
Supplementary Material for Inner Speech as Behavior Guides

A Algorithms for MIMIC Training and Simulation

This section provides the pseudocode for the algorithms described in Section 3.2.2.

Algorithm 1 MIMIC: Training

Require: Set of human demonstrations \mathcal{D} , Batch size B , Hyperparameters for training

- 1: Obtain demonstration $\in \mathcal{D}$ as images $\mathbf{I}_{1:T}^{(i)}$.
- 2: Inner speech $[m^{(i)} \dots m^{(i+B)}] \leftarrow \text{CLIP}(\text{VLM}([\mathbf{I}_{1:T}^{(i)}, \dots, \mathbf{I}_{1:T}^{(i+B)}]))$.
- 3: Construct \mathcal{D}_M by augmenting $\{m^{(i)}\}$ to \mathcal{D} .
- 4: Train π_θ, Ψ to minimize $\mathcal{L}_{\text{diff}}(\mathcal{D}_M)$ [Eq. 3] and $\mathcal{L}_{\text{is}}(\mathcal{D}_M)$ [Eq. 4] respectively.

Algorithm 2 MIMIC: Simulation

Require: Initial state s_1 of the environment, First update step t_0 , and update window W

- 1: Inner speech $m \leftarrow \mathbf{0}$
- 2: **for** $t = 1$ to T **do**
- 3: **if** $t \pmod{W} \equiv t_0$ **then**
- 4: Past images $\mathbf{I}_{t-W:t}$ from $s_{t-W:t}$
- 5: Sample $z \sim \mathcal{N}(0, I)$.
- 6: Update $m \leftarrow \Psi_{\text{dec}}(z, \mathbf{I}_{t-W:t})$
- 7: **end if**
- 8: Generate the action $a_t \sim \pi_\theta(\cdot \mid s_t, m)$
- 9: Update the state $s_{t+1} \leftarrow \mathcal{E}(s_t, a_t)$.
- 10: **end for**

B Discussions

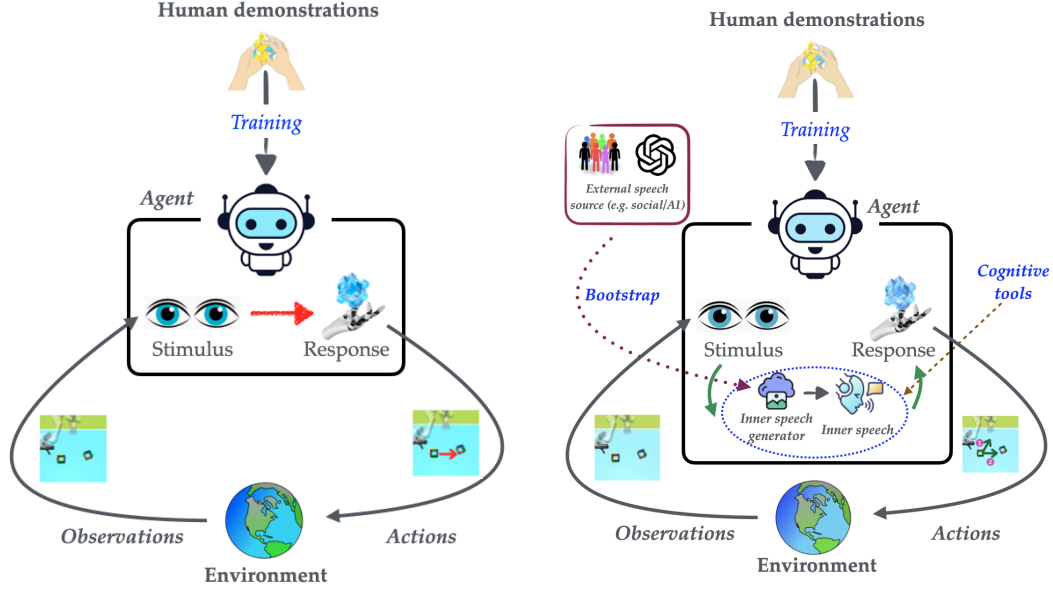
B.1 Inner Speech as a Behavior Guide

MIMIC provides an alternative to the conventional behaviorist framework in IL in the form of a mediated action selection framework that is grounded in cognitive science. The fundamental distinction between the behaviorist and cognitive approaches to IL, as illustrated in Figure 6, represents more than a technical architectural choice. It reflects competing theories of how intelligent behavior emerges.

The behaviorist paradigm (left panel) conceptualizes human action as direct responses to environmental stimuli, where an agent learns a mapping function from states to actions ($s_t \mapsto_{\mathcal{H}} a_t$). This approach, while computationally elegant, treats the human mind as a black box, assuming that behavioral patterns can be fully captured through observed input-output pairs. In contrast, the cognitive approach (right panel) recognizes that human actions are mediated by internal mental processes—what cognitive science literature terms ‘inner speech,’ internalized linguistic structures that guide behavior. Here, the same environmental state can produce diverse actions because it is filtered through an intermediate cognitive layer ($s_t \mapsto m_t \mapsto a_t$), where m represents the inner dialogue that shapes interpretation and response selection. This mediational architecture explains a fundamental observation about human behavior: why different individuals, or even the same individual at different moments, can respond differently to identical situations.

In this way, the cognitive model seeks to capture not just what humans do but to also approximate how they deliberate, through internal linguistic reasoning that weighs options, considers context, and reflects individual motivations. The computational instantiation of this cognitive framework leverages the latent space of inner speech as a principled mechanism for both behavioral diversity and designer control. The continuous latent representation m enables stochastic sampling during inference, naturally inducing behavioral variability that mirrors human decision-making heterogeneity, while simultaneously providing a semantic interface for control—designers can specify desired behaviors through natural language that constrains the latent distribution. Crucially, the vision-language model (VLM) shown in Figure 6 serves as developmental scaffolding, transforming visual observations into external linguistic descriptions that bootstrap the inner speech generator during training.

For artificial agents intended to collaborate with humans, the distinction between these two frameworks is critical: while behaviorist approaches may achieve high task performance, they fail to



(a) Behaviorist framework: Direct stimulus-response mapping (b) Cognitive framework: Linguistically-mediated action selection

Figure 6: Contrasting theoretical frameworks for IL. (a) The behaviorist approach models human behavior as a direct mapping from environmental states to actions ($s_t \mapsto_{\mathcal{H}} a_t$), treating cognitive processes as opaque transformations. (b) The cognitive approach instantiated by MIMIC introduces inner speech as a mediational layer ($s_t \rightarrow m_t \rightarrow a_t$), where m_t represents linguistically-structured internal deliberation that enables behavioral diversity and contextual adaptation.

Approach	Behavioral Diversity	Designer Control	Speech Type	Language Grounding	Language Annotations Required	Latent Space	Conditioning Type
Behavior Transformer [39]	✓ (discrete modes)	×	N/A	×	No	×	Unconditional
Diffusion BC [33]	✓ (continuous)	×	N/A	×	No	✓ (implicit)	Unconditional
BESO [36]	✓ (partial) ¹	✓ (goals only)	External	✓ (goals)	Yes (goals)	✓	Goal-conditioned
Thought Cloning [18]	×	×	External ³	✓ (thoughts)	Yes (per-step)	×	Unconditional
MIMIC (Ours)	✓ (stochastic)	✓ (general)	Internal	✓ (inner speech)	No ⁴	✓ (CVAE)	General linguistic

Table 4: Comparative analysis of IL approaches across key dimensions of behavioral modeling and control. **Legend:** ✓ = Full support; × = No support. **Annotations:** ¹Generates diversity through diffusion but only within goal-constrained trajectories. ²Conditions on fixed human-provided thoughts, limiting emergent diversity. ³Uses linguistic signals that remain external annotations rather than internally generated mediators. ⁴Bootstraps from visual observations using VLM-generated captions, eliminating the need for human language annotations.

generate the behavioral diversity and contextual adaptability that characterize human partners, limiting their effectiveness in real-world collaborative scenarios where understanding and predicting varied human responses is essential. Table 4 distills MIMIC’s unique position as the only approach that achieves language-grounded control without requiring exhaustive human linguistic supervision, instead leveraging vision-language models to automatically generate the necessary training signals from existing visual demonstrations.

B.2 Limitations and Future Extensions

While MIMIC represents a significant advance in cognitively-grounded IL, several limitations warrant consideration for robust deployment across diverse contexts. First, the fidelity of inner speech generation remains contingent upon the quality of linguistic annotations produced by vision-language

models during training, creating a dependency where advances in behavioral modeling are partially gated by progress in vision-language understanding—though notably, improvements in VLM capabilities will naturally enhance MIMIC’s performance without architectural modifications.

Second, the temporal granularity of inner speech generation, controlled through the W parameter, requires careful calibration for different task domains, as excessive polling may induce behavioral instability through frequent re-planning while insufficient polling reduces the framework’s capacity to correct for distributional drift.

Furthermore, while the CVAE’s latent representation enables stochastic behavioral generation, the mapping between latent codes and semantic content remains opaque; post-hoc clustering or CLIP-based projection to natural language recovers partial interpretability but potentially loses nuanced behavioral intentions encoded in the continuous space.

Finally, the framework’s efficacy in complex multi-agent environments with heterogeneous behavioral patterns remains unexplored, and coordination complexity may scale non-linearly with agent count in scenarios beyond dyadic interaction.

Future directions could include exploring semi-supervised approaches leveraging limited human annotations to calibrate VLM-generated captions so as to mitigate data quality dependencies, potentially through active learning frameworks that identify high-uncertainty trajectories for targeted human review. Adaptive polling mechanisms that dynamically adjust temporal granularity based on task complexity or behavioral uncertainty metrics would provide more robust default behaviors while reducing practitioner burden. Developing disentangled latent representations or incorporating discrete latent variables with explicit semantic grounding could enhance interpretability without sacrificing behavioral diversity. For multi-agent scalability, hierarchical inner speech architectures that model group-level intentions alongside individual cognition present a promising direction, enabling agents to reason about collective dynamics while maintaining individual behavioral authenticity.

B.3 Beyond diffusion based behavior policy

While our current implementation employs a diffusion-based behavior policy (DDPM-T), the MIMIC framework is fundamentally model-agnostic. The core insight—that inner speech serves as a stochastic mediator between perception and action—can be instantiated with any conditional behavior cloning architecture. The framework decomposes into two independent components: (1) an inner speech generator $p(m|\mathcal{H}_t)$ that produces linguistic representations from behavioral history, and (2) a behavior policy $p(a|s, m)$ that conditions on these representations. This modular design means the behavior policy can be implemented using transformers (e.g., Behavior Transformer), flow-based models, energy-based models, or even standard supervised learning approaches—any architecture capable of conditional generation.

The key requirement is that the base model accepts an additional conditioning signal. Since inner speech is represented as a continuous embedding $m \in \mathcal{Z}$, it can be naturally incorporated through concatenation with state features, cross-attention mechanisms, FiLM conditioning, or additive conditioning depending on the architecture.

The periodic generation mechanism (parameter W) is similarly architecture-agnostic, as it operates at the simulation level rather than within the model architecture. Any autoregressive policy can maintain a fixed inner speech representation for W steps before regenerating, making this a general inference-time control mechanism applicable across model families. Future work could explore this architectural flexibility to identify optimal policy architectures for different task domains while maintaining the core inner speech framework.

B.4 Cognitive Inspiration vs. Biological Plausibility

Our framework draws computational inspiration from cognitive theory without claiming biological fidelity or neurobiological correspondence. This distinction is crucial: we operationalize functional properties from cognitive theories of inner speech—semantic condensation, predicativity, and temporal regulation—through computational mechanisms (CVAE, transformer attention, diffusion policy) rather than attempting to replicate neural substrates.

This approach follows a productive tradition in AI where psychological theories inform architectural design without requiring neural isomorphism. Convolutional networks leverage principles of hierarchical visual processing without mimicking V1 neurons; attention mechanisms capture aspects of human focus without replicating neural attention circuits. Similarly, MIMIC extracts functional principles from inner speech theory to address behavioral diversity in imitation learning.

Our technical contributions lie in: (1) formalizing inner speech properties through information-theoretic and probabilistic frameworks (Section 3.1), (2) instantiating these properties through specific architectural choices (Section 3.2), and (3) empirically validating that these computational mechanisms improve behavioral fidelity and diversity. We make no claims about whether artificial agents experience phenomenological "inner speech" or whether our architectures replicate human cognitive processes at a mechanistic level.

The value of cognitive inspiration lies in generating testable hypotheses about computational mechanisms—in our case, that introducing linguistic mediation between perception and action can capture human behavioral diversity. Our empirical results validate this computational hypothesis while remaining agnostic about biological implementation.

B.5 Broader Impact

The development of cognitively-grounded artificial agents through MIMIC presents opportunities as well as ethical considerations for human-AI collaboration. The capacity to generate behaviorally-realistic human surrogates enables comprehensive pre-deployment safety validation, potentially preventing harmful interactions in high-stakes domains such as healthcare and autonomous systems. By incorporating linguistically-mediated control mechanisms, MIMIC also enhances transparency in AI decision-making while modeling cognitive diversity through stochastic inner speech generation—facilitating more inclusive systems that account for varied cultural reasoning patterns and individual differences in collaborative scenarios.

However, the same sophistication in replicating human-like behavioral patterns also introduces novel risks requiring careful governance. The ability to generate convincing behaviors could enable sophisticated social engineering attacks or deceptive AI personas designed to exploit human trust. When functioning correctly, such technology might enable unauthorized behavioral profiling; when producing incorrect outputs, it could generate inappropriate social behaviors violating cultural norms; and through intentional misuse, it could facilitate manipulative agents targeting human cognitive vulnerabilities. Additionally, biases embedded in vision-language models used for bootstrapping could perpetuate societal inequities if generated inner speech reflects discriminatory patterns in training corpora.

Mitigation strategies should focus on balancing innovation with principles of transparency and accountability. Disclosure of artificial agency through technical markers in inner speech generation could prevent deceptive practices. Establishing auditable logs of cognitive mediation processes would enable post-hoc analysis supporting accountability in high-stakes applications. These kinds of mitigations would help to promote further advances in cognitively-grounded AI enhancing rather than undermining human agency in increasingly automated societies.

C Extended Related Work

Imitation learning. IL algorithms are commonly organized into *behavior cloning*, *inverse-reinforcement learning*, and *distribution-matching* families [20, 32, 14]. Classic behavior cloning (BC) regresses actions from expert states; the seminal *ALVINN* system kept a vehicle in its lane by copying recorded steering commands [34]. Covariate-shift issues in BC motivated Dataset Aggregation (DAGger) [38], which iteratively queries experts on states visited by the learned policy to mitigate distribution mismatch. *Inverse-reinforcement learning (IRL)* infers a reward explaining the demonstrations [31, 1], while *generative adversarial IL (GAIL)* matches occupancy measures via an adversarial game [15]. The discriminator in GAIL can be interpreted as a potential-based reward, linking it back to IRL [12]. *Hierarchical IL* discovers latent sub-policies that can be sequenced for long-horizon manipulation [11]. While these approaches model imitation as direct mappings from states to actions or through inferred rewards, MIMIC introduces inner speech as a mediational

mechanism between perception and behavior, enabling both distributional matching and designer control through linguistic intervention—a capability absent in traditional IL paradigms.

Diverse behavior imitation: from single mode to multimodal. Human demonstrations are multimodal. Recognizing this, InfoGAIL augments GAIL with an information term so a latent captures hidden styles [25], employing mutual information maximization to discover discrete behavioral modes within expert demonstrations.

Variational methods [45] learn conditional VAEs to embed motor skills, allowing one-shot imitation by sampling different latent states. Such approaches encode behavioral diversity through continuous latent representations that can be manipulated to generate novel skill variations. A hierarchical VAE extends this idea to multi-scale variation [11], decomposing complex behaviors into temporal hierarchies where higher-level latent representations control long-horizon strategies while lower-level latent representations capture execution details. Sequence models such as Behavior Transformers tokenize continuous actions and clone k distinct modes using prompt tokens [39], leveraging the transformer architecture’s capacity for in-context learning to capture multimodal action distributions through discrete behavioral prototypes.

Score-based approaches fit diffusion models directly on trajectories, reproducing the full joint-action distribution [33]. These methods model the entire behavioral manifold through iterative denoising processes, achieving high-fidelity reproduction of demonstration diversity. BESO shows the same mechanism can be goal-conditioned with only three denoising steps [36], demonstrating computational efficiency while maintaining distributional expressiveness through accelerated diffusion sampling. These techniques broaden behavioral variety, but explicit control over which behavior type will appear at test time remains limited. This is a gap addressed by MIMIC’s language-grounded inner speech. MIMIC builds on diffusion BC but conditions the denoising step on an inner-speech vector learned via a vision–language scaffold, enabling fine-grained linguistic control without retraining.

Imitation learning for studying Human–AI coordination. The deployment of AI agents in collaborative human environments necessitates computational approaches that transcend purely algorithmic optimization to encompass the full spectrum of human behavioral patterns and coordination dynamics. In multi-agent coordination domains, empirical evidence demonstrates the critical role of human behavioral modeling: in *Overcooked*, self-play agents confuse human partners due to convergence in self-play to non-human equilibria, whereas agents fine-tuned after first imitating human gameplay coordinate effectively [5]. Similarly, in *Hanabi*, monte-carlo search regularized with a human-behavior prior achieves high human-partner win rates by incorporating human-like suboptimality and communication patterns. These systems employ a two-stage pipeline—learn a human model, then train a best response—yet this segregation introduces computational inefficiency and potential misalignment between the human model and the coordination policy.

The effectiveness of the above approaches is constrained by data availability: existing datasets exhibit significant limitations in capturing behavioral diversity. D4RL offers benchmark tasks but its demonstrations stem from synthetic experts, limiting stylistic variety [13]; RoboNet amasses large tele-operation sets yet focuses on narrow table-top primitives [8]; CALVIN provides language supervision but shows a single canonical solution per goal [26]; and *Overcooked* traces are short and stylistically homogeneous [5]. D3IL deliberately captures multiple human strategies for each manipulation task, making it a rare test-bed for diversity [21]. The BabyAI dataset is another exception, in providing explicit thought annotations for every action, enabling *thought cloning* [18] to learn from paired action-thought demonstrations. However, this kind of linguistic supervision is expensive and its availability is rare.

MIMIC addresses the architectural and data challenges: it unifies the two-stage pipeline as the diffusion policy trained by imitation already exhibits human-like variability and can serve directly as the partner model during new-agent optimization, while mitigating data bottlenecks by using automatically generated captions to train its inner-speech generator on existing video-only corpora—effectively bootstrapping linguistic mediation from visual demonstrations without requiring the exhaustive human annotation.

Language Interfaced Imitation Learning. *Thought cloning (TC)* [18] discussed above directly imitates human thoughts but requires access to annotation of actual human thought for each step in the demonstration trajectory. Further, the performance of TC is highly tied with goal (mission) conditioning and degrades by almost 40% in our experiments once the goal (mission) condition

is removed. Similarly, external speech has been used to steer agent behaviors through action re-ranking [30], though these speech mechanisms remain external to the agent. [43] enforce linguistic bottlenecks through auxiliary tasks but through approaches that operate outside the IL paradigm. Closest prior is [46], who frame intra-agent speech as *semi-supervised captioning*: a vision-language captioner is pretrained and then *frozen* to provide auxiliary caption and caption-matching supervision that improves behavior cloning and enables zero-shot object-level generalization with few additional captions. In contrast, we model *inner speech* as an explicit *latent mediator* that *conditions* the policy, i.e., $p(a | s) = \int p(a | s, m) p(m | s) dm$, and we *generate* m *online* from recent history via a conditional VAE. This shift from auxiliary language supervision to mediational control yields *steerable* and *distributionally realistic* imitation (designer-prompted behaviors, periodic refresh) rather than only improved supervision signals.

Cognitive Theories of Inner Speech and Behavioral Diversity. The relationship between inner speech and behavioral diversity has been extensively studied within cognitive psychology and neuroscience, providing rich theoretical foundations for our computational approach. Vygotsky’s seminal work [44] established inner speech as internalized social dialogue that mediates higher cognitive functions, proposing that the transformation of interpersonal communication into intrapersonal dialogue creates a mechanism for behavioral self-regulation. This theoretical framework was subsequently empirically observed by Sokolov [42], who characterized inner speech as possessing distinctive structural and functional properties that differentiate it from external communication.

Contemporary cognitive research has extended these foundations to explain behavioral diversity. Fernyhough’s [10] *dialogic theory* posits that inner speech maintains the dialogical characteristics of interpersonal communication, suggesting that behavioral diversity emerges from the multiplicity of internalized perspectives. As Alderson-Day and Fernyhough [2] note, “inner speech allows for the simulation of multiple action pathways before behavioral execution,” providing a cognitive mechanism for generating diverse behavioral responses to identical environmental stimuli.

Neuroimaging studies [3] have identified neural correlates of inner speech, showing activation in both language production regions and motor planning areas during inner speech episodes. These findings are consistent with the hypothesis that inner speech serves as a cognitive rehearsal mechanism for behavioral alternatives. Morin’s [29] self-regulatory framework further proposes that inner speech functions as a behavioral selection mechanism, where verbalized thoughts act as “cognitive filters” that modulate action selection based on contextual factors beyond immediate environmental stimuli.

This theoretical perspective aligns with empirical observations in human imitation learning. Meltzoff’s [27] “like me” framework demonstrates that human imitation is not merely mimicry but rather an inferential process that reconstructs the intentions and mental states underlying observed actions. The diversity in imitative behavior derives from this reconstructive process, where different individuals generate different internal models of the demonstrator’s cognitive states. Our computational architecture operationalizes this cognitive process, modeling how inner speech mediates between observation and action to produce diverse yet contextually appropriate behaviors.

AI Approaches to Modeling Cognitive Processes. Some AI research has explored computational implementations of cognitive processes. [6] propose “autotelic AI” that internalizes language for self-directed learning, emphasizing language as a tool for goal generation and intrinsic motivation. While sharing our interest in cognitive foundations, autotelic systems focus on autonomous learning rather than behavioral diversity, using language primarily for goal generation. [19] utilize large language models to simulate reasoning processes (“chain of thought”) before action selection, implementing serial, deterministic reasoning rather than the stochastic, parallel processing characteristic of inner speech in Vygotskian theory. Our framework models inner speech as a probabilistic process generating diverse behavioral patterns from identical environmental states, more closely aligning with cognitive theories of human behavioral diversity. [4] propose natural language as a latent space for reinforcement learning, using language to structure behavior hierarchically. While this approach shares with us the adoption of language as a cognitive tool, it focuses on decomposing complex tasks rather than generating behavioral diversity. Our framework uniquely combines the stochastic nature of inner speech with IL to capture a wide spectrum of human behavioral variation without requiring explicit linguistic supervision.

D Technical Background

In this section, we provide a detailed background on diffusion models and conditional variational autoencoders, which constitute the backbone of MIMIC’s architecture.

D.1 Diffusion Models

Diffusion models, specifically *denoising diffusion probabilistic models (DDPMs)*, constitute a class of generative models that transform noise distributions into target data distributions through iterative denoising processes. Their theoretical foundation derives from non-equilibrium thermodynamics and Markovian diffusion processes, establishing a principled approach to generative modeling through progressive noise injection and removal [17, 9, 40].

The diffusion framework comprises two fundamental stochastic processes operating in complementary directions. The **forward diffusion process** defines a Markov chain that incrementally incorporates Gaussian noise according to a predefined variance schedule, systematically destroying the data structure. Given data $\mathbf{x}_0 \sim q(\mathbf{x})$, this process generates increasingly noisy latents through:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (5)$$

where $\{\beta_t\}_{t=1}^T$ represents the noise schedule with $0 < \beta_t < 1$. The noise schedule can follow various strategies including linear, cosine, or learned schedules, each affecting the quality-efficiency trade-off during generation. This formulation admits a tractable closed-form expression for any timestep:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (6)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. As $T \rightarrow \infty$ with an appropriate schedule, \mathbf{x}_T approximates an isotropic Gaussian distribution, effectively erasing all information about the original data.

The **reverse diffusion process** recovers the original data distribution through learned denoising transformations, parameterized as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \quad (7)$$

where $p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. Following established practice, the variance is typically fixed as $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}$, with σ_t^2 either learned or set to β_t or $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. The mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ is parameterized through a neural network that predicts the noise component:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right), \quad (8)$$

where $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ predicts the noise added during the forward process. This parameterization establishes a fundamental connection to score-based generative models, as the predicted noise is proportional to the score function $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$.

The training objective, derived from variational inference principles, minimizes the negative evidence lower bound (NELBO). However, empirical investigations demonstrate that a simplified objective yields superior practical results:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t \sim \mathcal{U}[1, T], \mathbf{x}_0 \sim q(\mathbf{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2]. \quad (9)$$

This formulation enables efficient training through direct noise prediction across all timesteps simultaneously, while sampling necessitates iterative denoising from pure noise.

D.2 Conditional Variational Autoencoders

Conditional variational autoencoders (CVAEs) extend the traditional VAE framework by incorporating conditional information into the generative process, thereby enabling controlled generation based on specified attributes, contextual constraints, or structural specifications [41, 23]. This conditional paradigm addresses the fundamental limitation of standard VAEs in providing explicit control over generated outputs, establishing CVAEs as particularly valuable for applications demanding targeted generation capabilities.

CVAEs introduce a conditioning variable \mathbf{c} to model the conditional data distribution $p(\mathbf{x}|\mathbf{c})$ through a latent variable framework. The generative process is formulated hierarchically as:

$$p_{\theta}(\mathbf{x}|\mathbf{c}) = \int p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{c}) p_{\theta}(\mathbf{z}|\mathbf{c}) d\mathbf{z}. \quad (10)$$

The conditioning mechanism operates at multiple architectural levels: influencing the prior distribution $p_{\theta}(\mathbf{z}|\mathbf{c})$, the likelihood $p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{c})$, or both components simultaneously. Different conditioning strategies yield distinct modeling capabilities, ranging from simple attribute control to complex structural generation tasks requiring sophisticated conditional dependencies.

Since direct computation of the posterior $p_{\theta}(\mathbf{z}|\mathbf{x}, \mathbf{c})$ remains intractable, CVAEs employ variational inference with an approximate posterior $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c})$, yielding the conditional evidence lower bound (ELBO):

$$\log p_{\theta}(\mathbf{x}|\mathbf{c}) \geq \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c})} [\log p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{c})] - D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c}) \| p_{\theta}(\mathbf{z}|\mathbf{c})). \quad (11)$$

The conditional prior $p_{\theta}(\mathbf{z}|\mathbf{c})$ can be parameterized as a learned function of the conditioning variable, enabling the model to adapt the latent space structure based on conditional information. This adaptability fundamentally distinguishes CVAEs from simpler conditional generation approaches that merely concatenate conditions with inputs.

Neural networks parameterize both the encoder $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c})$ as a conditional Gaussian distribution and the decoder $p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{c})$, which reconstructs inputs based on both latent variables and conditioning information. The encoder produces conditional distributional parameters:

$$q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\phi}(\mathbf{x}, \mathbf{c}), \text{diag}(\boldsymbol{\sigma}_{\phi}^2(\mathbf{x}, \mathbf{c}))). \quad (12)$$

The reparameterization trick facilitates gradient-based optimization through:

$$\mathbf{z} = \boldsymbol{\mu}_{\phi}(\mathbf{x}, \mathbf{c}) + \boldsymbol{\sigma}_{\phi}(\mathbf{x}, \mathbf{c}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (13)$$

The complete training objective minimizes the negative conditional ELBO:

$$\mathcal{L}_{\text{CVAE}} = -\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c})} [\log p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{c})] + D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{c}) \| p_{\theta}(\mathbf{z}|\mathbf{c})). \quad (14)$$

This objective balances reconstruction fidelity against latent space regularization while incorporating conditional constraints. The KL divergence term encourages the approximate posterior to remain proximate to the conditional prior, enabling meaningful interpolation within the conditional manifold and ensuring that latent representations respect the conditioning structure.

E Experimental Setup

E.1 Additional Environment details

Figure 7 (a,b,c) illustrates the D3IL⁵ environments used in our experiments. We use the 4-box and vision-based setting for the Sorting environment, as we observed more stability and higher performance in the BC model under these settings. The following outlines some brief details on those environments.

Aligning: The Aligning task requires the robot to precisely manipulate a box such that it aligns with a target box within specified tolerances, with the constraint that colors must match for each side. The task admits two distinct behavioral modalities: pushing from the inside or from the outside of the box configuration, thereby introducing controlled multi-modality in the action space. The state representation encompasses end-effector position in Cartesian space, pushing box position and quaternion, and target box position and quaternion, with actions represented as desired Cartesian velocities. This task exemplifies the challenge of precision control under multi-modal behavioral strategies, requiring policies to master fine-grained manipulation while maintaining behavioral diversity.

Sorting: The Sorting task requires the robot to sort red and blue blocks into their color-matching target boxes, with task complexity scaling from 2 to 6 blocks. For the 6-block variant, the task exhibits

⁵<https://github.com/ALRhub/d3il> (MIT License)

20 distinct behaviors and demands complex manipulation sequences with high variation in trajectory lengths, challenging existing IL approaches. The state representation includes end-effector position, all boxes’ positions and tangent of Euler angles along the z-axis, with dimensionality scaling linearly with the number of objects. This environment tests an agent’s capacity to handle combinatorial complexity and maintain closed-loop sensory feedback across extended manipulation sequences.

Stacking: The Stacking task requires the robot to sequentially stack 1-3 blocks in a designated yellow target zone, employing a parallel gripper and augmented reality control interface for enhanced dexterity. The state representation includes robot joint positions, gripper width, and boxes’ positions with Euler angle tangents, while actions encompass both joint velocities and gripper width control. Success criteria demand not only lateral positioning within the target zone but also appropriate vertical heights confirming successful stacking. This task represents the pinnacle of manipulation complexity in the D3IL suite, requiring precise grasp-place sequences, dynamic stability maintenance, and adaptive recovery from perturbations.

Figure 7 (d,e,f) illustrates the Overcooked⁶ environments used in our experiments. **Note:** We use the term “Greedy agent” to report results for Overcooked environments, however, this agent is the same as the human proxy agent (split of the trajectories collected from humans) as reported in [5].

Cramped Room: Agents must navigate a confined workspace while executing sequential cooking tasks. The constrained spatial topology induces frequent collision possibilities, necessitating real-time trajectory adaptation and implicit coordination protocols that emerge through embodied interaction rather than explicit communication. The environment’s state space encompasses agent positions, object locations, and cooking progress indicators, with actions comprising discrete movement commands and object interactions. This layout operationalizes fundamental questions about emergent coordination strategies in spatially constrained multi-agent systems, where optimal policies must balance task efficiency against collision avoidance through anticipatory modeling of partner trajectories.

Asymmetric Advantages: The Asymmetric Advantages layout tests whether agents can develop high-level strategic reasoning that leverages differential access to resources, as players begin in distinct spatial regions with asymmetric proximity to cooking stations. This environmental structure necessitates role specialization and adaptive task allocation, where agents must infer and exploit comparative advantages based on spatial positioning and partner capabilities. The layout embodies game-theoretic coordination challenges where multiple Nash equilibria exist, each corresponding to different role assignments and workflow patterns. Success requires agents to transcend myopic task completion toward globally efficient coordination strategies that emerge through iterated interaction and mutual adaptation.

Coordination Ring: The Coordination Ring layout presents a topologically constrained environment where the ring-like spatial structure forces agents to establish and maintain directional conventions (clockwise or counterclockwise movement) to prevent deadlock scenarios. Agents must rapidly converge on shared behavioral protocols without explicit communication channels. The circular topology creates a coordination game with multiple equilibria, where misaligned conventions result in systematic inefficiencies through blocking behaviors. This layout thus serves as a minimal testbed for studying how artificial agents can develop and adapt to emergent social conventions, mirroring fundamental processes in human social coordination where arbitrary but stable behavioral patterns facilitate collective action.

E.1.1 Environment descriptions for the inner speech prompt

Aligning. The GIF(s) show a {camera} view of your actions and your task was to move a box starting from different positions using a robotic hand to align with the other box, which is fixed.

Sorting. The GIF(s) show a {camera} view of your actions where your goal was to sort red and blue blocks to their color-matching target box.

Stacking. The GIF(s) show a {camera} view of your actions where your goal to stack blocks with different colors in a (yellow) target zone.

⁶https://github.com/HumanCompatibleAI/overcooked_ai (MIT License)

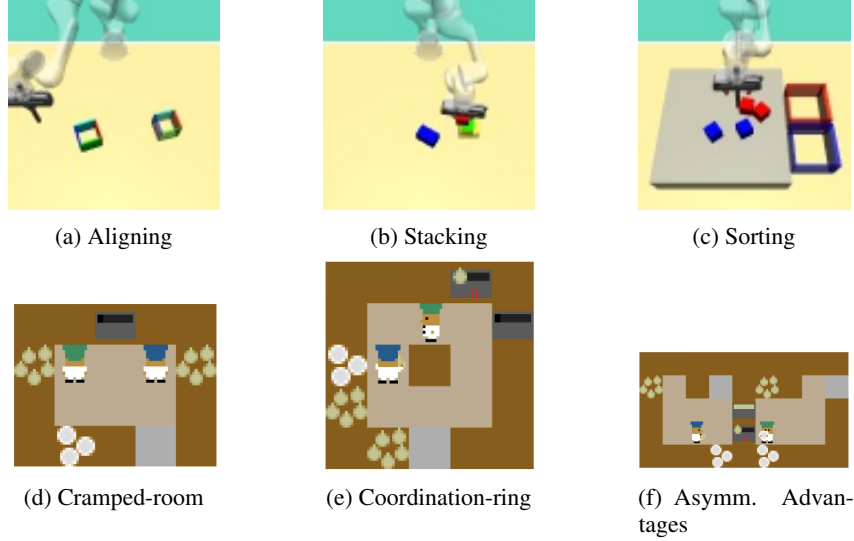


Figure 7: Environments used in our experiments. **(a-c)**: D3IL environments Aligning, Stacking and Sorting environments, respectively. **(d-f)**: Overcooked map layouts Cramped-room, Coordination ring, and Asymmetric Advantage, respectively.

Overcooked: The goal is to place three onions in a pot (dark grey), take out the resulting soup on a plate (white) and deliver it (light grey), as many times as possible within the time limit. The GIF(s) show how the {agent} moves and interacts with other agents in a cramped room.

E.2 Conditional evaluation

Evaluate the ability of the caption to describe the overall motion in the gif from 1 to 5.

Also give your reasoning. The caption is deliberately succinct and ignores the small motion so do not consider these points while evaluating.

Caption: {caption}

GIF: attached

Score:

E.3 Large language models

We use API version ‘2025-01-01-preview’ for all models and employ structured output API to obtain the behavior descriptions corresponding to each GIF.⁷

E.4 Computational environment

All experiments were conducted on a high-performance computing server, equipped with a 64-core x86_64 processor (128 threads) and 1007 GB of RAM, running Ubuntu 22.04 LTS (kernel 5.15.0-153-generic). For GPU-accelerated computations, we utilized an NVIDIA A100 80GB PCIe GPU with CUDA version 12.6. The experiments were implemented using Python 3.10.14 (conda-forge distribution) with PyTorch 2.7.0.

E.5 Extension to multi-agent speech

A novel implication of MIMIC’s flexibility arises in multi-agent settings where an agent can be conditioned not only on its own inner speech but also on the inner speech of other agents, inspired by

⁷<https://platform.openai.com/docs/guides/structured-outputs>

mirror neuron theory of social cognition [37]. This extension models how inner speech mediates social interaction, allowing agents to adapt their behavior based on their perception of others’ cognitive states⁸. We denote it as MIMIC-MA in our experiments.

F Complexity Analysis

Training. First, generating inner-speech captions with GPT-4o is inexpensive: we make N/B API calls with average context length of ~ 100 for text and $\sim B \cdot (85 + 170n)$ for images (with n tiles of 512×512 px), totaling $\sim N \cdot 170n$ tokens which is about \$ 2 for over 400 trajectories, even with high resolution 2048×2048 images. The CLIP model (~ 0.5 B parameters) is likewise lightweight and can be efficiently used during training to generate the inner speech.

Inference. Let T_{CVAE} and T_{diff} denote one forward pass through the CVAE and diffusion models, respectively. Over a simulation horizon H with window size W , we perform H diffusion passes and H/W CVAE passes, yielding a total complexity of $O(HT_{diff} + H/WT_{CVAE})$. Since both are vision-conditioned with similar runtimes and $H > H/W$, the diffusion term dominates. So, MIMIC adds no inference overhead.

G Additional Experiment Results

We first report the extended results in each environment along with the parameter configurations that correspond to the reported best performance. We then analyze the sensitivity of MIMIC to various hyper-parameters and different VLM models. We conclude with the visualization of behaviors obtained through designer specified control text for the Aligning and Sorting environments.⁹

G.1 How is inner speech represented in the embedding space?

Figure 8 shows the TSNE visualization of the CLIP-encoded inner speech of different environments, as generated using GPT-4o. We find that it tends to cluster together similar behaviors while separating distinct behaviors in this 2D space.

G.2 Which configurations maximize the efficacy of MIMIC?

Robotic Manipulation Task. Table 5 reports the hyperparameters corresponding to the best performance reported in Table 1 for D3IL benchmark. Here, p_{drop} denotes the probability of randomly dropping the m for \mathcal{L}_{diff} . We note that higher update windows are preferred for long horizon environments, such as Stacking and Sorting than Aligning, where smaller update windows (W) gives high performance. Initial steps show more variation while indicating that higher values are preferred in non-vision environments. This means that for the first few steps, the agent takes its action with no inner speech. On the other hand, random dropout probability of inner speech (p_{drop}) is found to be important for the Aligning environment for higher performance while no such dropout is more useful in others.

Overcooked Tasks. Table 6 shows the hyperparameters along with the Wasserstein distance between the actions for the OverCooked environment. We also include the reward collected by the $PPO_{H_{proxy}}$ model from [5]. This approach trains the PPO agent in partnership with the H_{proxy} model, essentially giving it access to ground truth. [5] calls this value the "gold standard" and reports only the PPO agent trained in the presence of an imitator reaching close to the gold standard. The results demonstrate that MIMIC already outperforms BC significantly and reaches closer to or surpasses the gold standard performance, demonstrating the capability of MIMIC agent to collaborate effectively with human proxy model. The results further show that the best hyperparameters often include a low initial step and a high update window with a non-zero dropping of probability. The Wasserstein distance between the generated and training actions to also small, following the trend in Table 1. This showcases the high fidelity of behavior imitation as compared to just task success that MIMIC achieves.

⁸In multi-agent settings, agents can observe each other’s behaviors and infer corresponding inner speech representations, or share inner speech explicitly in cooperative scenarios where communication is available.

⁹For other environments, it was difficult to visualize the behaviors via static images and so we defer them to the gif versions shared along with the source code at our project website.

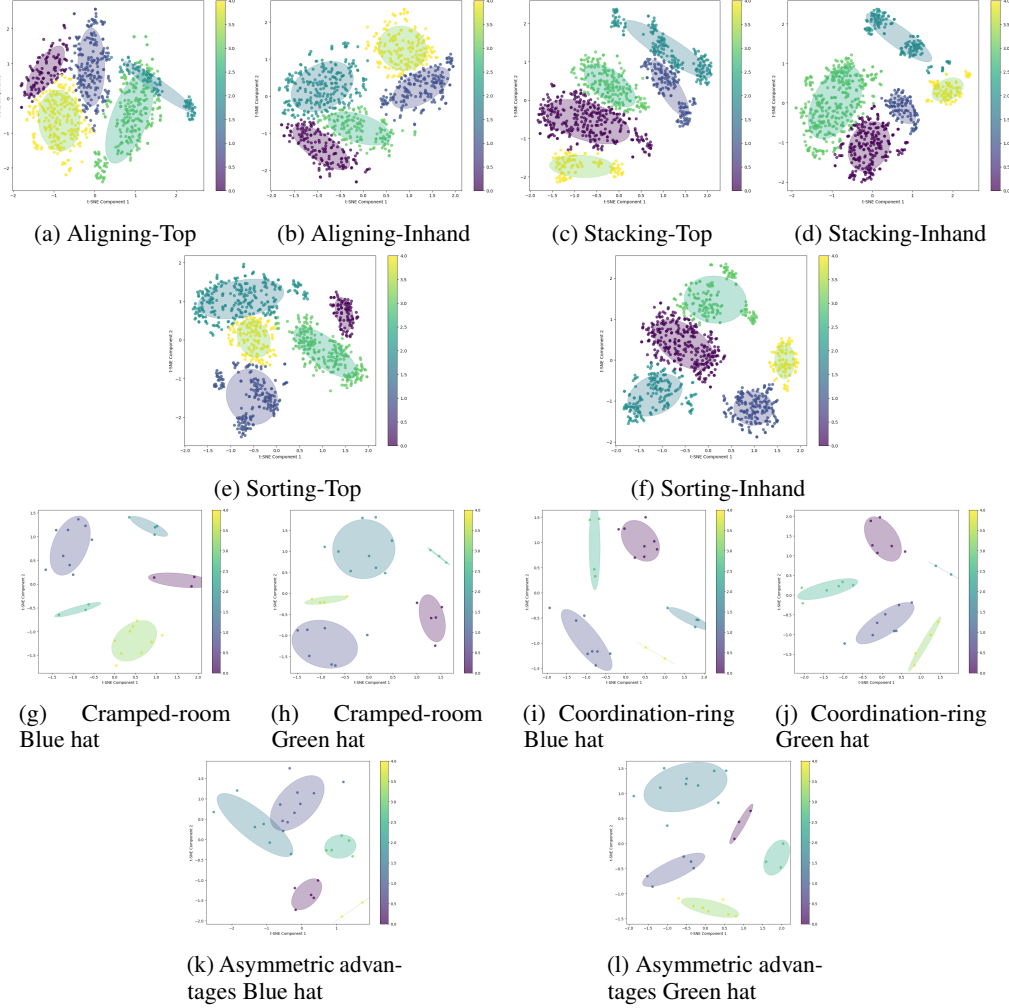


Figure 8: TSNE visualization of CLIP-encoded inner speech generated using GPT-4o for the environments used in our experiments. **(a-f)**: Top and inhand cameras in D3IL: Aligning, Stacking and Sorting environments, respectively. **(g-l)**: Blue and green hat agents in overcooked map layouts: Cramped-room, Coordination ring, and Asymmetric Advantages, respectively.

We also study the setting of using the other agent’s speech (denoted as MIMIC-MA) and find the performance to slightly improve in Coordination-ring, a layout where coordination becomes central to the task. Using the green hat agent’s speech in this case is found to be useful, whereas in all other layouts, the agents collect less reward when using just the blue hat agent’s own speech.

G.3 How sensitive is MIMIC to hyperparameters?

Figure 9 shows how performance varies with change in the initial step and polling window of the simulation in the D3IL benchmark. The performance goes down by delaying the first step of the inner speech update. The performance improves when increasing the update window in the Aligning dataset, but only up to a limit after which the success rate starts going down while the entropy increases. We find that higher polling windows are preferred in Stacking and Sorting environments, while other trends are similar to Aligning.

Figures 10b and 10c shows the hyperparameter sensitivity in Overcooked cramped room environment and we find a similar trend as D3IL benchmark of increasing the performance with increase in the initial step to some extent, while update/polling windows show a drop in performance after a point.

Table 5: Comparison of MIMIC against BC with the DDPM-T architecture on the D3IL benchmark.

Environment	Model	p_{drop}	t_0	W	Success rate \uparrow	Distance \downarrow	Entropy \uparrow
Aligning	BC				0.6645	0.1105	0.4743
	MIMIC-S	0.1	50	50	0.8021	0.0664	0.4184
	MIMIC-E	0.1	12	50	0.7229	0.0847	0.6148
Aligning-Vision	BC				0.1833	0.1875	0.0895
	MIMIC-S	0.1	1	20	0.2229	0.1885	0.0849
	MIMIC-E	0.0	1	20	0.2083	0.1849	0.1473
Sorting-Vision	BC				0.7972	-	0.3596
	MIMIC-S	0.0	100	200	0.8417	-	0.3719
	MIMIC-E	0.0	50	100	0.8083	-	0.4494
					1 box / 2 box	-	1 box / 2 box / 3 box
Stacking	BC				0.8027 / 0.4879	-	0.2058 / 0.1503 / 0.1049
	MIMIC-S	0.0	30	50	0.8129 / 0.6074	-	0.1774 / 0.0737 / 0.0394

Table 6: Comparison of MIMIC against BC with DDPM-T on the Overcooked environments. ‘-’ denotes “action Wasserstein” is not feasible or not available. * denotes values taken directly from [5]. Note that “state Wasserstein” is infeasible due to a large dimension (96) of state features.

Environment	Model	p_{drop}	t_0	W	Collective reward	Action Wasserstein
Cramped room	PPO* _{H_{proxy}}				$\sim 155 - 160$	-
	BC				115.8 ± 3.86	0.24
	MIMIC	0.1	10	100	151.8 ± 2.45	0.25
	MIMIC-MA	0.1	1	50	148.4 ± 2.17	0.25
Cramped room-Vision	BC				73.6 ± 6.18	-
	MIMIC	0.0	1	50	108.8 ± 4.84	-
	MIMIC-MA	0.0	1	20	103.6 ± 3.69	-
Coordination ring	PPO* _{H_{proxy}}				$\sim 145 - 150$	-
	BC				113.0 ± 2.21	0.08
	MIMIC	0.1	10	50	121 ± 1.93	0.09
	MIMIC-MA	0.1	10	20	128.6 ± 1.75	0.03
Asymmetric advantages	PPO* _{H_{proxy}}				$\sim 125 - 130$	-
	BC				215.8 ± 3.04	0.14
	MIMIC	0.1	10	200	227.6 ± 2.69	0.10
	MIMIC-MA	0.1	10	50	227.0 ± 1.84	0.11

We also evaluate the effect of changing the embedding and VLM in the overcooked environment. Figure 10a shows that CLIP-encoded and GPT-4o-scaffolded inner speech is most effective in obtaining the highest collective reward in the Overcooked cramped room. However, we find that even by changing the embedding and VLM, MIMIC still outperforms the BC variant.

G.4 How does MIMIC compare against other strong imitation learning approaches?

While our choice of BC (DDPM-T, [33]) is motivated by its high benchmark performance, we also compare against two additional approaches for comprehensiveness: BESO [36] and BeT [39]. For a fair comparison, we use the BESO’s diffusion model architecture as the underlying policy network in MIMIC instead of a DDPM-T architecture. Table 8 shows that MIMIC substantially outperforms these approaches as well, further highlighting the advantages of using inner speech.

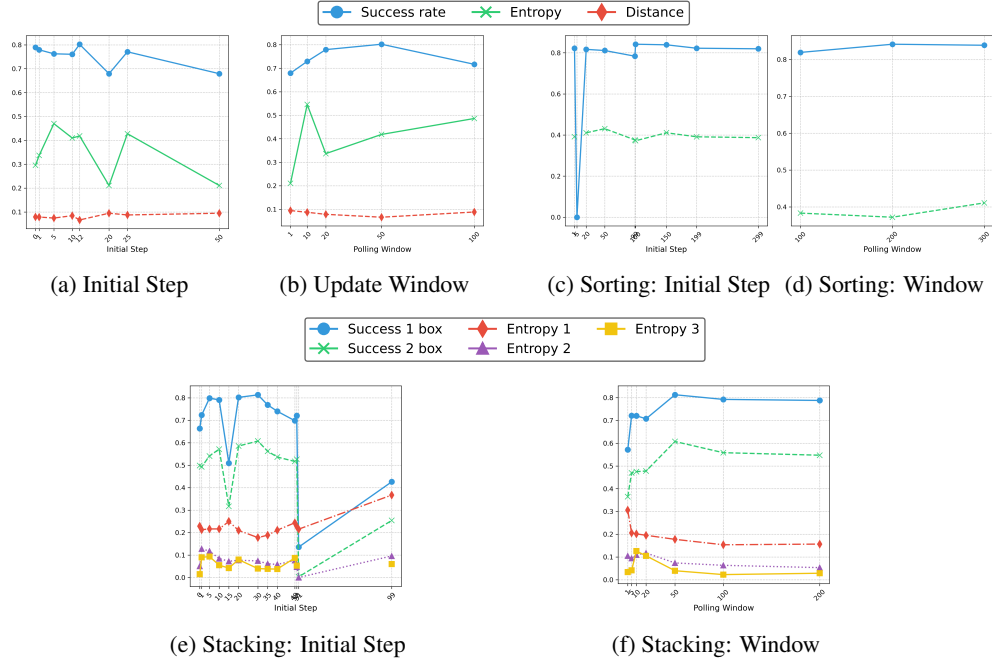


Figure 9: Hyperparameter sensitivity of MIMIC on the D3IL benchmark.

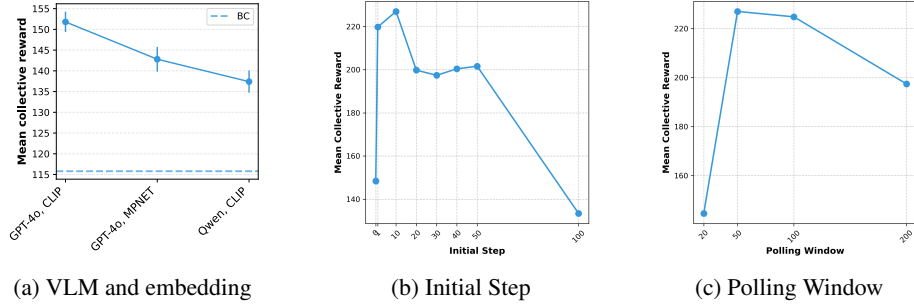


Figure 10: Sensitivity on overcooked Cramped-room

G.5 How efficient is MIMIC during inference?

We confirm the findings of Appendix F empirically by showing in Table 7 that MIMIC’s simulation runtime matches that of DDPM-T BC in vision-based environments. Since MIMIC’s CVAE is vision-based, it would be unfair to compare against non-vision policy networks.

Table 7: Runtime (s) ↓ for different vision environments.

Environment	BC	MIMIC
Aligning	40.69	57.16
Sorting	71.50	78.5
Overcooked	93.23	94.72

G.6 How well does MIMIC enable designer-specified control to generate desired behaviors?

Figure 11 shows examples of different behaviors produced by the MIMIC model conditioned with different descriptions of behaviors. Figure 11a shows a quick repositioning at the start, but due to mediating inner speech, it is not realized later. Figure 11b, on the other hand, shows an attempt to align/match the edges at the start according to the condition. Figure 11c shows an attempt to adjust the box position before pushing straight ahead after a mediation. We find a right side curve approach in Figure 11d, but due to misalignment, it does a lot of rotation at the end. A mediation would have helped here. We find that the zig-zag motion is exhibited in both Figures 11e and 11f while Figure 11f shows more adjustment at the final step as described in the input behavior description.

Table 8: Comparison with other imitation learning models. Here, we use BESO as the base diffusion policy network in MIMIC instead of DDPM-T for fairness.

Aligning				Overcooked cramped room	
Model	Success rate	Distance	Entropy	Model	Collective Reward
BeT	0.51667	0.12949	0.40475	BeT	47.2 ± 4.64
BeSO	0.85417	0.04954	0.6141	BESO	67.8 ± 4.55
MIMIC-S	0.88125	0.04234	0.7215	MIMIC	120.2 ± 2.86
MIMIC-E	0.86875	0.04759	0.7706	MIMIC-MA	141.2 ± 3.68

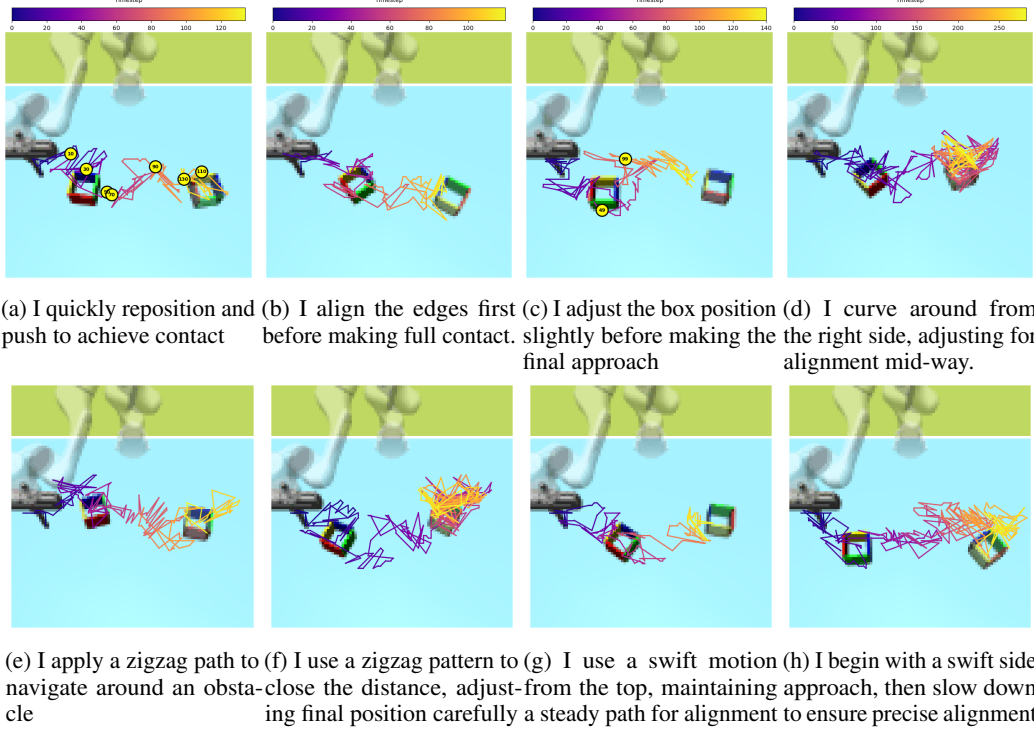


Figure 11: Conditional generation on the Aligning dataset. The color gradient (Plasma) shows the simulation time going from 0 (dark purple) to the end (bright yellow), going through red and orange. Inner speech updates are marked as yellow circles with corresponding times inside; the first inner speech is equal to the specified condition.

On the other hand, Figure 11g shows a swift motion moving from the top and fast convergence to the desired box, as mentioned in the input text while Figure 11h begins with a swift approach but then spends a lot of time in the final alignment with the desired box as mentioned in the text, similar to Figure 11f. These results demonstrate significant success achieved by MIMIC towards enabling steerable imitation of desired behaviors.

We also extend this analysis to the Sorting dataset as we find how it prioritizes the closest red block before any blue in Figure 12a, alternate color sorting in Figure 12b, and a behavior of grouping before moving into desired sorted places in Figure 12c.

G.7 What does generated inner speech look like during simulation?

Since CVAE-generated inner speech lies in the latent embedding space, it is hard to fully interpret and visualize them. We thus employ a heuristic technique to analyze the inner speech during simulation by retrieving the top-2 training descriptions in CLIP’s embedding space at each update step. We use the cosine similarity to find the nearest training description and provide the value in parentheses.

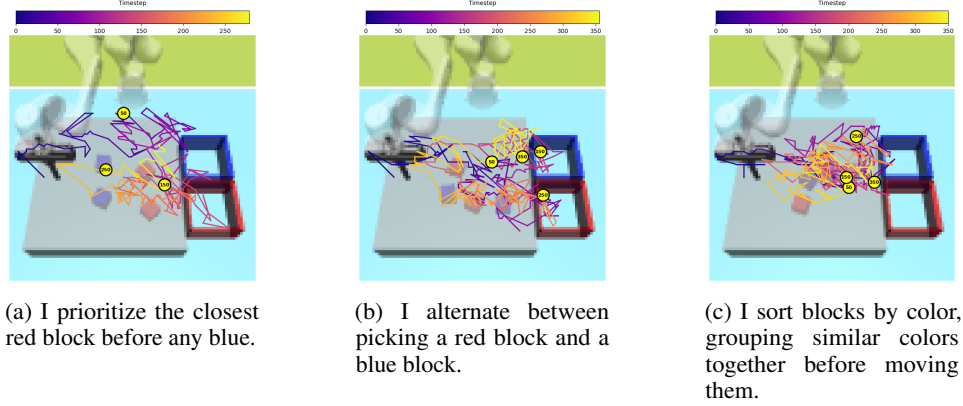


Figure 12: Conditional generation on the Sorting dataset. The color gradient (Plasma) shows the simulation time going from 0 (dark purple) to the end (bright yellow) going through red and orange. Inner speech updates are marked as yellow circles with corresponding times inside; the first inner speech is equal to the specified condition.

Here, we provide examples for both conditional and unconditional simulations and find how the top captions change along with the similarity score.

G.7.1 Conditional

I rotate the box first before aligning it with the target.

Timestep	Closest description	Similarity
t=49	I push the box directly without rotation, aiming for a straightforward alignment.	0.9183
t=99	I approach directly, then rotate in place to align perfectly.	0.9614
t=149	I approach directly, then rotate in place to align perfectly.	0.9614
t=249	Starting from a slightly rotated angle, I need a mid-action adjustment to align.	0.9616

I begin with a swift side approach, then slow down to ensure precise alignment.

Timestep	Closest description	Similarity
t=49	I approach the box from the side, rotating slightly to align smoothly with the fixed box.	0.9471
t=99	I approach directly, then rotate in place to align perfectly.	0.9614
t=149	I approach directly, then rotate in place to align perfectly.	0.9619
t=249	I approach with a direct path but make a last-second adjustment to align perfectly.	0.9611

I use a swift motion from the top, maintaining a steady path for alignment.

Timestep	Closest description	Similarity
t=49	I execute a direct push from behind, minimizing lateral movement.	0.9477
t=99	I carefully approach from the left, ensuring alignment from a diagonal perspective.	0.9613
t=149	Starting from a slightly rotated angle, I need a mid-action adjustment to align.	0.9613
t=249	I approach with a direct path but make a last-second adjustment to align perfectly.	0.9613

G.7.2 Unconditional

Aligning

Timestep	Closest description	Similarity
t=0	I start with a straight approach from a central position, requiring minimal rotation for alignment.	0.9693
t=50	I start with a straight approach from a central position, requiring minimal rotation for alignment.	0.9691
t=100	I start with a straight approach from a central position, requiring minimal rotation for alignment.	0.9696

Sorting

Timestep	Closest description	Similarity
t=100	I focus on sorting all blocks of one color first before switching to the other.	0.9607
t=300	I focus on sorting all blocks of one color first before switching to the other.	0.9604

Overcooked Cramped room

Timestep	Closest description	Similarity
t=100	I quickly grab onions from the pile and place them in the pot, prioritizing speed over precision.	0.9082
t=200	I quickly grab onions from the pile and place them in the pot, prioritizing speed over precision.	0.9096
t=300	I quickly grab onions from the pile and place them in the pot, prioritizing speed over precision.	0.9062
t=400	I quickly grab onions from the pile and place them in the pot, prioritizing speed over precision.	0.9086

Overcooked Coordination ring

Timestep	Closest description	Similarity
t=60	I adjust its movement pattern to avoid congestion, optimizing its task execution efficiency.	0.9521
t=110	I adjust its movement pattern to avoid congestion, optimizing its task execution efficiency.	0.9405
t=160	I adjust its movement pattern to avoid congestion, optimizing its task execution efficiency.	0.9466
t=210	I adjust its movement pattern to avoid congestion, optimizing its task execution efficiency.	0.9407
t=260	I adjust its movement pattern to avoid congestion, optimizing its task execution efficiency.	0.9395

Overcooked Asymmetric Advantages

Timestep	Closest description	Similarity
t=50	I maneuver around obstacles effectively.	0.8391
t=100	I balance interactions with other agents and time efficiency.	0.8416
t=150	I balance interactions with other agents and time efficiency.	0.8463
t=200	I maneuver around obstacles effectively.	0.8443
t=250	I balance interactions with other agents and time efficiency.	0.8401