Investigating the Role of Language Instructions in Robotic Manipulation Tasks

Anonymous ACL submission

Abstract

Instruction variety greatly impacts a model's ability to generalise outside of its training corpus. While language choices and paraphrases help models generalise to more complex tasks, embodied domain instructing models through multiple modalities (e.g., visual referents) can further help minimise ambiguities and improve the overall success rate. We investigate the impact of multimodal language instructions on a model's generalisation capacities on VIMA-Bench, an environment designed to evaluate generalisation performance through increasing levels of complexity. We design different perturbations that affect both the language and the visual referents in multimodal instructions. Our findings indicate that a VIMA model trained on multimodal instructions not only shows high performance when provided with gibberish instructions, but can even perform better on unseen tasks, casting doubts as to whether content from text in multimodal instructions is more useful than the necessary visual referents. Our findings suggest that current Transformer-based models for Embodied AI tasks are limited as to how way they integrate multiple modalities. Therefore, future work should focus on improvements in architecture design and training regimes to further facilitate multimodal fusion allowing the model to place more importance on the content of the instructions, thereby improving generalisation capabilities.[1]

1 Introduction

Designing artificial agents that can follow natural language instructions is a long-term goal of Artificial Intelligence (Winograd, 1972). In these scenarios, agents must understand the instructions within the context of the observations to predict the next action to take. Ideally, we expect an artificial agent to be able to generalise to previously unseen tasks by combining concepts and skills underpinning its training tasks in novel ways (Lake et al., 2017). Failure to do so means that a model is only good in the environment it was trained in, undermining its ability to adapt to novel scenarios that are likely to happen in the real world.

Previous work has proposed several language-guided tasks for tackling this long-term goal focused more on the ability to generalise to environments with different layouts from the training ones (e.g., ALFRED from Shridhar et al., 2020). Unfortunately, this represents only one facet of the generalisation ability of an agent.

Another limitation of language-guided action execution is that while providing instructions and context using only language can unlock benefits from in-context learning (Bhattacharyya et al., 2023), using language alone is not sufficient to capture all nuances and details within visual scenes (Pezzelle, 2023). Additionally, for multimodal tasks (e.g., action execution tasks (Shridhar et al., 2020)), using language alone is less efficient than using instructions that fuse language with visual representations (Li et al., 2023). For this reason, Jiang et al. (2023) presented VIMA-BENCH, the first benchmark aimed at studying several axes of generalisation involving novel concepts as well as novel tasks where the agent receives multimodal instructions combining both language and visual referents.

However, as multimodal instructions interleave both language and visual representations in a single instruction, the stark contrast between the latent spaces between the modalities might result in models using them as anchors (Wang et al., 2023): over-relying on seeing a visual representation at a specific position within the instruction for the task. If this were the case, it means that models are using superficial characteristics of the input (e.g., the shallow syntactic form of the instruction) to determine what actions to perform, which will directly impact its ability to generalise to unseen tasks and

[1] We will release the codebase on GitHub upon acceptance.

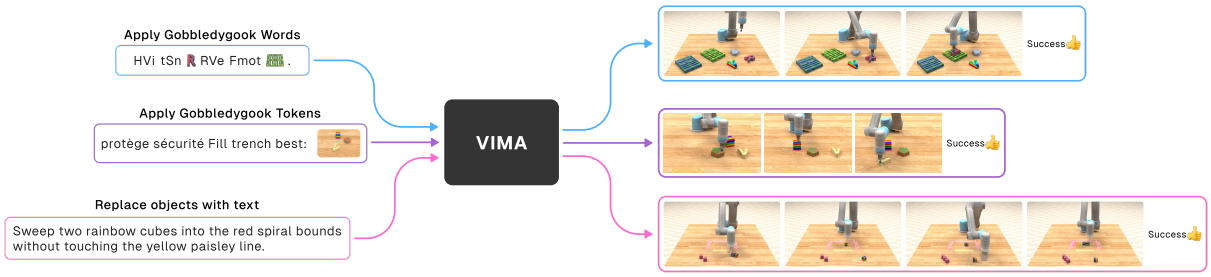

Figure 1: Example of how fused multimodal instructions from VIMA-BENCH (Jiang et al., 2023) impact on the model's evaluation performance. The model exhibits similar levels of success when given either incomprehensible words (Gobbledygood Words- top left), words that are syntactically but not grammatically sound (Gobbledygook Tokens - left), or fully syntactically and grammatically sound information (Replace objects with text - bottom left).

instruction forms as the model will be less good at performing multimodal fusion (Ahuja et al., 2017).

Our Contributions We investigate the impact of multimodal language instructions for state-of-the-art Embodied AI architectures designed for VIMA-BENCH. Motivated by Pythia (Biderman et al., 2023), we recognise the importance of providing the research community with both the training setup as well as training data used to train Embodied AI models because it can have an impact on the final downstream performance. For this reason, we build on top of the data released by VIMA-BENCH (Jiang et al., 2023) and release a reproducible training framework that includes: 1) specific dataset splits to train and evaluate model performance; 2) a training regime to reproduce the VIMA model (Jiang et al., 2023). Thanks to this controllable setup, we were able to design an evaluation framework aimed at studying the impact of properties of the multimodal prompts on the model's performance. For instance, we were able to investigate how well models can perform the task when visual referents are replaced with language descriptions as well as what is the impact of perturbing language instructions.

We found that a VIMA model trained on multimodal instructions can, to our surprise, still perform several tasks of the benchmark even when provided with gibberish instructions or can perform the task even better when visual referents are replaced with language descriptions of the objects. With this study, we aim to shed some light on state-of-the-art model performance and better understand what role language plays in the generalisation abilities of Embodied AI models designed for robotics tasks.

2 Related work

In this section, we provide a survey of the literature on Embodied AI with a focus on evaluating the generalisation ability of embodied agents as well as their ability to understand nuanced language instructions.

2.1 Instruction following in Embodied AI

Embodied AI focuses on designing artificial agents that are embodied in an environment (either simulated or real) and learning to generate actions to complete a given task, whose objective is typically specified in natural language (Das et al., 2018). In the literature, embodied tasks have been formulated in different ways depending on the degree of complexity of the action space. Vision+Language Navigation (e.g., VLN (Anderson et al., 2018), CVDN (Thomason et al., 2020), etc.) is one of the first examples of an agent that can follow instructions in natural language to reach a given destination. However, in this case, the agent's action space included navigation commands only. Thanks to more sophisticated 3D simulated environments such as AI2Thor (Kolve et al., 2017), researchers defined several tasks involving object interaction as well (e.g., ALFRED (Shridhar et al., 2020), Simbot (Shi et al., 2023)).

2.2 Assessing Generalisation in Embodied AI

When deploying Embodied AI systems in the real world they must possess generalisation abilities to dynamically adjust to the increasingly complex and novel tasks that they might face (Duan et al., 2022). Current Embodied AI benchmarks provide seen/unseen splits to assess generalisation across multiple environments/rooms. However, they assume that all the tasks that the agent has to complete, and the objects that the agent has to interact

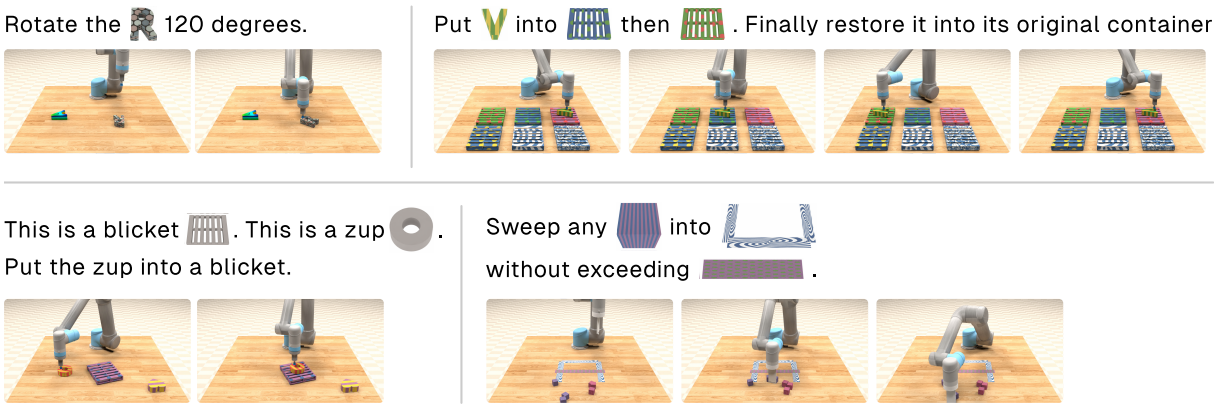

Figure 2: Sample tasks from VIMA-BENCH (Jiang et al., 2023). We refer readers to Appendix B within the VIMA paper for a comprehensive description of all tasks within the benchmark.

with are fully specified at training time. There is no notion of systematic generalisation to new concepts (Suglia et al., 2020), or novel tasks (Chung et al., 2022). To overcome some of these limitations, VIMA-BENCH (Jiang et al., 2023) evaluates generalisation in tabletop robotic manipulation tasks by defining different levels of complexity involving both novel object generalisation and novel task generation as well.

2.3 The Role of Language in Embodied AI

In most of the Embodied AI tasks, language instructions are typically hand-crafted via templates (e.g., VIMA-BENCH) and, in some instances crowd-sourced (e.g., ALFRED). Considering that language represents a generic interface for task learning (Laird et al., 2017), it is important to understand what role language plays in these tasks. Previous work from Akula et al. (2022) discovered that state-of-the-art models trained for the ALFRED benchmark are not sensitive to the language instructions. As reported by Zhu et al. (2023), a similar downside was showcased for the VLN benchmark where even nonsensical instructions seemed to improve downstream performance.

All these efforts therefore highlight the need for a systematic investigation of the role played by language in VIMA-BENCH and the interesting interplay between language and the systematic generalisation capabilities of the agent. Thanks to our reproducible experimental setup, we hope to shed light on many of these problems and provide the community with interesting directions for future work. We describe our experimental setup in the following section.

3 Experimental Setup

We explore the role of language for action execution by evaluating models in the VIMA-BENCH environment (Jiang et al., 2023). Built on top of the Ravens simulator (Zeng et al., 2021), VIMA-BENCH contains 17 tabletop object manipulation tasks to assess the capabilities learned by Vision+Language models through a four-level protocol that evaluates their systematic generalisation capabilities. 50K expert demonstrations are provided for each of 13 tasks, with 4 tasks held out for zero-shot evaluation.[2]

3.1 Skills that models are expected to perform

One of the benefits of VIMA-BENCH is that models must learn skills either in isolation or in combination with other skills, which is a desirable capability of intelligent systems (Lake et al., 2017). Figure 2 shows how skills overlap between tasks:

1. Simple Object Manipulation. Picking up objects from their name or a visual representation, and placing them in specific locations and positions.
2. Visual Goal Completion. Manipulating objects to match the scene in the provided frame.
3. Visual Memory. After performing actions, remember the previous state of the workspace and perform an action given information from that time.
4. Visual Reasoning. Only performing actions on objects that have the same colours/shapes as in the instruction.

[2]We outline our procedure for creating train-validation splits in Appendix A.2.

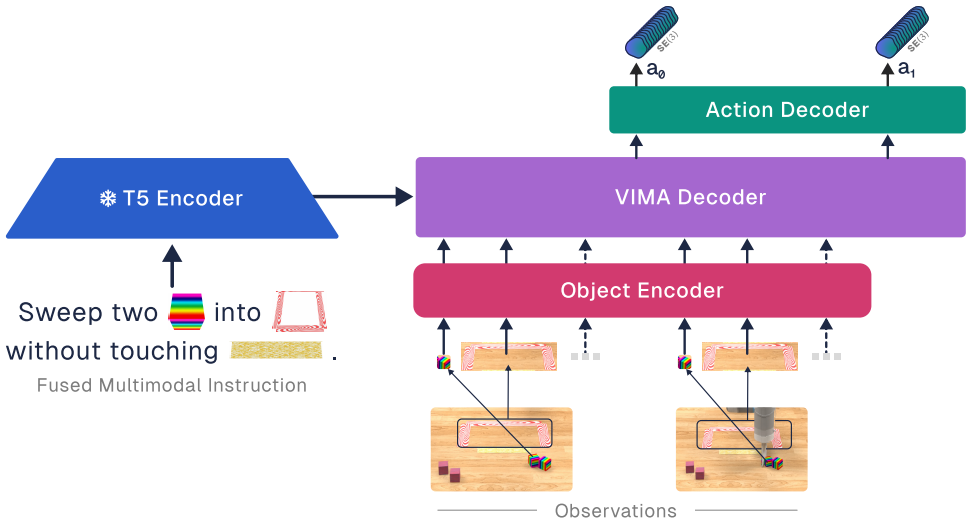

Figure 3: An overview of the VIMA architecture, adapted from Jiang et al. (2023). The VIMA architecture is an encoder-decoder model. A fused multimodal instruction with interweaved language and visual referents is provided to a pretrained T5 encoder. For each observation in the environment, the model must autoregressively predict the correct movement to perform given an instruction. The model must continue to predict movements for each observation until it successfully completes the task or performs the maximum number of moves.

5. One-Shot Imitation. Imitate the actions necessary to make the workspace look like a given sequence of frames.
6. Novel Concept Grounding. The prompt contains unfamiliar words like "dax" which are explained through visual referents and used within an instruction similar to multimodal in-context learning (Zhang et al., 2023).

3.2 Levels of Difficulty

VIMA-BENCH introduces specific levels of difficulty to precisely assess specific model capabilities. We review the different levels of difficulty below:

Placement Generalisation (L1) Identical to the training data; however, the starting location and orientation for each object have not been seen before, ensuring that models understand how to move around the physical space they are in.

Combinatorial Generalisation (L2) All of the tasks, objects, and textures have been seen at some point during training. However, the textures applied to a given object have not been seen together before. This ensures that models do not solely look at the given texture on an object, and can focus on the object itself and the movements necessary to complete the task.

Novel Object Generalisation (L3) Objects and their textures have not been seen during training, ensuring that models are capable of abstracting beyond the specifics of an object.

Novel Task Generalisation (L4) Tasks (including instructions and success criteria) have not been seen before, ensuring that models can abstract further from the training tasks and understand the underlying skills/movements needed to complete the task, combining them in new ways to properly understand and complete the new task.

3.3 VIMA: The Baseline Model

In the environment, models must learn a policy $\pi : \mathcal{P} \times \mathcal{H} \rightarrow \mathcal{A}$ that maps a multimodal prompt $p \in \mathcal{P}$ with a history trajectory of observations and actions $h_t = (o_0, a_0, o_1, \ldots, a_{t-1}, o_t) \in \mathcal{H}$ (up to some discrete timestep t) to the two-pose action primitive $a_t = (\mathcal{T}_{\text{start}}, \mathcal{T}_{\text{end}}) \in \mathcal{A}$. Each multimodal prompt with length l is an ordered sequence $\boldsymbol{p} = (x_1, \ldots, x_l)$ where each element x_i is either a word w_i or a visual representation of an object or frame of a scene v_i. An action token for the environment, a_t, defines a movement between the two end effector poses in $\mathbf{SE}(3)$[3]: the start pose $\mathcal{T}_{\text{start}}$ and the end pose $\mathcal{T}_{\text{end}}$.

Jiang et al. (2023) also proposed a model architecture (Figure 3) that we use for our evaluation.[4]

[3]State vector $(x, y, z, qw, qx, qy, qz)$ where x, y, z are Cartesian coordinates and qw, qx, qy, qz represent the orientation in a quaternion.

[4]We were unable to create a setup that is directly comparable to Jiang et al. (2023). We refer readers to Appendix B.2

The VIMA model follows a transformer encoder-decoder architecture with a frozen pretrained T5 encoder (Raffel et al., 2020; Tsimpoukelli et al., 2021) to encode the multimodal prompts, and a transformer decoder that predicts actions from the history trajectory by conditioning on the prompt through cross-attention layers. The model makes predictions for the position and rotation for each pose using four independent 2-layer MLPs which receive the decoder hidden state for that action as input. Following Pantazopoulos et al. (2023), the model is trained through behaviour cloning that minimises a loss function[5] for a trajectory of T actions given by Equation (1):

$$L(\theta) = \frac{1}{T} \sum_{t=0}^{T} \log \pi_\theta(a_t|p, h_t) \tag{1}$$

For model training, we follow the same training regime described in Jiang et al. (2023) and report additional details that are essential to fully replicate their setup in Appendix A.

4 Evaluating model robustness to language

We explore VIMA's robustness to changes in the multimodal instructions that are different to the ones it was trained on. In this section, we provide a selected number of results but please refer to Appendix B for our comprehensive evaluation.

4.1 Language Perturbation

We define two systematic methods to remove information from the language of a fused multimodal instruction: Gobbledygook Words ($\text{GDG}_{\text{WORDS}}$) and Gobbledygook Tokens ($\text{GDG}_{\text{TOKENS}}$). As illustrated in Figure 4, both methods remove information regarding the task from the language modality, leaving only the visual referents.

To avoid introducing additional difficulty into the tasks, we ensure that the length of the instruction is identical to before perturbing for either natural language words or the tokenized form. Table 1 further verifies this by indicating that the number of words in an instruction does not change for $\text{GDG}_{\text{WORDS}}$, and the number of tokens does not change for $\text{GDG}_{\text{TOKENS}}$. From this, we remove the meaning of the instruction from the natural language itself. It also allows for checking whether or not the length of the instruction in natural language has any impact on model performance.

	# Words	# Tokens
Original Instruction	12.9 ± 7.6	20.2 ± 13.6
$\text{GDG}_{\text{TOKENS}}$	15.2 ± 9.3	20.2 ± 13.6
$\text{GDG}_{\text{WORDS}}$	12.9 ± 7.6	49.7 ± 27.8

Table 1: Average length of the original instructions before and after transforming them through either $\text{GDG}_{\text{WORDS}}$, or $\text{GDG}_{\text{TOKENS}}$.

4.1.1 Experimental Setup

Here we concretely define both language perturbations that we perform on the language.

Gobbledygook Words Let $w_i = (c_1, c_2, \ldots, c_j)$ represent a word with j characters and each c_j is a character from a set $\mathbb{A}$ that contains all uppercase and lowercase alphabetical English characters. Given a multimodal prompt p consisting of multiple words, we transform the sequence in two steps. First, for each word $w_i \in p$, we replace each character with a random one chosen from $\mathbb{A}$ to modify every word within the sequence. Secondly, we randomly swap the positions of words within the sequence without changing the position of any visual representations within the sequence. As illustrated in Figure 4, $\text{GDG}_{\text{WORDS}}$ ensures that the number of characters and "words" within the multimodal prompt—and the number of words between each visual placeholder—does not change. However, the length of the prompt after tokenizing has increased on average because T5 uses a SentencePiece tokenizer that was trained on natural language text (Raffel et al., 2020).

Gobbledygook Tokens The $\text{GDG}_{\text{TOKENS}}$ method transforms the multimodal prompt by randomising each sub-word unit after tokenizing the instruction with any other token from the vocabulary such that the number of sub-word units is the same as the original instruction. See Figure 4 for an example where an instruction perturbed with $\text{GDG}_{\text{TOKENS}}$ does not contain any information in the language modality pertaining to the original task.

4.1.2 Results and Discussion

As we are removing any semblance of well-formed natural language from the instruction across both perturbation methods, we would naturally expect

for more details on the matter.

[5]This was modified from the original VIMA-BENCH loss function to prevent the model from being influenced by the trajectory length.

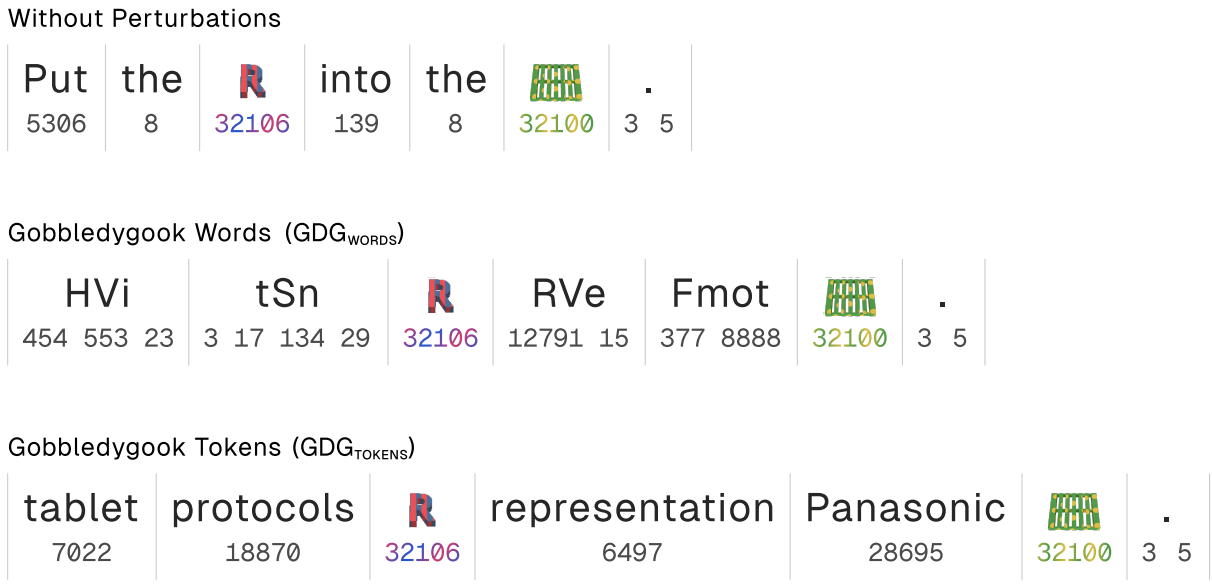

Figure 4: **Language perturbations** designed to challenge the robustness of VIMA. GDG$_{\text{TOKENS}}$ samples random words preserving the original sequence length, while GDG$_{\text{WORDS}}$ changes the words and intentionally increases the resulting number of input ids.

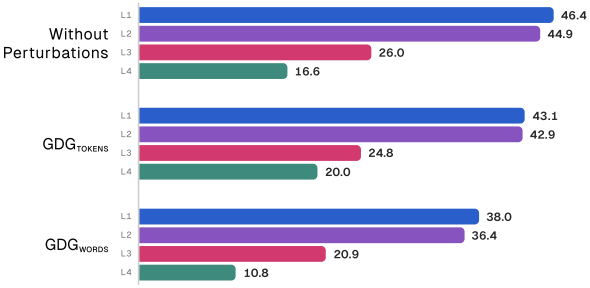

Figure 5: Comparison of the average success rate for each generalisation level when the model is instructed without any language perturbations, and when perturbing the language through GDG$_{\text{WORDS}}$ or GDG$_{\text{TOKENS}}$.

evaluation performance to plummet. However, Figure 5 reveals that when a model trained on the original instructions from VIMA-BENCH is exposed to the language perturbations, the model is still able to perform the task at each generalisation level, with little impact on the average performance. Figure 6 contains some examples where the model still succeeds in performing the task, even when provided with perturbed language from GDG$_{\text{WORDS}}$.

More specifically, from examples 1 to 3, the model successfully followed through on incomprehensible instructions and successfully performed the tasks of identifying the task to perform with the stated object of a choice of two, picking it up and putting it into a destination. Example 4 indicates interesting behaviour: the model incorrectly identified the correct object, placed it into the destination, picked the second object, and placed it in the destination. This would indicate that the environment has not been made aware of such failures; models are permitted to continue until a catastrophic failure occurs.[6] Such a failure is indicated in Example 4, where the model picked the object and placed it into the receptacle in a way that resulted in an unsalvageable scenario, potentially due to there not being any other objects for it to keep trying.

Surprisingly, Figure 5 also illustrates that when faced with GDG$_{\text{TOKENS}}$, the model performs better on unseen tasks than when given original instructions, with the average success rate on L4 increasing from 16.6% to 20% on tasks in L4. This performance increase implies that the semantic content provided by words somehow inhibits performance when faced with unseen tasks. This phenomenon is evidenced in Table 2, which shows model performance on tasks from L4. For example, when the model must understand the task from GDG$_{\text{TOKENS}}$, performance on T8—when faced with both a novel noun and a novel adjective for Novel Concept Grounding—is greater than when provided with linguistically and syntactically sound instructions. Our results indicate that the model might be relying on the visual referents as these remain unchanged across both perturbations which suggests that the fine-tuning of the T5 encoder might not work congruently. Following this, we consider the

[6] We outline the termination conditions for a given episode in Appendix A.4.

	T08	T10	T13	T14
Original Instructions	38.5	0.5	0.0	27.5
w/ GDG_{TOKENS}	67.5	0.5	0.0	12.0
w/ GDG_{WORDS}	42.5	0.0	0.0	0.5

Table 2: Evaluation performance of a model trained on the original instructions from VIMA-BENCH on **Novel Task Generalisation (L4)**.

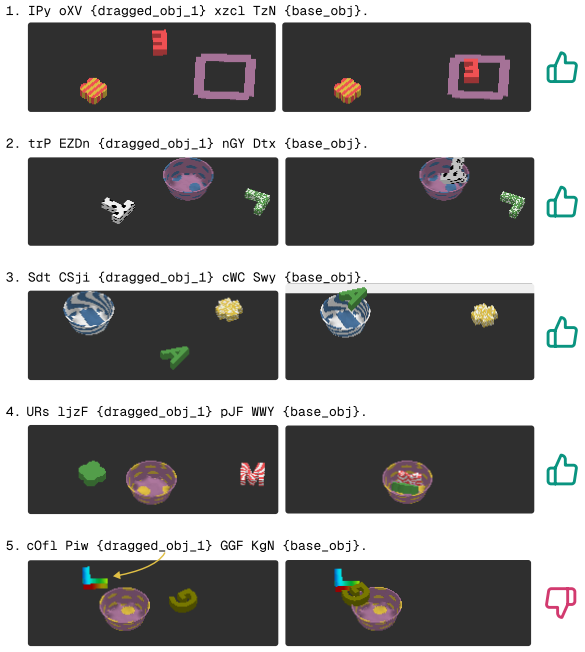

Figure 6: Shown above are in-environment observations seen by the model, showing task performance when using GDG_{WORDS}. Instructions given to the model are shown on top of the images, with the images themselves showing different iterations of either success (top and bottom images) or failure (middle).

possibility that—as the syntax of the instructions in VIMA-BENCH are mostly identical except for the visual referents used—the model has learned the structure and syntax of an instruction, including positions of the visual referents, over the language content of an instruction. We study this case in our next experimental setup.

4.2 Paraphrasing Prompts

Considering that VIMA-BENCH had no diversity in their language templates, we explore the importance of syntactical and lexical choices by creating paraphrases of the original instructions. We do so by manually inspecting the instructions and using meta-templates to construct variations. Table 4 shows some example paraphrases generated using the meta-templates, and we report further details

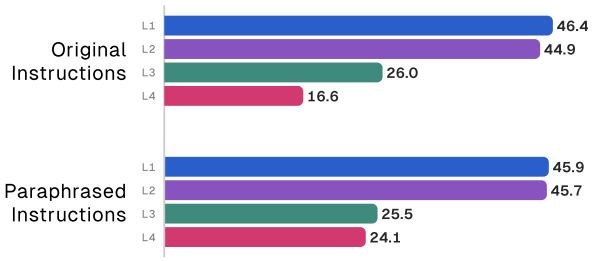

Figure 7: Evaluation performance before and after training on paraphrases.

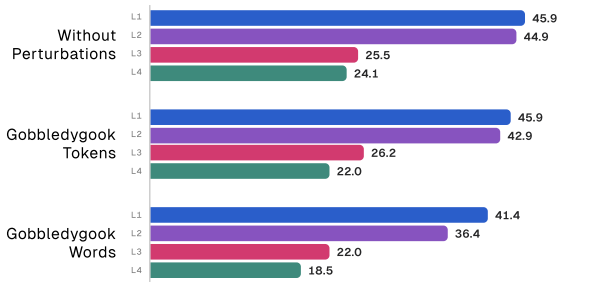

Figure 8: Comparison of model performance when trained on paraphrases and evaluated on language perturbations: GDG_{WORDS} and GDG_{TOKENS}.

and analyses in Appendix A.3.

Results Figure 7 shows the result of the model before and after training on paraphrases. Success rates for original instructions followed along with expected rates results indicating the highest success rates on L1, followed by L2, L3, and L4, which is to be expected due to the mounting difficulty of the tasks. Paraphrasing performed marginally worse on L1, marginally better on L2, worse on L3 and significantly better on L4.

These results showcase that a model trained on additional paraphrased instructions is more successful when faced with the most difficult task available in our environment. Although we expect models to generalise better on unseen tasks when trained on paraphrases, we also expect that they should pay more attention to the language instructions because they have to generalise over different linguistic variants during training.

However, as shown in Figure 8, these models are still robust to language perturbations as indicated by evaluation showing relatively similar results to the paraphrased instructions even though the instructions that they received make no sense in linguistically Figure 4. Additionally, perturbations with GDG_{TOKENS} can even improve the overall performance in L3—a negative result which seems common in previous work in VLN as well (Zhu

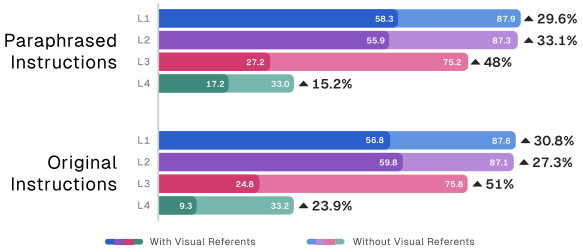

Figure 9: Evaluation performance across both models when all visual referents to objects within instructions have been replaced with natural language text.

et al., 2023).

Notably, similar to the trend in Figure 5, $\text{GDG}_{\text{WORDS}}$ negatively impacts performance more than $\text{GDG}_{\text{TOKENS}}$. While the model underperformed in all tasks compared to the $\text{GDG}_{\text{TOKENS}}$, it still performed relatively successfully through instructions that consisted of randomised characters.

This result implies that even once the meaning of the instruction is removed by the perturbation, a perturbation that results in the same number of tokens provided to the model is preferable to one that increases the overall tokenised instruction length. This suggests a potential problem in the absolute positional embeddings of the Transformer architecture used by VIMA—a problem highlighted by Sinha et al. (2022) for Transformer-based Large Language Models (LLMs).

4.3 Do we even need visual referents in multimodal instructions?

Considering that language instructions seem to have little impact on model performance, we introduce perturbations of the visual referents and explore how useful they are. As shown by Figure 2, a visual referent can either stand for an object or the equivalent of a frame from a scene. Therefore, we only explore the effect of removing visual referents to objects and, as such, limit our evaluation to the subset of tasks that only contain object visual referents (see Figure 2 for examples). To ensure models are given a fair shot during evaluation, we replace each visual referent with a natural language alternative. Concretely, as demonstrated in Figure 1, for every visual referent that refers to an object, we replace it with an adjective-noun pair that also uniquely describes the object.

As shown in Figure 9, regardless of the instruction provided to the model, the model is much better at generalising across all complexity levels. We consider this problem similar to the problem of spurious correlation in Visual Question Answering (e.g., (Selvaraju et al., 2019)), where models ignore the provided input, and output the most likely answer based on language priors. An additional reason for this suboptimal behaviour is the fact that VIMA finetunes an adapter for multimodal encoding (Tsimpoukelli et al., 2021) using only the action prediction loss. As shown by Liu et al. (2023), a preliminary "alignment pretraining stage" seems to be required to reliably align the visual and language modality for a pretrained language model.

5 Conclusion

Embodied AI is a field at the intersection between Robotics and NLP whose aim is to create artificial agents that are embodied in an environment and can execute actions to complete a task. Previous work focused on designing agents that can perform both visual navigation and object manipulation tasks. However, most of them have some drawbacks in terms of their ability to evaluate the ability of models to generalise to novel concepts or tasks. VIMA-BENCH was proposed to provide the community with a benchmark aimed at assessing different levels of systematic generalisation for robotic manipulation tasks.

Despite its coverage of tasks, the VIMA-BENCH ignored the role that language plays in Embodied AI tasks. To study this problem in a principled way, we built on top of VIMA-BENCH to propose a well-defined training setup which provides: 1) specific dataset splits to train and evaluate model performance; 2) a training regime to reproduce the VIMA model. Thanks to this controllable setup, we were able to design an evaluation framework aimed at studying the impact of properties of the multimodal prompts on the model's performance. Therefore, in this study, we investigate whether models proposed for the VIMA-BENCH challenge are: 1) robust to language perturbations; and 2) robust to visual perturbations. To our surprise, we showcase that the VIMA model (Jiang et al., 2023) still perform several tasks of the benchmark even when provided with gibberish instructions or can perform the task even better when visual referents are replaced with language descriptions of the objects. This highlights that there is still a long way to go to create *truly multimodal models* able to reliably perform multimodal fusion (Ahuja et al., 2017).

6 Limitations & Risks

In this study, we investigate the robustness of Embodied AI models proposed for the VIMA-BENCH challenge, a benchmark for robotics manipulation tasks. This benchmark proposes several tasks aimed at assessing the level of generalisation across several axes such as placement generalisation and combinatorial generalisation. We consider this benchmark as instrumental to analyse the capabilities of current Vision+Language models. However, we recognise that the VIMA-BENCH doesn't cover all possible ranges of tasks and conditions that might happen in other benchmarks (e.g., ALFRED) or other real-world scenarios. Therefore, we consider our research paper as an important milestone in investigating the robustness and generalisation of Embodied AI models and we hope to have raised awareness about the importance of creating ecologically valid and linguistically informed Vision+Language benchmarks.

References

Chaitanya Ahuja, Louis Philippe Morency, et al. 2017. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, pages 1–20.

Arjun Akula, Spandana Gella, Aishwarya Padmakumar, Mahdi Namazifar, Mohit Bansal, Jesse Thomason, and Dilek Hakkani-Tur. 2022. Alfred-l: Investigating the role of language for action learning in interactive visual environments. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9369–9378.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Aanisha Bhattacharyya, Yaman Singla, Balaji Krishnamurthy, Rajiv Shah, and Changyou Chen. 2023. A Video Is Worth 4096 Tokens: Verbalize Story Videos To Understand Them In Zero Shot. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9822–9839, Singapore. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. 2022. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. 2023. VIMA: General Robot Manipulation with Multimodal Prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14975–15022. PMLR.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

John E Laird, Kevin Gluck, John Anderson, Kenneth D Forbus, Odest Chadwicke Jenkins, Christian Lebiere, Dario Salvucci, Matthias Scheutz, Andrea Thomaz, Greg Trafton, et al. 2017. Interactive task learning. *IEEE Intelligent Systems*, 32(4):6–21.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253.

Jiachen Li, Qiaozi Gao, Michael Johnston, Xiaofeng Gao, Xuehai He, Suhaila Shakiah, Hangjie Shi, Reza Ghanadan, and William Yang Wang. 2023. Mastering Robot Manipulation with Multimodal Prompts through Pretraining and Multi-task Fine-tuning. ArXiv:2310.09676 [cs].

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Georgios Pantazopoulos, Malvina Nikandrou, Amit Parekh, Bhathiya Hemanthage, Arash Eshghi, Ioannis Konstas, Verena Rieser, Oliver Lemon, and

Alessandro Suglia. 2023. Multitask multimodal prompted training for interactive embodied task completion. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 768–789.

Sandro Pezzelle. 2023. Dealing with Semantic Underspecification in Multimodal NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12098–12112, Toronto, Canada. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. 2019. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2591–2600.

Hangjie Shi, Leslie Ball, Govind Thattai, Desheng Zhang, Lucy Hu, Qiaozi Gao, Suhaila Shakiah, Xiaofeng Gao, Aishwarya Padmakumar, Bofei Yang, et al. 2023. Alexa, play with robot: Introducing the first alexa prize simbot challenge on embodied ai. *arXiv preprint arXiv:2308.05221*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Koustuv Sinha, Amirhossein Kazemnejad, Siva Reddy, Joelle Pineau, Dieuwke Hupkes, and Adina Williams. 2022. The curious case of absolute position embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4449–4472.

Alessandro Suglia, Ioannis Konstas, Andrea Vanzo, Emanuele Bastianelli, Desmond Elliott, Stella Frank, and Oliver Lemon. 2020. Compguesswhat?!: A multi-task evaluation framework for grounded language learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7625–7641.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal Few-Shot Learning with Frozen Language Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 200–212. Curran Associates, Inc.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label Words are Anchors: An Information Flow Perspective for Understanding In-Context Learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. 2021. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Proceedings of the 2020 Conference on Robot Learning*, pages 726–747. PMLR. ISSN: 2640-3498.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Wang Zhu, Ishika Singh, Yuan Huang, Robin Jia, and Jesse Thomason. 2023. Does vln pretraining work with nonsensical or irrelevant instructions? *arXiv preprint arXiv:2311.17280*.

A Training details

To control for possible confounding variables across all models, we use the same training hyperparameters from Appendix D in Jiang et al. (2023) and from the various GitHub issues. We report a comprehensive table of hyperparameters in Table 3.

All models were trained using 4 NVIDIA A100 GPUs for each run, each taking approximately 10 hours. Each evaluation run on the environment took approximately 2 hours and did not require the use of any GPUs. Therefore, the total computational budget for this work is 122 hours.

A.1 Deviations from VIMA

While our training process is similar to Jiang et al. (2023), preliminary experiments showed that using cosine annealing that reduced the LR to the end of the 10th epoch performed better than annealing to 17K steps and training the model at 10^{-7} for 5 epochs.

A.2 Dataset Preparation for Training.

We start by parsing all 664 976 instances across the 13 tasks used for training, provided by Jiang et al. (2023), which contain action trajectories created by

Hyperparameter	Value
Dropout	0.1
Optimizer	AdamW (Kingma and Ba, 2014)
Weight Decay	0
Maximum Learning Rate	1e-4
Minimum Learning Rate	1e-7
Examples per step (Effective Batch Size)	128
Warmup steps	7K (896K examples)
Cosine Annealing steps	All remaining steps
Training epochs	10
Gradient Clip Threshold	1.0

Table 3: Hyperparameters used during model training.

an oracle; therefore, each trajectory is the optimal sequence of movements an agent could perform. We create a validation set using stratified sampling such that a total of 50 000 instances across all the tasks are held out.[7] We then prepare each instance for training in advance through tokenizing any natural language and preparing visual features for the model. <table> shows dataset statistics per task, per split, and across the entire dataset. We (will) release all instances, both before and after preprocessing, to aid in reproducibility.

A.3 Paraphrases

When creating the variations dataset for training, the instances are converted and then preprocessed in a similar fashion to above. When performing the transformation, only the natural language words are altered. The observations seen, the actions the model must perform, and the instances for each train-valid split are unchanged. We provide examples of some paraphrased alternatives of the original instruction in Table 4.

A.4 When does an evaluation episode end?

During the online evaluation, the episode is over when one of two conditions are met:

1. the model has successfully completed the instruction with the previous action it took; or,
2. the model has not successfully completed the instruction within a maximum of 10 actions.

A maximum length of 10 actions is longer than the default length used by Jiang et al. (2023).

[7] Authors state that they held out 50 000 examples for validation on their GitHub: https://github.com/vimalabs/VIMA/issues/8#issuecomment-1491255242.

B Experimental Results

We support all experimental results of our main paper with the per-task success rates for each generalisation level in Tables 5 to 8. In these tables, we have additionally compared performance on the original instructions from the pretrained checkpoint provided by Jiang et al. (2023) on our evaluation setup.

B.1 Each task has been sampled 200 times

Jiang et al. (2023) claimed to run each task in the environment for 100 steps.[8] However, we presume there is some inconsistency in the statement since the reported success rates consist of multiples of "0.5". As a result, we assume that each task was run 200 different times to get a similar result. Li et al. (2023) also sampled 200 instances of each task during evaluation.

B.2 Unable to reproduce reported results

Jiang et al. (2023) only provided the code for the model and the dataset did not contain a train-test split. After creating a working codebase, we were unable to reproduce the results reported by Jiang et al. (2023) using the provided model checkpoint. We spent several weeks trying to reproduce the results, including consulting the original authors on their experimental setup, but were unsuccessful in doing so.

C Reproducibility

VIMA-BENCH from Jiang et al. (2023), including all pre-existing model code, pre-trained checkpoints, and the environment are licensed under MIT, and all artefacts produced from this work will be released under the same license.

[8] While not reported within the paper, it was mentioned on their public GitHub repository: https://github.com/vimalabs/VIMA/issues/16#issuecomment-1622973970.

Task	Original	Alternative
1	Put the blue spiral object in {scene} into the wooden object.	From the {scene} stack the blue spiral object on the wooden thing.
2	Put the dragged_texture object in scene into the base_texture object.	Move objects in the scene so that the dragged_texture item is on one base_texture item.
3	Rotate the dragged_obj angle_in_degree degrees.	Turn the dragged_obj precisely angle_in_degree degrees.
4	Rearrange to this scene.	Rearrange things into this setup scene.
5	Rearrange objects to this setup {scene} and then restore.	Rearrange objects into this configuration scene and put it back.
6	demo_blicker_obj_1 is kobar than demo_blicker_obj_2. demo_blicker_obj_3 is kobar than demo_blicker_obj_4. Put the kobar dragged_obj into the base_obj.	object1 object3 and object5 are all kobar than objects object2 object4 and object6 respectively. move the kobar dragged_obj inside of the base_obj.
7	This is a blinket dragged_obj. This is a zup base_obj. Put a zup into a blinket.	This is a blinket object2. this is a zup object1. drop the zup inside of the blinket.
11	Stack objects in this order: frame1 frame2 frame3.	Move objects like this: frame1 frame2 frame3.
16	First put object1 into object2 then put the object that was previously at its direction into the same object2.	Set object1 in object2 then place the item that was at its direction before you placed it into the same place.
17	Put object1 into object2. Finally restore it into its original container.	Set object1 within object2 then restore it to its original place.

Table 4: Some of the alternative paraphrases generated from the meta-templates.

	T01	T02	T03	T04	T05	T06	T07	T09	T11	T12	T15	T16	T17	Overall
Provided Checkpoint (Jiang et al., 2023)														
Original Instruction	73.0	46.5	19.0	5.0	6.0	20.5	7.5	2.0	23.0	97.0	1.5	10.0	11.5	24.8
w/ GDG_{TOKENS}	56.0	68.0	22.0	12.0	4.0	81.0	75.0	6.5	14.0	90.0	1.0	6.5	3.5	33.8
w/ GDG_{WORDS}	44.0	16.5	12.0	5.5	2.0	34.0	29.5	0.5	15.5	89.5	0.5	0.0	4.5	19.5
w/ Paraphrases	51.0	37.5	15.5	17.5	3.5	46.0	12.0	1.5	20.0	92.5	0.0	12.5	5.5	24.2
w/ Objects As Text	91.0	—	100.0	—	—	—	86.5	—	—	98.0	9.0	35.0	1.5	60.1
+ w/ GDG_{TOKENS}	78.5	—	28.5	—	—	—	81.5	—	—	94.0	1.0	15.0	6.0	43.5
+ w/ GDG_{WORDS}	97.5	—	23.5	—	—	—	97.0	—	—	93.0	5.5	0.5	2.0	45.6
Trained on Original Instructions														
Original Instruction	88.5	72.5	2.5	7.0	1.0	96.5	61.0	1.5	27.5	97.0	65.0	14.0	69.5	46.4
w/ GDG_{TOKENS}	88.5	74.5	1.0	8.5	1.0	91.5	77.5	0.0	27.5	90.5	15.5	12.5	72.0	43.1
w/ GDG_{WORDS}	73.5	70.5	0.5	4.5	2.5	74.5	80.0	0.0	14.0	95.5	1.0	11.5	66.0	38.0
w/ Paraphrases	88.0	69.5	3.5	4.0	2.0	94.0	66.5	0.0	18.0	92.5	36.5	18.5	62.5	42.7
w/ Objects As Text	99.0	—	99.5	—	—	—	100.0	—	—	91.5	99.5	50.5	73.0	87.6
+ w/ GDG_{TOKENS}	97.0	—	14.5	—	—	—	93.0	—	—	91.0	26.0	11.5	70.5	57.6
+ w/ GDG_{WORDS}	99.0	—	10.5	—	—	—	99.5	—	—	90.5	65.5	17.0	8.0	55.7
Trained on Paraphrases														
Original Instruction	94.5	86.5	1.0	8.0	0.5	73.0	73.5	1.0	21.5	94.5	65.5	22.5	54.5	45.9
w/ GDG_{TOKENS}	85.5	81.0	0.5	7.5	1.0	81.0	84.0	0.5	23.0	93.0	38.5	18.0	60.5	44.2
w/ GDG_{WORDS}	82.0	81.0	0.5	4.5	2.5	76.5	81.0	0.5	16.0	89.5	21.0	16.5	67.0	41.4
w/ Paraphrases	93.0	84.5	1.0	6.5	2.0	75.0	73.0	0.0	19.5	91.5	53.0	25.0	49.5	44.1
w/ Objects As Text	100.0	—	99.5	—	—	—	100.0	—	—	95.5	99.5	47.0	74.0	87.9
+ w/ GDG_{TOKENS}	94.0	—	6.5	—	—	—	92.0	—	—	92.0	31.0	21.5	63.0	57.1
+ w/ GDG_{WORDS}	99.5	—	12.5	—	—	—	100.0	—	—	88.5	63.5	24.5	2.5	55.9

Table 5: Model evaluation performance at **Placement Generalisation (L1)** where the exact starting location and orientation of each object *were not seen* during training. 200 episodes were sampled for each task, and all results reported at precision of one decimal place.

	T01	T02	T03	T04	T05	T06	T07	T09	T11	T12	T15	T16	T17	Overall
Provided Checkpoint (Jiang et al., 2023)														
Original Instruction	65.0	45.5	19.0	5.5	5.5	30.0	11.5	3.0	15.0	92.5	0.5	12.5	11.0	24.3
w/ GDG_{TOKENS}	50.0	56.5	18.5	4.0	2.5	83.5	75.5	8.5	17.0	90.0	1.5	6.0	8.0	32.4
w/ GDG_{WORDS}	36.5	20.5	12.5	6.0	1.0	36.5	21.0	1.5	13.0	91.0	1.0	0.5	1.5	18.7
w/ Paraphrases	51.5	32.0	15.5	6.5	5.0	52.0	11.0	1.0	20.0	95.0	1.0	11.0	5.0	23.6
w/ Objects As Text	90.5	—	100.0	—	—	—	88.5	—	—	93.0	3.0	34.5	5.0	59.2
+ w/ GDG_{TOKENS}	78.0	—	28.5	—	—	—	76.5	—	—	97.0	1.0	7.5	7.0	42.2
+ w/ GDG_{WORDS}	95.5	—	23.5	—	—	—	96.5	—	—	93.0	4.0	2.0	2.0	45.2
Trained on Original Instructions														
Original Instruction	93.0	73.0	1.5	6.5	1.5	93.0	62.0	1.5	32.5	91.0	47.0	17.0	64.0	44.9
w/ GDG_{TOKENS}	86.0	71.5	1.0	7.0	1.5	92.0	84.5	0.0	27.0	92.5	14.0	11.0	70.0	42.9
w/ GDG_{WORDS}	70.0	62.0	0.0	7.5	2.0	78.5	69.0	1.5	10.0	94.0	2.5	12.5	64.0	36.4
w/ Paraphrases	88.0	59.5	2.5	3.5	2.5	91.5	65.0	0.0	14.5	90.5	34.5	21.5	59.5	41.0
w/ Objects As Text	100.0	—	100.0	—	—	—	100.0	—	—	93.5	98.5	45.5	72.0	87.1
+ w/ GDG_{TOKENS}	94.0	—	11.0	—	—	—	94.5	—	—	85.0	20.5	13.5	59.0	53.9
+ w/ GDG_{WORDS}	99.5	—	11.5	—	—	—	97.5	—	—	87.0	63.5	15.0	6.0	54.3
Trained on Paraphrases														
Original Instruction	90.5	82.5	0.5	7.0	0.5	89.0	75.5	0.0	23.0	93.0	65.5	22.5	44.0	45.7
w/ GDG_{TOKENS}	83.0	83.5	0.5	5.0	1.5	84.0	82.0	0.5	27.0	91.0	38.5	21.5	48.0	43.5
w/ GDG_{WORDS}	79.0	75.5	0.5	6.0	2.5	82.5	75.5	0.0	17.0	95.0	20.5	20.5	55.0	40.7
w/ Paraphrases	90.5	77.0	1.0	9.5	1.5	91.0	75.0	0.5	17.0	93.0	46.5	24.0	33.5	43.1
w/ Objects As Text	100.0	—	100.0	—	—	—	99.0	—	—	96.5	100.0	46.5	69.0	87.3
+ w/ GDG_{TOKENS}	93.0	—	6.5	—	—	—	90.5	—	—	90.0	29.0	19.5	53.0	54.5
+ w/ GDG_{WORDS}	99.0	—	12.5	—	—	—	99.5	—	—	87.0	62.5	21.0	1.5	54.7

Table 6: Model evaluation performance for **Combinatorial Generalisation (L2)** where *the textures used on a given object* were not seen during training. 200 episodes were sampled for each task and all results reported at precision of one decimal place.

	T01	T02	T03	T04	T05	T06	T07	T09	T11	T15	T16	T17	Overall
Provided Checkpoint (Jiang et al., 2023)													
Original Instruction	60.5	49.5	22.0	9.5	5.5	37.0	9.5	1.0	22.0	0.5	5.5	0.0	18.5
w/ GDG_{TOKENS}	42.5	66.0	30.0	9.5	3.5	79.5	66.0	10.0	13.5	0.5	2.0	1.5	27.0
w/ GDG_{WORDS}	36.5	14.0	10.5	4.0	2.0	31.0	25.0	2.0	18.5	0.0	0.0	2.0	12.1
w/ Paraphrases	43.0	35.0	14.0	10.0	6.0	39.0	12.0	1.0	17.0	2.0	5.0	1.0	15.4
w/ Objects As Text	90.5	—	100.0	—	—	—	89.5	—	—	3.0	33.5	1.5	53.0
+ w/ GDG_{TOKENS}	65.0	—	24.0	—	—	—	70.0	—	—	1.0	7.0	0.5	27.9
+ w/ GDG_{WORDS}	91.5	—	27.5	—	—	—	95.0	—	—	3.0	0.0	0.0	36.2
Trained on Original Instructions													
Original Instruction	65.5	50.0	1.5	3.5	2.5	78.0	51.0	0.0	28.5	19.0	11.0	1.0	26.0
w/ GDG_{TOKENS}	67.0	51.5	2.0	6.5	1.0	67.0	54.5	0.0	24.0	6.0	9.5	9.0	24.8
w/ GDG_{WORDS}	56.0	44.0	0.5	5.0	2.0	57.0	56.5	0.0	14.0	2.0	5.0	9.0	20.9
w/ Paraphrases	64.0	39.0	1.5	7.5	2.0	70.0	39.0	0.5	20.0	16.0	12.5	0.5	22.7
w/ Objects As Text	99.5	—	100.0	—	—	—	100.0	—	—	98.5	50.0	6.5	75.8
+ w/ GDG_{TOKENS}	73.0	—	6.5	—	—	—	68.5	—	—	6.5	12.5	3.5	28.4
+ w/ GDG_{WORDS}	88.0	—	8.5	—	—	—	90.0	—	—	30.5	7.0	0.0	37.3
Trained on Paraphrases													
Original Instruction	63.5	59.5	2.5	6.5	3.5	45.5	48.0	1.5	26.5	31.5	17.5	0.0	25.5
w/ GDG_{TOKENS}	58.5	59.0	4.0	7.0	1.5	61.0	55.0	1.0	30.5	20.5	16.0	0.0	26.2
w/ GDG_{WORDS}	55.0	56.5	0.5	6.0	1.5	52.5	58.5	1.0	14.0	6.5	12.0	0.0	22.0
w/ Paraphrases	68.0	54.5	2.5	8.0	2.0	46.5	51.0	1.0	21.0	21.0	16.5	0.0	24.3
w/ Objects As Text	99.0	—	100.0	—	—	—	99.0	—	—	97.5	55.5	0.5	75.2
+ w/ GDG_{TOKENS}	66.5	—	10.5	—	—	—	61.5	—	—	5.5	7.0	0.0	25.2
+ w/ GDG_{WORDS}	89.0	—	16.5	—	—	—	89.5	—	—	25.0	8.0	0.0	38.0

Table 7: Model evaluation performance for **Novel Object Generalisation (L3)** where all objects were not seen during training (and textures may or may not have been seen during training). 200 episodes were sampled for each task and all results reported at precision of one decimal place.

	T08	T10	T13	T14	Overall
Provided Checkpoint (Jiang et al., 2023)					
Original Instruction	9.5	0.0	0.0	1.0	2.6
w/ $\text{GDG}_{\text{TOKENS}}$	71.0	0.0	0.0	1.5	18.1
w/ $\text{GDG}_{\text{WORDS}}$	15.0	0.0	0.0	0.0	3.8
w/ Paraphrases	11.5	0.5	0.0	0.5	3.1
w/ Objects As Text	—	0.0	0.0	3.5	1.2
+ w/ $\text{GDG}_{\text{TOKENS}}$	—	0.0	0.0	0.5	0.2
+ w/ $\text{GDG}_{\text{WORDS}}$	—	0.0	0.0	5.0	1.7
Trained on Original Instructions					
Original Instruction	38.5	0.5	0.0	27.5	16.6
w/ $\text{GDG}_{\text{TOKENS}}$	67.5	0.5	0.0	12.0	20.0
w/ $\text{GDG}_{\text{WORDS}}$	42.5	0.0	0.0	0.5	10.8
w/ Paraphrases	40.5	0.0	0.0	21.0	15.4
w/ Objects As Text	—	0.0	0.0	99.5	33.2
+ w/ $\text{GDG}_{\text{TOKENS}}$	—	1.0	0.0	13.0	4.7
+ w/ $\text{GDG}_{\text{WORDS}}$	—	0.0	0.0	39.5	13.2
Trained on Paraphrases					
Original Instruction	45.0	0.0	0.0	51.5	24.1
w/ $\text{GDG}_{\text{TOKENS}}$	57.5	0.5	0.0	30.0	22.0
w/ $\text{GDG}_{\text{WORDS}}$	52.0	1.0	0.0	21.0	18.5
w/ Paraphrases	54.5	0.5	0.0	44.5	24.9
w/ Objects As Text	—	0.0	0.0	99.0	33.0
+ w/ $\text{GDG}_{\text{TOKENS}}$	—	0.0	0.0	20.5	6.8
+ w/ $\text{GDG}_{\text{WORDS}}$	—	0.0	0.0	45.5	15.2

Table 8: Model evaluation performance for **Novel Task Generalisation (L4)**, where tasks have not been seen before. Objects and their textures may or may not have been seen during training. 200 episodes were sampled for each task and all results reported at precision of one decimal place.