Rethinking Visual Rearrangement from A Diffusion Perspective

Tianliang Qi[1,2] , Xinhang Song[1,2*], Yuyi Liu[1,2] , Shuqiang Jiang[2,1]
[1]State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
{tianliang.qi, xinhang.song, yuyi.liu}@vipl.ict.ac.cn
sqjiang@ict.ac.cn

Abstract

Rearranging disarrayed objects to their intended goal states requires the agent to comprehend the changes that have occurred in the scene and to reason about the process of these changes. To address this, we propose a novel perspective on the visual rearrangement task, drawing inspiration from the diffusion processes in molecular thermodynamics. We model the room shuffle and unshuffle stages as the forward and reverse processes of diffusion. In contrast to conventional methods that rely on scene modeling and differential comparisons, our approach provides insight into the intrinsic evolution process between the goal and initial states of the scene, which allows for a more reasonable rearrangement of objects through fine-grained and progressive denoising steps with high confidence. By analyzing the task objectives, we represent the scene via spatial distributions of objects and model the visual rearrangement process using a diffusion bridge model. Building upon this, we introduce the Diffusion Rearrangement model, which takes point cloud data as input, fits it into Gaussian mixture distributions to represent the states of objects, and predicts the rearrangement target through an iterative denoising transformer. Experimental results on the RoomR dataset demonstrate the effectiveness of our approach.

1. Introduction

Room rearrangement is a common problem people face in daily life. As the environment becomes increasingly chaotic and disordered, people need to reorganize objects to their original location. A similar task is the visual rearrangement task[42], which is one of the most challenging tasks in the field of embodied AI. The goal of visual rearrangement task is for the agent to explore and memorize the initial room layout, then restore the room to its initial state after some objects have been randomly disarranged. The similarities between the daily room reorganization and visual rearrangement task can be understood in the context of thermodynamic analysis. For example, considering the Shannon entropy of object distribution as a measurement of the scene, the environment becoming disordered and human organizing behavior, as well as the room getting disarranged and agent restoring actions, are essentially process of entropy increasing and decreasing. Furthermore, if the process of randomly disarranging the room and restoring it are viewed as the forward and reverse process of a Markov random process[1], the state of the room can be described by a stochastic differential equation, for example, the Langevin dynamics equation[36]. Then it can be proven that the distribution probability of the objects in the room and the change in information entropy during this process conform to the diffusion equation, which means the change in room state can be modeled as a diffusion process.

Existing methods on visual rearrangement task can be classified into end-to-end reinforcement learning approaches[11, 42] and modular approaches[29, 31, 37]. The reinforcement learning methods attempt to leverage extensive experience during training and use a parameterized mapping mechanism to memorize environmental states. Modular methods, in the meanwhile, divide the task into perception and planning modules, explicitly modeling and comparing environmental states to make inference of the rearrangement goal. The commonality between both methods lies in building certain form of data representation to model and compare the initial and current room states, and then directly inferring the rearrangement targets from the difference. As a result, these methods rely on precise perception capabilities to a great extent and are sensitive to input noise. These methods take the initial and current room states as two isolated data distributions and do not further explore the inherent relationships and evolution processes between the two states.

Inspired by the diffusion process in nonequilibrium thermodynamics, we propose modeling the room state changes using a diffusion process to infer the process of object state changes. We redefine the rearrangement task from a diffu-

*Corresponding author

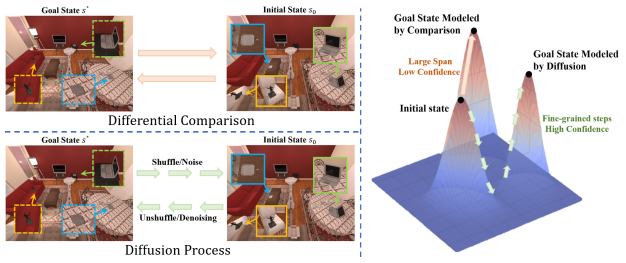

Figure 1. Different from methods based on differential comparison, our method build up the evolutionary process between the initial and goal states and predict the rearrangement targets step-by-step.

sion perspective, treating the process of randomly shuffle the room as the forward diffusion process, and the process of room restoration as the reverse process, as illustrated in Fig. 1. Unlike generative diffusion models that synthesize data from Gaussian noise, our task-specific diffusion bridge operates between two structured states, allowing interpretable evolution under task constraints. Our motivation is to simulate gradient descent, with fine-grained and step-by-step changes that gradually combine to restore the overall changes in the room state. Conventional comparison-based methods resemble finding the shortest path between two data distributions, due to the large inference span, the accuracy and optimality of the results are hard to be guaranteed. Scene change inference based on the diffusion modeling has higher confidence at each step of the diffusion process, and as the complexity of task increases, this method offers a higher chance of approaching the accurate goal state.

In this paper, we propose Diffusion Rearrangement, a novel modular method based on Brownian Bridge Diffusion Model[26] to tackle the visual rearrangement task. Different from previous methods that infer the scene changes through understanding and comparing the observations of the goal and initial states separately, we focus on modeling the changing process between the goal and initial states to predict interpretable rearrangement targets through a diffusion process. We take the differential point cloud[29] as input and employ a distribution-based method to represent the states of objects, which better aligns with the task definition and constrains the inference space to the acceptable range. The Diffusion Rearrangement framework uses a transformer-based architecture[38] as the denoising model, learning the reverse iterative process from observations of the goal and initial states to output the changes in the states of objects. The main contributions of this paper are as follows:

- A novel approach to understanding the visual rearrangement task through diffusion process, with proof at the level of probability and statistics.
- A scene representing approach that fits the spatial states of objects into Gaussian mixture models to align with the task objective.
- A diffusion-based method that learns the changing process between the goal and initial states and tackles the visual rearrangement task by iterative denoising steps.
- The proposed method outperforms existing methods in all metrics and the experimental results of ablation demonstrate the effectiveness of our method.

2. Related Work

2.1. Embodied AI

Embodied AI refers to artificial intelligence systems that possess a physical body and interact with the physical world in a human-like way that takes actions by perceiving and reasoning about its environment. In recent years, embodied AI has gained significant attention in multiple research domains and various embodied tasks have driven much progress, including object-goal navigation[27, 39, 40, 45–48], embodied question answering[5, 6, 43], object manipulation[10, 44], and object rearrangement[11, 29, 31, 37]. The development of methodologies and techniques for embodied AI has largely benefited from the availability of high-fidelity 3D simulation platforms, such as Habitat[32], RLBench[18], ThreeDWorld[12] and our work builds on AI2-THOR[23].

2.2. Diffusion Model

Diffusion models[8, 16, 17, 35, 36], which is first inspired by the diffusion phenomenon of non-equilibrium statistical physics[34], have recently emerged as a powerful class of generative models, achieving remarkable results in generating high-quality images and accurately modeling complex distributions. The forward diffusion process defined by Denoising Diffusion Probabilistic Model (DDPM)[17] starts from data $\boldsymbol{x}_0$ and ends at a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, which is suitable for generating data from certain domain. Aiming at learning a mapping between different domains, Brownian Bridge Diffusion Model (BBDM)[26] takes conditional input $\boldsymbol{x}_T$ as end of the diffusion process instead of pure Gaussian noise and defines the forward process as a stochastic Brownian Bridge process. Drawing inspiration from the forward and reverse processes of diffusion, we employ a diffusion bridge to model the rearrangement process, where each end of the bridge represents the initial and goal states of the scene.

2.3. Rearrangement

The conventional rearrangement problem in the field of task and motion planning[13–15, 19, 20], usually referred to as rearrangement planning[2–4, 9, 21, 22, 24], addresses the ability of interacting with and manipulating objects to achieve desired environmental configuration, with the assumption of full access to all object states. Recent rearrangement task[1, 42] in the filed of embodied AI, which is the focus of our work, requires not only the ability of scene un-

derstanding and object manipulation, but also the capability of perceiving the environment solely from raw visual input such as egocentric images. The problem scope of rearrangement is also not confined to a single tabletop with several objects but extends across multiple rooms with various forms of objects, introducing additional challenges in terms of understanding and reasoning about the environment. While previous works have addressed visual rearrangement tasks using diffusion-based approaches[28, 33, 41], our method differs in two key aspects: we do not assume known object states, and we focus on planning to rearrange the entire room rather than manipulating object poses within limited regions.

3. Preliminaries

3.1. Task Definition

The general form of the rearrangement task in the field of embodied AI is defined by Batra et al.[1], who specify rearrangement task using the mathematical model of Partially Observable Markov Decision Processes (POMDP). The rearrangement task requires an agent to transform an indoor environment from a starting state s_0 to a desired goal state $s^* \in S^* \subseteq \mathcal{S}$ with a sequence of actions $a \in A$. With only rigid bodies considered, the space of rigid-body poses can be denoted as $\mathcal{S}_i = SE(3) = \mathbb{R}^3 \times SO(3)$, where $\mathbb{R}^3$ and $SO(3)$ respectively represent 3D locations and rotations space. The world state space $\mathcal{S}$ is factorized as the Cartesian product of the rigid-body pose spaces corresponding to each of the n parts, which can be written as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \ldots \mathcal{S}_n$.

Since the agent does not have direct access to the goal state space and have to operate solely from observations o, the goal specification g is pre-defined to provide guidance for the actions of the agent toward goal states. The goal specification encompasses a variety of forms[1], such as GeometricGoal, ImageGoal, LanguageGoal, ExperienceGoal and PredicateGoal. We consider an instance of ExperienceGoal-based rearrangement task proposed by Weihs et al.[42], which defines rearrangement task as a two-stage process, including **walkthrough** and **unshuffle**. During the walkthrough stage, the agent is placed into a room with goal state configuration s^* and allowed to explore the environment according to its need. Then the agent is removed from the room and some changes of the objects are made to initialize a shuffled room s_0. During the unshuffle stage, the agent is placed back into the room and starts reorganizing the objects to convert s_0 to s^*. At each time step t, the agent receives egocentric RGB-D observations and executes a discrete action, where the action space consists of `move_ahead`, `turn_right`, `turn_left`, `look_down`, `look_up`, `put_down`, `open`, `stop`. The agent autonomously executes the action `stop` when it determines to complete the task.

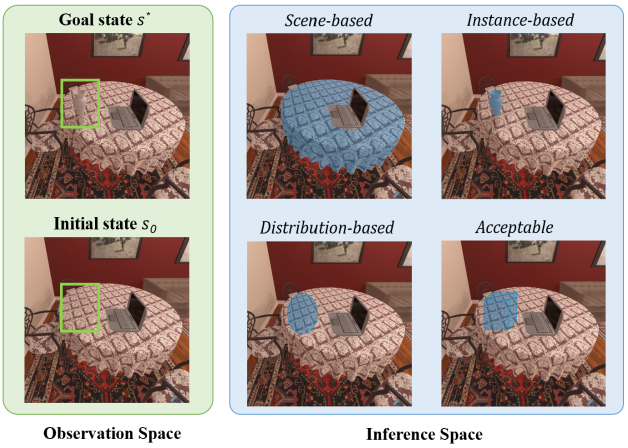

Figure 2. Observation and inference space of different methods. The blue area represents the possible positions to which the vase can be restored. Representing objects as distribution can better reflect the target space of rearrangement.

3.2. Analysis of Task Objective

For each task, while starting state s_0 is uniquely determined by the exact position and state of all objects, goal state s^* is an element from a set of acceptable states S^*. The acceptable states can be formulated as $S^* = S_1^* \times S_2^* \ldots S_n^*$, where S_i^* indicates the acceptable distributions of object i, optionally defined by the threshold of intersection over union (IOU) between 3D bounding boxes[42]. As a result, the principal objective of visual rearrangement task is not to restore the exact positions of target objects, but rather to strike a balance between location precision and task completeness.

We take one scene from dataset RoomR[42] as an example. As illustrated in Fig. 2, the agent gets observations of the same table in both goal and initial states, indicating that a vase needs to be repositioned on the table. Methods based on scene understanding[11, 31] typically establish the positional relationships between objects using a scene graph, which provides only coarse-grained information about the rearrangement target, in this case, expanding the space of inference and action to encompass the entire table. Methods based on instance identification[29, 37] generally compare the difference of observation from the same perspective, which can be minor at room scale, leading to a narrowed target. We propose representing objects as distribution over locations, which will be discussed in Sec. 4.1. This distribution-based approach restricts the inference space to a range close to the scope of acceptable states, avoiding unnecessary precision.

4. Method

In this section, we propose a modular approach motivated by diffusion process to tackle the visual rearrangement task. We begin the section by discussing the distribution form

of objects and representation of scene change (Sec. 4.1). Then we introduce the motivation and process of diffusion bridge model (Sec. 4.2). Finally, we present the Diffusion Rearrangement model (Sec. 4.3) which reasons about the process of scene change and generates the rearrangement target without explicit matching. The framework of our method is illustrated in Fig. 3. The pseudocode of training and inference are presented in Algorithm 1 and Algorithm 2.

4.1. Scene Representation via Distribution

As mentioned in Sec. 3, the goal state is confirmed by acceptable arrangements within the IOU threshold. To better reflect the target range of objects, we represent the room state using the distribution of the center point coordinates of the objects. The room state at time step t can be formulated as $\boldsymbol{x}_t = [x_1^t, \ldots, x_n^t]$, where x_i^t denotes the center point coordinate of object i at time step t. In particular, we use Gaussian mixture model to represent the distribution of $\boldsymbol{x}_t$, which can be formulated as:

$$p(\boldsymbol{x}_t) = \sum_{i=1}^{n} \pi_i \mathcal{N}\left(\boldsymbol{x}_t \mid \mu_i, \Sigma_i\right) \tag{1}$$

We choose Gaussian mixture model for three reasons: 1) the product of a Gaussian mixture distribution and a single Gaussian distribution remains a Gaussian mixture; 2) each component of Gaussian mixture corresponds to an object, with mean and covariance representing the center point and positional range of the object; 3) convenience for parameterization and robustness to noise.

In rearrangement tasks, we only get access to goal and initial states, which can be explicitly fitted by the Gaussian mixture model. Considering the forward transition probability $q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_T\right)$ defined in Sec. 4.2, we can derive the distribution of objects at time $t = 1$ given goal state $\boldsymbol{x}_0$ and initial state $\boldsymbol{x}_T$:

$$\begin{aligned} p\left(\boldsymbol{x}_1\right) &= p(\boldsymbol{x}_0) q\left(\boldsymbol{x}_1 \mid \boldsymbol{x}_0, \boldsymbol{x}_T\right) \\ &= \sum_{i=1}^{n} \pi_i \left[\mathcal{N}\left(\boldsymbol{x}_0 \mid \mu_i, \Sigma_i\right) \cdot \mathcal{N}\left(\boldsymbol{x}_0 \mid \mu, \Sigma\right)\right] \\ &= \sum_{i=1}^{n} \left(\pi_i Z_i\right) \mathcal{N}\left(\boldsymbol{x}_0 \mid \mu_i', \Sigma_i'\right) \end{aligned} \tag{2}$$

By properties of Markov chains, we can derive that distribution of objects at anytime within $[0, T]$ conforms to the form of Gaussian mixture model, which indicates that the process of rearrangement can be tackled in the latent space of Gaussian mixture model.

Since only a few objects in the room are repositioned and most objects keep the same position, the number n of components of Gaussian mixture can be reduced to the amount of objects to be rearranged, as objects that are not moved can be considered as components of $\pi_i \mathcal{N}\left(\boldsymbol{x}_i, \Sigma_i\right)$ with $\pi_i = 0$. Consequently, we consider a simplified approach that only focuses on differential changes among all objects to represent the room state. Following Liu et al.[29], we utilize point cloud as data representation to capture various scene changes precisely, as the point cloud contains rich geometric, positional, and scale information of objects, which can be intuitively represented by means and covariances of the Gaussian mixture model. In particular, during the walkthrough stage, the agent explores the environment and generates egocentric point cloud by depth observation. Then the agent records its current pose and update the global point cloud. During the unshuffle stage, the agent also explores the environment following the same trajectory and updates the global point cloud. The differential point cloud of walkthrough and unshuffle stage are obtained by extracting the moved and protruding parts of the two sets of global point cloud.

While the amount of objects to be rearranged is unknown, we fist apply clustering methods to determine the number of components in the Gaussian mixture model. Then the differential point cloud from both stages are fitted to Gaussian mixture models to represent the room state, denoted as $\boldsymbol{x}_0 \sim GMM(p_w)$ and $\boldsymbol{x}_T \sim GMM(p_u)$ with mean μ_i and covariance Σ_i representing the center coordinate and positional range of i th object.

4.2. Diffusion Process in Rearrangement

Drawn inspiration from the diffusion process in molecular thermodynamic motion, we propose redefining the rearrangement task as a reversible process of object movement. Given a room with a goal state configuration, the random shuffle process corresponds to the diffusion process, where molecules move towards a stable state driven by concentration differences. The unshuffle process, in turn, corresponds to the reverse diffusion process. What distinguishes rearrangement task from molecular motion is that the reverse motion process of molecular is impossible without external forces, while an agent with interaction capabilities can restore the room.

Assuming that the random shuffle process is executed by the agent with a sequence of actions $A = \{a_1, \ldots, a_m\}$, we can simply derive the reverse process by inverting the sequence and actions to $A' = \{a_m', \ldots, a_1'\}$, which offers a possible solution to the shuffle stage. This also suggests that any intermediate state between the goal state and initial state, which is segmented by a single action of the agent, is a valid room configuration. In contrast, the diffusion process in domains of image generation links the target image and Gaussian noise by adding noise and denoising, where intermediate states are meaningless noisy images. In rearrangement tasks, the forward process predicts the noise at time step t for training, which practically indicates the directional trend of object movement during shuffle. The reverse process, in turn, refines the inferred room state through gradual

denoising, indicating the process of objects moving towards the goal state.

Compared to conventional diffusion models[8, 17, 35] in the filed of image generation, where one end of both forward and reverse processes is standard Gaussian noise, the diffusion process in the visual rearrangement task focuses on the transition between the initial and goal states, taking two meaningful data representations as input instead of pure noise.

Given $\boldsymbol{x}_0$ and $\boldsymbol{x}_T$ as representations of goal and initial states, we consider a continuous-time stochastic model in the form of a Brownian Bridge[26], which can be formulated as:

$$p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{x}_T\right)=\mathcal{N}\left(\left(1-\frac{t}{T}\right)\boldsymbol{x}_0+\frac{t}{T}\boldsymbol{x}_T, \frac{t(T-t)}{T}\boldsymbol{I}\right) \tag{3}$$

It can be found that the process is conditioned on the starting point $\boldsymbol{x}_0$ at $t=0$ and ending point $\boldsymbol{x}_T$ at $t=T$, which forms a diffusion bridge process.

The forward diffusion process of Brownian Bridge[26] can be defined as:

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{x}_T\right)=\mathcal{N}\left(\boldsymbol{x}_t;\left(1-m_t\right)\boldsymbol{x}_0+m_t\boldsymbol{x}_T, \delta_t\boldsymbol{I}\right) \tag{4}$$

where $m_t=\frac{t}{T}$ and $\delta_t=2s(m_t-m_t^2)$ with factor s controlling the maximum variance and generally setting to 1.

Given goal state $\boldsymbol{x}_0$ and initial state $\boldsymbol{x}_T$, the transition probability $q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_T\right)$ can be derived by substituting the term of $\boldsymbol{x}_0$ with intermediate state $\boldsymbol{x}_{t-1}$:

$$\begin{aligned} q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_T\right)=\mathcal{N}\Bigg(\boldsymbol{x}_t; & \frac{1-m_t}{1-m_{t-1}}\boldsymbol{x}_{t-1} \\ & +\left(m_t-\frac{1-m_t}{1-m_{t-1}}m_{t-1}\right)\boldsymbol{x}_T, \delta_{t|t-1}\boldsymbol{I}\Bigg) \end{aligned} \tag{5}$$

where $\delta_{t|t-1}$ is derived as:

$$\delta_{t|t-1}=\delta_t-\delta_{t-1}\frac{\left(1-m_t\right)^2}{\left(1-m_{t-1}\right)^2} \tag{6}$$

The reverse process of Brownian Bridge[26] aims to predict $\boldsymbol{x}_{t-1}$ given $\boldsymbol{x}_t$ and initial state $\boldsymbol{x}_T$ by estimating the noise part of $\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, \boldsymbol{x}_T, t\right)$, which can be formulated as:

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_T\right)=\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, \boldsymbol{x}_T, t\right), \tilde{\delta}_t\boldsymbol{I}\right) \tag{7}$$

where $\tilde{\delta}_t$ is derived as:

$$\tilde{\delta}_t=\frac{\delta_{t|t-1}\cdot\delta_{t-1}}{\delta_t} \tag{8}$$

The training objective of optimizing the Evidence Lower Bound (ELBO) can be derived in a simplified form[26]:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_T, \boldsymbol{\epsilon}}\left[\left\|\tilde{\boldsymbol{\mu}}_t\left(\boldsymbol{x}_t, \boldsymbol{x}_T\right)-\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, \boldsymbol{x}_T, t\right)\right\|^2\right] \sim \\ &\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_T, \boldsymbol{\epsilon}}\left[\left\|m_t\left(\boldsymbol{x}_T-\boldsymbol{x}_0\right)+\sqrt{\delta_t}\epsilon-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)\right\|^2\right] \end{aligned} \tag{9}$$

$\tilde{\boldsymbol{\mu}}_t\left(\boldsymbol{x}_t, \boldsymbol{x}_T\right)$ denotes the mean value derived through Bayes'theorem and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ denotes the trained neural network to predict the noise.

4.3. Diffusion Rearrangement

Fig. 3 illustrates our approach to solving the visual rearrangement task. Our method takes the goal and initial configurations of the room as input and outputs predictions of the movement changes for each object to be rearranged. We model the visual rearrangement task as a diffusion bridge process[26], which starts from the goal configuration and ends at the initial configuration. Towards this goal, we design a denoising network, which adopts the Transformer[38] architecture, to predict the rearrangement targets by iterative denoising on distribution of objects. During inference, at each time step t, we predict the noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)$ which can be viewed as the gradient toward the goal state. By gradually combining the denoising steps, we restore the actual changes between initial and goal states.

Denoising Network Architecture: We adopt the conventional encoder-only transformer architecture as the backbone, which is composed of several standard transformer blocks[38], each consisting of the multi-head self-attention module and the position-wise feed-forward module. By enacting interactions between different object distributions through the self-attention mechanism, the model establishes the intrinsic connections between the distributions and predicts the noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)$ of the diffusion process, which is equivalent to predicting the tendency of object movement. For input of the Gaussian mixture models, we adopt a multilayer perceptron (MLP) as the network token encoder to map the parameters to high-dimensional latent space. The positional embedding of the transformer is implemented by the sinusoidal time embedding given time step t. Finally, the network decoder is also built as an MLP to output the predicted changes of objects, represented as changes in the parameters of the Gaussian mixture model.

Training and Inference: While sampling $\boldsymbol{x}_t$ from input $\boldsymbol{x}_0, \boldsymbol{x}_T, t$ can be computed in discrete form as $\boldsymbol{x}_t=\left(1-m_t\right)\boldsymbol{x}_0+m_t\boldsymbol{x}_T+\sqrt{\delta_t}\epsilon$, the training objective of optimizing the ELBO defined in Sec. 4.2 is derived as estimating $\boldsymbol{x}_t-\boldsymbol{x}_0$. This objective can be considered as minimizing the distance from current state $\boldsymbol{x}_t$ to goal state $\boldsymbol{x}_0$, denoted as:

$$\mathcal{L}_{rearrange}=\left\|\boldsymbol{x}_t-\boldsymbol{x}_0-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)\right\|^2 \tag{10}$$

The pseudocode of training process is presented in Algorithm 1.

During inference, the denoising model predict the goal state by iteratively applying the reverse transition process, which can also be computed in discrete form as $\boldsymbol{x}_{t-1}=\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, \boldsymbol{x}_T, t\right)+\sqrt{\tilde{\delta}_t}\epsilon$. Distinct from conventional BBDM[26] which serves as a generation model and has no access to the generation target $\boldsymbol{x}_0$, the rearrangement

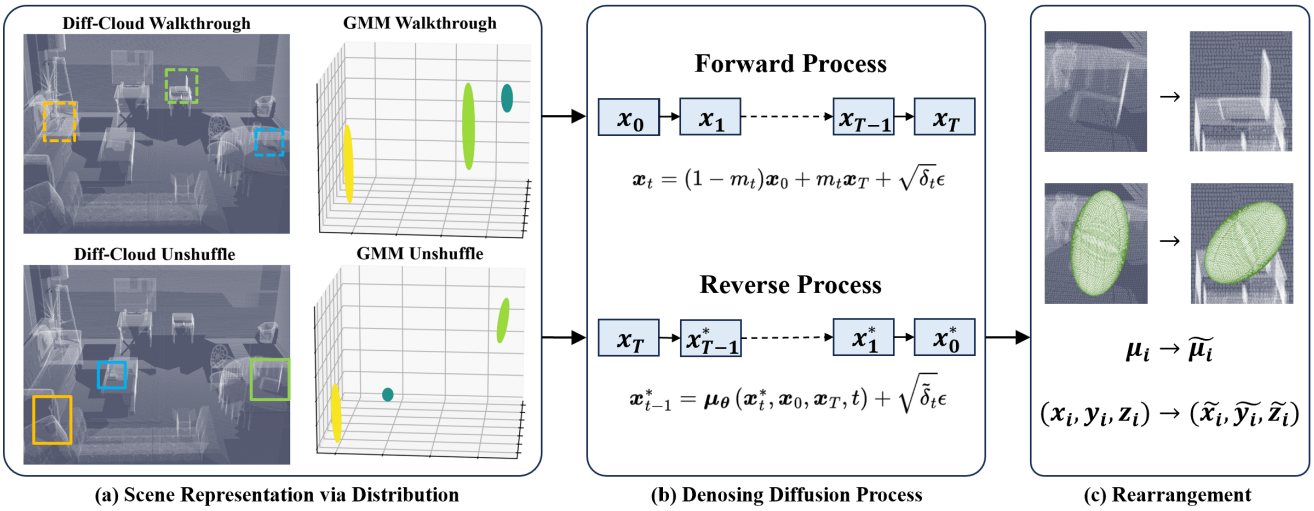

Figure 3. Framework of our Diffusion Rearrangement model. (a) We represent the scene changes via distributions of Gaussian mixture model fitted from differential point cloud during the walkthrough and unshuffle stage. (b) A diffusion denoising model is implemented, with forward process for training and reverse process for inference, to predict the target object states through iterative denoising steps. (c) The rearrangement targets are derived from changes in distributions with $\mu_i \rightarrow \tilde{\mu}_i$ indicating the movement of i th object.

task takes both $\boldsymbol{x}_T$ and $\boldsymbol{x}_0$ as input to predict the actual goal state $\boldsymbol{x}_0^*$. Thus the sampling process is reformulated as $\boldsymbol{x}_{t-1}^* = \boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t^*, \boldsymbol{x}_0, \boldsymbol{x}_T, t\right) + \sqrt{\tilde{\delta}_t}\epsilon$ to get observation of goal state $\boldsymbol{x}_0$ involved. The pseudocode of inference process is presented in Algorithm 2.

Algorithm 1 Training

repeat
  goal state s^*, initial state s_0
  differential point cloud p_w, p_u
  paired data $\boldsymbol{x}_0 \sim GMM(p_w), \boldsymbol{x}_T \sim GMM(p_u)$
  time step $t \sim \text{Uniform}(1, \ldots, T)$
  Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Forward diffusion $\boldsymbol{x}_t = (1 - m_t)\,\boldsymbol{x}_0 + m_t \boldsymbol{x}_T + \sqrt{\delta_t}\epsilon$
  Take gradient descent step on
  $\nabla_{\boldsymbol{\theta}}\left(\lambda \left\| m_t\left(\boldsymbol{x}_T - \boldsymbol{x}_0\right) + \sqrt{\delta_t}\epsilon - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)\right\|^2\right)$
until converged

5. Experiment

5.1. Experimental Setup

We evaluate our method using the AI2-THOR simulator[23], which provides near photo-realistic observation in 3D indoor scenes. We conduct experiments on the RoomR dataset[42], which consists of 80 rooms with 4000 tasks for training, and 20 rooms with 1000 tasks respectively for validation and test. In each task of RoomR dataset, the states of 1 to 5 objects are transformed.

Algorithm 2 Inference

goal state s^*, initial state s_0
differential point cloud p_w, p_u
input $\boldsymbol{x}_0 \sim GMM(p_w), \boldsymbol{x}_T \sim GMM(p_u), \boldsymbol{x}_T^* = \boldsymbol{x}_T$
for $t = T, \ldots, 1$ **do**
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\epsilon = \mathbf{0}$
  $c_{xt} = \frac{\delta_{t-1}}{\delta_t}\frac{1-m_t}{1-m_{t-1}}$
  $c_{yt} = \frac{\delta_{t|t-1}}{\delta_t}\left(1 - m_{t-1}\right)$
  $c_{\epsilon t} = \left(1 - m_{t-1}\right)\frac{\delta_{t|t-1}}{\delta_t}$
  $\boldsymbol{x}_{t-1}^* = c_{xt}\boldsymbol{x}_t^* + (1 - m_t)c_{yt}\boldsymbol{x}_0 + m_t c_{yt}\boldsymbol{x}_T + c_{\epsilon t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t^*, t\right) + (\sqrt{\tilde{\delta}_t} + c_{yt}\sqrt{\delta_t})\epsilon$
return $\boldsymbol{x}_0^*$

To verify the generalization of our method, we also build a dataset on ProcTHOR[7] simulator. We select 8000 scenes from ProcTHOR simulator and split 6000 scenes for training, 1000 scenes for validation and 1000 scenes for test. For each scene, we randomly generate one rearrangement task utilizing an official generation script. Our dataset expands the action space of the agent from single room to multiple rooms and introduces new patterns of state change of objects, which better characterizes the spatial complexity that fits the indoor environment in reality.

5.2. Implementation Details

The egocentric observation of the agent is set as 480*480 pixel RGB and depth images. The token encoder adopts a 2-layer MLP and produces 512-dimensional features as

input for the transformer. The transformer has 6 encoder layers with 8 heads of attention, where each encoder layer processes 512-dimensional features. Prior to diffusion model training, the agent executes an exploration policy in both the walkthrough and unshuffle stages for each task in the training set. The collected point cloud and pose data are subsequently used for training. To enhance robustness, slight Gaussian perturbations are added to the input point clouds during training. We adopt the Adam optimizer with the hyperparameters (lr, β_1, β_2) set to $(0.001, 0.9, 0.999)$. The batch size of training is selected as 64 and the denoising model is trained on an NVIDIA A40 GPU for 25,000 iterations. We also adopt the accelerated sampling processes proposed by DDIM[35], and the sampling steps is set to $S = 200$ to balance the sampling quality and efficiency.

Table 1. Test set performance on RoomR[42] dataset.

Method	Suc (%)↑	FS (%)↑	Mis↓	E↓
MaSS[37]	4.7	16.5	1.018	1.016
TIDEE[31]	11.7	28.9	0.734	0.715
CAVR[29]	14.2	33.1	0.707	0.714
Ours	**17.8**	**38.8**	**0.641**	**0.643**

5.3. Evaluation Metrics:

Following Weihs et al.[42], we employ four metrics to evaluate the performance of the agent from different perspectives. **%Success** metric is the strictest and most unforgiving metric, measuring the proportion of tasks in which the agent has restored all objects to their goal states. **%Fixed Strict** metric is an object-level metric, which measures the proportion of objects successfully rearranged per task and equals to 0 if there is any misplaced object that should not be rearranged. **Misplaced** metric is defined as the number of misplaced objects at the end of the episode divided by the number of misplaced objects at the start. Note that this metric can exceed 1 if, during the unshuffle stage, the agent misplaces more objects than were originally misplaced at the start. These metrics are quite strict and do not grant any partial credit, even if the agent restores objects to a state that is very close to the goal state. **Energy Remaining** metric is defined as the amount of energy remaining after the unshuffle stage, divided by the total energy at the start of the unshuffle stage. The energy represents the difference between two states of an object and is defined by an energy function $D : S \times S \Rightarrow [0, 1]$.

Table 2. Test set performance on ProcTHOR[7].

Method	Suc (%)↑	FS (%)↑	Mis↓	E↓
MaSS[37]	0.5	9.7	0.985	0.987
TIDEE[31]	0.7	10.8	0.924	0.917
CAVR[29]	4.9	17.0	0.849	0.851
Ours	**8.4**	**24.7**	**0.806**	**0.814**

5.4. Comparisons with Related Works

We compared our method with three modular methods on the latest AI2-THOR Rearrangement Challenge. The experimental results are reported in Tab. 1. We briefly introduce the three baselines as follows:

TIDEE[31]: This method maintains the 2D occupancy map for exploration and navigation, and keeps track of objects and their labels over time. After the exploration of two stages, it infers the spatial relationship changes for all objects to identify the moved ones that need to be rearranged.

MaSS[37]: This model uses a search-based policy to rapidly find objects and builds voxel-based semantic map of the environment, which is leveraged to identify the movement of objects. After the inference of all rearrangement goals, this model transports them to their goal state.

CAVR[29]: This model is designed for object movement, which leverages the observation distance map to explore the environment efficiently and performs scene change detection and scene change matching using point cloud, avoiding the reliance of category inference.

As shown in Tab. 1 and Tab. 2, our method outperforms related works by a significant margin across all metrics. In particular, our method improves the success rate of rearranging tasks by 3.6% and the proportion of objects successfully restored by 5.7%, compared to the state-of-the-art model[29]. Compared to scene-based method TIDEE [31] and instance-based method MaSS [37], our method, representing object states via Gaussian mixture model, outperforms them significantly in the Fixed Strict and Energy Remaining metrics, demonstrating the effectiveness of inferring in the distribution space. In comparison to the point cloud-based method CAVR [29], which uses a matching algorithm to infer object changes, our diffusion-based method shows improvements across all metrics, highlighting its superior performance in reasoning about the connection between the goal and initial states of the scene.

5.5. Ablation Study

We conduct several ablative experiments to verify the effectiveness of designs in our framework.

Ablation on matching algorithms: Since we use a transformer-based framework to process sequential data, the object states of input and output are directly corresponded without the need for matching. We replace the denosing transformer model with a direct matching method based on weights of the Gaussian mixture model and a bipartite graph matching algorithm, specifically the Kuhn-Munkres algorithm[25], which predicts the goal state by computing the cosine similarity of embeddings derived from the token encoder. The experimental results are presented in Tab. 3. Both the random matching the Hungarian algorithm[25] underperform, as the cosine similarity of embeddings fails to adequately capture the relationships between objects, and

Table 3. Ablation studies on matching algorithms and representations of input.

Matching	Representation	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓
Direct	✓	6.5	19.4	0.837	0.860
Kuhn-Munkres	✓	12.8	29.7	0.730	0.732
✓	coordinate	16.7	37.1	0.664	0.669
✓	feature	17.2	37.9	0.652	0.658
✓	✓	**17.8**	**38.8**	**0.641**	**0.643**

"✓" indicates our proposed corresponding modules, representing the denoising model in Matching column and distributions of Gaussian mixture model in Representation column.

the matching algorithm cannot reason about the process of object changes.

Ablation on representations of input: To evaluate the effectiveness of representing the scene via distributions of objects, we replace the input of Gaussian mixture models with center point coordinates calculated from the point cloud derived from the global point cloud using a clustering method. We also replace the input with geometric features extracted from PointNet++[30] for point cloud of each object instance. The token encoder and decoder of the framework are adjusted to corresponding sizes. The experimental results are presented in Tab. 3. Both coordinate and feature representations underperform, as coordinates only convey the positional information of objects, while features solely provide appearance information.

Table 4. Impact of sampling steps on validation set.

steps	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓
20	16.4	36.7	0.676	0.687
50	16.9	37.5	0.665	0.674
100	17.2	37.8	0.655	0.660
200	**17.9**	38.1	**0.648**	**0.656**
500	17.6	38.0	0.662	0.668
1000	17.4	**38.3**	0.655	0.660

Impact of sampling steps: We conduct experiments on the validation set of RoomR[42] dataset to evaluate the impact of sampling steps S, as shown in Tab. 4. We find that when the number of sampling steps is fewer than 200, all metrics improve rapidly with the increase of sampling steps, while the increase beyond 200 leads to a decrease in performance across three metrics and a slight improvement in Fixed Strict metric.

5.6. Robustness to Noise

While modular design helps decompose the complex rearrangement task into more manageable sub-tasks, it may face performance bottlenecks if certain modules function inefficiently or fail. Specifically, our method relies on distributions of Gaussian mixture models fitted from point cloud data. We evaluate our Diffusion Rearrangement model by introducing Gaussian noise ($\sigma = 0.01$ meters) to the depth input, assessing its robustness under sensor noise and measurement inaccuracies. Although the distribution-based representation offers a degree of robustness to noise, as shown in Tab. 5, the quality of the differential point cloud can still impact overall performance. To address this, our future work will focus on improving the robustness and generalization of the method.

Table 5. Test set performance on RoomR[42] dataset.

Method	Suc (%) ↑	FS (%) ↑	Mis ↓	E ↓
Ours	17.8	38.8	0.641	0.643
Ours + *noisy depth*	17.5	38.3	0.654	0.660

6. Conclusion

Motivated by principles of nonequilibrium thermodynamics, we rethink the visual rearrangement task from a diffusion perspective and model the shuffle and unshuffle processes as the forward and reverse diffuion processes. For building up the evolutionary process between the initial and goal states, we propose a denosing model based on diffusion bridge process, which takes distributions of Gaussian mixture model fitted from point cloud data as input and predicts the rearrangement targets by iterative denoising steps. We conduct experiments on the RoomR dataset and our self-built dataset and the experimental results demonstrate the effectiveness of the distribution-based scene representation and the denosing rearrangement model.

Acknowledgements

This work was supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123100).

References

[1] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rear-

rangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020. 1, 2, 3

[2] Ohad Ben-Shahar and Ehud Rivlin. Practical pushing planning for rearrangement tasks. *IEEE Transactions on Robotics and Automation*, 14(4):549–565, 1998. 2

[3] Akansel Cosgun, Tucker Hermans, Victor Emeli, and Mike Stilman. Push planning for object placement on cluttered table surfaces. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, pages 4627–4632. IEEE, 2011.

[4] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1614–1621. IEEE, 2019. 2

[5] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018. 2

[6] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural modular control for embodied question answering. In *Conference on Robot Learning*, pages 53–62. PMLR, 2018. 2

[7] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022. 6, 7

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 5

[9] Mehmet R Dogar, Michael C Koval, Abhijeet Tallavajhula, and Siddhartha S Srinivasa. Object search by manipulation. *Autonomous Robots*, 36:153–167, 2014. 2

[10] Linxi Fan, Yuke Zhu, Jiren Zhu, Zihua Liu, Orien Zeng, Anchit Gupta, Joan Creus-Costa, Silvio Savarese, and Li Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Conference on robot learning*, pages 767–782. PMLR, 2018. 2

[11] Samir Yitzhak Gadre, Kiana Ehsani, Shuran Song, and Roozbeh Mottaghi. Continuous scene representations for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14849–14859, 2022. 1, 2, 3

[12] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020. 2

[13] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021. 2

[14] Hector Geffner and Blai Bonet. *A concise introduction to models and methods for automated planning*. Morgan & Claypool Publishers, 2013.

[15] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning and acting*. Cambridge University Press, 2016. 2

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 5

[18] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2

[19] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pages 1470–1477. IEEE, 2011. 2

[20] Erez Karpas and Daniele Magazzeni. Automated planning for robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):417–439, 2020. 2

[21] Jennifer E King, Marco Cognetti, and Siddhartha S Srinivasa. Rearrangement planning using object-centric and robot-centric action spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3940–3947. IEEE, 2016. 2

[22] Jennifer E King, Vinitha Ranganeni, and Siddhartha S Srinivasa. Unobservable monte carlo planning for nonprehensile rearrangement tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4681–4688. IEEE, 2017. 2

[23] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 2, 6

[24] Athanasios Krontiris, Rahul Shome, Andrew Dobson, Andrew Kimmel, and Kostas Bekris. Rearranging similar objects with a manipulator using pebble graphs. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1081–1087. IEEE, 2014. 2

[25] Harold W Kuhn. The hungarian method for the assignment. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, page 29, 2009. 7

[26] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, pages 1952–1961, 2023. 2, 5

[27] Weijie Li, Xinhang Song, Yubing Bai, Sixian Zhang, and Shuqiang Jiang. Ion: Instance-level object navigation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 4343–4352, 2021. 2

[28] Weiyu Liu, Tucker Hermans, Sonia Chernova, and Chris Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. In *Workshop on Language and Robotics at CoRL 2022*, 2022. 3

[29] Yuyi Liu, Xinhang Song, Weijie Li, Xiaohan Wang, and Shuqiang Jiang. A category agnostic model for visual rearrangment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16457–16466, 2024. 1, 2, 3, 4, 7

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 8

[31] Gabriel Sarch, Zhaoyuan Fang, Adam W Harley, Paul Schydlo, Michael J Tarr, Saurabh Gupta, and Katerina Fragkiadaki. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In *European conference on computer vision*, pages 480–496. Springer, 2022. 1, 2, 3, 7

[32] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 2

[33] Anthony Simeonov, Ankit Goyal, Lucas Manuelli, Yen-Chen Lin, Alina Sarmiento, Alberto Rodriguez Garcia, Pulkit Agrawal, and Dieter Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. In *Conference on Robot Learning*, pages 2030–2069. PMLR, 2023. 3

[34] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 5, 7

[36] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 1, 2

[37] Brandon Trabucco, Gunnar Sigurdsson, Robinson Piramuthu, Gaurav S Sukhatme, and Ruslan Salakhutdinov. A simple approach for visual rearrangement: 3d mapping and semantic search. *arXiv preprint arXiv:2206.13396*, 2022. 1, 2, 3, 7

[38] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2, 5

[39] Xiaohan Wang, Yuehu Liu, Xinhang Song, Beibei Wang, and Shuqiang Jiang. Generating explanations for embodied action decision from visual observation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2838–2846, 2023. 2

[40] Xiaohan Wang, Yuehu Liu, Xinhang Song, Beibei Wang, and Shuqiang Jiang. Camp: Causal multi-policy planning for interactive navigation in multi-room scenes. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[41] Qiuhong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. Lego-net: Learning regular rearrangements of objects in rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19037–19047, 2023. 3

[42] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931, 2021. 1, 2, 3, 6, 7, 8

[43] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668, 2019. 2

[44] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 2

[45] Haitao Zeng, Xinhang Song, and Shuqiang Jiang. Multi-object navigation using potential target position policy function. *IEEE Transactions on Image Processing*, 32:2608–2619, 2023. 2

[46] Sixian Zhang, Xinhang Song, Yubing Bai, Weijie Li, Yakui Chu, and Shuqiang Jiang. Hierarchical object-to-zone graph for object navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15130–15140, 2021.

[47] Sixian Zhang, Weijie Li, Xinhang Song, Yubing Bai, and Shuqiang Jiang. Generative meta-adversarial network for unseen object navigation. In *European Conference on Computer Vision*, pages 301–320. Springer, 2022.

[48] Sixian Zhang, Xinhang Song, Weijie Li, Yubing Bai, Xinyao Yu, and Shuqiang Jiang. Layout-based causal inference for object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10792–10802, 2023. 2