Task-Context-Aware Diffusion Policy with Language Guidance for Multi-task Disassembly

Jeon Ho Kang[1], Sagar Joshi[1], Neel Dhanaraj[1], and Satyandra K. Gupta[1]

***Abstract*— Diffusion-based policy learning frameworks excel in learning diverse tasks and achieving high success rates. However, in manufacturing settings, success rate alone is insufficient for real-world deployment. Tasks must be executed efficiently, minimizing idle time while maintaining precision. Additionally, in assembly and disassembly settings, a single scene often contains multiple task goals that need to be completed—such as picking up an engine while simultaneously securing a suspension—requiring the robot to reason over multiple objectives within the same observation space. In human-robot collaboration, enabling humans to specify task preferences is crucial for flexible and intuitive interaction. In this paper, we address two key challenges: (1) improving task execution efficiency by structuring tasks into distinct sub-task modes via language, and (2) enabling human operators to select tasks using natural language commands. Additionally, we introduce adaptive parameter selection framework and reliance on different sensory modalities depending on these sub-task modes. We evaluate our approach on the NIST Task Board, a representative benchmark of real-world tasks where multiple task goals exist within the same scene. Our method improves execution speed by 57% and show 19% improvement in task success rates. Demonstration videos are available at `https://rros-lab.github.io/task-aware-diffusion/`.**

I. Introduction

For high-mix manufacturing applications, such as disassembly for maintenance or recycling, automation remains challenging due to the high variability of parts, precise and force-sensitive manipulations, and frequently changing task requirements. These settings, however, are well-suited for human-robot collaboration, where robots can handle repetitive physical operations—such as unplugging connectors and extracting components—while humans can provide oversight, perform inspections, or dynamically determine disassembly sequences.

Recent advances in imitation learning, particularly diffusion policies, have provided a promising approach to enable robots to perform complex manipulations—such as battery cell removal and enclosure disassembly—directly from multi-modal human demonstrations (e.g., visual, force/torque). These developments have brought collaborative robots closer to practical deployment in dynamic, high-mix manufacturing environments.

Despite recent successes, several drawbacks hinder the adoption of diffusion policies for multi-task disassembly problems. Typically, a diffusion policy is implemented as an end-to-end model, trained on extensive datasets of high-quality demonstrations, that directly maps raw sensory inputs to low-level actuator commands. While this approach avoids explicitly specifying individual tasks (e.g., removing a particular component) and their associated sub-tasks (e.g., approaching or grasping connectors), it introduces certain limitations. Empirically, we found performance degradation when demonstrations are suboptimal, particularly during critical transitions between sub-tasks. Furthermore, end-to-end policies often struggle when multiple tasks are simultaneously visible, as illustrated in Figure 1, causing ambiguity and task-selection confusion. Although training separate specialized diffusion models for each task could enhance reliability, this approach rapidly becomes impractical as the number of tasks increases due to increasing demonstration acquisition costs. Additionally, human operators frequently exhibit context-dependent preferences, prioritizing certain tasks based on specific assembly scenarios. To effectively capture preferences and clearly specify intended tasks, diffusion policies would need to accept flexible task sequences provided by human operators.

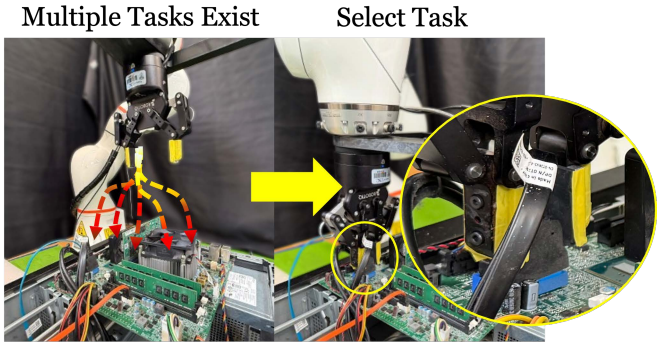

Fig. 1: Real-world manufacturing environments present robots with complex scenes containing multiple potential tasks, such as removing RAM, SATA ports, power connectors, and more (left). In these cases, the robot must be able to commit to a single task based on external input, such as language. We demonstrate an example where the robot selects the task of removing a wire from the motherboard on the desktop (right).

To address these challenges, we propose training a single diffusion policy capable of handling a family of related manipulation tasks, particularly those involving common task types (e.g., unplugging various connectors) or similar scenes (e.g., controller diagnostics). To clearly guide the robot through specific tasks and associated sub-tasks, we condition the diffusion policy on natural language instructions alongside visual and force inputs. Our key insight is that many manufacturing applications inherently involve multiple similar tasks, such as removing connectors of type A, B, or C, each composed of intuitive sub-tasks like approaching, grasping, rotating, and pulling. By explicitly incorporating task and sub-task information, a single diffusion policy can implicitly capture and leverage this underlying task struc-

[1]University of Southern California, Los Angeles, CA, USA.

ture. This approach enables effective operation in multi-task scenarios and provides human operators with the flexibility to dynamically specify or adjust task sequences during deployment. Furthermore, conditioning on sub-task information facilitates precise signaling of task transitions, significantly improving robotic system efficiency and adaptability in multi-task manufacturing settings. Our contributions are the following

1) We introduce a language-conditioned Diffusion Policy framework for manufacturing disassembly, enabling task selection within a single multi-task model.
2) We propose a task-context-aware, end-to-end policy learning approach that enhances data efficiency and reduces task execution time.
3) We propose a novel method for adaptively weighting different sensory modalities based on the specific sub-task the robot is performing, allowing dynamic reliance on relevant modalities at different stages.
4) We validate the effectiveness of our approach through experimental results and highlight the benefits of employing multi-task diffusion models for complex manufacturing tasks.

II. Related Works

Behavior Cloning: Behavior cloning is an imitation learning approach where a policy is learned by directly mimicking an expert's state-action pairs [1]. It has demonstrated remarkable potential across various real-world robotic manipulation tasks [2]–[7]. Transformer-based methods, such as Behavior Transformers [8], have notably advanced sequential decision-making by capturing long-range dependencies and the multi-modal nature of expert demonstrations, mitigating the covariate shift inherent in traditional behavior cloning. Recent implicit policy models [9]–[11] improve action representation, enabling smoother transitions and greater robustness in task execution. Although these methods highlight the ability of behavior cloning to simplify policy learning, they also raise concerns about generalization to novel objects and multi-modal tasks.

Diffusion Policy: Diffusion models are generative models that learn to gradually denoise random noise into coherent data samples [12], [13], and they have recently been applied to robot policy generation. [14]–[16] obtained remarkable performance by employing diffusion models to learn robot trajectory and visuomotor policies for robotic manipulation tasks. [17] further extended this line of work by incorporating force feedback through a cross-attention mechanism, enhancing robustness in contact-rich tasks. Additionally, diffusion policies have demonstrated state-of-the-art performance in various robotic manipulation tasks, including object reorientation, grasp synthesis, and precise tactile manipulation [18]–[23]. Beyond manipulation, diffusion models have effectively been used in imitation learning for general purpose agents [24], [25]. However, many of these existing frameworks have long task execution times, with a primary focus on task success rather than execution efficiency. In this paper, we aim to address this gap.

Language Guided Policy: Integrating natural language into policy learning enables robots to follow human instructions and perform a variety of specified tasks in words, such as using LLMs for higher-level decision-making and reasoning in robotics [26]–[30]. Recent transformer-based generalist robot policies have demonstrated enhanced ability to interpret diverse natural language instructions by aligning visual inputs and language-conditioned action primitives, significantly improving generalization across manipulation tasks described via language [31]. LLMs have also found applications in policy learning [32], [33], where they facilitate the translation of natural language task descriptions into reward functions and control objectives. Recent advances in language-guided policy learning leverage large language models and vision-language integration to bridge high-level task descriptions and low-level control [34]–[36]. For example, DISCO [37] employs vision-language-model-generated keyframes to condition diffusion policies, while other works demonstrate scalable skill acquisition through language-guided planning [38].Furthermore,such methods have inspired natural language-guided human-robot collaboration (HRC) applications [39], [40].

In manufacturing, the required tasks are relatively limited compared to home applications. However, the ability to scale to multiple task sets remains crucial. Therefore, we propose a language-guided policy, following the approach in [37], but specifically demonstrate its effectiveness in manufacturing disassembly tasks on a real robot.

III. Preliminaries

A. Problem Formulation

We aim to learn a multi-task policy π conditioned on the observation $O_t = \{I, F, l\}$ where I represents visual input, F denotes force measurements, and l is a language-based descriptor specifying the task. The language input $l \in L$ serves as a conditioning variable, informing the model about the current task and its sub-tasks. Let T_p be the parent task, which consists of a set of sub-tasks: $T_p = \{T_{s_i}\}_{i=1}^{n}$. Our objective is to learn a policy π that generates a sequence of actions $a_t \sim \pi(a|O_t)$ for real-world disassembly tasks. The learned policy should minimize task execution time while incorporating human preferences in task selection.

B. Diffusion Models

Diffusion models are probabilistic generative models that approximate the data distribution $p(x_0)$ by introducing a sequence of latent variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, which are progressively noisier versions of the clean data x_0. The forward process assumes a Gaussian distribution, with the mean centered around a scaled version of x_0.

$$\mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{1}$$

where $\bar{\alpha}_t$ is the cumulative product of derived variance schedule α_t at t diffusion time step. The forward process follows the Markov chain of the form

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{2}$$

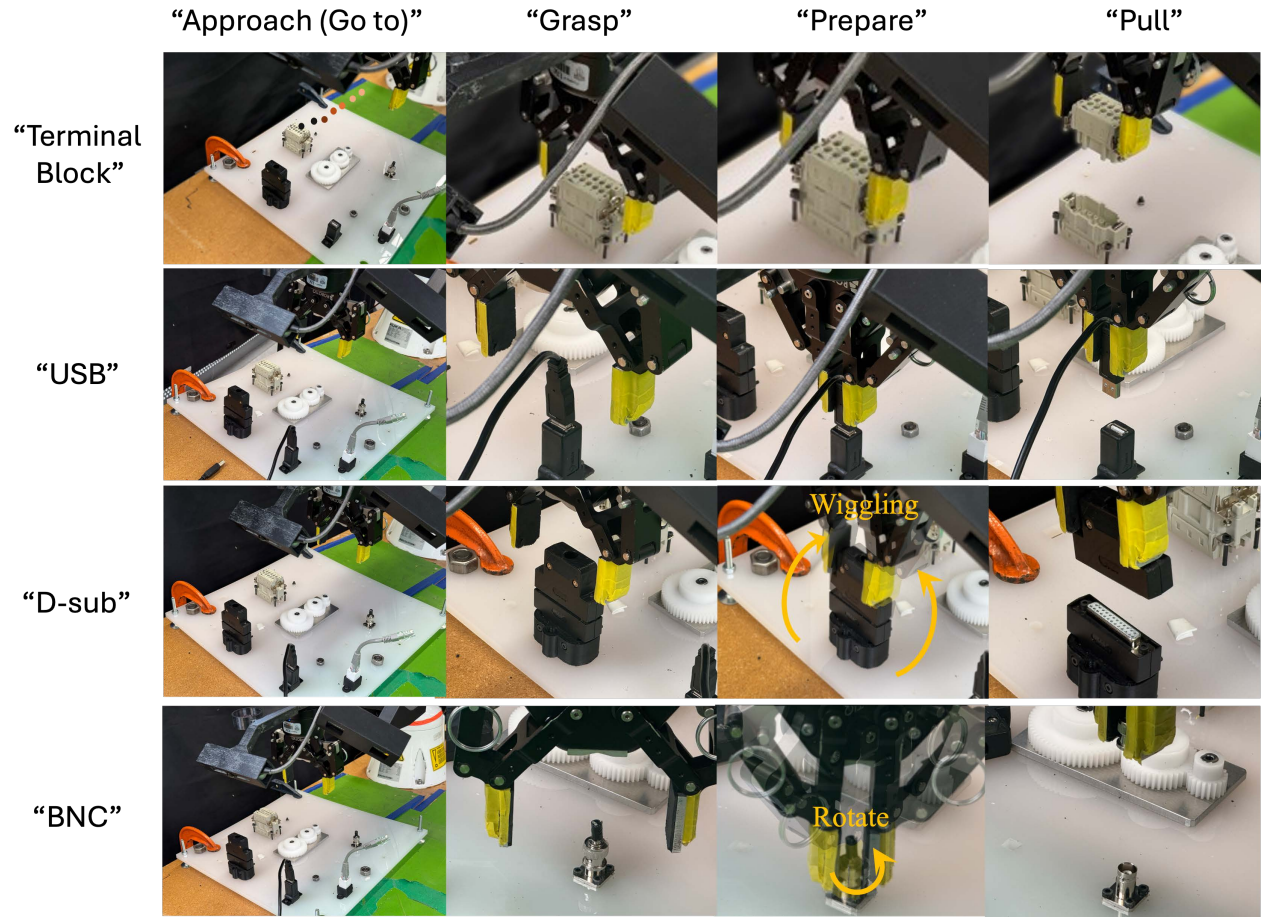

Fig. 2: **Task Description in Text**: We illustrate a sequence of parent tasks and sub-tasks present in our scenario. The robot receives textual input and selects the appropriate task to perform. Within a single policy, it executes different unlocking skills tailored to each mechanism. Our framework successfully adapts to various tasks, demonstrating a robust task selection capability and efficient sub-task mode switches.

Denoising Diffusion Probabilistic Model (DDPM) as one popular denoising algorithm reverses the diffusion process by

$$p_\theta(x_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \tag{3}$$

and $p_\theta(x_{0:T})$ takes the form of Markov chain

$$p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{4}$$

During the reverse process, the model learns to approximate the data distribution $p(x_0)$. However, due to the intractability of directly modeling the true posterior distribution, diffusion models are typically trained by maximizing the variational lower bound (ELBO). Under the assumption that the reverse process follows a Gaussian distribution with a fixed variance and a learnable mean function (as derived in [13]), the training objective simplifies to:

$$\mathcal{L}(\varepsilon_\theta) := \sum_{t=1}^{T} \mathbb{E}_{x_0, \varepsilon_t^{(t)}}[||\varepsilon_\theta(\sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\varepsilon_t) - \varepsilon_t||_2^2] \tag{5}$$

Recently, Denoising Diffusion Implicit Models (DDIM) have introduced an alternative formulation to Denoising Diffusion Probabilistic Models (DDPM), significantly reducing the number of diffusion steps, which was a major bottleneck for real-time control. Unlike DDPM, DDIM assumes a non-Markovian and deterministic sampling process, where the transition distribution is conditioned not only on the previous timestep but also on the original clean data, expressed as $P(x_t|x_{t-1}, x_0)$. This formulation enables a reduction in both forward and reverse time steps. While this approach eliminates the stochasticity inherent in standard diffusion models—removing randomness from the generation process—it remains well-suited for our application in this paper. However, the advantages of using DDIM over DDPM for policy learning require further investigation, as it may reduce diversity in sampling trajectories. In our previous work, we used 100 diffusion iterations with DDPM. In this work, we use 50 iterations for training and an adaptive denoising parameter with DDIM, ranging from 10 to 50.

C. Overview of Approach

To inform the diffusion policy of T_p and the T_s at each time step, we introduce a hierarchical structure. Specifically, in the connector disassembly task, the robot must handle various classes of connectors, each requiring distinct manipulation strategies. Within each class, the task can be further decomposed into sub-task modes such as approach, grasp, unlock, and pull, as illustrated in Figure 2.

As done in [15], we consider n observation frames for

conditioning the policy. We also use 16 action prediction horizon, T_a, and define an adaptive execution horizon, T_e, which is dependent on the current sub-task as described in Section IV-B. In our framework, the robot starts from an initial position and executes a sequence of tasks to achieve the overarching goal of disconnecting the connector. In this work, we focus on improving policy execution efficiency by incorporating additional task context and adaptively adjusting key model parameters to better suit the task requirements.

In a standard policy learning framework, mode transitions occur implicitly, meaning the robot switches from approach to grasp based solely on differences in image or force features. However, such implicit mode switches are inefficient, as image features exhibit only minor variations once the gripper is close to the object. The bottleneck makes it difficult for the model to precisely determine when a constraint (e.g., a pre-grasp pose) has been met before closing the gripper. We hypothesize that explicitly informing the model of sub-tasks as discrete states will improve performance in terms of success rate, data efficiency, and inference speed.

We use language as a tool to provide task context across different connector types and mode switches during task execution. The details of our language-conditioning mechanism are described in Section IV-A. Additionally, incorporating a hierarchical structure into the diffusion policy allows us to tune key design parameters, such as T_e and the number of denoising steps, as detailed in Section IV-B. Finally, to effectively weigh the importance of different sensory modalities, we introduce an importance weighing function within the Feature-wise Linear Modulation (FiLM) [42] conditioning mechanism, which is discussed in Section IV-C . An overview of our approach is outlined in Figure 3.

IV. Methodology

A. Using Language to Provide Task-Context for Policy Learning

We observe that when training a Diffusion Policy without explicit labels for different object classes, the robot struggles to perform any specific tasks. We hypothesize that this is due to the high ambiguity in the observation space—where the same visual and force observations correspond to multiple possible action sequences—causing confusion during inference.

Diffusion policy is known to commit to a single trajectory when presented with multiple options, a limitation often associated with multimodality [15]. Consequently, when trained on multiple tasks, we expected it to randomly commit to a single task. However, our experiments showed that, while a model trained on a single-task dataset for the same number of iterations performs the task reasonably well, performance degrades as more tasks are introduced into a single model. By the time the model is trained on four tasks, the robot struggles to commit to any specific task, often exhibiting idle motion instead of executing meaningful actions. This suggests that, without explicit task labels, the model faces difficulty distinguishing between different task modes, ultimately resulting in mode collapse.

To address this issue, we introduce language-based task labels to condition the model on the specific task mode the robot is executing. We flatten T_p and T_s into a language command during execution. For instance, when the robot is performing an approach sub-task for the USB, we use a natural language command such as "Approach USB" to guide the model. These commands can come from either human instructions or an autonomous classifier, as discussed in Section VI. To encode the text input, we leverage a CLIP text encoder, which generates a task-specific embedding. This embedding is then concatenated with the robot's observation space, which includes two camera-view image observations and force data, to condition the policy.

We use CLIP as our text encoder due to its proven effectiveness in Vision-Language Models (VLMs). There has been previous work that has used CLIP encoders for text encoding for policy learning [38]. CLIP encoders are designed to learn robust, semantically meaningful embeddings that align text and image representations in a shared latent space. This capability makes CLIP particularly well-suited for our task, where the robot needs to differentiate between various task modes and object classes based on language inputs.

Additionally, in this work, we constrain the vocabulary used to describe tasks to a predefined set of commands relevant to manufacturing disassembly. However, since CLIP is a general-purpose text encoder, our approach has the potential to scale to a more diverse set of tasks and sub-tasks, enabling robots to perform a broader range of operations in complex manufacturing environments.

B. Adaptive Parameter Selection

In previous work, hyperparameters such as T_e and the number of diffusion steps, t, are fixed throughout both training and inference. However, to optimize execution time and improve precision, it is beneficial to dynamically select these parameters based on the requirements of each sub-tasks, T_p and T_s. For example, closing the gripper does not require high precision when committing to a task, allowing for more aggressive predictions. In contrast, when the gripper approaches the object, fine adjustments in its pose are crucial to grasp success. Given these characteristics, the number of denoising steps can be reduced to decrease latency during action prediction. However, for tasks requiring greater precision, these hyperparameters need to be more conservative. Therefore, we select more denoising steps to obtain more precise actions for such tasks.

C. Importance Weighting Function for Different Sensory Modalities

Diffusion models often employ FiLM conditioning within a U-net architecture to integrate information from various sensory modalities. In these models, FiLM generates channel-wise scaling ($\gamma_{i,c}$) and shifting ($\beta_{i,c}$) parameters using functions f and h. These functions take as input a concatenation of observation modalities—such as images, force measurements, and proprioception—along with a positional embedding for the diffusion time step.

When performing contact-rich tasks, humans typically rely more on vision to guide their hand toward the object of interest. Once the object is grasped, they shift their reliance

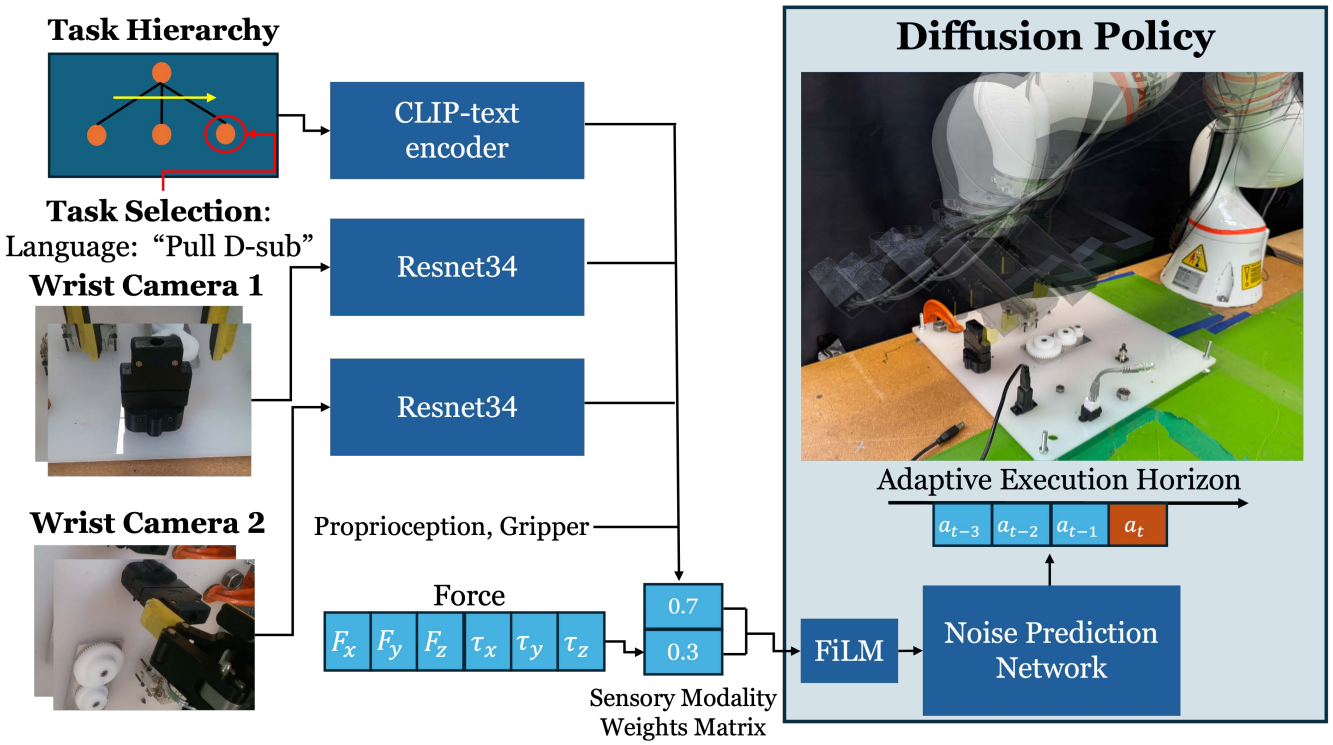

Fig. 3: Our framework utilizes text input for hierarchical task selection. The text description is processed through a CLIP-text encoder to generate text embeddings. For visual inputs, we employ two camera views and extract features using a pretrained ResNet34 [41]. Additionally, we incorporate 6D force data (force and torque), which is fed into a separate FiLM module to learn modulating parameters, as discussed in Section IV-C. This enables adaptive reliance on different sensor modalities based on the current T_s the robot is performing. Finally, the noise prediction network denoises the action sequence a_t using the previous n observations, following [15].

to force and tactile modalities. Inspired by this trait, we introduce a sensory modality weight matrix in the FiLM architecture that dynamically adjusts the reliance on each sensory modality based on $\mathbf{T_s}$. Specifically, we define $\gamma'_{i,c}$ and a shifting term $\beta'_{i,c}$ formulated as follows:

$$\begin{aligned} \gamma'_{i,c} &= w_{image}\alpha_{image} + w_{force}\alpha_{force} \\ \beta'_{i,c} &= w_{image}\beta_{image} + w_{force}\beta_{force} \end{aligned} \tag{6}$$

where $w_{image} + w_{force} = 1$. We use parameters w_{force} of 0.3 for the T_s that requires less precision, and w_{force} of 0.6 once the robot has grasped the object for better force-based guidance.

V. Experiments

A. Hardware Setup

We test our method on KUKA LBR IIWA14 arm with hardware setup shown in Figure 4. We choose the NIST task board #1 as the testbed that is representative of the electronics disassembly problem motivated in Figure 1. We have a BNC, Terminal block , D-sub, and USB connector in one scene (See Figure 3) to show that our method can multiple tasks efficiently while maintaining a single model. Additionally, these tasks require long horizon multi-step planning, which requires mode switching between sub-tasks.

B. Data Collection

For data collection, we use kinesthetic teaching. While teleoperation is a popular method for human demonstration, we find that it requires more time to collect demonstrations and has a greater tendency to introduce idle time

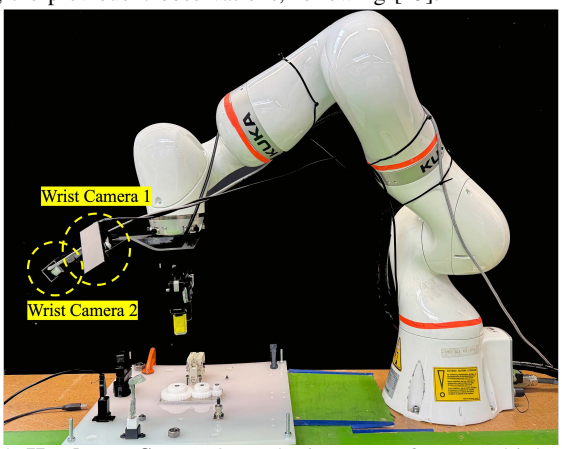

Fig. 4: **Hardware Setup**: Our robotic arm performs multiple tasks on the NIST task board. We utilize two wrist-mounted cameras: one providing a front view to monitor the gripper state and the board, and another offering a side view to assist with gripper and object alignment for successful grasping.

in demonstrations, which adds sub-optimality to the data. Additionally, it poses challenges when the tasks require high precision. During kinesthetic teaching, we record trajectory data. One issue with kinesthetic teaching is that during trajectory data collection, the human may be included in the scene. Therefore, once we collect trajectory data, we play back the trajectory data and collect two wrist camera RGB images, force, and proprioception.

Once we have the collected data, we label the trajectory data with language instructions by segment consisting of the

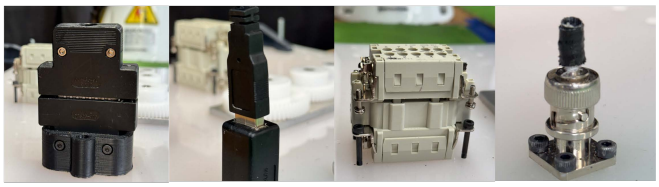

Fig. 5: Different connectors used in our experiment. Each connector requires a distinct unlocking mechanism: a wiggling motion for D-sub connectors, rotation for BNC connectors, and varying force levels for pulling out USB and Terminal block connectors.

hierarchical structure of the parent tasks and the sub-tasks as described in Figure 2. It is important to note that although we manually label all sub-task segments in the trajectories, there are many prior works on temporal action segmentation that can be incorporated for easy labeling [43].

C. Training

Our training data consists of 15 data points each except for the BNC connector, which includes 20 episodes of demonstration. Each demonstration consists of full trajectory from approaching the connector to pulling out the connector. We use delta action representation for our method and 6-D rotation representation for continuous representation for better interpretability in neural networks [44].

During inference, however, we use relative action representation with respect to the current pose of the robot similar to method shown in [16]. We find that this improves the smoothness of the robot trajectory, as delta action has to stop and calculate next action with respect to the current step.

We train our policy end-to-end with CLIP and two pre-trained resnet34 encoder on ImageNet Data and finetune the model with our own data as shown in Figure 3. For our resnet backbone, we replace Batch Normalization with Group Normalization. We use batch size of 64 with learning rate of 1×10^{-4} and weight decay of 1×10^{-3}. We use Exponential Moving Average (EMA) during training recommended for diffusion model training. We train our model on NVIDIA RTX 4080 GPU for 12 hours.

VI. Results

A. End-to-end Success-rate Comparison

To evaluate how each component of our proposed method contributes to performance, we benchmark against the following variants:

1) **DP-S (Single skill Diffusion Policy)** : Diffusion policy trained on single skill
2) **DP-M (Diffusion Policy with Language Instruction)**: Diffusion policy with task labels in language
3) **DP-M-AM (Multi-task Diffusion Policy with Adaptive Modality Weighing)**: A diffusion policy incorporating language-based task instructions and the adaptive modality weighing method described in Section IV-C.

Because the baseline diffusion policy (DP-B) trained in four different skills without language instructions cannot perform any tasks, we consider the success rate as 0% for all types of tasks and do not include it as the benchmark

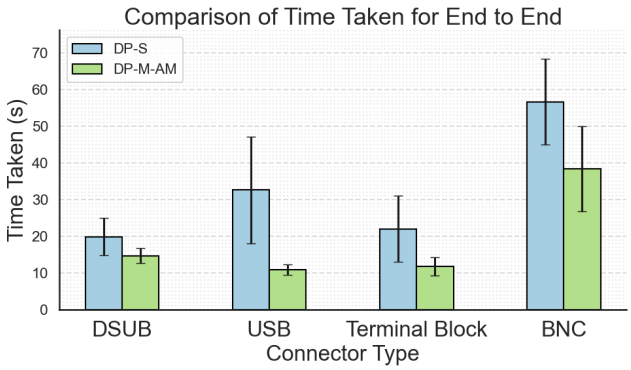

Fig. 6: **Comparison of Robot Execution Time**: We compare the time taken for two different models: DP-S and DP-M-AM(Ours). We see that our method reduces execution time significantly.

method. However, we include DP-S trained on a single task each and compare the success rate with our methods. In this experiment, human-provided task commands are used to guide the robot policy for DP-M and DP-M-AM models.

We repeat the experiment for 20 times per connector type. We define success as the robot approaching the connector and fully removing the connection end-to-end. Especially for the multi-task policies, we include being able to select the right parent task, T_p in the policy as success and consider it as a failure if the robot initiates a different task compared to the language instruction.

	D-Sub	USB	BNC	Terminal	Avg.
DP-S	0.80	0.75	0.65	0.80	0.76
DP-M	1.00	0.95	0.70	0.85	0.88
DP-M-AM	**1.00**	**1.00**	**0.85**	**0.95**	**0.95**

TABLE I: **Success Rate Comparison**: We compare the success rates of three different benchmark methods, as presented in Section VI-B. Our results show that incorporating adaptive sensor modality reliance at different sub-tasks improves performance. Among the connectors, the BNC connector presents the longest-horizon task and is the most challenging due to its higher reliance on contact states and small size, which causes the model to struggle the most.

B. Comparison in Time Taken End-to-end

To demonstrate that our method can effectively reduce the time required for the task, we compare the time taken for each task. We compare our method against a single diffusion policy without language labels trained separately on individual tasks to showcase that our method can efficiently change the modes compared to the regular policy learning framework. We run our experiment on four different connector types listed in Figure 5 twenty times and report the average time taken in Figure 6.

We find that our method reduces the time taken by 57% over all. Next section describes how much of this gain comes from the mode switch.

Failure Mode Analysis In the baseline methods without language labels, distinct failure modes arise. In particular, the most frequent errors occur during grasping. Without language instructions, models struggle to differentiate subtle differences in the pre-grasp pose and approach when relying solely on visual features. This problem is especially evident

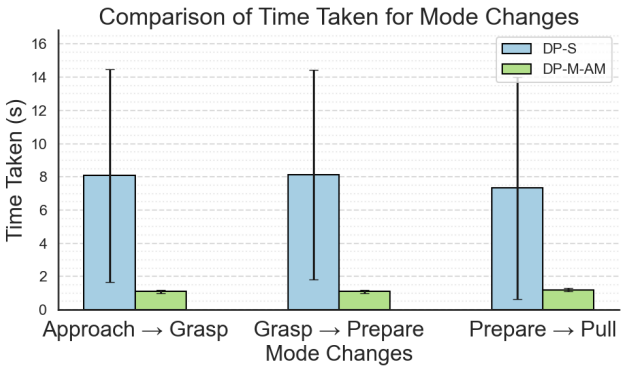

Fig. 7: **Comparison of Time Taken for Mode Change**: We analyze the contribution of mode change to overall efficiency. Our approach achieves significantly faster mode transitions compared to DP-S.

with smaller objects, such as the BNC connector, which is significantly smaller than other connectors. Consequently, the model attempts a premature grasp or lowers the gripper too far, causing it to inadvertently contact the board.

C. Comparison in Time Taken for Mode Switch

In this section, we quantify how much of the time improvement from Section VI-B is attributed to mode switches. We observe that language instructions can trigger an almost instantaneous mode switch, as the robot transitions modes immediately upon receiving an instruction. Figure 7 presents our findings. On average, our approach achieves a 6.7 seconds per mode switch reduction in time and overall attributes 49% of the time saved during execution.

D. Incorporating Mode-Switch Classification

In the previous experiments, we manually provided the policy with task-context input to indicate mode switches. However, when deployed in a manufacturing setting, an autonomous mechanism will be necessary to detect mode switches. To enable this, we investigate a classification-based approach that identifies sub-task transitions without human intervention. Specifically, we introduce a boundary classifier that determines whether the robot has reached the end of a sub-task, indicating to the policy to transition to the next task accordingly.

This framework detects boundaries based on the last n observations. Specifically, we reuse the observation feature embedding architecture from the proposed policy framework and extend it with a classification head for binary boundary detection. The classifier is trained on the same trajectory dataset used for the diffusion policy and achieves a classification accuracy of 97%. Notably, misclassifications occur, on average, just 1.2 ± 0.82 time steps away from the actual boundary, highlighting that the incorrect predictions are temporally close to true transitions.

While not the primary focus of our study, these findings suggest that our framework can operate autonomously by leveraging learned task context to enhance efficiency and robustness. Future work will further explore this direction to strengthen its applicability in fully autonomous scenarios.

E. Case Study

To demonstrate our architecture's capability for language-based interaction in collaborative manufacturing, we conduct a case study where a robot performs a sequence of assembly tasks based on human-specified language inputs. Inspired by practical scenarios involving electronic assemblies requiring diagnostics, we simulated examples of human task preferences to guide the robot's execution sequence. For our experiments, we employed the NIST task board due to its realistic representation of real-world electronic assemblies in terms of connector types and spatial layout. During the case study, the robot executed each task based on a provided language command, followed by manually dropping the connector, resetting to a home position, and receiving the next task instruction.

Case Study 1: The robot performed the following sequence of tasks specified by the human starting with unplugging a BNC connector, followed by disconnecting a D-sub connector, and finally a USB connector. The total execution time for this task sequence was 71 seconds.

Case Study 2: The robot executed another human-specified sequence, beginning with disconnecting a USB connector, followed by a D-sub connector, and concluding with a Terminal block. This sequence was completed in 36 seconds.

VII. Conclusion

In this paper, we extend the diffusion policy framework by incorporating task context, enhancing both robustness and efficiency in task execution while enabling human operators to specify tasks through natural language commands. Our framework improves success rates and accelerates task execution, making it well-suited for manufacturing-related applications. Experimental results demonstrate that our method reduces execution time by 57% and increases success rates by 19% compared to benchmark methods. Additionally, we present case studies illustrating the feasibility of our system, where a human operator commands the robot to execute a sequence of tasks, which the robot performs sequentially. In future work, we aim to extend our framework to more complex assembly and disassembly tasks and further investigate its scalability. Additionally, we plan to develop a more robust classifier with out-of-distribution detection to recover from boundary prediction errors automatically which can be train in conjunction with the diffusion policy.

References

[1] M. Bain and C. Sammut, "A framework for behavioural cloning." in *Machine Intelligence 15*, 1995, pp. 103–129.

[2] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[3] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *5th Annual Conference on Robot Learning*, 2021.

[4] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[5] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[6] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3758–3765.

[7] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 492–499, 2019.

[8] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *Advances in neural information processing systems*, vol. 35, pp. 22 955–22 968, 2022.

[9] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.

[10] D. Jarrett, I. Bica, and M. van der Schaar, "Strictly batch imitation learning by energy-based distribution matching," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7354–7365, 2020.

[11] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

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

[14] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

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

[16] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[17] J. H. Kang, S. Joshi, R. Huang, and S. K. Gupta, "Robotic compliant object prying using diffusion policy guided by vision and force observations," *IEEE Robotics and Automation Letters*, 2025, accepted for publication. Available on arXiv: arXiv:2503.03998.

[18] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5923–5930.

[19] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pre-trained image-editing diffusion models," in *The Twelfth International Conference on Learning Representations*, 2024.

[20] U. A. Mishra and Y. Chen, "Reorientdiff: Diffusion model based reorientation for object manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 867–10 873.

[21] Y. Wu, Z. Chen, F. Wu, L. Chen, L. Zhang, Z. Bing, A. Swikir, A. Knoll, and S. Haddadin, "Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2409.11047

[22] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, "Fmb: a functional manipulation benchmark for generalizable robotic learning," *The International Journal of Robotics Research*, p. 02783649241276017, 2023.

[23] Y. Hou, Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, S. Feng, B. Burchfiel, and S. Song, "Adaptive compliance policy: Learning approximate compliance for diffusion guided control," *arXiv preprint arXiv:2410.09309*, 2024.

[24] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *Advances in neural information processing systems*, vol. 36, pp. 9156–9172, 2023.

[25] I. Kapelyukh, V. Vosylius, and E. Johns, "Dall-e-bot: Introducing web-scale diffusion models to robotics," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3956–3963, 2023.

[26] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[27] S. S. Kannan, V. L. Venkatesh, and B.-C. Min, "Smart-llm: Smart multi-agent robot task planning using large language models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 140–12 147.

[28] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," in *Conference on Robot Learning*. PMLR, 2023, pp. 1769–1782.

[29] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.

[30] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*. PMLR, 2022, pp. 894–906.

[31] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *8th Annual Conference on Robot Learning*, 2024.

[32] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, "Text2reward: Reward shaping with language models for reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=tUM39YTRxH

[33] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson, "Rl-vlm-f: Reinforcement learning from vision language foundation model feedback," in *Proceedings of the 41th International Conference on Machine Learning*, 2024.

[34] J. Duan, W. Yuan, W. Pumacay, Y. R. Wang, K. Ehsani, D. Fox, and R. Krishna, "Manipulate-anything: Automating real-world robots using vision-language models," in *8th Annual Conference on Robot Learning*, 2024.

[35] H. Li, Q. Feng, Z. Zheng, J. Feng, and A. Knoll, "Generalizable robotic manipulation: Object-centric diffusion policy with language guidance," in *Workshop on Embodiment-Aware Robot Learning*.

[36] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 462–12 469.

[37] C. Hao, K. Lin, S. Luo, and H. Soh, "Language-guided manipulation with diffusion policies and constrained inpainting," *arXiv preprint arXiv:2406.09767*, 2024.

[38] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[39] P. Zheng, C. Li, J. Fan, and L. Wang, "A vision-language-guided and deep reinforcement learning-enabled approach for unstructured human-robot collaborative manufacturing task fulfilment," *CIRP annals*, vol. 73, no. 1, pp. 341–344, 2024.

[40] H. Liu, Y. Zhu, K. Kato, A. Tsukahara, I. Kondo, T. Aoyama, and Y. Hasegawa, "Enhancing the llm-based robot manipulation through human-robot collaboration," *IEEE Robotics and Automation Letters*, 2024.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[43] G. Ding, F. Sener, and A. Yao, "Temporal action segmentation: An analysis of modern techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 2, pp. 1011–1030, 2023.

[44] Y. Zhou, C. Barnes, L. Jingwan, Y. Jimei, and L. Hao, "On the continuity of rotation representations in neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.