STAR-Diff: Surgical Trocar-Adaptive Diffusion Policy with RCM-aware Action Space

Anonymous Authors

***Abstract*— Automating laparoscopic surgery holds significant promise for improving surgical outcomes, yet existing learning-based approaches assume fixed trocar positions during both training and deployment—a simplification that rarely holds in clinical practice. We present STAR-Diff (Surgical Trocar-Adaptive, RCM-aware Diffusion Policy), a novel imitation learning framework that enables generalization across varying trocar placements without retraining. Our key insight is that the Remote Center of Motion (RCM) constraint, which restricts instrument motion to pass through the trocar point, defines a 4-dimensional submanifold of $\mathrm{SE}(3)$. By directly parameterizing this constraint manifold as the action space, we eliminate the need for post-hoc constraint enforcement. Furthermore, we observe that the absolute end-effector position in the camera frame is trocar-invariant for visual servoing tasks, motivating a hierarchical architecture that decouples trocar-invariant position prediction (via diffusion) from trocar-dependent axial rotation estimation (via deterministic mapping). Designed for standard 6-DoF manipulators without specialized surgical wrists, STAR-Diff democratizes surgical automation beyond proprietary platforms. We validate our approach on multiple surgical tasks, demonstrating substantial improvements in generalization to unseen trocar configurations while maintaining near-zero RCM constraint violations.**

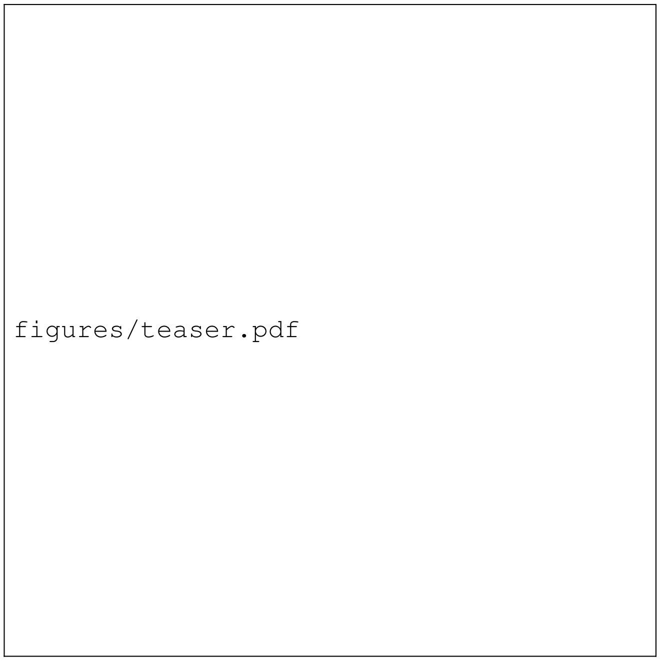

Fig. 1: **Problem Overview.** (a) Existing surgical policies are trained and deployed with a fixed trocar position, failing when trocar placement varies. (b) STAR-Diff learns a single policy that adapts to arbitrary trocar positions by leveraging an RCM-aware action space and hierarchical architecture that decouples trocar-invariant from trocar-dependent components.

I. Introduction

Minimally invasive surgery (MIS), particularly laparoscopic procedures, has revolutionized surgical practice by reducing patient trauma, hospital stays, and recovery times [?]. However, the inherent challenges of operating through small incisions—limited field of view, reduced haptic feedback, and constrained instrument motion—place significant cognitive and physical demands on surgeons. Robot-assisted surgical systems, exemplified by the da Vinci platform [?], have emerged as transformative tools that enhance precision and ergonomics. Yet, the potential for autonomous execution of surgical subtasks remains largely unrealized in clinical settings.

Recent advances in robot learning, particularly imitation learning and diffusion-based policies [?], [?], have demonstrated remarkable success in contact-rich manipulation tasks. Several works have applied these techniques to surgical automation [?], [?], achieving impressive results on benchmark tasks such as peg transfer, tissue manipulation, and suturing. However, a critical assumption pervades nearly all existing work: *the trocar position is fixed and known a priori during both training and deployment*.

This assumption fundamentally limits the practical applicability of learned surgical policies. In clinical practice, trocar placement varies substantially based on:

- **Patient anatomy**: Body habitus, organ size, and pathology location necessitate patient-specific port configurations [?].
- **Procedure type**: Different surgical procedures (cholecystectomy, appendectomy, hernia repair) require distinct trocar arrangements optimized for target anatomy access.
- **Surgeon preference**: Individual surgeons develop personalized ergonomic preferences for port placement [?].

When deployed in a trocar configuration different from training, existing policies face two failure modes: (1) **workspace mismatch**, where the learned end-effector trajectories become unreachable or kinematically infeasible, and (2) **RCM constraint violation**, where the instrument axis fails to pass through the trocar point, risking tissue damage at the incision site.

The Remote Center of Motion (RCM) constraint is the defining kinematic characteristic of laparoscopic surgery. It requires that the instrument's longitudinal axis always passes through the trocar point, effectively reducing the instrument's degrees of freedom from 6 to 4 (excluding gripper actuation). Prior work has addressed RCM compliance through various mechanisms: hardware-based solutions using specialized kinematic designs [**?**], software-based approaches using constrained optimization [**?**], and learning-based methods that incorporate RCM as a loss term [**?**]. However, these approaches either require specialized hardware, introduce computational overhead at inference time, or fail to guarantee strict constraint satisfaction.

In this work, we propose **STAR-Diff**, a diffusion-based imitation learning framework that achieves trocar-adaptive surgical manipulation through two key innovations:

(1) RCM-aware Action Space. Rather than learning in the ambient $\mathrm{SE}(3)$ space and projecting onto the RCM constraint manifold post-hoc, we directly parameterize the action space as the 4-dimensional RCM constraint manifold $\mathcal{M}_{\text{RCM}}$. This "constraint baking" approach guarantees RCM compliance by construction, eliminating the need for runtime constraint enforcement.

(2) Hierarchical Trocar-Adaptive Architecture. We make a crucial observation: for visual servoing tasks with a fixed camera, the desired end-effector position in world coordinates is *trocar-invariant*—it depends only on the target object's visual appearance, not on where the trocar happens to be placed. This insight motivates a hierarchical decomposition:

- A **diffusion policy** predicts the trocar-invariant absolute position $\mathbf{p}_{\text{abs}}$, conditioned only on visual observations.
- A **deterministic network** predicts the trocar-dependent relative rotation γ_{rel}, given the diffusion output and trocar parameters.

This decomposition enables the diffusion model to focus on the complex, multimodal distribution of task-relevant positions without being confounded by trocar variation, while the simpler deterministic mapping handles the geometric transformation induced by different trocar placements.

Our framework is designed for standard 6-DoF industrial manipulators (e.g., Fairino FR5) equipped with laparoscopic graspers, deliberately avoiding dependence on proprietary systems with specialized wrists (e.g., da Vinci EndoWrist). This design choice serves two purposes: (1) it *democratizes* surgical automation research by enabling experimentation with commodity hardware, and (2) it establishes *extensibility* to future systems where surgeons' manual laparoscopic data could be leveraged for policy learning.

We summarize our contributions as follows:

1) We formalize the **trocar-adaptive surgical manipulation** problem and identify the geometric structure of the RCM constraint manifold.
2) We propose an **RCM-aware action space** that parameterizes the constraint manifold directly, guaranteeing RCM compliance without post-hoc enforcement.
3) We introduce a **hierarchical diffusion architecture** that decouples trocar-invariant position prediction from trocar-dependent rotation estimation.
4) We demonstrate that STAR-Diff achieves strong generalization to unseen trocar configurations while maintaining near-zero RCM violations on multiple surgical tasks.

II. Related Work

A. Learning-based Surgical Automation

The application of machine learning to surgical robotics has grown substantially, spanning perception, planning, and control. Early work focused on learning from demonstrations (LfD) for isolated surgical subtasks [**?**], [**?**]. More recent approaches leverage deep learning for end-to-end visuomotor policies. Xu et al. [**?**] introduced SurRoL, a comprehensive simulation platform enabling reinforcement learning for surgical tasks. ORBIT-Surgical [**?**] extended this to GPU-accelerated simulation with photorealistic rendering.

Imitation learning has emerged as the dominant paradigm due to the difficulty of reward specification in surgery. Behavior cloning approaches have been applied to tissue manipulation [**?**], surgical cutting [**?**], and suturing [**?**]. The Action Chunking with Transformers (ACT) framework [**?**] demonstrated that predicting action sequences rather than single actions improves temporal consistency. Diffusion Policy [**?**] further advanced the state-of-the-art by leveraging denoising diffusion models to capture multimodal action distributions.

Despite these advances, **all existing methods assume fixed environmental configurations**, including trocar position. Our work addresses this limitation by explicitly conditioning on trocar parameters and designing an architecture that facilitates generalization.

B. Remote Center of Motion Constraints

The RCM constraint is fundamental to minimally invasive surgery, ensuring that instrument motion does not traumatize tissue at the insertion point. Solutions broadly fall into three categories:

Hardware-based RCM. Specialized kinematic designs mechanically enforce RCM compliance. The da Vinci system uses a parallelogram mechanism [**?**], while other designs employ spherical joints [**?**] or cable-driven architectures [**?**]. These solutions guarantee RCM by construction but require dedicated hardware.

Software-based RCM. For general-purpose manipulators, RCM can be enforced through control. Sandoval et al. [**?**] formulated RCM as a task-space constraint in a quadratic programming framework. Locke and Patel [**?**] derived analytical inverse kinematics for RCM-constrained systems. Control barrier functions (CBFs) have been applied to guarantee forward invariance of the RCM constraint set [**?**]. These approaches add computational overhead and may introduce tracking errors.

Learning-based RCM. Recent work incorporates RCM into learning objectives. Lu et al. [**?**] added RCM violation as a penalty term during policy optimization. Richter et al. [**?**] trained policies with RCM-constrained action spaces but did not address trocar variation. Our approach differs fundamentally: by parameterizing the RCM manifold directly as the action space, **constraint satisfaction is guaranteed**

by construction, not approximated through penalties or projections.

C. Generalization in Robot Learning

Achieving generalization across environmental variations is a central challenge in robot learning. Domain randomization [**?**] and domain adaptation [**?**] address visual domain shift. For spatial generalization, SE(3)-equivariant networks [**?**] exploit geometric symmetries. Contextual policies condition on task parameters [**?**], while meta-learning approaches [**?**] enable rapid adaptation.

In surgical robotics, generalization has received limited attention. Kazanzides et al. [**?**] discussed the need for adaptable surgical systems, but learning-based solutions remain scarce. The closest work is that of Shin et al. [**?**], who varied tissue properties but not kinematic configurations. **To our knowledge, STAR-Diff is the first learning-based approach to explicitly address generalization across trocar positions.**

D. Diffusion Models for Robot Control

Diffusion models [**?**], [**?**] have transformed generative modeling and recently entered robotics. Diffusion Policy [**?**] demonstrated that framing action prediction as conditional denoising yields policies that capture multimodal behavior and exhibit strong generalization. Follow-up work has applied diffusion to diverse domains: contact-rich manipulation [**?**], locomotion [**?**], and multi-task learning [**?**].

Architectural variants have emerged to address specific challenges. Chi et al. [**?**] introduced a universal manipulation interface using diffusion. 3D Diffusion Policy [**?**] incorporated point cloud observations for spatial reasoning. Consistency Policy [**?**] accelerated inference through distillation. Our work builds on this foundation, adapting the diffusion framework to the unique constraints of surgical manipulation through RCM-aware action spaces and hierarchical decomposition.

III. Problem Formulation

A. Notation and Coordinate Frames

We establish the following coordinate frames (Fig. **??**):

- $\{S\}$: Space (world) frame, fixed.
- $\{C\}$: Camera frame, fixed relative to $\{S\}$ with known transform ${}^{S}T_C \in \mathrm{SE}(3)$.
- $\{R\}$: Robot base frame, fixed relative to $\{S\}$ with known transform ${}^{S}T_R \in \mathrm{SE}(3)$.
- $\{T\}$: Trocar frame, centered at the trocar point $\mathbf{p}_T \in \mathbb{R}^3$ with z-axis aligned to gravity. The transform ${}^{S}T_T(\lambda_{\text{trocar}}) \in \mathrm{SE}(3)$ depends on trocar parameters.
- $\{E\}$: End-effector (TCP) frame, located at the jaw center of the laparoscopic grasper.

We assume a monocular camera observing the surgical workspace, with optical axis directed toward a reference point $\mathbf{o}_{\text{ref}}$ at distance L from the camera center.

B. Trocar Parameterization

The trocar position varies across procedures. We parameterize it using spherical coordinates centered at the reference point:

$$\lambda_{\text{trocar}} = (r, \theta, \phi) \in \mathbb{R}^{+} \times [0, \pi] \times [0, 2\pi), \tag{1}$$

where r is the radial distance from $\mathbf{o}_{\text{ref}}$, θ is the polar angle, and ϕ is the azimuthal angle. The trocar position in world coordinates is:

$$\mathbf{p}_T(\lambda_{\text{trocar}}) = \mathbf{o}_{\text{ref}} + r \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}. \tag{2}$$

This parameterization captures clinically relevant trocar variation while providing a compact, continuous representation suitable for conditioning neural networks.

C. The RCM Constraint Manifold

[Remote Center of Motion] A manipulator satisfies the RCM constraint at point $\mathbf{p}_T$ if the longitudinal axis of the end-effector always passes through $\mathbf{p}_T$. Formally, let $R_E \in \mathrm{SO}(3)$ and $\mathbf{p}_E \in \mathbb{R}^3$ denote the end-effector orientation and position. The RCM constraint requires:

$$\mathbf{p}_E = \mathbf{p}_T + d \cdot R_E \mathbf{e}_z, \quad d \in \mathbb{R}, \tag{3}$$

where $\mathbf{e}_z = [0, 0, 1]^\top$ is the instrument's longitudinal axis in the end-effector frame, and d is the insertion depth (signed).

[RCM Manifold Structure] The set of end-effector poses satisfying the RCM constraint forms a 4-dimensional submanifold of SE(3):

$$\mathcal{M}_{\text{RCM}}(\lambda_{\text{trocar}}) = \{(R_E, \mathbf{p}_E) \in \mathrm{SE}(3) \mid (\textbf{??}) \text{ holds}\}. \tag{4}$$

This manifold is diffeomorphic to $S^2 \times S^1 \times \mathbb{R}$, corresponding to:

- S^2: Instrument pointing direction (2 DoF)
- S^1: Axial rotation about the instrument axis (1 DoF)
- $\mathbb{R}$: Insertion depth (1 DoF)

Proof: The constraint (**??**) fixes the position $\mathbf{p}_E$ given R_E and d, reducing the 6 DoF of SE(3) by 2 (the position is constrained to lie on the line through $\mathbf{p}_T$ with direction $R_E\mathbf{e}_z$). The remaining DoF correspond to: choosing the direction $R_E\mathbf{e}_z \in S^2$ (2 DoF), choosing the rotation about this axis (1 DoF, giving S^1), and choosing the depth d (1 DoF, giving $\mathbb{R}$). ■

D. RCM-aware Action Space Parameterizations

Rather than learning in SE(3) and projecting onto $\mathcal{M}_{\text{RCM}}$, we seek explicit parameterizations of $\mathcal{M}_{\text{RCM}}$ as the action space. We identify four natural parameterizations:

$$\boxed{\begin{array}{ll} x_{\text{RCM}}^{(1)} = (d_{\text{rel}}, R_{\text{rel}}) & \text{[Relative depth, Relative rotation]} \\ x_{\text{RCM}}^{(2)} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{rel}}) & \text{[Relative roll, Relative position]} \\ x_{\text{RCM}}^{(3)} = (d_{\text{rel}}, R_{\text{abs}}) & \text{[Relative depth, Absolute rotation]} \\ x_{\text{RCM}}^{(4)} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}}) & \text{[Relative roll, Absolute position]} \end{array}} \tag{5}$$

TABLE I: Comparison of RCM Action Space Parameterizations

Property	$x_{\text{RCM}}^{(1)}$	$x_{\text{RCM}}^{(2)}$	$x_{\text{RCM}}^{(3)}$	$x_{\text{RCM}}^{(4)}$
Trocar-invariant component			R_{abs}	$\mathbf{p}_{\text{abs}}$
Geometric interpretability	High	Medium	Low	High
Learning efficiency	Low	Low	Low	**High**
Configuration consistency	–	–	Poor	**Good**

Here, "relative" quantities are expressed in the trocar frame $\{T\}$, while "absolute" quantities are expressed in the world frame $\{S\}$.

Analysis. Parameterizations $x_{\text{RCM}}^{(1)}$ and $x_{\text{RCM}}^{(2)}$ are fully trocar-relative: all components depend on λ_{trocar}. Consequently, a policy must learn to map trocar parameters to appropriate actions, potentially requiring extensive trocar coverage during training.

Parameterizations $x_{\text{RCM}}^{(3)}$ and $x_{\text{RCM}}^{(4)}$ are partially trocar-invariant: one component (rotation or position) is expressed absolutely. This asymmetry has profound implications:

[Trocar-Invariance of Absolute Position] For visual servoing tasks where the target is defined by visual features in a fixed camera frame, the desired absolute position $\mathbf{p}_{\text{abs}}^{*}$ is independent of trocar parameters λ_{trocar}.

Proof: Let the target object have world-frame position $\mathbf{p}_{\text{target}} \in \mathbb{R}^3$, determined solely by camera observations. The desired end-effector position for task completion is $\mathbf{p}_{\text{abs}}^{*} = f_{\text{task}}(\mathbf{p}_{\text{target}})$, where f_{task} encodes task-specific offsets (e.g., grasp approach). Since neither $\mathbf{p}_{\text{target}}$ nor f_{task} depends on λ_{trocar}, the result follows. ■

While $x_{\text{RCM}}^{(3)}$ also contains a trocar-invariant component (R_{abs}), it suffers from **configuration inconsistency**: the same R_{abs} corresponds to vastly different instrument configurations depending on λ_{trocar}. This makes learning difficult, as similar rotation targets yield dissimilar physical behaviors.

In contrast, $x_{\text{RCM}}^{(4)}$ exhibits **configuration consistency**: the same $\mathbf{p}_{\text{abs}}$ corresponds to similar instrument tip positions regardless of λ_{trocar}. The only variation is in the instrument's approach angle, which is captured by the trocar-dependent γ_{rel}.

Based on this analysis, we adopt $x_{\text{RCM}}^{(4)} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$ as our action space.

E. Problem Statement

We now state the trocar-adaptive surgical imitation learning problem:

Given:

- A dataset $\mathcal{D} = \{(\tau_i, \lambda_i)\}_{i=1}^{N}$ of expert demonstrations, where $\tau_i = \{(o_t, a_t, g_t)\}_{t=1}^{T_i}$ contains observations o_t (images), actions $a_t \in x_{\text{RCM}}^{(4)}$, and gripper states g_t, collected under trocar configuration λ_i.
- A distribution $p(\lambda_{\text{trocar}})$ over trocar configurations.

Find: A policy $\pi(a \mid o, \lambda_{\text{trocar}})$ that:

1) Maximizes task success rate across $p(\lambda_{\text{trocar}})$, including configurations not seen during training.
2) Guarantees RCM constraint satisfaction by construction.
3) Enables efficient inference for real-time control.

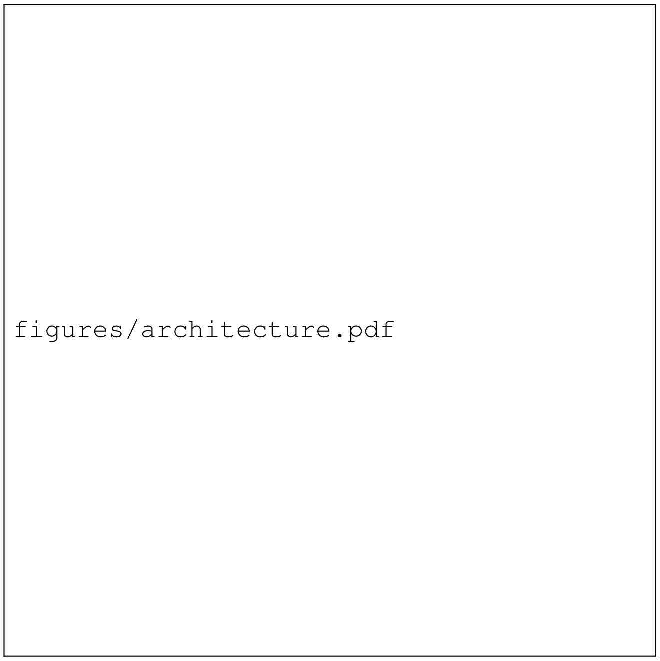

Fig. 2: **STAR-Diff Architecture.** The diffusion policy predicts trocar-invariant absolute positions from visual observations. The deterministic network then computes trocar-dependent relative rotations conditioned on the diffusion output and trocar parameters.

IV. Methodology

We present STAR-Diff, a hierarchical diffusion-based framework for trocar-adaptive surgical manipulation. Our approach exploits the decomposition of the RCM action space $x_{\text{RCM}}^{(4)} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$ into trocar-invariant (position) and trocar-dependent (rotation) components.

A. Architecture Overview

STAR-Diff comprises two modules (Fig. **??**):

1) **Absolute Position Diffusion Policy** π_{abs}: A conditional diffusion model that predicts future absolute positions $\mathbf{p}_{\text{abs}}^{t+1:t+H}$ and gripper actions $g^{t+1:t+H}$, conditioned on visual observations $I^{t-T_o:t}$ and position history $\mathbf{p}_{\text{abs}}^{t-T_o:t}$.
2) **Relative Rotation Network** f_{rel}: A deterministic MLP that predicts relative roll $\gamma_{\text{rel}}^{t+1:t+H}$, conditioned on the diffusion output, trocar parameters λ_{trocar}, and roll history $\gamma_{\text{rel}}^{t-T_o:t}$.

Here, T_o is the observation horizon and H is the prediction (action chunk) horizon.

1) Rationale for Hierarchical Decomposition: The decomposition is motivated by two observations:

Observation 1: Absolute position prediction benefits from generative modeling. End-effector trajectories in manipulation tasks often exhibit multimodality (e.g., multiple valid grasp approaches) and complex temporal correlations. Diffusion models excel at capturing such distributions [**?**].

Observation 2: Given $\mathbf{p}_{\text{abs}}$ and λ_{trocar}, the relative roll γ_{rel} is nearly deterministic. Once the tip position and trocar location are known, the instrument's approach direction is constrained to pass through both points. The only remaining freedom is axial rotation, which is typically dictated by task requirements (e.g., gripper orientation for grasping) rather than exhibiting multimodality.

B. Absolute Position Diffusion Policy

We adopt the DDPM framework [**?**] for position prediction. The diffusion model learns to reverse a forward noising process that gradually corrupts action sequences.

1) Forward Process: Given a clean action sequence $\mathbf{a}_0 = [\mathbf{p}_{\text{abs}}^{t+1:t+H}, g^{t+1:t+H}]$, the forward process produces increasingly noisy versions:

$$q(\mathbf{a}_k \mid \mathbf{a}_{k-1}) = \mathcal{N}(\mathbf{a}_k; \sqrt{1-\beta_k}\mathbf{a}_{k-1}, \beta_k \mathbf{I}), \tag{6}$$

where $\{\beta_k\}_{k=1}^{K}$ is the noise schedule with K diffusion steps. Using the reparameterization $\alpha_k = 1 - \beta_k$ and $\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i$, we can sample $\mathbf{a}_k$ directly:

$$\mathbf{a}_k = \sqrt{\bar{\alpha}_k}\mathbf{a}_0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{7}$$

2) Reverse Process: The reverse process learns to denoise, parameterized by a neural network ϵ_θ:

$$p_\theta(\mathbf{a}_{k-1} \mid \mathbf{a}_k, \mathbf{c}) = \mathcal{N}(\mathbf{a}_{k-1}; \boldsymbol{\mu}_\theta(\mathbf{a}_k, \mathbf{c}, k), \sigma_k^2 \mathbf{I}), \tag{8}$$

where $\mathbf{c} = (I^{t-T_o:t}, \mathbf{p}_{\text{abs}}^{t-T_o:t})$ is the conditioning information.

3) Training Objective: We train by minimizing the simplified denoising objective:

$$\mathcal{L}_{\text{abs}} = \mathbb{E}_{\mathbf{a}_0, k, \boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{a}_k, \mathbf{c}, k)\|^2 \right]. \tag{9}$$

4) Network Architecture: The noise prediction network ϵ_θ follows the conditional U-Net architecture of [**?**]:

Vision Encoder. Images $I^{t-T_o:t}$ are encoded using a ResNet-18 backbone, pretrained on ImageNet and fine-tuned during training. We apply spatial softmax pooling to obtain a fixed-dimensional feature vector per frame:

$$\mathbf{v}_t = \text{SpatialSoftmax}(\text{ResNet}(I_t)) \in \mathbb{R}^{d_v}. \tag{10}$$

State Encoder. Position history $\mathbf{p}_{\text{abs}}^{t-T_o:t}$ is encoded via a 1D convolutional network:

$$\mathbf{s} = \text{Conv1D}(\mathbf{p}_{\text{abs}}^{t-T_o:t}) \in \mathbb{R}^{d_s}. \tag{11}$$

Conditional U-Net. The concatenated condition $\mathbf{c} = [\mathbf{v}_{t-T_o:t}, \mathbf{s}]$ is injected into a 1D U-Net that operates on the temporal dimension of action sequences. Diffusion timestep k is encoded via sinusoidal embeddings and added to intermediate features.

5) Inference: At test time, we sample actions via iterative denoising:

$$\mathbf{a}_{k-1} = \frac{1}{\sqrt{\alpha_k}} \left(\mathbf{a}_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}} \epsilon_\theta(\mathbf{a}_k, \mathbf{c}, k) \right) + \sigma_k \mathbf{z}, \tag{12}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $k > 1$ and $\mathbf{z} = \mathbf{0}$ for $k = 1$. We use DDIM [**?**] for accelerated sampling with fewer steps.

Algorithm 1 RCM Action to End-Effector Pose

```
Require: Predicted action $(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$, trocar parameters $\lambda_{\text{trocar}}$
Ensure: End-effector pose $(R_E, \mathbf{p}_E)$
Compute trocar position: $\mathbf{p}_T \leftarrow$ Eq. (??)
Compute instrument direction: $\mathbf{v} \leftarrow \frac{\mathbf{p}_{\text{abs}} - \mathbf{p}_T}{\|\mathbf{p}_{\text{abs}} - \mathbf{p}_T\|}$
Compute base rotation aligning $\mathbf{e}_z$ to $\mathbf{v}$:
    $R_{\text{base}} \leftarrow \text{RotationAlign}(\mathbf{e}_z, \mathbf{v})$
Apply axial rotation:
    $R_E \leftarrow R_{\text{base}} \cdot R_z(\gamma_{\text{rel}})$
Set position: $\mathbf{p}_E \leftarrow \mathbf{p}_{\text{abs}}$
return $(R_E, \mathbf{p}_E)$
```

C. Relative Rotation Network

The relative roll network f_ϕ is a deterministic MLP that completes the action by predicting trocar-dependent components.

1) Input Representation: Given the diffusion output $\mathbf{p}_{\text{abs}}^{t+1:t+H}$, trocar parameters λ_{trocar}, and roll history $\gamma_{\text{rel}}^{t-T_o:t}$, we construct the input:

$$\mathbf{x}_{\text{rel}} = [\mathbf{p}_{\text{abs}}^{t+1:t+H}, \text{Embed}(\lambda_{\text{trocar}}), \gamma_{\text{rel}}^{t-T_o:t}], \tag{13}$$

where Embed$(\cdot)$ is a learnable embedding for trocar parameters.

2) Architecture: The network f_ϕ consists of:

- Linear projection: $\mathbf{x}_{\text{rel}} \mapsto \mathbb{R}^{256}$
- 3 residual MLP blocks with LayerNorm and GELU activation
- Output projection: $\mathbb{R}^{256} \mapsto \mathbb{R}^{H}$ (predicting $\gamma_{\text{rel}}^{t+1:t+H}$)

3) Training: The rotation network is trained jointly with the diffusion policy:

$$\mathcal{L}_{\text{rel}} = \mathbb{E}\left[\|\gamma_{\text{rel}}^{t+1:t+H} - f_\phi(\mathbf{p}_{\text{abs}}^{t+1:t+H}, \lambda_{\text{trocar}}, \gamma_{\text{rel}}^{t-T_o:t})\|^2 \right]. \tag{14}$$

The total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{abs}} + \lambda \mathcal{L}_{\text{rel}}, \tag{15}$$

where λ balances the two terms (we use $\lambda = 0.1$).

D. Action Execution via Forward Kinematics

Given the predicted RCM action $(\gamma_{\text{rel}}^{t+1}, \mathbf{p}_{\text{abs}}^{t+1})$, we recover the full end-effector pose for robot control:

The RotationAlign$(\mathbf{a}, \mathbf{b})$ function computes a rotation matrix that maps unit vector $\mathbf{a}$ to unit vector $\mathbf{b}$ using Rodrigues' formula:

$$R = I + [\mathbf{k}]_\times + [\mathbf{k}]_\times^2 \frac{1}{1 + \mathbf{a} \cdot \mathbf{b}}, \tag{16}$$

where $\mathbf{k} = \mathbf{a} \times \mathbf{b}$ and $[\cdot]_\times$ denotes the skew-symmetric matrix.

Finally, we compute joint commands via inverse kinematics:

$$\mathbf{q} = \text{IK}(R_E, \mathbf{p}_E), \tag{17}$$

using a Jacobian-based iterative solver with the current configuration as the initial guess.

E. RCM Constraint Guarantee

[RCM Compliance by Construction] Any action $(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}}) \in \mathbb{R} \times \mathbb{R}^3$ produced by STAR-Diff, when converted to an end-effector pose via Algorithm **??**, satisfies the RCM constraint (**??**).

Proof: By construction, $\mathbf{p}_E = \mathbf{p}_{\text{abs}}$ and the instrument direction is $\mathbf{v} = (\mathbf{p}_{\text{abs}} - \mathbf{p}_T)/\|\mathbf{p}_{\text{abs}} - \mathbf{p}_T\|$. Thus:

$$\mathbf{p}_E = \mathbf{p}_T + \|\mathbf{p}_{\text{abs}} - \mathbf{p}_T\| \cdot \mathbf{v} = \mathbf{p}_T + d \cdot R_E \mathbf{e}_z, \qquad (18)$$

where $d = \|\mathbf{p}_{\text{abs}} - \mathbf{p}_T\|$ and $R_E \mathbf{e}_z = \mathbf{v}$ by the alignment construction. This matches (**??**). ■

This guarantee is *unconditional*: regardless of the values predicted by the neural networks, the resulting pose always satisfies RCM. This contrasts with penalty-based or projection-based approaches where constraint satisfaction is approximate.

F. Training Data Collection

To enable trocar generalization, we collect demonstrations across a distribution of trocar configurations.

1) Trocar Sampling: We sample trocar parameters from a uniform distribution:

$$\lambda_{\text{trocar}} \sim \text{Uniform}([\underline{r}, \bar{r}] \times [\underline{\theta}, \bar{\theta}] \times [\underline{\phi}, \bar{\phi}]). \qquad (19)$$

Based on clinical guidelines for laparoscopic port placement [**?**], we set:

- $r \in [0, 5]$ cm (radial variation)
- $\theta \in [60, 120]$ (polar angle, avoiding extreme approach angles)
- $\phi \in [0, 360]$ (full azimuthal coverage)

2) Demonstration Protocol: For each sampled λ_{trocar}:

1) Configure the simulation/robot with the corresponding trocar position.
2) An expert teleoperator performs the task using a 6-DoF input device.
3) Record synchronized streams: images (30 Hz), end-effector pose, gripper state.
4) Convert poses to RCM action space representation $(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$.

We collect 20-30 demonstrations per trocar configuration across 10-15 configurations per task, yielding 200-450 demonstrations per task.

G. Implementation Details

Network Hyperparameters.

- Vision encoder: ResNet-18, output dimension $d_v = 512$
- State encoder: 3-layer Conv1D, output dimension $d_s = 256$
- U-Net: 4 downsampling/upsampling blocks, base channels 256
- Rotation MLP: 3 residual blocks, hidden dimension 256

Diffusion Hyperparameters.

- Diffusion steps: $K = 100$ (training), 10 (DDIM inference)
- Noise schedule: Cosine [**?**]
- Observation horizon: $T_o = 2$
- Prediction horizon: $H = 16$
- Action execution: First 8 actions executed before replanning

Training.

- Optimizer: AdamW, learning rate 1×10^{-4}, weight decay 1×10^{-6}
- Batch size: 256
- Training iterations: 100K
- Data augmentation: Random crop, color jitter, Gaussian noise

V. Experiments

We design a unified experiment to answer three research questions through systematic comparison:

- **RQ1**: Does STAR-Diff generalize to unseen trocar positions?
- **RQ2**: How effective is the RCM-aware action space compared to naive action spaces?
- **RQ3**: Does the hierarchical architecture provide benefits over a single-model approach?

By comparing all methods on the same task with the unified metric of **task success rate** across seen and unseen trocar configurations, we can simultaneously address all research questions.

A. Experimental Setup

1) Task: Peg Transfer: We evaluate on the **Peg Transfer** task from the Fundamentals of Laparoscopic Surgery (FLS) benchmark [**?**]. This task requires picking up a peg from one board and transferring it to a target location on another board. It tests fundamental skills including depth perception, hand-eye coordination, and precise manipulation under RCM constraints.

The task is implemented in a simulation environment based on SurRoL [**?**] with realistic physics and visual rendering. The robot is a 6-DoF Fairino FR5 manipulator equipped with a laparoscopic grasper.

2) Trocar Configuration: We define a trocar distribution covering clinically relevant variations:

- Radial distance: $r \in [0, 5]$ cm
- Polar angle: $\theta \in [70, 110]$
- Azimuthal angle: $\phi \in [0, 360]$

We discretize this space into 12 trocar configurations:

- **Training set**: 8 configurations (67%), uniformly distributed
- **Test set (Seen)**: Same 8 configurations as training
- **Test set (Unseen)**: 4 held-out configurations (33%), including boundary cases

3) Data Collection: Expert demonstrations are collected via teleoperation using a SpaceMouse 6-DoF input device. For each trocar configuration in the training set, we collect 30 successful demonstrations, yielding 240 total demonstrations recorded at 30 Hz (RGB images at 256×256, end-effector poses, gripper states).

TABLE II: Summary of Compared Methods

Group	Method	Action Space	RCM	Arch.
Naive	DP-EE	SE(3)	None	Single
	DP-EE-Proj	SE(3)	Post-hoc	Single
RCM Action Spaces	DP-RCM-1	$(d_{\text{rel}}, R_{\text{rel}})$	By constr.	Single
	DP-RCM-2	$(\gamma_{\text{rel}}, \mathbf{p}_{\text{rel}})$	By constr.	Single
	DP-RCM-3	$(d_{\text{rel}}, R_{\text{abs}})$	By constr.	Single
	DP-RCM-4	$(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$	By constr.	Single
Ours	STAR-Diff-Single	$(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$	By constr.	Single
	STAR-Diff	$(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$	By constr.	**Hier.**

All methods use Diffusion Policy as the base architecture. “By constr.” = RCM satisfied by construction. “Hier.” = Hierarchical (diffusion + MLP).

4) Evaluation Protocol: All methods are evaluated using **Task Success Rate (SR)**: the percentage of trials where the peg is successfully transferred within 60 seconds. Each method is evaluated over 30 trials per trocar configuration (240 trials for seen, 120 trials for unseen).

We define the **generalization gap** as:

$$\Delta_{\text{gen}} = \text{SR}_{\text{seen}} - \text{SR}_{\text{unseen}}. \tag{20}$$

A smaller gap indicates better generalization to novel trocar configurations.

B. Methods Compared

To systematically isolate the effects of action space design and architecture, we compare 8 methods organized into three groups (Table **??**).

Group 1: Naive Action Spaces.

- **DP-EE**: Vanilla Diffusion Policy predicting end-effector pose in SE(3). No RCM handling.
- **DP-EE-Proj**: Same as DP-EE, but with post-hoc projection onto the RCM constraint manifold at execution time.

Group 2: RCM Action Spaces (Single Model). Four variants corresponding to the parameterizations in Eq. (**??**):

- **DP-RCM-1**: $x^{(1)}_{\text{RCM}} = (d_{\text{rel}}, R_{\text{rel}})$ — fully relative, depth + rotation.
- **DP-RCM-2**: $x^{(2)}_{\text{RCM}} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{rel}})$ — fully relative, roll + position.
- **DP-RCM-3**: $x^{(3)}_{\text{RCM}} = (d_{\text{rel}}, R_{\text{abs}})$ — mixed, depth relative + rotation absolute.
- **DP-RCM-4**: $x^{(4)}_{\text{RCM}} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$ — mixed, roll relative + position absolute.

All use a single diffusion model conditioned on trocar parameters λ_{trocar}.

Group 3: Our Methods.

- **STAR-Diff-Single**: Uses $x^{(4)}_{\text{RCM}}$ with a single diffusion model (ablation for RQ3).
- **STAR-Diff**: Uses $x^{(4)}_{\text{RCM}}$ with hierarchical architecture—diffusion for $\mathbf{p}_{\text{abs}}$, deterministic MLP for γ_{rel}.

Note that **STAR-Diff-Single** and **DP-RCM-4** are identical, listed in both groups to clarify comparisons.

C. Results

Table **??** presents the unified comparison across all methods.

TABLE III: Task Success Rate (%) on Peg Transfer

Group	Method	Seen↑	Unseen↑	Δ_{gen}↓
Naive	DP-EE	[XX.X]	[XX.X]	[XX.X]
	DP-EE-Proj	[XX.X]	[XX.X]	[XX.X]
RCM Actions	DP-RCM-1 $(d_{\text{rel}}, R_{\text{rel}})$	[XX.X]	[XX.X]	[XX.X]
	DP-RCM-2 $(\gamma_{\text{rel}}, \mathbf{p}_{\text{rel}})$	[XX.X]	[XX.X]	[XX.X]
	DP-RCM-3 $(d_{\text{rel}}, R_{\text{abs}})$	[XX.X]	[XX.X]	[XX.X]
	DP-RCM-4 $(\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$	[XX.X]	[XX.X]	[XX.X]
Ours	STAR-Diff-Single	[XX.X]	[XX.X]	[XX.X]
	STAR-Diff	[**XX.X**]	[**XX.X**]	[**XX.X**]

Results averaged over 30 trials per trocar (240 seen, 120 unseen). Best in **bold**.

1) Analysis for RQ1: Generalization to Unseen Trocars: Comparing the “Unseen” column and Δ_{gen} across all methods reveals generalization capability.

Expected Findings:

- **DP-EE** should show the largest generalization gap, as the learned trajectories are tied to specific trocar configurations.
- **RCM action space methods** (DP-RCM-1 to 4) should show varying generalization depending on their trocar-invariant properties.
- **STAR-Diff** should achieve the smallest Δ_{gen} due to its trocar-invariant absolute position prediction.

2) Analysis for RQ2: RCM-aware vs. Naive Action Space: Comparing Group 1 (Naive) vs. Groups 2-3 (RCM-aware) isolates the effect of action space design.

Expected Findings:

- **DP-EE** should achieve reasonable success on seen trocars but with implicit RCM violations that may cause failures.
- **DP-EE-Proj** should improve over DP-EE by enforcing RCM, but post-hoc projection may introduce discontinuities that hurt performance.
- **All RCM action space methods** guarantee zero RCM violations by construction, leading to more reliable execution.
- Among RCM parameterizations, $x^{(4)}_{\text{RCM}} = (\gamma_{\text{rel}}, \mathbf{p}_{\text{abs}})$ should perform best due to configuration consistency (Section **??**).

3) Analysis for RQ3: Hierarchical vs. Single Architecture: Comparing **STAR-Diff-Single** (= DP-RCM-4) vs. **STAR-Diff** isolates the benefit of hierarchical decomposition.

Expected Findings:

- Both methods use the same action space $x^{(4)}_{\text{RCM}}$, so differences arise purely from architecture.
- **STAR-Diff-Single** must learn to predict both $\mathbf{p}_{\text{abs}}$ and γ_{rel} jointly, with γ_{rel} depending on trocar parameters.
- **STAR-Diff** decouples: the diffusion model predicts only trocar-invariant $\mathbf{p}_{\text{abs}}$, while a simple MLP handles γ_{rel}.
- The hierarchical design should yield better generalization (smaller Δ_{gen}) by allowing the diffusion model to focus on task-relevant multimodality without trocar-induced variation.

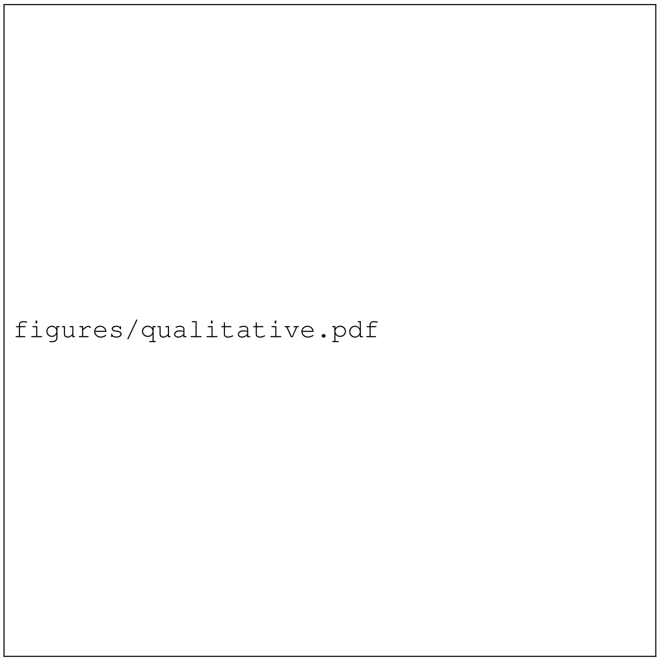

Fig. 3: **Qualitative Comparison.** Rollout trajectories across three trocar configurations (columns) for selected methods. Trocar points marked with ×. Top row: seen trocars. Bottom row: unseen trocars. [PLACEHOLDER]

TABLE IV: Inference Time Comparison

Method	Time (ms)	Freq. (Hz)
DP-EE / DP-RCM-* / Single	[XX]	[XX]
STAR-Diff	[XX]	[XX]

Measured on RTX 3090 with DDIM (10 steps).

D. Qualitative Analysis

Figure **??** visualizes representative rollout trajectories.

Expected Observations:

- **DP-EE**: Trajectories may deviate from RCM constraint, especially for unseen trocars.
- **DP-RCM-1/2** (fully relative): Valid trajectories but may struggle to adapt to unseen trocars.
- **DP-RCM-3**: Poor performance due to configuration inconsistency of R_{abs}.
- **STAR-Diff**: Smooth, RCM-compliant trajectories that adapt naturally across trocar positions.

E. Additional Analysis

1) Performance vs. Trocar Distance: Figure **??** analyzes how success rate degrades as test trocars deviate from the training distribution.

2) Computational Overhead: **Expected:** The additional MLP in STAR-Diff introduces <1ms overhead, maintaining real-time control.

VI. Discussion and Conclusion

A. Discussion

Limitations. Our current work has several limitations that suggest directions for future research:

Fig. 4: **Success Rate vs. Trocar Distance from Training Set.** Methods with trocar-invariant representations should degrade more gracefully. [PLACEHOLDER]

(1) 6-DoF Manipulator Constraint. STAR-Diff is designed for standard 6-DoF manipulators without articulated wrists. While this democratizes surgical automation research, it limits the dexterity compared to systems like da Vinci with EndoWrist. Extending our framework to 7+ DoF systems with redundant kinematics is an important direction.

(2) Single Fixed Camera. We assume a fixed monocular camera observing the workspace. Clinical laparoscopy typically uses an endoscopic camera inserted through a separate trocar, which moves during the procedure. Incorporating dynamic camera poses into our framework would enhance clinical applicability.

(3) Single-Arm Evaluation. Our experiments focus on single-arm peg transfer. Extending to bimanual tasks (e.g., suturing) where both arms must coordinate while respecting individual RCM constraints presents additional challenges.

(4) Simulation-Only Evaluation. While our simulation environment provides realistic physics and visuals, validation on physical hardware is necessary to assess sim-to-real transfer and real-world robustness.

Broader Impact. By enabling surgical policies that generalize across trocar configurations, STAR-Diff takes a step toward clinically deployable autonomous surgical systems. The use of commodity manipulators rather than proprietary platforms could reduce barriers to surgical robotics research and potentially lower costs of surgical automation. However, autonomous surgery raises significant safety and ethical considerations that must be carefully addressed before clinical deployment.

B. Conclusion

We presented STAR-Diff, a diffusion-based imitation learning framework for trocar-adaptive surgical manipulation. Our key contributions are:

1) **RCM-aware Action Space**: By directly parameterizing the RCM constraint manifold, we guarantee constraint satisfaction by construction, eliminating the need for post-hoc enforcement mechanisms.
2) **Hierarchical Architecture**: By decomposing the action space into trocar-invariant (absolute position) and trocar-dependent (relative rotation) components, we enable efficient learning that generalizes across trocar configurations.
3) **Theoretical Foundation**: We provided geometric analysis of the RCM constraint structure and formal guarantees of constraint satisfaction.

`[To be updated after experiments`: Our experiments on the peg transfer task demonstrate that STAR-Diff achieves [XX%] success rate on unseen trocar configurations, compared to [XX%] for the best baseline, while maintaining zero RCM violations.`]`

Looking forward, we envision extending STAR-Diff to more complex surgical tasks, bimanual manipulation, and real-world deployment. The principles of constraint-aware action space design and trocar-invariant decomposition may also apply to other constrained manipulation domains beyond surgery.

References

[1] M. J. Mack, "Minimally invasive and robotic surgery," *JAMA*, vol. 285, no. 5, pp. 568–572, 2001.

[2] Intuitive Surgical, "da Vinci Surgical System," 2023. [Online]. Available: `https://www.intuitive.com/`

[3] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proc. RSS*, 2023.

[4] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proc. RSS*, 2023.

[5] S. Scheikl *et al.*, "Movement primitive diffusion for dexterous surgical manipulation," in *Proc. ICRA*, 2024.

[6] J. Xu *et al.*, "SurRoL: An open-source reinforcement learning centered and dVRK compatible platform for surgical robot learning," in *Proc. IROS*, 2021.

[7] W. Reynolds, "The first laparoscopic cholecystectomy," *JSLS*, vol. 5, no. 1, pp. 89–94, 2001.

[8] D. Murphy *et al.*, "Surgeon preferences and peritoneal access," *Surg. Endosc.*, vol. 15, pp. 1236–1240, 2001.

[9] R. H. Taylor *et al.*, "A steady-hand robotic system for microsurgical augmentation," *Int. J. Robot. Res.*, vol. 18, no. 12, pp. 1201–1210, 1999.

[10] J. Sandoval *et al.*, "Collaborative framework for robot-assisted minimally invasive surgery," in *Proc. ICRA*, 2017.

[11] J. Lu *et al.*, "Super: A surgical perception framework for endoscopic tissue manipulation," *IEEE RA-L*, vol. 6, no. 2, pp. 3977–3984, 2021.

[12] Y. Yu *et al.*, "ORBIT-Surgical: An open-simulation framework for learning surgical augmented dexterity," in *Proc. ICRA*, 2024.

[13] J. van den Berg *et al.*, "Superhuman performance of surgical tasks by robots using iterative learning," in *Proc. ICRA*, 2010.

[14] J. Schulman *et al.*, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *Proc. IROS*, 2013.

[15] B. Thananjeyan *et al.*, "Multilateral surgical pattern cutting in 2D orthotropic gauze with deep reinforcement learning policies for tensioning," in *Proc. ICRA*, 2017.

[16] V. M. Varier *et al.*, "Collaborative suturing: A reinforcement learning approach," in *Proc. CASE*, 2020.

[17] G. S. Guthart and J. K. Salisbury, "The Intuitive telesurgery system: Overview and application," in *Proc. ICRA*, 2000.

[18] N. Simaan *et al.*, "Design and integration of a telerobotic system for minimally invasive surgery of the throat," *Int. J. Robot. Res.*, vol. 28, no. 9, pp. 1134–1153, 2009.

[19] R. C. Locke and R. V. Patel, "Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery," in *Proc. ICRA*, 2007.

[20] M. Selvaggio *et al.*, "Safe and efficient autonomous navigation for teleoperation in surgery," *IEEE RA-L*, vol. 3, no. 4, pp. 3373–3380, 2018.

[21] F. Richter *et al.*, "Open-sourced reinforcement learning environments for surgical robotics," in *Proc. IROS*, 2021.

[22] J. Tobin *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IROS*, 2017.

[23] E. Tzeng *et al.*, "Adapting deep visuomotor representations with weak pairwise constraints," in *Proc. WAFR*, 2020.

[24] A. Simeonov *et al.*, "Neural descriptor fields: SE(3)-equivariant object representations for manipulation," in *Proc. ICRA*, 2022.

[25] B. Da Silva *et al.*, "Learning parameterized skills," in *Proc. ICML*, 2012.

[26] C. Finn *et al.*, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017.

[27] P. Kazanzides *et al.*, "An open-source research kit for the da Vinci surgical system," in *Proc. ICRA*, 2014.

[28] C. Shin *et al.*, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *Proc. ICRA*, 2019.

[29] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020.

[30] Y. Song *et al.*, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, 2021.

[31] Y. Ze *et al.*, "3D diffusion policy," in *Proc. RSS*, 2024.

[32] A. Ajay *et al.*, "Is conditional generative modeling all you need for decision-making?" in *Proc. ICLR*, 2023.

[33] M. Reuss *et al.*, "Goal-conditioned imitation learning using score-based diffusion policies," in *Proc. RSS*, 2023.

[34] C. Chi *et al.*, "Universal manipulation interface," in *Proc. RSS*, 2024.

[35] V. Prasad *et al.*, "Consistency policy: Accelerated visuomotor policies via consistency distillation," in *Proc. RSS*, 2024.

[36] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. ICLR*, 2021.

[37] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. ICML*, 2021.

[38] J. H. Peters *et al.*, "Development and validation of a comprehensive program of education and assessment of the basic fundamentals of laparoscopic surgery," *Surgery*, vol. 135, no. 1, pp. 21–27, 2004.