URBAN-CENTRIC LATENT MULTI-TASK OFFLINE REINFORCEMENT LEARNING WITH MASK-EMPOWERED REWARD LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In urban environments, human decision-making strategies—such as commuting routes, delivery logistics, and public transit planning—are often shaped by individual preferences and localized constraints, leading to suboptimal outcomes and leaving substantial room for improvement. Offline reinforcement learning (RL) offers a promising paradigm for improving urban decision-making by optimizing policies from pre-collected datasets, without requiring real-time interaction that could raise safety concerns. However, despite its potential, the application of offline RL to urban settings, particularly to realistic multi-task urban environments, remains largely underexplored due to the unique challenges inherent to urban decision-making. In this paper, we propose MORSE—Latent Multi-Task Offline RL with Mask-Empowered Reward Learning, a novel framework for optimizing diverse human strategies in urban environments. MORSE operates in a latent state space through Urban State Compression, improving learning efficiency and scalability. Moreover, it introduces a Mask-Empowered Reward Learning framework that integrates a novel trajectory masked autoencoder (TrajMAE) with a reward ensemble network, jointly trained in a semi-supervised manner, leveraging sparse labeled data from the target task alongside abundant unlabeled data from related tasks. The learned reward function is then used for data relabeling and augmentation. To mitigate the risk of over-reliance on relabeled data, MORSE further employs an Adaptive Knowledge Fusion strategy, enabling robust and generalizable policy learning in complex urban environments. Extensive experiments on real-world datasets demonstrate that MORSE significantly outperforms state-of-the-art baselines in learning diverse urban human strategies.

1 INTRODUCTION

In urban environments, human activities are driven by strategies aimed at achieving specific objectives. For example, taxi drivers choose routes and pick-up locations to minimize idle time and maximize income, while daily commuters select transportation modes to reduce travel time. However, such strategies are typically shaped by individual preferences and localized constraints—for instance, a single commuter's experience may only cover a limited set of routes, time periods, or traffic conditions. As a result, these individually derived strategies are often suboptimal and leave substantial room for improvement. In recent years, offline reinforcement learning (RL) (Levine et al., 2020; Ernst et al., 2005; Lange et al., 2012) has emerged as a promising method for improving urban decision-making strategies by learning policies directly from pre-collected urban datasets. By eliminating the need for real-time interaction with the environment, offline RL enables policy learning and optimization purely from historical data. This property makes it particularly well suited for complex urban settings, where online trial-and-error exploration can be impractical or unsafe, such as in traffic management, transportation systems, and other safety-critical urban applications.

Prior works and limitations. A substantial body of prior work has explored learning urban decision-making strategies using offline RL. For instance, Wan et al. (2023) proposed an offline RL framework for scheduling in real-time wireless traffic systems. Shah et al. (2022) and Shi et al. (2021) developed offline RL approaches for autonomous driving in highway and parking scenarios, respectively. In addition, numerous offline RL methods have been investigated for traffic signal con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

trol (Kunjir & Chawla, 2022; Dai et al., 2021; Li et al., 2023; Liu et al., 2024a; Diehl et al., 2023; Ma & Wu, 2023). Despite these advances, most existing approaches are limited to single-task settings and assume the availability of abundant task-specific data. Extending offline RL to more realistic multi-task urban environments—where tasks share state and action spaces but differ in objectives, rewards, or contextual factors—remains largely underexplored. Several multi-task offline RL methods, such as CDS (Yu et al., 2021a) and UTDS (Bai et al., 2024), attempt to improve multi-task performance by selectively sharing data across related tasks, while SRTD (Yoo et al., 2022) generates synthetic data to facilitate knowledge transfer to target tasks. However, these methods are generic multi-task offline RL algorithms evaluated primarily on standard benchmark datasets, and they are not specifically designed to learn multi-task urban strategies in dynamic and safety-critical urban environments.

Goal and Challenges. These gaps motivate our central research question: *How can we effectively learn and enhance diverse human strategies in urban environments using offline RL?* To address this question, we identify the following key challenges.

Challenge 1: Data Sparsity. Learning effective urban strategies typically requires substantial task-specific data; however, in practice, collecting comprehensive mobility data for each individual is often infeasible. A single commuter's travel history may only cover a limited set of routes and time periods, leaving large gaps across space, time, and varying urban conditions. More broadly, available urban datasets are sparse and fragmented, consisting of incomplete and heterogeneous strategies from diverse agents performing different tasks.

Challenge 2: Implicit Rewards and Scarce Supervision. In urban settings, rewards for human agents are often implicit, evolving, and difficult to specify explicitly due to the complexity and variability of real-world environments. Consequently, labeled transitions are scarce, and human annotation is both expensive and unreliable at scale. Many existing generic multi-task offline RL methods (Yu et al., 2022; Zhao et al., 2024; Bai et al., 2024; Yoo et al., 2022) assume access to well-defined reward functions for each target task, which makes them impractical or unrealistic when applied to urban decision-making problems.

Challenge 3: High-Dimensional Urban State Spaces. Urban decision-making involves high-dimensional and complex state spaces that encode rich spatial–temporal context over large geographic areas and time horizons. These states may include locations, timestamps, traffic conditions, environmental factors, and points of interest (POIs), resulting in strong spatial–temporal dependencies and long-range correlations that significantly complicate policy learning.

The proposed work. We propose MORSE—Latent Multi-task Offline RL with MaSk-Empowered REward Learning—to learn urban policies from sparse target-task reward labels and unlabeled trajectories from related tasks. Our contributions are threefold:

- We develop Mask-Empowered Reward Learning in a compact latent space obtained through Urban State Compression. By jointly training a trajectory masked autoencoder (TrajMAE) and a reward ensemble on labeled and unlabeled trajectories, MORSE learns target-task rewards and relabels cross-task data for augmentation.
- We introduce Adaptive Knowledge Fusion to mitigate reliance on imperfect reward predictions. Periodic exponential moving average updates transfer knowledge from a policy trained on relabeled data to one trained on labeled data, balancing broader data coverage with reliable supervision.
- Experiments on taxi trajectory datasets from Shenzhen and Chengdu demonstrate improved policy performance over offline RL baselines, with reward prediction and ablation studies supporting the framework's components. Our data, code, and supplementary material are publicly available[1].

2 Overview

We model urban decision-making from human-generated spatial-temporal data (HSTD) (Zhang et al., 2023; 2020; Zhang et al.; Zhao et al., 2024) as a family of MDPs. Within each city, each individual corresponds to a task $\mathcal{M}_i = (\mathcal{S}, \mathcal{A}, P, \gamma, R_i)$. Tasks share the state space $\mathcal{S}$, action space

[1] https://anonymous.4open.science/r/anonymorse-0A70

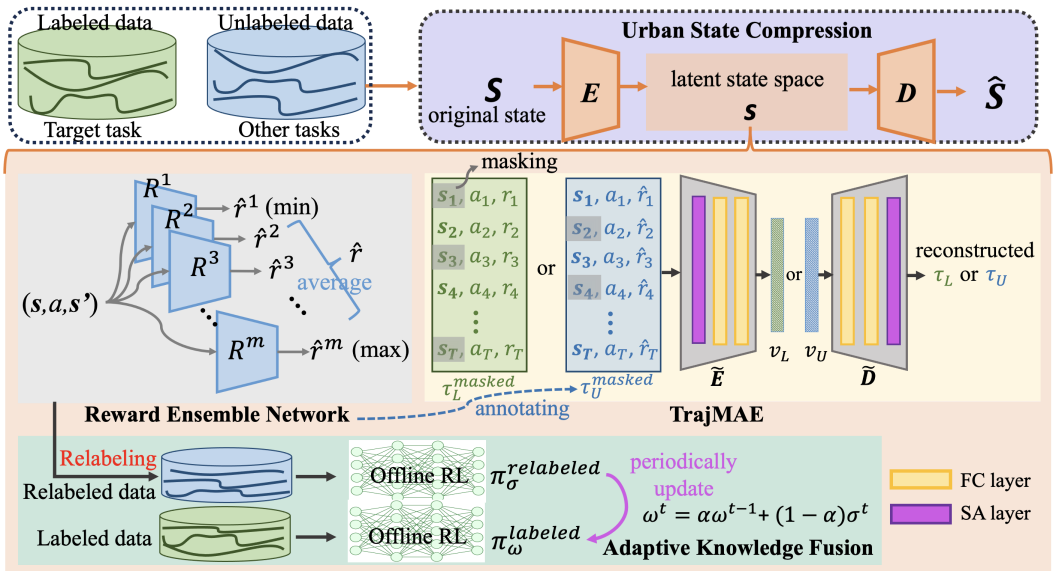


Figure 1: Overview of MORSE. MORSE contains **(1) Urban State Compression**, where we pretrain an autoencoder on states from all tasks to project high-dimensional urban observations into a compact latent space for efficient reward modeling and offline RL. **(2) Mask-Empowered Reward Learning**, where we jointly train a trajectory masked autoencoder (TrajMAE) and a reward ensemble using sparse labeled transitions from the target task and unlabeled trajectories from related tasks, and relabel unlabeled data to augment the target dataset. **(3) Adaptive Knowledge Fusion**, where we train offline policies on labeled and relabeled data respectively and periodically fuse them via an EMA-based update to balance reliability and generalization.

$\mathcal{A}$, transition dynamics P, and discount factor $\gamma \in [0, 1)$, but differ in their reward functions R_i, reflecting individual objectives and preferences.

An original urban state $\boldsymbol{S}$ summarizes spatial-temporal features of a grid cell and its surrounding $l \times l$ neighborhood. Urban State Compression maps it to a compact representation $\boldsymbol{s} = E(\boldsymbol{S}) \in \mathbb{R}^e$ (Section 3.1). Reward learning and policy optimization operate on these latent states, with $\pi(a \mid \boldsymbol{s}, i)$ denoting the policy for task i.

Problem Definition. For a target task i, we are given a small labeled dataset $\mathcal{D}^{\mathrm{L}} = \{(\boldsymbol{s}, a, \boldsymbol{s}', r)\}^i$ and an unlabeled dataset $\mathcal{D}^{\mathrm{U}} = \bigcup_{j \neq i} \{(\boldsymbol{s}, a, \boldsymbol{s}')\}^j$ from related tasks. The target reward function is not explicitly available; reward supervision comes only from $\mathcal{D}^{\mathrm{L}}$, while source-task rewards are unavailable or omitted. The objective is to learn a target-task policy $\pi(a \mid \boldsymbol{s}, i)$ using these fixed datasets, without additional environment interaction, that maximizes the expected discounted return $J_i(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t^i]$, where r_t^i is the reward under the target task.

3 Methodologies

In this section, we address the urban-centric multi-task offline RL problem by introducing an innovative framework, Latent Multi-Task Offline RL with Mask-Empowered Reward Learning (MORSE). As illustrated in Fig. 1, MORSE consists of three main components: (1) Urban State Compression (see Section 3.1), (2) Mask-Empowered Reward Learning (see Section 3.2) and (3) Multi-Task Offline RL with Adaptive Knowledge Fusion (see Section 3.3).

3.1 Urban State Compression

Urban state compression is a critical component of MORSE, enabling efficient and effective reward learning and multi-task offline RL processes. In urban environments, while actions and rewards typically reside in a low-dimensional space, the state space is considerably more complex. This complexity arises because human decision-making involves not only the conditions at the current

location—including spatial-temporal features such as latitude, longitude, time of day, traffic speed, inflow/outflow, travel demand, weather, and distances to points of interest (POIs)—but also similar contextual information from surrounding areas. The resulting high-dimensional and sparse state space poses significant challenges for reward learning and policy optimization, particularly in offline RL given limited and sparsely collected HSTD.

To address this issue, MORSE operates in a simplified latent state space, where each urban state is represented in a low-dimensional latent space using an autoencoder trained on all states across tasks within the same environment. Given a high-dimensional state $\boldsymbol{S}$ in the original urban state space, the encoder E maps $\boldsymbol{S}$ to a compact latent representation $\boldsymbol{s} = E(\boldsymbol{S})$, while the decoder D reconstructs the state as $\hat{\boldsymbol{S}} = D(\boldsymbol{s}) = D(E(\boldsymbol{S}))$. To ensure the latent representation retains critical information of the urban state and is well-structured, we define the objective function for the autoencoder as:

$$\mathcal{L}_{\text{Compress}} = \|\boldsymbol{S} - \hat{\boldsymbol{S}}\|^2 + \beta \text{KL}\left(q(\boldsymbol{s} \mid \boldsymbol{S}) \| p(\boldsymbol{s})\right), \tag{1}$$

where $q(\boldsymbol{s} \mid \boldsymbol{S})$ is the approximate posterior distribution, and $p(\boldsymbol{s})$ is a Gaussian prior. The first term minimizes the reconstruction error between the original and reconstructed states, while the KL-divergence regularization encourages a well-structured latent space. By learning compact and informative latents, the urban state compression approach addresses the complexity of high-dimensional state spaces, enabling MORSE to perform effective reward learning and policy optimization in multi-task urban environments.

Once the autoencoder is learned, original urban states $\{\boldsymbol{S}\}$ are transformed into their corresponding low-dimensional latent states $\{\boldsymbol{s}\}$ for both the labeled dataset $\mathcal{D}^{\text{L}}$ and the unlabeled dataset $\mathcal{D}^{\text{U}}$ using the encoder E. Subsequent reward learning and policy optimization, are conducted in this latent state space.

3.2 Mask-Empowered Reward Learning

For the target task i, we have access to a sparse set of labeled transitions $\mathcal{D}^{\text{L}} = \{(s, a, s', r)\}^i$ but lack the explicit reward function. The limited labeled data $\mathcal{D}^{\text{L}}$ is insufficient to learn a robust reward function for the target task. Directly applying existing approaches (Yu et al., 2020; Kidambi et al., 2020; Yu et al., 2021b) to train a reward network in a fully supervised manner using $\mathcal{D}^{\text{L}}$ would result in a reward function with high variance and poor generalization, as much of the vast state-action space in the urban environment remains unseen during training.

To address this issue, we leverage the large pool of unlabeled data $\mathcal{D}^{\text{U}}$ from other related tasks. While this data cannot provide direct supervision for reward learning due to the absence of reward annotations, it contains valuable information about the environment and the relationships between state-action pairs. This information can significantly benefit reward learning for the target task. For instance, in urban environments, if an unlabeled transition $(s, a, s') \in \mathcal{D}^{\text{U}}$ is structurally similar to a labeled transition $(s, a, s', r) \in \mathcal{D}^{\text{L}}$—*i.e.*, they correspond to similar spatial–temporal states under analogous urban conditions and actions—then their associated rewards are expected to be similar, as supported by recent studies (Yu et al., 2022; Hu et al., 2023). By capturing latent similarities among labeled and unlabeled transitions, the reward function for the target task can generalize more effectively across the environment. This latent similarity forms the foundation for improving reward learning.

Thus, learning a good reward function using the sparse labeled dataset $\mathcal{D}^{\text{L}}$ from the target task and the large pool of unlabeled transitions $\mathcal{D}^{\text{U}}$ from other tasks becomes a critical challenge. To address this, we propose the Mask-Empowered Reward Learning framework, which integrates two key components: a novel masked trajectory autoencoder (TrajMAE) to effectively capture latent similarities among transitions, and a reward ensemble network for reward learning. This framework operates in a semi-supervised manner, combining labeled and unlabeled data to improve the reliability of the learned reward function. Once the reward ensemble for the target task is successfully trained, it can be used to annotate the unlabeled data from other tasks, thereby augmenting the target task's dataset and enabling more effective offline RL learning.

Reward Ensemble Network. As shown in Figure 1, the reward ensemble network is designed to learn the reward function for the target task. Following prior works (Yu et al., 2020; Kidambi et al., 2020), we have an ensemble of m reward models, denoted as $\{R^i\}_{i=1}^m$, to reduce prediction bias. The input of each reward model in this ensemble is a transition without reward annotation

(s, a, s'), the output is a predicted reward $\hat{r}^i = R^i(s, a, s')$. To obtain the final predicted reward $\hat{r}$ for a transition, the highest and lowest predicted rewards are removed, and the remaining $m - 2$ predictions are averaged:

$$\hat{r} = \frac{1}{m-2} \sum_{i=1, i \neq \arg\min_j \hat{r}^j, i \neq \arg\max_j \hat{r}^j}^{m} \hat{r}^i. \tag{2}$$

Trajectory Masked Autoencoder (TrajMAE). The trajectory masked autoencoder (TrajMAE) is designed to learn a latent transition space that captures the spatial-temporal relationships among transitions, providing a comprehensive understanding of their underlying patterns. As illustrated in Figure 1, the TrajMAE comprises two components: a trajectory encoder $\tilde{E}$ and a trajectory decoder $\tilde{D}$. The trajectory encoder $\tilde{E}$ processes a masked trajectory, which can take one of two forms:

$$\tau_{\mathrm{L}}^{\text{masked}} = \left(\boldsymbol{s}_0^{\text{masked}}, a_0, r_0, \ldots, \boldsymbol{s}_T^{\text{masked}}, a_T, r_T\right),$$

if the trajectory is labeled and originates from the target task, or:

$$\tau_{\mathrm{U}}^{\text{masked}} = \left(\boldsymbol{s}_0^{\text{masked}}, a_0, \hat{r}_0, \ldots, \boldsymbol{s}_T^{\text{masked}}, a_T, \hat{r}_T\right),$$

if the trajectory is unlabeled and sampled from other tasks, where rewards $\hat{r}_t$ are annotated using the reward ensemble network. In both cases, a random proportion $p\%$ of the states in the trajectory are masked, such that $\boldsymbol{s}_t^{\text{masked}} = 0$ if masked, and $\boldsymbol{s}_t^{\text{masked}} = \boldsymbol{s}_t$ otherwise. The encoder $\tilde{E}$ maps the masked input trajectory $\tau_{\mathrm{L}}^{\text{masked}}$ or $\tau_{\mathrm{U}}^{\text{masked}}$ to an embedding $\boldsymbol{v}_{\mathrm{L}}$ or $\boldsymbol{v}_{\mathrm{U}}$, respectively. This embedding is then passed to the decoder $\tilde{D}$, which reconstructs the original trajectory $\tau_{\mathrm{L}} = (\boldsymbol{s}_0, a_0, r_0, \ldots, \boldsymbol{s}_T, a_T, r_T)$ or $\tau_{\mathrm{U}} = (\boldsymbol{s}_0, a_0, \hat{r}_0, \ldots, \boldsymbol{s}_T, a_T, \hat{r}_T)$.

Masking transitions within a trajectory is essential for learning a robust latent space that captures underlying similarities across transitions. Sequential transitions exhibit strong dependencies, and masking a portion of the states—primarily encoding spatial-temporal information—encourages the encoder to learn how transitions are interrelated. Through the decoder's reconstruction process, which relies on contextual cues from unmasked states, actions and rewards, the model can better understand how state-action pairs map to rewards. These dependencies also facilitate reward inference for unlabeled transitions by leveraging knowledge from similar labeled transitions. This whole learning process enhances TrajMAE's ability to capture both reward mapping and transition similarities, ultimately improving the generalization and robustness of the reward ensemble network when trained jointly.

Objective. To train the entire Mask-Empowered Reward Learning framework, including both the reward ensemble and the TrajMAE, we use labeled data from the target task and unlabeled data from other tasks in a semi-supervised manner. This involves two distinct cases: labeled data, where true rewards are available, and unlabeled data, where rewards are predicted by the reward ensemble. The objectives for each case are formulated separately as below.

Case 1: Labeled Data. For labeled data $\mathcal{D}^{\mathrm{L}} = \{(s, a, s', r)\}$ in the target task, the true rewards r are used to supervise both the reward ensemble and the TrajMAE. For the reward ensemble, each reward model R^i is trained independently with a supervised MSE loss:

$$\mathcal{L}_{R^i} = \frac{1}{n} \sum_{j=1}^{n} \left(r_j - R^i(s_j, a_j, s'_j)\right)^2, \tag{3}$$

where n is the number of transitions in the labeled dataset $\mathcal{D}^{\mathrm{L}}$.

For the TrajMAE, the labeled trajectory $\tau_{\mathrm{L}}^{\text{masked}}$ includes true rewards. The reconstruction process and objective are as follows:

$$\boldsymbol{v}_{\mathrm{L}} = \tilde{E}\left(\tau_{\mathrm{L}}^{\text{masked}}\right), \quad \tilde{\tau}_{\mathrm{L}} = \tilde{D}\left(\boldsymbol{v}_{\mathrm{L}}\right), \tag{4}$$

$$\mathcal{L}^{\mathrm{L}} = \frac{1}{T} \sum_{t=0}^{T} (\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2 + \|a_t - \tilde{a}_t\|^2 + \|r_t - \tilde{r}_t\|^2), \tag{5}$$

where $(\boldsymbol{s}_t, a_t, r_t) \sim \tau_{\mathrm{L}}$, and $\tilde{\tau}_{\mathrm{L}} = (\tilde{\boldsymbol{s}}_1, \tilde{a}_1, \tilde{r}_1, \cdots, \boldsymbol{s}_T, \tilde{a}_T, \tilde{r}_T)$ are reconstructed by the decoder $\tilde{D}$, *i.e.*, $(\tilde{\boldsymbol{s}}_t, \tilde{a}_t, \tilde{r}_t) \sim \tilde{D}$.

Case 2: Unlabeled Data. For unlabeled data $\mathcal{D}^{\mathrm{U}}$ from other tasks, the rewards are not directly available. In this case, the reward ensemble is used to annotate rewards for the unlabeled transitions with Eq. (2). The annotated trajectory is given by $\tau_{\mathrm{U}}^{\text{masked}}$. Using the annotated rewards, the reconstruction process and objective for the TrajMAE are formulated as:

$$\begin{aligned} \boldsymbol{v}_{\mathrm{U}} &= \tilde{E}\left(\tau_{\mathrm{U}}^{\text{masked}}\right), \quad \tilde{\tau}_{\mathrm{U}} = \tilde{D}\left(\boldsymbol{v}_{\mathrm{U}}\right), \\ \mathcal{L}^{\mathrm{U}} &= \frac{1}{T}\sum_{t=0}^{T}(\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|^2 + \|a_t - \tilde{a}_t\|^2 + \|\hat{r_t} - \tilde{r}_t\|^2), \end{aligned} \tag{6}$$

where $(\boldsymbol{s}_t, a_t, \hat{r_t}) \sim \tau_{\mathrm{U}}$, and $\tilde{\tau}_{\mathrm{U}} = (\tilde{\boldsymbol{s}}_1, \tilde{a}_1, \tilde{r}_1, \cdots, \boldsymbol{s}_T, \tilde{a}_T, \tilde{r}_T)$ are reconstructed by the decoder $\tilde{D}$, *i.e.*, $(\tilde{\boldsymbol{s}}_t, \tilde{a}_t, \tilde{r}_t) \sim \tilde{D}$.

Eq. (6) simultaneously updates the reward ensemble network, as the reward prediction process is inherently embedded in this objective (with the use of $\hat{r_t}$). This objective encourages the reward ensemble network to generalize more effectively by leveraging the transition similarities learned during the reconstruction process.

Final Objective. The overall objective function combines the losses from both labeled and unlabeled cases:

$$\mathcal{L} = \mathcal{L}_{\text{ensemble}} + \mathcal{L}^{\mathrm{L}} + \mathcal{L}^{\mathrm{U}}, \tag{7}$$

where $\mathcal{L}_{\text{ensemble}} = \sum_{i=1}^{m} \mathcal{L}_{R^i}$. With this final objective, the labeled data provides supervision for reward learning. The latent structural similarities among transitions are then learned by the TrajMAE with both labeled and unlabeled data, which in turn, enhances the reward learning by considering the underlying connections among transitions.

3.3 Multi-Task Offline RL with Adaptive Knowledge Fusion

To optimize the policy π for the target task, relying solely on the originally labeled data $\mathcal{D}^{\mathrm{L}}$ is insufficient. Due to the sparsity of $\mathcal{D}^{\mathrm{L}}$, π trained exclusively on this dataset is prone to significant distributional shift (Levine et al., 2020). Instead, MORSE leverages additional relabeled data. Using the learned reward ensemble network in the Mask-Empowered Reward Learning framework, the unlabeled data can be finally annotated, generating a relabeled dataset $\mathcal{D}^{\text{relabeled}}$. This augmented dataset, when combined with the originally labeled data $\mathcal{D}^{\mathrm{L}}$, has the potential to significantly enhance policy learning. However, directly relying on $\mathcal{D}^{\text{relabeled}} \cup \mathcal{D}^{\mathrm{L}}$ poses risks. Despite the well-trained reward ensemble network, the inherent complexity of urban environments makes it challenging for the reward model to perfectly generalize across all transitions. Consequently, some relabeled data may contain inaccurate or unreliable annotations, and over-reliance on such data can degrade policy performance.

To mitigate these risks, MORSE introduces a novel adaptive knowledge fusion strategy for offline RL. This strategy involves simultaneously training two policies using an offline RL algorithm: π^{labeled}, trained on the originally labeled dataset $\mathcal{D}^{\mathrm{L}}$, which produces reliable actions for states present in or similar to $\mathcal{D}^{\mathrm{L}}$ but struggles to generalize across the entire urban environment due to data sparsity; and $\pi^{\text{relabeled}}$, trained on the augmented dataset $\mathcal{D}^{\text{relabeled}}$, which covers a broader range of urban states but may occasionally yield suboptimal actions due to inaccuracies in the relabeled rewards. In this work, we employ IQL (Kostrikov et al., 2021) for learning π^{labeled} and $\pi^{\text{relabeled}}$, given its demonstrated reliability and strong performance across diverse applications (Chitnis et al., 2024; Tangri et al., 2024; Yang et al., 2024).

As shown in Figure 1, both π^{labeled} and $\pi^{\text{relabeled}}$ share the same architecture but maintain distinct weights. To balance policy reliability and generalization, we propose to progressively transfer knowledge from $\pi^{\text{relabeled}}$ to π^{labeled} using an exponential moving average (EMA) update:

$$\omega^t := \alpha\omega^{t-1} + (1-\alpha)\sigma^t, \tag{8}$$

where ω^t represents the parameters of π^{labeled} at training step t, σ^t represents the parameters of $\pi^{\text{relabeled}}$ at the same step, and $\alpha \in [0, 1]$ is a hyperparameter controlling the rate of knowledge fusion.

To ensure training stability and reliability, EMA updates are not applied at every training step. Instead, π^{labeled} and $\pi^{\text{relabeled}}$ are trained simultaneously, with π^{labeled} periodically updated using

EMA as defined in Eq. (8). The knowledge fusion interval between two successive EMA updates is governed by a hyperparameter η. If η is too small (indicating overly frequent EMA updates), the resulting policy may overly depend on $\pi^{\text{relabeled}}$, reducing reliability and destabilizing training. Conversely, if η is too large (indicating infrequent EMA updates), the resulting policy may not adequately incorporate the broader generalization capabilities of $\pi^{\text{relabeled}}$, limiting its effectiveness in complex urban environments.

This adaptive knowledge fusion allows π^{labeled} to progressively integrate the broader generalization capabilities of $\pi^{\text{relabeled}}$ while mitigating the impact of its inaccuracies and preserving the reliability of π^{labeled}. Consequently, the final π^{labeled} becomes more adaptable to the complexities of the urban environment, achieving an improved balance between reliability and generalization.

4 Experiments

In our experiments, we aim to evaluate whether MORSE can achieve satisfactory performance in the real-world multi-task urban setting. We will answer the following questions with extensive experiments: **Q1:** Can our MORSE effectively learn good policies for different tasks compared to state-of-the-art baselines in the multi-task urban setting? **Q2:** Can the reward learning method in MORSE effectively learn rewards? **Q3:** How do different components of MORSE affect the performance? **Q4:** How do different levels of labeled and unlabeled data quality (expert, medium, and non-expert generated by a random policy) affect performance? **Q5:** How do hyperparameters (including the qualities of the labeled and unlabeled data) affect the performance?

4.1 Dataset and Experiment Descriptions

Datasets. We evaluate MORSE on real-world taxi passenger-seeking trajectories from Shenzhen (50 drivers) and Chengdu (30 drivers), processed using the same pipeline. Within each city, each driver is modeled as a task with a distinct reward function under shared urban dynamics. Each observation comprises 125 urban and spatiotemporal features, compressed into a d-dimensional latent state via Urban State Compression (Section 3.1). The action space contains 18 discrete movement actions. Dataset statistics, feature specifications, and preprocessing details are provided in the appendix.

Evaluation Simulator. For each city, we construct a passenger-seeking simulator comprising seven independently trained dynamics models, with an out-of-distribution termination rule to restrict rollouts to supported regions (Kidambi et al., 2020; Yu et al., 2020). The simulator is trained separately from policy learning, used exclusively for evaluation, and shared by all methods. Simulator construction and validation are detailed in the appendix, and the code is released anonymously.

4.2 Baselines

(1) Offline RL. We compare MORSE with single-task methods IQL (Kostrikov et al., 2021), CQL (Kumar et al., 2020), BCQ (Fujimoto et al., 2018), and BEAR (Kumar et al., 2019). Multi-task baselines include No Sharing and Sharing All (Yu et al., 2022; 2021a), both using IQL; UDS (Yu et al., 2022); MODA (Zhao et al., 2024); SRTD (Yoo et al., 2022); Ludor (Wang et al., 2024b); CDS (Yu et al., 2021a); and HarmoDT (Hu et al., 2024). MORSE requires reward labels only for the target task. Baseline implementations and reward supervision settings are detailed in the appendix.

(2) Reward Learning. We evaluate Mask-Empowered Reward Learning against a Reward Ensemble Network (Yu et al., 2020; Kidambi et al., 2020), with and without urban state compression, as well as UDS (Yu et al., 2022) and the representation-based Reward Pred. (Hlynsson & Wiskott, 2021). Implementation details are provided in the appendix.

4.3 Empirical Results

Results of Q1 (Offline RL Performance). To validate that MORSE can effectively learn high-quality policies, we compare it against state-of-the-art offline RL baselines on two cities, Shenzhen and Chengdu, each with three drivers of varying expertise: an expert driver, a medium-expertise driver, and a non-expert driver following a random policy. As shown in Table 1, each value is the av-

Method	ShenZhen City				Chengdu City			
	Expert	Medium	Random	Avg	Expert	Medium	Random	Avg
MORSE	**687.37±28.26**	**597.22±26.31**	**469.67±37.84**	**584.75±18.02**	**590.14±24.73**	**504.33±26.45**	**378.20±32.92**	**490.89±16.31**
No Sharing+IQL	526.05±35.61	450.03±27.83	332.67±31.62	436.25±18.39	457.10±24.16	343.78±34.09	250.08±28.65	350.32±16.89
Sharing All+IQL	523.69±26.93	504.56±22.32	406.51±28.18	478.25±14.97	442.16±19.60	375.39±23.04	293.81±26.54	370.45±13.41
UDS	470.89±31.06	431.22±27.47	383.05±36.09	428.39±18.32	384.20±30.58	315.74±31.25	250.17±29.03	316.70±17.49
MODA	613.82±44.05	547.89±37.36	438.65±40.19	533.45±23.46	536.51±30.95	475.80±29.82	369.43±36.07	460.58±18.70
BCQ	404.07±11.64	323.20±35.25	260.03±37.83	329.10±17.67	390.44±23.86	305.30±22.17	219.89±18.36	305.21±12.46
BEAR	499.98±50.22	466.82±21.88	337.87±51.29	434.89±25.01	416.48±37.24	377.50±34.72	275.61±39.04	356.53±21.39
CQL	502.08±36.76	470.92±32.88	329.25±25.09	434.08±18.44	441.29±24.82	405.51±30.68	297.65±29.32	381.48±16.39
SRTD	512.53±56.88	489.93±48.92	401.98±41.34	468.15±28.55	446.30±22.79	397.62±33.68	333.49±25.11	392.47±15.93
CDS	537.91±38.73	493.16±31.56	382.78±34.46	471.28±20.23	486.10±20.77	445.32±22.93	343.48±28.26	424.97±13.97
Ludor	489.17±21.27	451.12±12.04	390.26±36.97	443.52±14.77	452.74±29.13	376.35±31.81	300.29±34.25	376.46±18.36
HarmoDT	597.67±24.01	522.70±38.27	368.19±31.69	496.19±18.39	502.61±25.76	443.08±32.95	311.42±26.42	419.04±16.49

Table 1: Policy learning performance across drivers and cities. All the learned policies are evaluated using 20 rollouts in the environment. The values are averaged over three random seeds. Standard deviations for individual drivers are computed over three seeds; the Avg standard deviation is obtained via error propagation across drivers.

Method	Expert	Medium	Random	Method	Expert	Medium	Random
MORSE	**7.76/13.90**	**8.44/14.42**	**7.68/14.65**	R-Ensemble w/o S.C.	14.30/18.88	14.19/17.73	20.56/30.40
MORSE w/o S.C.	10.92/14.48	11.28/15.24	12.13/24.41	UDS	51.58/59.30	39.09/47.17	43.98/52.33
R-Ensemble w/ S.C.	11.36/16.18	13.95/16.92	16.48/27.82	Reward Pred.	24.90/31.11	17.86/23.37	22.27/40.78

Table 2: Reward prediction errors on Shenzhen, reported as MAE/RMSE (lower is better). S.C. denotes state compression.

erage expected cumulative reward over 20 rollouts across three random seeds. MORSE achieves the best performance in every driver–city setting, outperforming not only single-task offline RL methods but also state-of-the-art multi-task data-sharing approaches (e.g., MODA, CDS) and the multi-task Decision Transformer HarmoDT. Consistent with the data-quality analysis (Table 4), MORSE's margin is largest on expert drivers and narrowest on random drivers—where the sparse target labels provide the weakest supervision—yet remains positive throughout. These results show that MORSE optimizes policies robustly across drivers of different expertise and across both cities.

Results of Q2 (Reward Learning Performance). To evaluate the effectiveness of the Mask-Empowered Reward Learning framework in MORSE, Table 2 presents reward prediction results in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for MORSE and baseline models across three drivers with different expertise (*i.e.*, expert, medium, and non-expert with random policy). The results show that MORSE consistently achieves the lowest MAE and RMSE values across all drivers, demonstrating the effectiveness of the Mask-Empowered Reward Learning framework in enhancing reward estimation. Furthermore, we observe that reward learning in the latent state space consistently outperforms learning in the original state space. This highlights the critical role of State Compression (S.C.) in improving reward prediction accuracy, ultimately enhancing the model's robustness in multi-task urban environments.

Results of Q3 (Ablation Studies). To investigate how different components of MORSE affect its performance, we conduct an ablation study by removing key modules including State Compression (S.C.), TrajMAE, and Knowledge Fusion (K.F.) from MORSE. Table 3 summarizes the results.

The results clearly show that the full MORSE model achieves the highest performance across all tasks, while the ablated variants exhibit noticeable performance degradation. This indicates that each component of MORSE plays a critical role in enhancing overall effectiveness. State Compression, TrajMAE, and Knowledge Fusion collectively contribute to the robustness and efficiency of the learned policy in multi-task urban settings. Furthermore, we investigate whether State Compression enhances learning efficiency.

Table 4 investigates the impact of labeled and unlabeled data quality (expert, medium, and non-expert with a random policy) on the performance of MORSE and baselines. From the results, we observe that MORSE consistently outperforms competitive baselines, demonstrating its effectiveness and robustness across different data quality settings. Moreover, MORSE achieves the highest performance when both labeled and unlabeled data are of high quality (*i.e.*, expert-level data). As unlabeled data quality degrades (*i.e.*, expert-labeled paired with random unlabeled), performance

Method	Expert	Medium	Random	Method	Expert	Medium	Random
MORSE	**687.37**±28.26	**597.22**±26.31	**469.67**±37.84	w/o TrajMAE	622.59±19.91	546.85±21.33	420.09±24.60
w/o S.C.	656.86±30.35	562.07±24.38	421.80±25.26	w/o K.F.	595.38±29.92	557.41±25.10	449.44±27.83

Table 3: Ablation results on Shenzhen. S.C.: state compression; K.F.: knowledge fusion.

Labeled Data	Unlabeled Data	MORSE	No Sharing	Sharing All	CDS	MODA
Expert	Expert	**724.18**±21.93	526.05±35.61	622.71±33.46	568.57±26.45	641.39±42.86
Expert	Random	**625.85**±28.90	526.05±35.61	544.17±19.17	523.48±28.69	559.11±25.91
Medium	Expert	**616.39**±27.25	483.18±30.01	536.95±28.23	518.11±33.08	552.75±30.68
Medium	Random	**523.39**±35.30	483.18±30.01	513.18±27.62	471.70±31.72	507.71±34.94
Random	Expert	**543.68**±24.47	385.26±29.58	507.15±25.30	455.81±33.67	531.03±28.17
Random	Random	**422.37**±31.14	385.26±29.58	407.05±26.50	415.88±45.05	416.09±27.64

Table 4: Performance under varying labeled and unlabeled data quality on Shenzhen. Expert/Medium denote driver expertise; Random denotes a random driving policy.

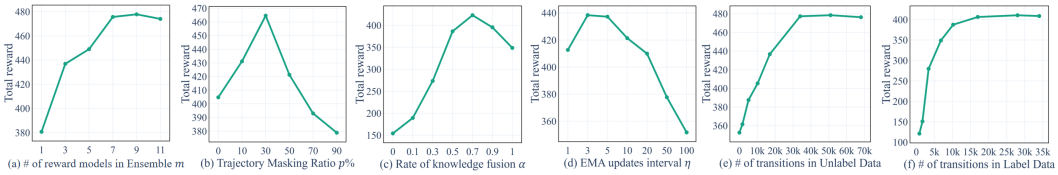


Figure 2: Impact of hyperparameters on MORSE.

declines noticeably. A similar trend is observed when labeled data quality decreases from expert to medium or random, regardless of the unlabeled data quality. These results highlight the importance of high-quality labeled and unlabeled data in achieving superior performance.

Results of Q5 (Hyperparameters). We further evaluate the impact of key hyperparameters on performance, including the number of reward models in the ensemble (m), the trajectory masking ratio ($p\%$) in TrajMAE, the Knowledge Fusion rate (α), and the EMA update interval (η). As shown in Figure 2(a), performance improves as m increases, but beyond $m = 7$, further gains are marginal. Figure 2(b) indicates an optimal masking ratio of around 30%. In Figure 2(c), when $\alpha < 0.5$, the model struggles to learn effectively from labeled data, whereas for $\alpha > 0.7$, it fails to fully integrate knowledge from relabeled data, hindering generalization. Figure 2(d) shows that an EMA update interval η between 3 and 5 yields the best performance; overly frequent updates degrades performance, while excessively infrequent updates lead to significant deterioration. Furthermore, Figure 2(e) and (f) examine the effect of labeled and unlabeled data size on performance. Increasing one type of data while keeping the other fixed consistently improves performance. However, when data volume becomes sufficiently large, performance gains plateau.

5 Conclusion

In this paper, we propose MORSE—Latent Multi-task Offline RL with MaSk-Empowered REward Learning. MORSE performs urban state compression to project high-dimensional states into a latent state space, improving the efficiency and effectiveness of subsequent learning. Besides, it features a novel Mask-Empowered Reward Learning framework that combines a masked trajectory autoencoder (TrajMAE) and a reward ensemble network, jointly trained in a semi-supervised way to enable reward relabeling and data augmentation. Additionally, MORSE incorporates an Adaptive Knowledge Fusion strategy that progressively integrates policies learned from labeled data with insights from relabeled data, ensuring reliable policy optimization and generalization. Extensive experiments in real-world multi-task urban settings validate MORSE's effectiveness and robustness. One limitation is that MORSE's gains hinge on the quality of the sparse target-task labels and on task relatedness, shrinking when supervision is weak or task reward functions diverge sharply (cf. Table 4); extending MORSE to weakly related or conflicting tasks is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing parts of the manuscript. We reviewed all AI-assisted suggestions and checked the incorporated text against the implemented method and reported results. We take responsibility for the final content of this work, including text and claims produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the MORSE method and training objectives. The appendix provides the detailed algorithm, implementation and training settings, dataset selection, and evaluation setup. Code is available at `https://anonymous.4open.science/r/anonymorse-0A70`.

REFERENCES

Chenjia Bai, Lingxiao Wang, Jianye Hao, Zhuoran Yang, Bin Zhao, Zhen Wang, and Xuelong Li. Pessimistic value iteration for multi-task data sharing in offline reinforcement learning. *Artificial Intelligence*, 326:104048, January 2024. ISSN 0004-3702. doi: 10.1016/j.artint.2023.104048. URL http://dx.doi.org/10.1016/j.artint.2023.104048.

Rohit Bokade and Xiaoning Jin. Offlight: An offline multi-agent reinforcement learning framework for traffic signal control. *arXiv preprint arXiv:2411.06601*, 2024.

Rohan Chitnis, Yingchen Xu, Bobak Hashemi, Lucas Lehnert, Urun Dogan, Zheqing Zhu, and Olivier Delalleau. Iql-td-mpc: Implicit q-learning for hierarchical model predictive control. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9154–9160. IEEE, 2024.

Xingyuan Dai, Chen Zhao, Xiaoshuang Li, Xiao Wang, and Fei-Yue Wang. Traffic signal control using offline reinforcement learning. In *2021 China Automation Congress (CAC)*, pp. 8090–8095, 2021. doi: 10.1109/CAC53003.2021.9728551.

Christopher Diehl, Timo Sievernich, Martin Krüger, F. Hoffmann, and Torsten Bertram. Uncertainty-aware model-based offline reinforcement learning for automated driving. *IEEE Robotics and Automation Letters*, 8:1167–1174, 2023. doi: 10.1109/LRA.2023.3236579.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556, dec 2005. ISSN 1532-4435.

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

Hlynur Davíð Hlynsson and Laurenz Wiskott. Reward prediction for representation learning and reward shaping, 05 2021.

Hao Hu, Yiqin Yang, Qianchuan Zhao, and Chongjie Zhang. The provable benefits of unsupervised data sharing for offline reinforcement learning. *arXiv preprint arXiv:2302.13493*, 2023.

Shengchao Hu, Ziqing Fan, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. Harmodt: Harmony multi-task decision transformer for offline reinforcement learning. *arXiv preprint arXiv:2405.18080*, 2024.

Xingshuai Huang, Di Wu, and Benoit Boulet. Traffic signal control using lightweight transformers: An offline-to-online rl approach. *arXiv preprint arXiv:2312.07795*, 2023.

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

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

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *CoRR*, abs/1906.00949, 2019. URL http://arxiv.org/abs/1906.00949.

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

Mayuresh Kunjir and Sanjay Chawla. Offline reinforcement learning for road traffic control. *CoRR*, abs/2201.02381, 2022. URL https://arxiv.org/abs/2201.02381.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. *Batch Reinforcement Learning*, pp. 45–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_2. URL https://doi.org/10.1007/978-3-642-27645-3_2.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Jianxiong Li, Shichao Lin, Tianyu Shi, Chujie Tian, Yu Mei, Jian Song, Xianyuan Zhan, and Ruimin Li. A fully data-driven approach for realistic traffic signal control using offline reinforcement learning, 2023.

Yicheng Liu, Chumeng Liang, Zherui Huang, and Guanjie Zheng. Motsc: Model-based offline traffic signal control. 2024a.

Zongkai Liu, Qian Lin, Chao Yu, Xiawei Wu, Yile Liang, Donghui Li, and Xuetao Ding. Offline multi-agent reinforcement learning via in-sample sequential policy optimization. *arXiv preprint arXiv:2412.07639*, 2024b.

Jinming Ma and Feng Wu. Effective traffic signal control with offline-to-online reinforcement learning. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–7, 2023. doi: 10.1109/IROS55552.2023.10341718.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7780–7791. PMLR, 2021.

Xiaoteng Pan, Zhaoran Wang, and Ji Liu. Plan better amid conservatism: Offline multi-agent reinforcement learning with conservative and risk-aware policies. In *International Conference on Machine Learning*, pp. 17427–17437. PMLR, 2022.

Vitchyr H Pong, Ashvin V Nair, Laura M Smith, Catherine Huang, and Sergey Levine. Offline meta-reinforcement learning with online self-supervision. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 17811–17829. PMLR, 2022.

Dhruv Shah, Arjun Bhorkar, Hrishit Leen, Ilya Kostrikov, Nicholas Rhinehart, and Sergey Levine. Offline reinforcement learning for visual navigation. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=uhIfIEIiWm_.

Tianyu Shi, Dong Chen, Kaian Chen, and Zhaojian Li. Offline reinforcement learning for autonomous driving with safety and exploration enhancement. *CoRR*, abs/2110.07067, 2021. URL https://arxiv.org/abs/2110.07067.

Arsh Tangri, Ondrej Biza, Dian Wang, David Klee, Owen Howell, and Robert Platt. Equivariant offline reinforcement learning. *arXiv preprint arXiv:2406.13961*, 2024.

Jialin Wan, Sen Lin, Zhaofeng Zhang, Junshan Zhang, and Tao Zhang. Scheduling real-time wireless traffic: a network-aided offline reinforcement learning approach. *IEEE internet of things journal*, 10(24):22331–22340, 2023.

Lei Wang, Yu-Xuan Wang, Jian-Kang Li, Yi Liu, and Jia-Tian Pi. Adaptive traffic signal control method based on offline reinforcement learning. *Applied Sciences*, 14(22):10165, 2024a.

Zhao Wang, Briti Gangopadhyay, Jia-Fong Yeh, and Shingo Takamatsu. Augmenting offline rl with unlabeled data, 2024b. URL https://arxiv.org/abs/2406.07117.

Kun Yang, Chengshuai Shi, Cong Shen, Jing Yang, Shu-ping Yeh, and Jaroslaw J Sydir. Offline reinforcement learning for wireless network optimization with mixture datasets. *IEEE Transactions on Wireless Communications*, 2024.

Minjong Yoo, Sangwoo Cho, and Honguk Woo. Skills regularized task decomposition for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 37432–37444, 2022.

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

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *CoRR*, abs/2109.08128, 2021a. URL https://arxiv.org/abs/2109.08128.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: conservative offline model-based policy optimization. *CoRR*, abs/2102.08363, 2021b. URL https://arxiv.org/abs/2102.08363.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. *CoRR*, abs/2202.01741, 2022. URL https://arxiv.org/abs/2202.01741.

Xin Zhang, Yanhua Li, Xun Zhou, Ziming Zhang, and Jun Luo. Trajgail: Trajectory generative adversarial imitation learning for long-term decision analysis. In *ICDM'20*.

Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. Cgail: Conditional generative adversarial imitation learning—an application in taxi drivers' strategy learning. *IEEE transactions on big data*, 8(5): 1288–1300, 2020.

Yingxue Zhang, Yanhua Li, Xun Zhou, Ziming Zhang, and Jun Luo. Stm-gail: Spatial-temporal meta-gail for learning diverse human driving strategies. In *2023 SIAM International Conference on Data Mining (SDM)*, 2023.

Xinbo Zhao, Yingxue Zhang, Xin Zhang, Yu Yang, Yiqun Xie, Yanhua Li, and Jun Luo. Urban-focused multi-task offline reinforcement learning with contrastive data sharing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 4512–4523, 2024. doi: 10.1145/3637528.3671823. URL https://doi.org/10.1145/3637528.3671823.

A Appendix

A.1 Related Work

Offline RL in Urban Environments. Offline RL has shown promise in addressing urban challenges, including traffic management and autonomous driving. Methods like Kunjir & Chawla (2022), Wang et al. (2024a), and Huang et al. (2023) have advanced traffic signal control using offline RL. Multi-agent frameworks, such as OffLight (Bokade & Jin, 2024), risk-aware methods by Pan et al. (2022), and sequential optimization by Liu et al. (2024b), address urban coordination challenges. However, these approaches typically assume abundant data availability, an unrealistic scenario in sparse multi-task urban environments, highlighting the need for novel data-efficient solutions.

Multi-Task Offline RL. Multi-task offline RL aims to optimize policies across multiple tasks using static datasets, enhancing both learning efficiency and generalization. However, it typically requires reward labels for every transition, which are prohibitively expensive to obtain in many practical applications (*e.g.*, healthcare, autonomous driving), as rewards are often manually annotated by humans. To address this, various data-sharing and augmentation strategies have been explored. For instance, CDS (Yu et al., 2021a) selectively shares transitions between tasks to identify data beneficial to the target task, while UTDS (Bai et al., 2024) employs uncertainty-based sharing to address distributional shifts. MODA (Zhao et al., 2024) leverages contrastive learning to identify task-relevant data, and UDS (Yu et al., 2022) simplifies sharing by treating all unlabeled data as having zero rewards. Additionally, SRTD (Yoo et al., 2022) generates synthetic transitions to augment datasets, and Ludor (Wang et al., 2024b) combines teacher and student networks to facilitate data sharing, demonstrating the value of task-agnostic information in offline RL. Meta-learning approaches have also been integrated into offline RL to enhance task generalization. For example, MACAW (Mitchell et al., 2021) and SMAC (Pong et al., 2022) combine meta-learning principles with offline RL constraints to improve adaptability across tasks by learning task-agnostic representations that can be fine-tuned for specific tasks. Despite these advancements, existing approaches struggle in urban-centric settings, where data is sparse, spatial-temporal dynamics are intricate, and cross-task reward functions are unavailable. These challenges highlight the need for innovative frameworks capable of leveraging diverse datasets and addressing the unique complexities of urban environments.

A.2 Experimental Settings

In Urban State Compression, we extract all state data and train an Autoencoder incorporating KL divergence for regularization. The model is trained for 2000 epochs with a batch size of 64. In Mask-Empowered Reward Learning, the reward ensemble network consists of multiple parallel reward networks and a TrajMAE. The training process is conducted over 1000 epochs with a batch size of 32. In Adaptive Knowledge Fusion, we apply the IQL (Kostrikov et al., 2021) algorithm for policy learning. Two separate policy models are trained on labeled and relabeled datasets, with their parameters progressively fused via EMA during training. The models are trained for 1000 epochs with a batch size of 64. In the main comparison (Table 1), each driver is the target task with its own sparse labeled data; single-task baselines (IQL, CQL, BCQ, BEAR) use only this labeled data, whereas MORSE and the multi-task data-sharing methods additionally draw on an unlabeled pool of mixed-expertise trajectories from the other drivers. In contrast, Table 4 controls the labeled and unlabeled data quality explicitly.

A.3 Detailed Dataset and Experiment Descriptions

Data Description. We evaluate MORSE using real-world taxi trajectory datasets to learn the passenger-seeking strategies employed by different drivers. Each driver follows a distinct strategy for finding passengers. As all drivers operate under the same urban dynamics but are driven by different reward functions, their passenger-seeking processes naturally constitute separate tasks in the multi-task offline RL setting. We use datasets from two cities: **Shenzhen**, with passenger-seeking trajectories from 50 taxi drivers collected between July 1 and September 30, 2016; and **Chengdu**, with 30 taxi drivers, processed using the identical data type and preprocessing pipeline. Each urban state is a multi-dimensional tensor encoding the features of a driver's current grid cell and its surrounding 5×5 neighborhood at a given time slot. Concretely, it aggregates traffic speed (25),

travel demand (25), traffic volume (25), and waiting time (25) over the neighborhood, distances to 21 categories of Points of Interest (PoIs) (21), and spatiotemporal context including grid position, time slot, and day-of-week (4), forming a 125-dimensional observation that is then compressed to a d-dimensional latent state via Urban State Compression (Section 3.1). At any state, a driver chooses among 18 discrete actions corresponding to movements in different directions. A full specification of the features and preprocessing is provided in the supplementary material.

Passenger-Seeking Simulation. To evaluate offline RL algorithms without costly or unsafe online interaction, we construct a passenger-seeking simulator from each dataset following the model-based offline RL paradigm (Kidambi et al., 2020; Yu et al., 2020). The simulator is an ensemble of 7 dynamics models, each trained independently to predict state transitions, equipped with an out-of-distribution (OOD) termination rule that halts a rollout once it enters unreliable regions of the state–action space, preventing evaluation in poorly supported areas. Importantly, the simulator is trained *entirely separately* from policy learning and is used *only* at evaluation time: no reward or policy information flows from the simulator into training, which rules out leakage between the evaluation environment and the learned policies. All methods are evaluated under the identical simulator to guarantee fair comparison, and each driver is associated with a distinct reward function derived from their realized outcomes and behavioral patterns, consistent with the multi-task formulation. The simulator is released as an anonymized repository for reproducibility, with full construction and validation details provided in the supplementary material.

A.4 Urban State Compression Details

We extract all state data and train an Autoencoder incorporating KL divergence for regularization. The model is trained for 2000 epochs with a batch size of 64, the learning rate is 0.0002. Under offline reinforcement learning in urban environment, the state dimensions include (traffic volume, travel demand, traffic speed, waiting time, and the distance to different Points of Interest (PoIs), etc.) is often hundreds of dimensions, and the dimensionality of the agent's entire trajectory often reaches thousands. So the low-dimensional lightweight state is especially important for TrajMAE with Mask-Empowered Reward Learning, which is supported by our experimental results. The results in Figure 3 also illustrate that in the case of offline reinforcement learning where the state dimension is high, reducing the state dimension through pre-training can effectively improve the model training efficiency. All training and evaluation are conducted on a single server equipped with four NVIDIA RTX A6000 GPUs (48 GB memory each), and the wall-clock times reported in Figure 3 are measured on one such GPU to ensure a fair comparison across methods.

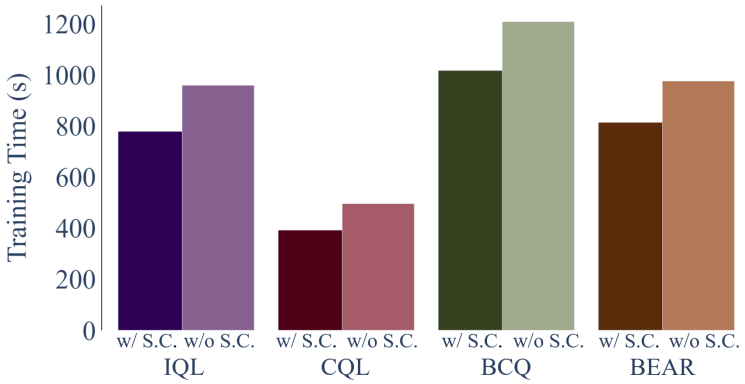


Figure 3: Time efficiency for baseline models with and without working on the latent state space.

A.5 Mask-Empowered Reward Learning Details

Mask-Empowered Reward Learning consists of a reward ensemble network and a trajectory masked autoencoder (TrajMAE). The ensemble contains m parallel reward models, each implemented as an MLP that predicts the reward from (s, a, s').

A.5.1 TRAJECTORY MASKED AUTOENCODER (TRAJMAE)

TrajMAE is a masked autoencoder designed to reconstruct the entire trajectory, including states, actions, and rewards, from a partially masked input trajectory. Each trajectory is represented as a sequence of `(value, type)` tokens, where `type` is `"state"`, `"action"`, or `"reward"`. TrajMAE comprises the following components:

1. Token Embedding:

- **Type Embedding:** An embedding layer maps each of the three token types to a vector.
- **Value Embedding:** A linear layer projects state vectors into the embedding space. Separate linear layers project scalar actions and rewards into the same space.

The embedding of each token is the sum of its type and value embeddings.

2. Transformer Encoder:

- The embedded trajectory is processed by a two-layer Transformer encoder with multi-head self-attention and feed-forward networks, producing contextualized representations for all token positions.

3. Decoder:

- The decoder processes the encoder representations at all token positions. A self-attention module allows these representations to capture dependencies across the trajectory. The resulting representations are decoded into values of the corresponding token types: state vectors, action scalars, and reward scalars.

Training Process:

1. **Masking:** For each trajectory, a proportion $p\%$ of state tokens is randomly selected, and their state values are replaced with zero vectors. Action and reward tokens remain in the input. The resulting trajectory is then embedded using the type and value embeddings.
2. **Forward Pass and Reconstruction:** The Transformer encoder processes the masked input trajectory. The decoder uses the encoder representations to reconstruct the state, action, and reward values at every timestep.
3. **Loss and Backpropagation:** The reconstruction loss combines the mean squared errors for states, actions, and rewards across the trajectory. For labeled trajectories, reward reconstructions are compared with the observed rewards. For unlabeled trajectories, they are compared with the rewards predicted by the ensemble. The parameters are updated using Adam.

TrajMAE is trained for 1000 epochs with a batch size of 32 on both labeled and unlabeled trajectories. For unlabeled trajectories, the reward ensemble first predicts the rewards, which are inserted into the trajectory before masking and reconstruction. As specified by the unlabeled objective in the main text, its reconstruction loss also updates the reward ensemble through these predicted rewards.

A.6 MULTI-TASK OFFLINE RL WITH ADAPTIVE KNOWLEDGE FUSION DETAILS

In Adaptive Knowledge Fusion, we apply the IQL (Kostrikov et al., 2021) algorithm for policy learning. Two separate policy models are trained on labeled and relabeled datasets, with their parameters progressively fused via EMA during training. The models are trained for 1000 epochs with a batch size of 64. Two models are maintained in parallel. Each model consists of three networks: Q-network, Value network and Policy network. Both models share the same network architectures but are trained on different data. In each epoch, one model (Model1) is trained on the original labeled dataset $\mathcal{D}^{\mathrm{L}}$, while the other model (Model2) is trained on the relabeled dataset $\mathcal{D}^{\text{relabeled}}$. Both models are optimized using the Adam optimizer with a learning rate of 1×10^{-3} and employ gradient clipping to ensure training stability.

EMA-Based fusion is applied to the Policy network parameters, ensuring that the reliable but limited π^{labeled} progressively incorporates the broader generalization capabilities of $\pi^{\text{relabeled}}$.

A.7 PASSENGER-SEEKING SIMULATION AND DATA SELECTION DETAILS

For Shenzhen, we use data from 20 of the 50 taxi drivers to construct the multi-task offline RL dataset, while data from all 50 drivers are used to train the environment simulator for evaluating the policies learned by MORSE and all baselines. For Chengdu, we follow the same procedure, using 20 of the 30 drivers for the offline RL dataset and all 30 drivers to train the simulator.

Specifically, the model is an ensemble model that contains seven sub-models, each containing two models for predicting the next state and the forward, respectively, based on (s, a). Because the data we used to train the ensemble model contains all the transitions in the dataset, which is several times more than the amount of data used to train the MORSE and other baseline models, the ensemble model can better simulate the environment. Of course, for complex urban environments, it is difficult to cover all the states with the data we have, so when the difference between the predictions of the seven dynamic models for the next state is too large, we consider that there is an undetected state and enter the termination state. In our experiments we have tried several plans based simulation environment scenarios, including trying different numbers of sub-models, merging models that predict rewards and states in sub-models, trying multiple thresholds to define unseen states, and we have also tested training multiple sub-models independently instead of jointly training them by ensemble, etc. Eventually we got the current MDP construction scheme.

Our simulation environment (MDP) is trained on the original states and for testing purposes, offline RL algorithms such as ablation study MORSE w/o S.C. or other offline RL algorithms that don't require State Compression, can apply their policies directly to the simulation environment. For policies generated by MORSE, it is necessary to perform State Compression on the state generated by the simulation environment before extracting the action.

A.8 DRIVER-SPECIFIC REWARD CONSTRUCTION

Each taxi driver is treated as a separate task with a distinct reward function. We construct rewards for complete passenger-seeking trajectory segments before assigning rewards to individual transitions. Let $\tau_{i,j}$ denote the j-th passenger-seeking segment of driver i, ending when a passenger is found. Its realized outcome includes the distance traveled while carrying that passenger, which serves as a proxy for trip income. We also estimate driver-specific spatial and temporal preferences from statistics of the regions and time periods in which that driver appears. Let $\Phi_i(\tau_{i,j})$ denote the resulting preference score. The total reward for the segment is computed as

$$G_{i,j} = F_i\big(d_{i,j}^{\mathrm{occ}}, \Phi_i(\tau_{i,j}); z\big)\,, \tag{9}$$

where $d_{i,j}^{\mathrm{occ}}$ is the realized occupied-trip distance, F_i combines the income proxy and preference weighting, and z denotes the random seed used for the stochastic component of reward simulation. Thus, the reward assigned to a segment depends on both its realized outcome and the behavioral preferences of its driver.

We distribute $G_{i,j}$ over the transitions in $\tau_{i,j}$ using nonnegative weights that increase toward the end of the passenger-seeking segment. For a segment with transitions $t = 0, \ldots, T_{i,j}$, the transition reward is

$$r_{i,j,t} = \frac{w_{i,j,t}}{\sum_{k=0}^{T_{i,j}} w_{i,j,k}}\, G_{i,j}, \qquad w_{i,j,T_{i,j}} > \cdots > w_{i,j,0} \geq 0. \tag{10}$$

Consequently, the transition at which the passenger is found receives the largest share, and the assigned shares decay progressively backward through the segment. The transition rewards sum to the original segment reward: $\sum_{t=0}^{T_{i,j}} r_{i,j,t} = G_{i,j}$.

For each target driver, only that driver's revealed training rewards are used for reward learning; rewards from the other drivers are treated as unknown. The reward prediction MAE and RMSE in Table 2 are computed on a held-out test split of the target driver's labeled transitions.

A.9 BASELINES DETAILS

- *IQL*. Since we adopt Implicit Q-Learning (IQL) as the offline RL algorithm for MORSE, we compare it against three baselines: (1) training IQL on labeled data only (**No Sharing + IQL**), (2) training IQL with direct sharing of all unlabeled data (**Sharing all + IQL**), (3) training IQL

on labeled data along with relabeled data, corresponding to an ablation study of MORSE without Knowledge Fusion.
For all these experiments, each IQL model is trained with a learning rate of 10^{-3}, a discount factor of 0.99, and an expectile parameter τ set to 0.7. The training runs for a total of 1000 epochs, with a batch size of 64.

- *CQL*. This model is trained with a learning rate of 10^{-3}, a discount factor of 0.99, and a regularization coefficient α set to 0.1. The expectile parameter τ for the target Q-network update is set to 0.005. The training runs for a total of 1000 epochs, with a batch size of 64.
- *BCQ*. This model is trained with a learning rate of 10^{-3}, a discount factor of 0.99, and a target network update parameter τ set to 0.005. The training runs for a total of 1000 epochs, with a batch size of 64. The VAE latent dimension is set to 32, and the perturbation network applies a maximum action adjustment of 0.05.
- *BEAR*. This model is trained with a learning rate of 10^{-3}, a discount factor of 0.99, and a target network update parameter τ set to 0.005. The KL regularization coefficient α is fixed at 0.1. The training runs for a total of 1000 epochs, with a batch size of 64.
- *CDS*. The Conservative Q-Network is trained with a learning rate of 10^{-3}, a discount factor of 0.99, and a target network update parameter τ set to 0.005. The conservative Q-learning regularization coefficient α is fixed at 0.1. Additionally, the Conservative Data Sharing threshold percentile is set to 90%.
- *MODA*. We trained Contrastive data sharing model We trained the model with a batch size of 32, a learning rate of 10^{-3}, and for 1000 epochs. The margin parameter is set to 2. Sliding window step length is set to 1. In GAN, we use batch size of 32 and trained over 1000 epochs. The learning rates for the generator and discriminator are 0.0005 and 0.0002. For Dynamics model, The learning rate is set to 0.02, with a batch size of 32, and the model is trained for 2000 epochs.
- *SRTD*. The Wasserstein Autoencoder (WAE) model is trained with a latent dimension of 32 for skill representation and a hidden layer size of 256 in the MLP architecture. The model is optimized using the Adam optimizer with a learning rate of 10^{-3}. The training process runs for 500 epochs with a batch size of 64. Reconstruction loss is weighted by $\lambda_{\text{rec}} = 1.0$ and a Wasserstein regularization term is $\lambda_{\text{wasserstein}} = 10.0$. For Skill Embedding. The skill embedding process uses a sub-trajectory length of 5, with a skill similarity threshold set to 0.8. For data augmentation, 3,000 imaginary demonstrations are generated using a skill interpolation factor of 0.7.
- *Ludor*. The model is trained with a batch size of 64, a learning rate of 3×10^{-4}, and a discount factor of 0.99. The expectile regression parameter τ is set to 0.7. The teacher model is pre-trained for 1000 epochs using behavior cloning, and the student model is trained for 1000 epochs using the IQL framework. The EMA update coefficient for knowledge transfer is set to 0.7.
- *HarmoDT*. HarmoDT is a multi-task offline RL method that learns a task-specific binary mask over the parameters of a shared Decision Transformer, isolating a "harmony subspace" for each task so as to alleviate gradient conflict and negative transfer across tasks; the per-task masks and the shared weights are optimized in a bi-level fashion. We implement it on a Decision Transformer backbone with a context length of 20, an embedding dimension of 128, 3 transformer layers, and 1 attention head, using a return-to-go conditioning consistent with the other tasks. The model is optimized with Adam at a learning rate of 10^{-4}, a batch size of 64, and a dropout rate of 0.1, and is trained for 1000 epochs. The task masks are updated in the outer loop every 5 epochs with a mask sparsity ratio of 0.5. As HarmoDT operates on full trajectories rather than on a relabeled transition buffer, it is trained on the labeled trajectories of all tasks jointly.

A.10 Detailed Algorithm

Algorithm 1 summarizes the three stages of MORSE. For each target task, the unlabeled dataset contains trajectories from the other tasks.

Algorithm 1 MORSE

Input: Target tasks $\mathcal{I}$; labeled trajectories $\mathcal{D}_i^{\mathrm{L}}$ and unlabeled trajectories $\mathcal{D}_i^{\mathrm{U}}$ for each $i \in \mathcal{I}$; ensemble size m, masking ratio p, EMA coefficient α, and fusion interval η.
Output: Target policies $\{\pi_{\omega_i}^{\mathrm{labeled}}\}_{i\in\mathcal{I}}$.

Stage 1: Urban State Compression
repeat
  Sample original states S from the datasets of all tasks.
  Update E_ζ and D_ξ using Eq. (1).
until the state-compression training budget is reached
Replace every S and S' in the datasets with $s = E_\zeta(S)$ and $s' = E_\zeta(S')$.
for each target task $i \in \mathcal{I}$ **do**
  Stage 2: Mask-Empowered Reward Learning
  Initialize TrajMAE $(\tilde{E}_\theta, \tilde{D}_\phi)$ and an ensemble $\{R_{\psi_k}^{(i)}\}_{k=1}^m$.
  repeat
    Sample labeled trajectories from $\mathcal{D}_i^{\mathrm{L}}$ and unlabeled trajectories from $\mathcal{D}_i^{\mathrm{U}}$.
    Predict rewards $\hat{r}_t$ for the unlabeled transitions using the trimmed ensemble mean in Eq. (2).
    Mask $p\%$ of the state tokens in each trajectory, retaining its action and reward tokens.
    Form masked labeled trajectories with observed rewards r_t and masked unlabeled trajectories with predicted rewards $\hat{r}_t$.
    Reconstruct the state, action, and reward tokens at every timestep using $\tilde{D}_\phi \circ \tilde{E}_\theta$.
    Compute the supervised ensemble loss on labeled transitions and both trajectory reconstruction losses.
    Update $(\tilde{E}_\theta, \tilde{D}_\phi)$ and $\{R_{\psi_k}^{(i)}\}_{k=1}^m$ using the total loss in Eq. (7).
  until the reward-learning training budget is reached
  Create $\mathcal{D}_i^{\mathrm{relabeled}}$ by assigning ensemble-predicted rewards to $\mathcal{D}_i^{\mathrm{U}}$.
  Stage 3: IQL with Adaptive Knowledge Fusion
  Initialize separate IQL learners $(Q_i^{\mathrm{L}}, V_i^{\mathrm{L}}, \pi_{\omega_i}^{\mathrm{labeled}})$ and $(Q_i^{\mathrm{R}}, V_i^{\mathrm{R}}, \pi_{\sigma_i}^{\mathrm{relabeled}})$.
  Set the training-step counter $t \leftarrow 0$.
  repeat
    Sample a transition batch from $\mathcal{D}_i^{\mathrm{L}}$ and update $(Q_i^{\mathrm{L}}, V_i^{\mathrm{L}}, \pi_{\omega_i}^{\mathrm{labeled}})$ with IQL (Kostrikov et al., 2021).
    Sample a transition batch from $\mathcal{D}_i^{\mathrm{relabeled}}$ and update $(Q_i^{\mathrm{R}}, V_i^{\mathrm{R}}, \pi_{\sigma_i}^{\mathrm{relabeled}})$ with IQL.
    $t \leftarrow t + 1$.
    if $t \bmod \eta = 0$ **then**
      $\omega_i \leftarrow \alpha\omega_i + (1-\alpha)\sigma_i$ *(policy parameters only).*
    end if
  until the offline-RL training budget is reached
end for