2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Multimodal Diffusion Forcing for Forceful Manipulation

Zixuan Huang, Huaidian Hou, Dmitry Berenson
University of Michigan, Ann Arbor

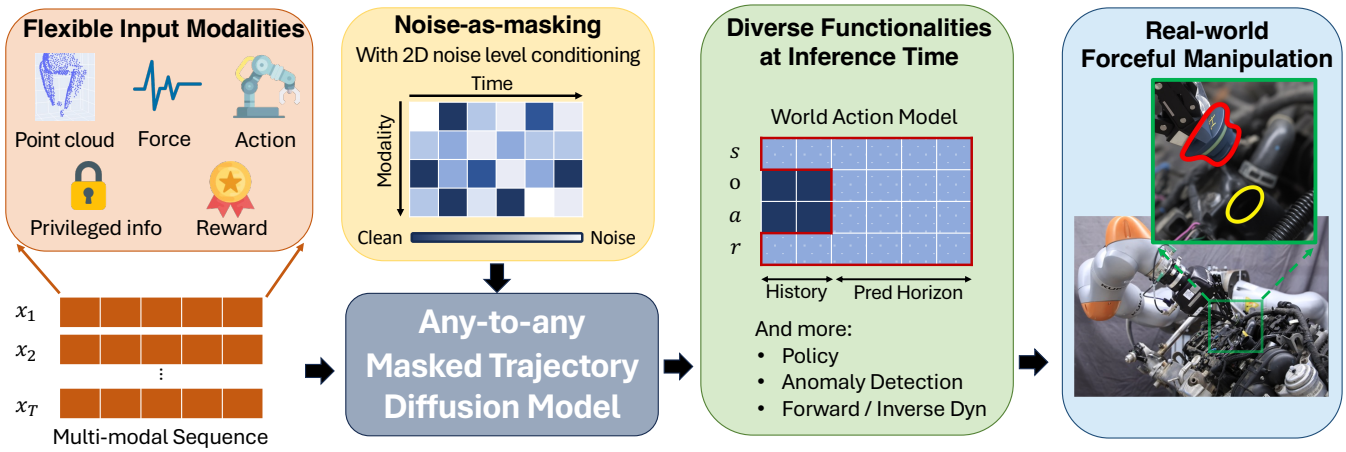

Fig. 1: We propose *Multimodal Diffusion Forcing*, a unified model that captures the interplay between modalities over time through partially masked training. At inference time, the model not only offers flexibility by allowing different input modalities, adjustable horizon lengths and prediction horizons, it also provides diverse functionalities — serving as a policy, world action model, dynamics model, and anomaly detector.

***Abstract*— Given a dataset of expert trajectories, standard imitation learning approaches typically learn a direct mapping from observations (e.g., RGB images) to actions. However, such methods often overlook the rich interplay between different modalities, i.e., sensory inputs, actions, and rewards — which is crucial for modeling robot behavior and understanding task outcomes. In this work, we propose *Multimodal Diffusion Forcing*, a unified framework for learning from multimodal robot trajectories that extends beyond action generation. Rather than modeling a fixed distribution, MDF applies random partial masking and trains a diffusion model to reconstruct the trajectory. This training objective encourages the model to learn temporal and cross-modal dependencies, such as predicting the effects of actions on force signals or inferring states from partial observations. We evaluate MDF on contact-rich, forceful manipulation tasks in simulated and real-world environments. Our results show that MDF not only delivers versatile functionalities, but also achieves strong performance, and robustness under noisy observations. More visualizations can be found on our website https://unified-df.github.io**

I. Introduction

Humans naturally integrate visual, audio, tactile, and proprioceptive signals to understand and interact with the physical world. For example, when inserting a key into a lock or tightening a bolt, we adjust our motion based on visual alignment and subtle resistance felt through touch. Similarly, robots performing contact-rich tasks must reason over diverse sensory inputs to perceive object states, predict outcomes, and react appropriately.

This work was supported in part by the Office of Naval Research under grants N00014-24-1-2036 and N6833525C0329 and NSF grants IIS-2113401 and IIS-2220876. Robotics Department, University of Michigan, Ann Arbor, MI, USA, contact zixuanh@umich.edu.

Despite the richness of multimodal sensory data in robotic systems, most existing learning methods [1], [2], [3], [4] focus on direct mappings from observations to actions. These approaches often overlook the complex interplay between modalities. Furthermore, existing approaches typically assume a fixed set of input modalities and lack robustness to partial or corrupted observation at inference time.

In this work, we propose *Multimodal Diffusion Forcing* (MDF), a unified framework for learning the joint distribution of multimodal robot trajectory. Unlike standard diffusion models that use a single, global noise level, MDF is trained using a 2D Time-Modality Noise Level Matrix. This unique training scheme endows MDF with the following properties:

Capturing cross-modal correlations over time. Our masked training strategy randomly corrupts input modalities and requires the model to recover them from the remaining context. This encourages the model to learn temporal dependencies both within and across modalities.

Flexibility at training and inference time. The model is trained to condition on arbitrary subsets of modalities and predict the rest, all controlled via the 2D noise level matrix. During training, we can include privileged modalities such as full point clouds, even if they are not available at test time. This form of privileged learning has been shown to improve robustness by encouraging the model to infer privileged information from partial observations [5], [6]. At inference time, a single MDF model can be flexibly deployed for a range of downstream tasks, including policy, dynamics modeling, and fine-grained anomaly detection.

Robustness to noisy inputs. Since MDF is trained with a *continuous* scale of corruption level, it is robust to a wide

range of noisy or missing observations at test time compared to models trained with binary masking.

We evaluate MDF in five contact-rich forceful manipulation tasks in simulation and the real world. All tasks require high-precision prediction and multimodal reasoning. We show that the performance of MDF is on par with specialized models and may surpass them when noisy observations are present. We also demonstrate the wide range of test-time capabilities of MDF.

II. Related work

A. Diffusion Models for Robotics

Diffusion models have been widely adopted in robotics for tasks such as policy learning [7], [8], [9], [10], forward dynamics modeling [11], [12], subgoal prediction [13], [14], [15], [16], and joint state-action modeling [17]. They have also been used for anomaly detection by estimating the log-likelihood via a variational bound [18]. Most prior models are trained for a single task. In contrast, our model's modality- and timestep-specific masking allows zero-shot adaptation to a variety of downstream tasks, including action generation, dynamics prediction, state estimation, and anomaly detection.

Recently, UWM [19] and UVA [20] introduced similar unified frameworks for behavior learning. While they primarily focus on video and action, our approach generalizes to multimodal sequences incorporating point clouds and force. Moreover, our method is more robust to sensory noise and offers greater test-time flexibility with configurable history lengths and input modalities. Due to the demanding compute requirement of UVA, we only compare with UWM.

B. Learning from Multimodal Data

There has been a growing interest in integrating multimodal information into robotic systems, including vision [21], [10], language [22], [23], audio [1], [24], force [25], [26], [27], and tactile signals [2], [28], [6], [29]. A common paradigm is to train policies that directly fuse multi-sensor inputs into actions. More recently, several works have explored jointly predicting both future actions and observations, such as images [30], force [29], or tactile signals [31], and have shown that this improves policy performance. However, these approaches typically rely on fixed input–output structures and assume access to a complete and consistent set of modalities. In contrast, MDF generalizes across varying input–output structures and supports multiple functionalities beyond action generation. Moreover, its noise-as-masking training paradigm provides richer supervision signals beyond dynamics learning alone.

C. Masked Training for Robotics

Recently, researchers attempted to apply masked training to robotics [32], [33], [34], [35], [20], [19], demonstrating the benefits of multimodal joint learning and the flexibility it offers for various downstream tasks. However, existing methods often rely on the low-dimensional state and struggle to handle high-dimensional observation [27], [33]. Moreover, many approaches adopt a binary masking scheme [27], [33], [34], [35], [20]. This limits the models to only reasoning about fully clean or completely missing data, but not anything in between. In contrast, we built on the idea of noise-as-masking [36] and train MDF with a continuous masking theme. This design enables fine-grained partial masking across time and modality, allowing the model to reason over partially corrupted observations robustly.

III. Preliminaries

A. Diffusion Models

Diffusion models gradually transforms data into noise through a forward Markov process, and then learns to reverse this process to reconstruct the original data distribution. The forward process $q(x^k \mid x^{k-1})$ adds Gaussian noise at each diffusion step k, defined as:

$$q(x^k \mid x^{k-1}) = \mathcal{N}\left(x^k \mid \sqrt{1-\beta_k}\, x^{k-1}, \beta_k \mathbf{I}\right), k = 1, \ldots, K$$

where β_k controls the noise variance at each step. The overall generative objective is to maximize the log-likelihood of reverse diffusion process under the learned model p_θ, which is intractable and thus optimized via a variational lower bound:

$$\mathbb{E}_{q(x^0)}[\log p_\theta(x^0)] \geq \mathbb{E}_{q(x^{0:K})}\left[\log \frac{p_\theta(x^{0:K})}{q(x^{1:K} \mid x^0)}\right]$$

As shown in DDPM [37], the distribution of the reverse process can be represented as $p_\theta(x^{k-1} \mid x^k) = \mathcal{N}\left(x^{k-1}; \boldsymbol{\mu}(x^k, k), \gamma_k \mathbf{I}\right)$, where γ_k is a fixed constant depending on k. Then, DPPM [37] proposed to reparametrize the mean μ with noise prediction according to $\epsilon = (\sqrt{1-\bar{\alpha}_k})^{-1} x^k - \sqrt{\bar{\alpha}_k}\mu$, which leads to the following objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{k, \mathbf{x}^0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon}^k - \boldsymbol{\epsilon}_\theta(\mathbf{x}^k, k)\right\|^2\right] \tag{1}$$

B. Problem Setup

We consider the problem of modeling multimodal robot trajectories from an offline dataset of expert demonstrations. We represent a trajectory as $\tau = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, where each timestep $\mathbf{x}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,m}\}$ consists of a set of M modality-specific features. Here, the notion of "modality" is general: it encompasses not only multimodal observations $o_t \in \mathcal{O}$ (e.g., images, point clouds, force signals), but also actions $a_t \in \mathcal{A}$, rewards $r_t \in \mathcal{R}$, and privileged states $s_t \in \mathcal{S}$ (e.g., object pose, velocity, or full-scene reconstructions) that are typically unavailable at test time. This abstraction allows us to build a unified model capable of handling heterogeneous data types within a single framework.

Our objective is to learn a unified generative model over multimodal robot trajectories that captures the complex interplays of different modalities and can be used for various downstream tasks.

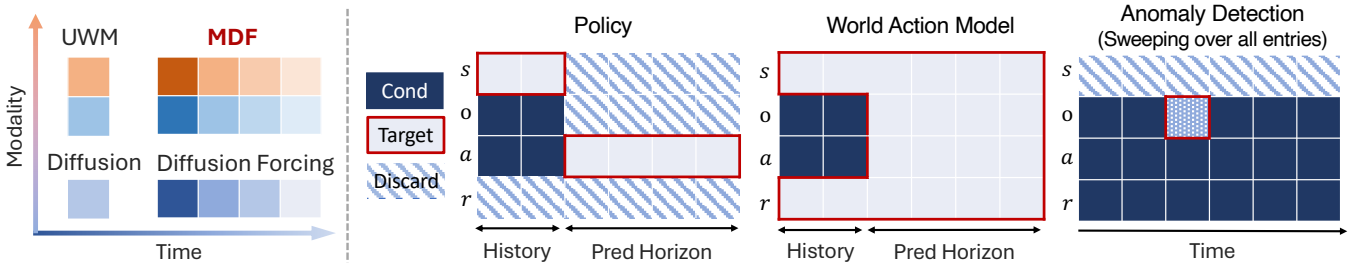

Fig. 2: Typical diffusion models use a scalar noise level to control the denoising process. Diffusion Forcing [36] proposes a time-varying noise vector to sample video sequences autoregressively. We further generalize this framework to the multimodal setting by introducing a time–modality varying noise matrix. This design enables versatile functionalities at test time such as policy, world action model, dynamics model, and fine-grained anomaly detector.

IV. Method

In section IV-A, we introduce the generalized multimodal diffusion forcing training scheme and the architecture. In section IV-B, we present the flexible inference-time capability of MDF. In section IV-C, we summarize the key training and implementation details.

A. Generalized Masked Training with Multi-modal Diffusion Forcing

a) Noise as masking: Masked training approaches [38], [39] usually train models to predict missing parts of the input with a binary mask. Diffusion models, which learn to iteratively denoise Gaussian-corrupted inputs, can be viewed as a continuous masking scheme where noise serves as partial masking [36]. A noise level of zero denotes an unmasked token, whereas a maximal noise level corresponds to complete masking. The partial masking aligns more with natural corruption in robotics (i.e. noisy state estimation, partial occlusion).

However, standard diffusion models typically apply a scalar noise level uniformly across all data, such as a full image [37], video [40] or trajectory [17]. This design introduces two limitations. First, during training, the model underutilizes the multimodal supervision by learning only a fixed global corruption pattern. Second, at test time, it lacks the flexibility to selectively condition on arbitrary subsets of modalities and timesteps, which is critical for partially observed or corrupted sequences.

To address these challenges, we extend Diffusion Forcing [36] to the multimodal setting by introducing a 2D *Time-Modality Noise Level Matrix* $\mathbf{K} \in \{0, \dots, K\}^{T \times M}$, where K is the number of diffusion steps, M is the number of modalities, and T is length of the trajectory (Fig. 2. $k_{t,m}$ specifies the noise level applied to modality m at timestep t. During training, the Gaussian noise ϵ and noise levels k are sampled independently across modalities and timesteps, allowing each part of the multimodal sequence to be partially corrupted to a different extent. Given a noise level matrix $\mathbf{K}$, the forward diffusion process can be written as follows:

$$\mathbf{x}_{t,m}^{k_{t,m}} = \sqrt{\bar{\alpha}_{k_{t,m}}}\, \mathbf{x}_{t,m}^{0} + \sqrt{1 - \bar{\alpha}_{k_{t,m}}}\, \boldsymbol{\epsilon}_{t,m}, \quad \boldsymbol{\epsilon}_{t,m} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Here, $\mathbf{x}_{t,m}^{k_{t,m}}$ denotes the noised version of the original clean feature $\mathbf{x}_{t,m}^{0}$ at timestep t for modality m, corrupted according to its assigned noise level $k_{t,m}$. Thus, the multimodal diffusion forcing model can be parameterized by $\epsilon_\theta\left(\tau^{\mathbf{K}}, \mathbf{K}\right)$, which takes the entire multimodal sequence and the 2D noise level matrix as input. The model is trained with the standard DDPM objective [37]:

$$\mathcal{L} = \mathbb{E}_{\mathcal{X}, \mathbf{K}, \boldsymbol{\epsilon}} \left[\sum_{t=1}^{T} \sum_{m=1}^{M} \left\| \epsilon_{t,m} - \epsilon_\theta\left(\tau^{\mathbf{K}}, \mathbf{K}\right)_{t,m} \right\|^2 \right],$$

In this work, we consider six modalities (Fig. 3): partial point cloud, full point cloud (training-time only), force, action, proprioception, and reward. Although full object point clouds are only available in simulation, we include it to encourage the model to implicitly reason about the objects interaction given partial point cloud. This can be seen as a form of occlusion reasoning or privileged learning [5]. As shown in the experiments, we find it to be critical to the performance of our method.

b) Architecture: Our multimodal diffusion forcing model is implemented as a bi-level diffusion framework (Fig. 3), consisting of diffusion-based point cloud autoencoders and a latent diffusion transformer.

Given a multimodal trajectory, we first transform the raw features of each modality into compact vector embeddings suitable for sequence modeling. While prior works [19], [20] primarily use sequences of images as observation, we focus on point clouds, as they capture rich geometric information that is critical for manipulation tasks. However, unlike images that can be efficiently encoded/decoded using a VQ-VAE [41], point clouds are high-dimensional and unordered, making them considerably more challenging to model. To obtain meaningful embeddings efficiently, we train a diffusion-based point cloud autoencoder [42] composed of a lightweight PointNet encoder and an expressive diffusion decoder. The autoencoder is trained to map point clouds into low-dimensional latent embeddings and reconstructs them through iterative denoising.

Then, we adopt a latent diffusion transformer [36] for sequence modeling. At each timestep, we concatenate encoded feature vectors of each modality together with their noise-level embeddings, producing a fused multimodal feature vector. The resulting sequence of fused vectors is then fed into a latent diffusion transformer, which models the bidirectional temporal dependencies and captures cross-modal interactions over time. Importantly, the multimodal diffusion model operates entirely in the latent space, and the expensive point cloud decoding step can be skipped at test time.

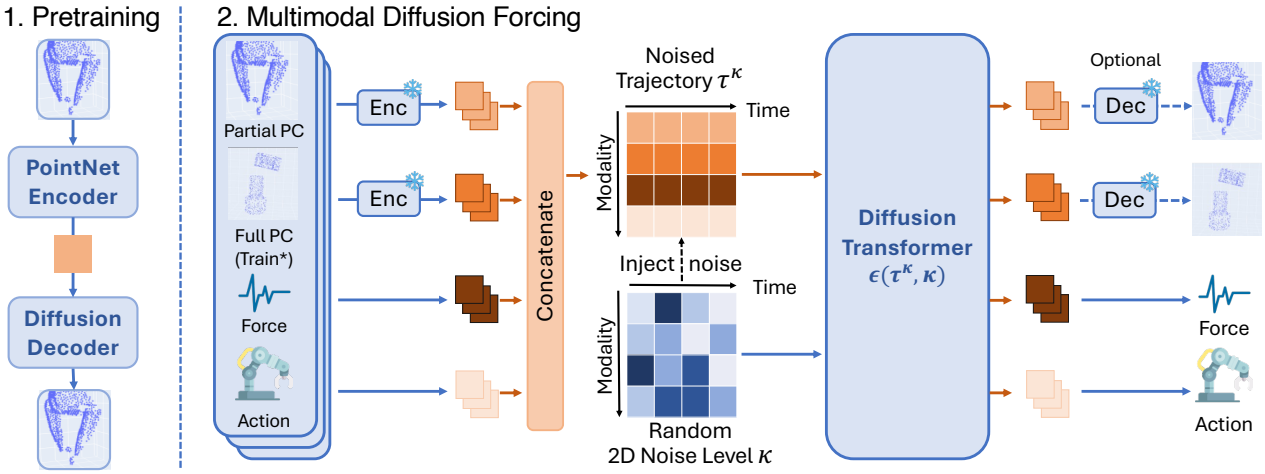

Fig. 3: Pretraining: MDF learns a diffusion-based autoencoder to compress point clouds into compact embeddings. **Multimodal masked training**: MDF processes six modalities: processes six modalities: partial point cloud, full point cloud (training only), force, action, reward, and proprioception; the reward and proprioception are omitted in the figure for clarity. The point clouds are tokenized using the pretrained PointNet encoder (frozen during MDF training). Data from all modalities are then concatenated and corrupted with noise according to a randomly sampled 2D noise level matrix. The diffusion transformer is trained to denoise this corrupted input, learning temporal and cross-modal dependencies.

B. Flexible Inference-time Capabilities

During inference, we can query the trajectory diffusion model ϵ_θ as an arbitrary conditional distribution by configuring the noise level matrix $\mathbf{K}$. As illustrated in Fig. 2, the matrix is partitioned into three types of blocks. **Condition blocks** are assigned near-zero noise levels throughout the diffusion process, ensuring that their information is preserved. **Target blocks** are initialized with Gaussian noise and assigned a high noise level. This noise level is then gradually denoised to zero according to a predefined noise schedule. Although different denoising schedules can be specified for each entry [36], we find that denoising the full sequence with a uniform timestep across target modalities is more efficient than autoregressive schemes. Finally, **discard blocks** are assigned the maximum noise level and remain noisy throughout the denoising process.

With these three types of blocks, we can flexibly configure the sampling distribution to support diverse functionalities. For instance, the model can function as a *policy* by conditioning on past observations to predict future actions, or as a *world action model* by additionally generating future states and observations. It can also serve as an *inverse dynamics model*, predicting actions from observations.

Practical Flexibility in Deployment. Beyond these roles, the same mechanism enables practical flexibility, such as varying the history length to suit task requirements or masking out unavailable sensor modalities while still producing coherent trajectories. For example, when deploying the model on a robot without a force sensor, the force modality can either be treated as a prediction target in forceful tasks or discarded entirely in others.

Fine-grained Anomaly Detection. MDF also enables localized likelihood estimation by injecting noise selectively into specific timesteps and modalities, rather than globally corrupting the entire trajectory. This design makes it possible not only to detect anomalies but also to precisely identify their source.

Diffusion models are trained to maximize a variational lower bound of the data log-likelihood (Eq. 1), and have therefore been applied to likelihood-based tasks such as density estimation [43] and anomaly detection [9], [18]. In this setting, anomalous samples are those with low likelihood under the learned distribution. Given a trajectory τ, its likelihood can be estimated by *measuring how well the diffusion model can recover τ after it has been corrupted by k steps of the forward diffusion process* [9]. Concretely, Gaussian noise is added to τ through the forward process $q(\tau^k)$, after which we compare the posterior $q(\tau^{k-1} \mid \tau^k, \tau^0)$ with the learned reverse process $p_\theta(\tau^{k-1} \mid \tau^k)$. We estimate the anomaly score D over a set of noise levels I as

$$D(I) = \frac{1}{|I|} \sum_{i \in I} D_{\mathrm{KL}} \left(q(\tau^{i-1} \mid \tau^i, \tau^0) | p_\theta(\tau^{i-1} \mid \tau^i) \right) \quad (2)$$

Building on this localized corruption mechanism, we design a fine-grained anomaly detection algorithm (Alg. 1) based on modality-time sweeping. Instead of perturbing the entire trajectory, our approach selectively injects noise into individual entries, each defined by a specific timestep and modality. By measuring how much each localized perturbation deviates from the model's expected behavior, the method can not only detect the presence of anomalies but also pinpoint their precise timestep and modality. For example, abnormal point cloud data may indicate a faulty or obstructed camera, while abnormal force readings likely suggest that the robot is experiencing an external disturbance

C. Training and implementation details

We first pretrain the point cloud tokenizers [42] separately on partial and full point clouds for 50k steps using a batch size of 512 and the Adam optimizer [44]. During MDF training, the pretrained tokenizers are frozen. The sequence length is set to 10, though both the history length and prediction horizon can be flexibly adjusted at test time.

As illustrated in Fig. 3, we randomly sample a 2D noise-level matrix during training. We adopt the Square Cosine

Schedule [45] with 1000 denoising steps for training. At inference time, we perform full-sequence denoising (sec. IV-B) with 200 steps using the DDIM sampler [46]. To further accelerate inference, we leverage CUDA Graphs, enabling the model to run at 10 Hz. The model is trained for 1 million gradient steps with the Adam optimizer [44], using a batch size of 1024 and a learning rate of 5×10^{-4}.

Algorithm 1: Anomaly Localization via Modality-Time Sweeping

Require: Trajectory $\mathbf{x}$, noise levels matrix $\mathbf{K} \in \{0, \dots, K\}^{T \times M}$, masked diffusion model p_θ

for each timestep t and modality m **do**
  Initialize noise matrix $\mathbf{K} \leftarrow \mathbf{0}$
  Set $\mathbf{K}_{t,m} \leftarrow i$
  Sample corrupted trajectory $\tau^{\mathbf{K}} \sim q(\tau^{\mathbf{K}} | \tau^0)$
  Compute and store KL divergence: $\mathcal{D}_{t,m} \leftarrow \mathrm{KL}\left(q(\tau^{k-1} \mid \tau^k, \tau^0) \,\|\, p_\theta(\tau^{k-1} \mid \tau^k)\right)$
  Compute KL divergence using Eq. 2
end for
return $(t^*, m^*) = \arg\max_{t,m} \mathcal{D}_{t,m}$

V. Experiments

We evaluate our methods on 3 contact-rich manipulation tasks in simulation and 2 forceful manipulation tasks in real world. Through the experiments, we seek to answer the following questions:

- **Performance**. MDF learns multiple probability distributions simultaneously. Can we match or even outperform the task-specific architecture for policy learning and anomaly detection?
- **Robustness**. Compared to baselines, does MDF provide additional robustness to sensory noise?
- **Flexibility**. To what extent can MDF be reconfigured at test time in terms of history length and input modalities?

A. Action Generation for Contact-rich Manipulation tasks

We first evaluate the robustness of our model in generating robot actions when provided with only partial or noisy observations. Specifically, we test on three simulated contact-rich manipulation tasks adapted from Nvidia Factory [47].

Nut Thread A KUKA iiwa robot equipped with a Robotiq 3F gripper is required to thread an M16 nut onto a fixed bolt. Success depends on aligning the nut within the thread tolerance before screwing. The primary difficulty stems from frequent occlusions of the nut by the rotating gripper.

Gear Mesh A Franka Emika Panda robot must insert a gear into a partially assembled gearbox. The gear must first be guided through a supporting peg, then simultaneously meshed with two neighboring gears of different sizes.

Peg Insert The robot must insert a cylindrical peg into a tight-fitting hole. The clearance is small, requiring accurate alignment and controlled contact.

1) Dataset collection: Teleoperation for contact-rich manipulation tasks in simulation is challenging due to the lack of feedback. We instead train a state-based RL policy using PPO to collect demonstrations paired with partial observations such as RGB image and point cloud. Collecting demonstrations with state-based RL for orbservation-based policy is a common technique in both online [5] and offline settings [48]. In this work, we focus on the offline supervised learning setting. We collect 10000 trajectories for Nut Thread and 7000 each for the Gear Mesh and Peg Insert.

Fig. 4: Contact-rich manipulation tasks in IsaacSim used in our experiments

Method	Nut Thread	Gear Mesh	Peg Insert
DP3 [10]	96%	80%	**84%**
UWM [19]	96%	54%	58%
MDF-Policy	**100%**	**86%**	80%
MDF-WA	92%	84%	78%
MDF-Policy-Noisy PC	**94%**	**84%**	**86%**
DP3-Noisy PC	78%	68%	76%
MDF-policy-No wrench	72%	78%	74%
MDF-Policy-No state estimation	74%	74%	70%

TABLE I: We evaluate every method for 50 random configurations and report the success rate.

2) Baselines and ablations: We select two state-of-the-art manipulation methods as baselines and ablate the design choice of our method.

Unified World Model (UWM) [19]. UWM trains a single model over video and action with independent noise levels. Similar to our method, UWM can also function as both a policy and a world model, but it lacks the capacity for geometric and force reasoning. Moreover, UWM has a fixed history length that cannot be adjusted at test time.

3D Diffusion Policy (DP3) [10]. DP3 incorporates 3D visual representations into diffusion policies [7] with a specially designed point encoder. DP3 is an optimized architecture for 3D policy learning.

MDF-Policy and MDF-WA. At test time, MDF can be sampled in two modes for action generation. In the *policy* mode, the model directly generates future actions. In the *world action* (WA) mode, the model additionally predicts future observations and states as consequences of the actions (see Fig. 2). In both modes, the model also estimates the history state, i.e., the full point cloud, which serves as an aggregated representation of object pose and geometry.

Ablations. We further examine the contributions of different components through ablation studies. Specifically, we evaluate *MDF-No Wrench*, which removes force signals to test the importance of force reasoning, and *MDF-Policy-No State Estimation*, where the history of full point cloud is not predicted during sampling.

3) Results: The success rates are reported in Table I.

Is MDF able to match the performance of state-of-the-art policy for manipulation? Although MDF is trained jointly to model multiple probability distribution, it is on-par with or even outperforms specialized architectures such

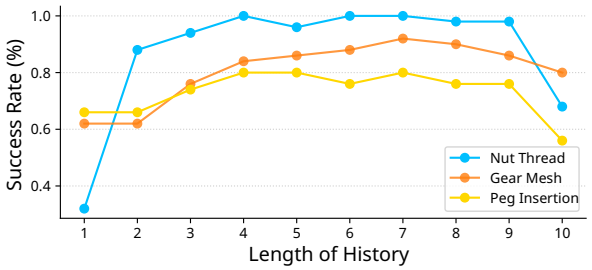

Fig. 5: The history length of MDF can be adjusted dynamically at test time to accommodate task requirements.

as DP3. For example, MDF-Policy achieves 100% success rate in Nut Thread, outperforming DP3's 96%, and remains competitive on other benchmarks. Compared to the previous unified model, UWM, MDF consistently achieves higher success rates, which we attribute to its ability to perform both 3D geometric and force reasoning. Moreover, MDF can additionally leverage the full point cloud if available, and achieve a 100% success rate on Peg Insert. While this setting provides privileged information, we argue that it remains realistic in semi-structured scenarios such as factories, where objects are known and specialized state estimation systems can be deployed to estimate poses reliably. The ability to accommodate different sensor setups is a major advantage over end-to-end policies with fixed inputs.

Does MDF training improve the robustness of the model to sensor noise? To assess the robustness of the model, we injected random translations into the point cloud input to mimic camera calibration errors. Under this perturbation, MDF maintains strong performance, dropping only 4% in Nut Thread and 2% in Gear Mesh, while DP3 dropped by 18% and 12%. This is because during training, MDF is trained to denoise the partially corrupted input.

Can MDF handle different history lengths? Unlike DP3 and UWM, which are trained with a fixed history length, the history length of MDF can be adapted at the test time based on the tasks requirements. The adaptibility to context length is a critical capability for large-scale multi-task learning. We demonstrate this capability of MDF in Fig. 5.

Is dynamics modeling helpful for action generation at test time? We also observe that MDF-WA, which incorporates explicit dynamics modeling into action generation, performs worse than MDF-Policy. We hypothesize that this is due to the relatively short-horizon nature of the evaluated tasks, which do not require complex, system-2–style reasoning.

Force reasoning and state estimation are important for contact-rich tasks. Removing the force input or restricting the model to sampling future actions without estimating history states leads to performance drops of 6%–28%.

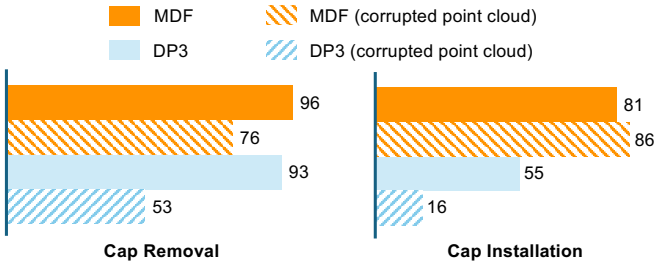

Fig. 6: MDF demonstrates better robustness to noisy sensory input.

Method	Wrench		Point cloud	
	Time	Time-Mod	Time	Time-Mod
MDF-sweeping	**73.8%**	**66.0%**	**100%**	**77.7%**
MDF-global	65.6%	48.9%	**100%**	63.6%
ImDiffusion	52.73%	3.52%	99.61%	5.47%

TABLE II: Results for anomaly localization

B. Anomaly Localizaiton

To evaluate the ability of our model to localize anomalous events, we design a fine-grained anomaly detection benchmark. Given a multimodal sequence, the goal is to precisely identify the timestep and modality in which anomaly occurs. To simulate anomalies, we corrupt point cloud observations by injecting random points and perturb wrench measurements with additive noise, mimicking external disturbances such as unexpected pushes. Each multimodal sequence is 10-steps long and contains 6 modalities, so a random guess will result in 1.67% accuracy.

Our proposed method, MDF-sweeping, localizes anomalies by selectively perturbing individual entries(Table 2). We compare against two baselines:

- **MDF-global** applies a global noise level uniformly across the trajectory (setting the same values for entire noise level matrix) and computes entry-wise likelihoods, but suffers from reduced accuracy since the corrupted context contaminates the reference distribution.
- **ImDiffusion** [49], a state-of-the-art approach that masks out time-series entries and imputes them with a diffusion model, using imputation errors as anomaly scores.

Table II summarizes the results. ImDiffusion achieves reasonable accuracy in identifying anomalous timesteps but fails to pinpoint the modality, as reflected by its sharp drop in the Time-Mod scores. MDF-global performs better but remains limited by global corruption. In contrast, MDF-sweeping achieves the highest localization accuracy across both wrench and point cloud modalities, particularly excelling in the challenging modality-time localization setting.

C. Real-World Car Maintenance Manipulation Tasks

We further evaluate our method on two challenging real-world car-maintenance tasks on a real vehicle engine with a 7 DoF Kuka LBR iiwa arm and Robotiq 3F gripper.

Oil Cap Installation A KUKA robot is required to install an oil cap in the oil filler port. We grade this task with three partial-credit checkpoints: alignment (0.5 pt), insertion and rotation (0.75 pt), and cap fully locked (1.0 pt).

Oil Cap Removal A KUKA robot is required to completely remove a locked oil cap from the filler port. Similar to the Installation task, we grade this task with four partial-credit checkpoints: alignment (0.25pt), grip (0.5pt), rotation to cap unlocked state (0.75pt), and removal (1.0pt).

1) Dataset Collection: Both Oil Cap Installation and Removal demonstration datasets are collected through teleoperation. Point cloud are captured by a Zivid 2 camera.

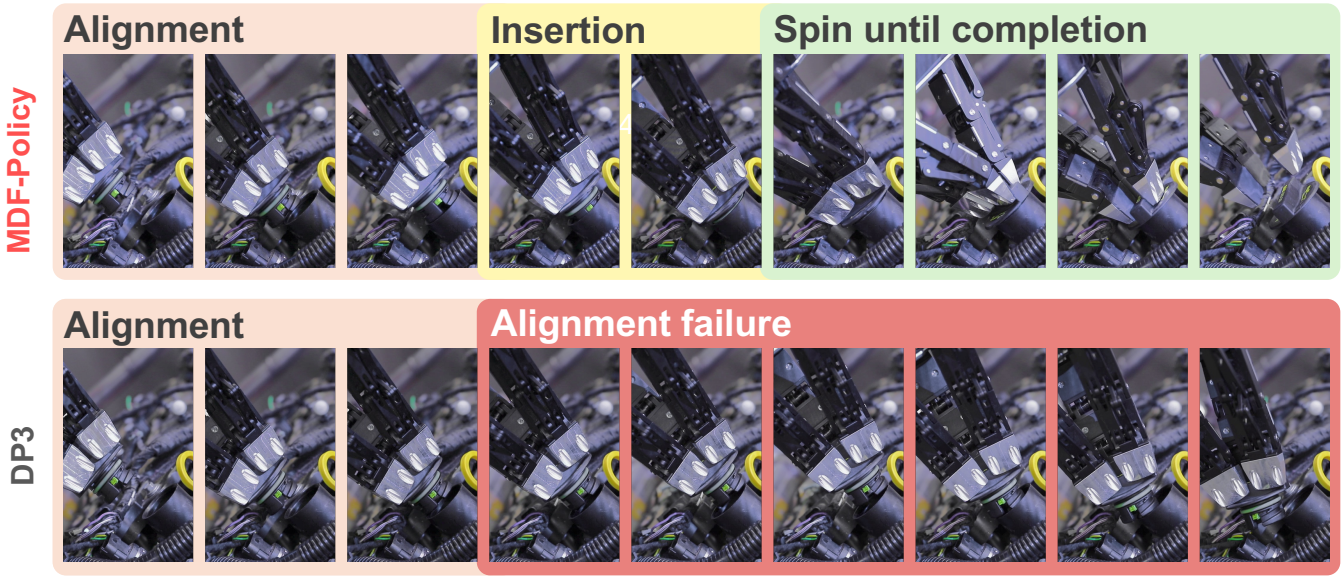

Fig. 7: We compare MDF with DP3 on two real-world forceful manipulation tasks. For each task, we compute the score over 20 trials (160 in total), the grading standards can be found in Section V-C. MDF's noise-as-masking training scheme allows it to be more robust to noisy observations.

2) Results: We compare our model with DP3 on both tasks. As shown in Fig. 6, MDF achieves up to a 26% higher success rate. In the Oil Cap Installation task, 3D Diffusion Policy often stops turning prematurely and loosens its grip before the cap is fully locked. We conjecture that the absence of force input prevents DP3 from accurately reasoning about the cap's locking state.

Robustness Against Corrupted Input. Further, we demonstrate MDF's robustness against noisy input. Specifically, we adopt an alternative camera profile with a shorter capture time, producing noisier point clouds with increased missing regions. MDF's ability to dynamically define input noise level on a specific modality excels in this scenario, outperforming DP3 by 23% and 70% respectively (Fig. 6).

An example rollout of the Oil Cap Installation task is presented in Figure 7. With identically corrupted point cloud perception, DP3 mis-aligns with the oil filler port and catastrophically fails to recover. Meanwhile, MDF is robust against noisy perception and successfully completes all three stages of this task.

Real-time Fine-grained Anomaly Detection. MDF enables fine-grained, per-modality anomaly detection at each timestep using Eq. 2. As illustrated in Fig. 8, the anomaly score rises selectively depending on the disturbance type. A visual distractor primarily increases the point cloud anomaly score while leaving the wrench modality largely unaffected. Conversely, an external physical push produces a pronounced spike in the wrench anomaly score, with minimal change in the score of point cloud. These results demonstrate MDF's ability to localize anomalies across modalities.

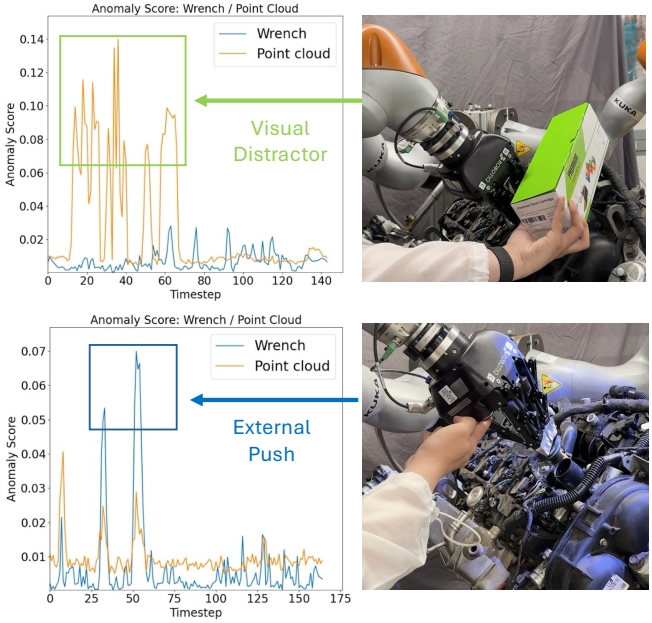

Fig. 8: MDF enables fine-grained, per-modality anomaly detection.

VI. Conclusion

We present Multimodal Diffusion Forcing (MDF), a unified framework for multimodal sequence modeling that captures the interplay of different modalities over time. MDF introduces a 2D *Modality-Time Noise Level Matrix* that enables fine-grained control over the sampling distribution. At test time, MDF demonstrates strong 3D and force reasoning. The framework is highly flexible, supporting variable sensory modalities and adaptable history lengths, and robust to noisy observations. Furthermore, we showcase its versatility through applications such as fine-grained anomaly detection.

Limitation and future work: 1. Training efficiency: MDF jointly learns to capture many distributions, which poses a challenging optimization problem. A more targeted training strategy could improve efficiency by focusing on the distributions most relevant to downstream tasks. 2. Heterogeneous training: MDF naturally supports learning from heterogeneous datasets containing different subsets of input modalities. Scaling this direction further could enhance generalization and improve performance across diverse mul-

timodal tasks.

REFERENCES

[1] M. Du, O. Y. Lee, S. Nair, and C. Finn, "Play it by ear: Learning skills amidst occlusion through audio-visual imitation learning," *arXiv preprint arXiv:2205.14850*, 2022.

[2] Y. Chen, A. Sipos, M. Van der Merwe, and N. Fazeli, "Visuo-tactile transformers for manipulation," *CoRL*, 2022.

[3] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song, "Maniwav: Learning robot manipulation from in-the-wild audio-visual data," in *8th Annual Conference on Robot Learning*, 2024.

[4] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik, "Learning visuotactile skills with two multifingered hands," *arXiv preprint arXiv:2404.16823*, 2024.

[5] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[6] I. Akinola, J. Xu, J. Carius, D. Fox, and Y. Narang, "Tacsl: A library for visuotactile sensor simulation and learning," *TRO*, 2025.

[7] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[8] L. Chen, S. Bahl, and D. Pathak, "Playfusion: Skill acquisition via diffusion from language-annotated play," in *Conference on Robot Learning*. PMLR, 2023, pp. 2012–2029.

[9] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu, "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," *arXiv preprint arXiv:2503.02881*, 2025.

[10] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[11] Z. Ding, A. Zhang, Y. Tian, and Q. Zheng, "Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning," *arXiv preprint arXiv:2402.03570*, 2024.

[12] M. Rigter, J. Yamada, and I. Posner, "World models via policy-guided trajectory diffusion," *arXiv preprint arXiv:2312.08533*, 2023.

[13] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" *arXiv preprint arXiv:2211.15657*, 2022.

[14] Z. Huang, Y. Lin, F. Yang, and D. Berenson, "Subgoal diffuser: Coarse-to-fine subgoal generation to guide model predictive control for robot manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 489–16 495.

[15] Z. Huang, Y. He, Y. Lin, and D. Berenson, "Implicit contact diffuser: Sequential contact reasoning with latent point cloud diffusion," *arXiv preprint arXiv:2410.16571*, 2024.

[16] C.-C. Hsu, B. Wen, J. Xu, Y. Narang, X. Wang, Y. Zhu, J. Biswas, and S. Birchfield, "Spot: Se (3) pose trajectory diffusion for object-centric manipulation," *arXiv preprint arXiv:2411.00965*, 2024.

[17] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *ICML*, 2022.

[18] S. Jiang, S. Ancha, T. Manderson, L. Brandt, Y. Du, P. R. Osteen, and N. Roy, "Anomalies-by-synthesis: Anomaly detection using generative diffusion models for off-road navigation."

[19] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta, "Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets," *arXiv preprint arXiv:2504.02792*, 2025.

[20] S. Li, Y. Gao, D. Sadigh, and S. Song, "Unified video action model," *arXiv preprint arXiv:2503.00200*, 2025.

[21] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[22] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control, 2024."

[23] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," *ICML*, 2022.

[24] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu, "See, hear, and feel: Smart sensory fusion for robotic manipulation," *CoRL*, 2022.

[25] M. Noseworthy, B. Tang, B. Wen, A. Handa, C. Kessens, N. Roy, D. Fox, F. Ramos, Y. Narang, and I. Akinola, "Forge: Force-guided exploration for robust contact-rich manipulation under uncertainty," *IEEE Robotics and Automation Letters*, 2025.

[26] J. H. Kang, S. Joshi, R. Huang, and S. K. Gupta, "Robotic compliant object prying using diffusion policy guided by vision and force observations," *IEEE Robotics and Automation Letters*, 2025.

[27] Y. Wu, Z. Chen, F. Wu, L. Chen, L. Zhang, Z. Bing, A. Swikir, S. Haddadin, and A. Knoll, "Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation," *arXiv preprint arXiv:2409.11047*, 2024.

[28] S. Cui, R. Wang, J. Wei, J. Hu, and S. Wang, "Self-attention based visual-tactile fusion learning for predicting grasp outcomes," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5827–5834, 2020.

[29] J. A. Collins, C. Houff, Y. L. Tan, and C. C. Kemp, "Forcesight: Text-guided mobile manipulation with visual-force goals," in *ICRA*, 2024.

[30] Y. Guo, Y. Hu, J. Zhang, Y.-J. Wang, X. Chen, C. Lu, and J. Chen, "Prediction with action: Visual policy learning via joint denoising process," *Advances in Neural Information Processing Systems*, vol. 37, pp. 112 386–112 410, 2024.

[31] L. Heng, H. Geng, K. Zhang, P. Abbeel, and J. Malik, "Vitacformer: Learning cross-modal representation for visuo-tactile dexterous manipulation," *arXiv preprint arXiv:2506.15953*, 2025.

[32] P. Wu, A. Majumdar, K. Stone, Y. Lin, I. Mordatch, P. Abbeel, and A. Rajeswaran, "Masked trajectory models for prediction, representation, and control," in *ICML*. PMLR, 2023, pp. 37 607–37 623.

[33] M. Carroll, O. Paradise, J. Lin, R. Georgescu, M. Sun, D. Bignell, S. Milani, K. Hofmann, M. Hausknecht, A. Dragan *et al.*, "Uni [mask]: Unified inference in sequential decision problems," *Advances in neural information processing systems*, vol. 35, pp. 35 365–35 378, 2022.

[34] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik, "Humanoid locomotion as next token prediction," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[35] F. Liu, H. Liu, A. Grover, and P. Abbeel, "Masked autoencoding for scalable and generalizable decision making," *Advances in neural information processing systems*, vol. 35, pp. 12 608–12 618, 2022.

[36] B. Chen, D. Martí Monsó, Y. Du, M. Simchowitz, R. Tedrake, and V. Sitzmann, "Diffusion forcing: Next-token prediction meets full-sequence diffusion," *Advances in Neural Information Processing Systems*, vol. 37, pp. 24 081–24 125, 2024.

[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[38] J. D. M.-W. C. Kenton, L. K. Toutanova *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, no. 2. Minneapolis, Minnesota, 2019.

[39] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[40] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022.

[41] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *NeurIPS*, vol. 30, 2017.

[42] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *CVPR*, 2021, pp. 2837–2845.

[43] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *Advances in neural information processing systems*, vol. 34, pp. 21 696–21 707, 2021.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[45] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International conference on machine learning*. PMLR, 2021, pp. 8162–8171.

[46] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[47] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu *et al.*, "Factory: Fast contact for robotic assembly," *arXiv preprint arXiv:2205.03532*, 2022.

[48] Z. Chen, Q. Yan, Y. Chen, T. Wu, J. Zhang, Z. Ding, J. Li, Y. Yang, and H. Dong, "Clutterdexgrasp: A sim-to-real system for general dexterous grasping in cluttered scenes," *arXiv preprint arXiv:2506.14317*, 2025.

[49] Y. Chen, C. Zhang, M. Ma, Y. Liu, R. Ding, B. Li, S. He, S. Rajmohan, Q. Lin, and D. Zhang, "Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection," *arXiv preprint arXiv:2307.00754*, 2023.