Dynamic RDMM: Scalable, Controllable Dataset Generation for Instruction-Grounded Robot Learning

blank

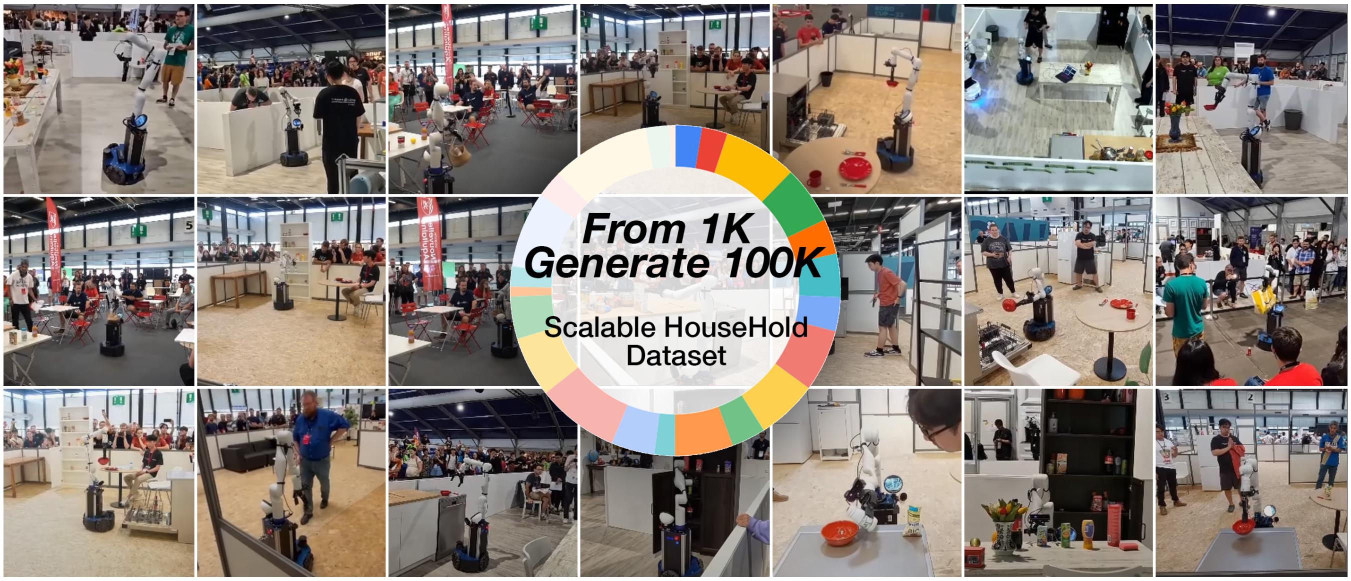

Fig. 1: **Dataset Overview.** Dynamic RDMM comprises 23 household tasks across manipulation, navigation, and interaction. It enables scalable, curriculum-driven data generation with precise control over difficulty and task type distribution. A model trained on this dataset was successfully deployed at RoboCup@Home.

***Abstract*— Robotic decision-making with large language models (LLMs) is increasingly constrained by the scarcity of datasets that are both linguistically expressive and grounded in symbolic robotic actions. We introduce the Dynamic RDMM Dataset (D-RDMM), a controllable, text-to-text dataset generator that maps natural-language instructions to structured action programs across 23 real-world household tasks.**

D-RDMM is built using a two-stage generation process. First, *hierarchical template expansion* recursively constructs multi-step task descriptions from nested logic templates, enabling variable instruction complexity and compositional diversity. Second, *constraint-aware content generation* fills these templates using curated embeddings (verbs, rooms, objects, names) while enforcing semantic and physical validity. This process produces 1,800 expert-verified instruction–action pairs and can be scaled to over 100,000 diverse, syntactically and semantically valid examples.

Crucially, D-RDMM supports task rebalancing and adjustable instruction difficulty, allowing researchers to generate lightweight commands (e.g.,“go to the kitchen”) or compound sequences (e.g., “deliver an item to Kai and follow him until he sits”). This makes D-RDMM a practical tool for curriculum learning, ablation studies, and controlled benchmarking.

We validate by fine-tuning LLaMA-3-8B, Mistral-7B, and Qwen-0.5 models, achieving 93% accuracy and outperforming ChatGPT-4o baselines. In real-world deployment at RoboCup@Home, D-RDMM-trained models successfully executed unseen multi-step commands in noisy, natural environments. We release all templates, Dataset, code, and generation tools to support reproducible, instruction-grounded robot learning.

Project website:blank.blank/blank

I. Introduction

Robotic systems that can follow natural-language instructions promise to make intelligent agents more accessible and useful in everyday environments. Recent advances in instruction-tuned Large Language Models (LLMs) have demonstrated compelling capabilities in translating user commands into structured plans, enabling robots to reason over complex task sequences. However, achieving robust, real-world instruction-following behavior remains challenging—especially when LLMs are deployed on embodied agents operating in noisy, dynamic household environments.

One of the central obstacles is the *shortage of domain-specific, symbolically grounded training datasets*. General-purpose web-scale corpora offer vast linguistic variety, but lack the structure, action alignment, and feasibility constraints needed for robotic execution. Conversely, existing robotics datasets—such as ALFRED [1] and TEACh [2] set—provide grounded examples, but are typically static, narrow in scope, or costly to expand. As a result, robot learning pipelines suffer from a *data bottleneck*: when models underperform on a specific task, researchers have no quick, controllable way to inject targeted supervision without manual annotation. This gap is especially apparent in open competition settings such as RoboCup@Home [3].

In this work, we introduce the **Dynamic RDMM Dataset**—and more importantly, the **controllable dataset generation engine** that produces it. RDMM is a text-to-text dataset in which each sample maps a natural-language instruction to a structured action sequence, composed of symbolic primitives (e.g., *MoveTo(kitchen)*, *Pickup(milk)*, *Respond("I'm here")*). These sequences can be interpreted directly by robotic planners or mapped onto platform-specific skills, making the dataset readily deployable. Unlike previous datasets, RDMM is not a static benchmark—it is a parametric data engine enabling controllable, symbolic instruction-action supervision at scale.

The core of D-RDMM is a two-stage generation process:

1) **Hierarchical template expansion**: A compact YAML library encodes 23 common household tasks (e.g., guiding a person, delivering an item, answering a question) using nested logic structures. These templates recursively expand into semantically valid multi-step instructions.
2) **Dynamic content generation**: Templates are populated with verbs, object types, room names, and personal references drawn from curated embedding dictionaries. Semantic constraints ensure physical plausibility (e.g., "put the pizza on the table" is allowed; "put the microwave on the sandwich" is not).

This system offers several advantages critical for robot learning:

- **Scalability**: From just 23 task templates, D-RDMM can generate over 100,000 valid instruction–action pairs.
- **Task controllability**: Researchers can dynamically rebalance data by adjusting task-specific weights without reauthoring templates.
- **Curriculum and ablation support**: Lexical variation, compositional complexity, and object diversity can all be programmatically adjusted to match experimental needs.

To validate D-RDMM, we fine-tune three open-source LLMs (LLaMA-3-8B, Mistral-7B, Qwen-0.5) using only the 1,800-sample seed set. These models achieve 93% accuracy on held-out samples and generalize to previously unseen instructions. Deployed on a mobile robotic platform at *RoboCup@Home*, D-RDMM-trained models reliably execute composite instructions in a real-world, multi-user environment.

Our contributions are three-fold:

- A **controllable dataset generation framework** for instruction-following tasks in robotic environments;
- The **Dynamic RDMM Dataset**, featuring 1,800 expert-validated pairs with the capacity to scale to 100k+ examples;
- **Empirical validation** showing that small, well-structured datasets can train LLMs to reason over symbolic action plans and generalize in real-world robotic applications.

By treating dataset generation as a parameterized process rather than a static artifact, we turn data design into a flexible tool in the robot learning loop—paving the way for faster, more adaptive development of LLM-based instruction-following systems.

II. Related work

Research on natural language instruction understanding for service robots spans a diverse set of approaches, which can be broadly grouped into: (1) semantic parsing and action prediction methods, (2) scalable dataset generation strategies, (3) multimodal instruction-following systems based on large pretrained models, and (4) LLM-based methods for robotic instruction understanding.

Semantic Parsing for Robot Instructions: A wide range of models have been proposed to translate natural language commands into executable action sequences or structured representations. For example, Misra et al. [4] predict low-level actions in simulated 3D environments, while Dong and Lapata [5] introduce hierarchical decoding for mapping sentences to logical forms. Other techniques include syntactic parsing [6], graph-based execution models [7], and transition systems [8]. More recent approaches like GRID [9] use scene graphs to support instruction-driven planning with structured environmental representations. These techniques often provide strong interpretability but depend on domain-specific annotations, limiting generalization.

Our work departs from these by offering a training resource explicitly designed to support structural parsing of robotic commands without domain-specific fine-tuning. We provide compositional, annotated examples aligned with symbolic robotic tasks, enabling more modular and transferable instruction understanding.

Template-Based Dataset Generation: Compositional data generation methods have been used to reduce the manual burden of annotation, such as Neural Module Networks [10] and probabilistic program synthesis [11]. These methods leverage templates with lexical placeholders to create large-scale datasets. Unlike VQA datasets like CLEVR, which focus on visual reasoning in synthetic environments, our dataset emphasizes symbolic outputs grounded in robot-executable actions. While Neural Module Networks decompose tasks into perceptual modules, our dataset targets structural symbolic parsing applicable across different planning pipelines.

Although prior efforts offer scalability, they often lack domain-general symbolic representations tailored for robotics. D-RDMM applies a similar compositional strategy but with emphasis on consistent symbolic outputs, ensuring high annotation quality and combinatorial diversity through structured expansion.

Multimodal Instruction-Following: Benchmarks like ALFRED [1], TEACh [2], RT-1 [16], and PaLM-E [17]

TABLE I: Comparison of approaches for robot natural language instruction understanding

Approach	**Data Gen. Method**	**Expandable**	**Input**	**Real-world**
ALFRED [1]	Hand Annotated	-	Text+Vision	-
TEACh [2]	Hand Annotated	-	Text+Vision	-
SayCan [12]	LLM	-	Text+State	✓
ProgPrompt [13]	LLM	-	Text	✓
GRID [9]	LLM + Scene Graphs	-	Text+Scene	-
ViLaIn [14]	VLM	-	Text+Vision	✓
GRAIL [15]	LLM +PDDL	-	Text+State	-
D-RDMM (ours)	Template + Constraints	✓	Text	✓

integrate visual and linguistic signals to enable end-to-end task execution. RT-2 [18] extends this idea by transferring internet-scale knowledge to robotics, while ViLaIn [14] builds a vision-language interpreter tailored to planning.

Our method contrasts with these end-to-end pipelines by isolating the language parsing component. Unlike ALFRED or TEACh, which measure overall success in visually-rich simulations, we decouple perception and language understanding, enabling symbolic planning and modular error analysis. Compared to RT-series and CoLLIE, which prioritize joint vision-language-action learning, we generate symbolic intermediate representations, supporting integration with existing task planners and broader robotic ecosystems.

LLM-Based Methods for Robotic Instruction Understanding: Large language models have recently shown promise in robotic instruction following. SayCan [12] grounds language in robotic affordances, while ProgPrompt [13] uses prompt engineering to generate symbolic plans. GRAIL [15] extends LLM capabilities by inducing grounded action rules. SayComply [19] builds on SayCan to align instructions with operational constraints, and Teriyaki [20] merges symbolic reasoning with neural networks.

These works demonstrate flexible language understanding but often produce unstructured or implicit outputs. Our approach complements them by providing explicit symbolic supervision that can guide or evaluate LLM outputs. By generating compositional action sequences in formal syntax, our dataset enhances the grounding, interpretability, and reliability of LLM-based robotic systems, especially in safety-critical or high-precision domains.

Comparative Analysis of Approaches: We summarize the differences between D-RDMM and existing methods in Table I.

III. Methods

The **Dynamic RDMM Dataset** is not a static collection, but the output of a controllable, parameterized generation engine. It is constructed through a two-stage algorithmic process that transforms a compact set of YAML templates into thousands of semantically grounded, text-to-text instruction–action pairs. This modular architecture enables researchers to programmatically scale, rebalance, and adapt the dataset to match specific training and evaluation needs in robot learning, summarized in Fig. 2.

In addition to dataset generation, the full end-to-end execution framework—including speech recognition (STT), text-to-speech (TTS), visual perception models, person tracking, and symbolic planning—was deployed on a mobile platform and orchestrated using RDMM-trained language models. This integrated AI stack allows natural language instructions to be grounded in multimodal real-world execution. Details of the complete robotic system, including software and hardware integration, are described in our companion paper [21].

A. Generation Pipeline Overview

The D-RDMM dataset is generated via a two-stage process that supports structured language generation and controllable task complexity.

Stage 1 — Hierarchical Template Expansion. Each task category (e.g., *follow*, *serve*, *guide*) is defined by high-level templates written in YAML. These templates are hierarchical: they include nested references to lower-level subtemplates that describe entities (e.g., "a person wearing item"), actions, or spatial configurations. The system performs recursive expansion, layer by layer, until all placeholders are replaced with terminal placeholders. This process naturally creates variations in instruction complexity:

- **Low-complexity instruction:** *"Follow a person"*
- **Medium-complexity:** *"Follow the person wearing glasses"*
- **High-complexity:** *"Follow the person wearing glasses and deliver the apple juice to them"*

Stage 2 — Dynamic Content Generation. Once the expanded templates contain only atomic placeholders, the system fills them using curated embedding dictionaries for verbs, object classes, rooms, and person names. Combinations are sampled using a task weighting vector w, and filtered through logical rules to eliminate physically implausible instructions (e.g., *Put(microwave, sandwich)* is invalid).

This two-stage pipeline enables large-scale, controllable generation of realistic instruction–action pairs while supporting task rebalancing and complexity tuning for curriculum learning.

B. Template and Ontology Design

Unlike prior RoboCup-style generators, D-RDMM outputs complete instruction–action pairs with symbolic structure, supports balanced sampling, and allows controlled scaling

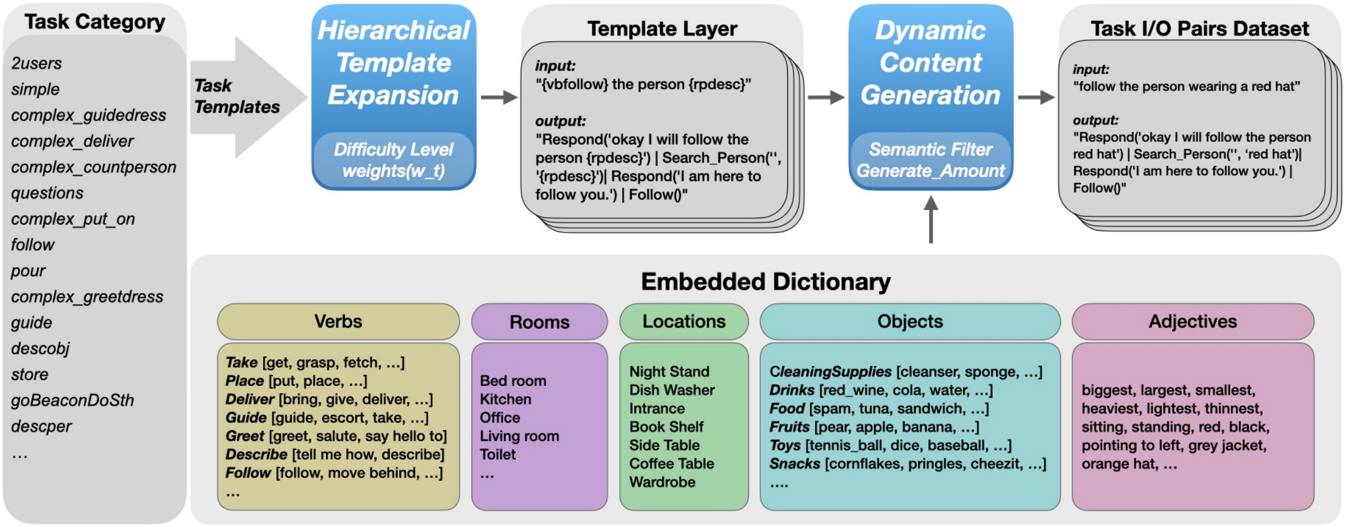
Task Category
2users
simple
complex_guidedress
complex_deliver
complex_countperson
questions
complex_put_on
follow
pour
complex_greetdress
guide
descobj
store
goBeaconDoSth
descper
…
Task Templates
Hierarchical Template Expansion
Difficulty Level weights(w_t)
Template Layer
input:
"{vbfollow} the person {rpdesc}"
output:
"Respond('okay I will follow the person {rpdesc}') | Search_Person('', '{rpdesc}')| Respond('I am here to follow you.') | Follow()"
Dynamic Content Generation
Semantic Filter Generate_Amount
Task I/O Pairs Dataset
input:
"follow the person wearing a red hat"
output:
"Respond('okay I will follow the person red hat') | Search_Person('', 'red hat')| Respond('I am here to follow you.') | Follow()"
Embedded Dictionary
Verbs
Take [get, grasp, fetch, ...]
Place [put, place, ...]
Deliver [bring, give, deliver, ...]
Guide [guide, escort, take, ...]
Greet [greet, salute, say hello to]
Describe [tell me how, describe]
Follow [follow, move behind, ...]
...
Rooms
Bed room
Kitchen
Office
Living room
Toilet
...
Locations
Night Stand
Dish Washer
Intrance
Book Shelf
Side Table
Coffee Table
Wardrobe
Objects
CleaningSupplies [cleanser, sponge, ...]
Drinks [red_wine, cola, water, ...]
Food [spam, tuna, sandwich, ...]
Fruits [pear, apple, banana, ...]
Toys [tennis_ball, dice, baseball, ...]
Snacks [cornflakes, pringles, cheezit, ...]
....
Adjectives
biggest, largest, smallest, heaviest, lightest, thinnest, sitting, standing, red, black, pointing to left, grey jacket, orange hat, ...

Fig. 2: **D-RDMM generation pipeline:** Hierarchical templates are expanded into multi-step instructions, then populated using curated embeddings and filtered for semantic validity. Control parameters like *generate_amount* and task weights w_t enable scalable, balanced generation of instruction–action training pairs.

without additional human annotation, D-RDMM defines an ontology of entities and their affordances:

- **21 locations** (e.g., *desk*, *wardrobe*, *coffee table*)
- **6 room types** (e.g., *kitchen*, *bedroom*, *office*)
- **7 object classes** 50+ items (e.g., *snacks*, *drinks*, *toys*)
- **14 person names** (e.g., *Kai*, *Noah*, *Riley*)

Placeholders are only filled with values that pass semantic filters. For example, *Pour(milk, red bowl)* is disallowed, while *Pour(sandwich, red bowl)* is permitted. Verb–object and object–location pairings are checked against pre-defined grammar constraints and affordance maps.

C. Controllability and Task Rebalancing

A key feature of D-RDMM is that dataset size and task balance are configurable. Researchers specify a task weighting vector $w \in \mathbb{R}^{23}$ and a global limit *generate_amount*. Increasing w_{follow} immediately generates more person-following samples without modifying templates.

This makes D-RDMM particularly well-suited for:

- Curriculum learning (increasing complexity or lexical diversity)
- Task-specific augmentation (targeted fine-tuning on underperforming behaviors)
- Ablation studies (removing or isolating specific instruction types)

D. Dataset Scale and Coverage

The seed release of dataset contains **1,860 expert-verified samples** across 23 task types (see Appendix Table II). By adjusting generation parameters, the same setup can scale to over **100,000 unique samples** in under a minute on a standard CPU.

Each sample references one or more elements from dataset's semantic ontology and exhibits natural linguistic variation in verb choice, object type, and phrasing (e.g., "go behind the person wearing yellow shoes" vs. "follow the person wearing blue pants").

E. Formal Definition

Formally, the dataset is generated as:

$$D = \bigcup_{t \in T} \bigcup_{g \in G_t} \left\{ \text{Apply}(g, \mathbf{e}) \;\middle|\; \mathbf{e} \in \prod_{j=1}^{n} E[p_j] \right\}$$

Where:

- T is the set of task categories.
- G_t is the set of templates for task t.
- $E[p_j]$ is the list of valid substitutions for placeholder p_j.
- $\mathbf{e} = (e_1, \ldots, e_n)$ is a sampled combination from the Cartesian product.
- *Apply* replaces placeholders in g with $\mathbf{e}$ to yield a resolved instruction–action pair.

This definition ensures every generated sample is syntactically correct and grounded in valid robotic semantics.

IV. RDMM Dataset Evaluation and Use Case Validation

We conducted both quantitative evaluation on the D-RDMM dataset and real-world deployment to validate its effectiveness as a training resource for robotic decision-making models. While detailed experiments, training procedures, and benchmarking results of D-RDMM-trained language models are presented in a separate research paper [21], this section summarizes the key findings relevant to the dataset's quality, usability, and practical impact.

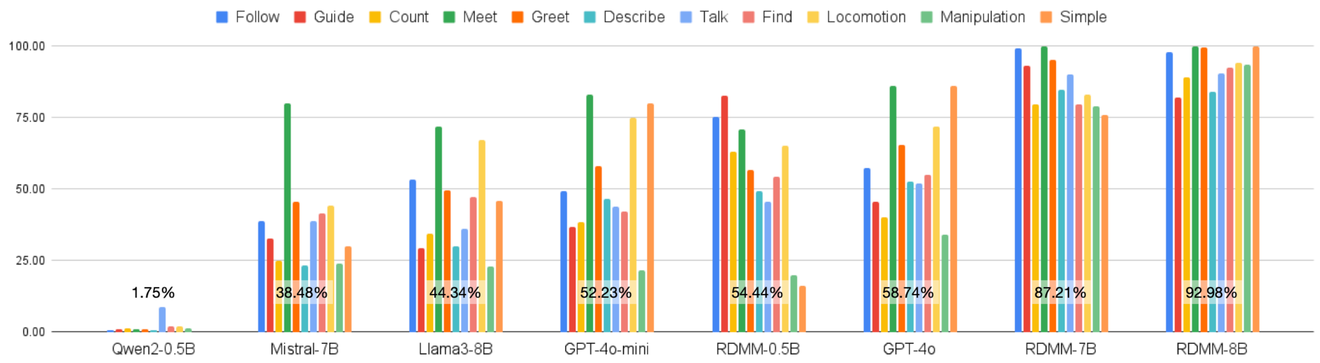

Fig. 3: **Accuracy Comparison.** Evaluation results of D-RDMM-trained LLMs (LLaMA-3, Mistral, Qwen) vs. 20-shot ChatGPT-4o and baselines, measured on held-out samples.

A. *Model Training Setup*

To validate the usability of D-RDMM, we fine-tuned three publicly available large language models: LLaMA3-8B [22], Mistral-7B [23], and Qwen-0.5 [24]. These models were trained end-to-end on the instruction-action pairs generated by D-RDMM across all task types. The models learned to map natural language commands to structured robot control sequences using only text-based input and output.

We additionally evaluated two prompting-based baselines — ChatGPT-4o and ChatGPT-4o-mini — using a 20-shot setup with representative D-RDMM samples. While these models demonstrate strong general language ability, they lack task-specific grounding and structured output alignment.

B. *Dataset-Level Accuracy*

We evaluated the D-RDMM-trained models on a held-out subset of the dataset. Accuracy was computed based on exact match between predicted output and ground truth action sequence. As shown in Fig.3, all three D-RDMM-trained models achieved consistently high accuracy across the dataset, validating the dataset's utility in teaching complex task reasoning and structured robotic behaviors.

C. *Real-World Deployment*

Beyond offline evaluation, we deployed D-RDMM-trained models on a physical robotic platform at the RoboCup@Home competition, where the robot was tasked with executing natural language instructions across a variety of household scenarios. The model-controlled system demonstrated reliable performance in person-following, object delivery, navigation, and multi-step planning tasks — even when interacting with previously unseen entities and descriptions.

Although systematic real-world metrics such as task success rate or response latency were not formally recorded, the robot was able to interpret natural instructions and complete task sequences effectively in a live, unstructured environment. As illustrated in Fig. 4 and Fig. 5, D-RDMM-trained models executed complex sequential behaviors such as breakfast preparation and grocery tidying in live competition settings.

V. Conclusion

We introduced the **Dynamic RDMM Dataset**, a controllable, scalable, and semantically grounded data generation framework for training language models to perform robotic decision making. Each data sample consists of a natural-language instruction paired with a structured symbolic action program, enabling precise instruction-following in domestic settings.

The dataset is constructed through a two-stage pipeline—hierarchical template expansion and dynamic content generation—that produces over 100,000 unique, task-aligned instruction–action pairs from a compact set of expert-defined templates. D-RDMM supports programmable control over dataset size, task balance, and linguistic diversity, making it a flexible tool for robot learning pipelines, ablation studies, and curriculum-driven training.

We validated the dataset by training multiple open-source LLMs, achieving high accuracy and robust real-world performance in the RoboCup@Home competition. By treating dataset generation as a dynamic process rather than a static artifact, D-RDMM transforms data curation into a tunable component of robot learning, accelerating experimentation and deployment.

We release all templates, code, and seed samples to enable reproducible research and further development of instruction-grounded robotics systems.

References

[1] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 740–10 749.

[2] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, "Teach: Task-driven embodied agents that chat," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2017–2025.

[3] "https://athome.robocup.org/." [Online]. Available: https://athome.robocup.org/

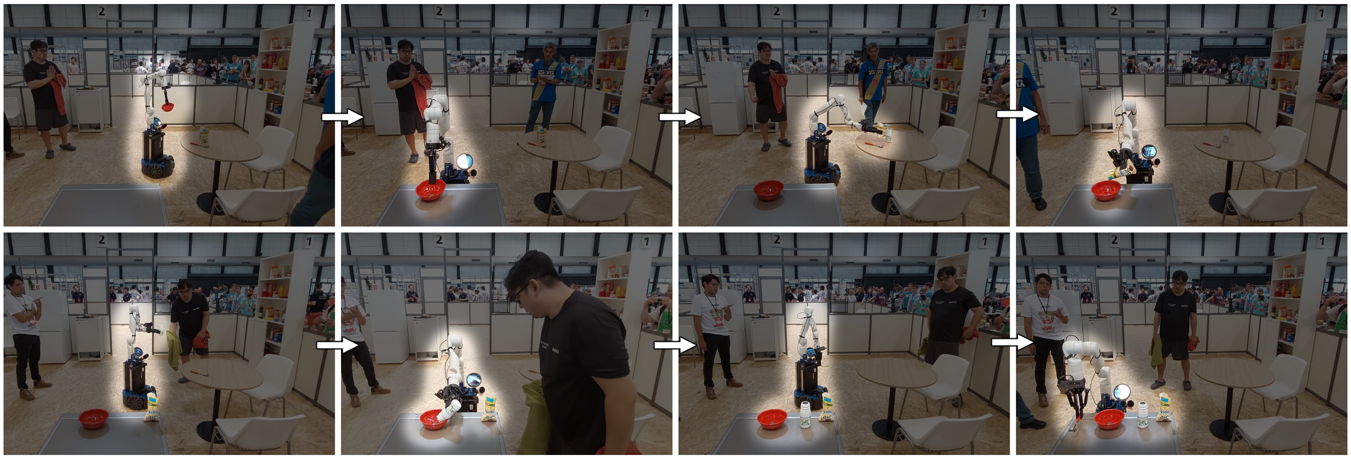

Fig. 4: **Making Breakfast Task:** During the RoboCup@Home competition, a single-arm mobile manipulator carries out a multi-step breakfast setup involving (i) bringing a bowl and placing it on the table, (ii) fetching cereal and setting it down, (iii) retrieving milk and pouring it into the bowl, and (iv) bringing a spoon and placing it alongside. Eight time-lapse frames showcase D-RDMM's ability to orchestrate long-horizon navigation, manipulation, and interaction in a realistic, human-populated environment.

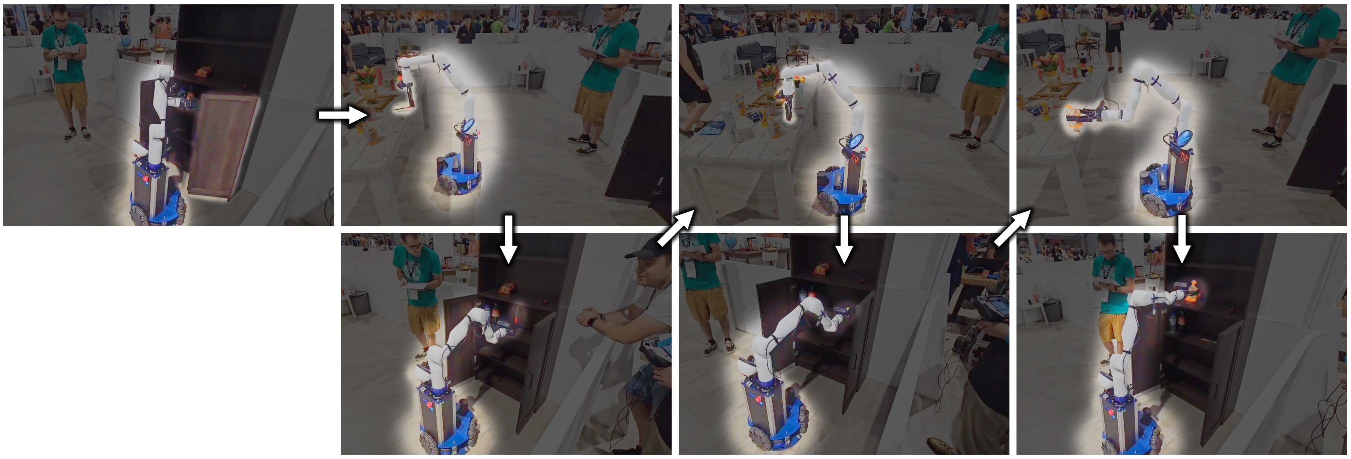

Fig. 5: **Storing Grocery:** In another RoboCup@Home run, the same platform performs a complex tidying routine involving (i) opening the shelf, (ii) retrieving a knife and placing it inside, (iii) fetching a washtub and storing it, and (iv) moving snacks into the shelf. Seven key frames capture the robot's coordinated navigation, object handling, and manipulation at varying heights—demonstrating D-RDMM's ability to drive robust sequential behavior in a dynamic competition environment.

[4] 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.

[5] L. Dong and M. Lapata, "Language to logical form with neural attention," *arXiv preprint arXiv:1601.01280*, 2016.

[6] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 25, no. 1, 2011, pp. 1507–1514.

[7] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, and D. Nardi, "Effective and robust natural language understanding for human-robot interaction," in *ECAI 2014*. IOS Press, 2014, pp. 57–62.

[8] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone, "Learning to interpret natural language commands through human-robot dialog." in *IJCAI*, vol. 15, 2015, pp. 1923–1929.

[9] Z. Ni, X. Deng, C. Tai, X. Zhu, Q. Xie, W. Huang, X. Wu, and L. Zeng, "Grid: Scene-graph-based instruction-driven robotic task planning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 13 765–13 772.

[10] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 39–48.

[11] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[12] 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.

[13] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[14] K. Shirai, C. C. Beltran-Hernandez, M. Hamaya, A. Hashimoto, S. Tanaka, K. Kawaharazuka, K. Tanaka, Y. Ushiku, and S. Mori, "Vision-language interpreter for robot task planning," in *2024 IEEE*

International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 2051–2058.

[15] J. Neidhoefer, J. Arkin, N. Roy, and C. Fan, "Grounded robotic action-rule induction through language models (grail)."

[16] 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.

[17] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang *et al.*, "Palm-e: An embodied multimodal language model," 2023.

[18] 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.

[19] M. F. Ginting, D.-K. Kim, S.-K. Kim, B. J. Krishna, M. J. Kochenderfer, S. Omidshafiei, and A.-a. Agha-mohammadi, "Saycomply: Grounding field robotic tasks in operational compliance through retrieval-based language models," *arXiv preprint arXiv:2411.11323*, 2024.

[20] A. Capitanelli and F. Mastrogiovanni, "A framework for neurosymbolic robot action planning using large language models," *Frontiers in Neurorobotics*, vol. 18, p. 1342786, 2024.

[21] S. Nasrat, M. Kim, S. Lee, J. Lee, Y. Jang, and S. joon Yi, "Rdmm: Fine-tuned llm models for on-device robotic decision making with enhanced contextual awareness in specific domains," 2025. [Online]. Available: https://arxiv.org/abs/2501.16899

[22] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[23] M. ai, "mistral-7b-instruct-0.3v," 2024. [Online]. Available: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[24] A. Y. et al., "Qwen2 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2407.10671

Appendix

This appendix summarizes the task types and symbolic actions used in the RoboCup@Home deployment. Table II lists the 23 instruction templates along with their generation ratios. Table III provides a description of the action primitives used in the competition.

TABLE II: Task Templates Description and Ratio

Task	**Description**	**Amount**
follow	Follow person to location	48
pour	Pour object into container	47
bringdesc	Bring object from location	134
complex_pose	Recognizing human posture	105
complex_countobj	Count object at location	64
countobj	Count object at location	79
descper	Describe person	62
2users	Answer the second user's question	113
goBeaconDoSth	Go to place and do something	111
serve	Put object onto a designated spot	62
guide	Guide person to location	106
store	Put object into storage	37
complex_put_on	Bring, put on object, and answer	79
complex_deliver	Deliver object and answer	150
mgreet	Greet person and answer	110
descobj	Describe object and answer	77
simple	Simple action and answer	18
complex_est	Identify extreme attributes	37
complex_greetdress	Greet person by outfit and answer	139
complex_countperson	Count people by outfit and answer	49
complex_guidedress	Guide person by outfit and answer	179
questions	Answer a simple question	42
time	Tell the time	12
Total		1860

TABLE III: Dataset Actions

Actions	**Description**
Respond	Respond to the user
Move_To	Move to a location
Pour_In	Pour an object into a container
Search_Object	Search for an object
Search_Person	Search for a person
Pickup	Pick up an object
Place_On	Place the picked-up object
Place_Next	Place the picked up object
Give_To	Give the object to the user
Open	Open the door
Close	Close the door
Vision_Ask	Ask the vision system and return the answer to Answer
Answer	Receive the answer from Vision_Ask, Count_Person, or Count_Object
Follow	Follow the person
New_Request	Listen to a question from second user and answer it
Count_Person	Count people with a given attribute and return the answer to Answer
Count_Object	Count a specific object and return the answer to Answer
Ask_Name	Ask a person for their name and return the answer to Answer
What_Time	Tell the time
What_Day	Tell the date
What_Tomorrow	Tell the tomorrow date