arXiv:submit/4778071 [cs.RO] 8 Mar 2023

Grounding Language with Visual Affordances over Unstructured Data

Oier Mees*[1], Jessica Borja-Diaz*[1], Wolfram Burgard[2]

***Abstract*— Recent works have shown that Large Language Models (LLMs) can be applied to ground natural language to a wide variety of robot skills. However, in practice, learning multi-task, language-conditioned robotic skills typically requires large-scale data collection and frequent human intervention to reset the environment or help correcting the current policies. In this work, we propose a novel approach to efficiently learn general-purpose language-conditioned robot skills from unstructured, offline and reset-free data in the real world by exploiting a self-supervised visuo-lingual affordance model, which requires annotating as little as 1% of the total data with language. We evaluate our method in extensive experiments both in simulated and real-world robotic tasks, achieving state-of-the-art performance on the challenging CALVIN benchmark and learning over 25 distinct visuomotor manipulation tasks with a single policy in the real world. We find that when paired with LLMs to break down abstract natural language instructions into subgoals via few-shot prompting, our method is capable of completing long-horizon, multi-tier tasks in the real world, while requiring an order of magnitude less data than previous approaches. Code and videos are available at http://hulc2.cs.uni-freiburg.de.**

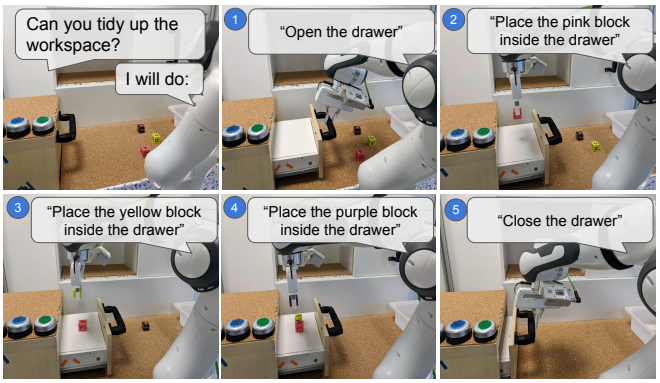

Fig. 1: When paired with Large Language Models, HULC++ enables completing long-horizon, multi-tier tasks from abstract natural language instructions in the real world, such as "tidy up the workspace" with no additional training. We leverage a visual affordance model to guide the robot to the vicinity of actionable regions referred by language. Once inside this area, we switch to a single 7-DoF language-conditioned visuomotor policy, trained from offline, unstructured data.

I. Introduction

Recent advances in large-scale language modeling have produced promising results in bridging their semantic knowledge of the world to robot instruction following and planning [1], [2], [3]. In reality, planning with Large Language Models (LLMs) requires having a large set of diverse low-level behaviors that can be seamlessly combined together to intelligently act in the world. Learning such sensorimotor skills and grounding them in language typically requires either a massive large-scale data collection effort [1], [2], [4], [5] with frequent human interventions, limiting the skills to templated pick-and-place operations [6], [7] or deploying the policies in simpler simulated environments [8], [9], [10]. The phenomenon that the apparently easy tasks for humans, such as pouring water into a cup, are difficult to teach a robot to do, is also known as Moravec's paradox [11]. This raises the question: how can we learn a diverse repertoire of visuomotor skills in the real world in a scalable and data-efficient manner for instruction following?

Prior studies show that decomposing robot manipulation into semantic and spatial pathways [12], [13], [6], improves generalization, data-efficiency, and understanding of multi-modal information. Inspired by these pathway architectures, we propose a novel, sample-efficient method for learning general-purpose language-conditioned robot skills from unstructured, offline and reset-free data in the real world by exploiting a self-supervised visuo-lingual affordance model. Our key observation is that instead of scaling the data collection to learn how to reach any reachable goal state from any current state [14] with a single end-to-end model, we can decompose the goal-reaching problem hierarchically with a high-level stream that grounds semantic concepts and a low-level stream that grounds 3D spatial interaction knowledge, as seen in Figure 1.

Specifically, we present Hierarchical Universal Language Conditioned Policies 2.0 (HULC++), a hierarchical language-conditioned agent that integrates the task-agnostic control of HULC [10] with the object-centric semantic understanding of VAPO [13]. HULC is a state-of-the-art language-conditioned imitation learning agent that learns 7-DoF goal-reaching policies end-to-end. However, in order to jointly learn language, vision, and control, it needs a large amount of robot interaction data, similar to other end-to-end agents [4], [9], [15]. VAPO extracts a self-supervised visual affordance model of unstructured data and not only accelerates learning, but was also shown to boost generalization of downstream control policies. We show that by extending VAPO to learn *language-conditioned affordances* and combining it with a 7-DoF low-level policy that builds upon HULC, our method is capable of following multiple long-horizon manipulation tasks in a row, directly from images, while requiring an order of magnitude less data

*Equal contribution.

[1]University of Freiburg, Germany.[2]University of Technology Nuremberg, Germany.

than previous approaches. Unlike prior work, which relies on costly expert demonstrations and fully annotated datasets to learn language-conditioned agents in the real world, our approach leverages a more scalable data collection scheme: unstructured, reset-free and possibly suboptimal, teleoperated *play* data [16]. Moreover, our approach requires annotating as little as 1% of the total data with language. Extensive experiments show that when paired with LLMs that translate abstract natural language instructions into a sequence of subgoals, HULC++ enables completing long-horizon, multi-stage natural language instructions in the real world. Finally, we show that our model sets a new state of the art on the challenging CALVIN benchmark [8], on following multiple long-horizon manipulation tasks in a row with 7-DoF control, from high-dimensional perceptual observations, and specified via natural language. To our knowledge, our method is the first explicitly aiming to solve language-conditioned long-horizon, multi-tier tasks from purely offline, reset-free and unstructured data in the real world, while requiring as little as 1% of language annotations.

II. Related Work

There has been a growing interest in the robotics community to build language-driven robot systems [17], spurred by the advancements in grounding language and vision [18], [19]. Earlier works focused on localizing objects mentioned in referring expressions [20], [21], [22], [23], [24] and following pick-and-place instructions with predefined motion primitives [25], [6], [26]. More recently, end-to-end learning has been used to study the challenging problem of fusing perception, language and control [4], [27], [28], [1], [10], [9], [15], [5]. End-to-end learning from pixels is an attractive choice for modeling general-purpose agents due to its flexibility, as it makes the least assumptions about objects and tasks. However, such pixel-to-action models often have a poor sample efficiency. In the area of robot manipulation, the two extremes of the spectrum are CLIPort [6] on the one hand, and agents like GATO [5] and BC-Z [4] on the other, which range from needing a few hundred expert demonstrations for pick-and-placing objects with motion planning, to several months of data collection of expert demonstrations to learn visuomotor manipulation skills for continuous control. In contrast, we lift the requirement of collecting expert demonstrations and the corresponding need for manually resetting the scene, to learn from unstructured, reset-free, teleoperated *play* data [16]. Another orthogonal line of work tackles data inefficiency by using pre-trained image representations [29], [6], [30] to bootstrap downstream task learning, which we also leverage in this work.

We propose a novel hierarchical approach that combines the strengths of both paradigms to learn language-conditioned, task-agnostic, long-horizon policies from high-dimensional camera observations. Inspired by the line of work that decomposes robot manipulation into semantic and spatial pathways [12], [13], [6], we propose leveraging a self-supervised affordance model from unstructured data that guides the robot to the vicinity of actionable regions referred in language instructions. Once inside this area, we switch to a single multi-task 7-DoF language-conditioned visuomotor policy, trained also from offline, unstructured data.

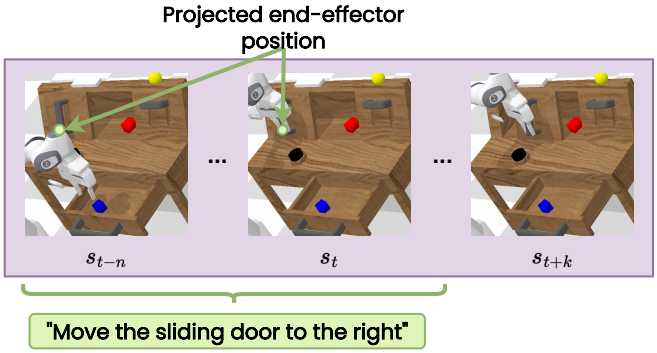

Fig. 2: Visualization of the procedure to extract language-conditioned visual affordances from human teleoperated unstructured, free-form interaction data. We leverage the gripper open/close signal during teleoperation to project the end-effector into the camera images to detect affordances in undirected data.

III. Method

We decompose our approach into three main steps. First we train a language-conditioned affordance model from unstructured, teleoperated data to predict 3D locations of an object that affords an input language instruction (Section III-A). Second, we leverage model-based planning to move towards the predicted location and switch to a local language-conditioned, learning-based policy π_{free} to interact with the scene (Section III-C). Third, we show how HULC++ can be used together with large language models (LLMs) for decomposing abstract language instructions into a sequence of feasible, executable subtasks (Section III-D).

Formally, our final robot policy is defined as a mixture:

$$\begin{aligned}\pi(a \mid s, l) \quad = \quad & (1-\alpha(s,l)) \cdot \pi_{mod}(a \mid s) \\ & +\alpha(s,l) \cdot \pi_{free}(a \mid s, l) \end{aligned} \tag{1}$$

Specifically, we use the pixel distance between the projected end-effector position I_{tcp} and the predicted pixel from the affordance model I_{aff} to select which policy to use. If the distance is larger than a threshold ϵ, the predicted region is far from the robots current position and we use the model-based policy π_{mod} to move to the predicted location. Otherwise, the end-effector is already near the predicted position and we keep using the learning-based policy π_{free}. Thus, we define α as:

$$\alpha(s,l) \quad = \quad \begin{cases} 0, & \text{if } |I_{aff} - I_{tcp}| > \epsilon \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

As the affordance prediction is conditioned on language, each time the agent receives a new instruction, our agent decides which policy to use based on $\alpha(s,l)$. Restricting the area where the model-free policy is active to the vicinity of regions that afford human-object interactions has the advantage that it makes it more sample efficient, as it only needs to learn local behaviors.

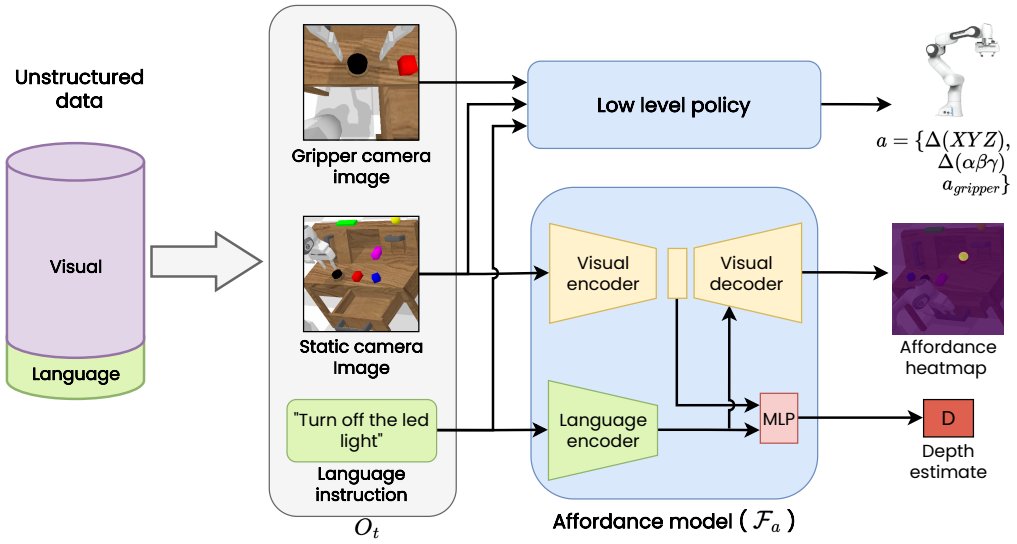

Fig. 3: Overview of the system architecture. HULC++ first processes a language instruction and an image from a static camera to predict the afforded region and guides the robot to its vicinity. Once inside this area, we switch to a language-conditioned imitation learning agent that receives RGB observations from both a gripper and a static camera, and learns 7-DoF goal-reaching policies end-to-end. Both modules learn from the same free-form, unstructured dataset and require as little as 1% of language annotations.

A. *Extracting Human Affordances from Unstructured Data*

We aim to learn an affordance model $\mathcal{F}_a$ that can predict a world location when given a natural language instruction. Unlike prior affordance learning methods that require manually drawn segmentation masks [31], we automatically extract affordances from unstructured, human teleoperated play data [16]. Leveraging play data has several advantages: it is cheap and scalable to collect, contains general behavior, and is not random, but rather structured by human knowledge of affordances. Concretely, play data consists of a long unsegmented dataset $\mathcal{D}$ of semantically meaningful behaviors provided by users teleoperating the robot without a specific task in mind. The full state-action stream $\mathcal{D} = \{(s_t, a_t)_{t=0}^{\infty}\}$ is relabeled to treat the preceding states and actions as optimal behaviour to reach a visited state [16]. Additionally, we assume that a small number of random sequences, less than 1% of the dataset, are annotated with a language instruction describing the task being completed in the sequence.

In order to extract visual affordances from unstructured data, we use the gripper action as a heuristic to discover elements of the scene that are relevant for task completion. Consider the following scenario: a random sequence $\tau = \{(s_0, a_0), ..., (s_k, a_k)\}$, where k denotes the window size, is annotated with a language instruction $s_g = l$. If for any state s_i in the sequence, the action a_i contains a gripper closing signal, we assume that there is an object that is needed for executing the task l at the position of the end-effector. To learn a visuo-lingual affordance model, we project the end-effector world position to the camera images to obtain a pixel p_t, and we annotate the previous frames with said pixel and the language instruction l, as shown in Figure 2. Intuitively, this allows the affordance model to learn to predict a pixel corresponding to an object that is needed for completing the task l.

During test time, given a predicted pixel location, assuming an existing camera calibration, depth information is needed to compute the 3D position where the model-based policy should move to. Instead of relying on the sensory depth observations, our model is trained to produce an estimated depth, by using the position of the end-effector during the gripper closing as supervision. A key advantage of our formulation is that by predicting the depth from visuo-lingual features, our model can better adapt to partial occlusions that might occur in the scene.

B. *Language-Conditioned Visual Affordances*

Our visuo-lingual affordance model, see Figure 3, consists of an encoder decoder architecture with two decoder heads. The first head predicts a distribution over the image, representing each pixels likelihood to be an afforded point. The second head predicts a Gaussian distribution from which the corresponding predicted depth is sampled. Both heads share the same encoder and are conditioned on the input language instruction. Formally, given an input consisting of a visual observation I and a language instruction l, the affordance model $\mathcal{F}_a$ produces an output o of (1) a pixel-wise heatmap $A \in \mathbb{R}^{H \times W}$, indicating regions that afford the commanded task and (2) a corresponding depth estimate d. We denote this mapping as $\mathcal{F}_a(I, l) \mapsto o = (A, d)$.

1) Visual Module: The visual prediction module produces a heatmap A given an input (I_t, l_t). To train it, we apply a softmax function over all the pixels of A. This results in a distribution V over the image where the sum of all the pixel values equals to one.

$$V = \text{softmax}(A) = \frac{\exp(a_i)}{\sum_{j=1}^{N} \exp(a_j)} \tag{3}$$

Similarly, the target T is constructed with the same shape as V, by initializing all its values to zero. Then, we generate a binary one-hot pixel map with the pixel of the projected

position that corresponds to the current state input. Finally, we optimize the visual prediction module with the cross-entropy loss:

$$\mathcal{L}_{aff} = -\sum_{i=1}^{N} t_i \log v_i, \tag{4}$$

where $t_i \in T$ and $v_i \in V$. This optimization scheme [32] allows the visual module to learn a multimodal belief over the image, where the pixel with the highest value denotes the most likely image location given the input. During inference, we use the dense pixelwise output prediction A to select a pixel location I_i:

$$I_i = \underset{(u,v)}{\operatorname{argmax}} V((u,v) \mid (I,l)) \tag{5}$$

The affordance prediction follows a U-Net [33] architecture, where we repeatedly apply language-conditioning to three of the decoder layers after the bottleneck, taking inspiration from LingUNet [34].

2) Depth Module: As aforementioned, we can compute a target for the depth module by transforming the pixel of interest p_t to the camera frame to obtain p_t^{cam}, where the z coordinate of this point corresponds to the ground truth depth $p_{t,z}^{cam}$. Although we compute the true value, typical depth sensors present measurement errors. Therefore, in order to design a system that models the depth error, we use the ground truth depth information to train a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ by maximizing the log likelihood.

$$\mathcal{L}_{depth} = \frac{1}{2}\left(\log \sigma^2 + \frac{(y-\mu)^2}{\sigma^2}\right) \tag{6}$$

As shown in Figure 3, the depth module consists of a set of linear layers that take as input the encoded visuo-lingual features. Here, the language-conditioning is done by concatenating the natural language encoding to the first two layers of the multilayer perceptron. The output of the network are the parameters of a Gaussian distribution $d \sim N(\mu, \sigma)$, which is sampled during inference to obtain the depth prediction d. The total loss function used to train the full affordance model is defined as a weighted combination of the affordance module and depth prediction module losses:

$$\mathcal{L} = \beta \mathcal{L}_{aff} + (1-\beta)\mathcal{L}_{depth} \tag{7}$$

C. Low-Level Language-Conditioned Policy

In order to interact with objects, we learn a goal-conditioned policy $\pi_\theta(a_t \mid s_t, l)$ that outputs action $a_t \in \mathcal{A}$, conditioned on the current state $s_t \in \mathcal{S}$ and free-form language instruction $l \in \mathcal{L}$, under environment dynamics $\mathcal{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. We note that the agent does not have access to the true state of the environment, but to visual observations. We model the low-level policy with a general-purpose goal-reaching policy based on HULC [10] and trained with multi-context imitation learning [9]. We leverage the same, long unstructured dataset $\mathcal{D}$ of semantically meaningful behaviors provided by users we previously utilized to learn affordances in Section III-A. In order to learn task-agnostic control, we leverage goal relabeling [35], by feeding these short horizon goal image conditioned demonstrations into a simple maximum likelihood goal conditioned imitation objective:

$$\mathcal{L}_{LfP} = \mathbb{E}_{(\tau, s_g)\sim D_{play}}\left[\sum_{t=0}^{|\tau|} \log \pi_\theta(a_t \mid s_t, s_g)\right] \tag{8}$$

However, when learning language-conditioned policies $\pi_\theta(a_t \mid s_t, l)$ it is not possible to relabel any visited state s to a natural language goal, as the goal space is no longer equivalent to the observation space. Lynch *et al.* [9] showed that pairing a small number of random windows with language after-the-fact instructions, enables learning a single language-conditioned visuomotor policy that can perform a wide variety of robotic manipulation tasks. The key insight here is, that solving a single imitation learning policy for either goal image or language goals, allows for learning control mostly from unlabeled play data and reduces the burden of language annotation to less than 1% of the total data. Concretely, given multiple contextual imitation datasets $\mathcal{D} = \{D^0, D^1, \ldots, D^K\}$, with different ways of describing tasks, multi-context imitation learning trains a single latent goal conditioned policy $\pi_\theta(a_t \mid s_t, z)$ over all datasets simultaneously.

D. Decomposing Instructions with LLMs

Guiding the robot to areas afforded by a language instruction with the affordance model and then leveraging the low-level policy to execute the task, enables in principle to chain several language instructions in a row. Although natural language provides an intuitive and scalable way for task specification, it might not be practical to have to continually input low level language instructions, such as "open the drawer", "now pick up the pink block and place it inside the drawer", "now pick up the yellow block and place it inside

```
state = 'drawer_open': False, 'blocks_on_table': ['red'],
'buttons_on': ['green']
# put away the red block.
open_drawer()
pick_and_place('red', 'drawer')
close_drawer()
...
state = 'drawer_open': False, 'blocks_on_table': [],
'buttons_on': ['yellow']
# turn off the lights.
push_button('yellow')
state = 'drawer_open': False, 'blocks_on_table': ['red',
'green', 'blue'], 'buttons_on': ['green', 'yellow']
# tidy up the workspace and turn off all the lights
open_drawer()
pick_and_place('red', 'drawer')
pick_and_place('green', 'drawer')
pick_and_place('blue', 'drawer')
close_drawer()
push_button('green')
push_button('yellow')
```

Fig. 4: Example prompt to decompose abstract instructions into sequences of subtasks. Prompt context is in gray, input task commands are magenta, and generated outputs are highlighted.

Training data	Method	Language Finetuned	Tasks completed in a row					
			1	2	3	4	5	Avg. Len.
100 %	Ours + R3M	✔	**93% (0.007)**	**79% (0.002)**	**64% (0.008)**	**52% (0.003)**	**40% (0.001)**	**3.30 (0.006)**
	Ours	✔	89% (0.014)	71% (0.018)	55% (0.025)	43% (0.028)	33% (0.015)	2.93 (0.090)
	HULC	✔	84% (0.009)	66% (0.023)	50% (0.023)	38% (0.030)	29% (0.029)	2.69 (0.113)
	HULC-original	✗	82.7% (0.3)	64.9% (1.7)	50.4% (1.5)	38.5% (1.9)	28.3% (1.8)	2.64 (0.05)
50 %	Ours + R3M	✔	88% (0.030)	69% (0.032)	52% (0.016)	38% (0.013)	27% (0.004)	2.75 (0.2705)
	Ours	✔	84% (0.035)	63% (0.061)	44% (0.062)	32% (0.064)	21% (0.053)	2.45 (0.274)
	HULC	✔	79% (0.031)	54% (0.067)	37% (0.072)	26% (0.066)	17% (0.045)	2.15 (0.278)
25 %	Ours + R3M	✔	78% (0.009)	56% (0.006)	36% (0.011)	23% (0.016)	14% (0.009)	2.068 (0.046)
	Ours	✔	81% (0.007)	56% (0.006)	37% (0.008)	24% (0.017)	15% (0.016)	2.15 (0.049)
	HULC	✔	72% (0.045)	45% (0.026)	27% (0.022)	17% (0.022)	9% (0.026)	1.72 (0.135)

TABLE I: Performance of our model on the D environment of the CALVIN Challenge and ablations, across 3 seeded runs.

the drawer" to perform a tidy up task for instance. Ideally, we would like to give the robot an abstract high level instruction, such as "tidy up the workspace and turn off all the lights". Similar to Zeng *et. al.* [7], we use a standard pre-trained LLM, to decompose abstract language instructions into a sequence of feasible subtasks, by priming them with several input examples of natural language commands (formatted as comments) paired with corresponding robot code (via few-shot prompting). We leverage the code-writing capabilities of LLMs [36], [3] to generate executable Python robot code that can be translated into manipulation skills expressed in language. For example, the skill expressed by the API call push_button('green'), is translated into "turn on the green light" and then used to execute an inference of the policy. The only assumption we make is that the scene description fed into the prompt matches the environments state. We show a example prompt in Figure 4.

IV. Experiments

Our experiments aim to answer the following questions: 1) Does integrating the proposed visuo-lingual affordance model improve performance and data-efficiency on following language instructions over using an end-to-end model? 2) Is the proposed method applicable to the real world? 3) When paired with LLMs, can the agent generalize to new behaviors, by following the subgoals proposed by the LLM?

A. Simulation Experiments

Evaluation Protocol. We design our experiments using the environment D of the CALVIN benchmark [8], which consists of 6 hours of teleoperated undirected play data that might contain suboptimal behavior. To simulate a real-world scenario, only 1% of that data contains crowd-sourced language annotations. The goal of the agent in CALVIN is to solve up to 1000 unique sequence chains with 5 distinct subtasks instructed via natural language, using onboard sensing. During inference, the agent receives the next subtask in a chain only if it successfully completes the current one.

Results and Ablations. We compare our approach of dividing the robot control learning into a high-level stream that grounds semantic concepts and a low-level stream that grounds 3D spatial interaction knowledge against HULC [10], a state-of-the-art end-to-end model that learns general skills grounded on language from play data. For a fair comparison, we retrain the original HULC agent to also finetune the language encoder, as this gives a boost in average sequence length from 2.64 to 2.69. We observe in Table I, that when combined with our affordances model, the performance increases to an average sequence length of 2.93. By decoupling the control into a hierarchical structure, we show that performance increases significantly. Moreover, when initializing our affordance model with pretrained weights of R3M [29], a work that aims to learn reusable representations for learning robotic skills, HULC++ sets a new state of the art with an average sequence length of 3.30.

In order to study the data-efficiency of our proposed approach, we additionally compare our model on smaller data splits that contain 50% and 25% of the total play data. Our results indicate that our approach is up to 50% more sample efficient than the baseline. As it might be difficult to judge how much each module contributes to the overall sample-efficiency gains, we investigate the effect of pairing our affordance model trained on 25% of the data with a low-level policy trained on the full dataset. We report little difference, with an average sequence length of 2.92.

B. Real-Robot Experiments

System Setup. We validate our results with a Franka Emika Panda robot arm in a 3D tabletop environment that is inspired by the simulated CALVIN environment. This environment consists of a table with a drawer that can be opened and closed and also contains a sliding door on top of a wooden base, such that the handle can be reached by the end-effector. Additionally, the environment also contains three colored light switches and colored blocks. We use an offline dataset from concurrent work [37], consisting of 9 hours of unstructured data and that was collected by asking participants to teleoperate the robot without performing any specific task. Additionally, we annotate less than 1% of the total data with language, 3605 windows concretely, by asking human annotators to describe the behavior of randomly sampled windows of the interaction dataset. The dataset contains over 25 distinct manipulation skills. We note that learning such a large range of diverse skills in the real world, from unstructured, reset-free and possibly suboptimal data, paired with less than 1% of it being annotated with language, is extremely challenging. Additionally, this setting contains an order of magnitude less data than related approaches [4].

Task\Method	Ours	HULC [10]	BC-Z [4]
Lift the block on top of the drawer	**70**%	60%	20%
Lift the block inside the drawer	**70**%	50%	10%
Lift the block from the slider	**40**%	20%	10%
Lift the block from the container	**70**%	60%	20%
Lift the block from the table	**80**%	70%	40%
Place the block on top of the drawer	**90**%	50%	30%
Place the block inside the drawer	**70**%	40%	20%
Place the block in the slider	**30**%	20%	0%
Place the block in the container	**60**%	30%	20%
Stack the blocks	**50**%	30%	0%
Unstack the blocks	**50**%	40%	0%
Rotate block left	**70**%	40%	10%
Rotate block right	**70**%	50%	20%
Push block left	**70**%	50%	20%
Push block right	**60**%	50%	10%
Close drawer	**90**%	70%	20%
Open drawer	**80**%	50%	10%
Move slider left	**70**%	10%	0%
Move slider right	**70**%	30%	0%
Turn red light on	**50**%	30%	0%
Turn red light off	**40**%	20%	0%
Turn green light on	**70**%	60%	10%
Turn green light off	**70**%	50%	10%
Turn blue light on	**70**%	50%	10%
Turn blue light off	**70**%	30%	10%
Average over tasks	**65.2**%	42.4%	16.6%
Average no. of sequential tasks	6.4	2.7	1.3

TABLE II: The average success rate of the multi-task goal-conditioned models running roll-outs in the real world.

Baselines. To study the effectiveness of our hierarchical architecture, we benchmark against two language-conditioned baselines: HULC [10] and BC-Z [4]. The first baseline serves to evaluate the influence of leveraging the affordance model to enable a hierarchical decomposition of the control loop, as the low-level policy is tailored to learning task-agnostic control from unstructured data. The BC-Z baseline, on the other hand, is trained only on the data that contains language annotation and includes the proposed auxiliary loss that predicts the language embeddings from the visual ones for better aligning the visuo-lingual skill embeddings [4]. For a fair comparison, all models have the same observation and action space, and have their visual encoders for the static camera initialized with pre-trained ResNet-18 R3M features [29]. For HULC++ this entails both, the visual encoder for the affordance model and the visual encoder for the static camera of the low-level policy. The encoder for the gripper camera is trained from scratch.

Evaluation. We start off by evaluating the success rate of the individual skills conditioned with language. After training the models with the offline play dataset, we performed 10 rollouts for each task using neutral starting positions to avoid biasing the policies through the robot's initial pose. This neutral initialization breaks correlation between initial state and task, forcing the agent to rely entirely on language to infer and solve the task. We recorded the success rate of each model in Table II. We observe that the BC-Z baseline has near zero performance in most tasks, due to insufficient demonstrations. HULC is more capable, as it leverages the full play dataset with an average of 42.4% over 10 rollouts, but struggles with long-horizon planning, as do most end-to-end agents trained with imitation learning. Overall, HULC++ is more capable with an average of 65.2% success rate over 25 distinct manipulation tasks, demonstrating the effectiveness of incorporating a semantic viso-lingual affordance prior for decoupling the control into a hierarchical structure.

Finally, we evaluate how many tasks in a row each method can follow in the real world, by leveraging GPT-3 to generate sequences of subgoals for abstract language inputs, such as "tidy up the workspace and turn off the lights". We report an average number of 6.4 subgoals being executed for our method, while the baselines tend to fail after completing 2 to 3 subgoals. See the supplementary video for qualitative results that showcase the diversity of tasks and the long-horizon capabilities of the different methods. Overall, our results demonstrate the effectiveness of our approach to learn sample-efficient, language-conditioned policies from unstructured data by leveraging visuo-lingual affordances.

V. Conclusion and Limitations

In this paper, we introduced a novel approach to efficiently learn general-purpose, language-conditioned robot skills from unstructured, offline and reset-free data containing as little as 1% of language annotations.The key idea is to extract *language-conditioned affordances* from diverse human teleoperated data to learn a semantic prior on where in the environment the interaction should take place given a natural language instruction. We distill this knowledge into an interplay between model-based and model-free policies that allows for a sample-efficient division of the robot control learning, substantially surpassing the state of the art on the challenging language-conditioned robot manipulation CALVIN benchmark. We show that when paired with LLMs to translate abstract natural language instructions into sequences of subgoals, HULC++ is capable of completing long-horizon, multi-tier tasks the real world, while requiring an order of magnitude less data than previous approaches.

While the experimental results are promising, our approach has several limitations. First, when sequencing skills in the real world, an open question is tracking task progress in order to know when to move to the next task. In this work, we acted with a fixed time-horizon for sequencing tasks in the real world, implicitly assuming that all tasks take approximately the same timesteps to complete. Second, the code-generation module to translate abstract language inputs to sequences of subgoals assumes that the prompted scene description matches the environment's state, which could be automated by integrating a perceptual system [2]. Finally, an exciting area for future work may be one that not only grounds actions with language models, but also explores improving the language models themselves by incorporating real-world robot data [38].

Acknowledgment

We thank Andy Zeng for fruitful discussions on few-shot prompting of LLMs. This work has been supported partly by the German Federal Ministry of Education and Research under contract 01IS18040B-OML.

REFERENCES

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

[2] 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," *arXiv preprint arXiv:2207.05608*, 2022.

[3] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," *arXiv preprint arXiv:2209.07753*, 2022.

[4] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[5] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[6] 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.

[7] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, *et al.*, "Socratic models: Composing zero-shot multimodal reasoning with language," *arXiv preprint arXiv:2204.00598*, 2022.

[8] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7327–7334, 2022.

[9] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," in *RSS*, 2021.

[10] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 11 205–11 212, 2022.

[11] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

[12] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3750–3757.

[13] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA, 2022.

[14] L. P. Kaelbling, "Learning to achieve goals," in *IJCAI*, 1993, pp. 1094–1099.

[15] D. I. A. Team, J. Abramson, A. Ahuja, A. Brussee, F. Carnevale, M. Cassin, F. Fischer, P. Georgiev, A. Goldin, T. Harley, *et al.*, "Creating multimodal interactive agents with imitation and self-supervised learning," *arXiv preprint arXiv:2112.03763*, 2021.

[16] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," *Conference on Robot Learning (CoRL)*, 2019.

[17] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 25–55, 2020.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[19] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[20] R. Paul, J. Arkin, N. Roy, and T. M Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators," in *RSS*, 2016.

[21] M. Shridhar and D. Hsu, "Interactive visual grounding of referring expressions for human-robot interaction," in *RSS*, 2018.

[22] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively picking real-world objects with unconstrained spoken language instructions," in *ICRA*, 2018.

[23] T. Nguyen, N. Gopalan, R. Patel, M. Corsaro, E. Pavlick, and S. Tellex, "Robot object retrieval with contextual natural language queries," in *RSS*, 2020.

[24] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. La, and N. Zheng, "Invigorate: Interactive visual grounding and grasping in clutter," in *RSS*, 2021.

[25] O. Mees and W. Burgard, "Composing pick-and-place tasks by grounding language," in *ISER*, 2021.

[26] W. Liu, C. Paxton, T. Hermans, and D. Fox, "Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6322–6329.

[27] S. Nair, E. Mitchell, K. Chen, B. Ichter, S. Savarese, and C. Finn, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *CoRL*, 2021.

[28] V. Blukis, R. Knepper, and Y. Artzi, "Few-shot object grounding and mapping for natural language robot instruction following," in *Conference on Robot Learning*. PMLR, 2021, pp. 1829–1854.

[29] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[30] W. Yuan, C. Paxton, K. Desingh, and D. Fox, "Sornet: Spatial object-centric representations for sequential manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 148–157.

[31] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[32] 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.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[34] D. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3d environments with visual goal prediction," *arXiv preprint arXiv:1809.00786*, 2018.

[35] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[36] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[37] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard, "Latent plans for task agnostic offline reinforcement learning," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, Auckland, New Zealand, 2022.

[38] Y. Bisk, A. Holtzman, J. Thomason, J. Andreas, Y. Bengio, J. Chai, M. Lapata, A. Lazaridou, J. May, A. Nisnevich, *et al.*, "Experience grounds language," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8718–8735.

[39] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *EMNLP*, 2019.

APPENDIX

A. Affordance Model Ablations

In this section we perform more ablation studies of our method on the CALVIN environment. Concretely, to better study the data-efficiency of our method, we perform ablation studies by pairing affordance and policy models trained with 25% and 100% of the training data. We observe in Table III that the performance does not change much, demonstrating the sample-efficiency of the visuo-lingual affordance model.

Training data		Tasks completed in a row					
Policy	Affordance	1	2	3	4	5	Avg. Len.
25%	25%	81%	56%	37%	24%	15%	2.15
25%	100%	82%	58%	38%	24%	15%	2.18
100%	100%	89%	71%	55%	43%	33%	2.93
100%	25%	89%	72%	55%	42%	31%	2.92

TABLE III: Ablation of our approach trained with different data quantities for the affordance and low-level policy networks.

Next, we perform similar ablation studies for the depth prediction module trained on 25%, 50% and 100% of the dataset. We report two metrics: mean pixel distance error and the mean depth error. We plot the pixel distance error for the validation split in Figure 5, and observe that the error increases only in $\sim$ 3 pixels when training the model with 25% of the data instead of the full dataset.

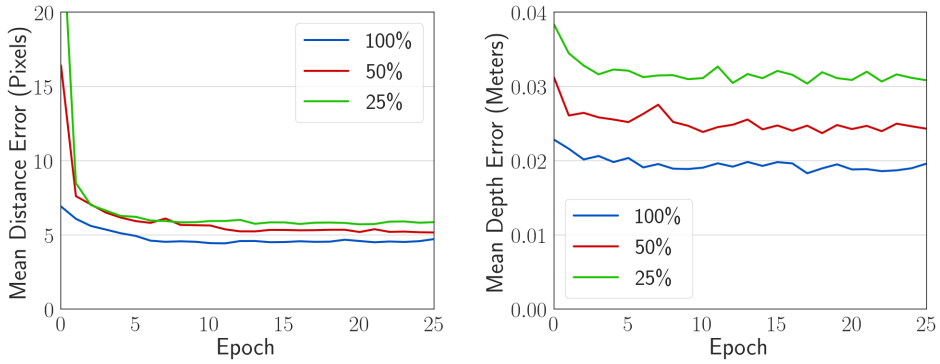

Fig. 5: Pixel distance and depth validation error for the affordance model's depth prediction module trained with different data quantities.

Similarly, we observe that the depth error increases in $\sim$2 cm when training the model with 25% of the data instead of the full dataset. These results show that the proposed visuo-lingual affordance model is very sample-efficient, making it attractive for real world robotic applications, where collecting robot interaction data and annotating them with natural language might be costly.

B. Hyperparameters

1) Low-Level Policy: To learn the low-level policy we train the model using 8 gpus with Distributed Data Parallel (DDP). Throughout training, we randomly sample windows between length 16 and 32 and pad them until reaching the max length of 32 by repeating the last observation and an action equivalent to keeping the end effector in the same state. We use a batch size of 64, which with DDP results in an effective batch size of 512. We train using the Adam optimizer with a learning rate of $2e-4$. The latent plan is a vector of categorical variables, concretely we use 32 categoricals with 32 classes each. The KL loss weight β is $1e-2$ and uses KL balancing. Concretely, we minimize the KL loss faster with respect to the prior than the posterior by using different learning rates, $\alpha = 0.8$ for the prior and $1-\alpha$ for the posterior. In order to encode raw text into a semantic pre-trained vector space, we leverage the paraphrase-MiniLM-L3-v2 model [39], which distills a large Transformer based language model and is trained on paraphrase language corpora that is mainly derived from Wikipedia. It has a vocabulary size of 30,522 words and maps a sentence of any length into a vector of size 384.

For the real world experiments, the static camera RGB images have a size of 150×200, we then apply a color jitter transform with contrast of 0.05, a brightness of 0.05 and a hue of 0.02. Finally, we use the values for the pretrained R3M normalization, i.e., mean $= [0.485, 0.456, 0.406]$ and a standard deviation, std $= [0.229, 0.224, 0.225]$. For the gripper camera RGB image, we resize the image from 200×200 to 84×84, we then apply a color jitter transform with contrast of 0.05, a brightness of 0.05 and a hue of 0.02. Then we perform stochastic image shifts of $0-4$ pixels to the and a bilinear interpolation is applied on top of the shifted image by replacing each pixel with

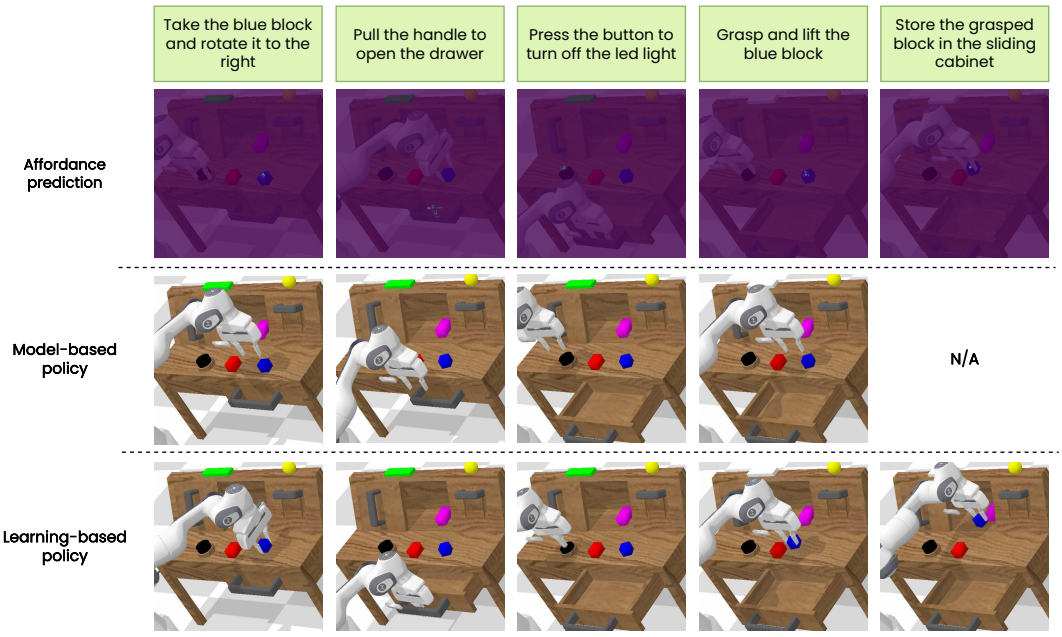

Fig. 6: Visualization of a sample rollout for our approach in the CALVIN environment. For each column, we show the input language instruction, the predicted affordance, the reached state by the model-based policy after executing the command, and the final reached state by the learning-based policy for completing the requested task.

the average of the nearest pixels. Finally, we normalize the input image to have pixels with float values between -1.0 and 1.0.

2) Affordance Model: For the affordance model we use a Gaussian distribution to model the depth estimate. We normalize the depth values with the dataset statistics. We train the network end-to-end using a learning rate of $1e-4$ with the Adam optimizer and a batch size of 32 in a single GPU. During training, we resize the input images to $224 \times 224 \times 3$, apply stochastic image shifts of 5 pixels and apply a color jitter transform with contrast of 0.05, a brightness of 0.05 and a hue of 0.02 as data augmentation. We use the paraphrase-MiniLM-L3-v2 pretrained model [39] to encode raw text into a semantic vector space. In our experiments, we observed that the affordance model starts learning accurate predictions for the 2d pixel affordance faster than making proper depth estimations. In order to balance both tasks, we define a higher weight for the depth loss $\mathcal{L}_{depth}$ than for the affordance loss $\mathcal{L}_{aff}$ by setting β to 0.1.

C. Qualitative Results

In order to better understand how the visuo-lingual affordance model, the model-based policy and the model-free policy interact with each other, we visualize a rollout for one chain of the CALVIN benchmark in Figure 6. Given a language instruction and a visual observation, the visuo-lingual affordance model predicts a location which affords the given instruction. The model-based policy guides the robot to the vicinity of the afforded region. Once inside this area, we switch to the model-free language-conditioned visuomotor policy that interacts with the environment.