Learning Tactile Manipulation Policies from Human Demonstrations

Erlernen taktiler Manipulationsstrategien anhand menschlisher Demonstrationen
Master thesis in the department of Computer Science by Changqi Chen
Date of submission: September 20, 2024

1. Review: M.Sc. Niklas Funk
2. Review: M.Sc. Tim Schneider
3. Review: Prof. Dr. Jan Peters
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

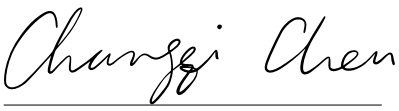
Hiermit erkläre ich, Changqi Chen, dass ich die vorliegende Arbeit gemäß § 22 Abs. 7 APB der TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 20. September 2024



C. Chen

Contents

Abstract

The sense of touch is regarded as one of the most crucial sensory modalities in humans, enabling the dexterous manipulation of tools in daily activities. However, the integration of tactile sensing and its impact on robotic manipulation is still underexplored. In this thesis, we investigate the importance of tactile sensing in robotic manipulation via solving a challenging task, i.e. match lighting, which is considered to be a typical task demonstrating that humans rely heavily on the sense of touch. As the whole task is impractical to simulate, we propose a novel imitation learning framework based on the multimodal visuomotor diffusion policy to learn match-lighting skills directly from human demonstrations. The performance of our framework for solving the task is extensively evaluated through real-world experiments, with respect to two main aspects: different sensor modality combinations and a wide range of match pose configurations. The evaluation results reveal that our policy is capable of solving the task robustly under different task configurations, and we also find that different sensor combinations can substantially affect policy performance. We believe that our results can highlight the fact that tactile sensing plays a significant role in improving policy performance for contact-rich manipulations, and provide practical experiences for choosing appropriate sensor combinations in similar tasks.

Zusammenfassung

Der Tastsinn gilt als eine der wichtigsten Sinnesmodalitäten des Menschen und ermöglicht die geschickte Handhabung von Werkzeugen bei täglichen Aktivitäten. Die Integration des Tastsinns und seine Auswirkungen auf die Robotermanipulation sind jedoch noch wenig erforscht. In dieser Arbeit untersuchen wir die Bedeutung des Tastsinns bei der Robotermanipulation anhand einer anspruchsvollen Aufgabe, dem Anzünden von Streichhölzern, einer typischen Aufgabe, die zeigt, dass der Mensch stark auf seinen Tastsinn angewiesen ist. Da die gesamte Aufgabe nicht simuliert werden kann, schlagen wir ein neuartiges Nachahmungs-Lernsystem vor, das auf der multimodalen visuomotorischen Diffusionspolitik basiert, um das Anzünden von Streichhölzern direkt von menschlichen Demonstrationen zu lernen. Die Leistung unseres Frameworks zur Lösung der Aufgabe wird in realen Experimenten umfassend evaluiert, und zwar in Bezug auf zwei Hauptaspekte: verschiedene Sensormodalitätskombinationen und eine breite Palette von Match-Positionskonfigurationen. Die Evaluierungsergebnisse zeigen, dass unsere Strategie in der Lage ist, die Aufgabe unter verschiedenen Aufgabenkonfigurationen robust zu lösen, und wir finden auch, dass verschiedene Sensorkombinationen die Leistung der Strategie erheblich beeinflussen können. Wir glauben, dass unsere Ergebnisse die Tatsache hervorheben können, dass die taktile Wahrnehmung eine bedeutende Rolle bei der Verbesserung der Leistung von Richtlinien für kontaktreiche Manipulationen spielt, und dass sie praktische Erfahrungen für die Auswahl geeigneter Sensorkombinationen bei ähnlichen Aufgaben liefern.

1 Introduction

Tactile perception is considered one of the fundamental human perceptual systems, allowing us to gather important information about the environment through touch, including pressure, texture, temperature, etc. In addition, tactile sensing also provides us with the ability to recognize and discriminate materials and positional features of the objects in our hands, i.e. the size and pose of the object, complementing other senses such as vision and hearing to guide us to perform all kinds of daily tasks e.g. opening drawers, turning keys, and folding clothes. However, in the field of robotics, despite the huge potential of tactile sensing in the manipulation of real-world objects, it has received less research attention compared to other sensory modalities such as vision [1]. Moreover, from the existing works, we notice that many tactile-involved manipulations are based on relatively slow and precise movements, e.g. pick-and-place, insertion, assembly, etc. [2]. The success of these tasks often primarily relies on the tactile sensor's ability to detect stable contact, identify object properties, or ensure precise alignment via haptic feedback. Nonetheless, tactile sensing in dynamic and fast-paced manipulation tasks, such as throwing and catching [3], and slip detection [4, 5], where the awareness of dynamic object displacements and rapid force variation are crucial, remains relatively underexplored.

To extend the research on these aspects, this thesis aims to investigate the importance of tactile sensing in dynamic manipulation tasks. Specifically, we focus on solving the match lighting task with the integration of tactile sensors. This task is considered to be one of the typical tasks in which humans rely heavily on the sense of touch [6]. Additionally, we investigate the effectiveness of solving the match lighting task with different sensor combinations.

In recent years, rapid advances in tactile sensory technology have improved the compactness, resolution, and robustness of tactile sensors available for robotic applications. Traditional tactile sensors, e.g. capacitive and piezoelectric types, offer less detailed measurements compared to modern optical tactile sensors like the popular GelSight Mini [7], which uses an RGB camera to capture high-resolution images of tactile deformations.

However, GelSight Mini is considered to lack temporal resolution due to the low updating rate of the RGB cameras (25 Hz). To address this drawback, a novel event-based optical tactile sensor, i.e. Evetac [5] was developed, employing a neuromorphic camera that captures brightness change on individual pixels rather than a whole frame of tactile measurement, enabling it to capture rapid deformations at 498 Hz—nearly 20 times faster than GelSight Mini, though with lower image spatial resolution. In this work, we employ both of the mentioned tactile sensors, i.e. GelSight Mini and Evetac, to provide tactile information during the task learning and execution, such as match in-hand pose and match displacement during the strike.

Considering the match-igniting task is generally unrealistic to simulate, reinforcement learning approaches are thus not ideal, as they require extensive trial-and-error interactions with the environment, which are not practical in our real-world scenarios. Therefore, we propose to employ a state-of-art imitation learning approach, i.e. Diffusion Policy [8], to learn the match-lighting skills directly from human demonstrations. Based on the Deep Denoising Probabilistic Models (DDPM), Diffusion Policy learns the gradient field of the action-distribution score function such that during the inference the expected action sequence can be sampled through a series of stochastic Langevin dynamics steps w.r.t. the learned gradient field [9]. Besides, the denoising process is conditioned on the multimodal observation sequence, improving the accuracy and consistency of the generated actions.

Since dynamic environment interactions are involved in our task, it is also necessary to apply constraints on the contact force during the task execution to prevent the breakage of the match. We thus involve a Cartesian impedance controller with empirical stiffness and damping values to regulate the robot's behavior more compliantly and gently during the interaction while preserving sufficient tracking accuracy.

Eventually, our imitation learning framework is comprehensively evaluated with multiple sensor combinations across various match pose configurations. Results indicate that our framework is capable of solving the match-lighting task and is robust against certain external perturbations. Furthermore, significant variations in policy performance with different sensor combinations are observed.

The thesis structure is organized as follows: Chapter 2 introduces previous works on learning tactile manipulation policies. Chapter 3 provides preliminary foundations of our proposed imitation learning framework. Subsequently, chapter 4 introduces the key components of our framework comprehensively. Chapter 5 reports the framework configurations and details of evaluation experiments. Chapter 6 presents the results of the evaluation experiments and corresponding discussions. Finally, in chapter 7 we summarize

our conclusions and propose future work against the limitation of our framework in chapter 8.

2 Related Work

In this chapter, we first motivate the popularity of optical tactile sensors in robotic applications. Subsequently, key approaches for learning tactile manipulation policies will be introduced. Lastly, we review recent advancements in addressing dynamic manipulation tasks using tactile sensing.

2.1 Optical Tactile Sensors

The role of tactile sensing is often considered an intermediate link between the exterior environment and the manipulator. Although vision systems can provide the robot with rich global information about the environment, extracting local features of the physical interaction between the object and the environment is often challenging due to visual occlusion [10]. Tactile sensors provide the robot with informative contact features and can potentially improve the manipulation performance.

Optical tactile sensors have long been developed and extensively studied to replicate the human sense of touch, which is a key sensory function in everyday life. Modern optical tactile sensors typically use cameras or other optical components to capture detailed tactile deformations on an elastic membrane placed above the camera in specific light conditions, enabling precise detection of texture, shape, and force. These features are quite beneficial for robotic applications, especially for object feature exploration [11, 12], dexterous manipulation [1, 3], human-machine interactions (HMIs) [13, 14], etc.

Nowadays, one of the most popular optical tactile sensors is GelSight Mini [7, 15], which can provide high spatial resolution RGB images that contain complex tactile information. Besides, the compact housing design and universal USB interfaces allow for easy integration with other robotic hardware. However, due to the limitations of the RGB camera, this tactile sensor can only measure tactile features at a relatively low frequency, i.e. $25\,\mathrm{Hz}$.

Recently, a novel event-based optical tactile sensor called Evetac [5] has resolved this problem. Evetac is equipped with a neuromorphic event-based camera that can measure light intensity change on every single pixel with a measurement frequency of $498\,\text{Hz}$. With this practical feature, Evetac is highly sensitive to dynamic tactile features and has enormous potential to be applied in downstream tasks like fast and small contact movement detection. Nonetheless, Evetac's high measurement speed comes with the trade-off of a relatively lower image spatial resolution. Moreover, due to the working principle of event-based cameras [16], when there are no movements across the membrane, i.e. no light intensity change, Evetac is not able to capture any touch events and preserve any static tactile features.

2.2 Tactile Manipulation Policies

To better leverage high-dimensional readings from tactile sensors, learning-based approaches are usually used to capture implicit relationships between sensor measurements and manipulator dynamics. Nonetheless, online predictive approaches have also demonstrated remarkable results.

2.2.1 Reinforcement Learning Approaches

Reinforcement learning has been one of the most popular approaches to learning robotic manipulation policies [17] for a long time, enabling robots to learn task-specific skills through iterative interactions in simulated environments. Recent works [18–21] demonstrated the effectiveness and remarkable performance of integrating tactile feedback into a reinforcement learning framework for solving complex contact-rich manipulation tasks. [10, 19, 22] learned policies first in a simulation environment and subsequently transferred to a real robot system.

However, these approaches are usually very computationally expensive during the training stage as extensive exploratory learning requires a large number of training steps. Moreover, physics engines in the simulation environment necessarily approximate the real-world dynamics, leading to a sim-to-real gap where dynamics and visuals differ between simulation and reality [23]. To tackle such shortcomings, [24, 25] proposed to first learn an initial policy representation directly from a real system, later the learned policy is adapted with the tactile feedback via reinforcement learning approaches to generalize to different

task settings. Nevertheless, in these works, only capacitive tactile sensors are employed that can provide dozens of measurement points at most. Although the computational complexity is lowered, additional features such as object texture and shape are unable to be captured and the generalization remains limited.

2.2.2 Imitation Learning Approaches

In contrast, imitation learning approaches focus more on learning skill insights from a set of expert demonstrations that can be obtained either through kinesthetic teaching or teleoperation. As one of the most fundamental categories in imitation learning, behavior cloning (BC) allows training policy in a supervised style, significantly reducing computational costs. However, traditional BC policies [26–29] are not suitable for incorporating high-dimensional multimodal observations from demonstration, and these models also tend to be sensitive to distribution shifts w.r.t. training data. Their generalization ability is therefore very brittle in dynamic or unseen environments. Recently, diffusion-based visuomotor policies [8, 30–32] have addressed most of these shortcomings, allowing the policy to make long-horizon consistent action predictions while receiving informative multimodal observation sequences. In terms of our goal, i.e. to investigate the role of various modes of tactile information in the dynamic manipulation task, our framework is based on the diffusion policy [8] that naturally supports multimodal tactile images as input and has promising trajectory generation quality.

2.2.3 Online Predictive Approaches

Besides learning-based policies, some prior works formulated tactile manipulation tasks as online optimization problems in closed-loop control systems. Shirai et al. [33], Tian et al. [34] proposed MPC closed-loop controllers for tactile object manipulation. The tactile feedbacks are used to guide the online optimization of the predictive model and the framework showed high robustness against external disturbances. In addition, Wilson et al. [35] proposed an online motion primitive predictive framework that solves challenging cable routing and assembly tasks. During the online policy execution, various motion primitives will be predicted based on current tactile features. Despite the motion efficacy and robustness in closed-loop control, these non-learning approaches suffer from poor generalization ability and lack of scene-motion consistency.

2.3 Tactile Manipulation in Dynamic Tasks

Although dynamical tactile feedback was exploited in many prior works, i.e. object movement on the deformable membrane, to detect contact status or pose change, most of the task executions were rather slow, such as object grasping, pushing, door opening, peg-in-hole, etc. Only a few works [3, 4, 36] investigated how tactile information will affect policy performance on dynamic tasks, such as stick swing-up and slip control, with swift movements. Similar to us, George et al. [37] has also investigated the role of tactile sensing in manipulation tasks with a BC policy. However, their task setting, i.e. cable plugging, also executes slow movements. In our work, the understanding of tactile features in a dynamic scenario is crucial for the match ignition task, as such information not only indicates the timing of the contact between the match and the matchbox but also whether the contact is properly maintained during the subsequent striking motion to provide sufficient friction to ignite the match.

3 Foundations

In this chapter, we present the foundations of our proposed imitation learning framework. We begin with the background of the Diffusion Policy for learning tactile match-lighting skills. Subsequently, we introduce the Cartesian impedance control strategy that executes the policy-generated robot actions with a compliant behavior. Finally, we introduce the functioning principles of the optical tactile sensors, i.e. GelSight Mini and Evetac.

3.1 Diffusion Policy

Based on the fundamental schema of behavior cloning, Diffusion Policy [8] practically mapped observations to actions by formulating the robot visuomotor policy as a conditional denoising diffusion process [9]. This formulation generates actions by indirectly inferring the denoising process, which removes the noise added to the expected action sequence step by step, guided by the observation conditions. Most of the advantages of diffusion models are thus preserved, including expressing multimodal action distributions, outputting sequences in high-dimensional space, stable training, etc. In this section, we will go through the major theoretical details of diffusion policy.

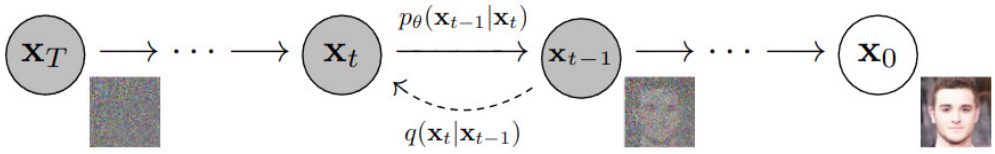


Figure 3.1: General demonstration of forward diffusion process (from $\mathbf{x}_T$ to $\mathbf{x}_0$), and reverse diffusion process (from $\mathbf{x}_0$ to $\mathbf{x}_T$). Figure is adapted from [9].

3.1.1 Behavior Cloning

Behavior cloning (BC), also called learning from demonstrations (LfD), is a commonly used technique in the field of imitation learning, where the policy learns task constraints and requirements from one or multiple expert demonstrations to eventually achieve adaptive behavior in unstructured environments [38]. A BC policy can be formulated as a mapping $\pi : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the observation space and $\mathcal{Y}$ is the action space. Both $\mathcal{X}$ and $\mathcal{Y}$ come from the expert demonstration dataset during training.

3.1.2 Diffusion Probabilistic Models

Diffusion Probabilistic Models [39] (also called diffusion models) are generative models that can model complex data distributions and generate high-quality samples.

Forward Process

Assume a data point $\mathbf{x}_0$ is sampled from a real data distribution $q(\mathbf{x})$, i.e. $\mathbf{x}_0 \sim q(\mathbf{x})$. As illustrated in 3.1, by adding a small amount of Gaussian noise to the sample for T times, a sequence of noised samples $\mathbf{x}_1, \cdots, \mathbf{x}_T$ can be obtained. This Markov chain is defined as the *forward process* of diffusion model:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

with

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

where t represents how many times (steps) the noise is added to the sample and $\{\beta_t \in (0,1)\}_{t=1}^{T}$ are used to control the variance of added noise in each step. During this process, the original sample becomes increasingly noisy and indistinguishable, and if $T \rightarrow \infty$ the final distribution $\mathbf{x}_T$ will become isotropic Gaussian. It's worth noting that $\mathbf{x}_t$ can be sampled at any time step t in closed form using the reparameterization trick [40]:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \tag{3.1}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Reverse Process

To remove added noise from time step t and reconstruct the sample at time step $t-1$, a model p_θ needs to be learned to estimate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ from which the true sample before noising can be sampled. This process is called the *reverse process* of diffusion models which is defined as a Markov chain as well with transition starts at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

with

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$$

Direct calculation of $p_\theta(\cdot)$ is intractable and thus we can obtain $p_\theta(\cdot)$ by optimizing the variational lower bound on negative log-likelihood [9]:

$$\begin{aligned}
L &= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] && (3.2)\\
&= \mathbb{E}_q \left[\log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] && (3.3)\\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^{T} \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] && (3.4)\\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] && (3.5)\\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] && (3.6)\\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \left(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \right) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] && (3.7)
\end{aligned}$$

$$= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \tag{3.8}$$

$$= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \tag{3.9}$$

$$= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \tag{3.10}$$

$$= \mathbb{E}_q \left[D_{\mathrm{KL}}\left(q(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T)\right) + \sum_{t=2}^{T} D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \tag{3.11}$$

It is noteworthy that in Eq. (3.5), $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is only tractable when conditioned on the start $\mathbf{x}_0$ in Eq. (3.6). Following Bayes' rule, the reverse conditional probability can then be transferred and further substitute $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$ in Eq. (3.7):

$$\begin{aligned} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) &= q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_t} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\sqrt{\bar{\alpha}_t}}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_t}\right)\mathbf{x}_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\mathbf{x}_{t-1}\right.\right. \\ &\quad \left.\left. + \mathrm{C}(\mathbf{x}_t, \mathbf{x}_0)\right)\right) \end{aligned} \tag{3.12}$$

where C is components that are without $\mathbf{x}_{t-1}$.
Decomposing the variational lower bound loss (3.11) into separate components based on various time steps, we have:

$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \qquad \text{if } t = 0 \tag{3.13}$$

$$L_t = D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t\mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) \qquad \text{if } 1 \le t \le T-1 \tag{3.14}$$

$$L_T = D_{\mathrm{KL}}\left(q(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T)\right) \qquad \text{if } t = T \tag{3.15}$$

The major advantage of using KL-Divergence is that the loss can eventually be calculated in closed form instead of high variance Monte Carlo estimates.

3.1.3 Denoising Diffusion Probabilistic Models (DDPMs)

The highly versatile implementation of diffusion models allows for the application of various simplifications to the reverse process, whereby latent variable models are transformed into a practical tool for generating high-quality samples. Ho et al. [9] proposed to fix L_T (3.15) as constant since $q(\cdot)$ has no learnable parameters after the reparameterization and thus can be ignored during training. Besides, for L_0 (3.13), an independent discrete decoder derived from $\mathcal{N}(\mathbf{x}_0; \mu_\theta(\mathbf{x}_1, 1), \sigma_1^2\mathbf{I})$ was introduced to obtain discrete log likelihoods.

Parameterization and Simplification of L_t

Recall that the goal of the reverse process is to learn a model

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \quad \text{for } 1 < t \leq T \tag{3.16}$$

that can estimate the forward process posteriors $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ which are tractable when conditioned on $\mathbf{x}_0$, i.e. $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}\mathbf{I})$. According to the standard Gaussian density function of the posterior (3.12), the mean and variance can be extracted through:

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)}{\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 \tag{3.17}$$

$$\tilde{\beta} = \frac{1}{\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{3.18}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ as defined before. Furthermore, following the reparameterization trick (3.1), $\mathbf{x}_0$ can be represented as:

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t) \tag{3.19}$$

plugging into the mean (3.17), we have:

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right) \tag{3.20}$$

Ultimately, $\boldsymbol{\mu}_\theta$ from Eq. (3.16) is trained to estimate $\tilde{\boldsymbol{\mu}}_t$ (3.20) given $\mathbf{x}_t$. Since $\mathbf{x}_t$ is available as model input during the training, similar parameterization can be applied to $\boldsymbol{\mu}_\theta$:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \tag{3.21}$$

As proposed by [9], L_t that used to minimize the difference between $\boldsymbol{\mu}_\theta$ and $\tilde{\boldsymbol{\mu}}_t$ can be written as:

$$L_t = \mathbb{E}_q \left[\frac{1}{2 \left\| \Sigma_\theta \right\|_2^2} \left\| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \right\|^2 \right] \tag{3.22}$$

Expanding Eq. (3.22) with Eq. (3.20) and Eq. (3.21):

$$L_t = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2 \left\| \Sigma_\theta \right\|_2^2} \left\| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \right\|^2 \right] \tag{3.23}$$

$$= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2 \left\| \Sigma_\theta \right\|_2^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_t \right) - \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \right\|^2 \right] \tag{3.24}$$

$$= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{(1-\alpha_t)^2}{2\alpha_t (1-\bar{\alpha}_t) \left\| \Sigma_\theta \right\|_2^2} \left\| \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right\|^2 \right] \tag{3.25}$$

Experimentally, Ho et al. [9] concluded that the unweighted version of L_t has advantages in improving the sample quality and reducing implementation effort. Thus Eq. (3.25) can be simplified as:

$$L_{t,\text{simple}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\left\| \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right\|^2 \right] \tag{3.26}$$

In summary, minimizing Eq. (3.26), which is generally a mean squared error (MSE), leads to minimizing the variational lower bound L (3.11) with KL-Divergences comparing distribution difference between forward and reverse process posteriors.

Once the model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in Eq. (3.16) is trained, we can sample $\mathbf{x}_{t-1}$ from this distribution with simplified variance $\sum_\theta(\mathbf{x}_t, t) = \sigma^2 \mathbf{I}$:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{3.27}$$

3.1.4 Observation Conditioned Denoising for Action-Sequence Prediction

To adapt DDPM to the generation of robot actions, Chi et al. [8] replaced the $\mathbf{x}$ that exists originally as images with a sequence of robot actions, which are typically M-dimensional end-effector Cartesian poses $\mathbf{x} \in \mathbb{R}^{T_p \times M}$, where T_p is the fixed length of the sequence. The action sequence generation is thus achieved by denoising a fully-noised action sequence for T steps. Another key modification of DDPM for visuomotor policy learning is to make the noise prediction network conditioned not only on the action sequence itself and the time step (3.27) but also on observations from external sensors e.g. cameras. More specifically, this conditioned denoising leads to predicting the distribution of the noised action sequence explicitly according to the sensor observations:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{O}_t, t) \right) + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{3.28}$$

where $\mathbf{O}_t$ is the observation data. It's worth noting that the observation data is not necessarily acquired from only a single time step but can be accumulated with multiple steps, i.e. $\mathbf{O}_t = \{o_0, \ldots, o_{T_o}\}$ where T_o indicates the history length of observation.

3.1.5 Training Diffusion Policy

As described in Section 3.1.4, the noise prediction network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{O}_t, t)$ will predict the noise added from the previous time step, conditioned on the observation history, the noised action sequence and sampled time step. Based on Eq. (3.26), the training criteria for this network can be written as:

$$L_{t,\text{conditioned}}(\theta) = \text{MSE}\left(\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{O}_t, t)\right) \tag{3.29}$$

where MSE stands for mean squared error.

Multimodal Representation of Visual Observations $\mathbf{O}_t$

Before sending the sensor observations to the noise prediction network, all of the image modalities of each observation step $o \in \{o_t, o_{t-1}, \ldots, o_{t-T_o}\}$ must be encoded into latent embedding space independently and concatenated with other low-dimensional modality embeddings as a multimodal representation.

Subsequently, all of the encoded single-step multimodal representations will be concatenated along time dimensions as the final observation condition embedding $\mathbf{O}_t$ for later conditional denoising. CNN-based models e.g. ResNet [41], VGG [42], AlexNet [43], etc are ideal candidates for image encoding. It's worth noting that as the model is used for feature extraction, corresponding structures thus must be modified to maintain the spatial information of the image. Experimentally, Chi et al. [8] suggested training the CNN-based image encoder end-to-end from scratch rather than using pre-trained weights.

Noise Prediction Network $\epsilon_\theta(\mathbf{O}_t, \mathbf{x}_t, t)$

Chi et al. [8] proposed two neural network architectures for noise estimation, i.e. CNN-based and transformer-based neural networks for estimating ϵ_θ, both networks accept observation sequence $\mathbf{O}_t$, noised action sequence $\mathbf{x}_t$ and corresponding noise step t and predict the noise added from last time step (theoretically, the network can also be trained to predict the original sample $\mathbf{x}_0$ directly but with the cost of worse sample quality [9]).

CNN-based noise prediction network is implemented by combining 1D temporal CNN [44], which encodes the noised action sequence along the time axis and Feature-wise Linear Modulation (FiLM) [45], which applies the observation conditioning to the action generation process.

Time-series transformer-based noise prediction network is generally based on a decoder-only transformer structure [46]. The condition data, i.e. time step t and observations $\mathbf{O}_t$ will be first concatenated with the order of t as first place, and then encoded into condition embeddings through an MLP encoder and later used as input features for the transformer decoder. Action sequences are then encoded by the same MLP and passed to the transformer decoder as the prediction target sequence. Finally, the decoder output will be the predicted noise with the same shape as the target sequence.

3.2 Cartesian Impedance Control

In our imitation learning framework, the policy-generated Cartesian trajectories will be directly sent to a Cartesian impedance controller with proper stiffness to keep the robot soft and compliant during the interaction.

Cartesian impedance control is a unique control strategy in robotics used to regulate the dynamic interaction between the robot manipulator and the external environment, which is very practical in the applications of contact-rich manipulation tasks.

Specifically, the impedance reflects how compliant the robot reacts to external perturbation. The Cartesian impedance control law abstracts the dynamic system between the robot end-effector and the environment into a virtual spring-damper system [47], and the external wrench applied to the end-effector frame will result in Cartesian displacements. The controller accepts the motion as system input and subsequently yields torque signals that make the robot react compliantly.

The rigid-body dynamics of the N joint robot can be written as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_c + \boldsymbol{\tau}_e \tag{3.30}$$

where $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ denote the joint parameters of the robot, i.e. joint position, velocity, and acceleration. $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{N\times N}$ is the inertia matrix, $\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) \in \mathbb{R}^{N\times N}$ is the Coriolis matrix, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^N$ is the gravity torque vector. $\boldsymbol{\tau}_c \in \mathbb{R}^N$ and $\boldsymbol{\tau}_e \in \mathbb{R}^N$ are control torque vector and torque caused by external wrench, respectively. Assume the gravity is compensated internally, Eq. (3.2) can be simplified as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} = \boldsymbol{\tau}_c + \boldsymbol{\tau}_e \tag{3.31}$$

3.2.1 Cartesian Impedance Control Law

The task space control torque $\boldsymbol{\tau}_{\text{task}} \in \mathbb{R}^N$ generated by Cartesian impedance control law [48] with M-dimensional Cartesian task space can be defined as:

$$\tau_{\text{task}} = \mathbf{J}^T(\mathbf{q})\left(\mathbf{K}_{\text{task}}\Delta\mathbf{x} - \mathbf{D}_{\text{task}}\Delta\mathbf{v}\right) \tag{3.32}$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{M\times N}$ is the Jacobian matrix relative to the end-effector frame, $\mathbf{K}_{\text{task}} \in \mathbb{R}^{M\times M}$ and $\mathbf{D}_{\text{task}} \in \mathbb{R}^{M\times M}$ are diagonal Cartesian stiffness and damping matrices, $\Delta\mathbf{x} = \mathbf{x}_d - \mathbf{x}, \mathbf{x} \in \mathbb{R}^M$ is Cartesian pose error with $\mathbf{x}_d$ as desired pose, detailed calculations under specific task configurations will be discussed in Sec. 4.5; $\Delta\mathbf{v} = \mathbf{v}_d - \mathbf{v}, \mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ is Cartesian velocity error with $\mathbf{v}_d$ as desired Cartesian velocity if applicable.

It's worth noting that we only consider the critical damping situation in this thesis, i.e. $\mathbf{D}_{\text{task}} = 2\sqrt{\mathbf{K}_{\text{task}}}$. Nevertheless, our policy generates Cartesian poses without extending to velocities, i.e. it's only applicable to positional control. In the damping part of Eq. (3.32), the velocity error thus becomes $\Delta\mathbf{v} = -\mathbf{v} = -J(q)\dot{q}$.

3.2.2 Nullspace Stiffness Regulation

For a robot with redundant joints, e.g. `Franka Emika Robot (Panda)` [49], constraining nullspace behavior is necessary while applying Cartesian impedance control, as Cartesian poses will have different joint space solutions with compliant flexible joints and inconsistent joint configurations will cause potential collisions in the environment. Nullspace stiffness regulation [50, 51] helps the joint configurations remain consistent w.r.t. a desired one by applying joint impedance control. Conveniently, this control behavior is also projected into the nullspace of the robot's Jacobian, therefore it will not affect the Cartesian motion. The nullspace control torque $\boldsymbol{\tau}_{\text{null}} \in \mathbb{R}^N$ can be written as:

$$\begin{aligned} \boldsymbol{\tau}_{\text{null}} &= \left(\mathbf{I} - \mathbf{J}^T(\mathbf{q})(\mathbf{J}^T(\mathbf{q}))^\dagger\right)\boldsymbol{\tau}_{\text{ref}} \\ \boldsymbol{\tau}_{\text{ref}} &= \mathbf{K}_{\text{null}}\Delta\boldsymbol{q} + \mathbf{D}_{\text{null}}\Delta\dot{\boldsymbol{q}} \end{aligned} \tag{3.33}$$

where $(\cdot)^\dagger$ is pseudo-inverse matrix, $\mathbf{I}$ is identity matrix, $\mathbf{K}_{\text{null}}$ and $\mathbf{D}_{\text{null}}$ are joint space diagonal stiffness and damping matrices. $\Delta\boldsymbol{q} = \boldsymbol{q}_d - \boldsymbol{q}$ and $\Delta\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}$ are joint space position and velocity error, respectively. Similar to the damping in task space torque, $\mathbf{D}_{\text{null}} = 2\sqrt{\mathbf{K}_{\text{null}}}$.

3.2.3 External Torques

A more precise impedance control can be achieved if torques exerted from the external wrench $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^N$ are involved in the controller:

$$\boldsymbol{\tau}_{\text{ext}} = \mathbf{J}^T(\mathbf{q})\mathbf{F}_{\text{ext}} \tag{3.34}$$

where $\mathbf{F}_{\text{ext}} \in \mathbb{R}^M$ is external wrench applied on the end-effector, can be measured from F/T sensor or estimated from joint torque measurements.

3.2.4 Ensemble Control Torque

The eventual control torque $\boldsymbol{\tau}_{\text{c}} \in \mathbb{R}^N$ in (3.31) for achieving compliant Cartesian behavior will be the superposition of Eq. (3.32), Eq. (3.33), and Eq. (3.34):

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_{\text{task}} + \boldsymbol{\tau}_{\text{null}} + \boldsymbol{\tau}_{\text{ext}} \tag{3.35}$$

More implementation details will be presented in Sec. 4.

3.3 Optical Tactile Sensing

In this thesis, we employ two types of optical tactile sensors, i.e. RGB-based and event-based tactile sensors. These sensors share similar working principles but are produced with different optical components and hardware designs to achieve diverse modes of tactile sensing. As illustrated in Figure 3.2, the most fundamental components of optical tactile sensors are the camera, elastic gel membranes with transparent acrylic support plates, and LED illumination. When an object is pressed on the gel membrane, it distorts to take on the shape of the object's surface [7], with LED illumination the camera will be able to capture the dynamic movement on the membrane. The key difference between these two tactile sensors is how they capture the image of deformation to interpret the tactile feedback.

3.3.1 GelSight Mini

Equipped with a regular RGB camera, GelSight Mini (lower row in Figure 3.2) captures the tactile image across three color channels and outputs the pixel intensity values of the whole frame at once in a fixed frequency. The global tactile information, such as the object shape and pressure magnitude, can be preserved (as shown in Figure 3.3), bringing the additional context of contact to the downstream tasks. However, this continuous capture of full frames leads to the generation of vast amounts of data, even in scenarios where there is minimal or no change on the membrane. Therefore, one of the major drawbacks of RGB-based tactile sensors is the low measurement frequency caused by heavy data processing load.

3.3.2 Evetac

We will first introduce the key working principle of the event-based camera mounted inside Evetac, and then explain how Evetac captures tactile features.

Event-Based Vision

Unlike traditional frame-based CMOS cameras that output an entire image frame, the pixels of event-based cameras work independently and only respond to the brightness

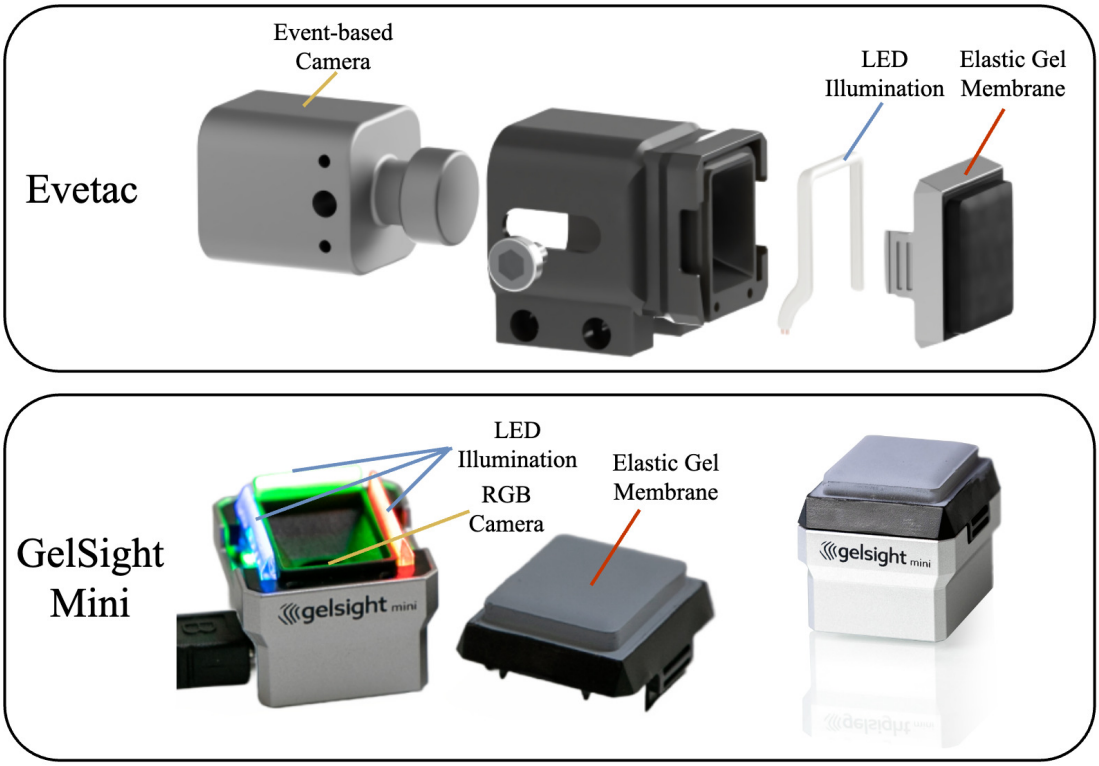


Figure 3.2: Components of vision-based tactile sensors. Upper row: Evetac [5] with an event-based camera and single-tone LED strip. Lower row: GelSight mini [7] with RGB camera and tri-colored LED. Both sensors have similar gel membranes that can be exchanged.

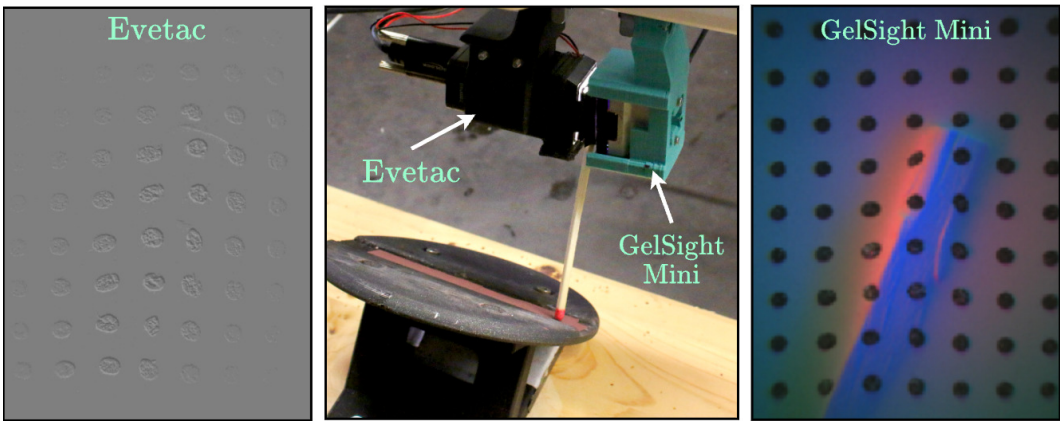


Figure 3.3: Tactile feature of a match contacting surface, captured by Evetac and GelSight Mini. Evetac (left) reads out the pixel intensity change of each pixel independently while GelSight Mini (left) outputs the whole frame at once. For better visualization, the Evetac image is the accumulation of the latest 10 measurements, i.e. the events triggered within the last $10\,\mathrm{ms}$.

magnitude change [52]. Assuming a constant illumination and a noise-free scenario, at time t_k, the brightness of a pixel located at (x, y) can be written as the log of the photocurrent $I(x, y, t_k)$:

$$L(x, y, t_k) = \log(I(x, y, t_k)) \tag{3.36}$$

After time interval Δt_k, the brightness change reaches a temporal contrast threshold $\pm C$, i.e.:

$$\Delta L(x, y, t_{k+1}) = L(x, y, t_{k+1}) - L(x, y, t_k) \tag{3.37}$$

$$= C p_{k+1} \tag{3.38}$$

where $t_{k+1} = t_k + \Delta t_k$, $C > 0$, and $p_k \in \{+1, -1\}$ is the polarity that indicates the brightness change tendency i.e. increase or decrease. At this change, an event $e_{k+1}(x, y, t_{k+1}, p_{k+1})$ will be instantly triggered. Subsequently, further events will be triggered w.r.t. the changed brightness $L(x, y, t_{k+1}) = L(x, y, t_k) + \Delta L(x, y, t_{k+1})$. The measurement time intervals Δt_k (also called temporal resolution) are typically extremely tiny, for instance, $65\,\mu\mathrm{s} - 200\,\mu\mathrm{s}$.

Empirically, to reduce the sparsity of the data and obtain a consistent output frequency, the events are configured to be accumulated for $1\,\mathrm{ms}$ before sending to the computer.

The events to be transmitted $\mathcal{S}_E(t_i)$ with a total amount of N_E that are measured in the previous millisecond t_i can be written as:

$$\mathcal{S}_E(t_i) = \{e_k(x_k, y_k, t_k, p_k), k \in N_E\} \tag{3.39}$$

where $t_k \in [t_{i-1}, t_i)$ with $t_i - t_{i-1} = 1\,\text{ms}$. The final output of the event-based camera will be a stream of these pre-accumulated events. The high temporal resolution allows the event-based camera to capture fast movement with a relatively lower power consumption due to a small data processing load.

The Evetac Sensor

The Evetac event-based optical tactile sensor is built with off-the-shelf hardware components, including 3D-printed housing, LED strip, gel membrane with imprinted black dots (exchangeable with GelSight Mini), and Inivation DVXplorer Mini event-based camera.

To visualize the tactile features as an image, Funk et al. [5] proposed to assign gray color to the pixels where no events are triggered, assign white to the event-triggered pixels with increasing brightness, and black to decreasing brightness. However, the functioning principle of the event-based camera results in rather sparse information on the tactile images, so further accumulation of events for obtaining more comprehensive tactile features is necessary.

As shown in 3.3, the example image is the accumulation of $10\,\text{ms}$ and the imprinted dots are therefore visible when there are distortions on the membrane such as the grasped match in contact with the surface. More configurations and implementation details of the pre-processing of tactile images will be introduced in Sec. 4.

4 Methodology

In this chapter, we start by introducing the challenging manipulation task that we aim to solve: lighting up a match, and providing an overview of the proposed imitation learning framework. Next, we detail the process of collecting human demonstrations and corresponding dataset preprocessing. Following that, we outline the training pipeline for Diffusion Policy. Finally, we present the action inference pipeline of Diffusion Policy based on multimodal sensor observation, integrating with other components of the framework, to achieve real-world task execution.

4.1 Task Specifications and Assumptions

Task Description

Since the motivation of this thesis is to investigate the importance of tactile sensing in dynamic manipulation tasks, we introduce the match-lighting task. This task requires the robot to automatically light up a match by striking it on the matchbox, then stop and wait for the fire to extinguish.

There are three main phases to this task:

1. Approach the matchbox, specifically the striker paper, and bring the match in contact with the striker paper.
2. Strike the match along the striker paper until the fire is lit.
3. Stop in a safe position and wait for the fire to extinguish.

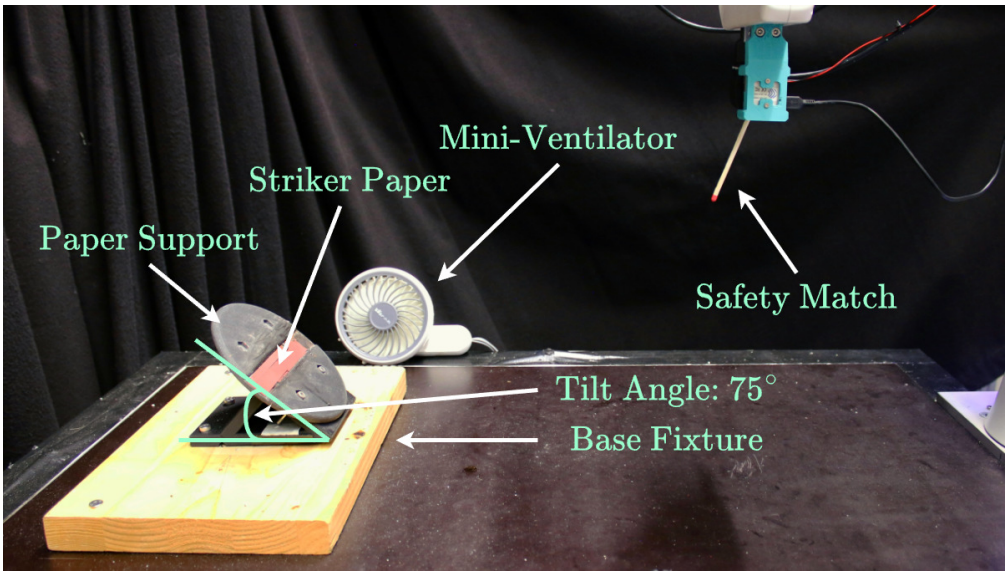


Figure 4.1: Environment setup for match ignition task. A safety match is a type of match that can only be ignited by striking against specially prepared striker paper.

Environment Setup

The regular matchboxes are made of paper, which results in short durability after a few experiments. Herein, we decompose the matchbox and only keep the striker paper for repeat use in experiments. As illustrated in Figure 4.1, we design and 3D-print a thin round plate with a diameter of $120\,\text{mm}$ as the substitution of the matchbox to support the striker paper. A rectangular shallow groove with a rough dimension of $20\,\text{mm} \times 120\,\text{mm} \times 3\,\text{mm}$ is made across the diameter to help locate the striker paper.

Moreover, to prevent potential robot joint singularities that occur when the plate is flat on the table, we raise the plate to form a tilt angle of $75°$. Close to the striker paper support, we place a mini-ventilator that can provide strong airflow to put the fire out and clear smoke. In addition, to ensure the safety of the experiments, we always kept fire extinguishers within handy reach. The matches we used for experiments are standard safety matches with a dimension of $(100\,\text{mm} \pm 5\,\text{mm}) \times (4\,\text{mm} \pm 1\,\text{mm}) \times (4\,\text{mm} \pm 1\,\text{mm})$.

Assumptions

For the task evaluation, we have the following assumptions:

1. Brightness of the ambient illumination is homogeneous.
2. The match is already properly gripped in the gripper and stays in a fixed start pose before the experiment trials.
3. The connection between the striker paper holder and the table is rigid.
4. The mounting positions of the sensors are fixed from data collection to experiments.

Requirements

This task was previously proposed by Kronander and Billard [53], and they solved this task by learning a variable impedance policy based on a time-conditioned Gaussian process (GP). Specifically, the policy only learns when to adjust the Cartesian stiffness to an appropriate value given human feedback, so that the robot doesn't break the match with too much force while maintaining sufficient tracking accuracy to successfully reach the matchbox. Despite the high efficiency of their approach, the generalization ability of the policy remains unfortunately at a lower level, since no sensors were involved to observe the external environment and the in-hand pose of the match.

With their experiences, reconsider this task with description in Sec. 4.1, the key requirements for finishing this task lie in the following perspectives:

1. Adjust robot end-effector pose according to different in-hand poses of the match before striking.
2. Correctly find out where the striker paper is and the correct contact area for the match tip.
3. Fast and robust contact status estimation, i.e. whether the match is in contact with the striker paper.
4. During striking, regulate the contact force appropriately to prevent the match from breaking while providing sufficient contact friction to ignite the match.

To solve this task with all its requirements, we propose a framework that is based on an imitation learning policy, i.e. Diffusion Policy, which accepts multimodal sensor observation sequence as input and output corresponding action sequence prediction. In addition, besides the wrist camera, we involve different kinds of tactile sensors to provide informative tactile feedback on the match e.g. in-hand pose and movements during dynamic contact. Lastly, since we are tackling a contact-rich task without complex joint-space motion planning, a Cartesian impedance controller is thus employed to make the robot's behavior more compliant and prevent excessive contact force during the interaction.

4.2 Data Collection

Since our policy typically learns tactile skills from human demonstrations, data collection is one of the most important preliminaries. In this section, we present the method of demonstration collection and corresponding dataset preprocessing.

4.2.1 Kinesthetic Teaching

The human demonstration collection is commonly performed in two ways: teleoperation and kinesthetic teaching. Despite teleoperation has advantages in reducing the physical effort of the demonstrator and providing a broad range of movement range, for the match ignition task, it is essential for the demonstrator to have an immediate haptic sense of the contact establishment and continuous precise contact feedback while striking the match to ensure that the match is successfully lit up without breaking. With these specific requirements, we choose to use kinesthetic teaching to collect human demonstrations.

We employ a `joint gravity compensation controller` [54] that automatically compensates the gravity applied to the robot and sensors, which allows the demonstrator to grab and move the robot without pressing the guiding buttons on the wrist, such that the task can be demonstrated easily and precisely.

4.2.2 Sensor Mounting

For further investigation of how different sensor combinations affect policy performance, we involve all the sensor modalities in the collected demonstrations. Experimentally, we

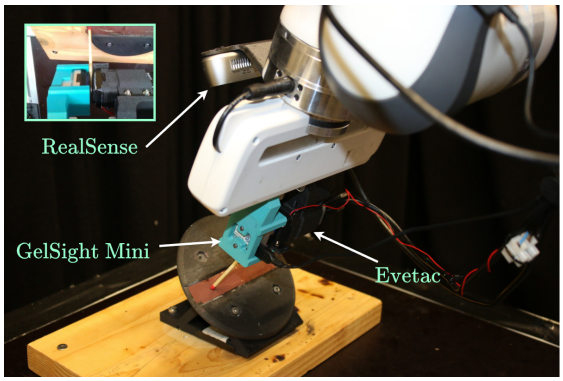


Figure 4.2: Sensor mounting positions. The RealSense is mounted on the wrist joint (7th joint) of Panda to provide a gripper view (mini figure on the top left corner). GelSight Mini and Evetact are mounted on both fingers of the gripper with 3D-printed holdings that guarantee the gel membranes of each sensor are aligned.

mount the RealSense camera with a 3D-printed holder on the wrist to provide a clear and complete view of the gripper and the gripped match as displayed inside Figure 4.2.

4.2.3 Demonstration Categorization

In our demonstration dataset, each demonstration contains information on GelSight Mini, Evetac, RealSense, and robot proprioception. The recording rate of each sensor can be found in Table 4.1. Note that for Evetac, there is an important trade-off between

Sensor Data	Recording Rate
Robot Proprioception	600 Hz
Evetac	100 Hz
RealSense	30 Hz
GelSight Mini	18 Hz

Table 4.1: Demonstration recording rates of all the involved sensors.

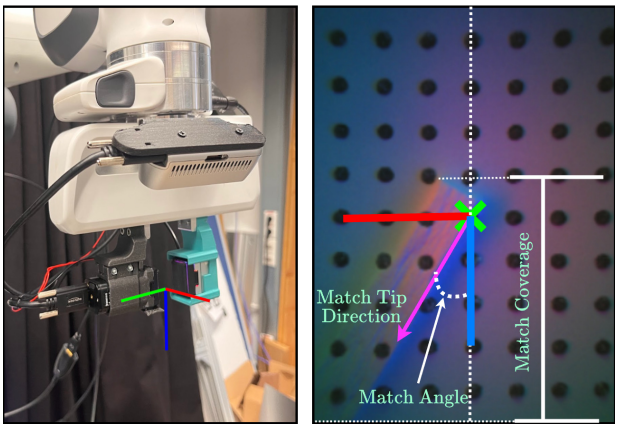


Figure 4.3: Characterizing the match pose. The left figure displays the robot end-effector frame with the red line indicating the x-axis, the green line indicating the y-axis, and the blue line indicating the z-axis. The axes are projected onto the right figure, which illustrates the match angle (angle between the match tip direction and the z-axis) and match coverage (the length of the match projected on the central vertical dotted white line of the tactile image).

the measurement rate and the information density on tactile image frames. We thus choose 100 Hz, i.e. 10 ms of event accumulation time, for a better representation of tactile features.

Due to the uniqueness and non-reusability of the matches, recording demonstrations with identical match poses is impossible. In order to make the dataset more structured in terms of the match pose, the demonstrations are characterized by two major factors that define the in-hand pose of the match, i.e. match angle and match coverage. Both factors are evaluated by observing the tactile image of GelSight Mini.

As illustrated in Figure 6.5, **match angles** is defined as the angle between the match tip direction and the z-axis of the robot end-effector frame (left part of the figure). The **match coverage** indicates how long the match is covered by the gel membrane, and is defined by the projection of the visible match length onto the vertical center reference line. We designed a match stand that can hold the match vertically with the match tip pointing to the table, a pre-defined trajectory will command the robot to grasp the match with a specific match angle and coverage.

4.2.4 Dataset and Preprocessing

Assume a dataset of N_D human demonstrations $\mathcal{S}_{D_i} = \{\mathcal{D}_i\}_{i=1}^{N_D}$, each demonstration contains time-stamped sensor data and robot end-effector trajectory, i.e. $\mathcal{D}_i = \{\boldsymbol{\Phi}_i, \boldsymbol{\mathcal{T}}_i\}$. The sensor data includes images from the Realsense camera $\boldsymbol{\phi}_{\text{rs},k} \in \mathbb{R}^{H_{\text{rs}} \times W_{\text{rs}} \times 3}$, GelSight Mini $\boldsymbol{\phi}_{\text{gs},k} \in \mathbb{R}^{H_{\text{gs}} \times W_{\text{gs}} \times 3}$, and Evetac $\boldsymbol{\phi}_{\text{et},k} \in \mathbb{R}^{H_{\text{et}} \times W_{\text{et}} \times 1}$, with $H_{(\cdot)}$ and $W_{(\cdot)}$ as height and width, respectively:

$$\boldsymbol{\Phi}_i = \{[\phi_{\text{rs},k}]_{k=1}^{l_{\text{rs}}}, [\phi_{\text{gs},k}]_{k=1}^{l_{\text{gs}}}, [\phi_{\text{et},k}]_{k=1}^{l_{\text{et}}}\} \tag{4.1}$$

where $l_{(\cdot)}$ indicates the total amount of images of each sensor. It's worth noting that these amounts are different due to the various sensor measurement rates. The end-effector trajectory is represented in the Cartesian space with C dimensions:

$$\boldsymbol{\mathcal{T}}_i = [\tau_k]_{k=1}^{l_\tau}, \quad \tau_k \in \mathbb{R}^C \tag{4.2}$$

where l_τ is the trajectory length. In our framework, we use the full Cartesian pose with quaternion rotation, i.e. $C = 7$.

To reduce the size of the dataset, we downsample both the sensor data and trajectory with a specific frequency f_{ds}, that is lower the frequency of the slowest sensor f_{min}. After downsampling, all data sequences have the same length of $\hat{l}_{\text{data}}$:

$$\begin{aligned} d_{\text{data}} &= \frac{f_{\text{data}}}{f_{\text{ds}}} \\ \hat{l}_{\text{data}} &= \frac{l_{\text{data}}}{d_{\text{data}}} \end{aligned} \tag{4.3}$$

where the subscript data indicates the data to be downsampled i.e. raw sensor data and trajectory; d_{data} is the downsample factor, $\hat{l}_{\text{data}}$ is the length after downsampling, and f_{data} is the data collection frequency.

Sequence Sampling

As described in Sec. 3.1.5, the noise prediction network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{O}_t, t)$ is trained upon the robot actions $\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon})$ and sensor observations $\mathbf{O}_t$, which are both temporal sequences with specific length. Herein, for the convenience of training, we split the demonstrations into short observation-action-paired temporal sequences:

$$\mathbf{D}_i = \{\mathcal{S}_k | \mathcal{S}_k = (\mathbf{O}_t, \mathbf{x}_t)\}_{k=1}^{N_S} \tag{4.4}$$

where $\mathbf{x}_t = [x_{t-T_0}, \ldots, x_t, x_{t+1}, \ldots, x_{t+(T_p-T_o)}]$, $\forall \mathbf{x}_t \in \boldsymbol{\mathcal{T}}_i$; $\mathbf{O}_t = [O_{t-T_o}, \ldots, O_{t-1}, O_t]$, $\forall O \in \{\boldsymbol{\Phi}_i, \boldsymbol{\mathcal{T}}_i\}$ are multi-modality observation history sequence with the length of T_o and corresponding expected future action sequence with the length of T_p with only low-dimensional-modalities, respectively. $N_S \in (0, \hat{l}_{\text{data}})$ is the total number of sequences. More details about observation modality combinations will be presented in Sec. 5.

Normalization

As suggested by [9], all the data used for training the denoising model should be normalized between $(-1, 1)$. Thus, for the low dimensional data in the sampled sequences, e.g. robot end-effector pose, we use the modified min-max normalization:

$$\hat{\mathbf{x}}_{\text{data}} = 2 \cdot \left(\frac{\mathbf{x}_{\text{data}} - \mathbf{x}_{\text{min, data}}}{\mathbf{x}_{\text{max, data}} - \mathbf{x}_{\text{min, data}}} \right) - 1 \tag{4.5}$$

where $\hat{\mathbf{x}}_{\text{data}} \in (-1, 1)$ is the normalized data while $\mathbf{x}$ is the original data. $\mathbf{x}_{\text{max, data}}$ and $\mathbf{x}_{\text{min, data}}$ are the maximum and minimum of the data across the whole dataset, and these values will be preserved for later denormalization. More details are described in Sec. 4.4.

For image modalities, e.g. tactile images, we employed mean-std normalization:

$$\hat{\mathbf{x}}_{\text{data}} = \frac{\mathbf{x}_{\text{data}} - \mathbf{x}_{\text{mean, data}}}{\mathbf{x}_{\text{std, data}}} \tag{4.6}$$

where $\hat{\mathbf{x}}_{\text{data}} \in (-1, 1)$ is the normalized data while $\mathbf{x}$ is the original data. $\mathbf{x}_{\text{mean, data}}$ and $\mathbf{x}_{\text{std, data}}$ are the configurable mean and standard deviation on each color channel.

4.3 Policy Training

Given a preprocessed dataset, we then train the noise prediction model of Diffusion Policy w.r.t. the observation and action sequences.

Visual Observation Encoder

To reduce the image dimensionality and extract informative features, in each time step O_t, we first encode all the image modalities into latent space. Subsequently, the encoded vectors will be concatenated with each other, and also with the rest of the low-dimensional modalities as a single flat vector:

$$O_t = [m_1|\dots|m_{N_m}], \quad O_t \in \mathbb{R}^{l_o} \tag{4.7}$$

where | operator indicates concatenation, m is the modality, N_m is the total number of observation modalities, d_o is the length of the vector after concatenation.

Finally, we concatenate the observation vectors along the time dimension as the final multi-modal observation representation vector $\mathbf{O}_t \in \mathbb{R}^{T_o \times l_o}$ as mentioned in (4.4). Empirically, each image modality should have its own encoder and not share weights with other encoders, and CNN-based models are ideal candidates due to their simplicity and efficiency.

In this thesis, we employ ResNet18 models from [55] that can be configured to accept both RGB images (e.g. GelSight and RealSense) and grayscale images (e.g. Evetac) and output latent vectors with specific lengths. The weights of encoders are randomly initialized and the batch norm layers are replaced by group norm layers to ensure training stability [8]. During the training, the optimization of encoder weights is guided by the same MSE error from noise prediction.

Noise Prediction Model

Considering the relatively high velocity of policy execution in our task configuration that will potentially cause a fast action changing rate, we choose the time-series transformer as the noise prediction model introduced in Sec. 3.1.5 to improve the action sequence prediction consistency in a dynamic scenario. Besides, to reduce vanishing or exploding gradients and accelerate the converging speed, we initialize the transformer decoder with He initialization [56].

Training Pipeline Overview

The complete training process is illustrated in Figure 4.4. Assume the preprocessed observation $\mathbf{O}_t$ and action $\mathbf{x}_t$ sequences from the dataset are available, we draw a noise

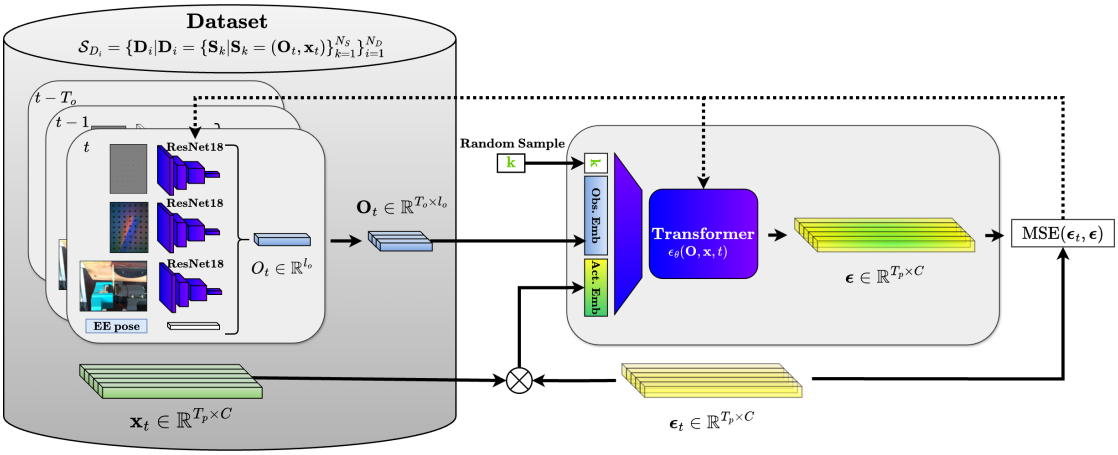


Figure 4.4: Traning pipeline of the noise prediction transformer. The dataset are split into observation-action paired temporal sequences. The image modalities are encoded into latent space and concatenated as a single vector. During the training, the clean trajectory from the dataset will be noised with randomly initialized Gaussian noise, the decoder-only transformer has to predict the added noise according to observation and time steps. MSE loss is calculated to guide the training process.

sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ from Gaussian distribution and a random time step of forward diffusion process $\mathbf{k}$. The sampled noise $\boldsymbol{\epsilon}_t$ is added to the original action sequence with the magnitude w.r.t. time step $\mathbf{k}$ and then the noisy sequence is encoded into embedding. Similarly, the observation sequence is combined with the time step and then also encoded into embedding. Both embeddings are passed through the multi-layer transformer decoder to obtain a predicted noise $\boldsymbol{\epsilon}$. As introduced in Sec. 3.29, the mean squared loss between the sampled noise $\boldsymbol{\epsilon}_t$ and predicted noise $\boldsymbol{\epsilon}$ is calculated and backpropagated through both the transformer and encoder networks, guiding the optimizer to update model weights.

4.4 Policy Inference

Once the noise prediction network is successfully trained, it can be integrated into the policy inference pipeline to generate task-oriented trajectories. Figure 4.5 demonstrates the whole inference-control pipeline similar to the paradigm of Model Predictive Control (MPC).

Assume the observation sampling is running at the same frequency as the dataset frequency after downsampling f_{ds} that's used in Sec. 4.2.4. The observations that include all of the required modalities are first stored in an observation buffer, which will only preserve T_o steps of history. As this buffer operates in a First-In, First-Out (FIFO) style, new observations will replace the old ones immediately when squeezed in and thus guarantee that the buffered observations are always the latest. Normalization and image encoding are applied to the sampled observation sequence and produce the multimodal observation history vector $\mathbf{O}_t = [O_{t-T_o}, \ldots, O_t]$ with O_t indicating the current time.

Subsequently, we randomly initialize a fully noised trajectory at K time step: $\mathbf{x}_t^K \in \mathbb{R}^{T_p \times C}$. With all the ingredients for conditional denoising ready, the noise prediction network starts predicting noise that has been added to the $K-1$ step conditioned on the observation sequence, and the noise scheduler will sample the trajectory from the previous time step w.r.t. the noise prediction using (3.27). This conditional denoising process will iterate for K times and eventually yield a clean trajectory. However, this trajectory is normalized and can't be directly sent to the controller, a denormalizing transform is thus necessary. Recall Eq. (4.5) in Sec. 4.2.4, assuming $\mathbf{x}_{\text{min, data}}$ and $\mathbf{x}_{\text{max, data}}$ are available, the denormalization can be written as:

$$\mathbf{x}_{\text{data}} = \frac{(\hat{\mathbf{x}}_{\text{data}} + 1) \cdot (\mathbf{x}_{\text{max, data}} - \mathbf{x}_{\text{min, data}})}{2} + \mathbf{x}_{\text{min, data}} \tag{4.8}$$

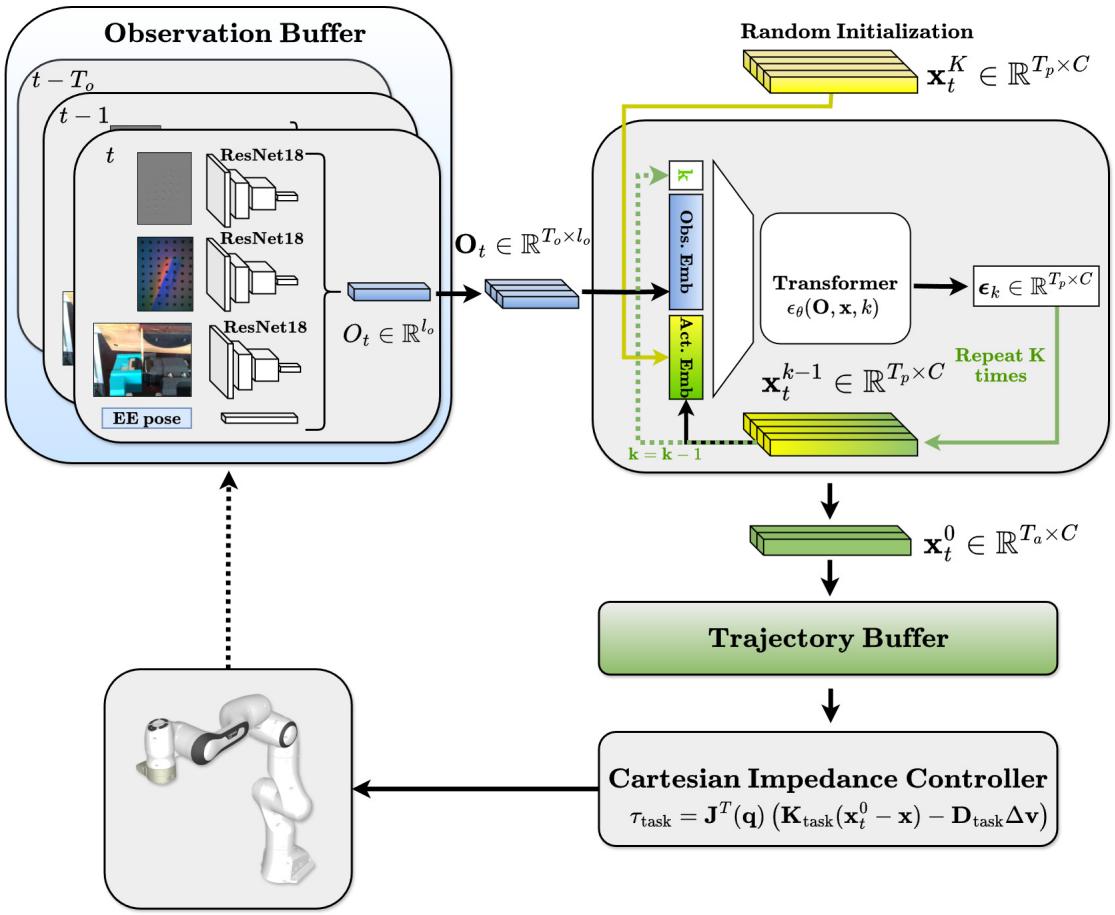


Figure 4.5: Policy inference and control pipeline. The observation buffer takes observations for T_o steps. Multimodal observation latent vectors are produced with specific frequency online. The conditional denoising process will remove noise from the randomly initialized trajectory step by step. The clean trajectories are buffered while the trajectory inference is still running for trajectories. The latest trajectory will be sent to the controller.

where $\hat{\mathbf{x}}_{\text{data}}$ is the normalized data and $\mathbf{x}_{\text{data}}$ is the original data.
The denormalized trajectories are first truncated with the length of T_a, then interpolated to the frequency of the robot controller which is 1000 Hz for Panda, and then they're ready to be sent. For the positional part of the Cartesian trajectory, we used linear interpolation, while for the quaternion rotation part, we used Spherical Linear Interpolation (Slerp). Additionally, in order not to block subsequent trajectory inference, we push the current trajectory into a trajectory buffer, which is also FIFO-like with a specific length.

4.5 Robot Control

Once the trajectory buffer is full, it will publish the latest interpolated trajectory **point-by-point**, since the controller we use only supports single target pose setting rather than a whole trajectory with motion planning. The trajectory publishing rate is the same as the controller torque command updating frequency, i.e. 1000 Hz.

4.5.1 Cartesian Error Calculation

In Sec. 4.2.4, we introduced that our framework uses full Cartesian poses as the trajectory representation. The calculation of the Cartesian error $\Delta\mathbf{x} \in \mathbb{R}^6$ is split into two parts, i.e. positional error $\Delta\mathbf{x}_{\text{pos}} \in \mathbb{R}^3$ and rotational error $\Delta\mathbf{x}_{\text{rot}} \in \mathbb{R}^3$ [57].

The positional error is simply the element-wise subtraction between the current $\mathbf{x}_{\text{pos}}$ and the target $\mathbf{x}_{\text{pos,d}}$ position:

$$\Delta\mathbf{x}_{\text{pos}} = \mathbf{x}_{\text{pos,d}} - \mathbf{x}_{\text{pos}} \tag{4.9}$$

For the rotational error, we first assume the rotation matrix of the current Cartesian pose is $\mathbf{R} \in SO(3)$, then we calculate the difference between the current quaternion $\boldsymbol{\theta}$ and target quaternion $\boldsymbol{\theta}_d$, i.e. $\Delta\boldsymbol{\theta} \in \mathbb{R}^4$:

$$\Delta\boldsymbol{\theta} = \boldsymbol{\theta}^{-1}\boldsymbol{\theta}_d \tag{4.10}$$

Subsequently, we rotate it back to the current task frame to obtain the final rotational error:

$$\Delta\mathbf{x}_{\text{rot}} = -\mathbf{R}\Delta\hat{\boldsymbol{\theta}} \tag{4.11}$$

where $\Delta\hat{\boldsymbol{\theta}} = \text{Im}(\Delta\boldsymbol{\theta}) \in \mathbb{R}^3$ is the imaginary vector of the difference quaternion. Note that in this thesis, the task frame is the nominal end-effector frame of Panda.

4.5.2 Error Smoothing

As introduced in Sec. 3.2.1, the task space control torque is significantly affected by the Cartesian error $\Delta\mathbf{x}$. The policy-generated trajectories are sometimes quite spiky, which will cause bumpy Cartesian errors and result in unexpected shaking of the task frame. To reduce this instability, we apply an online low-pass filter to the errors. Assuming at time t:

$$\Delta\mathbf{x}_{t+1} = \Delta\mathbf{x}_t(1-\eta) + \eta\Delta\mathbf{x}_d \tag{4.12}$$

where $\Delta\mathbf{x}_d$ is the Cartesian error between the current pose and the next desired pose, $\eta \in (0, 1]$ is a tunable value that adjusts the smoothness magnitude.

5 Experiments

In this chapter, we discuss the details of the experiment setup, including the software and hardware platforms for the framework implementation. Subsequently, we present the demonstration dataset categorized according to different match poses, and policy configurations in both training and inference scenarios.

5.1 Framework Implementation Overview

The implementation of our imitation learning framework is based on the open-sourced communication software `ROS (Noetic Ninjemys)` [58] and `Ubuntu 20.04` Linux operating system. The robot platform is `Franka Emika Robot (Panda)` [49] with parallel `Franka Gripper`.

The deep learning frameworks involved in the thesis are `PyTorch` [59] and `Huggingface Diffusers` [60]. The policy implementation is partially adopted from the GitHub repository of [8].

The implementation of ROS-based controllers is adapted from `libfranka` [57] and `Franka Interactive Controllers` [54].

5.2 Dataset Overview

We collected a total of 108 demonstrations of match lighting demonstrated by one demonstrator using kinesthetic teaching. As introduced in Sec. 4.2.3, we categorize the dataset regarding the match angle and coverage, the overview of the number of demonstrations collected according to match coverage and angle can be found in Table 5.1. Note that the match coverage has an average error of ± 6 mm and the match angle has an average

	$-20°$	$-10°$	$0°$	$10°$	$20°$	$30°$
7 mm (Short) Coverage	8	8	5	5	5	5
14 mm (Middle) Coverage	8	8	5	5	5	5
21 mm (Long) Coverage	8	8	5	5	5	5

Table 5.1: Number of demonstrations categorized by match coverage and angle. The match coverage (rows) has an average error of ± 6 mm while the match angle (columns) has an average error of $\pm 5°$.

error of $\pm 5°$. Besides, for negative match angles, $-20° \pm 5°$ is already the maximum angle that the human demonstrator can deal with, greater angles will lead to a high failure rate. Herein, to prevent an unbalanced dataset, we collected more negative-angle demonstrations.

5.3 Policy Configurations

In this section, we report the key parameter configurations for the training and inference of the Diffusion Policy.

5.3.1 Training Configurations

Policy meta parameters are reported in Table 5.2, hyperparameters for observation encoder, transformer, noise scheduler, and training can be found in Table 5.5.

Dataset Split

We split the demonstration dataset into two parts, i.e. train and test set. For the test set, we randomly pick 2 demonstrations from match poses $\{(\mathrm{C}, \mathrm{A}) | \mathrm{C} \in \{\mathrm{Short}, \mathrm{Middle}, \mathrm{Long}\}, \mathrm{A} \in \{-20°, -10°\}\}$, and randomly pick 1 demonstration from match poses of $\{(\mathrm{C}, \mathrm{A}) | \mathrm{C} \in \{\mathrm{Short}, \mathrm{Middle}, \mathrm{Long}\}, \mathrm{A} \in \{0°, 10°, 20°, 30°\}\}$. Eventually, $2 \times 6 + 1 \times 12 = 24$ demonstrations are used to compose the test set, and the rest 84 demonstrations are used as the train set.

Type	Steps
Observation History T_o	2
Action Prediction Length T_p	10
Action Execution Length T_a	8

Table 5.2: Policy Meta Parameters.

Sensor	Raw Shape	Final Shape
GelSight Mini	$(3, 3840, 2160)$	$(3, 144, 108)$
Evetac	$(1, 640, 480)$	$(1, 144, 108)$
RealSense	$(3, 640, 480)$	$(3, 120, 120)$

Table 5.3: Sensor image shapes before and after preprocessing.

Image Preprocess and Augmentation

After sequence sampling in 4.2.4, we preprocess the image modalities. Specifically, for GelSight Mini and Evetac, we resize the tactile images to the same size; for RealSense images, we first center crop the image to preserve only the ROI where the sensor fingers and the match are visible, then resize the image to a smaller size. During the training, we also apply data augmentation on the image modalities with transformations from `Torchvision` [61] to increase the generalization ability of the model. Experimentally, we apply `RandomCrop` on all of the images to increase data spatial variety; for GelSight Mini and RealSense images, we apply `ColorJitter` and `RandomAutoContrast` that randomly shifts the color and brightness of the image respectively to simulate sensor color change w.r.t. illumination; for Evetac we apply `GaussianNoise` to simulate event noise. For detailed values please refer to Table 5.3.

Sensor Modality Combination

For the evaluation experiments, we trained multiple combinations of sensors with other hyperparameters fixed. The sensor combinations are categorized with the number of modalities N_m in Table 5.4.

Combination	Num. Modalities N_m
RealSense, GelSight Mini, Evetac	3
RealSense, GelSight Mini	2
RealSense, Evetac	2
GelSight Mini, Evetac	2
RealSense	1

Table 5.4: Sensor combinations that are used for model training and evaluation.

5.3.2 Inference Configurations

The hyperparameters for inference are presented in Table 5.6.

ONNX Inference Acceleration

To increase the trajectory sampling speed of the policy, we use the ONNX runtime framework [62] to reduce the precision of transformer weights from `fp32` to `fp16`. This results in a $3\times$ faster inference speed than the full-precision model, and allows the trajectory generation to run at $10\,\text{Hz}$ on average.

Stiffness Tuning for Controller

Apart from the Cartesian error, another key factor that contributes to the task control torque is the task frame stiffness. It is necessary to tune the stiffness to carefully trade off both a compliant behavior of the task frame while interacting with the external environment, and sufficient tracking accuracy to ”honestly” follow the policy-generated trajectory. Experimentally, we found $\text{diag}([1200, 1200, 1200, 50, 140, 50]^T)$ meet our needs perfectly.

ResNet18	
Input Shape	$\{(3, 108, 144), (3, 120, 120), (1, 108, 144)\}$
Output Shape	128
Pre-trained	No
Transformer	
Input and Output Shape	$(T_p, 7)$
Observation Conditional Feature Length	$(N_m \times 128 + 7)$
Weight Initialization	He Initialization
Num. Decoder Layer	8
Num. Head	4
Len. Embedding	512
DDPM Noise Scheduler	
Num. Timesteps	100
β Range	$[0.0001, 0.2]$
β Schedule	Cosine
Training	
Training Epochs	2000
Batch Size	128
Learning Rate	0.0001
Learning Rate Scheduler	Cosine Annealing
Warmup Steps	500

Table 5.5: Hyperparameters for observation encoder, transformer, noise scheduler, and training. N_m indicates the sensor modalities involved in the model training.

DDPM Noise Scheduler	
Num. Timesteps	30
β Range	$[0.0001, 0.2]$
β Schedule	Cosine
Transformer	
Input and Output Shape	$(T_p, 7)$
Observation Conditional Feature Length	$N_m \times 128 + 7$
Weight Precision	`fp16`
Num. Decoder Layer	8
Num. Head	4
Len. Embedding	512
Cartesian Impedance Controller	
Nullspace Stiffness	$\text{diag}([1, 1, 1, 1, 1, 1]^T)$
Task Frame Stiffness	$\text{diag}([1200, 1200, 1200, 50, 140, 50]^T)$
Cartesian Error Smoothing Factor	0.1

Table 5.6: Inference parameter configurations. diag indicates a diagonal matrix.

5.4 Evaluation Experiments

In this section, we introduce the experiments that are used to evaluate the performance and robustness of our imitation learning framework on solving the match-lighting task, and the experiments for analyzing latent space observation embedding w.r.t. different sensor modalities.

5.4.1 Evaluating Policy Performance on Match-Lighting Task

As the main experiment of this thesis, we evaluate the performance of our imitation learning policy with the experiment setup described in Sec. 4.1 across 3 match coverages i.e. {Short, Middle, Long}, and 6 match angles i.e. $\{-20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ, 30^\circ\}$ that results in 18 match pose situations. In addition, we perform an ablation study on 5 sensor combination options, i.e. {RealSense, GelSight Mini, Evetac}, {RealSense, GelSight Mini}, {RealSense, Evetac}, {GelSight Mini, Evetac}, {RealSense}. For each match pose, we perform 20 trials of the experiment and repeat with all sensor combinations.

5.4.2 Robustness Test

To test the robustness of the policy, we apply external perturbations during the task execution of each sensor combination to see if the policy can resume approaching the striker paper. The perturbations are:

- Force perturbations, i.e. apply an external force temporarily to the end-effector in the direction of the y-axis (refer to Figure 4.2.3).
- Visual perturbations, i.e. block the camera view of RealSense for a certain time.
- Tactile perturbations, i.e. apply an external force temporarily to the match tip.

It's worth noting that we only apply these perturbations during the task phase of **approaching the striker paper** before the initial contact.

5.4.3 Temporal Responsiveness Analysis

This experiment aims to analyze how fast each sensor modality can react to specific interactions, e.g. match initial contact, striking movements, etc.

We first introduce a metric that expresses the dynamic variations of latent space embedding that occur across the observation history T_o, i.e. temporal responsiveness λ. Assume an end-to-end trained observation encoder as mentioned in Sec.4.3 which maps the sensor image into a latent embedding $\boldsymbol{\xi}_t$:

$$\boldsymbol{\xi}_t = \{\xi_1 | \dots | \xi_{N_m}\} \tag{5.1}$$

where t indicates the observation time, ξ represents the embedding of each modality, $|$ is the concatenation operation, and N_m is the total number of the modalities. With the observation history length T_o, the temporal responsiveness $\lambda \in \mathbb{R}$ can be written as:

$$\lambda = \frac{\sum_{t=1}^{T_o} \left\| \boldsymbol{\xi}_t - \boldsymbol{\xi}_{t-1} \right\|_2}{T_o} \tag{5.2}$$

where $\|\cdot\|_2$ is the L2-norm.

We then select a successful trial with all sensor modalities i.e. {RealSense, GelSight Mini, Evetac}, and obtain the temporal responsiveness of each modality.

5.4.4 Match Pose Recovery Analysis

This analysis experiment aims to investigate whether the end-to-end trained observation encoder is capable enough to recover the match in-hand pose in latent space under different sensor modalities to further discriminate different poses. However, from a single latent observation sequence of a fixed match pose, it is difficult to discover whether the corresponding pose information is recovered in the embedding. Thus, we propose to compare the embedding across various match poses to express the differences.

With this purpose, for each match pose, we continuously collect a sequence of T_o images from all the sensor combinations, i.e. $\mathbf{o} = \{o_{t-T_o}, \dots, o_{t-1}, o_t\}$, while keeping the robot in the same position to prevent background change, especially for RealSense. For each modality, we only keep the latest image o_t and encode it into a latent space embedding ξ.

By concatenating all of the latent embeddings from all the modalities under this specific match pose, we obtain a matrix of embeddings $\boldsymbol{X} \in \mathbb{R}^{N_m \times l_{\text{emb}}}$:

$$\boldsymbol{X} = \{\xi_{m_0} | \ldots | \xi_{m_{N_m}}\} \tag{5.3}$$

where $\xi_{m_{(\cdot)}}$ is the latent embedding of the latest step encoded by sensor m, l_{emb} is the length of the embedding, and $|$ is the concatenation operation. For the visualization of the most essential information of this embedding matrix, we first apply Principal Component Analysis (PCA) to $\boldsymbol{X}$:

$$\boldsymbol{\sigma} = \text{PCA}(\boldsymbol{X}, N_{\text{PCA}}) \tag{5.4}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^{N_m \times N_{\text{PCA}}}$ is the principal component vector, and $N_{\text{PCA}} = 1$ is the number of principal components, and we only pick the most significant one.

Subsequently, we apply this operation for all match poses and obtain an intermediate matrix $\bar{\boldsymbol{X}}(\boldsymbol{\sigma}) \in \mathbb{R}^{(N_{\text{cov}} \times N_{\text{ang}}) \times (N_m \times N_{\text{PCA}})}$, with N_{cov} and N_{ang} representing the number of match coverage and angle variations, respectively.

We then flatten this matrix into a vector $\bar{\boldsymbol{X}}_{\text{flat}}(\boldsymbol{\sigma})$, and calculate the pairwise cosine similarity between all the elements to evaluate a new matrix $\mathbf{M} \in \mathbb{R}^{N_{\text{cov}} \times N_{\text{ang}}}$:

$$\mathbf{M}_{ij} = \frac{\bar{\boldsymbol{X}}_{\text{flat},i}(\boldsymbol{\sigma}) \bar{\boldsymbol{X}}_{\text{flat},j}(\boldsymbol{\sigma})}{\left\| \bar{\boldsymbol{X}}_{\text{flat},i}(\boldsymbol{\sigma}) \right\| \left\| \bar{\boldsymbol{X}}_{\text{flat},j}(\boldsymbol{\sigma}) \right\|}, \quad i, j \in [0, 1, \ldots, N_{\text{cov}} \times N_{\text{ang}}] \tag{5.5}$$

Finally, this matrix $\mathbf{M}$ is used to demonstrate the encoder's ability to discriminate and recover the match poses when trained with different sensor modalities.

6 Results and Discussion

In this chapter, we report the results of all the experiments introduced in Sec. 5.4 and provide corresponding discussions.

6.1 Policy Performance Evaluation on Match-Lighting Task

As described in Sec. 5.4.1, we evaluated the policy performance across 3 match coverages i.e. {Short, Middle, Long}, and 6 match angles i.e. $\{-20°, -10°, 0°, 10°, 20°, 30°\}$ that results in 18 match pose situations. Moreover, we perform an ablation study on 5 sensor combination options, i.e. {RealSense, GelSight Mini, Evetac}, {RealSense, GelSight Mini}, {RealSense, Evetac}, {GelSight Mini, Evetac}, {RealSense}. It's worth noting that although the match poses used to evaluate the policy performance are the same as the dataset, they are actually different ones due to uncontrollable grasping errors and different match dimensions. More specifically, for each match grasp, match coverage has an average error of ± 6 mm and the match angle has an average error of $\pm 5°$. Therefore, the match poses that occurred in the experiments are generally distinct from the cases in the training dataset.

For each match pose, we conduct 20 trials and repeat with all sensor combinations. Besides, for a simpler expression, in the following sections, we use the tuple containing the match angle and coverage to represent the match pose, e.g. (Long, $20°$), and we also use GelSight to replace GelSight Mini.

6.1.1 Policy Performance Upon Different Sensor Combinations

Figure 6.1 presents the average success rates of the 5 different sensor combinations over all the 18 match poses. The pure tactile combination i.e. {GelSight, Evetac} achieved the

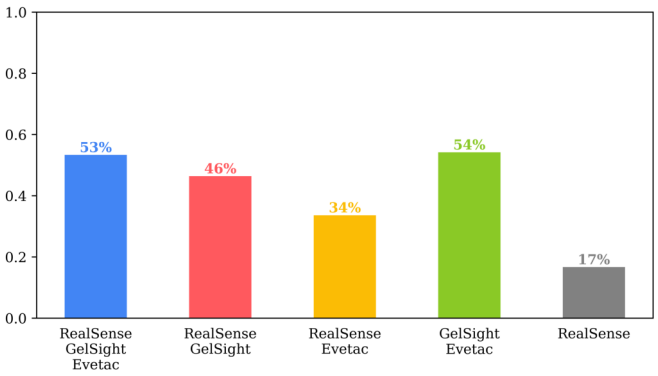


Figure 6.1: Average success rates of different sensor combinations over all 18 match poses, the y-axis indicates the success rate.

best results with 54%, while the vision-only combination i.e. {RealSense} has the lowest success rate of 17%. The full-sensor combination i.e. {RealSense, GelSight, Evetac} results in a strong performance of 53%, closely following the best performance. For the vision & tactile type of combinations, {RealSense, GelSight} yields 46% which is 12% higher than {RealSense, Evetac}. It turns out that the policies trained with combinations involving tactile modality generally outperform the vision-only policy, demonstrating that tactile information can make a significant contribution to improving manipulation performance.

For more details, Figure 6.2 shows the complete evaluation results of the policy performance over all the match poses w.r.t. sensor combinations.

For the policy trained with only vision i.e. {RealSense}, its best performance can only reach a success rate of 40% in the match pose of (Short, 10°). From the experiments, we observed that the most frequent failure mode is due to early strikes without even making initial contact with the striker paper, i.e. the robot reached above the striker paper and then proceeded to execute a strike movement directly. Other than that, it was also observed that the policy applied too much force on the contact and broke the match during striking.

When incorporated with tactile observation, i.e. GelSight and Evetac, the policy performance is enhanced in different aspects. In the case of {RealSense, Evetac}, the performance for all match coverages is greatly increased, especially for the angles of 10° and 20°, the success rates are all above 75%. We observed that in many trials, the initial contact can be detected quickly, followed by smooth strikes, indicating that the event-based tactile

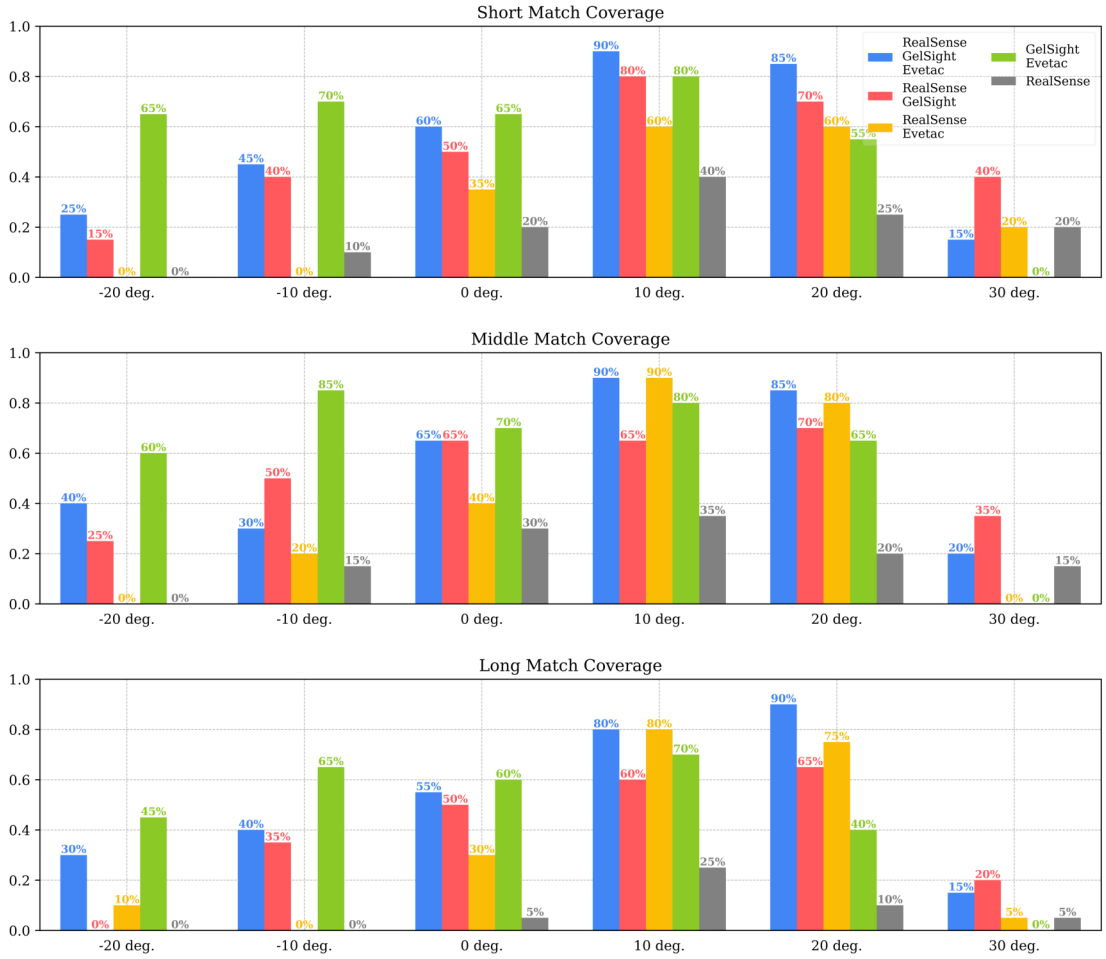


Figure 6.2: Evaluation results with y-axis as success rate. The three plots correspond to different match coverages, with each plot comparing policy performance over different angles.

signal might potentially enhance the contact awareness of the policy. However, with this combination, the policy is still unable to generalize to a wide range of angles, e.g. in negative angles of $-20°$, and $-10°$. Nevertheless, if the contact area was incorrectly selected, the subsequent strikes were also executed in the wrong direction immediately and resulted in a failure case. In contrast, when the policy is trained with the combination of {RealSense, GelSight}, although the best performance of 80% was not exceptionally high, the range of feasible match poses is expanded, especially in the negative match angles. We observed even a 50% success rate in the match pose of (Middle, $-10°$). However, the policy often stagnated for a long time when preliminary contact was established, and eventually resulted in a failed trial. Similar stuck cases are also observed in [8]. In summary, with the match pose information provided by GelSight, the policy might be able to better adjust the end-effector pose accordingly to enlarge generalization on extreme match angles.

Combining all three sensors, i.e. {RealSense, GelSight, Evetac}, our policy achieved the best performance of 90% in multiple match poses i.e. (Short, $10°$), (Middle, $10°$), and (Long, $20°$). With this combination, the correct contact area can be identified and proper end-effector pose adjustments can be achieved, resulting in a wide generalization of all match poses. Although this policy satisfied all the requirements for a successful strike, the stuck problem still exists but is greatly reduced in comparison with {RealSense, GelSight}. Moreover, during the execution, the policy might generate some redundant movements, such as a fast pull-back after striking, which results in a decrease in smoothness and completeness of task execution. Based on this behavior, we observed a special ability of the policy trained with this combination: in the situation that the policy initiates a strike without initial contact, it will promptly cease and perform a rapid retraction. Subsequently, it brings the end-effector slightly towards the striker paper and performs a strike again. If still no contacts appear, the policy will repeat the similar behavior until the contact is established, and eventually, it will perform a regular long strike to ignite the match.

When the policy is only observing both tactile modalities, i.e. the combination of {GelSight, Evetac}, its performance remains competitive in almost all match poses. Although its best performance of 85% is lower than e.g. the full-sensor combination (90%), this policy achieved remarkably high performance on negative angles, i.e. almost all the results are exceeding 60%. However, with the angle of $30°$ of all the match coverage, the policy always produces an over-rotated angle in the yaw-axis that leads to a zero success rate. Apart from this, one of the most significant improvements of this combination is the elimination of stuck behavior. Furthermore, the striking movement is more fluent than the vision-involved combinations. Other instances of failure exhibit a similar pattern, i.e.

the end-effector pose was not correctly adjusted or just reached outside the striker paper plate.

6.1.2 Policy Performance upon Match Poses

To gain a deeper statistical insight into the results of Figure 6.2, we provide Figure 6.3 that shows the distribution of policy performance over different match coverages at fixed angles and vice versa.

From the top figure, the largest performance change upon match coverage is 25%, while the smallest is 5% while in the bottom plot, the policy performance changes drastically w.r.t. match angles ranging from 15% to 90%, indicating the policy performance is more sensitive to the variations in match angles than coverages.

Looking at the top figure, the performance of nearly all combinations demonstrates reaching the best performance in the angle of 10° with 69% and dropping gradually as the angles vary to the negative (end with 20% at -20°) or positive direction (end with 14% at 30°). This trend suggests that policies are better suited for positive match angles, while their ability to accurately handle extreme angles is limited. The exception is the {GelSight, Evetac} combination, which shows more consistent performance across a wider range of angles, indicating a stronger capability for handling diverse match angle variations in the match-lighting task.

Furthermore, in the bottom figure, the performance of {GelSight, Evetac} and {RealSense} has the relatively smaller average range of variation (26%, 30%), in comparison to other three modalities ({RealSense, GelSight, Evetac} with 73%, {RealSense, GelSight} with 78%, and {RealSense, Evetac} with 76%). It turns out that when combining both tactile and vision modalities, the policy performance might exhibit inconsistency across different match angles, potentially due to a vision-tactile mismatch during the task execution.

6.2 Robustness Test of Match-Lighting Policy

In this experiment, we constrain the perturbations to certain magnitudes, e.g. for force perturbation, we apply a gentle push to the end-effector; for visual perturbation, we block to the camera view for 1 to 3 seconds; for tactile perturbation, we apply a quickly touch on the match tip within 1 second.

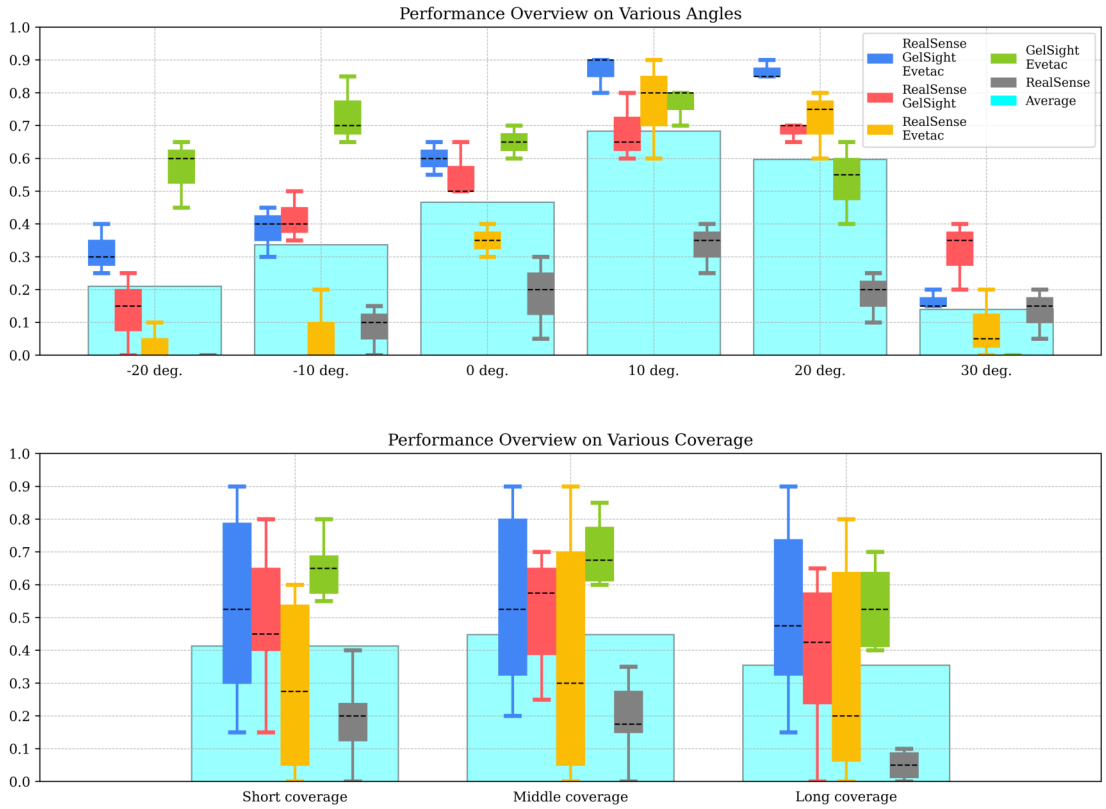


Figure 6.3: Results analysis over same match angle and coverage with y-axis as success rate. The top plot shows the distribution of policy performance over different coverages at fixed angles, and the bottom plot vice versa. The light blue bars are the average success rate across all sensor combinations under each match pose.

Combination	Force Perturb.	Visual Perturb.	Tactile Perturb.
RealSense, GelSight, Evetac	Robust	Robust	Robust
RealSense, GelSight	Robust	Robust	Robust
RealSense, Evetac	Robust	Robust	Not Robust
GelSight, Evetac	Robust	N/A	Not Robust
RealSense	Robust	Robust	N/A

Table 6.1: Policy robustness against various external perturbations.

The results are reported in Table 6.1. It turns out that all of the policies are robust to external force perturbations. With visual perturbations, all the policies that include RealSense tend to be robust against short visual occlusion before getting close to the striker paper. However, for the policies involved with tactile modalities, only the combination of {RealSense, GelSight, Evetac} and {RealSense, GelSight} exhibited robustness against unexpected tactile disturbs. Especially, for the full-sensor combination, i.e. {RealSense, GelSight, Evetac}, when applied with force on the match tip, the policy would try to strike but stopped shortly after, and conducted the pull-back behavior described in Sec. 6.1.1, then continued approaching the striker paper. For the rest two tactile combinations, i.e. {RealSense, Evetac} and {GelSight, Evetac}, the strike movement was immediately triggered when applied force on the match tip. It appears that the tactile features extracted from Evetac could cause overreactions to the policy, leading to sub-optimal decisions.

6.3 Temporal Responsiveness

For this experiment, we calculate the temporal responsiveness from a successful match lighting trail with the sensor combination of {RealSense, GelSight, Evetac}, presented in Figure 6.4. According to the results, Evetac has the lowest latency regarding the occurrence of contact and the largest magnitude for detecting a change in contact state due to the event-based working principle. Beyond that, the contact feedback remains active during the whole striking movement. These fast and strong responsive features could be considered as one of the reasons why the Evetac-involved policy can perform a strike shortly after contact establishment. However, despite sharp and swift contact

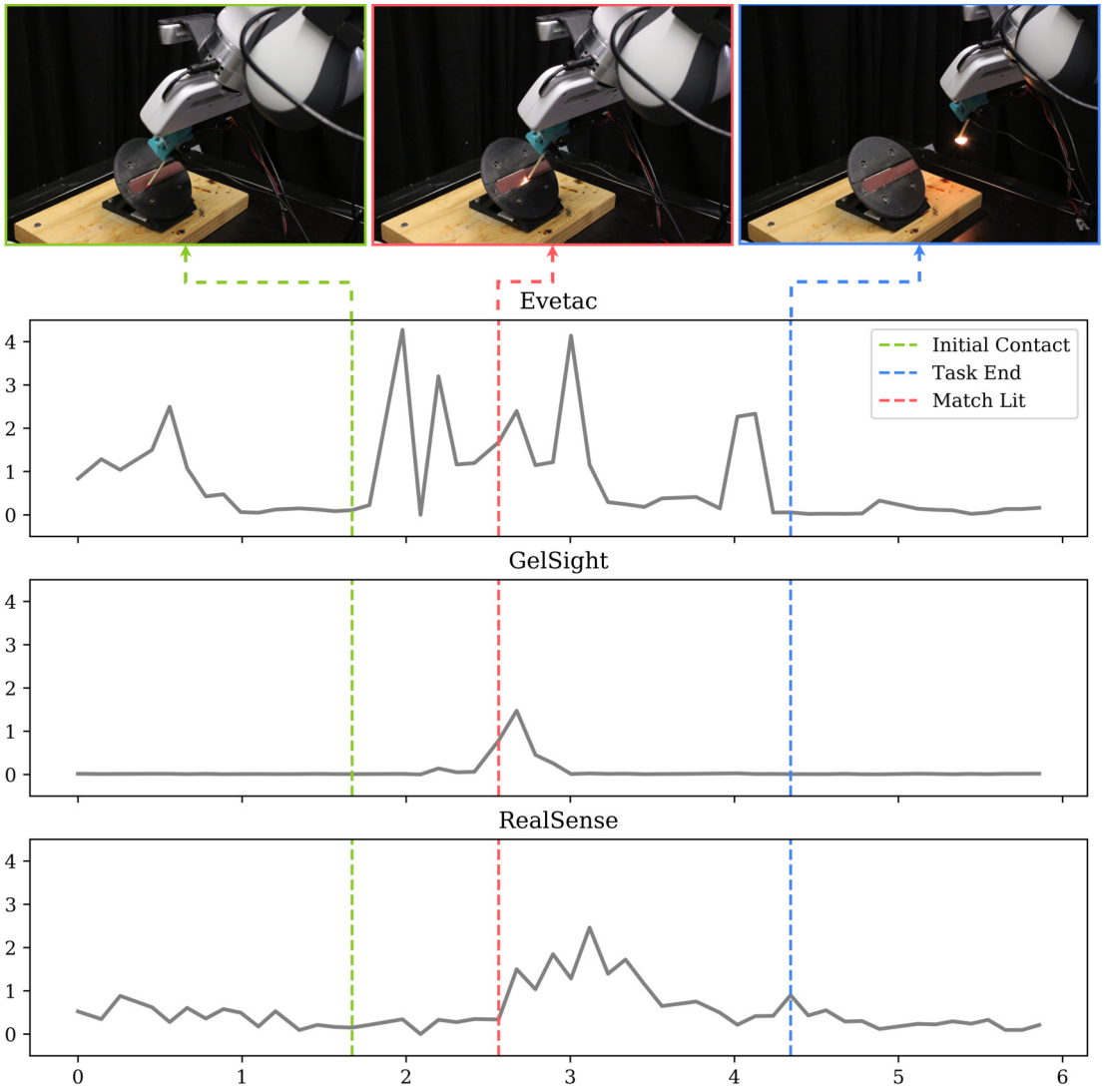


Figure 6.4: Dynamic temporal responsiveness λ of different sensors in terms of the encoded latent embeddings. The x-axis is time and the y-axis is the temporal responsiveness λ. The dotted line and corresponding snapshots indicate three key time points of the task execution, i.e. initial contact (green), match-lit (red), and task finished (blue).

detections, Evetac suffers from over-sensitive triggered events caused by tiny inertial displacements or mechanical vibrations of the match during movement, e.g. the unusual spike in the first row of Figure 6.4 at 0.5 and 4 second where no contact happened. This might result in the proneness of the policy to tactile perturbation and cause early strikes when the sensor combination involves Evetac.

Followed by GelSight, the contact is detected with a tiny signal spike and only lasts for a short moment at the beginning of the strike (i.e. around $2.1\,\mathrm{s}$) since during the strike the tactile images are mostly the same except for initial contact establishment. This might result in an unclear contact status feedback to the policy, leading to delayed decision-making for a strike movement and causing stagnation, as described in Sec. 6.1.1.

For RealSense, we are not able to directly confirm whether the responsiveness change of this sensor is caused by contact phenomena or visual cues. However, the continuous response due to the fast shift of visual cues w.r.t. the fire light is obvious to identify. During other task phases, the various visual cues also introduce noise to the embedding as small spikes suggest. For sensor combinations that include both RealSense and GelSight, the sensor noise and weak contact detection might lead to a performance decrease or redundant behaviors.

6.4 Match Pose Recovery

As introduced in Sec. 5.4.4, we sample the latent embeddings of each sensor combination and investigate the ability of the observation encoder to recover and discriminate the static feature of match poses.

The results are illustrated in Figure 6.5. Each heatmap demonstrates the pairwise similarity of the multimodal latent embedding between different match poses, brighter (in color white) pixels indicate a higher similarity while darker (in color blue) pixels indicate a lower similarity. Observing the overall brightness of the heatmaps, { GelSight, Evetac, RealSense } and { GelSight, Evetac } are darker in comparison to the other three combinations. This reveals a better ability of the encoder, because the darker the pixels, the lower the similarity between the latent embeddings of both match poses, which could indicate that the encoder is more capable of recovering the accurate match pose in latent space.

Recall our results in Sec. 6.1.1, we find that the policy with a better generalization on match poses corresponds to the sensor combination with a darker heatmap here. Apart from that, we also noticed that the encoders of almost all the combinations tend to extract

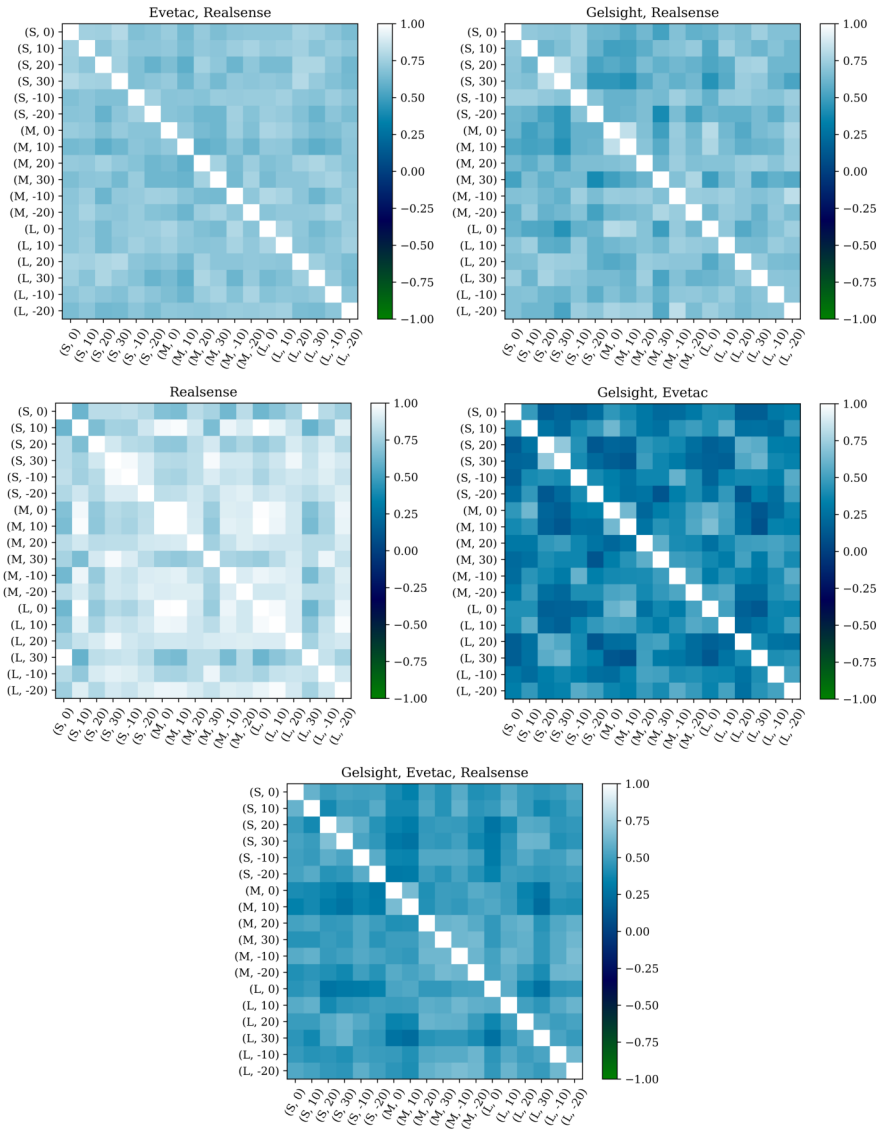


Figure 6.5: Static feature similarity spectrum of each sensor combination in terms of match poses. The x-axis and y-axis are the match poses with S, M, and L indicating Short, Middle, and Long match coverage. The pixel value of each heatmap is the pairwise cosine similarity of multimodal observation embedding. Pixels with darker blue indicate a lower similarity and whiter pixels indicate a higher similarity.

more information from the short and middle match coverage, which is consistent with the results in Figure 6.3.

In summary, the latent embedding similarity analysis provided evidence that the observation encoder trained with different sensor modality combinations may influence the performance of the match-lighting policy. However, the underlying factor that substantially affects the conditional denoising process and results in biased actions remains uninvestigated. Many recent works have also pointed out the difficulties of interpreting diffusion models, as the latent transformations in intermediate steps have no specific semantic meaning. This makes it difficult to attribute contributions to the overall content of the generation [63–66]. Therefore we leave the further advanced analysis for our future work.

7 Conclusion

In this thesis, we investigated the importance of tactile sensing in solving a challenging robotic manipulation task, i.e. lighting up a match. This task requires both precise contact status awareness and match in-hand features e.g. static pose and dynamic movements. Besides, proper contact-force control is also necessary since the match is quite brittle and fragile, any excessive force will result in breakage. To fulfill the task requirements, we proposed to learn the skills directly from human demonstrations through an imitation learning framework that's mainly based on diffusion policy and a Cartesian impedance controller. The action sequence prediction of diffusion policy guaranteed a fast and consistent trajectory generation, while the Cartesian impedance controller can regulate the robot behavior compliantly. In addition, we involved two types of vision-based tactile sensors, i.e. GelSight Mini, which captures global tactile features with whole RGB frames, and Evetac, which only captures dynamic tactile features with a rather low latency. Apart from that, by leveraging the multimodal input support of the diffusion policy, we are able to alter the combination of sensor observation modalities to investigate the performance change of the policy.

We extensively evaluated our framework with multiple task configurations and sensor combinations. It turns out that our policy is capable of lighting up a match with various task configurations and is robust against certain perturbations. From the results, we also find out that sensor combinations significantly influence the performance of the policy. Despite involving all the sensors results in the best performance in certain match poses and the widest range of feasible match angle, combining only the two types of tactile sensors can also achieve remarkable success rates. The vision & tactile combinations demonstrate performance highlights from different perspectives, while the policy trained with single vision modality yields the worst performance. Afterwards, we further discussed the underlying causalities between the sensor combinations and policy performance by investigating both static and dynamic features from latent observation embeddings.

Eventually, the evaluation results indicate that tactile sensing is a crucial sense modality for solving the match-lighting task with high success rates.

8 Limitations and Future Work

Although we have demonstrated the match can be successfully lit up with all kinds of match poses and sensors, there are still limitations that remain for future work.

The first of our concerns is the inference speed of our policy. Our policy typically runs at 10 Hz with 30 steps of denoising and `ONNX` precision downgrade as introduced in Sec. 5.3.2. This low frequency is due to the enormous calculation of the transformer-based diffusion policy. In our future work, we tend to introduce new lightweight and efficient backbone model structures for the diffusion policy that could accelerate the inference speed.

Secondly, beside model capacity, the data source is the key aspect of improving the model generalization ability. However, human demonstrations are expensive to collect, whether through kinesthetic teaching or teleoperation. Take our task as an example, a dataset of 108 demonstrations costs already about 10 hours of work. In our future work, we will try to investigate how to extract human demonstrations from existing videos, which could significantly reduce physical effort and increase the number of data. On the other hand, for contact-rich tasks, it's also challenging to figure out how to extract tactile skills from videos.

Moreover, the current training pipeline of our framework is limited to a single task. We are also very interested how to blend the trained individual policies to achieve more sophisticated manipulations.

Last but not least, as time goes by, the dataset can be outdated eventually, but re-training or fine-tuning the policy with a new dataset might eliminate some key features that are still useful. In our future work, we also tend to investigate how to maintain the trained policy continually.

Bibliography

[1] Akihiko Yamaguchi and Christopher G Atkeson. Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision? *Advanced Robotics*, 33(14):661–673, 2019.

[2] Pedro Silva Girão, Pedro Miguel Pinto Ramos, Octavian Postolache, and José Miguel Dias Pereira. Tactile sensors for robotic applications. *Measurement*, 46 (3):1257–1271, 2013.

[3] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5633–5640. IEEE, 2020.

[4] Yuri Gloumakov, Tae Myung Huh, and Hannah S Stuart. Fast in-hand slip control on unfeatured objects with programmable tactile sensing. *IEEE Robotics and Automation Letters*, 2024.

[5] N. Funk, E. Helmut, G. Chalvatzaki, R. Calandra, and J. Peters. Evetac: An event-based optical tactile sensor for robotic manipulation. submitted. URL `https://www.ias.informatik.tu-darmstadt.de/uploads/Team/NiklasFunk/evetac_paper.pdf`.

[6] Roland S. Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10:345–359, 2009. URL `https://api.semanticscholar.org/CorpusID:17298704`.

[7] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

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

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

[10] Tarik Kelestemur, Robert Platt, and Taskin Padir. Tactile pose estimation and policy learning for unknown object manipulation. *arXiv preprint arXiv:2203.10685*, 2022.

[11] Danny Driess, Peter Englert, and Marc Toussaint. Active learning with query paths for tactile object shape exploration. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 65–72. IEEE, 2017.

[12] Alina Böhm, Tim Schneider, Boris Belousov, Alap Kshirsagar, Lisa Lin, Katja Doerschner, Knut Drewing, Constantin A Rothkopf, and Jan Peters. What matters for active texture recognition with vision-based tactile sensors. *arXiv preprint arXiv:2403.13701*, 2024.

[13] Yazhan Zhang, Guanlan Zhang, Yipai Du, and Michael Yu Wang. Vtacarm. a vision-based tactile sensing augmented robotic arm with application to human-robot interaction. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 35–42. IEEE, 2020.

[14] Lac Van Duong and Van Anh Ho. Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation. *IEEE Transactions on Robotics*, 37(2): 390–403, 2021. doi: 10.1109/TRO.2020.3031251.

[15] Ian H Taylor, Siyuan Dong, and Alberto Rodriguez. Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787. IEEE, 2022.

[16] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413.

[17] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[18] Visak Kumar, Tucker Hermans, Dieter Fox, Stan Birchfield, and Jonathan Tremblay. Contextual reinforcement learning of visuo-tactile multi-fingered grasping policies. *arXiv preprint arXiv:1911.09233*, 2019.

[19] Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. See to touch: Learning tactile dexterity through visual incentives, 2023.

[20] Zhenjun Yu, Wenqiang Xu, Siqiong Yao, Jieji Ren, Tutian Tang, Yutong Li, Guoying Gu, and Cewu Lu. Precise robotic needle-threading with tactile perception and reinforcement learning. In *Conference on Robot Learning*, pages 3266–3276. PMLR, 2023.

[21] Yijiong Lin, Alex Church, Max Yang, Haoran Li, John Lloyd, Dandan Zhang, and Nathan F. Lepora. Bi-touch: Bimanual tactile manipulation with sim-to-real deep reinforcement learning. *IEEE Robotics and Automation Letters*, 8(9):5472–5479, September 2023. ISSN 2377-3774. doi: 10.1109/lra.2023.3295991. URL http://dx.doi.org/10.1109/LRA.2023.3295991.

[22] Johanna Hansen, Francois Hogan, Dmitriy Rivkin, David Meger, Michael Jenkin, and Gregory Dudek. Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8298–8304. IEEE, 2022.

[23] Alex Church, John Lloyd, Raia Hadsell, and Nathan F. Lepora. Tactile sim-to-real policy transfer via real-to-sim image translation, 2021.

[24] Yevgen Chebotar, Karol Hausman, Zhe Su, Gaurav S Sukhatme, and Stefan Schaal. Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1960–1966. IEEE, 2016.

[25] Y. Zheng, F.F. Veiga, J. Peters, and V.J. Santos. Autonomous learning of page flipping movements via tactile feedback. (5):2734 – 2749, 2022. URL https://ieeexplore.ieee.org/document/9786532.

[26] Auke Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. *Advances in neural information processing systems*, 15, 2002.

[27] Jens Kober and Jan Peters. Learning motor primitives for robotics. In *2009 IEEE International Conference on Robotics and Automation*, pages 2112–2118. IEEE, 2009.

[28] Sylvain Calinon, Zhibin Li, Tohid Alizadeh, Nikos G Tsagarakis, and Darwin G Caldwell. Statistical dynamical systems for skills acquisition in humanoids. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 323–329. IEEE, 2012.

[29] Nutan Chen, Justin Bayer, Sebastian Urban, and Patrick Van Der Smagt. Efficient movement representation by embedding dynamic movement primitives in deep autoencoders. In *2015 IEEE-RAS 15th international conference on humanoid robots (Humanoids)*, pages 434–440. IEEE, 2015.

[30] Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023.

[31] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[32] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

[33] Yuki Shirai, Devesh K. Jha, Arvind U. Raghunathan, and Dennis Hong. Tactile tool manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2023. doi: 10.1109/icra48891.2023.10160480. URL `http://dx.doi.org/10.1109/ICRA48891.2023.10160480`.

[34] Stephen Tian, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. Manipulation by feel: Touch-based control with deep predictive models. *2019 International Conference on Robotics and Automation (ICRA)*, pages 818–824, 2019. URL `https://api.semanticscholar.org/CorpusID:73728711`.

[35] Achu Wilson, Helen Jiang, Wenzhao Lian, and Wenzhen Yuan. Cable routing and assembly using tactile-driven motion primitives. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10408–10414, 2023. URL `https://api.semanticscholar.org/CorpusID:257636819`.

[36] Thomas Bi, Carmelo Sferrazza, and Raffaello D'Andrea. Zero-shot sim-to-real transfer of tactile control policies for aggressive swing-up manipulation. *IEEE Robotics and Automation Letters*, 6:5761–5768, 2021. URL `https://api.semanticscholar.org/CorpusID:230799584`.

[37] Abraham George, Selam Gano, Pranav Katragadda, and Amir Barati Farimani. Visuo-tactile pretraining for cable plugging. *arXiv preprint arXiv:2403.11898*, 2024.

[38] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3(1):297–330, 2020.

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

[40] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

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

[45] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[46] Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[47] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 354023957X.

[48] Neville Hogan. Impedance control: An approach to manipulation: Part ii—implementation. 1985.

[49] Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jähne, Lukas Hausperger, and Simon Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64, 2022.

[50] Alin Albu-Schaffer, Christian Ott, Udo Frese, and Gerd Hirzinger. Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms. In *2003 IEEE International conference on robotics and automation (Cat. No. 03CH37422)*, volume 3, pages 3704–3709. IEEE, 2003.

[51] Alexander Dietrich, Christian Ott, and Alin Albu-Schäffer. An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 34(11):1385–1400, 2015.

[52] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[53] Klas Kronander and Aude Billard. Online learning of varying stiffness through physical human-robot interaction. In *2012 IEEE International Conference on Robotics and Automation*, pages 1842–1849, 2012. doi: 10.1109/ICRA.2012.6224877.

[54] Nadia Figueroa, Bilkit Githinji, and Shen Li. Franka interactive controllers: A repository of controllers for franka robots. `https://github.com/nbfigueroa/franka_interactive_controllers`, 2022.

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

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. URL `http://arxiv.org/abs/1502.01852`.

[57] Franka Emika GmbH. libfranka: C++ library for franka robotics research robots. `https://github.com/frankaemika/libfranka`, 2024. Accessed: 2024-08-13.

[58] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL `http://arxiv.org/abs/1912.01703`.

[60] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[61] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

[62] ONNX Runtime developers. Onnx runtime. `https://onnxruntime.ai/`, 2021. Version: x.y.z.

[63] Ji-Hoon Park, Yeong-Joon Ju, and Seong-Whan Lee. Explaining generative diffusion models via visual analysis for interpretable decision-making process. *Expert Systems with Applications*, 248:123231, 2024.

[64] Hila Chefer, Oran Lang, Mor Geva, Volodymyr Polosukhin, Assaf Shocher, Michal Irani, Inbar Mosseri, and Lior Wolf. The hidden language of diffusion models. *arXiv preprint arXiv:2306.00966*, 2023.

[65] Eslam Mohamed Bakr, Liangbing Zhao, Vincent Tao Hu, Matthieu Cord, Patrick Perez, and Mohamed Elhoseiny. Toddlerdiffusion: Flash interpretable controllable diffusion model. *arXiv preprint arXiv:2311.14542*, 2023.

[66] Kamil Deja, Anna Kuzina, Tomasz Trzcinski, and Jakub Tomczak. On analyzing generative and denoising capabilities of diffusion-based deep generative models. *Advances in Neural Information Processing Systems*, 35:26218–26229, 2022.