Expert Systems With Applications 311 (2026) 131344

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

CORIMP: A correlation-driven imputation approach for offline reinforcement learning with incomplete action data

Yulin Shao [a,b], Yuanbo Xu [a,b,c,*], Ximing Li [c]

[a] College of Software, Jilin University, Changchun, 130012, China
[b] Mobile Intelligent Computing Lab (MIC Lab), Jilin University, Changchun, 130012, China
[c] College of Computer Science and Technology, Jilin University, Changchun, 130012, China

ARTICLE INFO

Keywords:
Offline reinforcement learning
Data imputation
Dimension-specific missing action data

ABSTRACT

Offline reinforcement learning (RL) is a data-driven paradigm that learns policies from static datasets without real-time interacting with the environment. However, action data collected from real-world are often incomplete due to issues such as sensor failures or communication disruptions, which can significantly impair the performance of offline RL. We focus on the *dimension-specific missing action data problem* (DSMADP) and utilize such expensive yet incomplete action data to enhance offline RL. Inspired by the coordinated nature of joint movements in physical systems, we propose that intrinsic correlations exist across dimensions within each action example—referred to as intra-example inter-dimension correlations. Based on this insight, we propose an effective MLP-based CORrelation-driven IMPutation model named CORIMP. It models the correlations by learning mappings from observed to missing action dimensions, which then guides the imputation of missing values using available data. Theoretically, we bound CORIMP's imputation error and its downstream impact on offline RL performance. Experimental results on variants of missing D4RL datasets demonstrate the effectiveness of our method. Notably, with the TD3BC algorithm, the CORIMP-imputed dataset achieves 95.15% of the Halfcheetah-medium-expert dataset performance (oracle). It provides an average improvement of 99.12% over zero-filled datasets with missing ratios from 0.1 to 0.9 across two dimensions.

1. Introduction

As a trial-and-error paradigm, online reinforcement learning (RL) has flourished over the past few years in various simulated tasks (Feng & Tan, 2016; Mnih et al., 2015; Silver et al., 2016). However, in many real-world applications, deploying online RL is complex, and collecting interactions is often costly (Kim et al., 2022; Kiran et al., 2021; Singh et al., 2022; Wang et al., 2025). Offline RL, as a data-driven paradigm, shines a light on a promising direction for learning policies from static offline datasets without further interacting with the environment (Lange et al., 2012; Levine et al., 2020).

The data-driven nature of offline RL dictates the need for high-fidelity offline datasets. Previous studies have mainly focused on simulated tasks, typically featuring stable environments that lack the widespread interferences common in the real world (Muratore et al., 2019; Niu et al., 2022; Park et al., 2024). Data from real-world interactions are naturally more representative, accurately reflecting complex scenarios (Zheng et al., 2024). Moreover, while acquiring such data is expensive, it is essential due to the unique insights it provides from challenging environments. Nevertheless, these datasets often suffer from the incomplete data problem due to complex factors (Fatyanosa et al., 2024). For instance, deep-sea AUV operations face sensor failures due to extreme pressures and mechanical impacts (Liu et al., 2025). These failures disrupt data collection, leading to incomplete datasets that degrade offline RL performance. In fact, missing data imputation is widely applied in practical engineering contexts like intelligent transportation (Fang et al., 2024; Xing et al., 2023; Zhou et al., 2025) and sensor networks (Fatyanosa et al., 2024; Ma et al., 2024; Xing et al., 2025), providing valuable insights for restoring incomplete RL datasets.

The integrity of real-world datasets is fundamentally constrained by the reliability asymmetry between perception and control subsystems. While state estimation typically relies on robust, high-bandwidth perception streams (e.g., cameras), action data is recorded via separate actuator-side feedback or control uplinks. Unfortunately, these action recording channels are often vulnerable to network instability (e.g., packet loss) or mechanical faults, frequently causing action data to go missing while the environmental observation stream remains intact.

In the domain of offline RL, prior works most closely related to ours (Yang et al., 2024; Zheng et al., 2023) have investigated scenarios involving imperfect action data. Crucially, they treat the action vector as

* Corresponding author.
E-mail addresses: shaoyl23@mails.jlu.edu.cn (Y. Shao), yuanbox@jlu.edu.cn (Y. Xu), liximing86@gmail.com (X. Li).

https://doi.org/10.1016/j.eswa.2026.131344
Received 18 July 2025; Received in revised form 23 January 2026; Accepted 23 January 2026
Available online 25 January 2026

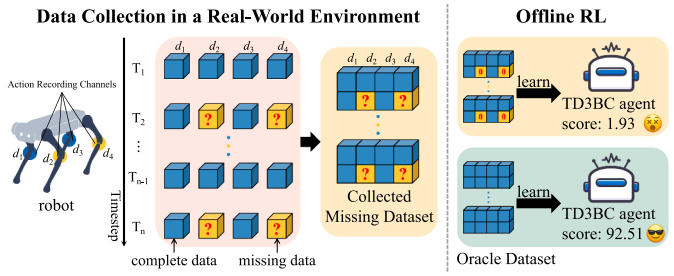


Fig. 1. Left: Illustration of DSMADP. Right: Normalized scores of TD3BC on the dataset with 70% missing data across two dimensions (zero-filled) and on the original D4RL Halfcheetah-medium-expert dataset (oracle). The scores are averaged over the final five evaluations, each with three seeds.

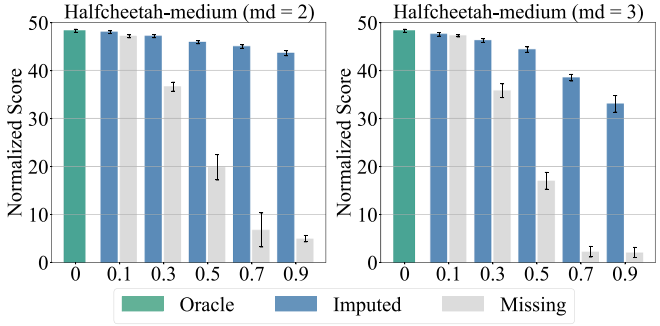


Fig. 2. Performance of TD3BC on the variant D4RL Halfcheetah-medium datasets, with two (left) and three missing dimensions (right), where missing rates vary from 0.1 to 0.9. "Oracle" refers to the performance on the original D4RL `Halfcheetah-medium` dataset; "Missing" refers to the performance on datasets with missing values filled with zeros; and "Imputed" refers to the performance on those missing datasets after applying CORIMP.

an atomic unit, relying on a coarse-grained assumption that anomalies uniformly affect all dimensions. However, this assumption overlooks the independent failure modes inherent in heterogeneous action recording channels. To bridge the gap between idealized assumptions and real-world failures, we delve into the granularity of individual dimensions. We specifically address the practical issue where specific action recording channels are disrupted, leading to incomplete action dimensions in specific samples, as illustrated in Fig. 1. We formally define this issue as the **Dimension-Specific Missing Action Data Problem (DSMADP)**.

To our knowledge, we are the first to investigate the incomplete data problem at the dimension level in offline RL. We argue that these expensive yet incomplete data obtained from real-world environments are irreplaceable, as they encapsulate complex environmental dynamics. Simply discarding such partially corrupted samples is wasteful, as they retain decision-critical information essential for enhancing model performance. Our insight lies in fully harnessing the intrinsic correlations within the data to recover these missing dimensions, thereby laying a solid foundation for robust offline policy learning.

In this work, we focus on DSMADP. Initially, we investigate how such incomplete data impacts the performance of offline RL. Specifically, we examine the situations where missing action data occurs respectively across two and three dimensions of the `Halfcheetah-medium` dataset, with missing rates ranging from 0.1 to 0.9. As illustrated in Fig. 2, it can be observed that: (1) given a constant number of missing dimensions, performance significantly declines as the missing rate increases; (2) for a fixed missing rate, a more significant number of missing dimensions is associated with poorer performance. To alleviate the impact of such incomplete data, we propose a simple yet effective MLP-based correlation-driven imputation model named CORIMP. It draws inspiration from the observation of an intriguing real-world phenomenon: there are complex interactions between the joints of moving objects, and they need to maintain specific relationships to stay balanced and achieve desired goals. Based on this insight, for a given action example, CORIMP leverages available data from non-missing dimensions to impute values for missing dimensions based on captured inter-dimension correlations. Extensive experiments demonstrate that CORIMP competently addresses various incomplete data scenarios, effectively alleviating the impact of incomplete data on offline RL and achieving performance levels on par with or surpassing those from oracle datasets.

The contributions of this work are summarized as follows:

- We are the first to formulate and address the dimension-specific missing action data problem (DSMADP) in offline RL.
- We systematically analyze the sensitivity patterns of different tasks and types of datasets when exposed to DSMADP.
- We propose CORIMP, a simple yet effective MLP-based correlation-driven imputation model that imputes missing action dimensions by learning their correlation with observed dimensions.
- We theoretically bound CORIMP's imputation error and its downstream impact on offline RL performance.
- We demonstrate the effectiveness of CORIMP through extensive experiments on variants of D4RL datasets.

2. Related works

2.1. Offline reinforcement learning

Offline RL aims to learn policies from pre-collected datasets without interacting with the environment (Lange et al., 2012; Levine et al., 2020). Existing work on offline RL can be generally categorized into model-based approaches (Chen et al., 2021; Janner et al., 2021; Kidambi et al., 2021, 2020; Li et al., 2025; Rigter et al., 2022; Sun et al., 2023; Uehara & Sun, 2021; Yu et al., 2021, 2020; Zhang et al., 2023; Zhu et al., 2025) and model-free approaches (An et al., 2021; Bai et al., 2022; Cheng et al., 2022; Fujimoto & Gu, 2021; Fujimoto et al., 2019; Jin et al., 2021; Kostrikov et al., 2021; Kumar et al., 2020; Nair et al., 2020; Xie et al., 2021; Zhou et al., 2021) approaches. The precondition for using them is that the dataset labels must be complete.

2.2. Imitation learning from observations

The problem setting of imitation learning (IL) from observations (ILFO) is that we do not have access to the full set of actions. FAIL (Sun et al., 2019) learns a sequence of time-dependent policies by minimizing an integral probability metric between the observation distributions of the expert policy and the learner. LAPO (Schmidt & Jiang, 2024) trains an inverse dynamics model (IDM) and a forward dynamics model (FDM) jointly to recover latent action information. Distinct from our work, IL-related works typically assume that trajectories are generated by expert policies, which isn't applicable in conventional offline RL. Moreover, it is essential to note that we are not concerned with entire action labels being missing, but rather with specific dimensions of the action labels having missing data.

2.3. Data imputation

Data imputation (Schafer & Graham, 2002) addresses missing data to complete datasets for downstream tasks. Traditional methods like mean imputation (Rubin, 1976) and regression imputation (Troyanskaya et al., 2001) may fail to capture complex data relationships. Machine learning techniques, such as k-nearest neighbors (k-NN) (Cover & Hart, 1967) and decision trees (Quinlan, 1986), enhance performance but can be computationally intensive and struggle with high-dimensional data. Deep learning approaches, including autoencoders (Hinton & Salakhutdinov, 2006) and GANs (Goodfellow et al., 2014), offer improved handling of high-dimensional data at the cost of increased computational re-

sources. Neural network-based regression models (Bishop, 1995; Rumelhart et al., 1986) effectively capture intricate data dependencies. In the relevant research, different approaches have been proposed in various application domains. For maritime sensor data, Fatyanosa et al. (2024) proposed a meta-learning framework that automatically recommends optimal imputation methods by extracting dataset characteristics. For traffic data, Fang et al. (2024) introduced a Multi-domain Generative Adversarial Transfer Learning Network (MDTGAN), which leverages transferable spatiotemporal patterns across urban networks to impute missing values under constrained local samples. Moreover, researchers have employed diverse strategies to address data sparsity such as representation learning (Xu et al., 2021) and generative models (Wang et al., 2022) for matrix-based imputation, and attention mechanisms for predicting missing points in sequential trajectories (Jiang et al., 2023). Despite substantial research, handling incomplete data in offline RL remains relatively unexplored. We introduce CORIMP, a novel MLP-based correlation-driven imputation model specifically designed to address the impact of DSMADP on offline RL.

3. Preliminaries

3.1. Dimension-specific missing action data

In the offline RL setting, we are provided with a fixed dataset $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_{i+1})\}_{i=1}^{N}$ collected by an unknown behavior policy π_β, where i indexes a transition (sample) in the dataset and N is the dataset size. Here, $\mathbf{s}_i$ is the state, $\mathbf{a}_i$ is the action, r_i is the reward, and $\mathbf{s}_{i+1}$ is the next state at each index i. The agent can only learn a policy $\pi(\mathbf{a}|\mathbf{s})$ from this dataset without further interaction. The dataset's quality plays a crucial role in ensuring the performance of the learned policy.

We consider datasets where action data is partially missing across certain dimensions. Specifically, such a dataset is defined as:

$$\mathcal{D}_{\text{missing}} = \{(\mathbf{s}_i, \tilde{\mathbf{a}}_i, r_i, \mathbf{s}_{i+1})\}_{i=1}^{N}, \tag{1}$$

where $\tilde{\mathbf{a}}_i$ represents the action data at index i, with potential missing values in some dimensions.

Assume the action $\tilde{\mathbf{a}}_i$ is a vector in $\mathbb{R}^C$, where C denotes the number of dimensions in the action space. Then the action data at index i can be denoted as:

$$\tilde{\mathbf{a}}_i = (\tilde{a}_{i1}, \tilde{a}_{i2}, \dots, \tilde{a}_{iC}), \tag{2}$$

where each $\tilde{a}_{ij}$ corresponds to the jth dimension of the action $\tilde{\mathbf{a}}_i$ at index i, for $j \in \{1, 2, \dots, C\}$.

To distinguish between missing and available values, we use an indicator variable m_{ij}. Specifically, $m_{ij} = 0$ indicates that the entry $\tilde{a}_{ij}$ is missing, and $m_{ij} = 1$ indicates that the entry is available and valid.

Let $\mathcal{M}_i \subseteq \{1, 2, \dots, C\}$ be the set of indices corresponding to the missing dimensions in the action data at sample i. We denote that for each sample i, there are $|\mathcal{M}_i|$ missing dimensions.

3.2. Problem statement

Given a dataset $\mathcal{D}_{\text{missing}}$ with incomplete action data across specific dimensions in various samples, which affects the fidelity of the dataset and consequently impairs the performance of offline RL. The goal is to develop an imputation method to estimate the missing entries in $\mathcal{D}_{\text{missing}}$, producing an imputed dataset $\mathcal{D}_{\text{imputed}}$ that closely approximates the quality of the oracle dataset $\mathcal{D}$. By addressing the incomplete data, it is expected to reduce the overall performance degradation of $\mathcal{D}_{\text{imputed}}$ relative to $\mathcal{D}$ in offline RL tasks.

4. Methodology

In this section, we begin by introducing a practical training pipeline tailored for DSMADP. Next, we offer an in-depth look at CORIMP, which consists of three parts. Firstly, we detail its crucial preliminary step: missing data identification. Secondly, we describe its architecture. Finally, we elaborate on the implementation details.

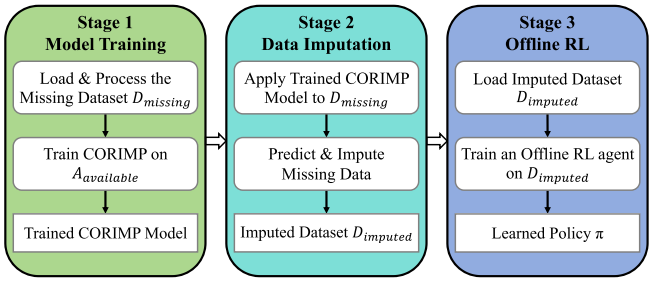


Fig. 3. Our proposed training pipeline.

4.1. Training pipeline

In conventional offline RL, datasets are typically assumed to be reliable and immediately usable for training. However, datasets with missing data require preprocessing to ensure their suitability for subsequent applications. Based on this insight, we propose a practical training pipeline to address DSMADP. This pipeline ensures dataset completeness and, consequently, supports effective learning in downstream tasks. As shown in Fig. 3, the pipeline comprises three stages: model training, data imputation, and offline RL.

- **Model Training:** This stage trains CORIMP using only the complete action samples, denoted as $\mathbf{A}_{\text{available}}$. Specifically, we first randomly split $\mathbf{A}_{\text{available}}$ into training (70%), validation (15%), and testing (15%) subsets. Then, to construct the training pairs, we extract values from specific dimensions of these samples: the values corresponding to the observed dimensions ($C_{\text{available}}$) serve as the input vector $\mathbf{x}$, while the values at the missing dimensions (C_{missing}) are used as the regression targets $\mathbf{y}$. The model learns to map $\mathbf{x}$ to $\mathbf{y}$ by minimizing prediction error, thereby effectively capturing the intra-example inter-dimension correlations.
- **Data Imputation:** In the second stage, the trained CORIMP model is applied to the dataset $\mathcal{D}_{\text{missing}}$. The model estimates and imputes the missing entries, producing a complete dataset $\mathcal{D}_{\text{imputed}}$.
- **Offline RL:** In the final stage, the imputed dataset $\mathcal{D}_{\text{imputed}}$ is used to train an offline RL agent. This allows the agent to benefit from comprehensive and accurate information, which is crucial for developing effective policies and enhancing decision-making capabilities in complex environments.

4.2. Missing data identification

We represent the action data from $\mathcal{D}_{\text{missing}}$ as a matrix $\mathbf{A} \in \mathbb{R}^{N \times C}$. Each row of this matrix corresponds to an action vector $\tilde{\mathbf{a}}_i$ from $\mathcal{D}_{\text{missing}}$, and each entry $\tilde{a}_{ij}$ in $\mathbf{A}$ corresponds to the value of the ith sample in the jth dimension.

$$\mathbf{A} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1C} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{N1} & \tilde{a}_{N2} & \cdots & \tilde{a}_{NC} \end{bmatrix}. \tag{3}$$

To identify missing data, we employ a binary mask matrix $\mathbf{M} \in \{0, 1\}^{N \times C}$, where $m_{ij} = 0$ indicates that $\tilde{a}_{ij}$ is missing, and $m_{ij} = 1$ signifies that $\tilde{a}_{ij}$ is available:

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1C} \\ m_{21} & m_{22} & \cdots & m_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N2} & \cdots & m_{NC} \end{bmatrix}. \tag{4}$$

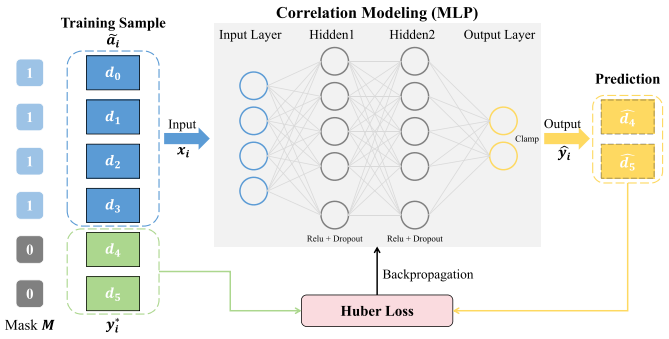


Fig. 4. Training-time architecture of CORIMP.

Based on this mask, we define $\mathbf{A}_{\text{available}}$ as the set of complete action samples (i.e., rows from $\mathbf{A}$ with no missing values). Only these samples are used for training CORIMP. Specifically, $\mathbf{A}_{\text{available}}$ can be expressed as:

$$\mathbf{A}_{\text{available}} = \{\mathbf{A}_i \mid \forall j \in \{1, \dots, C\}, m_{ij} = 1\}, \tag{5}$$

where $\mathbf{A}_i$ represents the ith sample of the matrix $\mathbf{A}$ (i.e., the action vector $\tilde{\mathbf{a}}_i$), which contains no missing values in any of its dimensions.

Moreover, we use $\mathbf{M}$ to classify dimensions as missing or available. A dimension j is missing if there is at least one sample i with $m_{ij} = 0$, and available if $m_{ij} = 1$ for all samples i. Formally, the sets are defined as:

$$C_{\text{missing}} = \{j \in \{1, \dots, C\} \mid \exists i \in \{1, \dots, N\} \text{ such that } m_{ij} = 0\} \tag{6}$$

$$C_{\text{available}} = \{j \in \{1, \dots, C\} \mid \forall i \in \{1, \dots, N\}, m_{ij} = 1\} \tag{7}$$

The set $C_{\text{available}}$ provides the input features for CORIMP, while C_{missing} comprises the target dimensions for imputation.

4.3. Model architecture

CORIMP is designed to capture and harness the correlations between dimensions in the action data by drawing inspiration from the correlated nature between joints in moving objects. The training-time architecture of CORIMP is illustrated in Fig. 4. For any given action sample, it uses data from available dimensions to impute missing values based on the captured inter-dimension correlations effectively. At inference time, CORIMP performs a single forward pass to impute the missing action dimensions using the trained network.

- **Input Layer:** Accepts an input vector $\mathbf{x}_i \in \mathbb{R}^{|C_{\text{available}}|}$, representing the observed dimensions (defined by $C_{\text{available}}$ in Eq. (7)) of an action sample $\tilde{\mathbf{a}}_i$.
- **Hidden Layers:** Two hidden layers, each with d_h units, transform the input $\mathbf{x}_i$ to produce activations $\mathbf{h}_{i,1}, \mathbf{h}_{i,2} \in \mathbb{R}^{d_h}$:

$$\mathbf{h}_{i,1} = \text{Dropout}(\text{ReLU}(W_1 \mathbf{x}_i + \mathbf{b}_1)) \tag{8}$$

$$\mathbf{h}_{i,2} = \text{Dropout}(\text{ReLU}(W_2 \mathbf{h}_{i,1} + \mathbf{b}_2)) \tag{9}$$

  where W_1, W_2 are weight matrices and $\mathbf{b}_1, \mathbf{b}_2$ are bias vectors. ReLU serves as the activation function, and Dropout is applied for regularization.
- **Output Layer:** Produces the predicted vector $\hat{\mathbf{y}}_i \in \mathbb{R}^{|C_{\text{missing}}|}$ for the target dimensions (defined by C_{missing} in Eq. (6)):

$$\hat{\mathbf{y}}_i = \text{Clamp}(W_3 \mathbf{h}_{i,2} + \mathbf{b}_3) \tag{10}$$

  where W_3 and b_3 are the output layer's weight matrix and bias vector, respectively, the Clamp$(\cdot)$ function ensures that output values are within a valid range.

This architecture effectively models the inter-dimension correlations, leading to accurate intra-example imputation and improved performance in downstream RL tasks.

4.4. Implementation details

The dataset is stored in HDF5 format and includes actions with missing values denoted by zeros. The preprocessing steps involve loading the dataset, classifying dimensions as missing or available (Eqs. (6) and 7), and filtering out complete samples (Eq. (5)). Only the complete samples are used for model training. For training preparation, the complete samples are split into training, test, and validation sets with ratios of 70%, 15%, and 15%, respectively. Data loaders with a batch size of 1024 are employed to feed data into the model during training efficiently.

CORIMP is trained using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1×10^{-4} and a batch size of 1024. The objective is to minimize the Huber loss with a threshold $\delta = 1.0$. Regarding the network architecture, we set the hidden layer size $d_h = 1024$ (see Eq. (8)) and apply a dropout rate of 0.1. To ensure optimal performance without overfitting, we employ early stopping with a patience of 35 epochs (halting training if the validation loss does not improve for 35 consecutive epochs).

For each training sample $\tilde{\mathbf{a}}_k \in \mathbf{A}_{\text{available}}$ and for each target dimension $j \in C_{\text{missing}}$ (Eq. (6)), let $\tilde{a}_{kj}$ be the ground truth value from $\tilde{\mathbf{a}}_k$, and $\hat{a}_{kj}$ be CORIMP's corresponding prediction (i.e., the component of $\hat{\mathbf{y}}_k = \hat{f}(\mathbf{x}_k)$ associated with dimension j). The component-wise Huber loss is:

$$\mathcal{L}_\delta(\tilde{a}_{kj}, \hat{a}_{kj}) = \begin{cases} \frac{1}{2}(\tilde{a}_{kj} - \hat{a}_{kj})^2 & \text{if } |\tilde{a}_{kj} - \hat{a}_{kj}| \le \delta \\ \delta|\tilde{a}_{kj} - \hat{a}_{kj}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}. \tag{11}$$

5. Theoretical analysis

This section analyzes the theoretical properties of CORIMP. We model the underlying data generation process and bound the imputation error, subsequently quantifying its impact on the downstream offline RL value estimation.

5.1. Assumptions and error decomposition

For any ground-truth action sample, denoted as $\mathbf{a}_i^*$, we decompose it into observed features $\mathbf{x}_i$ (values at dimensions $C_{\text{available}}$) and missing targets $\mathbf{y}_i$ (values at dimensions C_{missing}). Let $\hat{f}$ be the learned imputation model. We define the predicted missing values as $\hat{\mathbf{y}}_i = \hat{f}(\mathbf{x}_i)$. Consequently, the reconstructed (imputed) action $\hat{\mathbf{a}}_i$ is formed by combining the observed $\mathbf{x}_i$ and the predicted $\hat{\mathbf{y}}_i$.

Assumption 1 (Ground-truth correlation and noise)**.** We assume the missing values are governed by an underlying physical function f^*: $\mathbf{y}_i = f^*(\mathbf{x}_i) + \boldsymbol{\epsilon}_i$. Here, f^* represents the intrinsic system dynamics (e.g., non-linear kinematic constraints), and $\boldsymbol{\epsilon}_i$ represents irreducible aleatoric noise satisfying $\mathbb{E}[\boldsymbol{\epsilon}_i] = \mathbf{0}$ and $\mathbb{E}[\|\boldsymbol{\epsilon}_i\|_2^2] \le \sigma_{\text{noise}}^2$.

Assumption 2 (Bounded approximation error)**.** We assume the expected squared difference between the learned model $\hat{f}$ and the ground truth f^* is bounded: $\mathbb{E}_{\mathbf{x}}[\|\hat{f}(\mathbf{x}) - f^*(\mathbf{x})\|_2^2] \le \epsilon_{\hat{f}}^2$. This term represents the epistemic uncertainty (approximation and estimation error) minimized during training.

Theorem 1 (Imputation error bound)**.** *Under Assumptions 1 and 2, the expected squared L_2 error of the imputed action $\hat{\mathbf{a}}_i$ with respect to the ground truth $\mathbf{a}_i^*$ is bounded by:*

$$\mathbb{E}[\|\hat{\mathbf{a}}_i - \mathbf{a}_i^*\|_2^2] \le \epsilon_{\hat{f}}^2 + \sigma_{noise}^2 =: \epsilon_{CORIMP}^2 \tag{12}$$

Proof. Since the imputed action $\hat{\mathbf{a}}_i$ uses the true values $\mathbf{x}_i$ for the observed dimensions, the error is non-zero only in the missing dimensions. Thus, $\|\hat{\mathbf{a}}_i - \mathbf{a}_i^*\|_2^2 = \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2$. Substitute $\hat{\mathbf{y}}_i = \hat{f}(\mathbf{x}_i)$ and $\mathbf{y}_i = f^*(\mathbf{x}_i) + \boldsymbol{\epsilon}_i$, the error vector becomes $(\hat{f}(\mathbf{x}_i) - f^*(\mathbf{x}_i)) - \boldsymbol{\epsilon}_i$. Expanding the squared norm yields $\|\hat{f} - f^*\|_2^2 + \|\boldsymbol{\epsilon}\|_2^2 - 2(\hat{f} - f^*)^T\boldsymbol{\epsilon}$. Since the intrinsic noise $\boldsymbol{\epsilon}$ is zero-mean and independent of the model bias, the expectation of the cross-term vanishes. The result follows by summing the expected squared norms. □

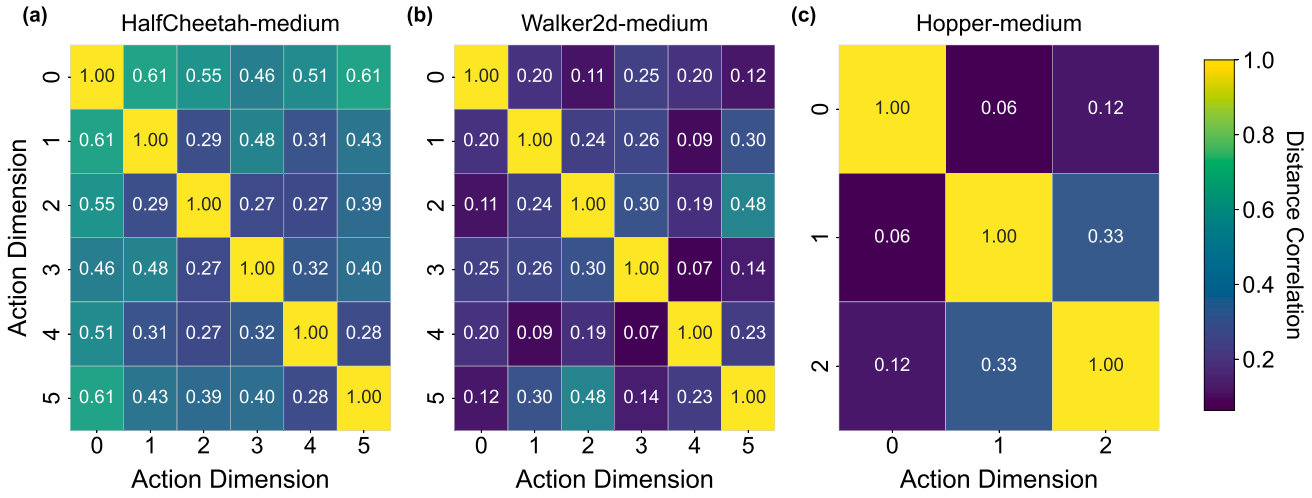


Fig. 5. Distance correlation matrices for action dimensions in three datasets: (a) Halfcheetah-medium, (b) Walker2d-medium, and (c) Hopper-medium. Each heatmap visualizes the pairwise distance correlation coefficients between action dimensions (numbered 0 to 5 for Halfcheetah and Walker2d, and 0 to 2 for Hopper), sampled at 0.1% of the original data. Warmer colors (e.g., yellow, green) indicate stronger correlations, while cooler colors (e.g., blue, purple) represent weaker correlations.

Remark 1 (Theoretical Advantage over Linear Baselines)**.** This decomposition explicitly highlights why CORIMP outperforms traditional methods. Methods like **Mean Imputation** (assuming constant f^*) or **Linear Regression** (assuming linear f^*) suffer from a significant *structural mismatch* when modeling complex robotic dynamics, leading to a high approximation error $\epsilon^2_{\hat{f}}$. In contrast, CORIMP leverages the universal approximation capability of MLPs to fit the non-linear manifold of f^*, theoretically minimizing $\epsilon^2_{\hat{f}}$ towards the irreducible noise limit.

5.2. Downstream impact on offline RL

We now link the imputation quality to the reliability of the Q-value estimation, which is critical for offline RL performance.

Assumption 3 (Lipschitz continuity)**.** The learned Q-function $Q(\mathbf{s}, \mathbf{a})$ is L_Q-Lipschitz continuous with respect to actions, such that $|Q(\mathbf{s}, \mathbf{a}_1) - Q(\mathbf{s}, \mathbf{a}_2)| \leq L_Q \|\mathbf{a}_1 - \mathbf{a}_2\|_2$.

Property 1 (Q-Value Perturbation Bound)**.** *Given Assumption 3 and Theorem 1, the expected error in Q-value estimation caused by imputation is bounded by:*

$$\mathbb{E}[|Q(\mathbf{s}_i, \hat{\mathbf{a}}_i) - Q(\mathbf{s}_i, \mathbf{a}_i^*)|] \leq L_Q \cdot \epsilon_{CORIMP} \tag{13}$$

Proof. By the Lipschitz condition, $|Q(\mathbf{s}_i, \hat{\mathbf{a}}_i) - Q(\mathbf{s}_i, \mathbf{a}_i^*)| \leq L_Q \|\hat{\mathbf{a}}_i - \mathbf{a}_i^*\|_2$. Taking the expectation and applying Jensen's inequality ($\mathbb{E}[X] \leq \sqrt{\mathbb{E}[X^2]}$ for non-negative variable X), we have $\mathbb{E}[\|\hat{\mathbf{a}}_i - \mathbf{a}_i^*\|_2] \leq \sqrt{\mathbb{E}[\|\hat{\mathbf{a}}_i - \mathbf{a}_i^*\|_2^2]} \leq \epsilon_{\text{CORIMP}}$. Multiplying by L_Q completes the proof. □

Insight. The Lipschitz constant L_Q acts as an amplifier. Since Q-functions in offline RL can be sharp near the data manifold (to penalize out-of-distribution actions), minimizing ϵ_{CORIMP} is mathematically essential. Property 1 confirms that CORIMP's ability to capture non-linear correlations (Remark 1) directly translates to a tighter theoretical bound on value estimation error compared to linear or constant baselines.

6. Experiments

6.1. Experimental setup

Missing Data Setup. Two key parameters, missing rate $p \in [0, 1]$, measuring the proportion of samples with missing data, and the number of dimensions with missing data $c \in \{0, 1, \ldots, C\}$, where C is the total number of dimensions in the action dataset. In our generation protocol, the subset of c dimensions is selected once and remains fixed across the entire dataset. For each sample, values in these dimensions are then masked with probability p.

Datasets. We evaluate our approach on three D4RL (Fu et al., 2020) MuJoCo locomotion tasks: `Halfcheetah`, `Walker2d`, and `Hopper`. The dataset types used for each task are `medium` and `medium-expert`.

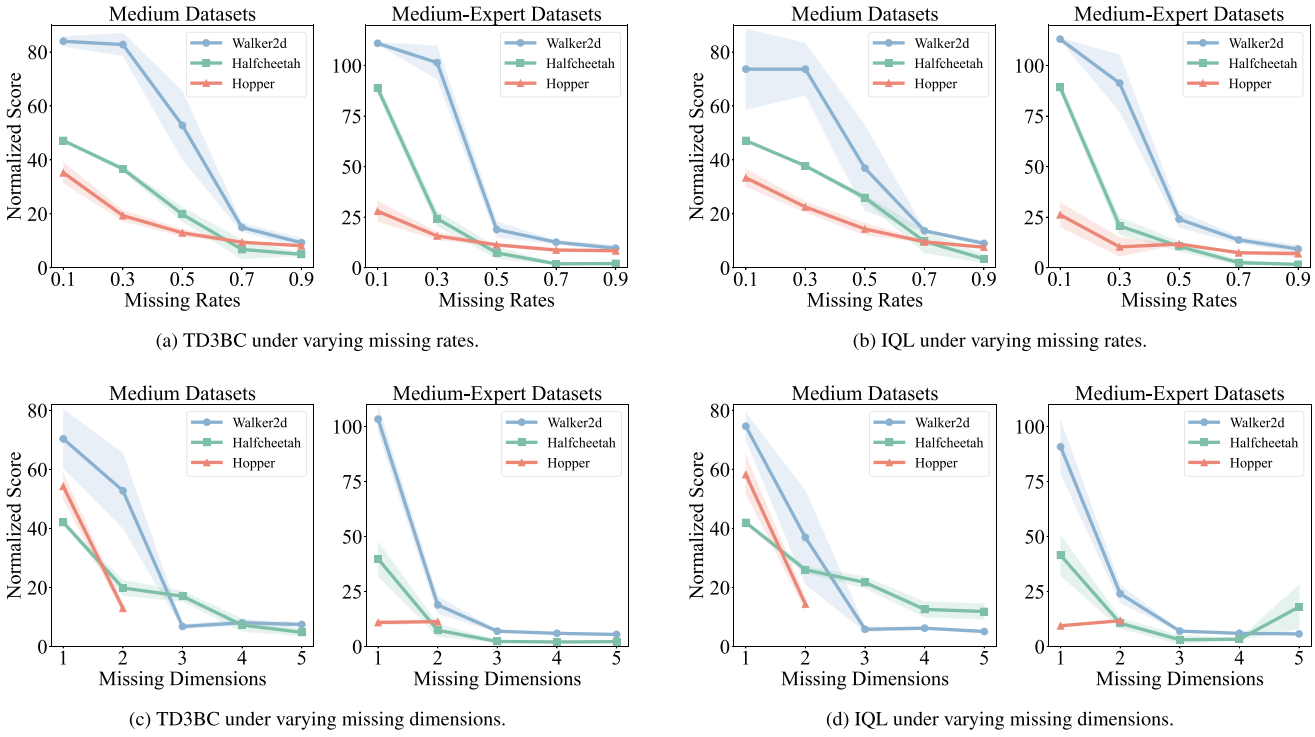
The `medium` dataset is generated by first training a policy online using Soft Actor-Critic (Haarnoja et al., 2018), early-stopping the training, and collecting 1M samples from this partially-trained policy. The `medium-expert` dataset combines equal amounts of expert demonstrations and suboptimal data generated via a partially trained policy or by unrolling a uniform-at-random policy. The `medium` dataset represents a more realistic scenario where the policy is only partially trained, reflecting typical conditions in real-world applications. The `medium-expert` dataset provides a diverse mix of high-quality expert demonstrations and suboptimal samples, enabling a more thorough assessment of our method's robustness and adaptability across varying data types.

Imputation Baselines and Offline RL Algorithms. To substantiate the effectiveness of our approach, CORIMP, we compare it against the following baselines:

- **Zero Filling (Zero):** Missing entries are replaced with zeros.
- **Mean Imputation (Mean):** Missing entries are replaced with the mean value of the corresponding dimension calculated from the training dataset.
- **Linear Regression (LR):** A parametric baseline implemented via Multivariate Imputation by Chained Equations (MICE). It iteratively models each missing feature as a linear function of others over 10 iterations to capture linear dependencies.
- **k-Nearest Neighbors (KNN):** A non-parametric baseline that imputes missing values using the mean of the k = 5 nearest neighbors found in the observed subspace, leveraging local similarity patterns.

We evaluate all imputation methods by training two offline RL algorithms, TD3BC (Fujimoto & Gu, 2021) and IQL (Kostrikov et al., 2021), on the corresponding imputed datasets.

Evaluation Metric. TD3BC and IQL are trained on each dataset for 1 million time steps and evaluated every 5000 time steps, each consisting of 10 episodes. We report the normalized score, calculated as $\frac{100\times(\text{score}-\text{random score})}{\text{expert score}-\text{random score}}$. All reported scores are averaged over the final five evaluations, each with three seeds. For clarity in all tables, "m" signifies "medium", and "me" signifies "medium-expert".



(a) TD3BC under varying missing rates.

(b) IQL under varying missing rates.

(c) TD3BC under varying missing dimensions.

(d) IQL under varying missing dimensions.

Fig. 6. Performance of TD3BC and IQL under (a)(b) different missing rates and (c)(d) different missing dimensions across various datasets and task domains.

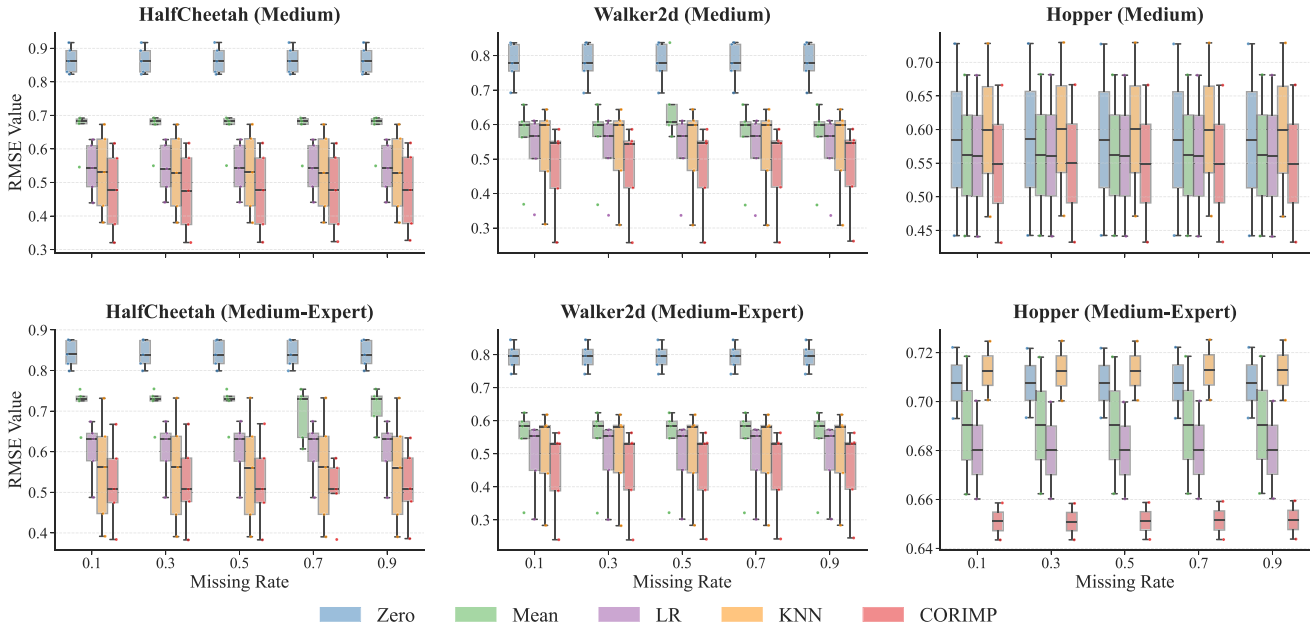


Fig. 7. Boxplot analysis of Imputation RMSE across 120 dataset variants. The evaluation covers three environments (`HalfCheetah`, `Walker2d`, `Hopper`) with two dataset qualities (`medium`, `medium-expert`). The x-axis represents the missing rate ranging from 0.1 to 0.9. CORIMP (pink) consistently demonstrates lower median error and lower variance compared to linear (LR) and non-parametric (KNN) baselines, particularly in high-dimensional tasks.

6.2. *Intra-example inter-dimension correlations*

To substantiate our core hypothesis that inherent correlations exist among the dimensions within individual action samples, referred to as *intra-example inter-dimension correlations*, we conducted a distance correlation analysis (Székely et al., 2007) on action data from representative Mujoco environments in the D4RL dataset (Halfcheetah-medium, Walker2d-medium, and Hopper-medium). This analysis aims to uncover synergistic patterns in the torques applied to different body parts, such as various joints. Distance correlation coefficients between these joint torque dimensions were computed from a 0.1% data subsample. Fig. 5 visualizes these coefficients as heatmaps, depicting the association strength between action dimensions across these environments.

For the **Halfcheetah-medium** dataset (Fig. 5(a)), the six-dimensional action space represents torques for the back thigh, back shin, back foot, front thigh, front shin, and front foot. Significant

Table 1
Results with varying missing rates while keeping the missing dimensions fixed at two. *Oracle*, *Zero*, *Mean*, *LR*, *KNN*, and *CORIMP* refer to TD3BC performance on the original datasets, datasets with missing actions filled with zeros, mean-imputed datasets, LR-imputed datasets, KNN-imputed datasets, and CORIMP-imputed datasets, respectively. *Rel. (%)* indicates the performance of TD3BC on the CORIMP-imputed datasets as a percentage of the original datasets. Best results among imputation methods are highlighted in bold..

		Oracle	Zero	Mean	LR	KNN	CORIMP	Rel. (%)
Halfcheetah-m	mr = 0.1	48.30 ± 0.32	47.16 ± 0.30	46.82 ± 0.32	47.69 ± 0.33	47.88 ± 0.28	**48.01 ± 0.28**	99.40%
	mr = 0.3		36.65 ± 0.93	39.95 ± 0.77	45.13 ± 0.47	46.71 ± 0.30	**47.19 ± 0.31**	97.70%
	mr = 0.5		19.84 ± 2.65	33.97 ± 0.43	42.27 ± 0.42	45.34 ± 0.39	**45.92 ± 0.30**	95.07%
	mr = 0.7		6.78 ± 3.55	27.71 ± 1.13	40.93 ± 0.30	43.89 ± 0.18	**45.02 ± 0.36**	93.21%
	mr = 0.9		4.97 ± 0.51	20.35 ± 1.23	40.25 ± 0.40	42.66 ± 0.20	**43.64 ± 0.66**	90.35%
Average		48.30	23.08	33.76	43.26	45.30	**45.96**	95.15%
Walker2d-m	mr = 0.1	83.16 ± 3.94	83.98 ± 1.86	84.87 ± 2.57	84.88 ± 1.15	82.95 ± 1.68	**85.11 ± 1.62**	102.34%
	mr = 0.3		82.77 ± 4.21	48.20 ± 18.30	**86.72 ± 2.93**	84.85 ± 1.58	85.40 ± 2.78	102.69%
	mr = 0.5		52.78 ± 12.77	16.96 ± 2.05	68.25 ± 17.24	**85.47 ± 5.27**	84.74 ± 4.34	101.90%
	mr = 0.7		14.95 ± 1.69	11.84 ± 1.83	21.17 ± 1.46	63.72 ± 13.49	**69.68 ± 14.05**	83.79%
	mr = 0.9		9.27 ± 1.62	11.82 ± 1.48	15.71 ± 0.69	18.33 ± 4.03	**47.82 ± 12.92**	57.50%
Average		83.16	48.75	34.74	55.35	67.06	**74.55**	89.65%
Hopper-m	mr = 0.1	57.20 ± 5.56	35.23 ± 3.83	36.27 ± 3.15	38.08 ± 6.10	36.13 ± 4.34	**38.92 ± 4.40**	68.04%
	mr = 0.3		19.34 ± 2.00	20.94 ± 1.70	19.58 ± 0.22	20.05 ± 2.28	**22.01 ± 1.08**	38.48%
	mr = 0.5		12.93 ± 1.33	14.96 ± 0.93	14.38 ± 1.58	13.64 ± 2.26	**15.23 ± 1.10**	26.63%
	mr = 0.7		9.45 ± 0.44	9.36 ± 0.65	**10.89 ± 0.42**	10.39 ± 0.35	9.71 ± 1.06	16.98%
	mr = 0.9		8.14 ± 0.28	7.48 ± 0.15	7.49 ± 0.07	4.03 ± 3.59	**15.11 ± 12.14**	26.42%
Average		57.20	17.02	17.80	18.09	16.85	**20.20**	35.31%
Halfcheetah-me	mr = 0.1	92.51 ± 3.23	88.75 ± 2.74	90.33 ± 2.19	91.73 ± 3.25	91.25 ± 4.51	**92.42 ± 3.11**	99.90%
	mr = 0.3		24.27 ± 3.71	35.90 ± 2.41	65.20 ± 4.62	89.12 ± 1.17	**91.76 ± 1.69**	99.19%
	mr = 0.5		7.36 ± 2.86	20.42 ± 2.64	37.23 ± 4.68	82.15 ± 2.22	**84.85 ± 4.81**	91.72%
	mr = 0.7		1.93 ± 0.42	19.45 ± 2.16	36.90 ± 4.29	71.35 ± 6.57	**80.93 ± 3.90**	87.48%
	mr = 0.9		2.08 ± 0.67	11.00 ± 2.22	29.79 ± 3.49	62.67 ± 5.82	**72.88 ± 4.90**	78.78%
Average		92.51	24.88	35.42	52.17	79.31	**84.57**	91.41%
Walker2d-me	mr = 0.1	110.31 ± 0.53	110.54 ± 0.56	110.39 ± 0.58	110.39 ± 0.49	110.37 ± 0.57	**110.56 ± 0.49**	100.23%
	mr = 0.3		101.43 ± 8.21	78.72 ± 33.95	108.75 ± 2.32	107.14 ± 2.68	**111.02 ± 1.33**	100.64%
	mr = 0.5		18.90 ± 3.88	13.29 ± 0.85	34.90 ± 12.79	74.17 ± 27.87	**94.95 ± 13.60**	86.08%
	mr = 0.7		12.62 ± 0.91	10.80 ± 0.34	14.75 ± 1.04	15.10 ± 1.53	**20.87 ± 11.14**	18.92%
	mr = 0.9		9.65 ± 1.65	10.57 ± 0.26	11.73 ± 0.57	11.54 ± 0.87	**11.86 ± 0.61**	10.75%
Average		110.31	50.63	44.75	56.10	63.66	**69.85**	63.32%
Hopper-me	mr = 0.1	98.94 ± 11.93	27.95 ± 5.05	29.43 ± 7.22	34.07 ± 6.22	30.16 ± 4.49	**36.62 ± 10.15**	37.01%
	mr = 0.3		15.71 ± 1.92	17.98 ± 2.60	15.23 ± 1.81	15.92 ± 3.48	**17.50 ± 2.12**	17.69%
	mr = 0.5		11.31 ± 0.73	11.63 ± 0.71	11.20 ± 0.29	10.51 ± 1.65	**11.90 ± 0.94**	12.03%
	mr = 0.7		8.73 ± 0.86	12.49 ± 0.87	**11.70 ± 0.71**	10.56 ± 0.76	8.95 ± 0.78	9.05%
	mr = 0.9		8.40 ± 0.28	7.06 ± 4.89	2.49 ± 0.71	6.47 ± 5.49	**10.80 ± 8.71**	10.92%
Average		98.94	14.42	15.72	14.94	14.72	**17.15**	17.34%

inter-dimensional correlations are observed. For instance, torque on the back thigh (dimension 0) exhibits correlation coefficients of 0.61, 0.55, and 0.61 with torques on the back shin (dimension 1), back foot (dimension 2), and front foot (dimension 5), respectively. This high degree of synergy clearly reflects the necessity for highly coordinated torques across the fore and hind limbs and their respective joints when a quadrupedal robot like the Halfcheetah executes gaits such as running or performs complex maneuvers. Such coordination is vital for balance, propulsion, and agile postural adjustments. These strong correlations in the action data reflect its efficient movement patterns.

Fig. 5(b) presents the correlation heatmap for the **Walker2d-medium** dataset. The action space of this bipedal robot also consists of six torque dimensions, controlling the right `thigh_joint`, right `leg_joint`, right `foot_joint`, and the corresponding joints on the left side. Inter-dimensional correlations are also evident in this environment. For instance, the torque on the right `foot_joint` (dimension 2) has a correlation coefficient of 0.48 with the torque on the left `foot_joint` (dimension 5). Concurrently, some association exists between the right `thigh_joint` torque (dimension 0) and the right `leg_joint` torque (dimension 1). This indicates that during bipedal locomotion, the actions of the left and right legs, as well as the torque outputs of different joints within the same leg, are interdependent. Such coordination is crucial for stable gaits, maintaining balance, and enabling effective center of mass transfer.

In the **Hopper-medium** dataset (Fig. 5(c)), this monodopal robot employs a concise three-dimensional action space: `thigh_joint`, `leg_joint`, and `foot_joint`. The heatmap reveals, for instance, a correlation of 0.33 between the `leg_joint` torque (dimension 1) and `foot_joint` torque (dimension 2). Although less prominent than in multi-legged environments, they still signify non-independent actuator torques, even in relatively simple hopping motions. Such correlations likely reflect subtle coordinations. For instance, adjustments in leg and foot posture to absorb landing impact—essential for stability and achieving motor objectives across various phases. Even if numerically less pronounced, these synergistic patterns are crucial for functions like balance or efficient energy transfer, providing a basis for learning-based imputation models, such as our proposed CORIMP, to leverage this underlying inter-dimensional information.

Experimental observations across agent environments with varying morphologies and locomotion modes consistently support our central hypothesis: varying degrees of correlation exist among dimensions within action data samples. This inherent relatedness not only reflects the kinematic and dynamic constraints imposed when agents interact

Table 2
Results with varying missing rates while keeping the missing dimensions fixed at two. *Oracle*, *Zero*, *Mean*, *LR*, *KNN*, and *CORIMP* refer to IQL performance on the original datasets, datasets with missing actions filled with zeros, mean-imputed datasets, LR-imputed datasets, KNN-imputed datasets, and CORIMP-imputed datasets, respectively. *Rel. (%)* indicates the performance of IQL on the CORIMP-imputed datasets as a percentage of the original datasets. Best results among imputation methods are highlighted in bold..

		Oracle	Zero	Mean	LR	KNN	CORIMP	Rel. (%)
Halfcheetah-m	mr = 0.1	48.30 ± 0.32	47.16 ± 0.24	47.39 ± 0.33	48.02 ± 0.28	48.36 ± 0.14	**48.55 ± 0.18**	100.52%
	mr = 0.3		37.77 ± 0.48	42.24 ± 0.56	46.68 ± 0.29	48.03 ± 0.30	**48.12 ± 0.25**	99.63%
	mr = 0.5		25.98 ± 1.25	34.57 ± 0.53	43.24 ± 0.36	46.42 ± 0.18	**46.74 ± 0.19**	96.77%
	mr = 0.7		9.75 ± 4.29	29.16 ± 0.83	40.59 ± 0.35	44.18 ± 0.39	**45.17 ± 0.30**	93.52%
	mr = 0.9		3.30 ± 1.47	20.05 ± 0.88	39.24 ± 0.30	42.72 ± 0.27	**43.72 ± 0.38**	90.52%
Average		48.30	24.88	34.68	43.55	45.94	**46.46**	96.19%
Walker2d-m	mr = 0.1	83.16 ± 3.94	73.64 ± 14.96	80.65 ± 7.03	77.11 ± 3.57	74.59 ± 7.93	**83.47 ± 3.72**	100.37%
	mr = 0.3		73.63 ± 9.72	32.51 ± 13.86	73.78 ± 7.66	**81.87 ± 5.80**	79.20 ± 4.18	95.24%
	mr = 0.5		36.93 ± 15.85	12.31 ± 0.56	63.33 ± 5.85	73.95 ± 6.18	**78.90 ± 5.66**	94.88%
	mr = 0.7		13.73 ± 0.47	10.60 ± 1.90	24.87 ± 0.78	46.21 ± 5.66	**70.69 ± 8.08**	85.00%
	mr = 0.9		9.03 ± 0.97	10.71 ± 2.27	15.50 ± 0.12	25.80 ± 2.90	**34.24 ± 11.61**	41.17%
Average		83.16	41.39	29.36	50.92	60.48	**69.30**	83.33%
Hopper-m	mr = 0.1	57.20 ± 5.56	33.36 ± 3.24	33.13 ± 2.13	30.48 ± 0.16	26.28 ± 0.06	**34.79 ± 5.57**	60.82%
	mr = 0.3		22.53 ± 1.96	22.26 ± 1.72	21.36 ± 0.14	**24.52 ± 0.03**	22.61 ± 1.59	39.53%
	mr = 0.5		14.36 ± 2.19	13.52 ± 2.06	11.21 ± 0.01	12.73 ± 0.01	**14.72 ± 3.01**	25.73%
	mr = 0.7		9.61 ± 0.36	9.41 ± 0.63	9.72 ± 0.01	7.20 ± 0.01	**10.47 ± 0.12**	18.30%
	mr = 0.9		7.61 ± 0.34	7.35 ± 0.23	8.46 ± 0.01	7.25 ± 0.01	**9.61 ± 0.36**	16.80%
Average		57.20	17.49	17.13	16.25	15.60	**18.44**	32.24%
Halfcheetah-me	mr = 0.1	92.51 ± 3.23	89.21 ± 2.36	90.23 ± 0.70	91.71 ± 0.38	91.94 ± 2.80	**92.43 ± 2.72**	99.91%
	mr = 0.3		20.69 ± 4.22	39.54 ± 6.49	73.19 ± 0.84	91.90 ± 1.60	**92.16 ± 2.01**	99.26%
	mr = 0.5		10.56 ± 2.45	13.23 ± 3.53	35.27 ± 3.98	88.42 ± 0.43	**90.77 ± 2.35**	98.12%
	mr = 0.7		2.51 ± 1.32	12.73 ± 2.14	25.57 ± 1.70	83.41 ± 2.45	**87.28 ± 1.05**	94.35%
	mr = 0.9		1.64 ± 0.01	6.15 ± 3.47	18.41 ± 3.19	70.67 ± 4.20	**77.76 ± 5.27**	84.06%
Average		92.51	24.92	32.38	48.83	85.27	**88.08**	95.21%
Walker2d-me	mr = 0.1	110.31 ± 0.53	113.02 ± 0.58	113.12 ± 0.47	111.73 ± 0.09	111.79 ± 0.20	**113.21 ± 1.15**	102.63%
	mr = 0.3		91.28 ± 14.04	90.09 ± 29.40	112.58 ± 0.07	114.13 ± 0.03	**114.63 ± 0.70**	103.92%
	mr = 0.5		23.99 ± 4.09	16.02 ± 1.77	42.42 ± 3.84	80.45 ± 13.63	**85.87 ± 19.49**	77.84%
	mr = 0.7		13.76 ± 1.64	12.27 ± 1.08	17.78 ± 0.12	17.13 ± 0.21	**30.29 ± 12.81**	27.46%
	mr = 0.9		9.23 ± 1.87	10.67 ± 0.94	13.18 ± 0.10	12.33 ± 0.21	**20.05 ± 6.27**	18.18%
Average		110.31	50.26	48.43	59.54	67.17	**72.81**	66.00%
Hopper-me	mr = 0.1	98.94 ± 11.93	26.19 ± 6.08	28.85 ± 2.17	31.69 ± 0.16	24.03 ± 0.36	**31.98 ± 5.66**	32.32%
	mr = 0.3		10.29 ± 4.72	17.78 ± 0.23	18.63 ± 0.15	17.71 ± 0.03	**18.71 ± 4.81**	18.91%
	mr = 0.5		11.72 ± 0.12	13.18 ± 1.55	11.50 ± 0.01	10.37 ± 0.01	**13.64 ± 1.89**	13.79%
	mr = 0.7		7.42 ± 0.62	9.46 ± 0.54	8.17 ± 0.16	**10.04 ± 0.01**	9.53 ± 0.95	9.63%
	mr = 0.9		6.99 ± 1.45	2.18 ± 0.14	**10.63 ± 0.01**	7.18 ± 0.01	8.38 ± 0.80	8.47%
Average		98.94	12.52	14.29	16.12	13.87	**16.45**	16.62%

with the physical world but also forms the critical premise and solid foundation for our CORIMP model to effectively impute missing action dimensions from the known ones.

6.3. Evaluation with varying missing rates

In this section, we keep the number of missing dimensions (denoted as "md") fixed at two and vary the missing rate (denoted as "mr") from 0.1 to 0.9.

6.3.1. Sensitivity analysis of tasks and dataset types

We examine how varying missing rates affect the performance of existing offline RL algorithms, focusing primarily on the TD3BC algorithm across different types of datasets and task domains, with results presented in Fig. 6(a). Our goal is to uncover the unique sensitivity patterns of different tasks and dataset types in response to changes in missing rates. While this analysis primarily focuses on the TD3BC algorithm, similar trends are observed with the IQL algorithm, as shown in Fig. 6(b).

The left plot reveals that for missing rates (mr) from 0.1 to 0.3, `Hopper`'s performance drops most, while that of `Walker2d` remains largely unaffected, indicating its resilience in this lower mr range. However, for mr 0.3-0.7, `Walker2d` deteriorates most significantly, reflecting its heightened sensitivity in this range. `Halfcheetah` maintains a relatively steady decline, whereas `Hopper` gradually decelerates. Finally, for missing rates from 0.7 to 0.9, all three tasks exhibit similar downward trends, culminating in notably poor results at a missing rate of 0.9.

In the right plot, TD3BC on `medium-expert` datasets shows sensitivity patterns like those on `medium` datasets. At missing rates from 0.1 to 0.3, `Halfcheetah` declines most sharply, reflecting its greatest sensitivity within this lower range. Meanwhile, `Walker2d` and `Hopper` remain relatively stable. As the missing rate increases from 0.3 to 0.5, `Walker2d` suffers a dramatic collapse, underscoring its sensitivity over `Halfheetah` and `Hopper`. In the range from 0.5 to 0.9, all three tasks display comparable performance declines, echoing the trends on the `medium` datasets. Ultimately, all tasks perform poorly at a 0.9 missing rate.

The performance patterns of the TD3BC algorithm across the `medium` and `medium-expert` datasets reveal both consistencies and variations. `Walker2d` shows high sensitivity, especially at higher missing rates, implying greater vulnerability to incomplete data. Conversely, `Halfcheetah` is more sensitive at lower missing rates on `medium-expert` datasets, suggesting dataset complexity can amplify its performance decline. Further, while `Hopper` consistently shows a relatively mild decline,

Table 3
Results with varying missing dimensions while keeping missing rate fixed at 0.5. *Oracle*, *Zero*, *Mean*, *LR*, *KNN*, and *CORIMP* refer to TD3BC performance on the original datasets, datasets with missing actions filled with zeros, mean-imputed datasets, LR-imputed datasets, KNN-imputed datasets, and CORIMP-imputed datasets, respectively. *Rel. (%)* indicates the performance of TD3BC on the CORIMP-imputed datasets as a percentage of the original datasets. Best results among imputation methods are highlighted in bold..

		Oracle	Zero	Mean	LR	KNN	**CORIMP**	Rel. (%)
Halfcheetah-m	md = 1	48.30 ± 0.32	42.04 ± 0.55	45.37 ± 0.44	45.68 ± 0.31	47.08 ± 0.34	**47.24 ± 0.3**	97.81%
	md = 2		19.84 ± 2.65	33.97 ± 0.43	42.27 ± 0.42	45.34 ± 0.39	**45.92 ± 0.30**	95.07%
	md = 3		17.02 ± 1.75	23.21 ± 1.26	39.29 ± 1.74	42.55 ± 0.79	**44.38 ± 0.55**	91.88%
	md = 4		7.26 ± 2.15	24.02 ± 1.55	35.20 ± 1.34	36.60 ± 1.44	**38.07 ± 0.99**	78.82%
	md = 5		4.78 ± 1.17	14.31 ± 1.95	23.07 ± 3.15	17.51 ± 3.07	**28.11 ± 2.13**	58.20%
Average		48.30	18.19	28.18	37.10	37.82	**40.74**	84.36%
Walker2d-m	md = 1	83.16 ± 3.94	70.37 ± 9.97	70.37 ± 9.97	81.76 ± 2.62	83.63 ± 1.22	**84.46 ± 0.84**	101.56%
	md = 2		52.78 ± 12.77	16.96 ± 2.05	68.25 ± 17.24	**85.47 ± 5.27**	84.74 ± 4.34	101.90%
	md = 3		6.78 ± 0.74	7.04 ± 0.33	11.01 ± 1.08	11.50 ± 0.43	**13.49 ± 2.02**	16.22%
	md = 4		8.04 ± 0.98	7.83 ± 0.67	8.77 ± 4.42	10.36 ± 0.93	**12.48 ± 1.91**	15.01%
	md = 5		7.45 ± 0.79	7.50 ± 0.57	7.79 ± 0.30	**9.00 ± 0.67**	8.79 ± 1.00	10.57%
Average		83.16	29.08	21.94	35.52	39.99	**40.79**	49.05%
Hopper-m	md = 1	57.20 ± 5.56	54.25 ± 4.41	53.62 ± 7.52	53.98 ± 7.40	51.12 ± 4.69	**54.76 ± 6.41**	95.73%
	md = 2		12.93 ± 1.33	14.96 ± 0.93	14.38 ± 1.58	13.64 ± 2.26	**15.23 ± 1.10**	26.63%
Average		57.20	33.59	34.29	34.18	32.38	**35.00**	61.18%
Halfcheetah-me	md = 1	92.51 ± 3.23	39.81 ± 7.79	66.45 ± 5.77	78.77 ± 1.72	**89.26 ± 1.60**	79.16 ± 2.39	85.57%
	md = 2		7.36 ± 2.86	20.42 ± 2.64	37.23 ± 4.68	82.15 ± 2.22	**84.85 ± 4.81**	91.72%
	md = 3		2.29 ± 0.63	11.90 ± 4.87	36.34 ± 5.32	55.04 ± 8.08	**55.76 ± 6.20**	60.27%
	md = 4		2.05 ± 0.05	2.69 ± 1.08	32.28 ± 1.79	34.82 ± 0.66	**36.40 ± 0.79**	39.35%
	md = 5		2.19 ± 0.53	2.32 ± 0.89	**28.46 ± 1.13**	15.34 ± 8.32	27.46 ± 5.01	29.68%
Average		92.51	10.74	20.76	42.62	55.32	**56.73**	61.32%
Walker2d-me	md = 1	110.31 ± 0.53	103.32 ± 6.14	110.27 ± 0.44	110.09 ± 0.36	110.28 ± 0.48	**110.34 ± 0.48**	100.03%
	md = 2		18.90 ± 3.88	13.29 ± 0.85	34.90 ± 12.79	74.17 ± 27.87	**94.95 ± 13.60**	86.08%
	md = 3		6.95 ± 0.42	6.00 ± 0.24	8.22 ± 1.66	**15.52 ± 2.60**	15.00 ± 1.92	13.60%
	md = 4		5.97 ± 0.34	5.13 ± 0.46	4.60 ± 0.56	**7.72 ± 1.35**	7.38 ± 1.93	23.62%
	md = 5		5.47 ± 0.38	5.40 ± 0.42	6.19 ± 0.73	5.77 ± 0.81	**6.49 ± 0.63**	18.65%
Average		110.31	28.12	28.02	32.80	42.69	**46.83**	42.45%
Hopper-me	md = 1	98.94 ± 11.93	10.91 ± 1.02	10.66 ± 1.64	**10.97 ± 1.19**	9.92 ± 0.88	10.92 ± 1.37	11.04%
	md = 2		11.31 ± 0.73	11.63 ± 0.71	11.20 ± 0.29	10.51 ± 1.65	**11.90 ± 0.94**	12.03%
Average		98.94	11.11	11.15	11.09	10.22	**11.41**	11.53%

this does not necessarily indicate robustness, as having two missing dimensions has already caused its performance to hit rock bottom. These observations suggest that task-specific data incompletion sensitivity is shaped by both dataset complexity and inherent task features.

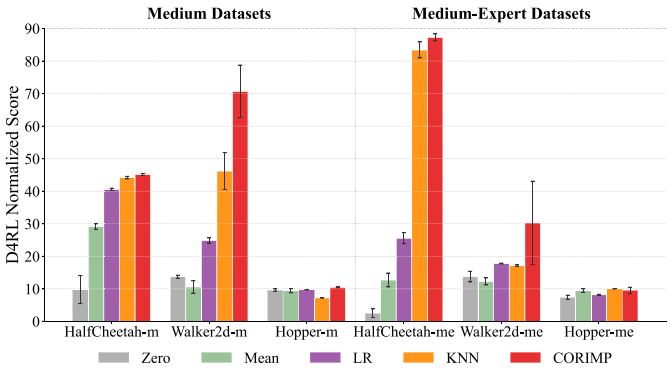


Fig. 8. Downstream offline RL performance comparison. Experiments are conducted under a fixed challenging setting: 2 missing dimensions with a 0.7 missing rate. CORIMP (red) demonstrates superior robustness, maintaining high performance in complex environments like `Walker2d` and `HalfCheetah` where linear (LR) and local (KNN) baselines degrade significantly.

6.3.2. *Performance with CORIMP*

With TD3BC, CORIMP demonstrates marked superiority over the Zero and Mean baselines, and consistently outperforms LR and KNN in the majority of scenarios, while achieving performance close to the oracle datasets, as illustrated in Table 1. Similar performance gains are seen with IQL in Table 2.

On `Halfcheetah`, CORIMP performs well on both dataset types, achieving over 90% of oracle performance. Remarkably, at a 0.7 missing rate on `medium-expert` dataset, CORIMP achieves a striking improvement of 4093.26% over Zero, highlighting its ability to recover even under extreme conditions. For `Walker2d`, CORIMP even surpasses oracle performance at missing rates from 0.1 to 0.5. This may be attributed to CORIMP enhancing data coverage, allowing the agent to learn from a more comprehensive dataset. Regarding `Hopper`, CORIMP shows modest improvements, with performance moderately below oracle levels. This is likely due to the high proportion of missing dimensions (two-thirds), which presents substantial challenges. In this specific low-dimensional setting, KNN occasionally matches or slightly exceeds CORIMP, suggesting that local approximation suffices when dynamics are less complex.

CORIMP excels in the `Halfcheetah` and `Walker2d` tasks. On the one hand, it achieves near-oracle performance at lower missing rates (0.1 and 0.3), underscoring its practical value, as low missing rates are frequently encountered in real-world scenarios. On the other hand, it also demonstrates strong recovery at higher missing rates, making it a powerful tool for extreme situations involving significant data incompleteness. The achievements across various missing rates underscore its

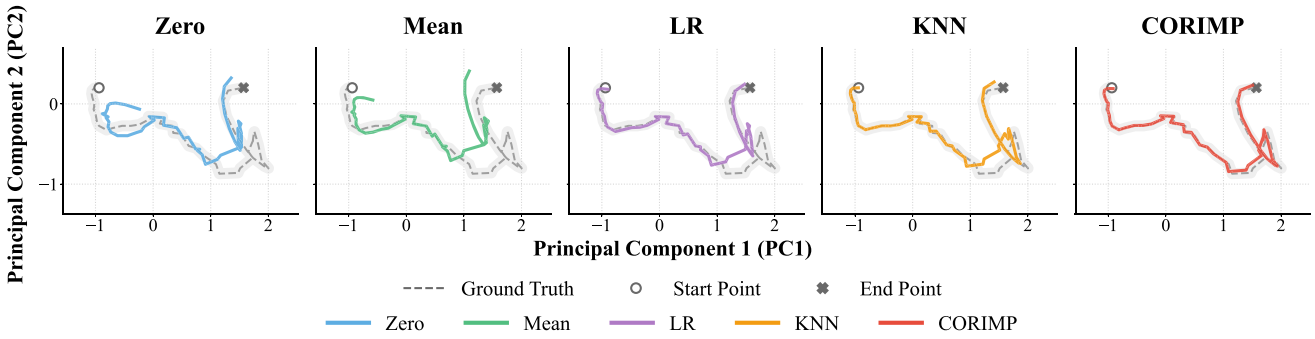


Fig. 9. Visualization of imputed action trajectories via PCA projection on `Walker2d-medium`. The gray corridor represents the ground truth manifold. (1) Initialization Error: Naive baselines (Zero/Mean) deviate immediately at the Start Point (Circle), failing to match the initial pose. (2) Baseline Limitations: While advanced baselines initialize correctly, they fail to track dynamics. LR (purple) produces an *oversmoothed* path that ignores jagged turns, while KNN (orange) exhibits visible *instability* and manifold shrinkage. (3) Ours: In contrast, CORIMP (red) accurately tracks both the initialization and the complex non-linear transitions, preserving the physical plausibility of the motion.

Table 4
Results with varying missing dimensions while keeping missing rate fixed at 0.5. *Oracle*, *Zero*, *Mean*, *LR*, *KNN*, and *CORIMP* refer to IQL performance on the original datasets, datasets with missing actions filled with zeros, mean-imputed datasets, LR-imputed datasets, KNN-imputed datasets, and CORIMP-imputed datasets, respectively. *Rel. (%)* indicates the performance of IQL on the CORIMP-imputed datasets as a percentage of the original datasets. Best results among imputation methods are highlighted in bold..

		Oracle	Zero	Mean	LR	KNN	CORIMP	Rel. (%)
Halfcheetah-m	md = 1	48.30 ± 0.32	42.00 ± 0.57	45.42 ± 0.24	45.96 ± 0.32	47.18 ± 0.10	**47.46 ± 0.14**	98.26%
	md = 2		25.98 ± 1.25	34.57 ± 0.53	43.24 ± 0.36	46.42 ± 0.18	**46.74 ± 0.19**	96.77%
	md = 3		21.70 ± 1.95	29.19 ± 0.78	43.19 ± 0.19	44.52 ± 0.37	**45.44 ± 0.35**	94.08%
	md = 4		12.57 ± 2.65	28.96 ± 2.55	39.12 ± 0.42	39.13 ± 0.21	**39.41 ± 0.22**	81.59%
	md = 5		11.85 ± 2.65	27.79 ± 1.99	**35.86 ± 0.35**	32.92 ± 2.05	35.51 ± 2.56	73.52%
Average		48.30	22.81	33.19	41.47	42.03	**42.91**	88.84%
Walker2d-m	md = 1	83.16 ± 3.94	74.62 ± 4.58	74.62 ± 4.58	78.20 ± 3.56	79.83 ± 3.63	**80.21 ± 4.68**	96.45%
	md = 2		36.93 ± 15.85	12.31 ± 0.56	63.33 ± 5.85	73.95 ± 6.18	**78.90 ± 5.66**	94.88%
	md = 3		5.82 ± 0.79	6.29 ± 0.75	9.94 ± 0.16	11.52 ± 0.77	**12.08 ± 0.79**	14.53%
	md = 4		6.20 ± 0.49	8.28 ± 2.68	10.51 ± 0.03	10.70 ± 0.03	**12.66 ± 2.20**	15.22%
	md = 5		5.07 ± 0.37	7.95 ± 0.75	**9.88 ± 0.02**	8.67 ± 0.03	9.75 ± 0.64	11.72%
Average		83.16	25.73	21.89	34.37	36.75	**38.72**	46.56%
Hopper-m	md = 1	57.20 ± 5.56	58.26 ± 6.57	56.81 ± 4.80	57.33 ± 3.48	52.76 ± 3.08	**61.12 ± 5.61**	106.85%
	md = 2		14.36 ± 2.19	13.52 ± 2.06	11.21 ± 0.01	12.73 ± 0.01	**14.72 ± 3.01**	25.73%
Average		57.20	36.31	35.17	34.27	34.86	**37.92**	66.29%
Halfcheetah-me	md = 1	92.51 ± 3.23	41.48 ± 9.22	53.71 ± 6.83	80.86 ± 1.70	**86.68 ± 5.85**	77.39 ± 6.00	83.66%
	md = 2		10.56 ± 2.45	13.23 ± 3.53	35.27 ± 3.98	88.42 ± 0.43	**90.77 ± 2.35**	98.12%
	md = 3		3.07 ± 1.97	14.09 ± 1.45	38.72 ± 1.65	**49.64 ± 6.94**	46.97 ± 7.67	50.77%
	md = 4		3.31 ± 0.44	9.72 ± 3.00	30.30 ± 1.83	32.91 ± 2.22	**33.16 ± 3.07**	35.84%
	md = 5		18.01 ± 9.95	25.50 ± 9.84	27.45 ± 1.25	32.53 ± 0.16	**35.71 ± 0.89**	38.60%
Average		92.51	15.29	23.25	42.52	**58.04**	56.80	61.40%
Walker2d-me	md = 1	110.31 ± 0.53	90.75 ± 12.08	111.66 ± 0.41	111.45 ± 0.95	111.63 ± 0.41	**111.74 ± 0.93**	101.30%
	md = 2		23.99 ± 4.09	16.02 ± 1.77	42.42 ± 3.84	80.45 ± 13.63	**85.87 ± 19.49**	77.84%
	md = 3		6.99 ± 0.37	6.15 ± 0.24	9.60 ± 0.10	13.03 ± 0.08	**13.87 ± 2.14**	12.57%
	md = 4		5.98 ± 0.09	5.57 ± 0.19	4.53 ± 0.01	6.98 ± 0.02	**8.76 ± 3.72**	7.94%
	md = 5		5.72 ± 0.11	4.79 ± 0.43	5.17 ± 0.01	4.52 ± 0.02	**5.88 ± 0.41**	5.33%
Average		110.31	26.69	28.84	34.63	43.32	**45.22**	40.99%
Hopper-me	md = 1	98.94 ± 11.93	9.42 ± 0.34	11.04 ± 0.90	11.30 ± 0.01	10.36 ± 0.01	**12.44 ± 2.16**	12.57%
	md = 2		11.72 ± 0.12	13.18 ± 1.55	11.50 ± 0.01	10.37 ± 0.01	**13.64 ± 1.89**	13.79%
Average		98.94	10.57	12.11	11.40	10.37	**13.04**	13.18%

effectiveness in addressing different levels of data incompleteness and robustness under extreme data sparsity, demonstrating its value in alleviating the impact of incomplete data on offline RL.

6.4. Evaluation with varying missing dimensions

Here, we fix the missing rate at 0.5 and vary missing dimensions from 1 to $C - 1$.

6.4.1. Sensitivity analysis of tasks and dataset types

We examine how varying missing dimensions affect the performance of existing offline RL algorithms, focusing primarily on the TD3BC algorithm across different types of datasets and task domains, as shown in Fig. 6(c). Our goal is to uncover the unique sensitivity patterns of different tasks and dataset types that react to changes in missing dimensions. While focusing on TD3BC, IQL exhibits comparable sensitivity patterns as Fig. 6(d)) shows.

In the left plot, Hopper exhibits a sharp performance drop as missing dimensions increase from one to two. This decline is expected, as missing data in two of the three action dimensions poses a significant challenge. Halfcheetah and Walker2d also decline sharply as their missing dimensions increase from one to three. Both tasks' performance plunges to critical lows when three dimensions are missing. The decline is most severe for Walker2d. These results suggest that Walker2d's sensitivity to missing dimensions arises from both its inherent characteristics and task complexity. These observations reveal a threshold effect: beyond a certain level of missing data, performance stabilizes at a very low level.

In the right plot, Hopper consistently performs poorly. As missing dimensions rise from one to two, Halfcheetah and Walker2d decline significantly, Walker2d especially sharply. When the number of missing dimensions reaches two or more, the decline in performance for both datasets slows down gradually. With two missing dimensions, performance once again plummets to a notably low level, mirroring previous observations.

6.4.2. Performance with CORIMP

CORIMP demonstrates marked superiority across varying missing dimensions, outperforming the Zero, Mean, LR, and KNN baselines in most settings while maintaining near-oracle performance. Detailed results are available in Tables 3 and 4. Here, we primarily focus on the performance of TD3BC, as shown in Table 3.

For Halfcheetah, CORIMP achieves over 90% of the oracle performance with fewer than three missing dimensions on the medium dataset, with an average performance of 84.36% relative to the oracle. Although CORIMP achieves a modest 61.32% on the medium-expert dataset, it yields a substantial 428.18% improvement over Zero. Regarding Walker2d, CORIMP surpasses oracle performance in three out of four cases with one or two missing dimensions. On medium-expert, even with two missing dimensions, CORIMP reaches 86.08% of the oracle performance, highlighting its practical significance. Notably, in this complex environment, LR often fails to capture the dynamics, leading to unstable performance (e.g., Table 3 shows LR scoring only 34.90 on Walker2d-me at md = 2, while CORIMP reaches 94.95). However, performance drops with more than two missing dimensions, as handling a 0.5 missing rate on over half the dimensions is challenging. For Hopper, CORIMP demonstrates moderate performance relative to the oracle: 61.18% for the medium dataset and 11.53% for the medium-expert dataset. Modest improvements indicate that a high proportion of missing dimensions (two-thirds) makes effective data imputation more challenging.

Overall, CORIMP performs effectively with few missing dimensions, underscoring its practical significance. Compared to baselines like LR and KNN which degrade rapidly as dimension loss increases (indicating structural mismatch or manifold shrinkage), CORIMP maintains superior robustness in preserving policy-relevant features. Nevertheless, as the number of missing dimensions approaches extreme levels, performance inevitably deteriorates, highlighting the persistent challenge of handling severe data incompleteness.

6.5. Analysis of imputation fidelity

To explicitly validate the quality of data reconstruction independent of downstream RL tasks, we benchmark CORIMP's imputation accuracy against four baselines: Zero, Mean, LR, and KNN. We report the Root Mean Square Error (RMSE) between the imputed and ground-truth values across missing dimensions.

Fig. 7 presents the RMSE distributions aggregated over 120 dataset variants (covering diverse missing rates and missing dimension combinations).

- **High-Dimensional Complexity:** In environments with complex dynamics like HalfCheetah and Walker2d, CORIMP (pink) consistently achieves the lowest median RMSE and variance. This confirms that the MLP architecture effectively captures non-linear kinematic correlations that LR fails to model.
- **Comparison with Non-Parametric Methods:** While KNN performs competitively in lower-dimensional tasks (e.g., Hopper), CORIMP outperforms KNN in high-quality datasets (medium-expert), suggesting superior capability in learning the underlying data manifold rather than relying on local neighborhood retrieval.

6.6. From imputation accuracy to downstream performance

To verify whether the superior imputation fidelity observed in Section 6.5 translates to robust downstream policy learning, we further evaluate the offline RL performance using the imputed datasets. We include Zero, Mean, LR, and KNN to investigate how different imputation paradigms affect the final agent scores. We conduct this evaluation using the IQL algorithm under a highly challenging setting: 2 missing dimensions with a 0.7 missing rate.

The results, summarized in Fig. 8, reveal a clear correlation between reconstruction accuracy and policy performance:

- **Robustness in Complex Dynamics: CORIMP** (red) consistently achieves the highest average normalized scores across most tasks. This advantage is most pronounced in the Walker2d-medium and Walker2d-medium-expert environments, where CORIMP significantly outperforms both LR and KNN. This confirms that minimizing reconstruction error via non-linear modeling is critical for recovering actionable signals in complex locomotion tasks.
- **Limitations of Baselines:** The parametric baseline **LR** struggles significantly in high-quality datasets, highlighting the failure of linear assumptions to capture expert-level dynamics. Similarly, the non-parametric baseline **KNN**, while competitive in simpler tasks (e.g., Hopper), falters in Walker2d, suggesting that local neighborhood retrieval is less robust than CORIMP's global manifold learning when handling sparse, high-dimensional data.

6.7. Qualitative analysis via trajectory visualization

To provide a deeper mechanistic understanding of CORIMP's statistical superiority (evidenced by RMSE in Section 6.5 and RL scores in Section 6.6), we further analyze the geometric structure of the imputed action trajectories using Principal Component Analysis (PCA). Visualizing the low-dimensional manifold offers direct insight into how different models handle complex physical dynamics. Fig. 9 displays a representative trajectory segment from the Walker2d-medium dataset (2 missing dimensions with a 0.7 missing rate), characterized by complex non-linear dynamics.

The visualization reveals critical insights into the *structural fidelity* of different imputation paradigms:

- **Immediate Divergence of Naive Baselines:** As observed at the Start Point (Circle), naive baselines like Zero (blue) and Mean (green) fail to match the initial state of the ground truth. This immediate deviation confirms that simple heuristics cannot even recover the static starting pose, leading to catastrophic trajectory errors.
- **Oversmoothing in Linear Models:** While LR (purple) correctly identifies the starting point, it produces an oversmoothed path during the motion. The ground truth exhibits high-frequency jagged turns near the terminal phase. LR fails to capture these sharp geometric fluctuations, due to its inability to model the fine-grained physical constraints required for stable locomotion.
- **Instability in Non-Parametric Models:** KNN (orange) captures the general trend but exhibits noticeable instability. The trajectory appears jittery and suffers from manifold shrinkage at the bottom of the curve, indicating that sparse local neighborhoods lead to discontinuous estimates.
- **Geometric Consistency of CORIMP:** In contrast, CORIMP (red) demonstrates superior geometric fidelity. It accurately initializes

from the start point and faithfully reconstructs the local sharp turns near the endpoint. This qualitative evidence reinforces that CORIMP effectively learns the non-linear manifold structure, validating its strong performance in downstream offline RL tasks.

7. Conclusion

This paper explores the dimension-specific missing action data problem (DSMADP) in offline RL. On the one hand, we thoroughly investigate how different tasks and types of datasets react to DSMADP. On the other hand, we propose CORIMP, a correlation-driven imputation model, to alleviate the impact of DSMADP. Experimental results on variants of missing D4RL datasets demonstrate its effectiveness. Our findings emphasize the significance of addressing DSMADP in offline RL and provide a practical solution through CORIMP. Future work can build on these findings by refining CORIMP and applying it to diverse real-world scenarios to improve its robustness and adaptability.

CRediT authorship contribution statement

Yulin Shao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization; **Yuanbo Xu:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition; **Ximing Li:** Methodology, Validation, Formal analysis, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no competing interests or financial conflicts to disclose.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 92567204, National Natural Science Foundation of China No. 62472196, Jilin Science and Technology Research Project 20230101067JC.

References

An, G., Moon, S., Kim, J.-H., & Song, H. O. (2021). Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in Neural Information Processing Systems*, *34*, 7436–7447.

Bai, C., Wang, L., Yang, Z., Deng, Z., Garg, A., Liu, P., & Wang, Z. (2022). Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. arXiv preprint arXiv:2202.11566.

Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, *34*, 15084–15097.

Cheng, C.-A., Xie, T., Jiang, N., & Agarwal, A. (2022). Adversarially trained actor critic for offline reinforcement learning. In *International conference on machine learning* (pp. 3852–3878). PMLR.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Fang, J., He, H., Xu, M., & Chen, H. (2024). Mdtgan: Multi domain generative adversarial transfer learning network for traffic data imputation. *Expert Systems with Applications*, *255*, 124478.

Fatyanosa, T. N., Firdausanti, N. A., Prayoga, P. H. N., Kuriu, M., Aritsugi, M., & Mendonca, I. (2024). Meta-learning for vessel time series data imputation method recommendation. *Expert Systems with Applications*, *251*, 124016.

Feng, S., & Tan, A.-H. (2016). Towards autonomous behavior learning of non-player characters in games. *Expert Systems with Applications*, *56*, 89–99.

Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219.

Fujimoto, S., & Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, *34*, 20132–20145.

Fujimoto, S., Meger, D., & Precup, D. (2019). Off-policy deep reinforcement learning without exploration. In *International conference on machine learning* (pp. 2052–2062). PMLR.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*, 2672–2680.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (pp. 1861–1870). PMLR.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Janner, M., Li, Q., & Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems*, *34*, 1273–1286.

Jiang, Y., Yang, Y., Xu, Y., & Wang, E. (2023). Spatial-temporal interval aware individual future trajectory prediction. *IEEE Transactions on Knowledge and Data Engineering*, *36*(10), 5374–5387.

Jin, Y., Yang, Z., & Wang, Z. (2021). Is pessimism provably efficient for offline rl? In *International conference on machine learning* (pp. 5084–5096). PMLR.

Kidambi, R., Chang, J., & Sun, W. (2021). Mobile: Model-based imitation learning from observation alone. *Advances in Neural Information Processing Systems*, *34*, 28598–28611.

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

Kim, M., Kim, J., Jung, M., & Oh, H. (2022). Towards monocular vision-based autonomous flight through deep reinforcement learning. *Expert Systems with Applications*, *198*, 116742.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(6), 4909–4926.

Kostrikov, I., Nair, A., & Levine, S. (2021). Offline reinforcement learning with implicit q-learning. arXiv preprint arXiv:2110.06169.

Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, *33*, 1179–1191.

Lange, S., Gabel, T., & Riedmiller, M. (2012). Batch reinforcement learning. In *Reinforcement learning: State-of-the-art* (pp. 45–73). Springer.

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

Li, X., Wang, B., & Ling, X. (2025). Conservative reward enhancement through the nearest neighbor integration in model-based offline policy optimization. *Expert Systems with Applications*, *274*, 126888.

Liu, J., Xu, Y., Song, S., & Jiang, L. (2025). Reducing AUV energy consumption through dynamic sensor directions switching via deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 18843–18851). (*vol. 39*).

Ma, L., Wang, M., & Peng, K. (2024). A missing manufacturing process data imputation framework for nonlinear dynamic soft sensor modeling and its application. *Expert Systems with Applications*, *237*, 121428.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Muratore, F., Gienger, M., & Peters, J. (2019). Assessing transferability from simulation to reality for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(4), 1172–1183.

Nair, A., Gupta, A., Dalal, M., & Levine, S. (2020). Awac: Accelerating online reinforcement learning with offline datasets. arXiv preprint arXiv:2006.09359.

Niu, H., Qiu, Y., Li, M., Zhou, G., Hu, J., Zhan, X. et al. (2022). When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. *Advances in Neural Information Processing Systems*, *35*, 36599–36612.

Park, Y., Margolis, G. B., & Agrawal, P. (2024). Position: Automatic environment shaping is the next frontier in RL. In *Forty-first international conference on machine learning*. https://openreview.net/forum?id=dslUyy1rN4.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Rigter, M., Lacerda, B., & Hawes, N. (2022). Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, *35*, 16082–16097.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*(2), 147.

Schmidt, D., & Jiang, M. (2024). Learning to act without actions. In *The twelfth international conference on learning representations*. https://openreview.net/forum?id=rvUq3cxpDF.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Singh, B., Kumar, R., & Singh, V. P. (2022). Reinforcement learning in robotic applications: A comprehensive survey. *Artificial Intelligence Review*, *55*(2), 945–990.

Sun, W., Vemula, A., Boots, B., & Bagnell, D. (2019). Provably efficient imitation learning from observation alone. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 6036–6045). PMLR (*vol. 97*). Proceedings of Machine Learning Research. https://proceedings.mlr.press/v97/sun19b.html.

Sun, Y., Zhang, J., Jia, C., Lin, H., Ye, J., & Yu, Y. (2023). Model-bellman inconsistency for model-based offline reinforcement learning. In *International conference on machine*

learning (pp. 33177–33194). PMLR.
Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, *35*(6), 2769–2794. https://doi.org/10.1214/009053607000000505
Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, *17*(6), 520–525.
Uehara, M., & Sun, W. (2021). Pessimistic model-based offline reinforcement learning under partial coverage. arXiv preprint arXiv:2107.06226.
Wang, E., Zhang, M., Xu, Y., Xiong, H., & Yang, Y. (2022). Spatiotemporal fracture data inference in sparse urban crowdsensing. In *IEEE infocom 2022-IEEE conference on computer communications* (pp. 1499–1508). IEEE.
Wang, S., Wang, Z., Wang, X., Liang, Q., & Meng, L. (2025). Intelligent vehicle driving decision-making model based on variational autoencoder network and deep reinforcement learning. *Expert Systems with Applications*, *268*, 126319.
Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., & Agarwal, A. (2021). Bellman-consistent pessimism for offline reinforcement learning. *Advances in Neural Information Processing Systems*, *34*, 6683–6694.
Xing, J., Liu, R., Anish, K., & Liu, Z. (2023). A customized data fusion tensor approach for interval-wise missing network volume imputation. *IEEE Transactions on Intelligent Transportation Systems*, *24*(11), 12107–12122.
Xing, R., Zheng, Z., Wu, F., & Chen, G. (2025). User context generation for large language models from mobile sensing data. *IEEE Transactions on Mobile Computing*, *24*, 13678–13695.
Xu, Y., Wang, E., Yang, Y., & Chang, Y. (2021). A unified collaborative representation learning for neural-network based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, *34*(11), 5126–5139.
Yang, R., Zhong, H., Xu, J., Zhang, A., Zhang, C., Han, L., & Zhang, T. (2024). Towards robust offline reinforcement learning under diverse data corruption. In *The twelfth international conference on learning representations*. https://openreview.net/forum?id=5hAMmCU0bK.
Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., & Finn, C. (2021). Combo: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems*, *34*, 28954–28967.
Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., & Ma, T. (2020). Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, *33*, 14129–14142.
Zhang, J., Lyu, J., Ma, X., Yan, J., Yang, J., Wan, L., & Li, X. (2023). Uncertainty-driven trajectory truncation for model-based offline reinforcement learning. arXiv preprint arXiv:2304.04660.
Zheng, J., Jia, R., Liu, S., He, D., Li, K., & Wang, F. (2024). Sample-efficient reinforcement learning with knowledge-embedded hybrid model for optimal control of mining industry. *Expert Systems with Applications*, *254*, 124402.
Zheng, Q., Henaff, M., Amos, B., & Grover, A. (2023). Semi-supervised offline reinforcement learning with action-free trajectories. In *Proceedings of the 40th international conference on machine learning* (pp. 42339–42362). PMLR.
Zhou, W., Bajracharya, S., & Held, D. (2021). Plas: Latent action space for offline reinforcement learning. In *Conference on robot learning* (pp. 1719–1735). PMLR.
Zhou, W., Shen, G., Zhang, Y., Deng, Z., Kong, X., & Xia, F. (2025). Sequence-to-sequence traffic missing data imputation via self-supervised contrastive learning. *IEEE Transactions on Intelligent Transportation Systems*, *26*, 9948–9961.
Zhu, Z., Tian, H., Chen, X., Zhang, K., & Yu, Y. (2025). Offline model-based reinforcement learning with causal structured world models. *Frontiers of Computer Science*, *19*(4), 194347.