# CONTROLLABLE CROWD GENERATION THROUGH WORLD-MODEL PLANNING

**Anonymous authors**
Paper under double-blind review

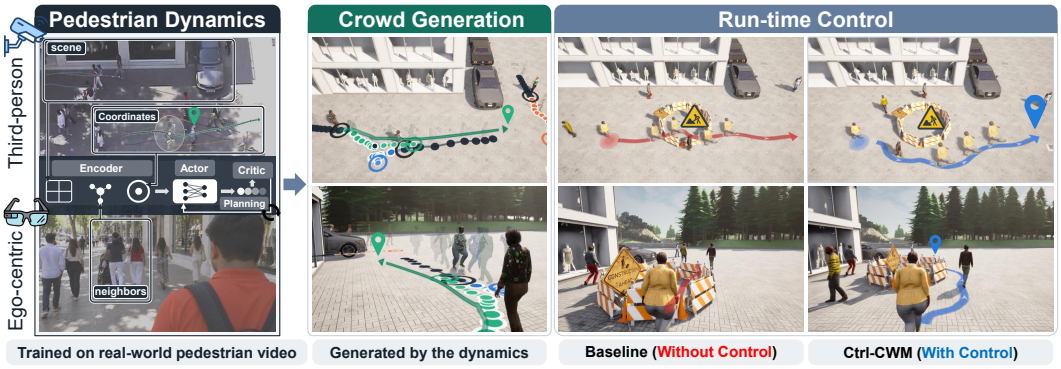



Figure 1: **Ctrl-CWM.** Human dynamics learned from real-world pedestrian video **(left)** are used for continuous crowd generation **(center)** and runtime crowd control **(right)**. User-defined objectives can be introduced at run time without retraining the model.

## ABSTRACT

Crowd simulation plays a central role in robot navigation, autonomous driving, and urban planning. For these applications, realistic simulation requires crowds to adapt their behavior to environmental changes and user objectives. However, existing methods that rely on predefined control settings have limited flexibility in accommodating new user-specified objectives. To address this limitation, we propose **Ctrl-CWM**, a multi-agent **C**on**tr**o**l**lable **C**rowd **W**orld **M**odel that integrates crowd generation and run-time control. Our key idea is to adapt the world-model principle of planning using imagined futures to crowd simulation. To this end, Ctrl-CWM consists of an encoder that learns a representation of human motion dynamics, an actor that proposes pedestrian displacements, a critic that evaluates imagined crowd trajectories, and a planner that selects actions. We first learn human motion dynamics through trajectory prediction on real-world pedestrian videos and then freeze the encoder to preserve them. Using this representation, the actor generates imagined crowd trajectories through repeated state updates, and the planner combines the critic's scores with user costs to select actions. Repeated planning advances the simulated crowd, while additional user costs introduce new control objectives without retraining. We extensively evaluate crowd generation under varied agent arrival conditions and run-time control across avoidance and attraction scenarios. Ctrl-CWM outperforms the state-of-the-art method on most crowd realism and collision metrics, and adapts crowd behaviors to user-specified objectives introduced during simulation.

## 1 INTRODUCTION

Crowd simulation is important for urban planning as well as for developing autonomous vehicles and social robots. Training robots to navigate reliably and evaluating urban designs require simulations that reproduce human crowd behaviors, not just scenes populated with moving pedestrians. In the real world, the environment changes over time and people adjust their motions in response. For example, when a construction site blocks a walkway, pedestrians take another route rather than



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.

continue along their original paths. Therefore, a realistic crowd simulator should capture not only how people move but also how they adjust their motion when conditions change. To this end, we aim to generate run-time controllable crowd motion that responds to user-specified objectives while preserving human motion dynamics.

Traditional crowd simulators use hand-designed rules for navigation and interactions (Reynolds, 1987; Helbing & Molnar, 1995; Van Den Berg et al., 2010). These rules make control straightforward, but they do not directly learn human motion dynamics from real-world pedestrian videos. Learning-based approaches capture human motion dynamics from pedestrian data (Charalambous et al., 2023; Rempe et al., 2023; Bae et al., 2025) and support control over generated crowd motion (Rempe et al., 2023; Bae et al., 2025). However, their control is limited to a predefined population and agent parameters (Bae et al., 2025) or to gradient guidance and sample filtering (Rempe et al., 2023). TRACE (Rempe et al., 2023) also reports trajectory sampling times of $1 \sim 3$ seconds for a single pedestrian, which can delay responses to control during simulation. We aim to develop a crowd model that retains learned human motion dynamics while supporting run-time control.

Recent advances in world models (Hafner et al., 2025; Hansen et al., 2024) have shown promise for learning and controlling behavior across different tasks. In approaches such as Dreamer (Hafner et al., 2019a), an encoder maps observations to latent states and an actor proposes actions. A dynamics model predicts the next state and reward from the current state and action, producing imagined futures through repeated prediction. A critic estimates the expected return, and both the actor and critic can learn from imagined experience. For planning, a planner can evaluate candidate action sequences using predicted rewards and a terminal value from the critic (Hansen et al., 2022; 2024). The planner iteratively refines the sampling distribution toward higher-scoring candidates, executes only the first action, and replans from the updated state. These ideas motivate our use of learned pedestrian dynamics to imagine and evaluate crowd futures for generation and run-time control.

We propose Ctrl-CWM, a multi-agent Controllable Crowd World Model for crowd generation and run-time control (Fig. 1). Tasks where world models have been applied, such as video games and robot control, specify the observations and the available actions, and use rewards or goals to evaluate outcomes. Pedestrian videos instead record how people moved but do not define actions or specify which actions meet new user objectives. Therefore, we must define the crowd state and pedestrian actions and train a critic that evaluates the resulting motion for planning. In this paper, we define the state as the scene and each pedestrian's position, recent motion, and local goal, and an action as a pedestrian's next-step displacement. Human trajectory prediction provides a basis for learning how pedestrians move and interact with their surroundings (Alahi et al., 2016; Gupta et al., 2018; Sadeghian et al., 2019). We learn a shared representation through trajectory prediction and then freeze the encoder to preserve the learned human motion dynamics. A single actor shared across all pedestrians learns to propose each pedestrian's next action from this representation.

Applying these actions updates the explicit crowd state, and repeating these updates generates imagined crowd trajectories. Our critic learns to rank these imagined crowd trajectories using supervision derived from ground-truth trajectories.

At run time, the actor proposes displacements as during training. For a controlled pedestrian, the model imagines the crowd trajectories that follow several candidate displacements, while the actor generates the actions of all other pedestrians. The critic ranks these trajectories by their quality, and a user cost measures how well they satisfy the current user-specified objective, such as avoiding or approaching a specified region. A planner combines the critic's ranking with the user cost and selects the displacement to execute. Repeating this process advances the crowd, and the critic keeps the motion realistic while the user cost steers it toward the objective. By planning with the critic in imagination, Ctrl-CWM performs crowd generation and run-time control with a single model.

To demonstrate the effectiveness of Ctrl-CWM, we construct run-time control scenarios with objectives introduced during simulation. Generated crowds are directed away from an avoidance area such as construction or toward an attraction such as a fountain. As a crowd generator, Ctrl-CWM outperforms the state-of-the-art method on most crowd realism and collision metrics. For avoidance objectives, it achieves the highest average compliance among the related methods across the tested region configurations. Additionally, qualitative results demonstrate run-time control in reconstructed real-world scenes beyond the benchmark. Finally, competitive trajectory prediction results indicate that the shared representation preserves human motion dynamics.

# 2 Related Work

## 2.1 Human Trajectory Prediction

Human trajectory prediction has evolved to capture social interaction, scene context, destination intent, and uncertainty in future motion. Social-LSTM (Alahi et al., 2016) captures dependencies among neighboring pedestrians, whereas Social-GAN (Gupta et al., 2018) introduces multimodal forecasting. Subsequent methods encode social structure through graphs (Mohamed et al., 2020; Shi et al., 2021), groups (Sun et al., 2020; Zhou et al., 2022), hypergraphs (Xu et al., 2022), and agent-aware attention (Yuan et al., 2021; Shi et al., 2023), while other approaches incorporate learnable social-physics formulations (Yue et al., 2022). Environmental context is represented through visual features (Sadeghian et al., 2019) or semantic maps (Mangalam et al., 2021), with intent captured by sampled endpoints (Mangalam et al., 2020), expert examples (Zhao & Wildes, 2021), or goal and waypoint heatmaps (Mangalam et al., 2021). Future uncertainty has also been addressed using conditional VAEs (Salzmann et al., 2020), diffusion (Gu et al., 2022; Mao et al., 2023; Li et al., 2023), flow-based methods (Maeda & Ukita, 2023; Fu et al., 2025), and, more recently, language models (Bae et al., 2024; Kim et al., 2025; Xu et al., 2026). We build on these advances to learn an agent-centric representation through trajectory prediction and use it within a world model for crowd generation and runtime control.

## 2.2 Crowd Simulation and Generation

Classical crowd simulators model local interactions through flocking (Reynolds, 1987), social forces (Helbing & Molnar, 1995), and reciprocal collision avoidance (Van Den Berg et al., 2010). Continuum methods (Treuille et al., 2006) combine global navigation with crowd flow. Data-driven approaches (Lee et al., 2007; Lerner et al., 2007; Ju et al., 2010) select or blend observed motions. Later methods (Lee et al., 2018) learn navigation policies through reinforcement learning. GREIL-Crowds (Charalambous et al., 2023) derives rewards from real crowd examples to learn pedestrian behaviors. SPDiff (Chen et al., 2024) combines social-physics information with diffusion and roll-out training for crowd simulation. CEDRL (Panayiotou et al., 2025) learns diverse behaviors from multiple crowd datasets and provides control over behavioral complexity.

User control has also been studied through interactive editing of existing crowd animations (Kim et al., 2014). TRACE (Rempe et al., 2023) supports diffusion guidance and replanning and can use values learned by a physics-based animation controller. Its filtering variant selects among unguided samples using a guidance loss. CrowdES (Bae et al., 2025) combines continuous agent emission with learned behavior switching and exposes population and agent controls. Text-guided methods (Ji et al., 2024) generate crowd distribution and velocity fields from descriptions.

## 2.3 World Models and Planning

World models (Ha & Schmidhuber, 2018; Hafner et al., 2019a; 2025) learn environment dynamics for policy learning and online planning (Hafner et al., 2019b; Hansen et al., 2022). Planning may operate in a learned latent space (Hansen et al., 2022; 2024) or over explicit states (Chua et al., 2018), commonly through sampling-based model predictive control. Short model rollouts (Janner et al., 2019) help reduce accumulated prediction errors. Learned value estimates (Hansen et al., 2022) can account for outcomes beyond the planning horizon. Recent world models (Zhou et al., 2024; Assran et al., 2025) use pretrained representations and offline trajectories to support visual-goal planning without task-specific rewards. Multi-agent extensions (Egorov & Shpilman, 2022) also model interactions among decision-making agents.

Ctrl-CWM builds an action-conditioned crowd transition on a representation learned through trajectory prediction. A critic trained from ground-truth behavior evaluates imagined futures for generation. Adding user costs to the same planner enables runtime control.

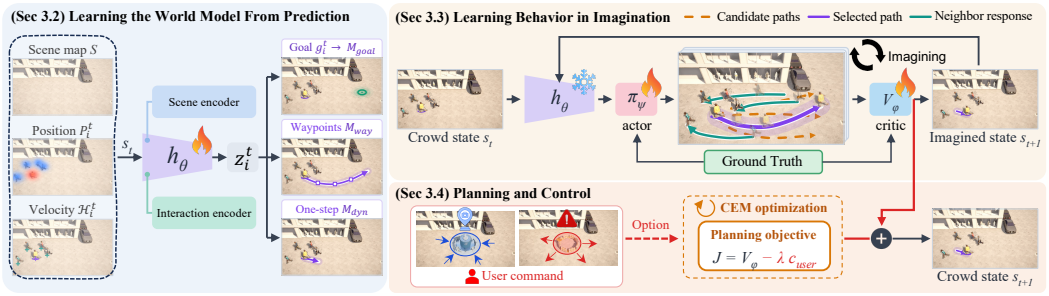



Figure 2: **Overview of Ctrl-CWM.** Trajectory prediction trains the state encoder $h_\theta$ on real-world pedestrian data, which is then frozen. The actor $\pi_\psi$ and critic $V_\phi$ are learned over imagined rollouts on this representation, and CEM adds a user cost to the critic score for run-time control.

# 3 Method

## 3.1 Problem Formulation

Building on human dynamics learned through trajectory prediction, we formulate a multi-agent world model for controllable crowd generation. We define its state, actions, and model components, which enable imagination-based planning and runtime control.

Each pedestrian is an agent. At time $t$, the crowd state is $s_t = \big(S, \{(p_i^t, \mathcal{H}_i^t, g_i^t)\}_{i\in\mathcal{A}_t}\big)$, where $S$ represents the static scene representation and $\mathcal{A}_t$ denotes the set of agents present at time $t$. For agent $i$, $p_i^t \in \mathbb{R}^2$ is its position, $\mathcal{H}_i^t$ is its recent velocity history, and $g_i^t \in \mathbb{R}^2$ denotes its local navigation goal. All positions are expressed in scene-grid coordinates. An action $a_i^t \in \mathbb{R}^2$ specifies a one-step displacement. Goals remain fixed within each $L$-step window and are updated for the next window. We use $L = 12$ as the prediction and behavior-learning horizon.

Let $\hat{s}_{t+1:t+H}$ denote a sequence of $H$ imagined future crowd states, where $H < L$ is the planning horizon. Ctrl-CWM uses a feature extractor $h_\theta$, an actor $\pi_\psi$, and a critic $V_\phi$, with parameters $\theta$, $\psi$, and $\phi$, respectively:

$$\begin{aligned} &\text{Feature extractor} && z_i^t = h_\theta(s_t)_i && \triangleright \text{ agent-centric representation} \\ &\text{Actor} && \pi_\psi(z_i^t, g_i^t, v_i^t) && \triangleright \text{ one-step displacement proposal} \\ &\text{Critic} && V_\phi(\hat{s}_{t+1:t+H}, i) && \triangleright \text{ agent-level rollout score} \end{aligned} \tag{1}$$

The critic scores agent $i$'s motion within the imagined crowd-state sequence. The actor and critic are shared across agents.

At each transition, $\mathcal{C}_t \subseteq \mathcal{A}_t$ denotes the agents whose actions are selected by the planner, collected as $a_{\mathcal{C}_t}^t = \{a_i^t\}_{i\in\mathcal{C}_t}$. All remaining agents follow the actor proposal $\bar{a}_i^t = \pi_\psi(z_i^t, g_i^t, v_i^t)$. The planner selects $a_i^t$ by evaluating candidate displacements through imagined rollouts and optimizing the planning objective (Sec. 3.4).

The crowd transition $\mathcal{T}$ updates each agent's position using its assigned displacement:

$$s_{t+1} = \mathcal{T}(s_t, a_{\mathcal{C}_t}^t), \qquad p_i^{t+1} = p_i^t + \begin{cases} a_i^t, & i \in \mathcal{C}_t, \\ \bar{a}_i^t, & i \notin \mathcal{C}_t. \end{cases} \tag{2}$$

Velocity histories are updated from the resulting positions.

## 3.2 Learning the World Model from Prediction

Crowd simulation requires modeling how pedestrians move and interact with the scene and nearby agents. We learn these human motion dynamics through trajectory prediction on real-world data. We adopt the scene-aware, goal-conditioned design of Y-Net (Mangalam et al., 2021) to provide spatial features and local goal predictions for simulation.

**Agent-centric representation.** The feature extractor $h_\theta$ combines scene and interaction cues to form the agent-centric representation $z_i^t$ (Fig. 2, left). The scene encoder adopts a U-Net architecture

to extract features from the scene map $S$ together with Gaussian heatmaps of the target agent and its neighbors on a $G \times G$ grid, where $G$ is the grid side length.

In parallel, the interaction encoder captures social and motion context using an attention module and a gated recurrent unit (GRU) (Cho et al., 2014). The attention module models interactions among agents using position embeddings and pairwise-displacement biases, while the GRU encodes each agent's velocity history $\mathcal{H}_i^t$. Their outputs are concatenated and projected into an interaction embedding. This embedding modulates the U-Net bottleneck through feature-wise linear modulation (FiLM) (Perez et al., 2018). The modulated bottleneck, U-Net skip features, and interaction embedding together form the representation $z_i^t$.

**Prediction heads.** To capture both navigation intent and local motion dynamics, we use three prediction heads: goal, waypoint, and dynamics. The goal decoder predicts the endpoint of the $L$-step prediction window as a local navigation goal using a goal heatmap $M_{\text{goal}}$. This goal represents the target for the current window rather than the pedestrian's final destination.

Conditioned on the ground-truth goal during training and the predicted goal at inference, a shared decoder predicts $L$ waypoint heatmaps $\{M_{\text{way},j}\}_{j=1}^{L}$. An auxiliary head predicts the next-position heatmap $M_{\text{dyn}}$ for one-step supervision but does not advance rollouts.

During training, the waypoint and dynamics heads are conditioned on the ground-truth goal, while the waypoint decoder uses the predicted goal at inference time. For crowd simulation, a coordinate extracted from the predicted goal heatmap serves as the local navigation goal $g_i^t$.

**Objective.** We jointly train $h_\theta$ and the prediction heads using three prediction objectives. Let $p_j^*$ denote the ground-truth position at future step $j$, where $j = 1, \dots, L$. The target heatmaps $M_{\text{goal}}^*$, $M_{\text{way},j}^*$, and $M_{\text{dyn}}^*$ are Gaussians centered at $p_L^*$, $p_j^*$, and $p_1^*$, respectively. The training objective is

$$\begin{aligned}\mathcal{L}_{\text{wm}} = \; & \text{BCE}(M_{\text{goal}}, M_{\text{goal}}^*) + \left\|\text{softargmax}(M_{\text{goal}}) - p_L^*\right\|_2 \\ & + \frac{\lambda_{\text{way}}}{L} \sum_{j=1}^{L} \Big[\text{BCE}(M_{\text{way},j}, M_{\text{way},j}^*) + \left\|\text{softargmax}(M_{\text{way},j}) - p_j^*\right\|_2\Big] \\ & + \lambda_{\text{dyn}} \, \text{BCE}(M_{\text{dyn}}, M_{\text{dyn}}^*).\end{aligned} \tag{3}$$

Here, BCE denotes binary cross-entropy with positive-class weighting, and softargmax extracts a coordinate from a predicted heatmap. We empirically set the loss weights to $\lambda_{\text{way}} = 1.0$ and $\lambda_{\text{dyn}} = 0.5$.

### 3.3 Learning Behavior in Imagination

After prediction training, we freeze $h_\theta$ to preserve the learned human motion dynamics. The actor learns to generate crowd motion by predicting pedestrian actions within the learned dynamics model. The critic learns to rank imagined crowd trajectories based on their motion quality for planning.

**Actor.** The actor uses a multilayer perceptron (MLP) with two hidden layers to predict each pedestrian's next displacement from the agent-centric representation $z_i^t$, the goal direction, and the current velocity normalized by $G$. We train it through $L$-step rollouts of $\mathcal{T}$, with all agents following the actor. At each step, predicted positions update the crowd state, which is re-encoded for the next prediction. Training goals are nearby waypoints sampled from ground-truth trajectories.

Let $\tau = (\hat{p}_1, \dots, \hat{p}_L)$ and $\tau^* = (p_1^*, \dots, p_L^*)$ denote an agent's generated and ground-truth trajectories. The imitation loss combines pointwise position error with soft-DTW (Cuturi & Blondel, 2017):

$$\mathcal{L}_{\text{imit}}(\tau, \tau^*) = \lambda_{\text{pos}} \frac{1}{LG} \sum_{j=1}^{L} \|\hat{p}_j - p_j^*\|_2 + \lambda_{\text{dtw}} \, \text{soft-DTW}_\gamma(\tau, \tau^*). \tag{4}$$

Here, $\gamma > 0$ is the soft-DTW smoothing parameter. The weights $\lambda_{\text{pos}}$ and $\lambda_{\text{dtw}}$ control the pointwise position and soft-DTW terms, respectively.

The actor additionally matches speed profiles and their successive changes to the ground-truth trajectory. Let $v_j$ and $\hat{v}_j$ denote ground-truth and generated speeds, respectively.

$$\mathcal{L}_{\text{kin}} = \frac{1}{L-1}\sum_{j=1}^{L-1}|\hat{v}_j - v_j| + \frac{1}{L-2}\sum_{j=1}^{L-2}|(\hat{v}_{j+1} - \hat{v}_j) - (v_{j+1} - v_j)|, \tag{5}$$

where $v_j = \|p^*_{j+1} - p^*_j\|_2/G$ and $\hat{v}_j = \|\hat{p}_{j+1} - \hat{p}_j\|_2/G$.

Let $\Delta\tau$ and $\Delta\tau^*$ denote the corresponding velocity sequences. A GRU discriminator $D$ distinguishes generated velocity sequences from ground-truth velocity sequences. The actor and discriminator are optimized with least-squares adversarial objectives (Mao et al., 2017):

$$\begin{aligned}\mathcal{L}_{\text{act}} &= \mathcal{L}_{\text{imit}}(\tau, \tau^*) + \lambda_{\text{kin}}\mathcal{L}_{\text{kin}} + \lambda_{\text{adv}}(D(\Delta\tau) - 1)^2,\\ \mathcal{L}_D &= (D(\Delta\tau^*) - 1)^2 + D(\Delta\tau)^2.\end{aligned} \tag{6}$$

Here, $\mathcal{L}_D$ denotes the discriminator objective, while $\lambda_{\text{kin}}$ and $\lambda_{\text{adv}}$ weight the kinematic and adversarial terms in the actor objective, respectively.

**Critic.** The critic learns to rank imagined crowd trajectories generated by candidate actions. For each training state and agent $i$, we sample $K$ candidate displacements around the actor proposal. Each candidate replaces agent $i$'s first-step displacement, while the actor supplies all remaining actions throughout the $L$-step rollout.

For candidate $k$, let $\tau_i^{(k)}$ denote agent $i$'s trajectory and $\hat{s}^{(k)}_{t+1:t+H}$ denote the first $H$ imagined crowd states. We define a cost for each candidate:

$$y_i^{(k)} = \mathcal{L}_{\text{imit}}(\tau_i^{(k)}, \tau_i^*) + \lambda^T_{\text{kin}}\mathcal{L}_{\text{kin}}(\tau_i^{(k)}, \tau_i^*) + \lambda_{\text{coll}}C_i^{(k)}. \tag{7}$$

The collision term is computed as:

$$C_i^{(k)} = \frac{1}{L}\sum_{t=1}^{L}\sum_{j\neq i}\mathbf{1}[\|\hat{p}_{i,t}^{(k)} - \hat{p}_{j,t}^{(k)}\|_2 < r]. \tag{8}$$

The weights $\lambda^T_{\text{kin}}$ and $\lambda_{\text{coll}}$ control the kinematic and collision penalties, respectively.

A GRU followed by an MLP processes agent $i$'s velocity, speed, nearest-neighbor distance, goal distance, and velocity-goal alignment over the first $H$ steps to estimate $V_\phi(\hat{s}^{(k)}_{t+1:t+H}, i)$. Let $\mathbf{y}$ and $\mathbf{q}$ denote candidate costs and critic scores. The critic is trained to assign higher scores to lower-cost candidates using a listwise objective:

$$\mathcal{L}_{\text{critic}} = -\sum_{k=1}^{K}\text{softmax}_k(-\mathbf{y}/\tau_c)\log\text{softmax}_k(\mathbf{q}/\tau_c). \tag{9}$$

At inference, the critic evaluates imagined rollouts without future ground-truth trajectories.

### 3.4 Planning and Control

At each simulation step, the planner selects the next displacement for each controlled agent $i \in \mathcal{C}_t$ using the cross-entropy method (CEM) (Fig. 2, lower right). CEM initially samples $K$ candidate displacements from $\mathcal{N}(\bar{a}_i^t, \sigma^2_{\text{plan}}\mathbf{I}_2)$. Each candidate is evaluated through an $H$-step imagined rollout.

Each candidate $k$ is scored by

$$J^{(k)} = V_\phi(\hat{s}^{(k)}_{t+1:t+H}, i) - \lambda_{\text{user}}c_{\text{user}}(\hat{s}^{(k)}_{t+1:t+H}), \tag{10}$$

where the critic evaluates behavioral quality and $c_{\text{user}}$ specifies a task-dependent spatial objective. The avoidance and attraction costs used in our experiments are defined in Appendix D.

CEM iteratively refines the sampling distribution toward higher-scoring candidates. After $I$ iterations, its final mean is selected as $a_i^t$. Once actions are selected for all controlled agents, the crowd advances according to Eq. 2, and planning repeats from the updated state. Setting $\lambda_{\text{user}} = 0$ yields critic-guided crowd generation.

Table 1: **Crowd behavior generation under shared arrivals.** Results use identical replayed random-surface (left) and diffusion (right) arrivals across methods. AVG averages five ETH–UCY folds, SDD, and GCS. Lower is better except Div.; Col. is in %. **Bold**/underline mark the best/second-best distinct scores.

*Arrivals from the RS emitter*

| Dataset/Fold | Model | Scene-Level Realism | | | | Agent-Level Accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Dens.↓ | Freq.↓ | Cov.↓ | Pop.↓ | Kinem.↓ | DTW↓ | Div.↑ | Col.(%)↓ |
| ETH | ORCA | **0.019** | **0.016** | **0.016** | **0.205** | 0.839 | 2.848 | 0.092 | **0.004** |
| | CrowdES | 0.044 | 0.025 | 0.025 | 0.461 | 1.138 | 2.608 | **0.240** | 2.660 |
| | **Ours** | 0.034 | **0.016** | **0.016** | 0.398 | **0.358** | **1.511** | 0.171 | 0.265 |
| HOTEL | ORCA | 0.044 | 0.017 | 0.017 | 0.396 | 0.623 | 1.094 | 0.179 | **0.015** |
| | CrowdES | 0.015 | 0.016 | 0.016 | 0.130 | 0.524 | 0.907 | 0.184 | 2.471 |
| | **Ours** | **0.012** | **0.012** | **0.012** | **0.098** | **0.419** | **0.746** | **0.224** | 0.828 |
| UNIV | ORCA | **0.311** | **0.210** | **0.210** | **0.658** | **0.357** | 1.878 | 0.221 | **0.007** |
| | CrowdES | 0.365 | 0.230 | 0.230 | 0.771 | 0.457 | 1.394 | 0.249 | 2.758 |
| | **Ours** | 0.362 | 0.216 | 0.216 | 0.765 | 0.543 | **1.184** | **0.327** | 0.733 |
| ZARA1 | ORCA | 0.063 | 0.027 | 0.027 | 0.953 | 1.435 | 2.530 | 0.181 | **0.002** |
| | CrowdES | 0.011 | 0.009 | 0.009 | 0.167 | **0.358** | 0.706 | **0.283** | 2.010 |
| | **Ours** | **0.010** | **0.008** | **0.008** | **0.137** | 0.691 | **0.703** | 0.258 | 1.015 |
| ZARA2 | ORCA | 0.023 | 0.010 | 0.010 | 0.247 | 0.742 | 1.358 | 0.110 | **0.015** |
| | CrowdES | **0.020** | **0.008** | **0.008** | **0.191** | **0.340** | 0.580 | **0.333** | 2.983 |
| | **Ours** | 0.026 | 0.012 | 0.012 | 0.258 | 0.581 | **0.569** | 0.306 | 1.457 |
| SDD | ORCA | 0.061 | 0.045 | 0.043 | 0.472 | 0.684 | **5.168** | 0.360 | **0.011** |
| | CrowdES | **0.043** | 0.034 | 0.033 | **0.458** | **0.662** | 6.303 | 0.375 | 1.643 |
| | **Ours** | 0.046 | **0.033** | **0.031** | 0.476 | 0.718 | 5.511 | **0.383** | 0.227 |
| GCS | ORCA | 6.787 | **0.369** | **0.369** | 15.683 | 2.423 | 5.120 | 0.232 | **0.011** |
| | CrowdES | 4.726 | 0.573 | 0.573 | 10.921 | 0.755 | 2.928 | **0.397** | 9.613 |
| | **Ours** | **2.847** | 0.483 | 0.483 | **6.578** | **0.503** | **1.766** | 0.327 | 7.645 |
| **AVG** | ORCA | 1.044 | **0.099** | **0.099** | 2.659 | 1.015 | 2.857 | 0.196 | **0.009** |
| | CrowdES | 0.746 | 0.128 | 0.128 | 1.871 | 0.605 | 2.204 | **0.294** | 3.448 |
| | **Ours** | **0.477** | 0.111 | 0.111 | **1.244** | **0.545** | **1.713** | 0.285 | 1.739 |

*Arrivals from the diffusion emitter*

| Dataset/Fold | Model | Scene-Level Realism | | | | Agent-Level Accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Dens.↓ | Freq.↓ | Cov.↓ | Pop.↓ | Kinem.↓ | DTW↓ | Div.↑ | Col.(%)↓ |
| ETH | ORCA | 0.026 | 0.013 | 0.013 | 0.265 | 0.867 | 3.794 | 0.096 | **0.006** |
| | CrowdES | 0.020 | 0.020 | 0.020 | 0.208 | 0.377 | 1.649 | **0.203** | 0.697 |
| | **Ours** | **0.017** | **0.006** | **0.006** | **0.193** | **0.214** | **1.458** | 0.181 | 0.620 |
| HOTEL | ORCA | 0.022 | 0.010 | 0.010 | 0.199 | 1.052 | 1.588 | 0.112 | **0.013** |
| | CrowdES | 0.013 | **0.009** | **0.009** | 0.117 | 0.336 | 0.643 | 0.242 | 1.197 |
| | **Ours** | **0.011** | **0.009** | **0.009** | **0.087** | **0.305** | **0.548** | **0.259** | 1.046 |
| UNIV | ORCA | 0.348 | 0.214 | 0.214 | 0.738 | 0.732 | 1.884 | 0.221 | **0.014** |
| | CrowdES | 0.347 | 0.204 | 0.204 | 0.734 | **0.420** | **1.121** | 0.340 | 0.645 |
| | **Ours** | **0.339** | **0.196** | **0.196** | **0.717** | 0.436 | **1.121** | **0.349** | 0.609 |
| ZARA1 | ORCA | 0.064 | 0.024 | 0.024 | 0.974 | 1.950 | 2.594 | 0.181 | **0.002** |
| | CrowdES | 0.018 | 0.017 | 0.017 | 0.254 | **0.327** | **0.675** | **0.304** | 1.018 |
| | **Ours** | **0.010** | **0.012** | **0.012** | **0.150** | 0.866 | 0.822 | **0.304** | 0.506 |
| ZARA2 | ORCA | 0.013 | **0.007** | **0.007** | 0.136 | 1.017 | 2.501 | 0.101 | **0.025** |
| | CrowdES | **0.009** | 0.013 | 0.013 | **0.100** | **0.227** | **0.579** | **0.355** | 1.021 |
| | **Ours** | 0.012 | 0.011 | 0.011 | 0.124 | 0.505 | 0.675 | 0.334 | 1.000 |
| SDD | ORCA | 0.052 | 0.036 | 0.032 | **0.421** | 0.674 | **5.674** | 0.311 | **0.392** |
| | CrowdES | **0.038** | 0.033 | **0.030** | 0.463 | **0.650** | 6.352 | **0.354** | 0.411 |
| | **Ours** | 0.044 | **0.031** | **0.030** | 0.462 | 0.971 | 5.993 | 0.326 | 1.140 |
| GCS | ORCA | 6.677 | 0.439 | 0.439 | 15.429 | 2.097 | 4.558 | 0.240 | **0.004** |
| | CrowdES | 0.584 | 0.066 | 0.066 | **0.329** | 1.341 | 3.864 | 0.279 | 8.519 |
| | **Ours** | **0.340** | **0.056** | **0.056** | 0.870 | **1.325** | **1.547** | **0.414** | 6.080 |
| **AVG** | ORCA | 1.029 | 0.106 | 0.106 | 2.595 | 1.198 | 3.228 | 0.180 | **0.065** |
| | CrowdES | 0.147 | 0.052 | 0.051 | **0.315** | **0.525** | 2.126 | 0.297 | 1.930 |
| | **Ours** | **0.110** | **0.046** | **0.046** | 0.372 | 0.660 | **1.738** | **0.310** | 1.572 |

# 4 EXPERIMENTS

We evaluate generation and region control on ETH–UCY (Pellegrini et al., 2009; Lerner et al., 2007), SDD (Robicquet et al., 2016), and GCS (Yi et al., 2015). The main comparisons average seven benchmark entries: five ETH–UCY folds, SDD, and GCS. Component and planning analyses use the five ETH–UCY folds. Prediction-head evaluation separately assesses the pretrained representation. Appendix B.2 distinguishes training costs, generation metrics, and control compliance.

## 4.1 CROWD BEHAVIOR GENERATION

We compare ORCA (Van Den Berg et al., 2010), CrowdES (Bae et al., 2025), and Ctrl-CWM on the CrowdES benchmark for scene-level realism and agent-level accuracy. Here, an emitter determines when and where pedestrians enter. We use random-surface and diffusion emitters, replaying identical entries across methods to isolate motion generation.

As shown in Table 1, ORCA explicitly optimizes collision avoidance and attains the lowest average collision rate, but the highest average density error and DTW under both protocols. This contrast shows that collision minimization alone does not ensure realistic crowd behavior. CrowdES improves density error and DTW over ORCA, but its average DTW and collision rate remain higher than Ctrl-CWM.

Ctrl-CWM ranks first on nine of sixteen average scores and second on the remaining seven, with the lowest average density error and DTW under both protocols. Relative to CrowdES, average DTW falls by $22\%$ and $18\%$ under random-surface and diffusion arrivals, respectively, while collision rates also decrease. On GCS under diffusion arrivals, DTW drops from $3.864$ to $1.547$. Overall, these gains suggest that learning motion dynamics from real-world trajectories helps preserve realistic pedestrian behavior, while planning over imagined futures accounts for interactions and collisions.

## 4.2 RUN-TIME CONTROL

**Protocol and compliance.** We introduce avoidance and attraction through the user cost in Eq. 10. We design the avoidance zones in Fig. 3 and encode them as obstacles for ORCA and CrowdES. For avoidance, the first third of each Ctrl-CWM episode is shared, after which we compare runs with $\lambda_{\text{user}} = 10$ and without the user objective ($\lambda_{\text{user}} = 0$). Baselines use the obstacles from initialization. All methods use identical arrivals, with 10 trials per scene evaluated over the final two thirds.

Table 2: **Run-time avoidance control.** Ctrl-CWM receives objectives online, whereas baseline obstacles are fixed from initialization. Higher compliance is better.

| Zone | Method | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | SDD | GCS | **AVG** |
|---|---|---|---|---|---|---|---|---|---|
| Disc | ORCA + obstacle | **1.000** | −1.154 | 0.530 | 0.428 | **1.000** | 0.344 | **0.999** | 0.450 |
| | CrowdES + map | 0.599 | 0.546 | 0.727 | 0.469 | 0.598 | 0.317 | 0.436 | 0.527 |
| | **Ctrl-CWM (ours)** | 0.837 | **0.988** | **0.831** | **0.752** | 0.714 | **0.877** | 0.827 | **0.832** |
| Rect | ORCA + obstacle | **1.000** | −1.484 | 0.526 | −0.049 | **1.000** | −1.827 | 0.595 | −0.034 |
| | CrowdES + map | 0.883 | 0.808 | **0.779** | 0.790 | 0.824 | 0.114 | 0.301 | 0.643 |
| | **Ctrl-CWM (ours)** | 0.896 | **0.881** | 0.683 | **0.988** | 0.912 | **0.934** | **0.932** | **0.889** |
| Multi | ORCA + obstacle | 0.973 | **0.999** | 0.601 | −1.952 | −0.304 | −0.473 | 0.768 | 0.087 |
| | CrowdES + map | 0.479 | 0.542 | 0.549 | 0.781 | 0.751 | 0.525 | 0.424 | 0.579 |
| | **Ctrl-CWM (ours)** | **0.992** | 0.997 | **0.993** | **0.983** | **0.971** | **0.908** | **0.808** | **0.950** |

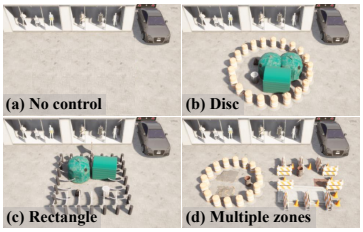



Figure 3: **Control zones.** (a) No control. (b∼d) Disc, rectangle, and multiple avoidance regions.

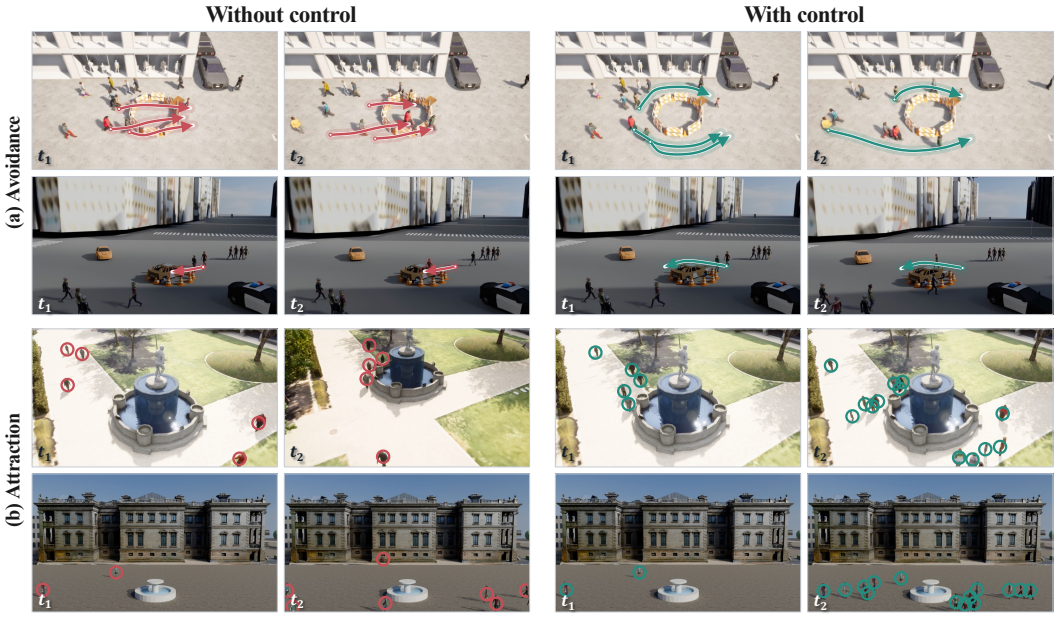



Figure 4: **Run-time control.** The left and right pairs show uncontrolled (red) and controlled (green) rollouts, respectively, at $t_1$ and $t_2$. (a) An avoidance objective redirects pedestrians away from the hazardous zone. (b) An attraction objective guides them toward the fountain.

Let $\mathrm{occ}_{\mathrm{cmd}}$ and $\mathrm{occ}_{\mathrm{free}}$ denote the fractions of agent steps inside the avoidance region with and without control. Avoidance compliance is $C_{\mathrm{avoid}} = 1 - \mathrm{occ}_{\mathrm{cmd}}/\mathrm{occ}_{\mathrm{free}}$ for $\mathrm{occ}_{\mathrm{free}} > 0$. A value of one corresponds to zero controlled occupancy, zero denotes no reduction, and negative values indicate increased occupancy. Ctrl-CWM achieves the highest average $C_{\mathrm{avoid}}$ for every tested zone configuration (Table 2). For multiple disjoint zones, compliance is $0.950$, versus $0.579$ for CrowdES and $0.087$ for ORCA. Figure 4 shows avoidance of the hazardous zone and attraction toward the fountain. Quantitative attraction results and the compliance–realism trade-off appear in Appendix D.2.

### 4.3 Analysis

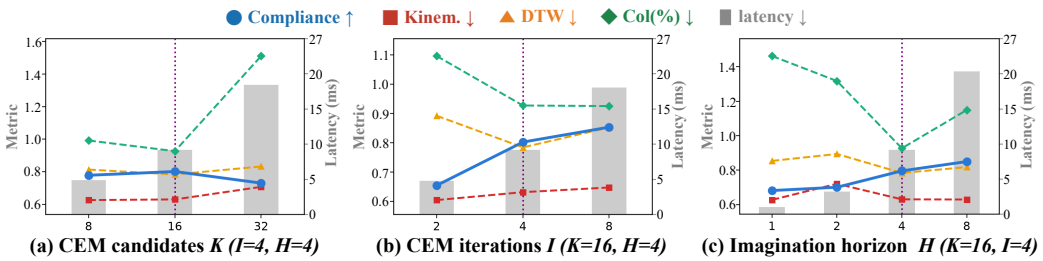
We analyze the shared representation and planning, with further ablations in Appendix C.

**Prediction pretraining.** Table 3 evaluates prediction pretraining by replacing the pretrained encoder with a randomly initialized one while retaining the same behavior-learning procedure. Pretraining reduces kinematic error from 1.205 to 0.465 and increases diversity from 0.157 to 0.285. The trajectory-prediction heads also achieve 0.21/0.31 m average ADE/FDE on ETH–UCY and 5.89/10.26 px on GCS under best-of-20 evaluation (Appendix E). These results indicate that human motion dynamics learned from real-world data transfer effectively to subsequent behavior learning.

**Planning and learned ranking.** Critic-guided planning reduces DTW from 0.970 to 0.925 and collisions from 0.931% to 0.756% relative to actor-only generation, with a small increase in kinematic error from 0.432 to 0.465. Under the same control protocol, user-cost only achieves higher compli-

Table 3: **Component ablations under diffusion arrivals.** Metrics are averaged over five ETH–UCY folds. User-cost only retains the actor, imagined rollouts, and CEM procedure but ranks candidates using only the user cost. Comp. reports avoidance compliance $C_{\text{avoid}}$ under the same protocol. **Bold**/underline mark the best/second-best scores. N/A denotes generation-only evaluation.

| Variant | Scene-level realism | | | | Agent-level accuracy | | | | Control |
|---|---|---|---|---|---|---|---|---|---|
| | Dens.↓ | Freq.↓ | Cov.↓ | Pop.↓ | Kinem.↓ | DTW↓ | Div.↑ | Col.(%)↓ | Comp.↑ |
| Random encoder | 0.101 | 0.067 | 0.067 | 0.304 | 1.205 | 1.031 | 0.157 | 1.101 | N/A |
| Actor only | **0.077** | **0.047** | **0.047** | 0.271 | **0.432** | 0.970 | 0.280 | 0.931 | N/A |
| User-cost only | 0.089 | 0.075 | 0.075 | 0.294 | 0.812 | 1.178 | 0.078 | 2.398 | **0.858** |
| **Ctrl-CWM (full)** | 0.078 | **0.047** | **0.047** | **0.254** | 0.465 | **0.925** | **0.285** | **0.756** | 0.824 |
| w/o soft-DTW | 0.082 | 0.051 | 0.051 | 0.275 | 0.846 | 0.989 | 0.182 | 0.838 | N/A |
| w/o collision term | 0.088 | 0.049 | 0.049 | 0.266 | 0.472 | 0.965 | 0.283 | 0.919 | 0.645 |
| w/o kinematic loss | 0.081 | 0.053 | 0.053 | 0.271 | 0.502 | 0.933 | 0.281 | 0.817 | 0.772 |




(a) CEM candidates *K (I=4, H=4)* (b) CEM iterations *I (K=16, H=4)* (c) Imagination horizon *H (K=16, I=4)*

Figure 5: **Ablation of planning parameters.** Lines show compliance and trajectory metrics over the post-activation control window. Gray bars report call latency in milliseconds on the right axes. Dotted lines mark the default settings.

ance than Ctrl-CWM (0.858 vs. 0.824) but performs worse on all eight realism and accuracy metrics. This indicates that the user cost drives the control objective, while the learned critic helps preserve realistic behavior during planning. Removing collision or kinematic supervision lowers compliance to 0.645 and 0.772, respectively, and degrades the related motion metrics.

**Training objectives.** The loss ablations show complementary contributions to motion quality. Removing soft-DTW increases DTW to 0.989 and kinematic error to 0.846 while reducing diversity to 0.182. Removing collision supervision raises collisions to 0.919%, while removing the kinematic loss increases kinematic error to 0.502. These results support complementary roles in trajectory matching, collision avoidance, and motion dynamics.

**Effect of planning parameters.** Figure 5 varies candidate count $K$, CEM iterations $I$, and horizon $H$ individually. Among tested counts, $K = 16$ maximizes compliance and minimizes DTW and collisions. Increasing $I$ or $H$ from 4 to 8 raises compliance, latency, and DTW. At $H = 8$, collisions also increase relative to $H = 4$. At $H = 1$, latency decreases, but compliance drops and collisions increase. Thus, our default $K = 16$, $I = 4$, $H = 4$ balances these criteria at about 10 ms per call.

## 5 Conclusion

We introduce Ctrl-CWM, a crowd world model that unifies generation and run-time control through a representation learned from real-world pedestrian videos. An actor proposes motion, while a critic ranks imagined futures to guide planning. User-defined costs steer behavior without retraining or gradient-based guidance. Extensive experiments demonstrate realistic crowd generation and adaptation to avoidance and attraction objectives introduced during simulation.

**Limitations and future work.** Our current framework requires user objectives to be specified as explicit spatial costs rather than high-level semantic commands. Future work will ground language-based instructions into planning objectives to enable semantically grounded control while preserving human-like crowd behavior.

## AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks with required disclosure, such as generating synthetic data, developing theoretical models, formulating or proving mathematical claims, designing research methodology, implementing methods, or interpreting results. All text was drafted by the authors. We used generative AI tools only to edit the text for readability. The authors reviewed all edited text and take responsibility for the final content of this work.

## ETHICS STATEMENT

This work uses publicly available datasets and 3D scenes constructed using open-source resources or purchased assets. The generated crowd behaviors may reflect biases in the source data and limitations of the simulated environments, and should not be assumed to represent human behavior across all populations or settings. Results obtained in simulation do not establish the safety of real-world applications, which would require additional validation and appropriate safeguards. Any materials released with this work will be subject to the applicable licenses and redistribution restrictions of the source datasets, software, and third-party assets.

## REPRODUCIBILITY STATEMENT

For reproducibility, we describe the model formulation and learning objectives in the main paper and provide network architectures and key training and planning settings in Appendix A. Evaluation protocols and metrics are detailed in Appendix B, while run-time control costs and sensitivity analyses are provided in Appendix D. We expect to release our code and calibration data upon acceptance, subject to internal approval.

## REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Inhwan Bae, Junoh Lee, and Hae-Gon Jeon. Can language beat numerical regression? language-based multimodal trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Inhwan Bae, Junoh Lee, and Hae-Gon Jeon. Continuous and locomotive crowd behavior generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Panayiotis Charalambous, Julien Pettre, Vassilis Vassiliades, Yiorgos Chrysanthou, and Nuria Pelechano. Greil-crowds: Crowd simulation with deep reinforcement learning and examples. *ACM Transactions on Graphics (TOG)*, 2023.

Hongyi Chen, Jingtao Ding, Yong Li, Yue Wang, and Xiao-Ping Zhang. Social physics informed diffusion model for crowd simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2018.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Vladimir Egorov and Aleksei Shpilman. Scalable multi-agent model-based reinforcement learning. *arXiv preprint arXiv:2205.15023*, 2022.

Yuxiang Fu, Qi Yan, Lele Wang, Ke Li, and Renjie Liao. Moflow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 2025.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 1995.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2019.

Xuebo Ji, Zherong Pan, Xifeng Gao, and Jia Pan. Text-guided synthesis of crowd animation. In *Proceedings of ACM SIGGRAPH*, 2024.

Eunjung Ju, Myung Geol Choi, Minji Park, Jehee Lee, Kang Hoon Lee, and Shigeo Takahashi. Morphable crowds. *ACM Transactions on Graphics (TOG)*, 2010.

Jongmin Kim, Yeongho Seol, Taesoo Kwon, and Jehee Lee. Interactive manipulation of large-scale crowd animation. *ACM Transactions on Graphics (TOG)*, 2014.

Sungsik Kim, Janghyun Baek, Jinkyu Kim, and Jaekoo Lee. Guide-cot: Goal-driven and user-informed dynamic estimation for pedestrian trajectory using chain-of-thought. *Autonomous Agents and Multiagent Systems (AAMAS)*, 2025.

Jaedong Lee, Jungdam Won, and Jehee Lee. Crowd simulation by deep reinforcement learning. In *Proceedings of ACM SIGGRAPH*, 2018.

Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 109–118, 2007.

Seongju Lee, Junseok Lee, Yeonguk Yu, Taeri Kim, and Kyoobin Lee. Mart: Multiscale relational transformer networks for multi-agent trajectory prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, 2007.

Rongqing Li, Changsheng Li, Dongchun Ren, Guangyi Chen, Ye Yuan, and Guoren Wang. Bcdiff: Bidirectional consistent diffusion for instantaneous trajectory prediction. *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023.

Ruochen Li, Zhanxing Zhu, Tanqiu Qiao, and Hubert PH Shum. Vite: Virtual graph trajectory expert router for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

Takahiro Maeda and Norimichi Ukita. Fast inference and update of probabilistic density estimation on trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Andreas Panayiotou, Andreas Aristidou, and Panayiotis Charalambous. Cedrl: Simulating diverse crowds with example-driven deep reinforcement learning. In *Computer Graphics Forum*, volume 44, pp. e70015. Wiley Online Library, 2025.

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2009.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of ACM SIGGRAPH*, 1987.

Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. *ACM Transactions on Graphics (TOG)*, 2006.

Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Optimal reciprocal collision avoidance for multi-agent navigation. In *Proc. of the IEEE International Conference on Robotics and Automation, Anchorage (AK), USA*, 2010.

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zirui Xu, Biao Yang, Rongrong Ni, Zhongkai Zhou, and Shaobo Shen. W2w: Language-model-based trajectory prediction with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Shuai Yi, Hongsheng Li, and Xiaogang Wang. Understanding pedestrian behaviors from stationary crowd groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

He Zhao and Richard P Wildes. Where are you heading? dynamic trajectory prediction with expert goal examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Rui Zhou, Hongyu Zhou, Huidong Gao, Masayoshi Tomizuka, Jiachen Li, and Zhuo Xu. Grouptron: Dynamic multi-scale graph convolutional networks for group-aware dense crowd trajectory forecasting. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

# A Implementation Details

This appendix describes the model architecture, training, and planning used by Ctrl-CWM. Appendix B separates the evaluation protocols and metrics. Appendices C and D report component and control analyses. Appendices E and F provide prediction-head results and additional visualizations. Detailed per-benchmark results are provided in Appendix G.

## A.1 Model Architecture

**Scene representation.** A per-scene similarity transform maps scene geometry and agent positions to a $G \times G$ grid with $G = 192$. Distances are in grid pixels unless explicitly normalized by $G$. The static representation $S$ contains a binary walkability mask and eight semantic channels for background, building, structure, bush, grass, tree, sidewalk, and road, following the CrowdES categories with background added (Bae et al., 2025). Two additional channels encode the focal agent and the co-present crowd as Gaussian heatmaps with standard deviation 5 grid pixels. The focal channel contains one agent, and the crowd channel accumulates co-present agents. The resulting input has eleven channels.

**Scene and interaction encoders.** At width multiplier $w = 0.25$, the U-Net has four skip levels with 8/16/32/64 channels. Four pooling stages produce a $64 \times 12 \times 12$ bottleneck. The actor samples an eight-dimensional local scene feature $\ell_i^t$ from the full-resolution skip feature at $p_i^t$. The interaction encoder embeds positions with a $2 \to 64$ linear layer. Attention uses pairwise-displacement biases from a $2 \to 16 \to 1$ MLP, while a $2 \to 64$ GRU encodes the velocity history $\mathcal{H}_i^t$. The attention output and GRU state are concatenated and projected through $128 \to 96 \to 96$ layers to obtain $c_i^t \in \mathbb{R}^{96}$. A FiLM layer maps $c_i^t$ to channel-wise scales and shifts for the scene bottleneck. The modulated bottleneck, skip features, and interaction embedding constitute $z_i^t$. Table 4 lists the layer interfaces.

**Prediction heads.** The goal decoder predicts $M_{\text{goal}}$ from the modulated bottleneck and scene skips. Its target is the endpoint of the $L$-step prediction window, not necessarily the final destination. The waypoint and auxiliary dynamics heads share a goal-conditioned decoder trunk with separate $1 \times 1$ output layers. A four-level module extracts goal features with 16/32/64/128 channels. Each decoder level upsamples its input and concatenates the corresponding scene and goal features before a $3 \times 3$ convolution, GroupNorm with eight groups, and SiLU activation. The outputs are $L = 12$ waypoint heatmaps and one next-position heatmap $M_{\text{dyn}}$. Conditioning uses the ground-truth goal heatmap during prediction training and the predicted goal during forecasting. The auxiliary head provides one-step supervision and is not used to advance rollout states.

**Actor.** The actor receives

$$[\, \ell_i^t \,\|\, c_i^t \,\|\, u_i^t \,\|\, v_i^t/G \,] \in \mathbb{R}^{108},$$

where $u_i^t \in \mathbb{R}^2$ is the unit direction to the local goal and $v_i^t \in \mathbb{R}^2$ is the current velocity. Two 128-unit hidden layers with ReLU activations precede a two-dimensional displacement output. The MLP has approximately 31K parameters, and its output-layer weights use an initialization scale of 0.01. The vector $v_i^t$ differs from the scalar speeds in Eq. 5.

**Critic and discriminator.** For each imagined step, the critic processes six geometric features,

$$[\, v_x/G,\ v_y/G,\ d_{\text{nbr}}/G,\ d_{\text{goal}}/G,\ \cos(v, g-p),\ \|v\|_2/G \,],$$

where $d_{\text{nbr}}$ and $d_{\text{goal}}$ are the nearest-neighbor and goal distances. A $6 \to 128$ GRU followed by a $128 \to 128 \to 1$ MLP produces a scalar score from the final hidden state. This network has approximately 69K parameters. The critic receives geometric features of imagined states rather than encoder features directly. The discriminator uses a $2 \to 128$ GRU and a $128 \to 128 \to 1$ MLP with LeakyReLU(0.2). It produces a scalar score for an $L$-step velocity sequence, including the first displacement from the observed starting position. The actor and discriminator objectives are defined together in Eq. 6. Table 5 summarizes these networks.

Table 4: **Encoder and prediction-head interfaces.** Spatial dimensions are per focal agent, with batch axes omitted. $A$ is the number of co-present agents and $T_h$ is the velocity-history length. Widths use $w = 0.25$. GN(8) denotes GroupNorm with eight groups.

| ID | Operation | Input | Output |
|---|---|---|---|
| *Inputs and scene encoder* | | | |
| 1-1 | Similarity transform | Scene, trajectories | Grid, $G = 192$ |
| 1-2 | Walkability and semantic channels | Transformed scene | $9 \times 192 \times 192$ |
| 1-3 | Focal-agent Gaussian, $\sigma = 5$ | Agent position | $1 \times 192 \times 192$ |
| 1-4 | Crowd Gaussian accumulation | Crowd positions | $1 \times 192 \times 192$ |
| 1-5 | Channel concatenation | 1-2, 1-3, 1-4 | $11 \times 192 \times 192$ |
| 2-1 | Encoder block, 8 channels | 1-5 | $8 \times 192 \times 192$ |
| 2-2 | Pool $2\times$, encoder block | 2-1 | $16 \times 96 \times 96$ |
| 2-3 | Pool $2\times$, encoder block | 2-2 | $32 \times 48 \times 48$ |
| 2-4 | Pool $2\times$, encoder block | 2-3 | $64 \times 24 \times 24$ |
| 2-5 | Pool $2\times$, bottleneck | 2-4 | $64 \times 12 \times 12$ |
| 2-6 | Local sampling at $p_i^t$ | 2-1, position | $\ell_i^t \in \mathbb{R}^8$ |
| *Interaction encoder* | | | |
| 3-1 | Position embedding, linear $2 \to 64$ | Positions | $A \times 64$ |
| 3-2 | Pairwise bias, MLP $2 \to 16 \to 1$ | Displacements | $A \times A \times 1$ |
| 3-3 | Attention with pairwise biases | 3-1, 3-2 | $A \times 64$ |
| 3-4 | History GRU $2 \to 64$ | $A \times T_h \times 2$ | $A \times 64$ |
| 3-5 | Concatenation | 3-3, 3-4 | $A \times 128$ |
| 3-6 | Projection $128 \to 96 \to 96$ | 3-5 | $c_i^t \in \mathbb{R}^{96}$ |
| 3-7 | FiLM scale and shift | 3-6 | Two 64-vectors |
| 3-8 | Bottleneck modulation | 2-5, 3-7 | $64 \times 12 \times 12$ |
| *Goal features and prediction heads* | | | |
| 4-1 | Goal decoder | 3-8, scene skips | $M_{\text{goal}}$ |
| 4-2 | Goal feature level 1 | Goal heatmap | $16 \times 192 \times 192$ |
| 4-3 | Goal feature level 2 | 4-2 | $32 \times 96 \times 96$ |
| 4-4 | Goal feature level 3 | 4-3 | $64 \times 48 \times 48$ |
| 4-5 | Goal feature level 4 | 4-4 | $128 \times 24 \times 24$ |
| 4-6 | Upsample, concatenate, conv, GN(8), SiLU | 3-8, 2-4, 4-5 | $64 \times 24 \times 24$ |
| 4-7 | Upsample, concatenate, conv, GN(8), SiLU | 4-6, 2-3, 4-4 | $32 \times 48 \times 48$ |
| 4-8 | Upsample, concatenate, conv, GN(8), SiLU | 4-7, 2-2, 4-3 | $16 \times 96 \times 96$ |
| 4-9 | Upsample, concatenate, conv, GN(8), SiLU | 4-8, 2-1, 4-2 | $8 \times 192 \times 192$ |
| 4-10 | Waypoint output, conv $1 \times 1$ | 4-9 | $L \times 192 \times 192$ |
| 4-11 | Auxiliary output, conv $1 \times 1$ | 4-9 | $1 \times 192 \times 192$ |

Table 5: **Behavior-learning networks.** The actor receives frozen encoder features in the full model. The critic uses $H$ imagined steps, while its training targets use $L$ steps. The discriminator uses $L$ displacement vectors.

| Module | Layers or input | Output |
|---|---|---|
| Actor input | $[\ell_i^t \| c_i^t \| u_i^t \| v_i^t / G]$ | 108 features |
| Actor hidden | Linear $108 \to 128 \to 128$, ReLU | 128 features |
| Actor output | Linear $128 \to 2$ | Displacement |
| Critic input | Six geometric features at each imagined step | $H \times 6$ |
| Critic recurrent | GRU $6 \to 128$, final hidden state | 128 features |
| Critic output | MLP $128 \to 128 \to 1$ | Rollout score |
| Discriminator input | Normalized displacement sequence | $L \times 2$ |
| Discriminator recurrent | GRU $2 \to 128$, final hidden state | 128 features |
| Discriminator output | MLP $128 \to 128 \to 1$, LeakyReLU(0.2) | Sequence score |

### A.2 Training Procedure

**Prediction pretraining.** We jointly optimize $h_\theta$ and the prediction heads with Eq. 3 for 80 epochs. Training uses AdamW with learning rate $3 \times 10^{-4}$, weight decay $10^{-5}$, cosine annealing, and gradient-norm clipping at 1.0. A single RTX 3090 GPU processes three scenes per optimizer update using batch size one and three gradient-accumulation steps. The heatmap BCE terms use positive-class weight 50. The full model freezes $h_\theta$ after pretraining.

**Actor and discriminator.** Behavior learning uses four trajectory windows per batch, 40 epochs, and $L = 12$ rollout steps. All agents follow the actor during actor training. Predicted positions update both the crowd state and velocity histories, and scene and interaction features are recomputed at each step. Future ground-truth positions do not replace the generated states. Gradients pass through position updates, while the rasterization and feature-extraction branches are detached. The actor uses nearby waypoints sampled from ground-truth trajectories as training goals. Its objective combines imitation, kinematic matching, and adversarial supervision, with no direct collision penalty, critic score, or run-time user cost. Actor and discriminator losses are averaged over valid agents.

The kinematic loss in Eq. 5 compares normalized scalar speeds and successive changes in speed. It does not compare two-dimensional acceleration vectors. Its $L$ predicted positions give $L-1$ speeds and $L-2$ speed differences. The discriminator instead receives $L$ normalized displacement vectors, including the displacement from the observed starting position to the first prediction. These two temporal conventions should not be interchanged.

**Critic.** For each training state and focal agent, $K$ candidate displacements are sampled around the actor proposal. Each candidate changes only the focal agent's first action, and the actor supplies every other action over the $L$-step rollout. The candidate cost in Eq. 7 uses all $L$ steps, while the critic observes geometric features from only the first $H$ steps. The listwise objective in Eq. 9 assigns larger scores to lower-cost candidates. Its softmax is over candidates for the same state and agent. The temperature is $\tau_c = 1$. The critic is trained from trajectory-derived costs rather than a bootstrapped return target. Neither prediction targets nor future ground-truth trajectories are needed for planning at inference.

The collision target is the number of neighbors within $r = 4$ grid pixels, averaged over the $L$ imagined steps. Repeated proximity at multiple steps contributes multiple times before temporal averaging. This term supervises the critic and is distinct from the benchmark's collision-rate metric.

**Calibrated loss weights.** The actor weight $\lambda_{\text{kin}}$ and critic-target weights $\lambda_{\text{kin}}^{T}$ and $\lambda_{\text{coll}}$ use separate calibration rules. During an initial eight-epoch calibration period, the actor kinematic term is inactive. Its eventual weight is 0.3 times the ratio of the mean actor-gradient norm of imitation-plus-adversarial loss to that of the kinematic loss. For each critic-target term, the weight is 0.3 times the ratio of the mean candidate standard deviation of the imitation cost to that of the corresponding term. Thus, 0.3 is a calibration ratio, not any of the three realized loss weights. These weights are also distinct from $\lambda_{\text{user}}$. Table 6 separates fixed settings from calibrated quantities.

### A.3 Planning, Goal Updates, and Execution

At the start of each $L$-step navigation window, the goal decoder predicts a heatmap from the current state. An extracted coordinate defines $g_i^t$ and remains fixed within that window. This goal-update interval is distinct from the shorter planning horizon $H$. The actor uses ground-truth waypoints during behavior learning and predicted local goals during simulation.

At each simulation step, CEM initializes its sampling distribution at the actor proposal with isotropic standard deviation $\sigma_{\text{plan}}$. Each iteration draws $K$ candidates, scores their imagined rollouts, and refits the mean and standard deviation to the highest-scoring quarter. The final mean after $I$ iterations is executed. The refit adapts the distribution to selected samples and does not imply that its spread decreases monotonically. Algorithm 1 states which actions are replaced in each imagined crowd rollout.

The reported planning experiments use $\mathcal{C}_t = \mathcal{A}_t$. Each agent's next displacement is optimized independently with other agents following the actor inside that candidate rollout. The resulting displace-

Table 6: **Training and planning settings.** The three calibrated weights are not fixed to the calibration ratio. User-cost weights are reported with each control protocol.

| Setting | Value |
|---|---|
| *Prediction pretraining* | |
| Optimizer | AdamW |
| Learning rate / weight decay | $3 \times 10^{-4}$ / $10^{-5}$ |
| Learning-rate schedule / gradient clipping | Cosine / 1.0 |
| Epochs / scenes per optimizer update | 80 / 3 |
| Batch size / accumulation steps | 1 / 3 |
| Prediction horizon $L$ | 12 |
| Waypoint / auxiliary weights | $\lambda_{\text{way}} = 1$, $\lambda_{\text{dyn}} = 0.5$ |
| Heatmap positive-class weight | 50 |
| *Behavior learning* | |
| Epochs / trajectory windows per batch | 40 / 4 |
| Rollout length | $L = 12$ |
| Actor and critic optimizer | AdamW |
| Actor/critic learning rate / weight decay | $3 \times 10^{-4}$ / $10^{-5}$ |
| Discriminator optimizer | Adam |
| Discriminator learning rate / $(\beta_1, \beta_2)$ | $2 \times 10^{-4}$ / $(0.5, 0.9)$ |
| Gradient clipping / critic updates per actor update | 1.0 / 2 |
| Position / soft-DTW weights | $\lambda_{\text{pos}} = 1$, $\lambda_{\text{dtw}} = 0.5$ |
| Soft-DTW smoothing / adversarial weight | $\gamma = 1$, $\lambda_{\text{adv}} = 0.3$ |
| Actor kinematic weight $\lambda_{\text{kin}}$ | Gradient-norm calibrated |
| Actor calibration period / ratio | 8 epochs / 0.3 |
| Critic-target weights $\lambda_{\text{kin}}^{T}$, $\lambda_{\text{coll}}$ | Candidate-spread calibrated |
| Critic-target calibration ratio | 0.3 for each term |
| Collision threshold / listwise temperature | $r = 4$ grid pixels / $\tau_c = 1$ |
| *CEM planning* | |
| Candidates / iterations / horizon | $K = 16$, $I = 4$, $H = 4$ |
| Elite fraction | Highest-scoring quarter |
| Initial standard deviation | $\sigma_{\text{plan}} = 1.5$ grid pixels |

**Algorithm 1** One simulation step of Ctrl-CWM

**Require:** State $s_t$, selected agents $\mathcal{C}_t$, user cost $c_{\text{user}}$, weight $\lambda_{\text{user}}$, and $K, I, H, \sigma_{\text{plan}}$
Compute $z_i^t = h_\theta(s_t)_i$ and $\bar{a}_i^t = \pi_\psi(z_i^t, g_i^t, v_i^t)$ for all agents
**for** each $i \in \mathcal{C}_t$ in parallel **do**
  Initialize a Gaussian at $\bar{a}_i^t$ with isotropic standard deviation $\sigma_{\text{plan}}$
  **for** $j = 1, \ldots, I$ **do**
    Draw $K$ candidate displacements $a_i^{t,(k)}$
    **for** $k = 1, \ldots, K$ **do**
      Copy $s_t$ and replace only agent $i$'s first action with $a_i^{t,(k)}$
      Imagine $H$ steps, using the actor for all other actions
      Update interaction features and sample cached scene features at each step
      Score $\hat{s}_{t+1:t+H}^{(k)}$ using Eq. 10
    **end for**
    Refit the Gaussian mean and standard deviation to the highest-scoring quarter
  **end for**
  Set $a_i^t$ to the final mean
**end for**
Apply all selected displacements simultaneously using Eq. 2
Use actor proposals for agents outside $\mathcal{C}_t$ and update velocity histories

ments are then executed simultaneously. This is an approximation to joint multi-agent planning, not a jointly optimized action sequence. Actor-only generation sets $\mathcal{C}_t = \varnothing$. Full-model generation retains CEM with $\lambda_{\text{user}} = 0$, and control adds the user cost to the same score.

For efficiency, planning caches the U-Net maps computed from the scene and agent-position heatmaps at the start of an imagined rollout. The maps remain fixed for $H$ steps, while interaction embeddings and local sampling locations are updated at every step. Because the raster includes agent positions, this cache approximates the transition used during behavior learning, where both scene and interaction features are recomputed.

### A.4 Checkpoint Selection

The actor and critic are selected as a pair using held-out checkpoint-selection data. Actor quality is measured by full-rollout position error. Critic quality is measured by Spearman correlation between critic scores and negative candidate costs, so larger values indicate better motion in both quantities. Among checkpoints within $5\%$ of the lowest selection error, the pair with the highest ranking correlation is selected. The reported correlations range from 0.60 to 0.76 across folds.

## B Evaluation Protocols and Metrics

### B.1 Evaluation Scope

Table 7 distinguishes generation, control, and prediction evaluations. A shared model configuration does not imply an identical evaluation window or aggregation rule. In particular, the generation columns of Table 3 and the post-activation metrics in Fig. 5 evaluate different conditions. Their values should not be combined as if they came from the same controlled rollout.

Table 7: **Evaluation protocols.** Repeat counts are shown where reported. The generation and control columns of Table 3 use separate evaluations.

| Evaluation | Setting and aggregation | Repeats and window |
|---|---|---|
| Generation, Table 1 | Two replayed arrival protocols. Mean of five ETH–UCY folds, SDD, and GCS. | Full episodes. |
| Avoidance, Table 2 | Disc, rectangle, multiple zones. $\lambda_{\text{user}} = 10$. Same seven-entry mean. | 10 trials per scene. Final two thirds. |
| Components, Table 3 | Diffusion arrivals over five folds. Generation ablations use full episodes; the user-cost-only comparison uses the same avoidance-control protocol as the full model. | Full episodes for generation; post-activation window for control. |
| Planning, Fig. 5 | Five folds. $K$, $I$, and $H$ varied individually. | Post-activation window. |
| Additional components, Appendix C | Five folds. Disc avoidance, $\lambda_{\text{user}} = 10$. | 20 trials per scene. Final two thirds. |
| Weight/population sweeps, Appendix D | Five folds. Weights and emitter multipliers varied as reported. | One trial per configuration. Post-activation window. |
| Prediction heads, Appendix E | Eight observed, twelve predicted steps. Best-of-20 ADE/FDE. | Prediction windows, not controlled crowd episodes. |

**Shared arrivals.** An emitter determines pedestrian entries into the scene. Within each generation protocol, all compared methods receive identical replayed entries from the random-surface or diffusion emitter. This controls for the arrival process when comparing motion generation. The seven-entry main average gives equal weight to each ETH–UCY fold, SDD, and GCS rather than equal weight to the three dataset families.

**Run-time avoidance.** For Ctrl-CWM, runs with and without the objective share the first third of the episode. After that point, the controlled run uses $\lambda_{\text{user}} = 10$, while the run without the objective

retains critic-guided planning with $\lambda_{\text{user}} = 0$. ORCA and CrowdES are evaluated with or without the corresponding obstacles from initialization. All methods use identical arrivals, and the main avoidance comparison uses 10 trials per scene evaluated over the final two thirds. The baseline comparison therefore has different intervention times and is not a matched comparison of online objective activation. The weight and repeat settings of this main avoidance experiment do not apply to the attraction sweeps.

### B.2 Generation Metrics

We use the eight CrowdES generation metrics (Bae et al., 2025). Distributional metrics compare generated and ground-truth statistics using Earth Mover's Distance (EMD). Density (Dens.), frequency (Freq.), and coverage (Cov.) use quadrat sampling on a $10 \times 10$ evaluation grid, with statistics collected every second. Population (Pop.) compares histograms of concurrent agent counts at the same interval. This grid is distinct from the $192 \times 192$ model-input grid.

Kinematics (Kinem.) averages EMDs for travel velocities, accelerations, travel distances, and travel times. DTW averages the minimum pairwise Dynamic Time Warping distance between generated and ground-truth trajectories under the benchmark's coordinate conversion. Diversity (Div.) measures coverage of ground-truth trajectories by the matched pairs. Collision rate (Col.) is the benchmark's inter-agent collision percentage. Lower is better except for diversity.

Freq. and Cov. are computed with respect to agent categories. They coincide on the single-category ETH–UCY and GCS data, but can differ on SDD. Consequently, ranks across the displayed metrics are not independent measurements. The main generation mean uses seven benchmark entries, whereas component and sensitivity means use five ETH–UCY folds.

**Training losses versus evaluation metrics.** The training loss $\mathcal{L}_{\text{kin}}$ matches scalar speed profiles and their successive differences. It is not the distributional Kinem. metric. Likewise, the candidate collision cost $C_i^{(k)}$ counts nearby neighbors over imagined steps and is not the benchmark Col. percentage. The soft-DTW training term is also distinct from the trajectory-matching DTW evaluation. These quantities should retain separate names and units. Control costs and compliance measures are defined in Appendix D.

## C Additional Component Analysis

### C.1 Interpretation of the Main Ablations

Table 3 evaluates prediction pretraining, planning, learned ranking, and selected training terms. The no-pretraining variant starts from a randomly initialized encoder, rather than the prediction-pretrained initialization. The full model uses a frozen prediction-pretrained encoder. The comparison does not by itself establish that freezing is preferable to fine-tuning.

Actor-only generation bypasses CEM. The user-cost-only planner retains the actor, imagined rollouts, and CEM procedure but removes the critic score, ranking candidates solely by $-\lambda_{\text{user}} c_{\text{user}}$. Its comparison with the full model uses the same avoidance protocol and post-activation evaluation window. User-cost only achieves higher compliance (0.858 vs. 0.824), while the full model performs better on all eight motion metrics.

The loss-removal variants retain the planner and critic. Collision supervision refers to the collision term in the critic target, not a direct actor collision loss. The reported decreases in compliance to 0.645 and 0.772 are from a separate control evaluation, whereas the associated motion columns report generation performance. They do not establish the control–realism trade-off within a single set of controlled trajectories.

### C.2 Interaction and Auxiliary Supervision

The additional component experiments use five ETH–UCY folds, diffusion arrivals, disc avoidance, $\lambda_{\text{user}} = 10$, and 20 trials per scene. Each disc covers $7.5\%$ of the fold's walkable area, with radii between 18.9 and 28.5 grid pixels. Runs share the first third of each episode, replayed arrivals,

and region geometry. Only the controlled run receives the user cost, and evaluation uses the post-activation window. These results are separate from the main component table and the one-trial sweeps.

The interaction encoder improves compliance on four of five folds, increasing the mean from 0.578 to 0.880 (Table 8). Collision effects are not uniform. The ablated model has fewer collisions on HOTEL, ZARA1, and ZARA2, but lower compliance on those folds. The large ETH collision increase without interaction contributes strongly to the mean difference.

The auxiliary dynamics head also improves compliance on four folds (Table 9). Mean compliance changes from 0.584 without the head to 0.880 with it. The largest decreases occur on HOTEL and UNIV, while ZARA1 improves without the head. Because this head is unused for state updates, this ablation concerns training supervision rather than an inference-time transition module.

Table 8: **Interaction-encoder ablation.** Disc avoidance with $\lambda_{\text{user}} = 10$ and 20 trials per scene. AVG is calculated from the displayed fold values.

| | Full model | | w/o interaction | |
|---|---|---|---|---|
| Fold | Col. (%) ↓ | Comp. ↑ | Col. (%) ↓ | Comp. ↑ |
| ETH | 0.458 | 0.932 | 20.32 | 0.950 |
| HOTEL | 1.522 | 0.992 | 1.292 | 0.497 |
| UNIV | 0.766 | 0.882 | 1.133 | 0.468 |
| ZARA1 | 3.508 | 0.839 | 0.868 | 0.466 |
| ZARA2 | 2.033 | 0.754 | 1.062 | 0.510 |
| AVG | 1.657 | 0.880 | 4.935 | 0.578 |

Table 9: **Auxiliary dynamics-head ablation.** Avoidance compliance under the same disc-control protocol. AVG is calculated from the displayed fold values.

| Fold | Full model | w/o dynamics head |
|---|---|---|
| ETH | 0.932 | 0.880 |
| HOTEL | 0.992 | 0.215 |
| UNIV | 0.882 | 0.245 |
| ZARA1 | 0.839 | 0.880 |
| ZARA2 | 0.754 | 0.699 |
| AVG | 0.880 | 0.584 |

### C.3 Critic Training Objective

The listwise critic has lower mean kinematic error, DTW, and collision rate, and higher compliance than pointwise regression (Table 10). The listwise soft targets depend on both the ordering and differences of candidate costs. The collision improvement is concentrated in ETH, with 0.458% for the listwise critic and 7.874% for pointwise regression. This comparison evaluates the critic's training objective, not removal of the critic.

Table 10: **Critic-objective comparison.** Five-fold means under the 20-trial disc-control protocol.

| Variant | Kinem. ↓ | DTW ↓ | Col. (%) ↓ | Comp. ↑ |
|---|---|---|---|---|
| Pointwise critic | 0.433 | 0.994 | 2.454 | 0.870 |
| **Full model (listwise)** | **0.404** | **0.985** | **1.657** | **0.880** |

# D Run-time Control Costs and Sensitivity

## D.1 Spatial Costs and Compliance

**Planning costs.** For a disc, let $Z = \{p : \|p - z_c\|_2 \le r_Z\}$, where $z_c$ and $r_Z$ are its center and radius. The zone radius differs from the inter-agent collision threshold $r$. For candidate $k$, $\hat{p}_b^{t+h,(k)}$ is agent $b$'s position at imagined step $h$. With $\mathcal{A}_t$ fixed over the short rollout, the costs are

$$c_{\text{avoid}}(\hat{s}^{(k)}) = \frac{1}{H|\mathcal{A}_t|} \sum_{h=1}^{H} \sum_{b \in \mathcal{A}_t} \mathbf{1}[\hat{p}_b^{t+h,(k)} \in Z], \tag{11}$$

$$c_{\text{attract}}(\hat{s}^{(k)}) = \frac{1}{H|\mathcal{A}_t|} \sum_{h=1}^{H} \sum_{b \in \mathcal{A}_t} \frac{\|\hat{p}_b^{t+h,(k)} - z_c\|_2}{G}. \tag{12}$$

Avoidance penalizes the fraction of imagined agent steps inside the zone. For a rectangle or multiple zones, $Z$ is the corresponding rectangle or union. Attraction penalizes mean distance to the target center. Its region radius is used for occupancy evaluation but not in the distance cost. Changing the target or zone changes the planning objective without retraining the model or changing the static scene map.

**Evaluation measures.** Let $\text{occ}_{\text{cmd}}$ and $\text{occ}_{\text{free}}$ be the fractions of agent steps inside $Z$ during the evaluation window, with and without the user objective. Compliance is

$$C_{\text{avoid}} = 1 - \frac{\text{occ}_{\text{cmd}}}{\text{occ}_{\text{free}}}, \qquad C_{\text{attract}} = \frac{\text{occ}_{\text{cmd}} - \text{occ}_{\text{free}}}{1 - \text{occ}_{\text{free}}}. \tag{13}$$

Avoidance requires $\text{occ}_{\text{free}} > 0$, and attraction requires $\text{occ}_{\text{free}} < 1$. For avoidance, one means zero controlled occupancy and zero means no reduction. Negative values mean increased occupancy for avoidance or decreased occupancy for attraction. These measures quantify relative occupancy changes, not route completion or the percentage of agents obeying a command.

**Paired sweep protocol.** The weight and population sweeps compare runs that share the first third of the episode and identical arrivals. After activation, only one run receives the user cost. The other continues critic-guided generation with $\lambda_{\text{user}} = 0$. Each configuration uses one trial unless stated otherwise. All reported changes use the post-activation window and are computed as controlled minus uncontrolled values. Collision-rate changes are in percentage points (pp). The supplied sweeps do not provide variability estimates, so their trends are descriptive rather than statistical significance claims.

## D.2 Effect of the User-Cost Weight

Avoidance and attraction use costs with different scales. Their numerical weights should therefore not be compared directly. Tables 11–13 report the corresponding results and changes in motion statistics.

**Avoidance.** Mean compliance increases from 0.607 at weight 5 to 0.863 at 10 and 0.985 at 25. Between weights 25 and 100, it stays between 0.968 and 0.985 without improving monotonically. The largest mean increase in windowed kinematic error is 0.099. At weights of at least 25, some rollouts on ETH, HOTEL, and UNIV slow or stop near the region. Thus, high compliance can reflect reduced motion rather than successful rerouting.

**Attraction.** Mean compliance increases from 0.381 at $\lambda_{\text{user}} = 1.5$ to 0.645 at 3.0 and 0.951 at 5.0. At 3.0, the mean changes in kinematic error and collision rate are +6.44 and +2.8 pp. At 5.0, these changes reach +7.76 and +26.5 pp. Stronger attraction therefore increases target occupancy but can substantially increase collisions and change crowd statistics. The high-compliance setting should not be interpreted as preserving realism or safety.

Table 11: **Avoidance weight sweep.** Five folds, one trial per configuration, post-activation evaluation. Kinematic changes are controlled minus uncontrolled. $\lambda$ is $\lambda_{\text{user}}$.

| *Avoidance compliance* ↑ | | | | | |
|---|---|---|---|---|---|
| Fold | $\lambda = 5$ | $\lambda = 10$ | $\lambda = 25$ | $\lambda = 50$ | $\lambda = 100$ |
| ETH | 0.657 | 0.949 | 0.987 | 0.955 | 0.997 |
| HOTEL | 0.661 | 0.823 | 0.990 | 0.964 | 0.992 |
| UNIV | 0.694 | 0.931 | 1.000 | 1.000 | 1.000 |
| ZARA1 | 0.422 | 0.759 | 0.963 | 0.973 | 0.938 |
| ZARA2 | 0.600 | 0.855 | 0.986 | 0.949 | 0.967 |
| AVG | 0.607 | 0.863 | 0.985 | 0.968 | 0.979 |
| *ΔKinem. (controlled minus uncontrolled)* | | | | | |
| Fold | $\lambda = 5$ | $\lambda = 10$ | $\lambda = 25$ | $\lambda = 50$ | $\lambda = 100$ |
| ETH | $+0.006$ | $+0.058$ | $+0.042$ | $+0.040$ | $+0.032$ |
| HOTEL | $+0.143$ | $+0.165$ | $+0.157$ | $+0.157$ | $+0.158$ |
| UNIV | $-0.017$ | $-0.029$ | $-0.013$ | $+0.000$ | $-0.018$ |
| ZARA1 | $+0.037$ | $+0.032$ | $+0.173$ | $+0.186$ | $+0.127$ |
| ZARA2 | $+0.074$ | $+0.090$ | $+0.117$ | $+0.113$ | $+0.127$ |
| AVG | $+0.049$ | $+0.063$ | $+0.095$ | $+0.099$ | $+0.085$ |

Table 12: **Attraction weight sweep.** Avoidance and attraction weights have different cost scales. One trial is used per configuration. AVG is calculated from the displayed fold values.

| Fold | $\lambda_{\text{user}} = 1.5$ | $\lambda_{\text{user}} = 3.0$ | $\lambda_{\text{user}} = 5.0$ |
|---|---|---|---|
| ETH | 0.480 | 0.674 | 0.977 |
| HOTEL | 0.406 | 0.580 | 0.907 |
| UNIV | 0.244 | 0.670 | 0.960 |
| ZARA1 | 0.390 | 0.572 | 0.930 |
| ZARA2 | 0.386 | 0.727 | 0.983 |
| AVG | 0.381 | 0.645 | 0.951 |

Table 13: **Motion changes under attraction.** Post-activation controlled minus uncontrolled values from the supplied sweeps. Collision changes are percentage points (pp), not relative percentages.

| | $\lambda_{\text{user}} = 3.0$ | | $\lambda_{\text{user}} = 5.0$ | |
|---|---|---|---|---|
| Fold | ΔKinem. | ΔCol. (pp) | ΔKinem. | ΔCol. (pp) |
| ETH | $+13.92$ | $+11.2$ | $+13.78$ | $+37.9$ |
| HOTEL | $+5.41$ | $-0.4$ | $+10.66$ | $+22.2$ |
| UNIV | $+3.14$ | $+3.0$ | $+3.45$ | $+32.8$ |
| ZARA1 | $+4.09$ | $-0.5$ | $+4.98$ | $+15.6$ |
| ZARA2 | $+5.63$ | $+0.7$ | $+5.93$ | $+24.1$ |
| AVG | $+6.44$ | $+2.8$ | $+7.76$ | $+26.5$ |

### D.3 Sensitivity to the Emitter Population

We vary the emitter population multiplier over $0.5$, $1$, and $2$ at avoidance weights 10 and 25. This is a fourfold range of emitter settings, not a measurement of a fourfold change in realized crowd density. At weight 25, mean compliance is between 0.928 and 0.985, the lowest per-fold compliance is 0.847, and collision changes have magnitude at most 0.6 pp. At weight 10, mean compliance decreases from 0.863 at multiplier 1 to 0.701 at multiplier 2. Collision changes are within 0.5 pp except on ETH at multiplier $0.5$, where the increase is 9.5 pp. Both control strength and the emission setting therefore affect the observed trade-off.

Table 14: **Emitter-population sensitivity.** Compliance and post-activation collision-rate changes at each emission multiplier. One trial is used per configuration. All signs, including rounded zero changes, follow the source.

| $\lambda_{\text{user}} = 10$ | | | | | | |
|---|---|---|---|---|---|---|
| | $\times 0.5$ | | $\times 1$ | | $\times 2$ | |
| Fold | Comp. ↑ | ΔCol. (pp) | Comp. ↑ | ΔCol. (pp) | Comp. ↑ | ΔCol. (pp) |
| ETH | 0.928 | $+9.5$ | 0.949 | $-0.0$ | 0.836 | $-0.1$ |
| HOTEL | 0.843 | $+0.0$ | 0.823 | $+0.0$ | 0.632 | $+0.0$ |
| UNIV | 1.000 | $-0.0$ | 0.931 | $-0.0$ | 0.732 | $+0.0$ |
| ZARA1 | 0.651 | $-0.2$ | 0.759 | $-0.2$ | 0.744 | $+0.5$ |
| ZARA2 | 0.683 | $-0.1$ | 0.855 | $-0.1$ | 0.563 | $-0.1$ |
| AVG | 0.821 | $+1.8$ | 0.863 | $-0.1$ | 0.701 | $+0.1$ |
| $\lambda_{\text{user}} = 25$ | | | | | | |
| | $\times 0.5$ | | $\times 1$ | | $\times 2$ | |
| Fold | Comp. ↑ | ΔCol. (pp) | Comp. ↑ | ΔCol. (pp) | Comp. ↑ | ΔCol. (pp) |
| ETH | 0.964 | $+0.1$ | 0.987 | $+0.1$ | 0.976 | $-0.1$ |
| HOTEL | 0.931 | $-0.1$ | 0.990 | $+0.0$ | 0.896 | $+0.1$ |
| UNIV | 1.000 | $+0.2$ | 1.000 | $-0.0$ | 1.000 | $+0.2$ |
| ZARA1 | 0.943 | $-0.2$ | 0.963 | $-0.1$ | 0.920 | $+0.6$ |
| ZARA2 | 0.913 | $-0.1$ | 0.986 | $-0.1$ | 0.847 | $-0.0$ |
| AVG | 0.950 | $-0.0$ | 0.985 | $-0.0$ | 0.928 | $+0.2$ |

## E Trajectory Prediction

This evaluation uses the prediction heads trained jointly with $h_\theta$, not newly fitted heads or the run-time planner. Eight observed steps are used to predict $L = 12$ future steps, with best-of-20 ADE/FDE. ETH–UCY errors are in meters, while SDD and GCS errors are in pixels. The ETH–UCY result averages the five folds. Table 15 compares prediction accuracy across these benchmarks. These results assess the predictive information learned during pretraining and complement, rather than replace, closed-loop generation and control evaluation.

**Scope of the pretraining comparison.** Table 3 shows lower generation kinematic error and higher diversity with prediction pretraining than with random initialization. This supports the usefulness of the learned representation in the evaluated system. It does not determine whether freezing outperforms fine-tuning, and forecasting accuracy alone does not establish the accuracy of all counterfactual crowd responses encountered during control.

Table 15: **Trajectory prediction on ETH–UCY, SDD, and GCS.** Best-of-20 ADE/FDE is reported in meters for ETH–UCY and pixels for SDD and GCS. AVG is the mean over the five ETH–UCY folds. **Bold** and underline denote the best and second-best distinct values. N/A denotes unreported results.

| Model | Year | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | AVG | SDD | GCS |
|---|---|---|---|---|---|---|---|---|---|
| AgentFormer Yuan et al. (2021) | 2021 | 0.46/0.80 | 0.14/0.22 | 0.25/0.45 | 0.18/0.30 | 0.14/0.24 | 0.23/0.40 | 8.7/14.9 | 10.2/16.9 |
| MID Gu et al. (2022) | 2022 | 0.57/0.93 | 0.21/0.33 | 0.29/0.55 | 0.28/0.50 | 0.20/0.37 | 0.31/0.54 | 7.6/14.3 | 10.7/18.2 |
| EqMotion Xu et al. (2023) | 2023 | 0.40/0.61 | 0.12/0.18 | 0.23/0.43 | 0.18/0.32 | 0.13/0.23 | 0.21/0.35 | 7.9/11.9 | 7.6/13.1 |
| MART Lee et al. (2024) | 2024 | 0.35/0.47 | 0.14/0.22 | 0.25/0.45 | 0.17/0.29 | 0.13/0.22 | 0.21/0.33 | **7.4**/11.8 | 10.6/14.1 |
| LMTraj Bae et al. (2024) | 2024 | 0.41/0.51 | 0.12/0.16 | 0.22/0.34 | 0.20/0.32 | 0.17/0.27 | 0.22/0.32 | 7.8/**10.1** | 7.1/**9.6** |
| MoFlow Fu et al. (2025) | 2025 | 0.40/0.57 | **0.11**/0.17 | 0.23/0.39 | **0.15**/**0.26** | **0.12**/0.22 | **0.20**/0.32 | 7.5/12.0 | 9.1/11.6 |
| GUIDE-CoT Kim et al. (2025) | 2025 | 0.38/0.43 | 0.13/**0.15** | 0.34/0.48 | 0.19/0.29 | 0.17/**0.21** | 0.24/0.31 | N/A | N/A |
| ViTE Li et al. (2026) | 2026 | 0.35/0.49 | **0.11**/0.17 | 0.23/0.42 | 0.18/0.30 | 0.13/0.22 | **0.20**/0.32 | **7.4**/11.9 | N/A |
| W2W Xu et al. (2026) | 2026 | 0.35/**0.41** | 0.12/**0.15** | **0.20**/**0.32** | 0.19/0.29 | 0.17/0.26 | 0.21/**0.29** | **7.4**/10.1 | N/A |
| **Ours** | — | **0.34**/0.42 | 0.15/0.20 | 0.26/0.43 | 0.16/**0.26** | 0.14/0.24 | 0.21/0.31 | 7.8/11.9 | **5.9**/10.3 |

## F Additional Qualitative Results

Figure 6 extends the main visualizations with three reconstructed scenes for each objective. The left column shows the initial scene, while the middle and right columns compare motion without and with control. Avoidance redirects motion away from a specified region, and attraction draws motion toward a target. These visualizations illustrate the spatial objectives but do not independently establish quantitative transfer performance or collision-free operation.




Figure 6: **Run-time control in additional reconstructed scenes.** Columns show the initial scene, motion without control (red), and motion with control (green). Each objective is illustrated in three scenes. (a) Avoidance redirects pedestrians away from a hazardous region. (b) Attraction guides them toward a specified target.

## G Detailed Benchmark Results

Tables 16 and 17 reproduce the complete generation values from Table 1 at a larger scale. Shared arrivals control the emission process, while the measured outcomes still depend on the generated motion. The aggregate improvements are not uniform across datasets or metrics. For example, diffusion-arrival kinematic error is higher than CrowdES on several entries, and the SDD collision rate is also higher. Table 18 provides the per-benchmark avoidance results from Table 2. The different intervention times described in Appendix B apply to these results.

Table 16: **Random-surface-arrival generation results.** Complete values from Table 1. Col. is a percentage. Bold/underline identify the best/second-best distinct displayed values within each benchmark entry.

| Data | Method | Dens.↓ | Freq.↓ | Cov.↓ | Pop.↓ | Kin.↓ | DTW↓ | Div.↑ | Col.↓ |
|---|---|---|---|---|---|---|---|---|---|
| ETH | ORCA | **0.019** | **0.016** | **0.016** | **0.205** | 0.839 | 2.848 | 0.092 | **0.004** |
| | CrowdES | 0.044 | 0.025 | 0.025 | 0.461 | 1.138 | 2.608 | **0.240** | 2.660 |
| | Ctrl-CWM | 0.034 | **0.016** | **0.016** | 0.398 | **0.358** | **1.511** | 0.171 | 0.265 |
| HOTEL | ORCA | 0.044 | 0.017 | 0.017 | 0.396 | 0.623 | 1.094 | 0.179 | **0.015** |
| | CrowdES | 0.015 | 0.016 | 0.016 | 0.130 | 0.524 | 0.907 | 0.184 | 2.471 |
| | Ctrl-CWM | **0.012** | **0.012** | **0.012** | **0.098** | **0.419** | **0.746** | **0.224** | 0.828 |
| UNIV | ORCA | **0.311** | **0.210** | **0.210** | **0.658** | **0.357** | 1.878 | 0.221 | **0.007** |
| | CrowdES | 0.365 | 0.230 | 0.230 | 0.771 | 0.457 | 1.394 | 0.249 | 2.758 |
| | Ctrl-CWM | 0.362 | 0.216 | 0.216 | 0.765 | 0.543 | **1.184** | **0.327** | 0.733 |
| ZARA1 | ORCA | 0.063 | 0.027 | 0.027 | 0.953 | 1.435 | 2.530 | 0.181 | **0.002** |
| | CrowdES | 0.011 | 0.009 | 0.009 | 0.167 | **0.358** | 0.706 | **0.283** | 2.010 |
| | Ctrl-CWM | **0.010** | **0.008** | **0.008** | **0.137** | 0.691 | **0.703** | 0.258 | 1.015 |
| ZARA2 | ORCA | 0.023 | 0.010 | 0.010 | 0.247 | 0.742 | 1.358 | 0.110 | **0.015** |
| | CrowdES | **0.020** | **0.008** | **0.008** | **0.191** | **0.340** | 0.580 | **0.333** | 2.983 |
| | Ctrl-CWM | 0.026 | 0.012 | 0.012 | 0.258 | 0.581 | **0.569** | 0.306 | 1.457 |
| SDD | ORCA | 0.061 | 0.045 | 0.043 | 0.472 | 0.684 | **5.168** | 0.360 | **0.011** |
| | CrowdES | **0.043** | 0.034 | 0.033 | **0.458** | **0.662** | 6.303 | 0.375 | 1.643 |
| | Ctrl-CWM | 0.046 | **0.033** | **0.031** | 0.476 | 0.718 | 5.511 | **0.383** | 0.227 |
| GCS | ORCA | 6.787 | **0.369** | **0.369** | 15.683 | 2.423 | 5.120 | 0.232 | **0.011** |
| | CrowdES | 4.726 | 0.573 | 0.573 | 10.921 | 0.755 | 2.928 | **0.397** | 9.613 |
| | Ctrl-CWM | **2.847** | 0.483 | 0.483 | **6.578** | **0.503** | **1.766** | 0.327 | 7.645 |
| AVG | ORCA | 1.044 | **0.099** | **0.099** | 2.659 | 1.015 | 2.857 | 0.196 | **0.009** |
| | CrowdES | 0.746 | 0.128 | 0.128 | 1.871 | 0.605 | 2.204 | **0.294** | 3.448 |
| | Ctrl-CWM | **0.477** | 0.111 | 0.111 | **1.244** | **0.545** | **1.713** | 0.285 | 1.739 |

Table 17: **Diffusion-arrival generation results.** Complete values from Table 1. Col. is a percentage. Bold/underline identify the best/second-best distinct displayed values within each benchmark entry.

| Data | Method | Dens.↓ | Freq.↓ | Cov.↓ | Pop.↓ | Kin.↓ | DTW↓ | Div.↑ | Col.↓ |
|---|---|---|---|---|---|---|---|---|---|
| ETH | ORCA | 0.026 | 0.013 | 0.013 | 0.265 | 0.867 | 3.794 | 0.096 | **0.006** |
| | CrowdES | 0.020 | 0.020 | 0.020 | 0.208 | 0.377 | 1.649 | **0.203** | 0.697 |
| | Ctrl-CWM | **0.017** | **0.006** | **0.006** | **0.193** | **0.214** | **1.458** | 0.181 | 0.620 |
| HOTEL | ORCA | 0.022 | 0.010 | 0.010 | 0.199 | 1.052 | 1.588 | 0.112 | **0.013** |
| | CrowdES | 0.013 | **0.009** | **0.009** | 0.117 | 0.336 | 0.643 | 0.242 | 1.197 |
| | Ctrl-CWM | **0.011** | **0.009** | **0.009** | **0.087** | **0.305** | **0.548** | **0.259** | 1.046 |
| UNIV | ORCA | 0.348 | 0.214 | 0.214 | 0.738 | 0.732 | 1.884 | 0.221 | **0.014** |
| | CrowdES | 0.347 | 0.204 | 0.204 | 0.734 | **0.420** | **1.121** | 0.340 | 0.645 |
| | Ctrl-CWM | **0.339** | **0.196** | **0.196** | **0.717** | 0.436 | **1.121** | **0.349** | 0.609 |
| ZARA1 | ORCA | 0.064 | 0.024 | 0.024 | 0.974 | 1.950 | 2.594 | 0.181 | **0.002** |
| | CrowdES | 0.018 | 0.017 | 0.017 | 0.254 | **0.327** | **0.675** | **0.304** | 1.018 |
| | Ctrl-CWM | **0.010** | **0.012** | **0.012** | **0.150** | 0.866 | 0.822 | **0.304** | 0.506 |
| ZARA2 | ORCA | 0.013 | **0.007** | **0.007** | 0.136 | 1.017 | 2.501 | 0.101 | **0.025** |
| | CrowdES | **0.009** | 0.013 | 0.013 | **0.100** | **0.227** | **0.579** | **0.355** | 1.021 |
| | Ctrl-CWM | 0.012 | 0.011 | 0.011 | 0.124 | 0.505 | 0.675 | 0.334 | 1.000 |
| SDD | ORCA | 0.052 | 0.036 | 0.032 | **0.421** | 0.674 | **5.674** | 0.311 | **0.392** |
| | CrowdES | **0.038** | 0.033 | **0.030** | 0.463 | **0.650** | 6.352 | **0.354** | 0.411 |
| | Ctrl-CWM | 0.044 | **0.031** | **0.030** | 0.462 | 0.971 | 5.993 | 0.326 | 1.140 |
| GCS | ORCA | 6.677 | 0.439 | 0.439 | 15.429 | 2.097 | 4.558 | 0.240 | **0.004** |
| | CrowdES | 0.584 | 0.066 | 0.066 | **0.329** | 1.341 | 3.864 | 0.279 | 8.519 |
| | Ctrl-CWM | **0.340** | **0.056** | **0.056** | 0.870 | **1.325** | **1.547** | **0.414** | 6.080 |
| AVG | ORCA | 1.029 | 0.106 | 0.106 | 2.595 | 1.198 | 3.228 | 0.180 | **0.065** |
| | CrowdES | 0.147 | 0.052 | 0.051 | **0.315** | **0.525** | 2.126 | 0.297 | 1.930 |
| | Ctrl-CWM | **0.110** | **0.046** | **0.046** | 0.372 | 0.660 | **1.738** | **0.310** | 1.572 |

Table 18: **Full avoidance-control results.** Each zone block reports $C_{\text{avoid}}$. Baseline obstacles are present from initialization, while Ctrl-CWM activates the objective after a shared first third. AVG is the unweighted mean of the seven entries.

| *Disc* | | | |
|---|---|---|---|
| Data | ORCA + obstacle | CrowdES + map | Ctrl-CWM |
| ETH | 1.000 | 0.599 | 0.837 |
| HOTEL | -1.154 | 0.546 | 0.988 |
| UNIV | 0.530 | 0.727 | 0.831 |
| ZARA1 | 0.428 | 0.469 | 0.752 |
| ZARA2 | 1.000 | 0.598 | 0.714 |
| SDD | 0.344 | 0.317 | 0.877 |
| GCS | 0.999 | 0.436 | 0.827 |
| AVG | 0.450 | 0.527 | 0.832 |
| *Rectangle* | | | |
| Data | ORCA + obstacle | CrowdES + map | Ctrl-CWM |
| ETH | 1.000 | 0.883 | 0.896 |
| HOTEL | -1.484 | 0.808 | 0.881 |
| UNIV | 0.526 | 0.779 | 0.683 |
| ZARA1 | -0.049 | 0.790 | 0.988 |
| ZARA2 | 1.000 | 0.824 | 0.912 |
| SDD | -1.827 | 0.114 | 0.934 |
| GCS | 0.595 | 0.301 | 0.932 |
| AVG | -0.034 | 0.643 | 0.889 |
| *Multiple zones* | | | |
| Data | ORCA + obstacle | CrowdES + map | Ctrl-CWM |
| ETH | 0.973 | 0.479 | 0.992 |
| HOTEL | 0.999 | 0.542 | 0.997 |
| UNIV | 0.601 | 0.549 | 0.993 |
| ZARA1 | -1.952 | 0.781 | 0.983 |
| ZARA2 | -0.304 | 0.751 | 0.971 |
| SDD | -0.473 | 0.525 | 0.908 |
| GCS | 0.768 | 0.424 | 0.808 |
| AVG | 0.087 | 0.579 | 0.950 |