A Robotic Skill Learning System Build Upon Diffusion Policies and Foundation Models

Nils Ingelhag*, Jesper Munkeby*, Jonne van Haastregt*, Anastasia Varava, Michael C. Welle, Danica Kragic

***Abstract*— In this paper, we build upon two major recent developments in the field, Diffusion Policies for visuomotor manipulation and large pre-trained multimodal foundational models to obtain a robotic skill learning system. The system can obtain new skills via the behavioral cloning approach of visuomotor diffusion policies given teleoperated demonstrations. Foundational models are being used to perform skill selection given the user's prompt in natural language. Before executing a skill the foundational model performs a precondition check given an observation of the workspace. We compare the performance of different foundational models to this end as well as give a detailed experimental evaluation of the skills taught by the user in simulation and the real world. Finally, we showcase the combined system on a challenging food serving scenario in the real world. Videos of all experimental executions, as well as the process of teaching new skills in simulation and the real world, are available on the project's website[1].**

I. Introduction

How can we ensure that robots have the necessary skills to accomplish the various tasks that a specific user might need them to do in its specific environment?

While certain skills are potentially more universal than others, the long-tailed nature [1] of the necessary skills is a major challenge to overcome to make autonomous agents truly ubiquitous in everyday environments.

In this work, the user can continuously show and teach new skills using intuitive teleoperation. Our approach is able to receive instructions via natural language and its current observation - a procedure that most humans are now familiar with thanks to the widespread adaptation of Large Language Models [2]. The framework then consults its skill library - a repository of skills it has learned in the past - and assesses if any of the currently available skills are applicable to execute the given task. If, however, no suitable skill is available, the system will simply ask the user to provide a number of demonstrations (around $50-150$). The new skill can then be trained on external hardware and loaded onto the system when completed. In this way, the system's capabilities can be continuously expanded by the user. A schematic overview of our Robotic Skill Learning System (RSLS) is given in Fig. 1. The system uses a Large Language Model (LLM) and a Large Visual Language model (VLM) as foundational models to assess if any of the learned skills are applicable to fulfill the user's instructions. New skills can be easily integrated by performing demonstrations using an off-the-shelf Oculus VR controller, and training using a visuomotor diffusion policy. We assess the RSLS in the real world on complex skills that have to be executed sequentially such as removing a lid from a bowl before being able to access its content as well as extend the use of diffusion policies to contact-rich and granular material manipulation tasks. We can summarize our contributions as follows:

- we present a fully functional Teaching by Demonstration framework both in simulation and the real world;
- we compare different large Language and Visual Language Models for skill selection;
- we apply diffusion-based visuomotor policy to contact-rich and granular material tasks;
- we conduct an extensive experimental evaluation of each skill and component of the full framework.

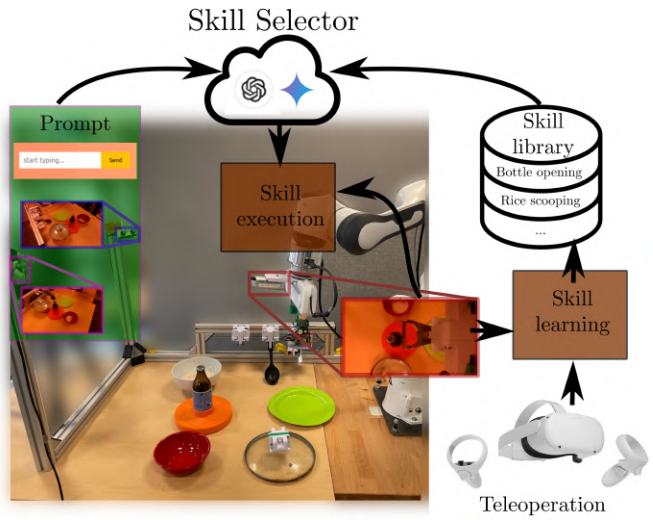

Fig. 1. Conceptual overview of our Robotic Skill Learning System. The system receives the user's instructional prompt and an image of the current state. The skill selector module - realized through a foundational model - selects an appropriate skill to perform the task. If no suitable skill is available, the system asks the user to perform a number of demonstrations and train a new skill using visuomotor diffusion policies.

II. Background & Related Work

In this section, we first briefly explain the required background knowledge and related work in regards to VLM and visuomotor diffusion policies. We end the section by discussing relevant work with respect to our framework.

Large Language/Visual Language Models as Foundational Models for Robotic Skill Selection:

*These authors contributed equally (listed in alphabetical order).

KTH Royal Institute of Technology Stockholm, Sweden, {*ingelhag, munkeby, jmvh, mwelle, varava, dani*}*@kth.se*

[1] https://roboskillframework.github.io/

Mapping natural language descriptions to robotic actions allows for simple and relatable interaction with a robotic system. Approaches to achieve this vary, including direct conditioning of models on actions[3][4][5], as well as strategies that combine Large Language Models (LLMs) with visual data, creating Visual Language Models (VLMs) that can take multimodel inputs i.e. text and images or video as input. The integration of VLMs into robotics is a step towards robots that can interpret and act on natural language instructions in a visual context. Examples demonstrating the usefulness of this approach include frameworks that emulate human cognitive processes for nuanced task execution [6], and the processing of complex, multimodal task descriptions [7]. These advancements collectively highlight the promise of VLMs as a strategy to enhance robotic capabilities.

Visuomotor diffusion policies for robotic manipulation:

Diffusion, originally popularized in the domain of generative image models, have recently emerged as a powerful tool in the field of robotics. In particular, diffusion-based policy models used for behavioral cloning have shown promising results [8]. At their core, diffusion models learn to gradually construct complex data distributions, starting from random noise and progressively refining this noise into structured outputs. In the context of robotics, these outputs are sequences of actions (in task space) of the robot to perform a given task.

One key benefits of diffusion-based policy models is the few number of human demonstrations needed to clone a specific behavior. [8] showed that only around 100 demonstrations were needed to clone complex behaviors such as flipping objects and handling liquids with over 70% success rate while being more robust to disturbances and idle actions than other approaches such as [9]. Another benefit of using diffusion-based policy models is the lack of explicit modeling that needs to be done of the task and environment. [10] shows how the same underlying learning method can be used to control a bi-manual mobile manipulation system to navigate indoor environments and perform kitchen tasks. In another example of it's versatility, [11] shows how diffusion policy is used to handle deformable objects for the the purpose of robot-assisted surgery.

However, while those frameworks expand the use case of [8] significantly, which skill is executed and when is still entirely decided manually by an expert.

General Skills Learning Frameworks There have been several proposed frameworks that leverage language models in combination with learned skills to make robots capable of completing queried tasks in a scene.

[12] utilizes a visual language model to first decompose an overall goal into a series of subtasks given a scene of interactable objects. By superposing the camera view with a grid and keypoints on relevant objects, the VLM is able to communicate well-defined spatial planning for robotic manipulation. Keypoint-based navigation helps to mitigate some of the shortcomings still found in the spatial reasoning of current VLMs, but they also restrict the planning to primitive and non-complex tasks.

Another approach [13] leverages a 3D-LLM [14] together with a "goal imagination" diffusion model to generate actions given a scene and a goal. The 3D-LLM is first used to condition the goal imagination process given the stated goal. When an imagined 3D scene of the goal has been generated, the 3D-LLM is used again to generate a sequence of action tokens for the manipulator to execute. While this has the potential to create more intricate action plans due to its more native 3D understanding, it is still limited to open-loop control.

The approach in [15] attempts to retrieve skills from unstructured play data. The play data is language labeled in hindsight which is used to condition a diffusion-based next action predictor. By introducing a quantization bottleneck in the diffusion process, this method is able to discretize the learned representations into individual finer skills. The discrete skills can then be used in new combinations to achieve novel goals, showcased on tasks such as pick and place action in a dinning table setting.

III. Robotic Skill Learning System

In this section, we describe the Robotic Skill Learning System (RSLS) framework in detail. Fig. 2 shows an overview of our RSLS setup in the real world. It consists of a Skill Selector which is based on a foundational model. Given the user's input prompt, the first step of the skill selector is to find a suitable skill in the library of skills. If no skill is found the system requests to be taught a new skill by the user. In this case, the RSLS enters demonstration mode, and the user can perform repeated demonstrations of the new skill which will be, after training, added to the skill library. If a skill is found, the skill selector checks the preconditions of the particular skill given an image of the current workspace using the multimodal aspect of the foundational model. If all preconditions are met, the skill is then sent to the Skill Execution Module.

A. Teaching a Skill

We record the demonstrations in such a way that they are compatible with the diffusion policy training framework introduced in [8]. To facilitate teleoperating the robot we make use of the stand-alone app Quest2ROS [16] for the Oculus Quest. This way, a user can easily record demonstrations for an arbitrary task using readily available hardware. We extend the range of tasks via diffusion policy to contact-rich tasks (bottle opening) as well as the handling of granular material (rice scooping task) in the real world.

B. Training/Executing a Skill

Once a number of demonstrations (depending on the skill, between $50 - 150$) are collected for any particular task, we train a visuomotor diffusion policy as detailed in [8]. Specifically, we use a CNN-based denoising network along with separate RestNet18 encoders for each camera view. Once the skill is trained, it is added to the Skill library with a short description of the skill, as well as what preconditions have to be fulfilled, and a method to execute to use the skill, as shown in1.

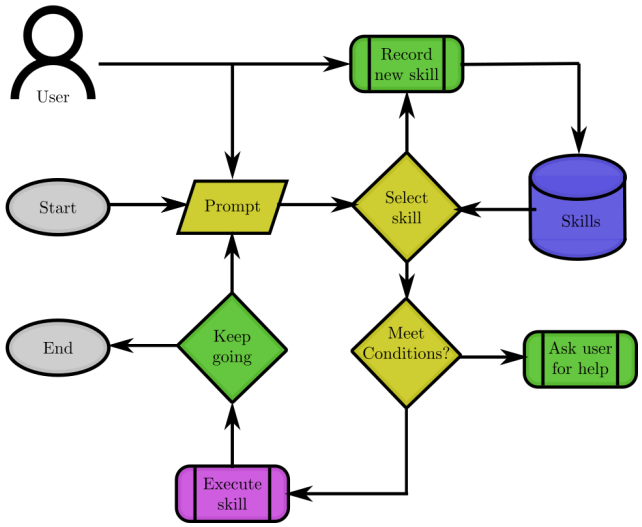

Fig. 2. Flowchart overview of our RSLS method. Yellow boxes indicate the Skill Selector realized through a foundational model, green indicates the user's activity, and purple shows the visuomotor diffusion policy.

Listing 1. An entry in the skill library

```
Skill(
    "SERVE RICE",
    "This skill serves rice from a white bowl into a red bowl",
    "The white bowl needs to contain rice. A red bowl needs to visible in the
    workspace."
    ),
```

C. Skill Selector

The Skill Selector is realized using a large pre-trained Foundational Model. The input is an image of the current scene as well as the user's instructional prompt. In the first step, the user prompt is fed into an LLM, which has access to the names and descriptions of the Skill library. The task of the LLM is to select a suitable skill given the user's request and the names and descriptions of the skills in the skill library. If no suitable skill is identified, the system will ask the user to teach it the new required skill.

If a matching skill is found by the LLM, a second step is performed to check if all preconditions for performing the skill are met. The preconditions are sent to the VLM along with an image of the scene. As output, the VLM will have to make a "YES" or "NO" decision on whether the preconditions are met or not. If any of the preconditions are violated, the system will inform the user which can amend the situation. If no preconditions are violated, the execution method of the skill is called and the skill is executed.

This flow is depicted schematically in Figure 2. Determining a suitable skill using only the LLM in the first step of the skill selector saves the query to the more computationally expensive VLM if no suitable skill is found in general. If a skill is found the VLM can be prompted more specifically to only check if the by the skill given precondition is fulfilled and the skill can be executed.

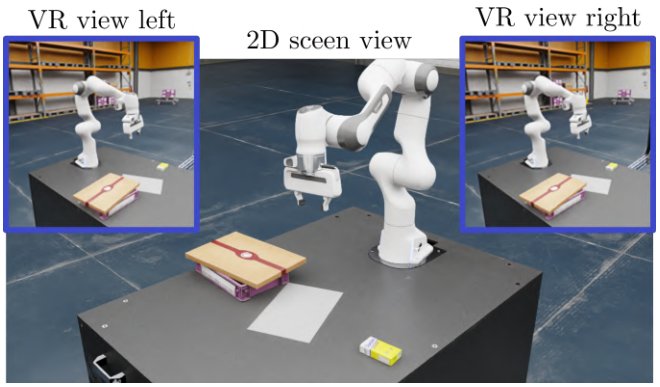

Fig. 3. Setup of the simulation environment, including the VR views for the left and right eye. Note that as the user is free to traverse the virtual environment he can obtain different views than those shown on a 2D screen.

D. Simulation Setup

To obtain as realistic conditions for the simulation as possible we adopt Isaac Sim with the Orbit framework [17] for simulation. To make the teleoperation similar to the real world, we adapted the teleoperation framework from [16]. In order to successfully teleoperate a robot in 3D space, it is beneficial to display the environment in a 3D space as well; receiving teleoperation feedback from a 2D screen can result in confusion [18][19]. To overcome this issue, a VR camera rig is set up in Orbit which records stereoscopic images using a virtual camera which are streamed to the Oculus headset. The cameras have an asymmetric frustum and matched parameters to the human eye to enhance the 3D experience. The movements of the Oculus in the real world in turn determine the change in position of the virtual camera in Orbit. This way the user is placed in an interactive 3D scene when performing demonstrations which make it possible to immersively and naturally teleoperate the robot in simulation using the same control scheme as in the real world. The setup can be seen in Fig 3 including the left and right eye images.

E. Real World Setup

The robotic setup can be seen in Fig. 4, and consists out of a Franka Panda manipulator with a Realsense camera mounted on the end-effector, the robot is able to interact/manipulate with the food-related items such as a bottle, a bowl of rice with a lid and a plate of sausages, in its workspace. As each task in the real world requires a specific tool a tool change station is also mounted in the workspace of the robot. Furthermore, two additional real-sense cameras are mounted on the opposite side of the robot providing the skill selector with an unobstructed view of the scene.

Tool Changer: the Toolchanger holds three different tools as shown in Fig. 4: *i)* a bottle opener, *ii)* a large serving spoon, and *iii)* a compliant custom gripper. If a skill needs a specific tool the robot will first exchange/equip the appropriate tool for the skill. Not having any tool is also a viable option, for instance when performing a pick-and-place task such as the

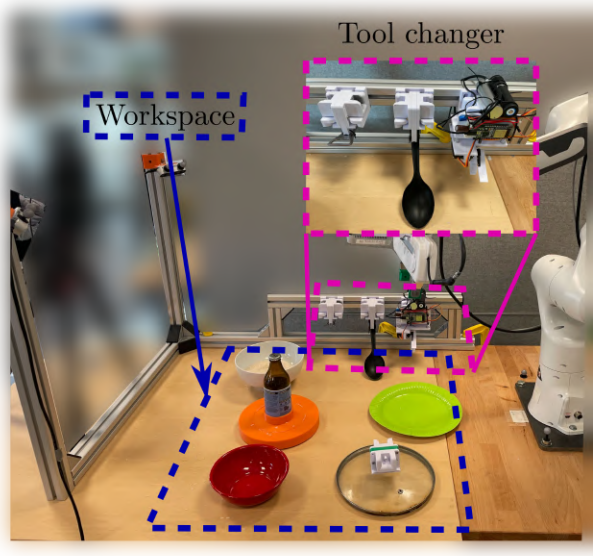

Fig. 4. The real world setting indicating the workspace (blue) and the tool changer (pink) containing a bottle opener, a serving spoon, and a custom gripper for the sausages.

lid removal.

IV. Experimental Evaluation

We first report insights from the training of new skills using experienced operators. Next, we report the performance of the trained skills individually as well as the performance of the skill selector. Finally, we validate the functionality of the full system.

A. Human in the Loop

In this work, we collected approximately 100 demonstrations per skill in simulation and the real world by two experienced operators – authors of the paper. A total of 7 different skills have been trained. The demonstrations are done using the stand-alone Oculus app Quest2ROS [16] in both settings. In the simulation, the functionality of the app has been extended to include head-tracking and VR rendering. Fig. 5 shows the histograms of the demonstration time for the skills learned in simulation (left) and the real world (right) respectively. The mean time for each skill is indicated by the dashed lines. We can show that rather complex tasks such as opening a bottle can be demonstrated in a short amount of time i.e. around $100 \cdot 17.1s = 28.5$ min. Overall we can see that the demonstrations follow an expected normal distribution with a mean time of 19.5, 15.6, and 58.1 seconds for the cap removal, pick and placing, and crate pushing in simulation respectively. Naturally, pushing the crate to a target configuration is more complex and therefore takes a much longer time on average than the other tasks. In the real world, the tasks' mean durations are: bottle opening - 17.1s, lid removal - 22.3s, rice scooping - 25.5s, and sausage placing - 32.2s. The sausage placing

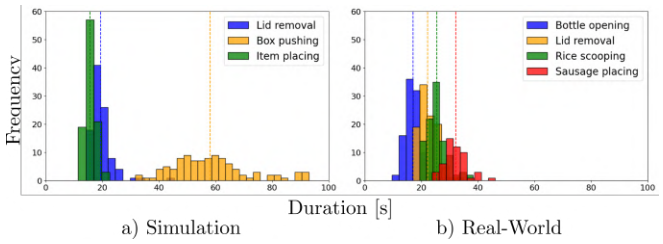

Fig. 5. Demonstration time histograms for the lid removal (blue), box pushing (orange), and item placing (green) in simulation (left) as well as for the real-world tasks Bottle opening (blue), lid removal (orange), rice scooping (green), and sausage placing (red) on the right. The dashed lines indicate the mean duration of the respective task.

only received 50 demonstrations, as a single demonstration includes placing three sausages into the red bowl.

Observations are collected at 0.1s intervals during demonstrations. The simulation is played at half speed to ensure a sufficiently low computational load and ensure physics interactions can be computed correctly.

B. Simulation Setting

As shown in Fig. 3, the simulation setting consists of three different tasks. *i)* Cap removal - remove the cap from the box, *ii)* crate pushing - push the crate onto the designated target area, and *iii)* picking and placing - picking up a box of sugar and putting it into the box. We report the results of all three tasks in Table I, including the number of demonstrations used for training and the success criteria. Furthermore, the videos of all demonstrations and evaluations can be found on the project website[1].

Implementation details: The data collected for the training of the policies is similar to the real world. With a frequency of 10Hz, the following observations are logged: A 4x4 transformation matrix from the robot base to the end-effector, A 240x320 image from an end-effector camera, and the actions recorded from the oculus controller, which is a 6D velocity vector in end-effector space along with a scalar value that acts as the gripper activation signal. After each demonstration, the scene can be easily reset by the user, which randomly initializes the pose of the purple crate and its cap, the gray plane, as well as the box of sugar.

When playing the policies, a parallel thread is started to perform inference at a frequency of 4Hz. The action horizon is then updated as soon as the inference has finished.

All trained policies are evaluated by 100 trial runs. The success criteria and rates are shown in table I. The results are discussed for each experiment below.

Cap removal: This task reaches a success rate of 83%, the only failure cases occur when the robot did not only push the cap but the crate as well.

Crate pushing: When training and running this task on only the end-effector view the performance is very poor - only 20% reached a IoU > 0.8. This is expected, as the end-effector view barely contains any information about the position of the crate relative to the goal. When adding an

Name	N_{Demo}	Success criteria	Time limit	Success rate	Notes
Cap removal	100	The geometric center of the lid is more than it's length away from the geometric center of the crate and the crate is on top of the table.	20s	83/100	
Crate pushing (single view)	100	The Intersection over Union (IoU) of the crate and goal marking is larger than 0.8.	60s	20/100	Average IuO: 39.0%
Crate pushing (multi view)	100	The Intersection over Union (IoU) of the crate and goal marking is larger than 0.8.	60s	74/100	EE, front & side view. Average IuO: 83.4%
Pick and Placing	100	The geometric center of the sugar box is inside the convex hull of the crate.	20s	96/100	

TABLE I

OVERVIEW OF ALL SIMULATED POLICY EXPERIMENTS.

additional front and side view the policy receives additional information containing more spatial cues that can inform the actions leading to a success rate of 74%. Still, the policy can end up in cases where the diffused actions are very small resulting in the robot staying in place and not pushing the box anymore. Another failure case is created due to errors in the physics simulation. Occasionally, the gripper would clip into the crate during pushing and the two objects can not move independently anymore.

Pick and placing: With 96% this is the highest scoring task, the failures happened because of the sugar box being dropped on the edge of the crate.

C. Real World Tasks

As shown in Fig. 4 the real-world setting consists of four different skills in a food serving setting, and each skill requires a dedicated tool. The pose of the objects of the table is varied between all experiments. The individual skills are: *i)* bottle opening - using the opener to remove the bottle cap from the bottle, *ii)* lid removal - removing the lid on top of the white rice bowl using the gripper, *iii)* rice scooping - transferring rice from the white bowl into the red bowl using the serving spoon, and *iv)* sausage placing - placing sausages from the green plate into the red bowl using the custom gripper.

Implementation details: The policy model was configured to operate at a control frequency of 10 Hz, using the observations from 2 previous iterations and predicting 14 timesteps into the future. The first 8 of these timesteps were then executed on the robot before generating 14 new actions and repeating the process. All skills were conditioned on the observations from the end-effector camera. The camera feed was first resized to $240x320$ before being fed into the encoder and the overall policy model. The robot state represented as a $4x4$ transformation matrix from the base to the end effector was also used as observations. Each skill was trained for 600 epochs, taking approximately 10 h on a NVIDIA GeForce RTX 4090.

The performance of all individual skills is shown in Table II. All execution videos of all experiments are available on the project website[1].

Bottle opening: Constitutes the most challenging task reaching 60%. The main failure case is when the robot misses the bottle when approaching it from above. When the bottle opener gets in a position relative to the bottle that is out of the dataset distribution. In one such failure case, the robot was able to recover but just outside the time window for successful task completion. Whenever the robot was able to latch onto the cap, a successful task completion always followed swiftly.

Lid removal: Succeded 90%, in the single failure case, the grippers was not successful in centering itself when approaching the lid from above. This resulted in one of the gripper fingers getting stuck on the handle and the robot was unable to recover.

Rice scooping: Reached 90% as well. This policy was given 90 seconds due to its complex nature. In the single failure case of the nine tries, the robot was able to scoop up a sufficient amount of rice and transfer it over to the red bowl. However, it never tilted the spoon to let the rice into the bowl.

Sausage placing: Performed successfully in 90% of the cases and the single failure case occurred when the gripper dropped one of the sausages over the green plate and it landed in a position it was not able to be picked up from.

D. Skill Selector

The skill selector's job is to select an adequate skill given a user prompt or request a new skill, once a skill is found an image of the scene is used to determine if the preconditions for the skill are fulfilled. We compare two state-of-the-art foundational models, GPT-4 [20] and Gemini [21] (accessed 15/03/2024).

Evaluating foundational models is notoriously difficult and still an open research direction [22]. For that reason, the models are evaluated on our specific use cases. One evaluation is performed for the skill matching step and one for the precondition validation step.

The skill matching evaluation is set up as follows: For each 16 combinatorial variation of the four skills in the skill library (including no skill at all), each four skills are requested using two variations for the user prompt. The experiment is repeated five times resulting in a total of 640 prompts and responses to evaluate. The skills in the library and the user prompt are fed into the foundational model using the template in Listing 2. The descriptions and preconditions of the skills are stated in Listing 3. The user prompts and corresponding skills are stated in Listing 4. The response is

Name	N_{Demo}	Success criteria	Time limit	Success rate
Rice scooping	101	At least 5 grams of rice has been moved from the white bowl into the red bowl.	90s	9/10
Bottle opening	100	The bottle cap is fully removed from the bottle.	60s	6/10
Lid removal	100	The lid is placed onto the table.	60s	9/10
Sausage placing	50	All three sausages have been moved from the green plate into the red bowl.	60s	9/10

TABLE II

OVERVIEW OF ALL REAL-WORLD POLICY EXPERIMENTS.

considered correct if it includes the skill name the user asked for and this skill is indeed in the library. If the requested skill is not in the library, the response is considered correct if it requests a new skill. The results are shown in Table III.

Listing 2. Prompt template to the LLM the skills and user input get injected to the placeholders

```
You are an expert skill selector that has to match skills that are given to a
user's request. If none of the skills given to you are fulfilling the users
request, answer with "NEW SKILL".

Your skills are:
[[[SKILL NAMES AND DESCRIPTIONS]]]

User request:
[[[PROMPT]]]

Structure your answer in this format:
[reasoning without metioning the names of skills]
[Skill Name]
```

Listing 3. Description and preconditions of the four skills in the skill library

```
Skill(
    "SERVE RICE",
    "This skill serves rice from a white bowl into a red bowl",
    "The white bowl needs to contain rice. A red bowl needs to visible in the
        workspace."
    ),
Skill(
    "OPEN BEER",
    "This action opens the beer bottle by removing the metal cap",
    "The bottle needs to be closed with a metal cap"
    ),
Skill(
    "SERVE SAUSAGE",
    "This skill picks up one or more sausages from a green plate and puts them
        into a red bowl",
    "A green plate with sausages on them needs to be visible in the workspace. A
        red bowl needs to be visible in the workspace which could contain
        something already.",
    ),
Skill(
    "REMOVE LID",
    "This skill removes the glass pan cover from the white bowl of rice.",
    "A glass pan cover has to be present and not on the table.",
    )
```

Listing 4. Mock user requests used for the evaluation of skill matching.

```
Rice scooping: ["Serve the rice please.", "I want rice!"]
Bottle opening: ["Open the bottle!", "I would like something to drink please."]
Sausage placing: ["Give me some meat!", "Please move the sausages from the green
    plate to the red bowl"]
Lid removal: ["Please remove the lid from the rice.", "Uncover the rice!"]
```

In order to evaluate the VLM, we collected 10 images of the scene for each permutation of skills that are able to be performed. In total 110 pictures were collected. For each picture, the preconditions are validated for all four skills, leading to 440 prompts and responses to evaluate.

The preconditions shown in Listing 3 are fed into the VLM using the template in Listing 5 along with an image of the scene. A response is considered correct if a "YES" or "NO" is retrieved from the last line of the response which matches the ground truth of whether all preconditions are met.

The experiment is repeated three times, using images from a camera on the left side of the scene, images from the right side, and once using both images. The results are shown in Table III.

Listing 5. Prompt template to the VLM the precondtion gets injected into the placholder

```
Please check if the following conditions are met in the image:
[[[PRECONDITIONS]]]

Answer format for each precondition:
[Short Reasoning]
[YES/NO]

End the response with a definitive answer (YES/NO) on whether ALL conditions are
met on a new line.
```

Model	LLM	VLM-l	VLM-r	VLM-lr	LLM·VLM-r
GPT-4	**96.3**%	71.1%	**77.5**%	71.9%	**74.6**%
Gemini	93.0%	69.1%	75.7%	65.0%	70.4%

TABLE III

COMPARISON OF GPT-4 AND GEMINI AS FOUNDATIONAL MODELS FOR SKILL SELECTION. BEST RESULTS IN BOLD.

All prompts and responses for both the LLM and VLM evaluation can be found on the website[1]. From Table III, it can be seen that the LLM variant of both foundational models is able to achieve a high result matching the user request to the skill library at hand. Precondition checking seems to be slightly more difficult. Having a good camera angle seems to matter, as well as that more cameras do not necessarily result into better performance, which might be attributed to the fact that more information can lead to confusing answers of the VLM. A multiplicative result is shown taking the best result from the LLM evaluation together with the best result of the VLM evaluation. This is a good indicator for the success rate of a request to the robot in the full system. GPT-4 performed slightly better than Gemini and was subsequently used in the full systems.

Peculiar failure case: Fig. 6 shows a particular peculiar failure case of the foundational model when prompted to check the preconditions for the bottle opening, the model responded in Chinese characters instead with the prompting and location language of English. This kind of unexpected failure mode highlights the need for further studies and additional output checks when deploying these models in combination with other technology.

E. Validation of SSLE Framework

As a user study lies outside of the scope of this work we validate the full framework by playing out an interaction with a typical user. The validation aims to show the human-in-the-loop interactions when new skills are required and when skills can not be performed due to precondition validation. We recommend to watch the validation video on the website[1].

The interaction is as follows: the user arrives at the scene as shown in Fig. 7. For now, the tasks in the skill library are

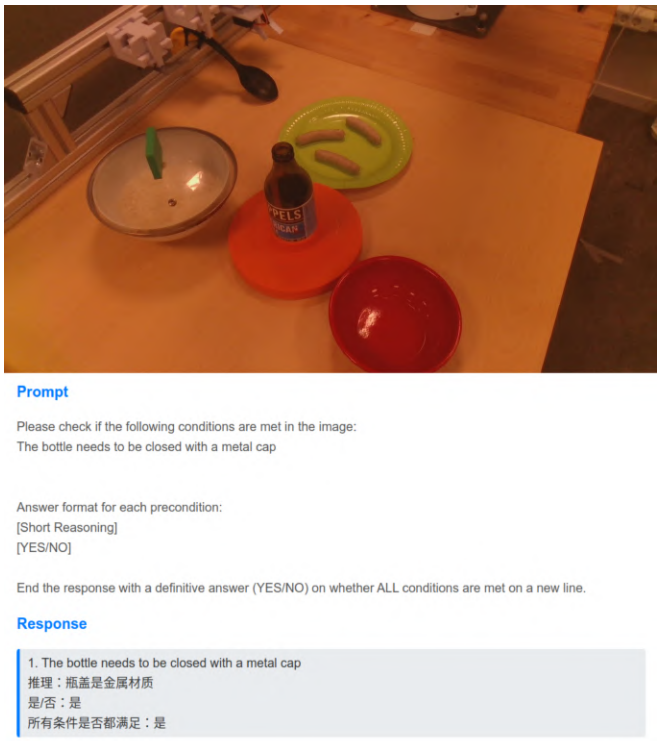

Fig. 6. Peculiar failure case of the model when given an image and a prompt, the response is largely in Chinese.

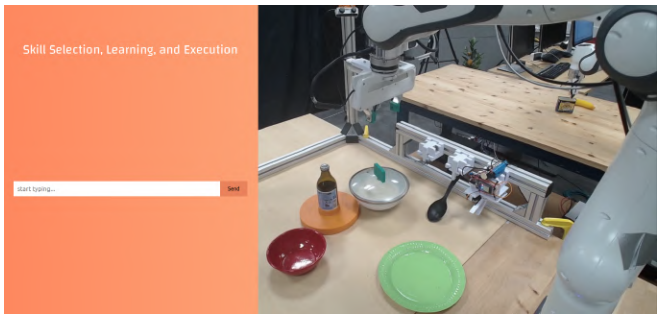

Fig. 7. The setup used for validation, with a prompting window (left) and the workspace (right)

”Bottle opening”, ”Rice scooping” and ”Sausage placing”. The user starts by saying they are thirsty and would like a refreshment. The framework finds a suitable skill (Bottle opening), validates the preconditions, and executes the skill. Next, the user asks to remove the pan cover from the rice. The system replies that no suitable skill is available and requests to teach a ”new” lid removal skill. At this point, the skill library is reloaded with the additional ”Lid removal” skill to simulate the act of having shown and learned a new skill. When the user repeats the prompt in regards to removing the pan cover, this action is now executed. After successfully removing the pan cover, the user asks to move the rice to the red bowl. The system matches the rice scooping skill, validates the preconditions, and executes the action. Finally, the user prompts the system to provide some sausages. Since there are no sausages on the plate yet, the system returns that while it has the skill to do so, it can not perform it since the preconditions are not met. After placing some sausages on the plate and re-prompting, the system executes the sausage placing skill.

V. CONCLUSION

In this work, we presented a Robotic Skill Learning System that builds upon diffusion policies and foundational models. Our system is able to learn novel tasks via diffusion policies using approximately 100 demonstrations per task given by the user. We compared two state-of-the-art foundational LLMs/VLMs in their role to select a known skill from a skill library or ask for a new skill as well as their capability to check preconditions and determine if the skill should be executed or not. We extensively evaluated the individual skills, and parts of the system with all detailed results public on the project website[1], and validated the whole framework as shown in the supplementary video.

REFERENCES

[1] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, “Deep long-tailed learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[2] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, “Gpts are gpts: An early look at the labor market impact potential of large language models,” *arXiv preprint arXiv:2303.10130*, 2023.

[3] Z. Liang, Y. Mu, H. Ma, M. Tomizuka, M. Ding, and P. Luo, “Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution,” *arXiv preprint arXiv:2312.11598*, 2023.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.

[6] M. Zhu, Y. Zhu, J. Li, J. Wen, Z. Xu, Z. Che, C. Shen, Y. Peng, D. Liu, F. Feng, *et al.*, “Language-conditioned robotic manipulation with fast and slow thinking,” *arXiv preprint arXiv:2401.04181*, 2024.

[7] J. Yang, Y. Dong, S. Liu, B. Li, Z. Wang, C. Jiang, H. Tan, J. Kang, Y. Zhang, K. Zhou, *et al.*, “Octopus: Embodied vision-language programmer from environmental feedback,” *arXiv preprint arXiv:2310.08588*, 2023.

[8] 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.

[9] 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.

[10] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *arXiv*, 2024.

[11] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, *Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects*, 2023. arXiv: 2312.10008 [cs.RO].

[12] F. Liu, K. Fang, P. Abbeel, and S. Levine, *Moka: Open-vocabulary robotic manipulation through mark-based visual prompting*, 2024. arXiv: 2403.03174 [cs.RO].

[13] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan, *3d-vla: A 3d vision-language-action generative world model*, 2024. arXiv: 2403.09631 [cs.CV].

[14] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, *3d-llm: Injecting the 3d world into large language models*, 2023. arXiv: 2307.12981 [cs.CV].

[15] L. Chen, S. Bahl, and D. Pathak, *Playfusion: Skill acquisition via diffusion from language-annotated play*, 2023. arXiv: 2312.04549 [cs.RO].

[16] M. C. Welle, N. Ingelhag, M. Lippi, M. Wozniak, A. Gasparri, and D. Kragic, "Quest2ros: An app to facilitate teleoperating robots," in *7th International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions*, 2024.

[17] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, 2023.

[18] R. Hetrick, N. Amerson, B. Kim, E. Rosen, E. J. de Visser, and E. Phillips, "Comparing virtual reality interfaces for the teleoperation of robots," in *2020 Systems and Information Engineering Design Symposium (SIEDS)*, IEEE, 2020, pp. 1–7.

[19] M. Moletta, M. K. Wozniak, M. C. Welle, and D. Kragic, "A virtual reality framework for human-robot collaboration in cloth folding," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, IEEE, 2023, pp. 1–7.

[20] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, *et al.*, *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL].

[21] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, D. Silver, S. Petrov, M. Johnson, I. Antonoglou, J. Schrittwieser, A. Glaese, J. Chen, E. Pitler, T. Lillicrap, *et al.*, *Gemini: A family of highly capable multimodal models*, 2023. arXiv: 2312.11805 [cs.CL].

[22] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, Jan. 2024, ISSN: 2157-6904. DOI: 10.1145/3641289. [Online]. Available: https://doi.org/10.1145/3641289.