Dynamics-Compliant Trajectory Diffusion for Super-Nominal Payload Manipulation

Anuj Pasricha, Joewie Koh, Jay Vakil, Alessandro Roncone
Department of Computer Science
University of Colorado Boulder
firstname.lastname@colorado.edu

Abstract: Nominal payload ratings for articulated robots are typically derived from worst-case configurations, resulting in uniform payload constraints across the entire workspace. This conservative approach severely underutilizes the robot's inherent capabilities—our analysis demonstrates that manipulators can safely handle payloads well above nominal capacity across broad regions of their workspace while staying within joint angle, velocity, acceleration, and torque limits. To address this gap between assumed and actual capability, we propose a novel trajectory generation approach using denoising diffusion models that explicitly incorporates payload constraints into the planning process. Unlike traditional sampling-based methods that rely on inefficient trial-and-error, optimization-based methods that are prohibitively slow, or kinodynamic planners that struggle with problem dimensionality, our approach generates dynamically feasible joint-space trajectories in constant time that can be directly executed on physical hardware without post-processing. Experimental validation on a 7 DoF Franka Emika Panda robot demonstrates that up to 67.6% of the workspace remains accessible even with payloads exceeding 3 times the nominal capacity. This expanded operational envelope highlights the importance of a more nuanced consideration of payload dynamics in motion planning algorithms.

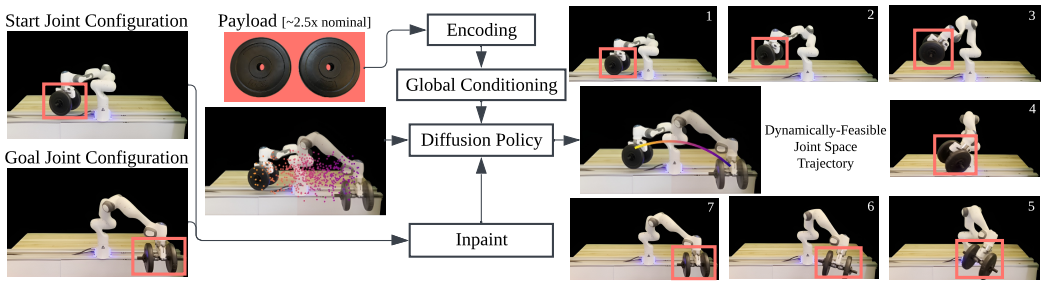

Figure 1: The diffusion model presented in this work learns to generate dynamically feasible trajectories directly in joint angle, velocity, and acceleration space, enabling super-nominal payload manipulation.

1 Introduction

Manipulator design establishes fundamental performance boundaries through the interplay of mechanical structure, actuation system, and control architecture. At the hardware level, these capabilities manifest as explicit physical constraints: joint angle limits that define the reachable workspace, maximum velocities bounded by motor specifications, and torque limits determined by the drivetrain capacity. However, a crucial distinction exists between these hardware limitations and the operational capacities specified by manufacturers for general-purpose deployment. While hardware limits are absolute physical constraints, nominal specifications—like payload capacity and effective workspace boundaries—are *derived* conservatively by considering worst-case scenarios across

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

possible configurations and trajectories. In practice, system integrators and end-users are locked into treating these derived operational capacities as hard limits rather than contextual capabilities, constrained by warranty requirements to design processes well within these conservative bounds. This well-established approach of *over-provisioning* hardware to ensure ample safety buffers has system-wide cascading effects—from the need to procure oversized actuators and power supplies, to reinforcing mounting structures for added load tolerance, to expanding facility requirements to accommodate larger equipment. For instance, when an application requires handling a 35kg payload, end-users are forced to select a 50kg-rated robot over a 30kg-rated system, despite the latter potentially being capable of safely executing the task within broad regions of its operational space.

Importantly, it is possible to overcome the issue of hardware over-provisioning through software-based optimizations, which allow to expand the safe operational envelope of a fixed/given robot embodiment and enable it to perform beyond nominal payload specifications. This requires unified handling of both geometric (i.e., kinematic) and dynamic constraints, as payload characteristics interact with joint configurations and higher-order derivatives to determine feasible motions. Even if they have not solved this specific problem, prior approaches in this space have investigated the challenge from multiple perspectives: geometric planners with dynamics post-processing [1, 2], kinodynamic planning [3], optimization-based methods [4], and control-based frameworks [5]. However, these methods face inherent trade-offs between design complexity, constraint handling, and computational efficiency; these become particularly critical when operating near system limits where the feasible solution space becomes highly constrained. These limitations highlight the need for a method that balances constraint satisfaction with generalizability—i.e., the ability to adapt to varying payloads, environmental conditions, and task requirements while maintaining consistent performance.

Of note, diffusion models [6, 7] present a compelling opportunity address the "curse of dimensionality" inherent in trajectory planning with simultaneous kinematic and dynamic constraint satisfaction. *By learning from successful trajectories generated by multiple (imperfect) planning methods*, these models can extract various homotopic solution classes that overcome the limitations of individual approaches. This distills successful solutions from diverse planning approaches into a unified representation that generalizes across the entire feasible space, including regions where individual planners fail. The inherent stochasticity in the denoising process enables the exploration of diverse, multimodal trajectories [6], while their capacity to model high-dimensional distributions allows incorporation of rich contextual information such as sensor data and problem constraints. However, existing diffusion-based approaches have been confined to position-controlled systems [8, 9], leaving the fundamental challenge of integrating both kinematic and dynamic constraints unaddressed.

In this work, we propose a method that directly tackles this problem by jointly learning valid distributions of positions, velocities, and accelerations that satisfy both constraint types simultaneously—moving beyond prior work that only denoises in position space. Through investigation of payload encoding strategies, we demonstrate that diffusion models effectively learn the complex relationships between joint states and payload-dependent dynamics, <u>enabling constant-time generation ($\approx$ 10ms) of feasible trajectories that preserve 67.6% workspace accessibility at 3$\times$ nominal payload capacity</u>—without explicit constraint checking at runtime. This demonstrates the importance of dynamics-informed generative models in expanding the operational boundaries of robotic systems.

2 Related Work

Manufacturer-specified operational parameters for general deployment of robots are typically far more conservative than hardware-level physical limits and often lack transparency in their derivation. This has led to significant under-utilization of robotic capabilities, as evidenced by geometric motion planning and time parameterization approaches dominating most industrial use [2, 10, 1] and recent advances in foundation models for manipulation in unstructured environments being restricted to exclusively geometric reasoning tasks [11, 12, 13]. A nuanced, configuration-dependent understanding of the robot's specifications that considers both robot and environment dynamics could enable a broader spectrum of manipulation capabilities within hardware limits.

Higher-order constraints such as joint torques and end-effector accelerations can be addressed through several approaches that range from "plan-and-filter" methods which post-process geometric paths [14, 15] to kinodynamic planners that directly incorporate dynamics during planning [16, 17, 18]. While these approaches offer completeness guarantees, they face practical limitations in high-dimensional manipulation spaces and often exhibit high variance in planning times especially when faced with state or model uncertainty. These challenges are particularly acute when planning near operational limits, where the feasible solution space becomes increasingly constrained [19]. Optimization-based methods naturally accommodate dynamics constraints but rely heavily on good initialization, leading to hybrid approaches that bootstrap the optimization process with learning or sampling-based solutions [20, 15, 21]. Optimal control formulations for dynamics-aware manipulation also exist [5, 22]. Despite their theoretical rigor, they face practical limitations in handling complex constraints and achieving reliable convergence for real-world manipulation problems.

Importantly, there exist a large body of works that demonstrate how a robot's behavioral repertoire can be significantly expanded when incorporating dynamic constraints—e.g. nonprehensile manipulation [23, 24, 25, 26, 27] or fast inertial transport [5, 4, 28, 29]. These capabilities and considerations are particularly relevant for the specific problem of payload manipulation, where the interaction between payload and induced joint torques significantly influence task success. Interestingly, payload manipulation presents unique challenges that intersect robot design and control. Traditionally, prior work has focused on mechanical modeling for specific low-DoF systems [30], with emphasis on payload capacity optimization [31] and joint torque considerations [32]. Other work in this domain tackles super-nominal payload handling through the use of multi-arm systems [33, 34]. However, the field lacks generalizable approaches for high-DoF systems that can reason about dynamic payload-robot interactions across diverse tasks.

Learning-based approaches offer promising directions for both generalizability and efficient trajectory generation in high-dimensional constrained spaces while respecting system dynamics. Autoregressive models implicitly encode dynamics through dataset design [35, 36], while recent diffusion-based approaches provide non-autoregressive alternatives with improved sampling diversity [8, 9]. Task-level diffusion policies, conditioned on language and visual embeddings [37, 6], demonstrate potential for generating diverse behaviors from high-level task specifications. Our method conditions a denoising diffusion model on the target payload to generate dynamically feasible trajectories.

3 Diffusion Models for Trajectory Generation

Diffusion models are a class of generative models that prdouce high-quality and diverse samples across multiple data modalities [38]. These models operate through a two-phase process: first, gradually corrupting training data with Gaussian noise in a forward process, then learning to reverse this corruption through an iterative denoising process. In the context of robot motion planning, we use diffusion to generate collision-free and dynamically feasible trajectories.

A joint space trajectory with t waypoints at denoising step k is defined as $\boldsymbol{\pi}^k = [X_0^k, X_1^k, \ldots, X_{t-1}^k]^T$ where $X_i = (q_i, \dot{q}_i, \ddot{q}_i) \in \mathbb{R}^{3n}$ represents the complete state (joint angles, velocities, and accelerations) of an n-DoF robot arm at timestep i. The training process for the diffusion model approximates the conditional distribution $p(\boldsymbol{\pi} \mid \mathcal{P})$, where $\mathcal{P}$ is the target payload representation (Section 4.2). It does so by sampling $\boldsymbol{\pi}$ from dataset $\mathcal{D}$, adding noise according to a noise schedule parameterized by α, γ, and σ, and learning to predict the underlying sample. The noise prediction network ϵ_θ, where θ represents the parameters of the 1D U-Net architecture, is trained to minimize $\mathcal{L}(\theta) = MSE(\epsilon^k, \epsilon_\theta(\mathcal{P}, \boldsymbol{\pi}^0 + \epsilon^k, k))$.

Given a trained ϵ_θ, the K-step iterative denoising process $\boldsymbol{\pi}^{k-1} = \alpha \cdot (\boldsymbol{\pi}^k - \gamma\epsilon_\theta(\mathcal{P}, \boldsymbol{\pi}^k, k) + \mathcal{N}(0, \sigma^2 I))$ starts from $\boldsymbol{\pi}^K \sim \mathcal{N}(0, I)$ and generates a trajectory $\boldsymbol{\pi}^0$ that can support a target payload and obeys start, goal, joint limit, and collision constraints. In practice, start and goal states are enforced via inpainting [8], i.e., $\boldsymbol{\pi}_0^k = X_{start}$ and $\boldsymbol{\pi}_{t-1}^k = X_{goal}$ where $\dot{q}_{start} = \dot{q}_{goal} = \ddot{q}_{start} = \ddot{q}_{goal} = \mathbf{0}$. Collisions are avoided via inference-time gradient guidance $\boldsymbol{\pi}^{k-1} = \boldsymbol{\pi}^k - \beta \cdot \nabla J(\boldsymbol{\pi}; E)$ where β is a tunable guidance weight, J defines the collision cost function, and E represents the set

of all obstacles in the environment [9]. Additionally, joint limits are enforced via clamping during the training and inference processes. Specific to our work, $p(\boldsymbol{\pi} \mid \mathcal{P})$ implicitly models dynamically feasible payload-induced torques through an object property-based global conditioning mechanism that influences the entire trajectory by appending target payload representation m to the diffusion timestep embedding in the noise prediction network architecture [6].

4 Payload-Conditioned Trajectory Denoising

In this section, we present our payload-conditioned diffusion model to generate pick-and-place trajectories that transport super-nominal payloads while obeying kinematic and dynamic constraints.

4.1 Training Data

Our payload-conditioned diffusion model (Section 4.2) is trained on $25,000$ time-parameterized, collision-free joint space trajectories (state dimension = DoF$\times$3, corresponding to joint angles, velocities, and accelerations) for tabletop pick-and-place motions. These trajectories are generated using a plan-and-filter pipeline that uses cuRobo, a GPU-accelerated trajectory optimization framework [39], followed by a dynamics validation step (Figure 2).

First, trajectories are generated for the 7DoF Franka Emika Panda robot arm with an effective payload mass of 0kg. Second, to ensure that payload-induced joint torques remain within hardware limits, each trajectory is verified through an analytically-derived dynamics model parameterized on real-world data [40]. This model captures both the standard dynamic elements (inertia, Coriolis, and gravity) and velocity-dependent joint friction. By using a dynamics model fitted to real robot trajectory data, we ensure that our training dataset reflects the physical constraints observed in hardware.

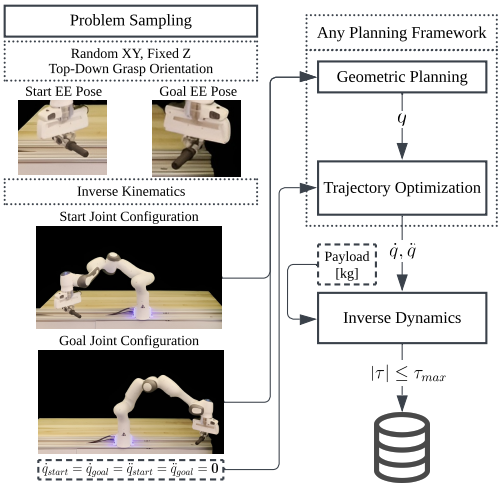

Figure 2: Plan-and-filter process to create training data.

This dynamics verification step utilizes a complete robot model where q, $\dot{q}$, $\ddot{q}$, and τ are 7×1 vectors representing joint angles, velocities, accelerations, and torques respectively. The dynamics matrices include: $M(q)$ (7×7 positive-definite, symmetric joint inertia matrix), $C(q, \dot{q})$ (7×7 Coriolis/centrifugal matrix), $g(q)$ (7×1 gravity compensation torque vector), and $f(\dot{q})$ (7×1 joint friction torque vector). External task space forces F_{ext} (6×1) map to joint torques through the Jacobian $J(q)$ (7×7). The dynamics model is parameterized as:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) + J^{-1}(q)F_{ext} \tag{1}$$

Note that this analytically-derived model represents the robot without external payloads which can affect model accuracy when considering varying payload masses in real-world deployment. To account for payload-induced joint torques, we model the gravitational effects as a 6D Cartesian wrench transformed from the world frame to the end-effector frame. This frame transformation is critical for accurately projecting gravitational forces acting on the payload along the robot's embodiment. We represent the external gravitational wrench $\boldsymbol{F}_g \in \mathbb{R}^6$ in the world frame with a simplified point mass model: $\boldsymbol{F}_g = mg \cdot [0,\ 0,\ -1,\ 0,\ 0,\ 0]^T$ where $m \in \mathbb{R}^+$ denotes payload mass and $g \approx 9.81$ m/s^2 is gravitational acceleration. This representation assumes force application at the object's center of mass, neglecting potential moment contributions. The wrench's structure combines linear forces (first three components) and angular forces (last three components), which are mapped to induced joint torques through the manipulator Jacobian, contributing to the total joint torque vector (eq. (1)).

The training dataset $\mathcal{D}$ is constructed from these trajectories that satisfy both kinematic and dynamic constraints. Each element $d_i \in \mathcal{D}$ is a tuple of trajectory $\boldsymbol{\pi}_i$ and maximum supported payload m_i. Each m_i is the maximum supported payload mass for trajectory i subject to $|\tau(\boldsymbol{\pi}_i, m_i)| \leq \tau_{max}$, $\forall t \in [0, T]$ where τ_t is computed via the dynamics equation (eq. (1)). Having established our dataset $\mathcal{D}$ of dynamically feasible trajectories and their corresponding maximum payload limits, we now consider different ways of incorporating payload conditioning into the diffusion model architecture.

4.2 Payload-Conditional Trajectory Generation

We extend the diffusion policy architecture to condition trajectory generation on payload mass through global conditioning [6]. Unlike local conditioning where features influence each waypoint individually, global conditioning applies the mass embedding uniformly across the entire trajectory generation process, preserving dynamic feasibility across the generated motion (Figure 3). For each training trajectory i with corresponding maximum supported payload mass $m_i \in \mathcal{D}$, we investigate four approaches to incorporate a target payload mass p_i into an encoding vector $\mathcal{P}_i$. The payload encoding $\mathcal{P}$ is concatenated with a diffusion timestep embedding to form a global conditioning vector, which is then integrated into ϵ_θ through either additive or affine transform conditioning [41].

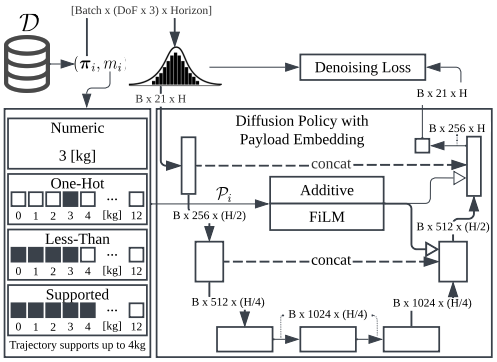

Figure 3: Our model conditions a 1D UNet denoising architecture [6] on various payload embeddings.

We consider the following approaches for designing $\mathcal{P}$. **Numeric Input.** A random supported mass scalar $p_i \sim \mathcal{U}(0, m_i)$ is sampled and fed directly to the network. This approach naturally supports continuous payload values during inference, despite training on a discrete set of payload values. **One-Hot Encoding.** For a system supporting payloads from $0 - 18$kg in 1kg increments, a random supported mass $p_i \sim \mathcal{U}(0, m_i)$ is encoded as a 19-dimensional binary vector with a single non-zero entry at index $\lceil p_i \rceil$. While this categorical representation simplifies the learning task by discretizing the payload space, it constrains inference to the discrete payload values seen during training. During inference, continuous payload masses must be quantized to their ceiling value $\lceil p \rceil$ before input to the denoising network, ensuring a tight upper bound for conservative operation. **Less-Than Encoding.** This encoding exploits the logical constraint that trajectories supporting a given payload must also support all lighter payloads. For a random supported mass $p_i \sim \mathcal{U}(0, m_i)$, we construct a 19-dimensional binary vector where entries are 1 for indices $j \leq \lceil p_i \rceil$ and 0 otherwise. This representation directly encodes the downward compatibility constraint into the input space, potentially improving the network's ability to learn and respect payload-dependent constraints. During inference, continuous payload masses are similarly mapped to their ceiling value $\lceil p \rceil$ to construct the binary encoding. **Supported-Range Encoding.** This approach encodes the complete range of supported payloads for a given trajectory. For training trajectory i with maximum supported mass m_i, we construct a 19-dimensional binary vector where entries are 1 for indices $j \leq \lceil m_i \rceil$ and 0 otherwise. During training, this encoding preserves the full payload compatibility information from $\mathcal{D}$. At inference time, for a target payload mass p, the encoding can be constructed either as a one-hot vector with a single non-zero entry at index $\lceil p \rceil$, or as a less-than encoding with ones for indices $j \leq \lceil p \rceil$. This dual interpretation enables both trajectory generation for specific payloads and exploration of the learned manifold of payload-trajectory relationships. Broadly, each encoding method presents trade-offs between generalization capability and training simplicity.

5 Evaluation

To validate our method, we focus on pick-and-place motions—a canonical industrial robotics requirement where super-nominal payload manipulation can significantly extend the operational en-

velope of robot arms while obeying strict hardware limits.[1] In our experiments, payloads were rigidly attached to the end-effector with a geometrically centered and uniform mass distribution. This was a deliberate experimental choice necessitated by the limited grasping force of our parallel gripper, ensuring repeatable execution across trials. Importantly, this choice is not a fundamental limitation of our method: as shown in the supplementary video, our approach is capable of handling non-rigidly attached objects and payloads with modest center-of-mass offsets, though some failure modes at higher payloads are also highlighted and explained there. During execution, invalid trajectories generated by the diffusion model are simply not executed. For valid trajectories that fail due to unmodeled dynamics (e.g., payload oscillations), standard robot safety stops are triggered automatically. Our approach relies on the manufacturer's joint velocity controller for trajectory tracking, which provides sufficiently accurate state tracking for the tasks evaluated in this work.

5.1 Comparison of Payload Encodings

Our evaluation of different payload encodings, conducted with FiLM conditioning and payload encoding normalization, reveals several key insights in terms of planning success rate (Figure 4). Numeric and one-hot approaches demonstrate comparable effectiveness, particularly for payloads within nominal capacity ($0-3$kg). However, all methods show a tendency to underestimate success rates for higher payloads, potentially due to data imbalance in the training set. When interpreted as a less-than encoding during inference, the supported-range encoding achieves competitive performance, but its one-hot interpretation leads to significantly lower success rates. While the encoding is designed to capture richer payload-trajectory relationships, the network struggles to fully utilize this information during generation. The inability of numeric, supported one-hot, and supported less-than encodings to achieve 100% success rates in the most loosely-constrained and data-rich portions of the state space, i.e., the nominal operating regime ($0-3$ kg), suggest fundamental limitations in how these encoding schemes enable the diffusion model to obey kinematic and dynamic constraints. For subsequent experiments, we use with the one-hot encoding scheme, which demonstrates superior performance in terms of success rate.

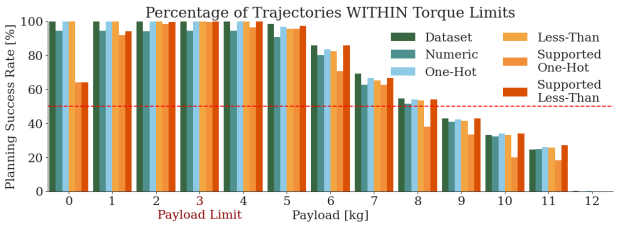

Figure 4: The aggregate success rate metric shows one-hot diffusion closely matching the underlying training data distribution.

5.2 Comparison across Planner Types

For the purposes of our discussion, we refer to DDIM (One-Hot) as a primary point of comparison. Denoising Diffusion Implicit Models (DDIM) offer accelerated inference due to fewer denoising steps (in our case, 5) [42]. We also present results for DDPM [38] with 25 denoising steps. Notably, our experiments demonstrate that DDIM achieves identical success rates to DDPM despite requiring significantly less planning time (≈ 0.01s), validating our approach for realtime use (Figure 5c).

We evaluate our approach against a diverse set of trajectory planning methods using mean planning time to first solution and the corresponding success rate over 500 tabletop manipulation tasks as the primary metric (Figure 5c). This metric provides a fair basis for comparison across fundamentally different planning paradigms. Additional metrics include planning time factor (mean planning time of baseline / mean planning time of DDIM) (Figure 5a), relative success rate change over DDIM as a percentage (Figure 5b), and standard deviations of planning times (Figure 5d).

Plan-and-Filter Methods Broadly, plan-and-filter methods employ a sequential process: first planning a geometric path via sampling-based motion planning, then time-parameterizing it to derive joint velocities and accelerations, and finally validating trajectory feasibility for the target payload

[1] Please refer to the supplementary material for more details on compute hardware, problem domain, experimental setup, and baseline methods. Videos are available here: `https://payload-diffusion.github.io/`

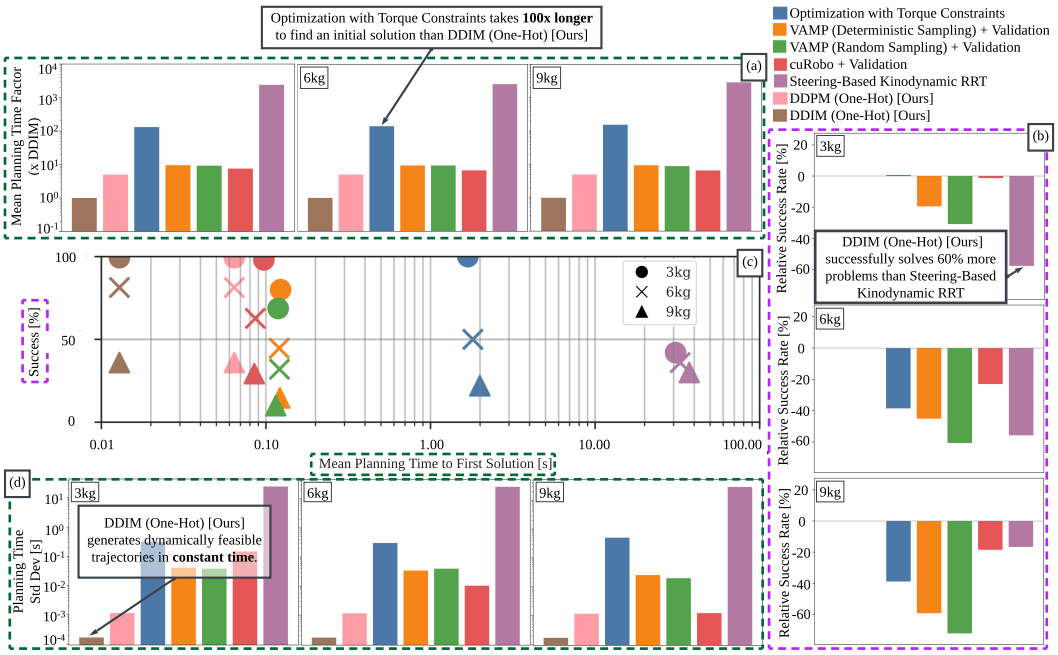

Figure 5: Comparative analysis of trajectory planning methods for payloads of 3kg, 6kg, and 9kg. DDIM (One-Hot) significantly outperforms a variety of baseline methods in both planning time to first solution and success rate, generating dynamically feasible trajectories in constant time across all payload conditions.

using the robot dynamics model (eq. (1)). This iterative process continues until a feasible trajectory is found that satisfies all joint torque constraints induced by the payload [14]. We evaluate two CPU-accelerated geometric planning variants: VAMP (RRTConnect) with deterministic sampling and VAMP (RRTConnect) with random sampling [43]. We select Ruckig for time parameterization due to its ability to generate time-optimal trajectories in realtime [2]. Geometric motion planning and time parameterization can alternatively be unified within a single optimization framework (in our case, cuRobo) which can be followed by the dynamics validity checking.

Since there is a decoupling between kinematic reasoning and dynamic reasoning, further exacerbated by the separate time parameterization step in the VAMP variants, these approaches demonstrate inferior performance compared to our one-hot DDIM model (Fig. 5c). VAMP with deterministic sampling achieves marginally higher success rates than VAMP with random sampling due to its low-dispersion sampling strategy, which provides more uniform exploration of the configuration space. Despite this sampling advantage, both variants exhibit higher standard deviations (Fig. 5d) and lower overall success rates, particularly with heavier payloads (Fig. 5b). These methods also exhibit significant planning time variability (Fig. 5a). The temporal predictability of our approach is particularly valuable for industrial pipelines, where high standard deviations in planning times can complicate integration with broader automation workflows. The success rate differential between cuRobo and DDIM particularly demonstrates that our model effectively learns from successful trajectories generated by imperfect planners. While cuRobo with validation often requires multiple attempts to find feasible solutions, our method leverages these curated trajectories during training to achieve higher success rates, especially under super-nominal payload conditions.

Kinodynamic Motion Planning Plan-and-filter methods generate paths requiring post-processing to satisfy dynamic constraints, potentially invalidating the solution. By planning directly in the state space that includes joint angles, velocities and accelerations, kinodynamic planning ensures dynamically feasible solutions from the outset. Specifically, we employ steering-based kinodynamic RRT as our baseline, planning in the joint angle, velocity, and acceleration space [44]. Ruckig serves as our steering function owing to its ability to operate in realtime while respecting third-order constraints [2]. Each edge undergoes validity checking that includes dynamics (eq. (1)) and

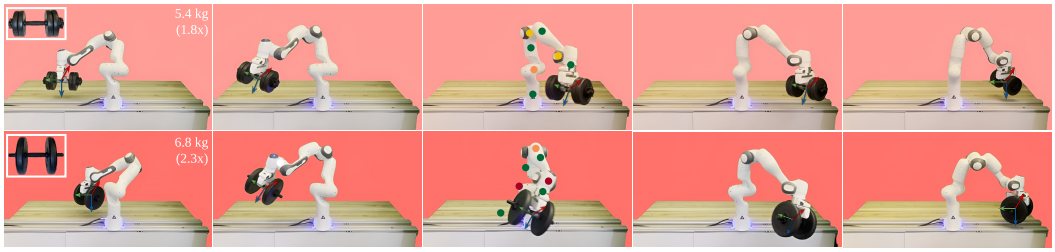

Figure 6: Two qualitative motions of the robot carrying super-nominal payloads of 5.4kg (1.8× nominal capacity) and 6.8kg (2.3× nominal capacity) are shown. Colored dots indicate joint torques as a percentage of joint limits: $0-20\%$, $20-40\%$, $40-60\%$, $60-80\%$, $80-100\%$. Joints 1, 3, and 5 exhibit higher torques, with joint 1 bearing the structural load of the arm and joints 3 and 5 positioning and orienting the payload.

smoothness checks before addition to the planning graph. Results not only indicate high planning times to find the first solution ($> 1000\times$ compared to DDIM) due to "the curse of dimensionality" (Figure 5a), but also show high standard deviations due to random sampling in the high-dimensional state space (Figure 5d). The clustering of success rates observed across all three payloads suggests an inherent planning bias in kinodynamic RRT, indicating insufficient exploration in this highly non-Euclidean state space (Figure 5a). Furthermore, kinodynamic approaches require extensive tuning of hyperparameters that interact with system dynamics in complex ways. Suboptimal parameter selection frequently results in planning timeouts or physically implausible trajectories.

Trajectory Optimization with Torque Constraints This baseline implements Sequential Least SQuares Programming (SLSQP) optimization for trajectory generation, minimizing sum-squared jerk to ensure smoothness [4]. To facilitate direct comparison with our method, we constrain trajectory durations instead of performing time optimization [39, 4], isolating evaluation to payload manipulation capabilities. The formulation incorporates linearized representations of both dynamic payload constraints and collision avoidance parameters. The success rate disparity between optimization and DDIM becomes particularly pronounced with higher payloads as the optimization landscape becomes increasingly constrained. This significant performance gap warrants discussion in the context of our evaluation choices. We implemented a basic SLSQP optimization formulation to establish a baseline that directly handles dynamic constraints without sophisticated warm-starting or acceleration techniques. Recent work has shown that optimization-based planners can achieve substantially better performance both in planning time and success rate through strategies like learning-based seeding [20, 45]. However, our baseline still highlights the computational challenge of finding feasible solutions in high-dimensional spaces with nonlinear dynamic constraints.

6 Conclusion

This work demonstrates that manipulators can safely operate well beyond their nominal payload ratings through diffusion-based trajectory generation, challenging the traditional approach of hardware over-provisioning in industrial automation. By training on trajectories that span the full range of payload-dependent dynamic constraints and leveraging the successes of "imperfect" planners, our model learns to generate motions for complex tasks in constant time that naturally respect system limitations without explicit constraint checking at runtime. This enables safe operation even at super-nominal payloads, maintaining 67.6% of the nominal workspace at $3\times$ rated capacity.[2]

Potential avenues for future work include studying how to determine the optimal composition of training data from different planners operating in varied environments. Moreover, the notion of conditioning on object-level characteristics can be generalized to handle objects with internal dynamics like liquids or articulated parts, moving beyond rigid-body assumptions. Such a framework could significantly reduce robotic system integration complexity and design burden by eliminating separate subsystems for different planning aspects, while maintaining the computational efficiency advantages and higher-order constraint satisfaction demonstrated in this work.

[2]More details on accessible workspace regions in the supplementary material.

7 Limitations

While our results establish the viability of super-nominal payload manipulation on fixed/given hardware, several aspects warrant further discussion. **Rigid Attachment Choice.** Even though the rigid attachment experimental choice simplifies analysis by eliminating offset moments, it introduces two critical limitations in real-world execution (see accompanying video): dynamic effects from payload oscillation when objects are not perfectly rigid, and unmodeled torques when the object's center of mass is significantly offset from the end-effector's origin. These assumptions can lead to trajectory tracking errors or even task failures, particularly when handling flexible materials or asymmetric objects. Looking ahead, extending our approach to objects with internal dynamics will require explicit robot-object dynamics models to capture effects such as rotational inertia, slippage, and uneven mass distributions. Incorporating these nonlinearities, together with friction modeling for fingered grippers and suction force modeling for suction-based end-effectors, remains an important direction. Moreover, expanding the conditioning space to whole-body contact forces, fingertip wrench cones, and suction dynamics, as well as embedding more advanced controllers (e.g., impedance or admittance), will allow the same "constraint-aware by design" principle to scale to interaction-rich manipulation tasks. **Test-Time Robustness.** Assessing test-time robustness requires long-term deployment studies and, in many cases, proprietary Original Equipment Manufacturer (OEM) data on wear-and-tear. Whether robots should operate continuously at or near maximum capacity is also a design question for robot OEMs. While our method generates trajectories that respect manufacturer-specified torque limits, joints do run hotter under high payloads, motivating further study of safety margins under sustained operation. **Embodiment Generalization.** We acknowledge that embodiment understanding must be more directly incorporated into the training process to enable generalization across different robotic platforms. To this end, we are developing more general embodiment-constrained conditioning mechanisms as part of ongoing work.

Our work also raises important theoretical questions about completeness and optimality guarantees for learning-based motion planning. Even though we can argue that diffusion models trained on optimal trajectories will generate similarly optimal solutions, establishing formal completeness guarantees remains an open challenge. Moreover, while our method demonstrates advantages in inference speed, traditional optimization-based approaches offer clearer constraint satisfaction guarantees and greater flexibility in adapting to new constraints without retraining. That said, training is not a bottleneck: we generate training trajectories across multiple payload conditions, with each trajectory verified for kinematic and dynamic feasibility throughout the robot's operational workspace. The compact dimensionality of our denoised samples enables fast retraining of the UNet and efficient creation of a high-quality dataset with comprehensive operational space coverage, but without the need for real-world payload manipulation data (assuming the existence of a robot dynamics model, which also lends to system modularity).

More rigorous evaluations comparing CPU versus GPU performance and Python versus C++ implementations would enable direct baseline comparisons but standardizing these benchmarks remains challenging. Our evaluation of baseline methods maintains industry-standard implementations of baselines without additional optimizations for deployment infrastructure. Several other under-explored areas of research remain: optimal training data selection for maximizing planner generalization, disambiguating data imbalance from model generalization to explain the high variance in torques for higher payloads, leveraging diffusion models' continual learning capabilities at industrial scale, and conditioning on real-world sensor modalities like point clouds for deployment.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback and constructive questions, which greatly improved the final version of this paper. This work was supported by the Office of Naval Research under Grant N00014-22-1-2482.

References

[1] A. Orthey, C. Chamzas, and L. E. Kavraki. Sampling-based motion planning: A comparative review. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

[2] L. Berscheid and T. Kröger. Jerk-limited real-time trajectory generation with arbitrary target states. *Robotics: Science and Systems XVII*, 2021.

[3] R. Shome and L. E. Kavraki. Asymptotically optimal kinodynamic planning using bundles of edges. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9988–9994. IEEE, 2021.

[4] J. Ichnowski, Y. Avigal, Y. Liu, and K. Goldberg. Gomp-fit: Grasp-optimized motion planning for fast inertial transport. In *2022 international conference on robotics and automation (ICRA)*, pages 5255–5261. IEEE, 2022.

[5] J. Arrizabalaga, L. Pries, R. Laha, R. Li, S. Haddadin, and M. Ryll. Geometric slosh-free tracking for robotic manipulators. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1226–1232, 2024. doi:10.1109/ICRA57147.2024.10610813.

[6] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[7] A. Sridhar, D. Shah, C. Glossop, and S. Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[8] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1916–1923. IEEE, 2023.

[9] K. Saha, V. Mandadi, J. Reddy, A. Srikanth, A. Agarwal, B. Sen, A. Singh, and M. Krishna. Edmp: Ensemble-of-costs-guided diffusion for motion planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10351–10358. IEEE, 2024.

[10] D. Coleman, I. Sucan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

[11] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[12] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[13] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[14] A. Abderezaei, A. Pasricha, A. Klausenstock, and A. Roncone. Clutter-aware spill-free liquid transport via learned dynamics. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9908–9915. IEEE, 2024.

[15] A. Kuntz, C. Bowen, and R. Alterovitz. Fast anytime motion planning in point clouds by interleaving sampling and interior point optimization. In *Robotics Research: The 18th International Symposium ISRR*, pages 929–945. Springer, 2019.

[16] Y. Li, Z. Littlefield, and K. E. Bekris. Sparse methods for efficient asymptotically optimal kinodynamic planning. In *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*, pages 263–282. Springer, 2015.

[17] S. Nayak and M. W. Otte. Bidirectional sampling-based motion planning without two-point boundary value solution. *IEEE Transactions on Robotics*, 38(6):3636–3654, 2022.

[18] A. Pasricha and A. Roncone. The virtues of laziness: Multi-query kinodynamic motion planning with lazy methods. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14286–14292. IEEE, 2024.

[19] Z. Kingston, M. Moll, and L. E. Kavraki. Sampling-based methods for motion planning with constraints. *Annual review of control, robotics, and autonomous systems*, 1(1):159–185, 2018.

[20] J. Ichnowski, Y. Avigal, V. Satish, and K. Goldberg. Deep learning can accelerate grasp-optimized motion planning. *Science Robotics*, 5(48):eabd7710, 2020.

[21] L. Yan, T. Stouraitis, J. Moura, W. Xu, M. Gienger, and S. Vijayakumar. Impact-aware bimanual catching of large-momentum objects. *IEEE Transactions on Robotics*, 2024.

[22] S. Kim, A. Shukla, and A. Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.

[23] M. R. Dogar and S. S. Srinivasa. A framework for push-grasping in clutter. In *Robotics: Science and systems*, volume 2, 2011.

[24] A. Pasricha, Y.-S. Tung, B. Hayes, and A. Roncone. Pokerrt: Poking as a skill and failure recovery tactic for planar non-prehensile manipulation. *IEEE Robotics and Automation Letters*, 7(2):4480–4487, 2022.

[25] F. Ruggiero, V. Lippiello, and B. Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

[26] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 30–37. IEEE, 2016.

[27] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020.

[28] R. I. C. Muchacho, R. Laha, L. F. Figueredo, and S. Haddadin. A solution to slosh-free robot trajectory optimization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 223–230. IEEE, 2022.

[29] Y. Avigal, J. Ichnowski, M. Y. Cao, and K. Goldberg. Gomp-st: Grasp optimized motion planning for suction transport. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 488–505. Springer, 2022.

[30] C.-Y. Wang, W. K. Timoszyk, and J. E. Bobrow. Payload maximization for open chained manipulators: finding weightlifting motions for a puma 762 robot. *IEEE Transactions on Robotics and Automation*, 17(2):218–224, 2001.

[31] M. Korayem and A. Nikoobin. Maximum payload for flexible joint manipulators in point-to-point task using optimal control approach. *The International Journal of Advanced Manufacturing Technology*, 38:1045–1060, 2008.

[32] H. C. Nho and P. Meckl. Intelligent feedforward control and payload estimation for a two-link robotic manipulator. *IEEE/ASME transactions on mechatronics*, 8(2):277–282, 2003.

[33] R. Kim, S. Balakirsky, K. Ahlin, M. Marcum, and A. Mazumdar. Enhancing payload capacity with dual-arm manipulation and adaptable mechanical intelligence. *Journal of Mechanisms and Robotics*, 13(2):021012, 2021.

[34] L. Yan, Z. Mu, W. Xu, and B. Yang. Coordinated compliance control of dual-arm robot for payload manipulation: Master-slave and shared force control. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2697–2702. IEEE, 2016.

[35] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox. Motion policy networks. In *Conference on Robot Learning*, pages 967–977. PMLR, 2023.

[36] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip. Motion planning networks: Bridging the gap between learning-based and classical motion planners. *IEEE Transactions on Robotics*, 37 (1):48–66, 2020.

[37] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sP1fo2K9DFG.

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

[39] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023.

[40] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. De Luca. Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4):4147–4154, 2019.

[41] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[42] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[43] W. Thomason, Z. Kingston, and L. E. Kavraki. Motions in microseconds via vectorized sampling-based planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8749–8756. IEEE, 2024.

[44] S. M. LaValle and J. J. Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5):378–400, 2001.

[45] H. Huang, B. Sundaralingam, A. Mousavian, A. Murali, K. Goldberg, and D. Fox. Diffusionseeder: Seeding motion optimization with diffusion for rapid motion planning. *arXiv preprint arXiv:2410.16727*, 2024.

Supplementary Material

1 Other Design Considerations

Input Normalization. The trajectories in our training set are normalized to the range $[-1, 1]$ to improve training stability. Given that payload masses in our dataset, $m \in \mathcal{D}$, are bounded within well-defined minimum and maximum values, we apply an analogous normalization transformation to map these payload values to $[-1, 1]$. This normalization ensures trajectory and payload features exist in comparable numerical ranges, which empirically improves training stability and convergence by preventing scale mismatches in the network's learning dynamics.

Conditioning Mechanism. We consider two global conditioning mechanisms to incorporate payload encoding and diffusion timestep information into ϵ_θ [1]. The first approach uses additive conditioning, where the global feature vector formed by a concatenation of the payload encoding and diffusion timestep embedding is projected through a learned embedding layer and added to the output of each convolutional block. The second approach implements an affine transform through Feature-wise Linear Modulation (FiLM) [2], where the global feature vector determines scale and bias parameters that modulate outputs from the convolutional layers. This method provides more expressive control over the generated trajectories due to its ability to selectively amplify or attenuate features based on payload requirements, thereby giving higher planning success rates relative to the additive approach.

2 Evaluation

2.1 Evaluation Hardware

All model training and evaluations are performed on a system with an Intel Xeon W5-2445 processor (3.10 GHz), 64 GB of RAM, and a single NVIDIA GeForce RTX 4090 GPU.

2.2 Problem Domain and Experimental Setup

We evaluate our approach on tabletop pick-and-place tasks using dumbbells as payloads to enable controlled experimentation. Our trained models and experiments assume that the payload's center of mass corresponds to its geometric center, and that the payload is rigidly attached to the robot's end-effector. The evaluation comprises 500 planning problems in a tabletop manipulation scenario using the 7DoF Franka Emika Panda robot arm, which has a manufacturer-specified nominal payload capacity of 3kg. Each test scenario consists of randomly sampled start and goal joint configurations, with the end-effector positioned above a table at a fixed height ($z = 0.2m$) for each terminal state and oriented downward to facilitate object grasping from the table surface. We initialize all start and goal states with zero joint velocities and accelerations.

2.3 Super-Nominal Payload Manipulation as a Novel Capability

Torque Distribution. The ridgeline plots in Figure 3 demonstrate systematic changes in torque distributions for the elbow (joint 3) and wrist (joint 5) with increasing payload mass. Both joints exhibit distributions that shift progressively toward more positive torques, consistent with their role in counteracting gravitational forces on the payload. The wrist joint displays notably broader distributions compared to the elbow joint across all payload conditions, likely due to its position further

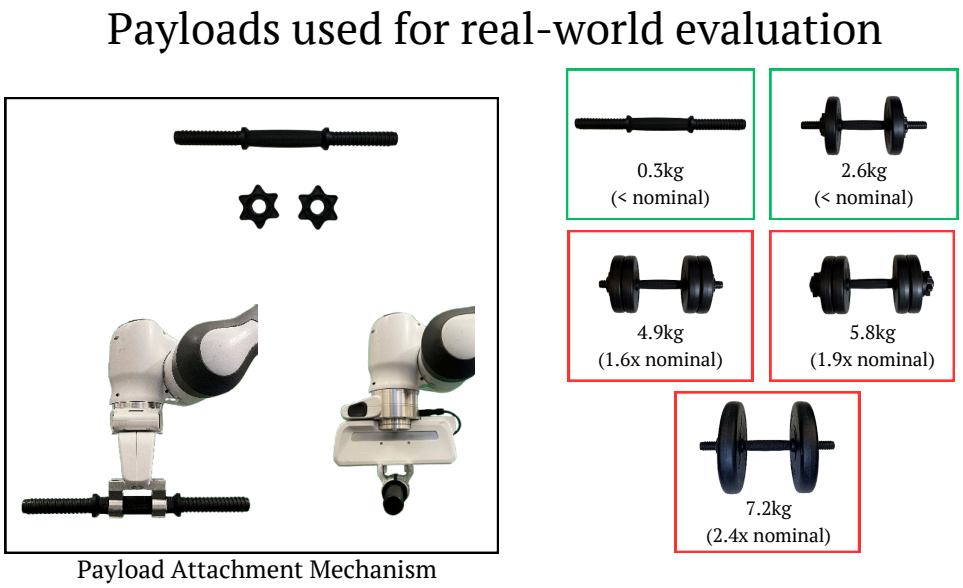

Figure 1: Red indicates super-nominal payloads. Green indicates payloads within the nominal capacity. See website for demos.

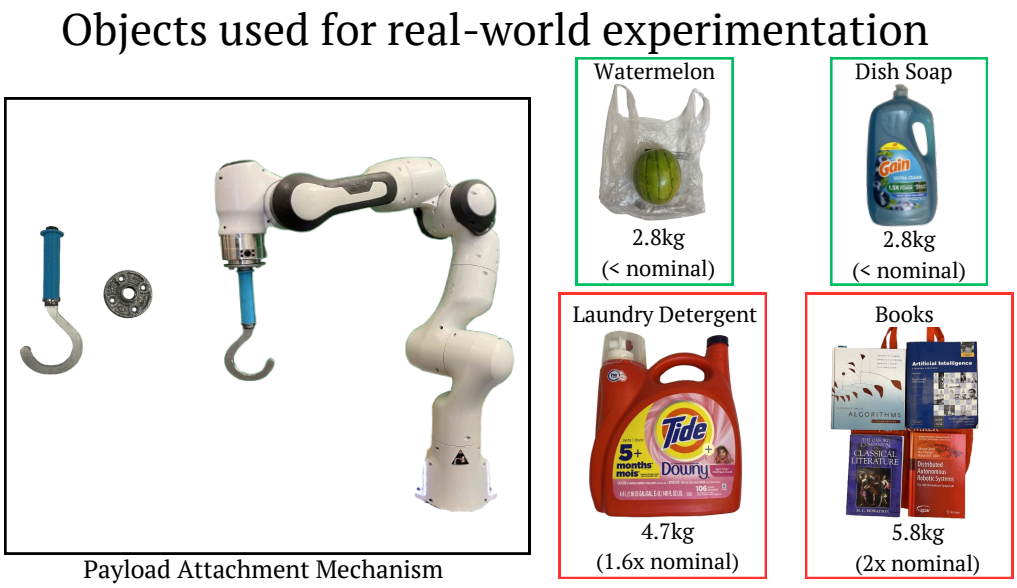

Figure 2: Red indicates super-nominal payloads. Green indicates payloads within the nominal capacity. See website for demos.

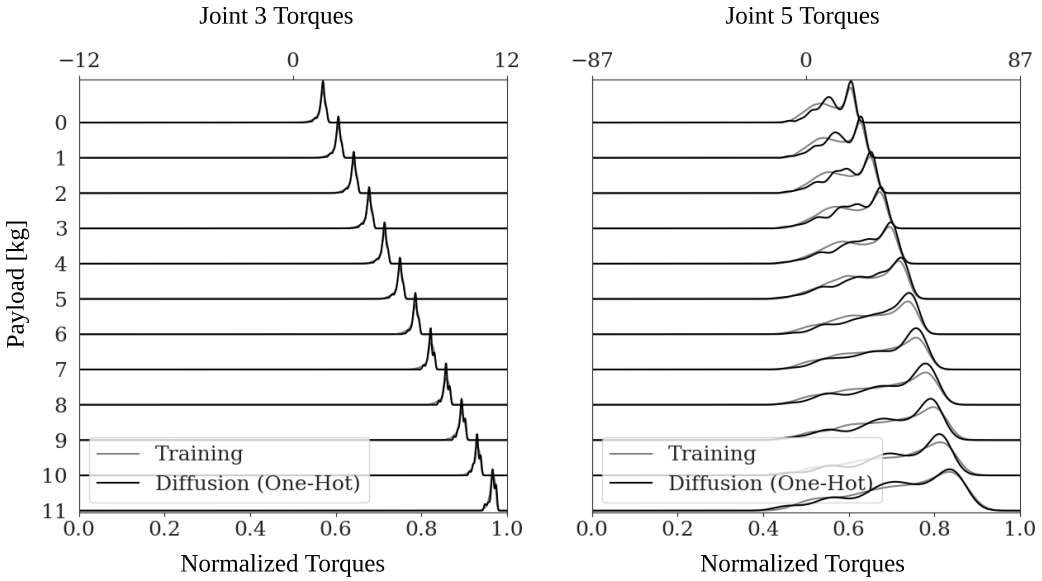

Figure 3: Distribution of joint torques generated by one-hot diffusion at varying payload masses, shown for the elbow (joint 3) and wrist (joint 5). The wrist joint exhibits broader torque distributions compared to the elbow joint, reflecting its position in the kinematic chain.

along the kinematic chain where it experiences the compound effects of proximal joint motions and must operate with smaller torque limits.

The diffusion model generates torque distributions that closely align with the training data distribution across all joints, including joint 3 and 5, and this consistency holds across different payload masses. This alignment suggests two key insights. First, the model has successfully learned to replicate the statistical properties of valid trajectories, capturing the underlying dynamics constraints and super-nominal operational behavior. Second, diffusion maintains stable performance under more constrained conditions as payload mass increases, especially with limited availability of valid training trajectories that satisfy torque constraints for heavier payloads.

Reachable Workspace and Joint Failures. The heatmaps in Figure 4 and Figure 6 visualize the robot's accessible workspace in the x-y plane at a fixed height of $z = 0.2m$, with each pixel representing a 1.73×1.73cm bin (area = 2.98 cm^2). The heatmap intensity indicates the success rate of generating dynamically-feasible trajectories for end-effector poses starting or ending at each spatial location. This fine-grained spatial resolution reveals how the robot's operational boundaries progressively contract under increasing payload conditions, quantifying the relationship between load capacity and accessible workspace. The reachable workspace area at nominal capacity (3kg) spans 1.578 m^2, reducing to 1.455 m^2 at $2\times$ nominal load (6kg)—a 92.2% retention. At $3\times$ nominal capacity (9kg), the robot maintains a reachable area of 1.066 m^2 or 67.6% of its nominal workspace. This demonstrates significant workspace preservation even under substantial payload increases.

The 3×3 joint failure maps (Figure 4), corresponding to subsections of the workspace reachability map, indicate which joints operate near their torque limits in these sections of the workspace. Joints 0, 4, and 6 operate well within their torque limits. In contrast, joints 1, 2, 3, and 5 frequently operate close to their torque limits, as their axes lie predominantly perpendicular to the gravity vector, requiring them to generate substantial counteracting torques for payload support. Joint 3 displays a characteristic failure pattern, suggesting limited maneuverability in the negative-x region of the task space. This pattern reveals an inherent asymmetry in the robot's mechanical design across the x-axis. Joint 5, despite its relatively limited torque capacity compared to the base-mounted joint 1, bears significant responsibility for maintaining payload position and orientation throughout

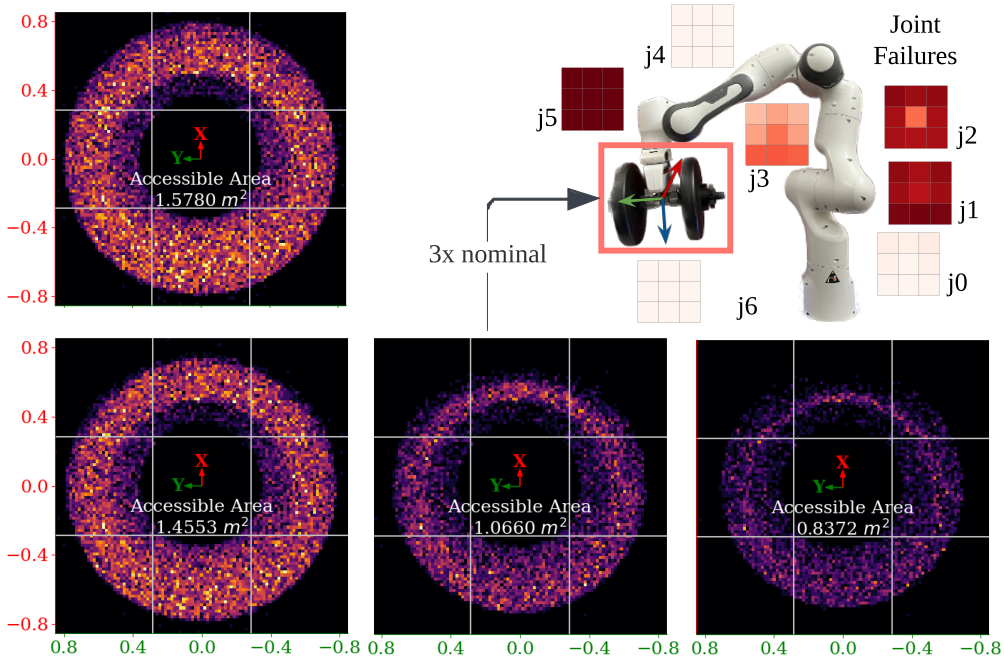

Figure 4: Analysis of diffusion success, workspace reachability, and joint constraints across a range of payloads. Heatmaps: success rate distribution across the robot's reachable workspace, where color intensity indicates the probability of successful trajectory completion for trajectories starting or ending at each location in the x-y plane ($z = 0.2$m). Each pixel in the heatmap represents a 1.73×1.73cm spatial bin, providing high-resolution insight into the robot's operational boundaries and joint-specific limitations. Top right: A jointwise 3×3 grid of spatial patches corresponding to the 9kg heatmap indicating which joints reach their torque limits in different regions of the workspace.

trajectories. This is particularly noteworthy given that joint 1 is primarily engineered to support the weight of the subsequent kinematic chain links and payload.

2.4 Real-World Trajectory Execution

We validate our approach on a Franka Emika Panda robot arm using trajectories generated by our diffusion model ($\Delta t = 0.15s$, equivalent to $6.67Hz$). The generated trajectories were executed through the robot's velocity control interface, with linear interpolation to match the required 1kHz control frequency.[1] Notably, the diffusion-generated trajectories required no additional post-processing or filtering beyond this basic interpolation. Our real-world testing encompassed payload masses ranging from 0 to 7.2kg—Figure 5 presents the joint torque measurements recorded across 15 trajectory executions at maximum payload for different start and goal joint configurations. While our simulation work demonstrates successful planning capability for payloads up to 11kg with an approximate success rate of 20% (Figure 4), we constrained our physical experiments to 7.2kg to maintain safe operating conditions.

The measured joint torques exhibit distinct patterns reflecting the robot's kinematic chain and load-bearing constraints. Joints 1-3 demonstrate substantial torque variations during dynamic motion phases, consistent with their primary role in payload manipulation. Joint 5 exhibits notably consistent torque readings (approximately 8Nm) across trials, primarily managing gravitational loads to maintain end-effector orientation. Joints 0 and 6, which mainly control vertical axis rotation, show minimal torque requirements throughout the trajectories. A characteristic feature emerges at

[1]The accompanying video material demonstrates the robot manipulating super-nominal payload masses and various real-world objects.

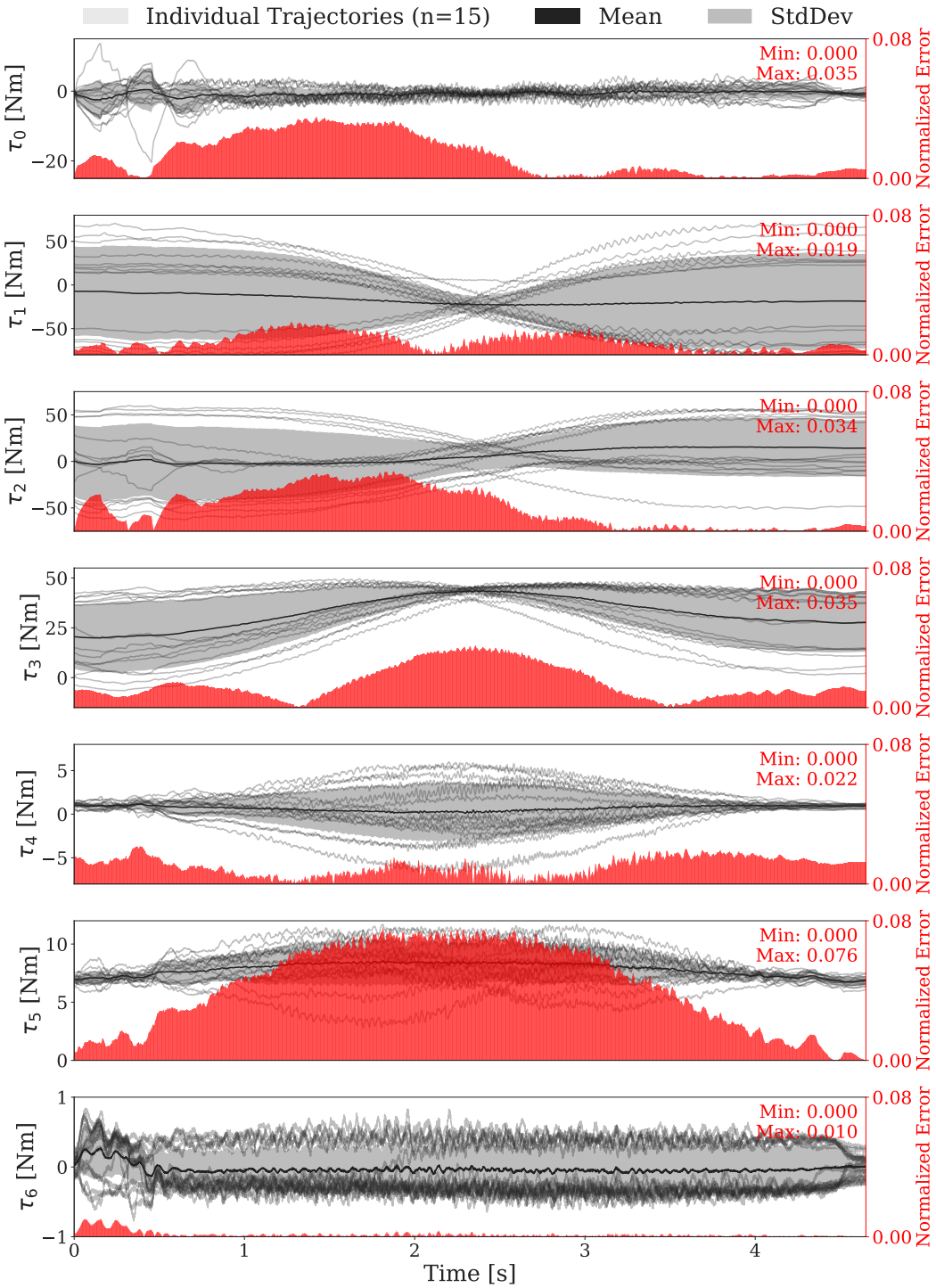

Figure 5: Joint torques and tracking errors during trajectory execution with a 7.2kg payload on the Franka Emika Panda robot. Grey shaded regions show mean $\pm$ one standard deviation of joint torques across 15 different trajectories, while red bars indicate the normalized tracking error for each joint. Note the consistent $\approx$ 8Nm loading on joint 5 and increased tracking errors in joints 1, 3, and 5 due to gravitational loading effects.

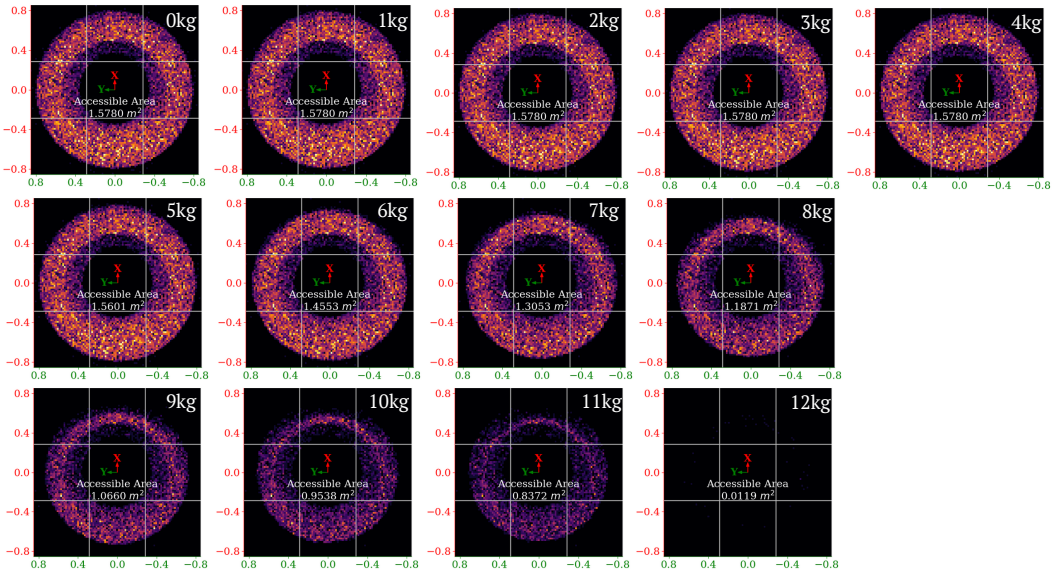

Figure 6: These heatmaps visualize the robot's reachable workspace resulting from diffusion trajectories generated under different payload attachments for pick-and-place to/from $z = 0.2$m. Even at $3\times$ nominal payload capacity, we see a 67.6% retention of accessible workspace.

the trajectory apex ($2 - 3$ seconds), where torque standard deviation significantly decreases across all joints. This reduction corresponds to a mechanically advantageous configuration where the robot momentarily pauses vertical motion, resulting in more consistent torque distributions across multiple executions.

The normalized tracking error (shown in red in Figure 5) exhibits the highest magnitudes in joints 1, 3, and 5—those perpendicular to the payload's z-axis. This increased tracking error can be attributed to: 1) heightened sensitivity to mass and center-of-mass estimation errors due to significant gravitational loading at these joints; 2) higher joint velocities leading to increased joint friction modeling uncertainties [3]; and, 3) joint compliance effects, where high torques cause elastic deformation in the transmission elements leading to position deviations beyond encoder measurements. Our physical hardware experiments demonstrate successful execution of diffusion-planned trajectories with payloads beyond the derived operational threshold, though safety considerations ultimately bound rigorous real-world testing.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[2] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[3] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. De Luca. Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4):4147–4154, 2019.