LEGIBILITY DIFFUSER: OFFLINE IMITATION LEARNING FOR INTENT-EXPRESSIVE ROBOT MOTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In human robot collaboration, legible motion that clearly conveys a robot's intentions and goals is known to improve safety, task efficiency, and user experience. Legible robot motion is typically generated using hand designed cost functions and classical motion planners. But with the rise of deep learning, there is a need for data driven robot policies trained end-to-end on offline demonstration data. In this paper we propose Legibility Diffuser, a diffusion model based policy that learns intent expressive motion directly from human demonstrations. By variably combining the noise predictions from a goal conditioned diffusion model, we are able to guide the robot's motion towards the most legible trajectory. Additionally, we design two empirical studies for automated evaluation of this important property of motion. We find our model is able to optimize for intent expressive motion while maintaining a competitive success rate.

1 INTRODUCTION

Imitation learning (IL) is a powerful paradigm that allows robot policies to be trained on previously collected human demonstrations. Offline IL allows robotics to scale with big data, eliminating the need for costly environment interaction. When training robots for human environments, leveraging offline IL is especially important for safety and effectiveness. For this reason, developing learning from demonstrations (LfD) algorithms that are amenable to HRI is an important avenue of this research. One important characteristic of cooperative robots is legible motion that clearly conveys the robot's intentions and goals. It then seems natural to ask: *How can we directly learn legible robot motion from previously collected human demonstrations?*

As robots become more integrated into our daily lives, it is critical that they move in a way that is not only efficient and functional, but also legible and understandable to humans. In Human-Robot Interaction (HRI), legible motion conveys a robot's intentions and goals in an intuitive and interpretable manner Dragan et al. (2013); Lichtenthäler et al. (2011). Making a robot's actions more transparent will allow humans to better anticipate and respond to the robot. This can reduce the risk of accidents and collisions, which is important in safety critical environments. Concretely, legible robot motion allows an observe to make early and accurate predictions of an agent's target goal. Studies have shown that in collaborative environments, this leads to faster task completion times and more fluent collaboration Breazeal et al. (2005); Dragan et al. (2015).

Mathematically, a legible trajectory is one that maximizes $p(g^*|\xi_{s_0 \to s_t})$ where g^* is the goal and $\xi_{s_0 \to s_t}$ is the ongoing trajectory. Methods for generating legible motion traditionally leverage hand design cost functions and classical motion planning algorithms to maximize this term Dragan et al. (2013). Methods for evaluating legibility involve time consuming human studies where participants predict the target goal and respond to qualitative likert scales. In order for legible robot motion to be a feature of robot policies moving forward, we need deep learning methods that optimize for legibility and tasks on which this property can be benchmarked.

To this end, our paper establishes a connection between legible motion generation and diffusion model guidance. After training a diffusion model conditioned only on goal states, we variably combine noise gradients at evaluation time to produce legible motion. Our method requires access to multi-task and multi-modal data, which are key features of realistic human demonstrations Grauman et al. (2022); Lynch et al. (2020). We do not assume access to reward, constraint, or goal labels. Our other key contribution is designing two emperical studies for automated evaluation of legibility. The first is an object reaching task that is based on classical legibility literature. The second utilizes the Franka Kitchen environment and datasets, but modifies the goal conditions to test a model's capacity to clone the most intent expressive modes from the demonstrations.

Our results show that Legibility Diffuser is able to clone the most legible trajectory from a demonstration dataset while still maintaining a high success rate. We find that in the action generation domain, the class specific features that are enhanced by diffusion model guidance are the same features that make a trajectory legible. We hope that our work will encourage the machine learning community to consider optimizing for legible motion when designing robot policies.

2 Previous Work

2.1 Legible Robot Motion

Shared intentionality is an important aspect of human cognition, and being able to read intentions is critical for how we collaborate as a species Tomasello et al. (2005). Intent expressive actions (legible actions) are a form of non-verbal communication that allows groups of agents to coordinate their behaviors. This is useful for HRI because if robots forecast their next move, they can fluidly interact and improvise with humans Hoffman & Weinberg (2010). Robots are more readable and understandable if they have the capability to express forethought and respond to task outcomes. This increases people's perception of robots and will make users more willing to engage in interactions with legible robots Takayama et al. (2011). Experiments have shown that legible motion allows for faster completion time of collaborative tasks and increased user satisfaction Dragan et al. (2015). Importantly, motion produced by robots can be legible if it allows for quick and confident predictions of the goal state.

Standard methods for generating legible motion Dragan et al. (2013) involve hand designed cost functions as described in section 3.1. With these cost functions, classical motion planners such as Covariant Hamiltonian Optimization Zucker et al. (2013) are able to generate trajectories that maximize $P(g^*|\xi_{s->q})$. In our method, we directly learn this distribution from the training data. The authors of Zhao et al. (2020) use an actor critic approach to train a legible policy in an online setting. In this paper we are specifically interested in learning an end-to-end *offline* policy for legible motion.

2.2 Offline Imitation Learning

Learning effective policies from demonstration data is an important open problem in robotics. There are many benefits to learning from offline datasets such as scalability, portability, and reproducability. These factors are particularly important for deep learning; compiling larger datasets is routinely used to dramatically improve deep vision and language models Deng et al. (2009); Krizhevsky et al. (2017); Devlin et al. (2018); Floridi & Chiriatti (2020); Touvron et al. (2023); Chowdhery et al. (2022). Acheiving similar success with policy generators has proven more difficult. Thus learning from demonstration data is still a commonly used technique. The two main paradigms for learning from offline demonstrations are IL Pomerleau (1988); Zhang et al. (2018); Mandlekar et al. (2020) and Offline Reinforcement Learning (RL) Levine et al. (2020); Lange et al. (2012); Cabi et al. (2019). These algorithms assume access to datasets of state action pairs and reward labels in the case of offline RL. Most formulations of IL assume access to expert demonstrations, but empirical studies have gotten state of the art performance across a variety of tasks even with sub-optimal data Mandlekar et al. (2021); Florence et al. (2022); Hahn et al. (2021).

In this paper we are concerned with training an agent to learn legible motion from multi-modal and multi-task datasets. Multi-modal distributions don't have a singular deterministic action output, rather there can be multiple plausible actions from any given state. Behavioral cloning algorithms with an Recurrent Neural Network backbone (BC-RNN) are known to produce successful manipulation policies when trained on mixed quality demonstrations Mandlekar et al. (2021). Conditional Behavioral Transformers (C-BeT) have even more powerful multi-modal capabilities and can learn policies from unstructured play data Shafiullah et al. (2022); Cui et al. (2022). Recently, Denoising Diffusion Probabilistic Models (DDPMs) emerged as state of the art deep generative models for offline learning Janner et al. (2022); Ajay et al. (2022); Chi et al. (2023). DDPM guidance, described in 3.3, allows for controllable generation at evaluation time and has proven useful for a range of tasks including offline reinforcement learning Ajay et al. (2022), traffic simulation Zhong et al. (2023), and image generation Ho & Salimans (2022). With legibility diffuser, we show that guided generation from diffusion models can produce intent expressive motion.

In the context of HRI, learning from demonstrations (LfD) provides a whole other set of benefits. LfD allows for non-expert programming of desired behaviors through kinesthetic teaching, teleoperation, or passive observation Ravichandar et al. (2020). Because of this, fine-tuning by end users is much simpler and allows for greater adaptability. This is particularly useful when training generative models as they have the capacity for continual learning through techniques like deep generative replay Shin et al. (2017). Another important factor is the safety offered by offline learning. Deep reinforcement learning algorithms have achieved excellent results across a wide range of domains, but they necessitate online interaction with the environment Levine et al. (2020). Deploying agents that learn through environment interaction is dangerous because actions with low reward (such as hitting a human) will be taken in the process of exploration. This danger is mitigated with LfD algorithms because the agent imitates the demonstration; there is no environment exploration.

3 Preliminaries

3.1 Equations for Legible Motion

Mathematically, a legible trajectory ξ from start state s_0 to goal state g^* optimizes the following equation Dragan et al. (2013):

$$legibility(\xi) = \frac{\int p(g^*|\xi_{s_0 \to s_t}) f(t) dt}{\int f(t) dt} \tag{1}$$

Here $f(t)$ is a function of time that puts higher weight on earlier parts of the trajectory. Typically, $p(g|\xi_{s_0 \to s_t})$ is computed using a cost function ζ that models what the observer expects the robot to do:

$$p(g|\xi_{s_0 \to s_t}) \propto \frac{exp(-\zeta[\xi_{s_0 \to s_t}] - v_g(s_t))}{exp(-v_g(s_0))} p(g) \tag{2}$$

Here $v_y(x)$ is the lowest cost path from x to y. In order to maximize $p(g^*|\xi_{s_0 \to s_t})$, one must minimize $p(g \neq g^*|\xi_{s_0 \to s_t})$, i.e., the likelihood of trajectory going for other non-target (opposing) goals. This is done by following an ongoing path $\xi_{s_0 \to s_t}$ such that $v_{g \neq g^*}(s_t) \gg v_{g^*}(s_t)$. For pick and place tasks, experiments have shown Dragan & Srinivasa (2014) that the cost function C is:

$$\zeta[\xi] = \frac{1}{2} \int \xi'(t)^2 dt \tag{3}$$

A trajectory is legible if the cost of reaching an opposing goal is high while the cost of reaching the target goal is low. From this equation, it is clear that longer, slower paths have higher costs. For a pick-and-place task, straight line paths that move quickly toward an object will have a low cost.

While these methods are useful for estimating $p(g^*|\xi_{s_0 \to s_t})$, as generative models get more powerful, directly learning this distribution becomes feasible. So in order to produce legible motion, it should be possible to train a model to generate actions that maximize $p(g^*|\xi_{s_0 \to s_t})$.

3.2 Sequential Decision Making

We view robot action generation as a sequential decision making problem and model it as a discrete-time infinite-horizon Markov Decision Process (MDP), $\mathcal{M} = (S, A, T, R, \gamma, \rho_0)$, where S is the state space, A is the action space, $T(\cdot|s, a)$ is the state transition distribution, $R(s, a, s_0)$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and $\rho_0(\cdot)$ is the initial state distribution. At every step, an agent observes a state s_t and queries a policy π to choose an action $a_t = \pi(s_t)$. The agent performs the action and observes the next state $s_{t+1} \sim T(\cdot|s_t, a_t)$ and reward $r_t = R(s_t, a_t, s_{t+1})$.

We augment this MDP with a set of absorbing goal states $G \subset S$, where $g \in G$ corresponds to a specific state of the world in which the task is considered to be solved. Every pair (s_0, G) of an initial state $s_0 \sim \rho_0(\cdot)$ and goals for a task G corresponds to a new task instance. For the purposes of legibility, we also define a target goal g^* and an opposing goal g^-. We want our motion to g^* to be distinct from motion to any opposing goal such that $P(g^*|s_t) > P(g^-|s_t)$

We assume access to a dataset of N task demonstrations $D = \{\tau_i\}_{i=1}^N$ where each demonstration is a trajectory $\tau_i = (s_{i0}, a_{i0}, s_{i1}, a_{i1}, \ldots, s_{iT})$ that begins in a start state $s_{i0} \sim \rho_0(\cdot)$ and terminates in a final (goal) state $s_{iT} = g_i$.

3.3 Conditional Denoising Diffusion Probabilistic Models

Conditional DDPMs aim to estimate an unknown conditional distribution $q(\mathbf{x}_0|\mathbf{c})$ using a parameterized model $\pi_\theta(\mathbf{x}_0, \mathbf{c})$ based on sampled data $\mathbf{x}_0$ drawn jointly with conditioning information $\mathbf{c}$. The process consists of a *forward noising process* and a *reverse denoising process*. The forward process injects Gaussian noise into samples, producing noised distributions $q_t(\mathbf{x}_t|\mathbf{c})$. The distribution of $\mathbf{x}_t$ based on $\mathbf{x}_0$ is given as $q_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2\mathbf{I})$, with σ_t representing noise levels. During the reverse denoising process, a prediction network ϵ_θ estimates the noise added at time t: $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) \approx \nabla_\mathbf{x} \log q_t(\mathbf{x}_t|\mathbf{c})$. This allows the the uncorrected data $\mathbf{x}_0$ to be recovered using a stochastic differential equation originally laid out in Ho et al. (2020).

$$\mathbf{x}_{t-1} = \phi_t(\mathbf{x}_t - \psi_t \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) + \mathcal{N}(0, \sigma_t^2 I)) \tag{4}$$

Here ϕ_t, ψ_t, and σ_t are hyperparameters of the noise scheduling process that can be tuned. In classifier free guidance Ho & Salimans (2022), the noise score $\bar{\epsilon_\theta}(\mathbf{x}_t, t, \mathbf{c})$ is calculated by combining the noise scores from a conditional estimate $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ and an unconditional estimate $\epsilon_\theta(\mathbf{x}_t, t, \emptyset)$:

$$\bar{\epsilon_\theta}(\mathbf{x}_t, t, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t, t, \emptyset) \tag{5}$$

The unconditional score estimation $\epsilon_\theta(\mathbf{x}_t, t, \emptyset)$ is trained at the same time as the conditional score estimation $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ by setting class conditioning information $\mathbf{c}$ to a null token $\emptyset$ with probability p_uncond. Guidance weight w has the effect of up-weighting the probability of data for which a classifier $p_\theta(\mathbf{c}|\mathbf{x}_t)$ would assign high likelihood to the correct label. This formulation has the benefit of not requiring a classifier trained on partially corrupted data $\mathbf{x}_t$.

4 Legibility Diffuser

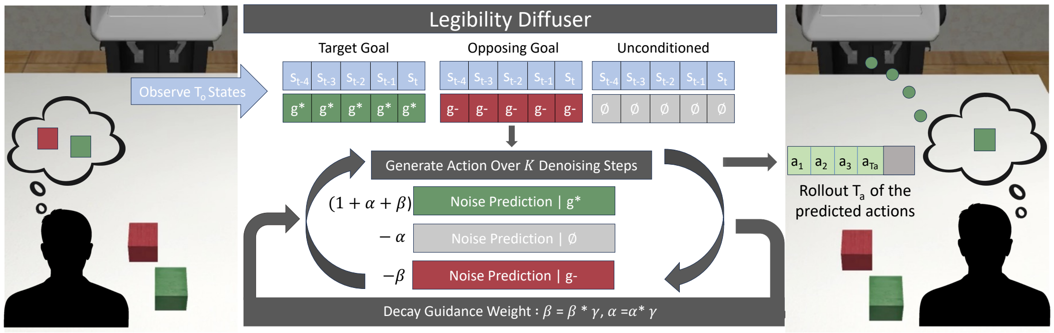
Legibility Diffuser
Target Goal
Opposing Goal
Unconditioned
Observe T_o States
s_{t-4} s_{t-3} s_{t-2} s_{t-1} s_t
g* g* g* g* g*
g- g- g- g- g-
∅ ∅ ∅ ∅ ∅
Generate Action Over K Denoising Steps
$(1+\alpha+\beta)$
Noise Prediction | g*
$-\alpha$
Noise Prediction | ∅
$-\beta$
Noise Prediction | g-
Decay Guidance Weight : $\beta = \beta * \gamma$, $\alpha = \alpha * \gamma$
a_1 a_2 a_3 a_{Ta}
Rollout T_a of the predicted actions

Figure 1: **Legibility Diffuser:** This diagram shows the evaluation process for legibility diffuser. The training process is the same as standard classifier free guidance. At each step, before denoising our action we generate the outputs from or conditional diffusion model. We have an unconditional output that is only conditioned the current state, an output that is additionally conditioned on the target goal state g^*, and a final output conditioned on the opposing goal state g^-. Each of these models is given T_o previous states and T_o repetitions of the goal state (or null character).

4.1 Overview

The key contribution of our paper is framing legible motion generation as a conditional generative modeling problem. As described in section 3.3, a conditional DDPM π_θ estimates an unknown conditional distribution $q(\mathbf{x}_0|\mathbf{c})$ by drawing samples $\mathbf{x}_0$ jointly with class conditioning information $\mathbf{c}$. These models can be guided to generate outputs such that a classifier $p_\theta(\mathbf{c}|\mathbf{x}_0)$ trained on the same samples as π_θ would assign a high likelihood to the correct class label Ho & Salimans (2022). In the setting of robotic policy generation we train a generative model to estimate $q(a_t|s_t, g^*)$, the distribution of actions a_t given the current state s_t and a target goal g^*. Using diffusion guidance, we can generate outputs such that a classifier $p(g|a_t, s_t)$ would assign a high probability to g^*. We notice that this resembles the objective of legible motion generation, i.e. maximizing $p(g^*|\xi_{s_0 \rightarrow s_t})$ (Eq. 2) where $\xi_{s_0 \rightarrow s_t}$ is the ongoing trajectory. In this formulation, (a_t, s_t) can be thought of as a small section (the most recent section) of $\xi_{s \rightarrow \xi(t)}$. Furthermore, in the context of HRI the human our robot is interacting with can be thought of as the classifier $p(g|a_t, s_t)$ that assigns a high probability to $p(g^*)$. This allows the observer to quickly and accurately predict the target goal, leading to all the benefits of legible motion that are critical for the HRI community (sec. 2.1). Below we detail how our method, Legibility Diffuser, is able to leverage this property of diffusion model guidance to generate legible robot motion.

4.2 Conditional Generative Policy Formulation

Legibility Diffuser (Fig. 1) is a DDPM that serves as a visuomotor robot policy. Our work builds upon recent advancements in DDPM-based policies, most notably Diffusion Policy Chi et al. (2023). At each time step, we generate $A_t = \{a_{t:t+\tau_p}\}$ actions for an agent given $0_t = \{s_{t:t+\tau_o}\}$ initial observations and a goal state g. This is done by training a CNN-based diffusion model policy $\pi(A_t|O_t, g)$. Once the actions are generated, the agent carries out $\tau_a < \tau_p$ steps in open loop. During training, our dataset is broken into sequences of length τ_p that include actions, observations, and the goal state. We define g as the final state in the demonstration. We construct our conditioning term by concatenating O_t with g (in practice we use τ repetitions of g). In the style of classifier free guidance Ho & Salimans (2022), we zero out g to train $\pi(A_t|O_t, \emptyset)$ with probability p_{uncond}. Our training procedure follow the standard formulation for DDMP training as laid out in 3.3. Generating legible motion does not require any additional steps during training. At evaluation, we require access to a target goal and opposing goal(s) in order to guide π to generate legible actions.

4.3 Legibility Guidance

We guide our diffusion model π towards the most legible action sequence A_t at each timestep t. To do this, we require access to a target goal g^* that we hope our agent reaches. We also require an opposing goal g^- that we do not want our agent to reach. For our motion to be legible, at every timestep t an observer should be able to predict that the agent is going to g^* and not g^-. In other words, $p(g^*|O_t, A_t) > p(g^-|O_t, A_t)$. We achieve this by guiding our diffusion model during the denoising process.

At each denoising step k, we calculate a noise score conditioned on the target goal $\epsilon_\theta(A_t^k, O_t, g^*, k)$, the opposing goal $\epsilon_\theta(A_t^k, O_t, g^-, k)$, and a null token $\epsilon_\theta(A_t^k, O_t, \emptyset, k)$. We combine these scores in a manner similar to classifier free guidance (Eq. 5) to get a final noise score as follows:

$$\bar{\epsilon}_\theta(A_t^k, O_t, g^*, k) = (1 + \alpha_t + \beta_t)\epsilon_\theta(A_t^k, O_t, g^*, k) - \alpha_t \epsilon_\theta(A_t^k, O_t, \emptyset, k) - \beta_t \epsilon_\theta(A_t^k, O_t, g^-, k)$$

Here, α and β are guidance weights that can be tuned based on the task at hand. The noise score is used to recover an uncorrupted action sequence A_t^0 following the standard stochastic process for DDPMs (Eq. 4). By increasing α, we guide π to generate actions that are distinct from the outputs of an unconditional model. By increasing β, we guide π to generate actions that are distinct from the outputs of a model conditioned on opposing goal g^-. We expect a larger value for β as this should directly lead to actions where an observer would predict $p(g^*|O_t, A_t) > p(g^-|O_t, A_t)$ (sec. 4.1). Empirically, we find that a $\beta \approx 2$ and an α that is an order of magnitude smaller lead to the most legible motion.

4.4 Time-varying Legibility Guidance Decay

Inspired by techniques for legible motion generation in HRI, we introduce a decaying term γ to our guidance weights. This causes legibility guidance to be strongest at the beginning of the trajectory.

Various HRI papers have shown that maximizing legibility is most important at the beginning of a trajectory Dragan et al. (2013; 2015) as we want observers to be able to quickly infer the goal state. The guidance weight only decays after τ_a action steps, it is constant for every denoising step while generating the open loop action sequence of length τ_a:

$$\alpha_{t+1} = \gamma \alpha_t, \; \beta_{t+1} = \gamma \beta_t$$

Empirically, we find that a gamma value slightly under 1 promotes legible motion while maintaining a high success rate.

5 EXPERIMENTS

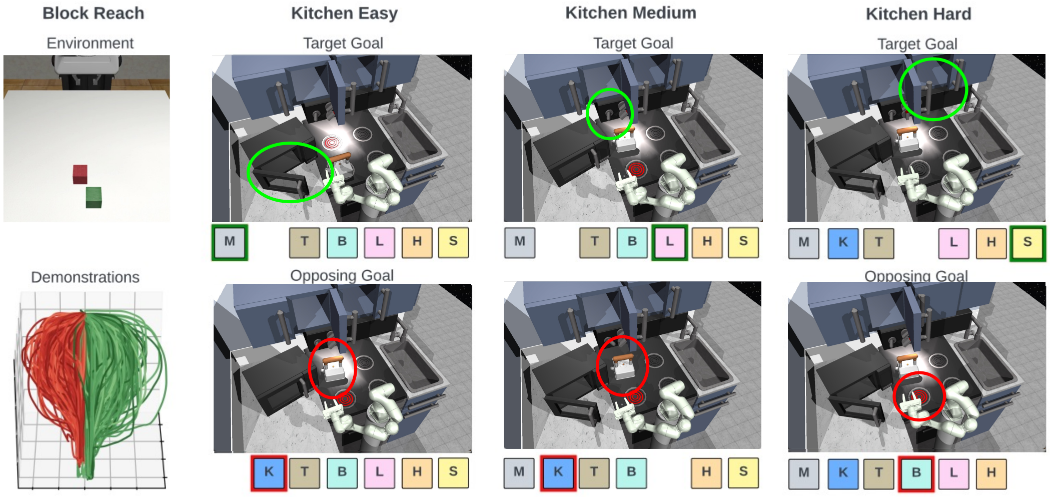

Figure 2: **Task Environments:** This diagram shows the environments that we use in our experiment

Through our experiments, we aim to address the following questions: Can Legibility Diffuser generate the most intent-expressive mode given multi-modal demonstrations? Does optimizing for legibility effect success rate? How important are the individual components of Legibility Diffuser?

5.1 TASKS AND ENVIRONMENTS

Below is an explanation of the tasks we design to evaluate legibility and the environments in which the agent interacts (Fig. 2). Both environments are simulation environments where we have access to low dimensional states and proprioception. For each task, we define a target goal $g*$ that the agent should reach and an opposing goal g^- that the agent should avoid.

Block Reach: We collect a demonstration dataset modeled off the classical legible motion experiment described in Dragan et al. (2013). In this task, an agent reaches across a table to lift up one of two offset block. The proximity of the two blocks makes a straight line path ambiguous for an observe who is trying to predict the target block. A legible robot will take an exaggerated path towards the goal, emphasizing whether it is headed to the left block or to the right block. Our dataset consists of 780 demonstrations that cover a wide range of paths towards each goal. This environment is challenging due to the high level of multi-modality in the demonstrated trajectories.

Franka Kitchen: We use the Franka Kitchen datasets that are originally proposed in Gupta et al. (2019). This environment is challenging due to the multi-modal and long-horizon nature of the human demonstrations. The demonstrations include the robot interacting with four out of the seven items in the kitchen. Importantly, the demonstrations always follow a fixed interaction order:

Microwave(M) → Kettle(K) → TopBurn(T) → BottomBurn(B) → Light(L) → HingeCabinet(H) → SlideCabinet(S)

We conduct three legibility experiments in this environment. For each experiment, we define a target goal and an opposing goal, each including a set of N object interaction subgoals. The target and opposing goals have $N-1$ objects in common. We call the distinguishing object in the target goal the *distinguishing object* and that of the opposing goal as the *opposing object*. For example, for a target goal $\{$M, K, T, B, H, S$\}$ and an opposing goal $\{$M, T, B, L, H, S$\}$, the distinguishing and opposing objects are K (Kettle) and L (Light), respectively. A legible agent should interact with the distinguishing object early in the trajectory so that an observer can quickly identify the target goal. A legible agent should *not* interact with the opposing object as this would confuse the observer, resulting in a wrong goal identification. These experiments pose significant challenges for generalization. Agents attempt to solve longer-horizon tasks than were shown in the training data ($N = 6$ as opposed to $N = 4$) and must interact with objects in a different order than demonstrated to be legible. The target and opposing goals for each experiment are shown in Fig. 2.

Kitchen-Easy: The distinguishing object K (kettle) and the opposing object M (Microwave) are the first two items in the interaction sequence of the training data. This task is considered easy because all the decisions that impact legibility have a short horizon.

Kitchen-Medium: The distinguishing object L (Light) is significantly farther along the interaction sequence than in the easy task. The opposing item K (Kettle) still comes early in the sequence.

Kitchen-Hard: Both the distinguishing object S (SlideCabinet) and the opposing object B (BottomBurn) are late in the training data interaction sequence. This is the most challenging task because all the decision that impact legibiltiy have a long horizon.

5.2 Metrics

Evaluating legibility is a challenging problem. Classically, this is done by recording how long it takes an observer to predict an agent's intended goal (Dragan et al., 2013). Faster prediction times means the motion is more legible. However, such human evaluations are difficult to scale and may be subject to various factors such as observation angles. Hence, in this work, we design methods to measure legibility in an automated manner using experiment-specific metrics. Each of these metrics capture the notion that an observer would predict $p(g^*) > p(g^-)$.

Legibility - Block Reach: We evaluate the legibility for the block reaching task using a distance based heuristic. Many experiments have shown that for pick and place tasks, legible trajectories maximize the distance to the opposing goal (Dragan et al., 2013). We draw inspiration from the cost functions defined in these papers and calculate the legibility using the following equation.

$$L(\xi_{s_0 \to g^*}) = \sum_{s_i \in \xi_{s \to g^*}} \frac{||g^- - s_i||_2}{i} \tag{6}$$

The legibility values reported are normalized based on the minimum and maximum values in the training dataset.

Legibility - Franka Kitchen: To evaluate legibility in this environment, we record the ordinality of the distinguishing object in the roll-out as x_d, and the ordinality of the opposing object as x_o, the legibility score of one rollout can then be defined as $\frac{1}{x_d} \cdot \mathbb{1}_{x_d < x_o}$, where $x_d \leftarrow \infty$ if the agent never interact with the distinguishing object. Failure to interact with the distinguishing object results in a zero score for the roll-out, and engaging with the opposing object before the distinguishing one also leads to a zero score. For example, in Kitchen-Medium the distinguishing object is L and the opposing object is K. Roll-outs M → L → B → H and M → L → S both get a legibility score of $\frac{1}{2}$. Roll-outs K → L → B → H and M → T → H both get a legibility score of 0.

Success Rate: Our model is focused on producing legible motion, not optimizing for success rate, but we evaluate this metric as well for the sake of analysis. We define a rollout to be successful if the conditioned goal is reached before the step limit. For the block reaching task, we define the step limit to be the length of the longest demo in the demonstration dataset. For Franka Kitchen, each goal contains six items. We separately report the success rate for reaching 1 - 4 of these items (the horizon is not long enough for an agent to interact with six items). The success of interacting with i items can be calculated as $\mathbb{1}[(|\mathbb{S}| - \mathbb{1}_{x_o \neq \infty}) \geq i]$, where $\mathbb{S}$ is the set of items that are reached during

	Block Reach	Kitchen-Easy	Kitchen-Med	Kitchen-Hard
Diffusion Policy	.36	.9	.12	.03
CFG Diffusion	.31	**1**	.13	.02
Legibility Diffuser (Ours)	**.73**	**1**	**.2**	**.09**
Ablation - No Decay	-	1	.17	.09
Ablation - No g^-	-	.98	.1	.07

Table 1: Model legibility.

the rollout. Interacting with the opposing object does not count towards the success rate as it is not part of the conditioned goal.

5.3 Baselines and Ablations

Legibility Diffuser is an offline IL algorithm trained on multi-modal, multi-task datasets. We compare our model to two other diffusion model variants that are known to effectively generate policies from such data. Each model is goal conditioned in the same way as Legibility Diffuser - concatenating the target goal state to the current state. These baselines allow us to evaluate if our method is actually optimizing for legibility to a greater extent than standard diffusion models.

Diffusion Policy (Diff Policy): Diffusion Policy is the DDPM that Legibility Diffuser is built upon, it is known to effectively capture the multi-modality in a dataset. This baseline isolates the effect of our novel contributions.

CFG Diffusion: We implement a DDPM with vanilla classifier free guidance. Although the implementations differ, this baseline draws inspiration from Decision Diffuser Ajay et al. (2022). Namely, CFG Diffusion models $p(a|s)$ instead of $p(a, s)$ and conditions on goal states as opposed to class/constraint/reward labels. We base the guidance weights for CFG Diffusion off the values reported for Decision Diffuser.

Ablation 1 (No Decay): For this ablation we evaluate the role of the decaying guidance weights. We keep the same parameters as Legibility Diffuser, but we set $\gamma = 1$.

Ablation 2 (No g^-): For this ablation, we evaluate the role of guidance away from g^-. This is classifier free guidance with a decaying guidance weight. We use the same parameters as Legibility Diffuser, except we combine the guidance weights $w = \alpha + \beta$ and only apply this to the null conditioned noise score.

6 Results

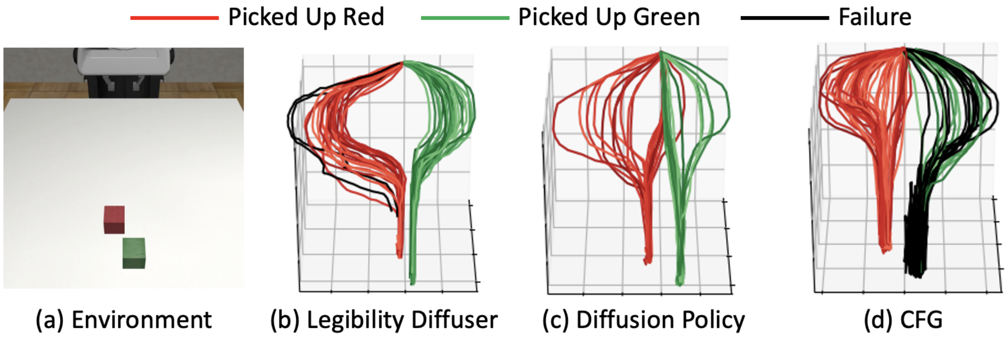

Figure 3: Visualization of task rollouts for Legibility Diffuser and baselines.

		Number of Conditioned Tasks Completed											
	Block Reach	**Kitchen-Easy**				**Kitchen-Med**				**Kitchen-Hard**			
		1	2	3	4	1	2	3	4	1	2	3	4
Diffusion Policy	1	1	.98	.95	.8	1	.98	.95	.8	1	.95	.93	.8
CFG Diffusion	.83	1	1	.85	.8	1	1	.85	.8	.93	.9	.75	.53
Legibility Diffuser (Ours)	.98	1	1	.93	.85	.95	.93	.68	.6	1	.9	.43	.3
Ablation - No Decay	-	1	.98	.9	.7	.85	.75	.53	.48	1	.9	.43	.3
Ablation - No g^-	-	.98	.88	.65	.45	.98	.88	.65	.45	0.88	.73	.45	.25

Table 2: Model Success Rates

6.1 Does Legibility Diffuser imitate legible modes from the demonstrations?

We find that on all tasks, Legibility Diffuser is able to imitate a more legible mode than any of the other baselines. For the block reaching task, our model avoids paths that go straight towards the target block. This is what we would expect based on how legible trajectories for this task are classically designed Dragan et al. (2013). For Kitchen-Easy, all of the models perform well which is also expected. Legibility Diffuser distinguished itself from the other models on the Kitchen-Medium and Kitchen-Hard tasks, achieving legibility scores that are 160% and 300% higher than the next best baselines respectively. Generally, the legibility performance on the kitchen tasks is limited by the current capabilities of generative models. Performing these sub-goals in an out of distribution order is challenging for imitation learning algorithms.

6.2 Does optimizing for legibility effect success rate?

Our results show an interesting connection between success rate and legibility. When generating actions, diffusion model guidance is known to help with constraint satisfaction Ajay et al. (2022). This is why we see Legibility Diffuser perform best on Block Reach and Kitchen-Easy. In order to be successful on block reach, the agent must pick up the target block. For Kitchen-Easy, interacting with the opposing object wastes valuable time. By being legible, we ensure that we are satisfying these constraints

6.3 How important are the individual components of Legibility Diffuser?

We see that decaying the guidance weight is important for legibility diffuser to maintain a high success rate. Without this decay, the model seems to have a harder time staying within distribution. Including guidance away from the opposing goal g^- has the effect of improving legibility.

7 Discussion and Future Works

We find that Legibility Diffuser is able to optimize for legible motion generation by taking advantage of diffusion model guidance. By guiding our action generation to be distinct from both an unconditional model and a model conditioned on an opposing goal we are able to produce intent expressive motion.

The true value of legible motion lies in human robot collaboration, so an extensive study that investigates how Legibility Diffuser performs in these scenarios is needed. Specifically, we would like to see that our method leads to faster task completion times and improved user satisfaction. There is also no need for motion to be the only mode of legible generation, generating goal specific text or images could also be useful for collaborative tasks.

A strength of Legibility Diffuser is that it does not require any goal, reward, of constraint labels. This lack of supervision gives us the potential to learn from large scale unstructured datasets on the internet. However, we do require our data to have a defined start and end, which is less common than completely unstructured ”play” data. Ideally algorithms should be able to learn from such data, and future work will look into how Legibility Diffuser performs in this setting.

Our algorithm assumes access to low dimensional states for both observations and goal conditioning. In order for our method to be deployed on a real robot, the algorithm needs to be adapted to work with image inputs. While many papers have shown that robot manipulation tasks can be learned from image observations, this is less explored in the context of diffusion guidance. We believe our method is a step towards showing that diffusion guidance is effective in the continuous domain and not just for discrete classes and values. We hope our exploration encourages the machine learning community to consider legible motion when designing robot policies.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ international conference on intelligent robots and systems*, pp. 708–713. IEEE, 2005.

Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

Zichen Jeff Cui, Yibin Wang, Nur Muhammad, Lerrel Pinto, et al. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Anca Dragan and Siddhartha Srinivasa. Familiarization to robot motion. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pp. 366–373, 2014.

Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308. IEEE, 2013.

Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. Effects of robot motion on human-robot collaboration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pp. 51–58, 2015.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Guy Hoffman and Gil Weinberg. Shimon: an interactive improvisational robotic marimba player. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pp. 3097–3102. 2010.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. *Reinforcement learning: State-of-the-art*, pp. 45–73, 2012.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Christina Lichtenthäler, Tamara Lorenz, and Alexandra Kirsch. Towards a legibility metric: How to measure the perceived value of a robot. In *International Conference on Social Robotics, ICSR 2011*, 2011.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pp. 1113–1132. PMLR, 2020.

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3:297–330, 2020.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Leila Takayama, Doug Dooley, and Wendy Ju. Expressing thought: improving robot readability with animation principles. In *Proceedings of the 6th international conference on Human-robot interaction*, pp. 69–76, 2011.

Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences*, 28(5): 675–691, 2005.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5628–5635. IEEE, 2018.

Xuan Zhao, Tingxiang Fan, Dawei Wang, Zhe Hu, Tao Han, and Jia Pan. An actor-critic approach for legible robot motion planner. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5949–5955. IEEE, 2020.

Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3560–3566. IEEE, 2023.

Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10): 1164–1193, 2013.

A Appendix

You may include other additional sections here.