$

where $\mathcal{D}_r$ denotes relabeled data, $g = \phi(s_i)$ denotes the relabeled goal for $i \geq t$. The weighted function $w$ exists various forms in GCWSL methods (Yang et al. 2022; Ma et al. 2022a; Hejna, Gao, and Sadigh 2023; Sikchi et al. 2024) and can be considered as the scheme choosing optimal path between $s$ and $g$. Therefore GCWSL includes typical two process, acquiring sub-trajectories corresponding to $(s, g)$ pairs and imitating them. In the process of imitation, GCWSL first train the specific weighted function $w$, and then extract the policy with the Equation 3.

## Generalized Principles and Related Works of Goal Data Augmentation for GCWSL

Data augmentation has become a common approach for improving generalization. In offline GCRL scenarios, it can also serve as an effective strategy to tackle unseen skills. Given the emergence of various methods, we utilize the characteristics of data augmentation and develop more general methods for competitive GCWSL methods in offline GCRL. This section proposes for the first time three principles of satisfying goal data augmentation for GCWSL methods. These principles also can be applied to other goal-conditioned SL methods. Moreover, we compare the differences among following goal augmentation methods with our MGDA using a counter example: SGDA (Yang et al. 2023) proposes a method that randomly choose augmented goals from different trajectories. TGDA (Ghugare et al. 2024) proposed a another goal data augmentation approach from the perspective of combinatorial optimization. It employs k-means to cluster the goal and certain states into a group, and samples goals from later stages of these state trajectories as augmented goals.

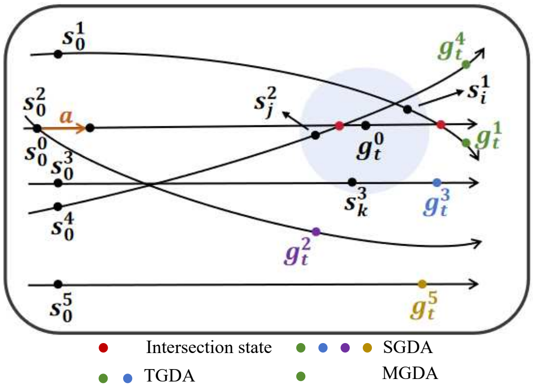

Figure 1: Counter examples of generalized principles and related goal data augmentation methods. The states $s_0^0$ through $s_0^5$ correspond to trajectories $\tau_0$ through $\tau_5$, respectively. The goals $g_t^0$ through $g_t^5$ represent the relabeled goals for each respective trajectory. Note that $s_0^0$ is equal to $s_0^2$ and red points denote intersection state of two trajectories. The light blue circles represent the nearby states of $g_t^0$ after k-means clustering within the TGDA (Ghugare et al. 2024) . Compared to the original goal $g_t^0$, SGDA (Yang et al. 2023) randomly select all goals $g \in \left[g_t^1, g_t^2, g_t^3, g_t^4, g_t^5\right]$ as augmented goals, while TGDA selects the goals $g \in \left[g_t^1, g_t^3, g_t^4\right]$ from later in the trajectory corresponding to the nearby states $s \in \left[s_i^1, s_j^2, s_k^3\right]$. Our MGDA method will select more appropriate goals $g \in \left[g_t^1, g_t^4\right]$, building upon TGDA by avoiding unreachable goals.

Here we assume that we need to perform goal data augmentation for $(s_0^0, g_t^0)$ pair as shown in Figure 1. To select more appropriate augmentation goals for $g_t^0$, we established three guiding principles for goal data augmentation. Below, we elaborate on the specifics and motivation behind these three principles.

***Goal Diversity.*** This is a fundamental principle for goal data augmentation, which implies that for a given state $s_0^0$, multiple goals can be associated with it. These multiple goals should come from different trajectories.

***Action Optimality.*** This principle asserts that after selecting a goal, the original action corresponding to state $s_0^0$ can still be considered the optimal action for the augmented goal. It is also recognized in the TGDA (Ghugare et al. 2024). This principle helps to reduce the redundancy in goal data augmentation. In other words, the stitched trajectories of augmented goals can be treated as the optimal trajectories corresponding to the $(s_0^0, g_t^0)$ pair, thereby simplifying the training complexity of GCWSL. For instance, we choose $g_t^2$ as the augmented goal for the original state $s_0^0$. Even if it is reachable for $s_0^0$, the optimal action from $s_0^0$ to $g_t^2$ no longer align with the original action $a$. In this way $g_t^2$ cannot be used as an augmented goal for $g_t^0$, but only as another goal for $s_0^2$ that needs to be reached by re-learning. Since SGDA selected $g_t^2$, it does not satisfy *Action Optimality.*

***Goal Reachability.*** This principle emphasizes that when selecting augmented goals from other trajectories for the original state $s_0^0$, the chosen goals must be reachable. Specifically, the original trajectory and other trajectories must share a intersection state. For instance, in Figure 1, trajectories $\tau_3$ and $\tau_5$ do not intersect with the original trajectory $\tau_0$, rendering the chosen goals $\left[g_t^3, g_t^5\right]$ ineffective. SGDA exists cases where the augmented goal data are unreachable such as $\left[g_t^3, g_t^5\right]$. TGDA also faces the existence of unreachable goals. As shown in Figure 1, assume a wall between trajectory $\tau_0$ and $\tau_3$, k-means might still identify unreachable goals $g_t^3$ as augmented goals, which is unreasonable.

In summary, ***Goal Diversity*** is a fundamental guideline that most data augmentation methods aim to achieve, while ***Action Optimality*** and ***Goal Reachability*** are specifically tailored for offline GCRL, building upon ***Goal Diversity***. Our MGDA approach overcomes the limitations of previous goal data augmentation methods and adheres to all three principles. Detailed comparisons can be found in Table 1.

| **Methods** | **Metric** | ***Goal Diversity*** | ***Action Optimality*** | ***Goal Reachability*** |
|---|---|---|---|---|
| SGDA | - | ✓ | ✗ | ✗ |
| TGDA | L2 | ✓ | ✓ | ✗ |
| **MGDA** | Dynamics | ✓ | ✓ | ✓ |

Table 1: Comparison of different goal data augmentation methods for GCWSL.

## MGDA: Model-based Goal Data Augmentation

In this section, we describe MGDA in detail. The core idea is to **identify nearby states relative to the original goal and then sample an augmented goal from the latter part of the trajectory corresponding to these nearby states to serve an augmentation of the original goal. And these nearby states must successfully reach the goal**. In this case, the original goal functions as an intersection state, linking two different trajectories.

We begin by detailing the process of learning a dynamics model that leverages the local Lipschitz continuity assumption to accurately predict nearby states and sample augmented goals. Following this, we provide both practical implementation and theoretical justification for MGDA.

### Sample Augmented Goals with MGDA

A critical step in our approach involves learning the dynamics model from data and utilizing it for state prediction by leveraging the local continuity of the environment. When the dynamics model is locally Lipschitz bounded, small variations in state and action lead to correspondingly small changes in transitions. A dynamics function with local continuity enables us to accurately identify nearby states around the original goal that satisfy the dynamic transition conditions observed in the training data within this region, the reliability of the learned model can be assured. We identify the nearby state under the metric defined by the dynamics model, as illustrated in Figure 2. In this figure, we denote the correct nearby state as $s_n$ and the learned dynamics function is represented as $\hat{f}$. Additionally, the action corresponding to $s_n$ is denoted as $a_n$. To guarantee accurate predictions of nearby state $s_n$, we follow the model-based reinforcement learning frameworks proposed in Gulrajani et al. (2017) and Ke et al. (2024) to enforce local continuity on the dynamics model's learning process:

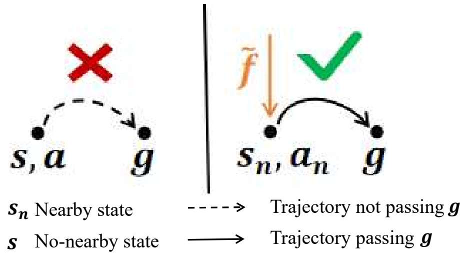

Figure 2: (*left*) State searched by other goal data augmentation methods. (*right*) State searched by our MGDA, constrained by the dynamics model in its relationship to the goal. MGDA ensures that the searched state and goal $g$ satisfy the one-step transition criterion, thereby defining this searched state as a nearby state $s_n$.

$$\begin{aligned} \arg\min_{\hat{f}} \mathbb{E}_{s_n, a_n, g \sim \mathcal{D}} \big[ \sigma(\lambda_n) \cdot \| g - s_n - \hat{f}(s_n, a_n) \| \\ + \sum \sigma(\lambda_n) \big], \text{while } W \rightarrow W / \max(\frac{\|W\|_2}{\lambda}, 1) \end{aligned} \quad (4)$$

where $\lambda_n$ is a state-dependent slack variable used to dynamically adjust the weight of each data point in the loss function, $\lambda$ is a global regularization parameter in the training objective that controls the overall penalty strength for enforcing local Lipschitz continuity in the model, $\sigma$ is the Sigmod function and $W$ is a weight matrix that scales the dynamics model's predictions. Equation 4 introduces a modified form of the mean squared error (MSE) loss, which we employ to fit the function $\hat{f}$ with a neural network and predict the changes in both the nearby state $s_n$ and the goal $g$.

Then we can ensure that the approximate model remains predominantly $L$- Lipschitz bounded while accurately predicting transitions in the given offline data. As a result, this approximate dynamics model can be effectively used to predict corrective labels.

**Theorem 1 (model smoothness)** *As described in Figure 2, under the assumption of local Lipschitz continuity, when the error generated by training the model is $\epsilon$, the 1-step residual dynamics model $\hat{f}$ is subject to the following boundary for predicting the correct nearby states $s_n$ of goal $g$.*

$$\left\| f\left(s_n, a_n\right) - \hat{f}\left(s_n, a_n\right) \right\| \leq \epsilon + \left(K + \Delta(\lambda_n)\right) \|s_n - g\|, \quad (5)$$

*where $K$ is the Lipschitz constants for true environment dynamics $f$ and $\Delta(\lambda_n)$ is the Lipschitz error controlled by $\lambda_n$ for learned dynamics model $\hat{f}$.*

The full proof can be found at supplementary material. This theorem demonstrates that when the dynamics function is governed by a local Lipschitz constant, the prediction error of the dynamics model can be quantified accordingly based on that constant.

After searching for the nearby states of original goal, we sample a new augmented goal from later in that trajectory, as shown in Figure 1. This approach provides a straightforward method for sampling cross trajectory goals while ensuring that the action remains optimal at original state.

## Practical Implementation of MGDA and Analysis

To expedite the identification of nearby states, our MGDA algorithm first employs the k-means technique to cluster all states into multiple groups, as proposed by (Ghugare et al. 2024). Subsequently, we train a transition model on the data and search for nearby states within the permissible range of model error. Specifically, we first randomly select a initial state within the same group as the original goal and apply the dynamic function for joint forward inference to generate the resulting state. And then if this resulting state reach the original goal (i.e, the L2 distance between the resulting state and the original goal is less than $\delta$) , we designate this initial state as nearby state to original goal. Finally, a new goal is randomly chosen as an augmentation goal for original goal from the trajectory following nearby state. Therefore MGDA can be considered as model-augmented version of TGDA. The complete algorithm for adding MGDA to GCWSL is illustrated in Algorithm 1. The blue text represents the entire process of MGDA.

Algorithm 1: MGDA for GCWSL Methods

```
Input: Offline Dataset $\mathcal{D}$.
Initialize: Policy $\pi_\theta(a|s,g)$ with parameter $\theta$.
Function Learn Dynamics $\hat{f}$
    Optimize the objective in Equation 4.
while a fixed number of iteration do
    for $t = 1, \cdots, m$ do
        Relabel and sample $\{(s_t, a_t, g) \sim \mathcal{D}$, where $g = \phi(s_i), i \geq t$.
        Get the group of the goal with k-means: $k = d_{t+}$, where $d_l$ = CLUSTER($s_l$).
        Randomly sample candidate state from the same group: $u \sim \{s_j; \forall j$ such that $d_j = k\}$.
        if $\|g - u - \hat{f}(s_t, a_t)\| < \delta$ then
            Identify $s_n := u$ as a nearby state.
            Sample new goal $\tilde{g}$ from the later stages in the trajectory of $s_n$.
            Augment the goal $g = \tilde{g}$.
        end if
        Calculate the weight $w$ in original GCWSL methods and Update $\pi_\theta$ with maximize goal-conditioned policy optimization with Equation 3.
    end for
end while
Return: Goal-conditioned policy $\pi_\theta(a|s,g)$
```

While data augmentation methods generally lack theoretical guarantees, we demonstrate that MGDA approximates a one-step stitching policy under the smooth assumption of certain distributions. The proposed new model-based method MGDA can generate (state, goal) combinations that were not encountered during training, effectively mimicking the stitching process observed in TD learning.

If we assume that our offline dataset $\mathcal{D} := \{s_0^i, a_0^i, \ldots\}_{i=1}^N$ is collected by a set of policies $\{\beta(a \mid s, h)\}$ where $h$ specifies some context from distribution $p(h)$ and $\beta_h := \beta_h(a|s,h)$ (Ghugare et al. 2024). We denote $p_+^\pi(g \mid s, a) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^\pi(s_t = g \mid s_0 = s)$ is the discounted state occupancy distribution for a goal-conditioned policy $\pi$. We first have follow assumption (Drawing inspiration from Appendix D.2 in Ghugare et al. (2024)):

**Assumption 1 (distribution smoothness)** *For all $s, a, g$ pairs and all data collecting policies $\beta_h$, $s_n$ and $s_n'$ are the states corresponding to the reachable and unreachable goals in $u$, respectively. The $p_+^{\beta_h}(g \mid s, a)$ is $L_1$-Lipschitz continuous with respect to $s_n$ and $L_2$-Lipschitz continuous with respect to $s_n'$:*

$$|p_+^{\beta_h}(g \mid s, a) - p_+^{\beta_h}(s_n \mid s, a)| \leq L_1(||g - s_n||) \tag{6}$$

*and:*

$$|p_+^{\beta_h}(g \mid s, a) - p_+^{\beta_h}(s_n' \mid s, a)| \leq L_2(||g - s_n'||) \tag{7}$$

Intuitively, the above all smoothness result ensures that all intra-group states under the data collection policy have similar probabilities. For example, after taking action $a$ from state $s$, there is an equal probability of reaching nearby states $s_n$ and $g$. And then we have follow guarantee for MGDA:

**Theorem 2** *Given the aforementioned smoothness assumption, the model-based goal data augmentation $p^{MGDA}(g \mid s, a)$ approximates the sampling process of goals according to the distribution defined by the one-step goal-reaching stitching policy $p^{1\text{-}step}(g \mid s, a)$ for all $s, a, g$ pairs:*

$$p^{MGDA}(g \mid s, a) = p^{1\text{-}step}(g \mid s, a) \pm \mathcal{O}(\epsilon_k L_1), \tag{8}$$

*where $\epsilon_k$ is the maximum cutoff distance between all states within the group of $g$ after clustering.*

This theorem demonstrates that a single application of model-based goal data augmentation samples $(s, g)$ pairs from the 1-step goal-reaching distribution.

## Experiments

**Datasets**. To rigorously evaluate the stitching capabilities of GCWSL methods, we employ the offline point maze dataset configuration as outlined in (Ghugare et al. 2024). For this evaluation, we modify the GCWSL policy to navigate between previously unseen combinatorial (state, goal) pairs and subsequently measure the success rate.

**Baselines**. We conducted a series of comparative experiments by implementing the GCWSL methods within the same framework, as well as related goal data augmentation approaches. Specifically, all GCWSL implementations are based on DWSL (Hejna, Gao, and Sadigh 2023), with hyperparameter values set to the default values specified in

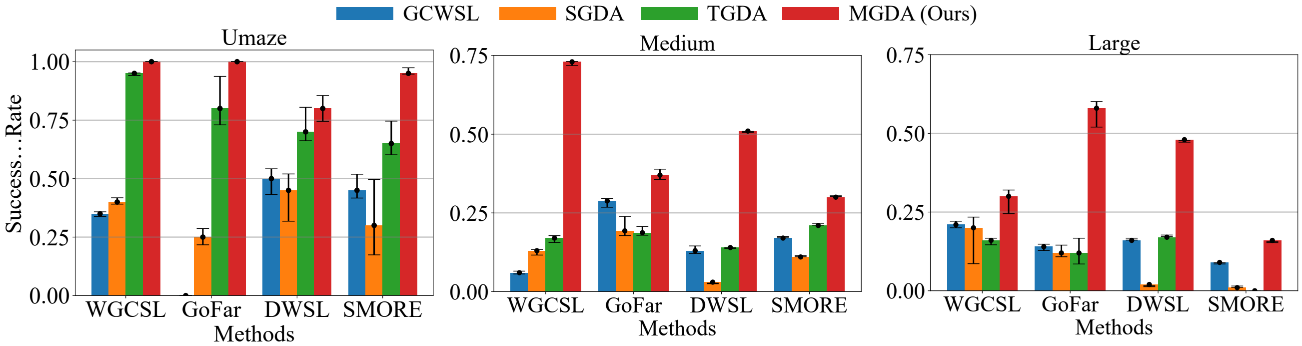

Figure 3: Performance of the original GCWSL methods and the impact of incorporating different goal augmentation approaches in state-based datasets. We use the final mean success rate as the report. Error bars denote 95% bootstrap confidence intervals.

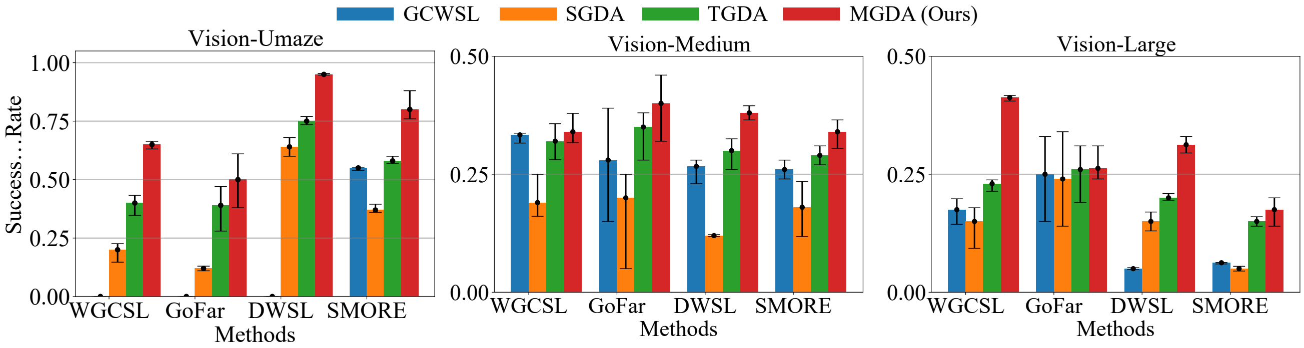

Figure 4: Performance comparison between the original GCWSL approach and its enhancement with MGDA on vision-based datasets. We also use the final mean success rate as the report. Error bars denote 95% bootstrap confidence intervals.

the original studies. We consider four competitive baseline GCWSL algorithms: **WGCSL** (Yang et al. 2022), **GoFar** (Ma et al. 2022a), **DWSL** (Hejna, Gao, and Sadigh 2023), and **SMORE** (Sikchi et al. 2024).

All goal data augmentation implementations including **SGDA** (Yang et al. 2023) and **TGDA** (Ghugare et al. 2024) are based on TGDA (Ghugare et al. 2024), while utilizing the same augmentation probability. All experiments are conducted using five random seeds. Detailed algorithm implementations and hyperparameter settings are provided in the supplementary material.

## Experimental Results

In this section we seek to answer the following questions: 1) *How does GCWSL perform when combined with our MGDA method? Does more data remove the need for goal data augmentation?* 2) *Can MGDA be effective for high-dimensional tasks?* 3) *How does MGDA compare in performance to other data augmentation methods?* 4) *How essential is the local Lipschitz continuity assumption?*

**State-based Dataset Results**. As shown in Figure 3, it is evident that all GCWSL algorithms struggle to demonstrate stitching properties, particularly in the complex Medium and Large tasks, where their performance is notably poor. However, when MGDA is incorporated into the GCWSL methods, performance improvements were observed across all tasks, albeit to varying degrees. This enhancement is attributed to the fact that goal data augmentation allows for the sampling of unseen (state, goal) combinations during the training phase, thereby improving the generalization and stitching capabilities of the models.

**Vision-based Datasets Results.** Figure 4 demonstrates that MGDA similarly enhances the performance of all GCWSL methods, indicating its effectiveness in high-dimensional goal-reaching tasks and its overall robustness. However, in the complex Vision-Medium and Vision-Large tasks, the results were inconsistent, with some instances showing minimal improvement over the original GCWSL methods. This suggests that while MGDA offers benefits, its robustness may be limited in certain scenarios, and future research should focus on developing more robust and scalable approaches to address these challenges.

**Comparison with Related Goal Data Augmentation Approaches.** Based on previous discussion, we selected SGDA and TGDA as baseline methods, as outlined in Table 1. As shown in Figure 3, our MGDA method consistently outperforms the other data augmentation approaches across all tasks, particularly in the more complex Medium and Large datasets. SGDA consistently exhibits poor performance across all tasks. While TGDA performs well in the Umaze dataset, it fails to achieve comparable results in the more complex Medium and Large tasks, often underperforming relative to SGDA, which suggests a lack of robustness. In contrast, MGDA not only achieves superior perfor-

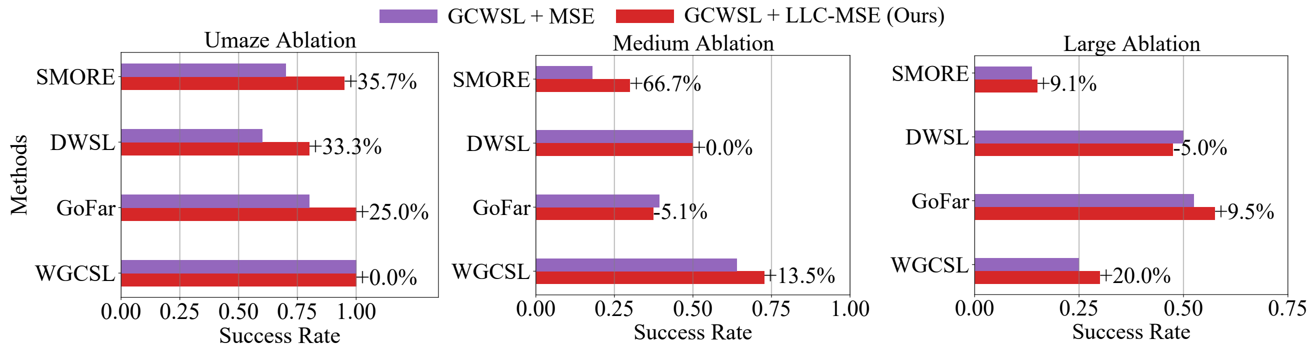

Figure 5: Ablation study on the local Lipschitz continuity (LLC) assumption. The results clearly show that the modified MSE version, incorporating LLC, outperforms the standard MSE-based dynamics model. This highlights the crucial role of the LLC assumption in enhancing performance.

| Dataset | DWSL with Different Dataset Size | | | | **DWSL** ($10^5$) | **DWSL** ($10^6$) | **DWSL** ($10^7$) | **DWSL** ($10^8$) |
|---|---|---|---|---|---|---|---|---|
| | $10^5$ | $10^6$ | $10^7$ | $10^8$ | | | | |
| Umaze | 50.2±3.9 | 50.6±8.0 | 55.5±4.5 | 63.7 ±2.9 | 72.4±3.3 | 80.7±2.7 | 87.6±8.6 | 91.2±1.8 |
| Medium | 21.3±2.3 | 22.3±2.3 | 24.1±1.5 | 28.3±5.8 | 52.6±9.2 | 56.8±5.3 | 58.2±6.1 | 63.4±2.6 |
| Large | 11.3±4.9 | 16.3±6.7 | 15.8±8.9 | 21.3 ±7.3 | 36.7±4.5 | 47.5±6.1 | 52.5±1.8 | 56.1±3.9 |
| *Total* | *82.8* | *89.2* | *95.4* | *113.3* | 161.7 | *179.0* | 198.3 | 210.7 |

Table 2: We report the mean success rate (%) across different dataset sizes, computed over five seeds, with each seed evaluated on 100 trajectories. **DWSL** ($10^n, n = 5, 6, 7, 8$) indicates the performance of the DWSL algorithm with MGDA integration on datasets of size $10^n$.

mance across all tasks but also demonstrates greater robustness compared to both SGDA and TGDA.

**Ablation Study.** Here we investigate the necessity of goal data augmentation for the highly-regarded algorithm DWSL (Hejna, Gao, and Sadigh 2023) within the GCWSL framework. The conventional wisdom suggests that larger datasets generally result in better generalization. To empirically test whether this is the case, we trained original DWSL on four different dataset sizes and then adding it with MGDA on 10 million transitions. As shown in Table 2, increasing the dataset size did not result in improved performance for DWSL. Due to space constraints, additional results are provided in the supplementary material. Based on the results of all necessary experiments, simply increasing the size of the dataset does not enhance the generalization capability of the GCWSL methods. This suggests that traditional methods of expanding datasets may not effectively increase trajectory diversity. However, goal data augmentation can address this issue.

Finally, to assess the significance of the local Lipschitz continuity assumption, we compared MGDA with a variant of goal data augmentation that relies solely on standard MSE loss for learning the dynamics model, thereby omitting the local Lipschitz continuity assumption. As shown in Figure 5, the local Lipschitz continuity assumption generally leads to superior performance across most goal-reaching tasks when compared to the MSE-based dynamics model learning method. Although there are instances where the MSE approach outperforms our continuity-based model method in certain datasets and algorithms, we contend that incorporating local continuity in dynamics learning is valuable and merits further exploration. In summary, the local Lipschitz continuity assumption plays a crucial role in the effective learning of dynamics model.

## Conclusion and Future Work

In this paper, we explored goal data augmentation methods for advanced GCWSL algorithms. While these methods enable GCWSL to exhibit stitching capabilities, existing approaches still struggle to select appropriate augmented goals for GCWSL. To address this issue, we proposed a unified set of goal data augmentation principles specifically tailored for GCWSL methods and introduced a novel model-based approach called MGDA. MGDA leverages the local Lipschitz continuity assumption to quantify model learning errors, facilitating more accurate prediction of augmented goals. Experimental results assessing stitching capabilities indicate that MGDA significantly enhances the performance of GCWSL methods across most offline maze datasets. Compared to other data augmentation techniques, MGDA consistently delivers superior results.

However, this work has two primary limitations. First, our experiments indicate that the effectiveness of MGDA may be influenced by the number of layers in the neural network. Second, in certain offline datasets, MGDA did not consistently outperform existing data augmentation methods. Future research should explore more robust and scalable approaches to further enhance performance.

## Acknowledgments

This work was supported by the STI2030-Major Projects under Grant No. 2022ZD0208800. This research was supported by the Provincial Key Research and Development Program (Project No. 2023-YBGY-033).

## References

Akakzia, A.; Colas, C.; Oudeyer, P.-Y.; Chetouani, M.; and Sigaud, O. 2020. Grounding language to autonomously-acquired skills via goal generation. *arXiv preprint arXiv:2006.07185*.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; and Zaremba, W. 2017. Hindsight experience replay. *Advances in neural information processing systems*, 30.

Bonnard, B.; Caillau, J.-B.; and Trélat, E. 2007. Second order optimality conditions in the smooth case and applications in optimal control. *ESAIM: Control, Optimisation and Calculus of Variations*, 13(2): 207–236.

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

Char, I.; Mehta, V.; Villaflor, A.; Dolan, J. M.; and Schneider, J. 2022. BATS: Best Action Trajectory Stitching. arXiv:2204.12026.

Cheikhi, D.; and Russo, D. 2023. On the statistical benefits of temporal difference learning. In *International Conference on Machine Learning*, 4269–4293. PMLR.

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

Chen, L.; Paleja, R.; and Gombolay, M. 2020. Learning from suboptimal demonstration via self-supervised reward regression. *arXiv preprint arXiv:2010.11723*.

Dargazany, A. 2021. DRL: Deep Reinforcement Learning for Intelligent Robot Control–Concept, Literature, and Future. *arXiv preprint arXiv:2105.13806*.

Ding, Y.; Florensa, C.; Abbeel, P.; and Phielipp, M. 2019. Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32.

Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*.

Eysenbach, B.; Salakhutdinov, R.; and Levine, S. 2020. C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*.

Eysenbach, B.; Zhang, T.; Levine, S.; and Salakhutdinov, R. R. 2022. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 35603–35620.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Ghosh, D.; Gupta, A.; Reddy, A.; Fu, J.; Devin, C. M.; Eysenbach, B.; and Levine, S. 2021. Learning to Reach Goals via Iterated Supervised Learning. In *International Conference on Learning Representations*.

Ghugare, R.; Geist, M.; Berseth, G.; and Eysenbach, B. 2024. Closing the Gap between TD Learning and Supervised Learning–A Generalisation Point of View. *arXiv preprint arXiv:2401.11237*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Hansen, N.; and Wang, X. 2021. Generalization in Reinforcement Learning by Soft Data Augmentation. arXiv:2011.13389.

Hejna, J.; Gao, J.; and Sadigh, D. 2023. Distance Weighted Supervised Learning for Offline Interaction Data. *arXiv preprint arXiv:2304.13774*.

Kaelbling, L. P. 1993. Learning to achieve goals. In *IJCAI*, volume 2, 1094–8. Citeseer.

Kahveci, N. E. 2007. *Robust Adaptive Control For Unmanned Aerial Vehicles*. Ph.D. thesis, University of Southern California.

Kalashnikov, D.; Varley, J.; Chebotar, Y.; Swanson, B.; Jonschkowski, R.; Finn, C.; Levine, S.; and Hausman, K. 2021. MT-Opt: Continuous Multi-Task Robotic Reinforcement Learning at Scale. arXiv:2104.08212.

Ke, L.; Zhang, Y.; Deshpande, A.; Srinivasa, S.; and Gupta, A. 2024. CCIL: Continuity-Based Data Augmentation for Corrective Imitation Learning. In *The Twelfth International Conference on Learning Representations*.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Kostrikov, I.; Yarats, D.; and Fergus, R. 2021. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. arXiv:2004.13649.

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

Li, W.; and Todorov, E. 2004. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *First International Conference on Informatics in Control, Automation and Robotics*, volume 2, 222–229. SciTePress.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, M.; Zhu, M.; and Zhang, W. 2022. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*.

Lu, C.; Huang, B.; Wang, K.; Hernández-Lobato, J. M.; Zhang, K.; and Schölkopf, B. 2020. Sample-Efficient Reinforcement Learning via Counterfactual-Based Data Augmentation. arXiv:2012.09092.

Lynch, C.; Khansari, M.; Xiao, T.; Kumar, V.; Tompson, J.; Levine, S.; and Sermanet, P. 2020. Learning latent plans from play. In *Conference on robot learning*, 1113–1132. PMLR.

Ma, J. Y.; Yan, J.; Jayaraman, D.; and Bastani, O. 2022a. Offline goal-conditioned reinforcement learning via $f$-advantage regression. *Advances in Neural Information Processing Systems*, 35: 310–323.

Ma, Y. J.; Yan, J.; Jayaraman, D.; and Bastani, O. 2022b. How Far I'll Go: Offline Goal-Conditioned Reinforcement Learning via $f$-Advantage Regression. *arXiv preprint arXiv:2206.03023*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Oroojlooyjadid, A.; Nazari, M.; Snyder, L. V.; and Takáč, M. 2022. A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management*, 24(1): 285–304.

Paster, K.; McIlraith, S. A.; and Ba, J. 2020. Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*.

Paster, K.; Pitis, S.; McIlraith, S. A.; and Ba, J. 2023. Return Augmentation gives Supervised RL Temporal Compositionality.

Plasencia-Salgueiro, A. d. J. 2023. Deep Reinforcement Learning for Autonomous Mobile Robot Navigation. In *Artificial Intelligence for Robotics and Autonomous Systems Applications*, 195–237. Springer.

Prudencio, R. F.; Maximo, M. R.; and Colombini, E. L. 2023. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*.

Quiroga, F.; Hermosilla, G.; Farias, G.; Fabregas, E.; and Montenegro, G. 2022. Position control of a mobile robot through deep reinforcement learning. *Applied Sciences*, 12(14): 7194.

Sarangapani, J. 2018. *Neural network control of nonlinear discrete-time systems*. CRC press.

Schaul, T.; Horgan, D.; Gregor, K.; and Silver, D. 2015. Universal value function approximators. In *International conference on machine learning*, 1312–1320. PMLR.

Sestini, A.; Bagdanov, A. D.; et al. 2023. Deep reinforcement learning for the design and validation of modern computer games.

Seto, D.; Annaswamy, A. M.; and Baillieul, J. 1994. Adaptive control of nonlinear systems with a triangular structure. *IEEE Transactions on Automatic Control*, 39(7): 1411–1428.

Sharifani, K.; and Amini, M. 2023. Machine Learning and Deep Learning: A Review of Methods and Applications. *World Information Technology and Engineering Journal*, 10(07): 3897–3904.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6: 1–48.

Sikchi, H.; Chitnis, R.; Touati, A.; Geramifard, A.; Zhang, A.; and Niekum, S. 2024. Score Models for Offline Goal-Conditioned Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Srinivas, A.; Laskin, M.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*.

Srivastava, R. K.; Shyam, P.; Mutz, F.; Jaśkowski, W.; and Schmidhuber, J. 2019. Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*.

Stone, A.; Ramirez, O.; Konolige, K.; and Jonschkowski, R. 2021. The Distracting Control Suite – A Challenging Benchmark for Reinforcement Learning from Pixels. arXiv:2101.02722.

Yamagata, T.; Khalil, A.; and Santos-Rodriguez, R. 2023. Q-learning Decision Transformer: Leveraging Dynamic Programming for Conditional Sequence Modelling in Offline RL.

Yang, C.; Ma, X.; Huang, W.; Sun, F.; Liu, H.; Huang, J.; and Gan, C. 2019. Imitation learning from observations by minimizing inverse dynamics disagreement. *arXiv preprint arXiv:1910.04417*.

Yang, R.; Lu, Y.; Li, W.; Sun, H.; Fang, M.; Du, Y.; Li, X.; Han, L.; and Zhang, C. 2022. Rethinking Goal-conditioned Supervised Learning and Its Connection to Offline RL. *arXiv preprint arXiv:2202.04478*.

Yang, W.; Wang, H.; Cai, D.; Pajarinen, J.; and Kämäräinen, J.-K. 2023. Swapped goal-conditioned offline reinforcement learning. *arXiv preprint arXiv:2302.08865*.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2021. Mastering Visual Continuous Control: Improved Data-Augmented Reinforcement Learning. arXiv:2107.09645.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, 1433–1438. Chicago, IL, USA.