CLARITY: Counteracting Language Alteration Risks in Intelligent Systems

Nicholas Gray
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816
ni585527@ucf.edu

Ashton Frias
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816
as700045@ucf.edu

Abhinav Kotta
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816
ab828858@ucf.edu

Wen-Kai Chen
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816
we337236@ucf.edu

Tyler VanderMate
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816
ty111606@ucf.edu

Abstract

There has been significant work in recent years investigating the usage of large language models (LLMs) and vision-language models (VLMs) in robotic applications. However, a recent work from the University of Maryland has demonstrated that popular LLM and VLM based robotic systems are greatly vulnerable to rephrasing attacks, where a prompt is changed to use different words and architecture while still possessing the same semantic meaning. To answer questions raised in this work, we use the VIMA architecture to investigate the cause of the prompt attacks working and how they can be defended against. Our findings revealed that rewording attacks caused a significant deviation in the latent space representation of the prompt, causing notable changes in performance. We then introduce CLARITY, a novel alignment layer to the VIMA architecture and train on three prompt attack types across three tasks on two complexities. The results of our technique lead near-complete performance recovery on the trained tasks and on two additional zero-shot attack types and zero-shot higher task complexity. We believe this demonstrates a quick but effective defensive layer against basic rephrasing prompt attacks, allowing for potential drop-in additions to other robotic systems without the need for significant retraining or architecture changes. All code can be found at https://github.com/SandysPappy/LASER.

1. Introduction

Since the rise in popularity of Large Language Models (LLMs) and Vision-Language Models (VLMs), there has been significant interest in extending the models to robotics to take advantage of their natural language processing and visual recognition capabilities to enhance performance and interactivity. The potential benefits for LLM/VLM enhanced robotic systems are great, with potential benefits in industries such as healthcare [12, 21, 31], manufacturing [43, 44], and the service industries [3, 8]. However, the introduction of LLMs/VLMs into robotic systems introduces new risks and vulnerabilities as well, with limited existing defense mechanisms to defend against potential attacks. For example, language model systems are prone to hallucinations, which could limit or distort a robot's understanding of its environment and lead to undesired action. Using LLM/VLM systems also introduces concerns over reasoning capabilities, as it makes the robotic system at least partially rely on the potentially faulty logic learned from a text or multi-modal input space. Furthermore, LLMs/VLMs can have ambiguity in understanding contextual information provided by text and images, which can be another source of failure [25]. A significant potential risk to robotic systems is a lack of flexibility in the training prompts, as they often are trained on a limited set of prompt formats and a limited set of word options [11, 16, 23], with a lack of synonyms and sentence variants leading to potential prompt misunderstandings [18, 39].

Recent work from the University of Maryland investigates these concerns and found significant evidence that not only has there been no research into understanding the potential vulnerabilities of LLM/VLM-based robotic systems, but that these systems are highly vulnerable to adversarial attacks on both the text and image inputs into the models [45]. Their work studied popular LLM/VLM-based robotic models Instruct2Act [14], VIMA [16], and KnowNo

[36] and found that all three architectures are vulnerable to prompt attacks that control the initial instruction of the model and perception attacks that affect a model's perception of its environment. Of specific interest to us was the prompt-based attacks, which work by rephrasing the instruction prompt using GPT-3.5 into a semantically similar but structurally different prompt, which led to the models performing incorrect actions. This work leaves open questions though on why these attacks work on these systems and how they can be defended against rigorously and efficiently.

The goal of our work was to investigate why these prompting attacks are able to hurt model performance, and how they can be defended against to improve model performance against these attacks. We investigate why the prompt attacks, which should not have a significant effect on models due to having the same semantic meaning as the original instructions, cause changes in operation, and how the prompt changes correlate to overall changes in model success rate. For our investigation, we chose to use the VIMA architecture [16], as it uses a multi-modal prompt input at the start of the simulation, which controls the rest of the model's operation. This allows us to analyze directly how the adversarial prompts are changing the output token embedding of the T5 LLM, and how much of a change causes VIMA pipeline to fail.

We offer two main contributions to this work:

- **Latent space analysis on prompt-based adversarial attacks.** We conduct an analysis on the cosine similarity between the token embeddings of original instruction prompts and the adversarial prompts, finding a notably low cosine similarity and a correlation between low similarity and decreased success rate in tasks.
- **Novel defensive alignment layer.** This paper introduces CLARITY, a simple yet effective multi-layer perceptron layer on top of the T5 multi-modal LLM to align adversarial instruction prompts. This alignment layer is able to recover lost performance against adversarial attacks while minimally hurting performance in normal instruction prompts, and achieving greater similarity between the normal instructions and adversarial instructions in the token embeddings.

2. Related Works

2.1. Language Models in Robotics

There has been significant literature in recent years demonstrating the benefits of integrating Large Language Models (LLMs) and Vision Language Models (VLMs) with robotic systems, representing a significant advancement in embodied AI [6, 7, 42] . The integration of LLMs and VLMs allows robots to use the inference and human understanding capabilities of Language Models to improve decision making and reasoning tasks. Recent research has primarily investigating the usage of LLM/VLM-based robotic systems in navigation and manipulation tasks, where the model needs to have both an understanding of its surroundings and a comprehension of the human instruction [19, 37]. In navigation tasks, the model must be able to navigate an environment while completing human given instructions given via a natural language prompt. These tasks require Vision Language Models to be trained on large image datasets and be able to understanding language input, recognize seen objects and their positions, navigate correctly and efficiently, and detect and pinpoint both in- and out-of-domain objects [13, 29, 33]. Manipulation tasks involve similar but distinct processes, where the vision-language model must be able to understand text instruction and visual perception to generate actions of robotic manipulators [4, 5, 16, 27, 41]. This task type involves have scene understanding, grasping, and arranging objects in both real-world and simulated environments, requiring a continuous understanding of its current, future, and end states.

Task complexity is another metric to classify tasks undertaken by LLM/VLM-based robotic systems, as the complexity of tasks can span from basic perception and manipulation to long-term advanced planning and reasoning. Models performing perception-based tasks are relatively simple, as they can either gather training data directly from scene observation or through large Web-sourced datasets. Reasoning and planning tasks, on the other hand, need to engage in complex decision making, requiring greater comprehension and common-sense understanding [4, 24, 32]. There have been recent efforts to enhance the capabilities of these systems in these tasks, such as pre-training for task prioritization [1] or converting instructions into detailed reward-based tasks [46]. There has been work in human-in-the-loop learning as well, such as enabling comprehension and learning from human demonstrations and instructions [40] or through additional guidance from human overseers when the system is unsure in making a decision [36]. Despite the fact that there has been extensive research in integrating and improving LLM/VLM-based robotic systems, there is a lack of investigation into the potential risks of using language models in these systems. Language models are known to have multiple points of vulnerability and failure, which may be leaving integrated robot systems with failure points that could be attacked by malicious actors or unexpectedly fail when unintended behavior is introduced, leaving potentially serious consequences.

2.2. Adversarial Attacks on Language Models

Adversarial attacks are defined as input that reliably trigger erroneous output from language models [30]. There is a wide range of attack strategies against language models, including gradient-based attacks, jailbreak prompting, model red-teaming, and token manipulation, to give a few examples. With token manipulation, prompts are altered by switching words with synonyms, random token insertion, and swapping words in the prompt [17, 22, 28]. In gradient-based attacks, the model's gradients are analyzed to find vulnerabilities that can lead to adversarial prompts. Jailbreak prompting is a popular techniques that involves crafting prompts to specifically bypass model restrictions and defenses, leading to undesired behavior in the model. Model Red-Teaming involves testing model against these adversarial prompts to understand the model's robustness. Language Models can also be attacked indirectly, such as through external documents or websites via Retrieval Augmented Generation (RAG), introducing new avenues for adversarial prompts and attacks [10].

Attacks can be performed on vision language models as well, through both the text input and the vision input. A recent study showed that multi-modal foundation models are vulnerable to adversarial attacks via images by near imperceptible perturbations to the input images, generating non-aligned or off-topic responses [38]. Alignment strategies have been shown to be vulnerable as well, as [9, 25, 26] demonstrate that one-dimensional alignment strategies are limited and fail against multi-modal inputs.

2.3. Defense Techniques for Language Models

Research on defenses against adversarial attacks have primarily been oriented towards defending against attacks against aligned language models [15]. Most defenses can be categorized into three main concepts: detection, preprocessing, and adversarial training. In detection based systems, an adversarial prompt is checked to see if it breaks alignment, and rejects the prompt if it is detected as adversarial.

A recent study demonstrated a detection defense against jailbreaking attacks [48] with a perplexity filter, where perplexity is defined as the average negative log likelihood of each next text token appearing, and where rejection occurs if the text perplexity is over a given threshold [15]. Preprocessing defenses involve preprocessing and altering the prompt or its tokens in some way to preserve the prompt's original meaning while destroying the original adversarial token sequence. An example of a preprocessing defense is paraphrasing, where an adversarial prompt is fed through a generative model to create a new but semantically similar prompt that does not preserve the original token sequence [20] . Another defense is retokenization, where the original prompt tokens are broken up into multiple smaller tokens at random, better preserving the original prompt while distorting the exact token sequence itself [34]. Adversarial training is a technique where a language model has adversarial prompts injected into its training set and is taught to generate rejection text against the adversarial prompts [15, 47].

2.4. Safety Concerns of LLM/VLM Integration in Robotics

While there is substantial evidence put forth in current literature on the effectiveness integrating LLMs and VLMs in robotic systems and their performance benefits [], there has been a lack of exploration into the safety concerns on the integration of these systems, especially in relation to adversarial techniques being used against these models. The first major work to study the effects of adversarial attacks, and the motivator of our work, is ”On the Safety Concerns of Deploying LLMs/VLMs in Robotics: Highlighting the Risks and Vulnerabilities.” [45] This work addresses the mentioned gap, and demonstrates on three popular LLM/VLM-based robotic systems that current systems are vulnerable to adversarial attacks on the controlling prompt given to the LLM/VLM and the robot's perception of its environment.

While this work addresses the gap in literature on the potential of attacks against LLM/VLM-based robotic systems, there is still a gap present on understanding and defending against these system vulnerabilities. Specifically, there is a need to address the prompt-based attacks against these systems, as the prompt will be the primary means of interaction between the human and robot system and is of crucial importance in human-robot interactions. [2] Our work aims to fill this gap by investigating the potential root cause of the attack's success and introducing a defense technique against these attacks that is quick to train and simple to integrate and deploy.

3. Methodology

A major motivation for this work was to develop a system that is versatile and easy to implement across potentially all robotics-based Visual Language Models. To achieve this, we prioritized creating a lightweight additional alignment layer that does not require many resources to train, has swift training cycles, and is independent of a tested model. We chose training on an MLP (Multi-Layer Perceptron) network which aligns with our objectives, meeting all three criteria with precision and accuracy. We chose to investigate our alignment layer technique using the VIMA architecture [16]. VIMA was chosen as it is a well-cited publication in LLM/VLM-based robotics, uses a multi-modal prompt input, and can be separated into two distinct parts: the initial prompt encoding and the secondary interaction layers. This separation allowed us to easily add the alignment layer on top of the T5 LLM [35] used in VIMA for prompt encoding, allowing for fast training, implementation, and testing.

We solved this problem in three steps.

- (1) We develop a dataset of three manipulation and understanding tasks across three task complexities with 5 different prompt attacks for each task. We separated this

dataset into a training and testing set, measuring zero-shot performance on the highest complexity task.
- (2) We investigate the similarity between the embeddings of the base prompts and attack prompts after being passed through the T5 LLM to gain an understanding of how the attack prompts position in the latent space and how they may relate to the current task.
- (3) We develop a novel alignment layer system designed to be a drop-in addition to VIMA on top of the T5 LLM, enabling performance improvements by re-aligning attack prompts back to the base prompt area of the latent space. We train the alignment layer using our dataset, and demonstrate near-total recovery of lost performance from the original prompt attacks.

3.1. Training Data

In order to properly investigate the similarity between the base prompts and the attack prompts in the T5 latent space and train and test the novel alignment layer, we needed to develop a dataset that encompassed a variety of tasks types, task complexities, and attack types. For our dataset, we explored three tasks available in VIMA-Bench [16]: Visual Manipulation, where the robot must pick up a specific object(s) and place it (them) in a specified container, Rearrange, where the robot must rearrange target objects to match a given configuration, and Scene Understanding, where the robot must put objects with a specified texture in a given image into a container object with a specified color. For these tasks, we investigate across three task generalizations, placement, combinatorial, and novel object, whose definitions can be found in Figure 1. For our prompt attacks, we look at five attack types: simple, extend, color rephrase, object rephrase, and noun rephrase. The definition of these tasks can be found in Figure 2. The attack prompts are generated by calling GPT-3.5 with the original prompt and rephrasing instruction, which returns the rephrased adversarial attack prompt.

For the training dataset, we chose all three tasks, 3 of the 5 attacks (Extend, Noun, and Color Rephrase), and 2 partitions (Placement Generalization, and Combinatorial) which resulted in 12 distinct combinations. For each combination, we developed 10 attack prompts for each of the 150 base prompts, ensuring that our MLPs can effectively learn to align the base-to-attack prompt embeddings. For the other prompt attacks (simple, object rephrase) and the highest task complexity (novel object), we perform zero-shot testing on these tasks with the same number of attacks per iteration per task.

3.2. Experiment Metrics

We employed two metrics, success rate and cosine similarity for interpreting the results that were produced. The success rate, given as a percentage, is used to evaluate how the model is affected by the prompt attacks, and how well our system is able to recover from the attacks. Success rate is defined as the number of successfully completed iterations of a task divided by the total number of runs for a task. The simulation and its success is evaluated using VIMA-Bench [16], and as there are 150 iterations per task as defined in the training dataset, each task if run 150 times. For success rate, a higher percentage is better. The equation for success rate can be found at 1.

Placement Generalization: All prompts, including actions, objects, and their textures, are seen during training, but only the placement of objects on the tabletop is randomized in the evaluation.

Combinatorial Generalization: All textures and objects are seen during, training, but new combinations of them appear in the evaluation.

Novel Object Generalization: In the evaluation, prompts and the simulated workspace include novel textures and objects that are unseen during training.

Figure 1. Definitions of investigated task generalization.

$$\text{Success Rate} = \frac{\text{\# of Successful Instances}}{\text{\# of Total Instances}} \tag{1}$$

The second metric, cosine similarity, is used in the latent space analysis to analyze the similarity between the T5 embeddings of base prompts and the associated embeddings from attack prompts. A higher cosine similarity indicates a greater similarity between the base prompt and attack prompt embeddings. We leverage cosine similarity to understand by much how much the attack prompts differ from the base prompts in the VIMA model, and how well our system is able to realign the attack prompts back to the area of the latent space where base prompts reside. Cosine similarity is defined at 2, and for calculating the similarity we flatten the base prompt tokens and attack prompt tokens into one-dimensional vectors.

$$S_C(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \tag{2}$$

3.3. CLARITY Alignment Layer

As seen in Figure 3, we develop a novel additional alignment layer to the VIMA architecture to defend against the prompt attacks and align them back to being more similar to the base prompts. We developed two approaches for incorporating the Multilayer Perceptions (MLPs) in our model: The first accepts padded prompt token embeddings, and the second accepts unpadded prompt token embeddings. Both

Simple Rephrase: Changing the structure of the prompt while preserving the original meaning.

Extend Rephrase: Extends the prompt by using more words while preserving the original meaning.

Color Rephrase: Changes the adjectives within the prompt that describe the object's color.

Object Rephrase: Changes the adjectives within the prompt that describe the object's properties.

Noun Rephrase: Replaces the noun in the prompt, while preserving the original meaning.

Figure 2. Definitions of prompt attack types.

MLPs utilize three fully connected layers and match the output from the T5 transformer.

For the first MLP, we pad the output token sequence with zero padding tokens until the sequence length reaches 30 tokens. We chose this upper boundary because none of the prompt embeddings exceeded a length of 30 tokens. For this MLP we flatten the tokens before passing through the alignment layer and during inference we crop the padded tokens from the post-alignment layer output back to the original T5 output sequence length. The second MLP does not utilize any padding or flattening, and the sequence length inputted into the alignment layer is the same as the sequence length after the alignment layer both in training and inference.

4. Experiments

4.1. Latent Space Analysis

For the latent space analysis, we explored the cosine similarity between the base prompts and the attack prompts, and the trends in the cosine similarity as prompts move further from the original base prompt space via adding noise. For both analyses, we pass the original prompts through the T5 LLM and obtain the embedding tokens, which we flatten into a longer one-dimensional vector. For base vs. attack prompts, we directly compare the similarity of the base prompt and attack prompt and average the similarity of the 150 iterations of the task. For the base prompt plus noise, we investigate only on the visual manipulation task, and perform a similar process for obtaining the average similarity.

Tables 1 and 2 show the results of the cosine similarity analysis on the base prompts versus the attack prompts under the Model name ”VIMA Baseline”. The results of the analysis show the the attack prompts are notably dissimilar from the original base prompts, and occupy a different position in the latent space compared to the base prompts. However, while the cosine similarities are all significantly lower, the correlation between cosine similarity and success rate is not equal across the tasks, as while visual manipulation and rearrange are notably affected by attacks, scene understanding is only minorly affected.

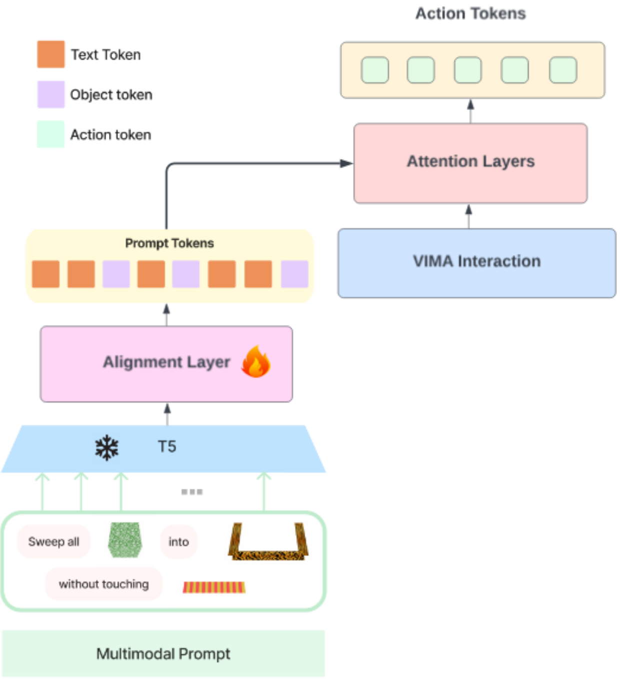


Figure 3. Modified VIMA Architecture with CLARITY Alignment Layer

Figure 4 shows the results of the investigation on the visual manipulation base prompts plus noising. Our results show that, at least for visual manipulation, lower cosine similarity in the prompt has a slow descent in the success rate up until around 0.2-0.3 similarity, where performance rapidly decreases. This is similarity to the results we found in our base prompt vs attack prompt investigation, as the similarities in visual manipulation are around 0.2-0.3 and dropped in success rate around 10-20% for each attack.

4.2. Training

We trained both MLPs with the following specifications: (1) a learning rate of 1e-3, (2) the AdamW optimizer, (3) batch size of 32, (4) 5 epochs for the Unpadded MLP, and 20 epochs for the Padded MLP, and (5) using mean square error as our loss function. We train the model using attack prompt embeddings as our input and base prompt embeddings as our target, enabling learning to align attack prompts to the base prompt space. Following training, both models were then integrated back into the VIMA model for testing.

		Placement ↑			Combinatorial ↑			Novel Object ↑		
Model	Attack	VM	R	SU	VM	R	SU	VM	R	SU
VIMA Baseline	Simple	89.3	68.0	98.7	91.3	73.3	100	80.7	72.0	98.0
	Extend	87.3	32.7	96.7	82.7	36.6	96.6	68.7	26.0	90.0
	Color	82.7	70.0	94.7	83.3	74.7	97.3	72.0	78.7	89.3
	Noun	86.7	93.3	99.3	84.0	88.7	98.7	71.3	90.7	97.3
	Object	83.3	58.0	97.3	79.3	60.7	95.3	68.7	54.0	88.7
	No Attack	100	99.3	100	100	99.3	100	99.3	100	99.3
MLP w/ unpad	Simple	100	99.6	99.3	99.3	98.7	99.3	99.3	99.3	100
	Extend	100	99.6	98.6	99.3	98.7	99.3	99.3	97.3	100
	Color	100	97.2	99.3	99.3	98.7	99.3	99.3	97.3	99.3
	Noun	100	97.3	97.3	99.3	97.2	98.6	99.3	98.0	100
	Object	99.3	97.1	99.3	99.3	96.7	98.6	99.3	98.6	100
	No Attack	100	97.3	99.3	100	98.7	98.6	100	97.3	100
MLP w/ pad	Simple	100	70.0	99.3	100	71.3	98.6	99.3	76.0	100
	Extend	100	68.6	99.3	99.3	68.7	99.3	99.3	75.3	100
	Color	100	66.7	98.6	99.3	67.3	100	99.3	72.7	100
	Noun	100	75.3	98.7	99.3	72.6	99.3	99.3	79.3	100
	Object	100	67.3	98.7	99.3	69.3	99.3	99.3	76.0	100
	No Attack	100	67.3	99.3	99.3	66.0	99.3	99.3	72.7	100

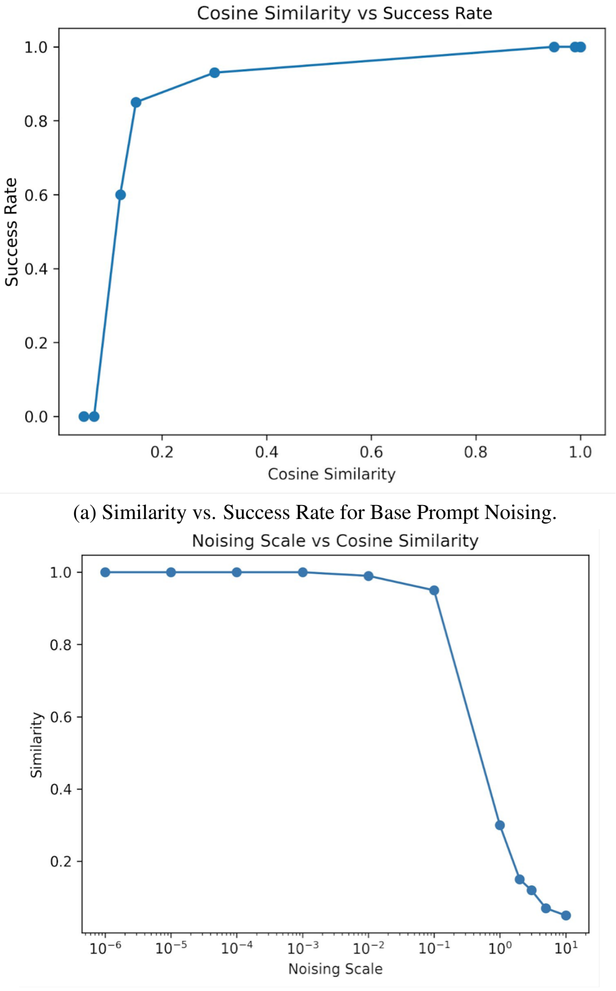
Table 1. Success rate for all models for the different tasks, rephrasing attacks, and partition combination. Greater number is better. VM: Visual Manipulation. R: Rearrange. SU: Scene Understanding.

		Placement ↑			Combinatorial ↑			Novel Object ↑		
Model	Attack	VM	R	SU	VM	R	SU	VM	R	SU
VIMA Baseline	Simple	25.4	45.4	40.7	25.2	45.1	34.8	23.8	45.8	37.2
	Extend	24.9	26.0	25.1	24.8	24.8	23.3	23.5	24.5	26.2
	Color	24.8	36.3	21.8	25.2	36.4	21.9	24.0	37.0	22.0
	Noun	25.2	63.7	26.8	25.0	62.9	27.0	24.5	64.8	26.9
	Object	25.2	34.4	21.3	24.8	33.9	20.2	24.3	32.4	21.6
	No Attack	100	100	100	100	100	100	100	100	100
MLP w/ unpad	Simple	99.6	94.5	83.5	96.9	94.7	82.4	97.0	94.5	82.8
	Extend	99.6	94.1	82.6	96.9	94.6	81.7	96.8	93.9	81.5
	Color	97.2	95.2	82.2	97.0	95.0	80.5	96.9	95.0	80.4
	Noun	97.3	95.7	84.6	97.2	95.6	82.8	97.1	95.8	82.5
	Object	97.1	94.4	82.3	97.2	93.4	81.7	97.1	94.7	80.8
	No Attack	97.3	89.7	79.5	83.8	89.7	78.6	84	89.6	78.2
MLP w/ pad	Simple	99.6	92.5	83.2	99.5	93.7	78.4	97.8	93.9	80.9
	Extend	99.6	93.5	84.1	99.5	93.6	79.4	97.7	93.8	80.9
	Color	99.6	93.5	83.8	99.5	93.8	78.3	97.6	93.7	80.0
	Noun	99.6	93.9	84.2	99.5	93.9	79.5	97.8	94.1	81.4
	Object	99.6	93.7	84.5	99.5	93.6	78.5	97.7	94.0	80.3
	No Attack	97.3	92.5	81.8	97.3	92.5	77.2	96.3	93.0	79.7

Table 2. Cosine Similarity for all models for the different tasks, rephrasing attacks, and partition combination. Greater numbers is better. VM: Visual Manipulation. R: Rearrange. SU: Scene Understanding

4.3. Results

Table 1 provides a comprehensive overview of the success rates obtained through the replication of baseline tasks as



(a) Similarity vs. Success Rate for Base Prompt Noising.

(b) Noise Level vs. Similarity

Figure 4. Result of token embedding analysis on base prompt noising. There can be significant deviation from base prompt embeddings with only minor performance loss, but similarity below 0.3 causes significant performance loss.

detailed in the adversarial attack paper [45] and the results of the MLP models. We achieved similar results to the original VIMA paper, and we can see that results have drastically improved with the addition of our MLP-based model in visual manipulation (VM), rearranging (R), and scene understanding (SU).

Table 2 offers a comprehensive overview of the cosine similarity scores across all attacks and tasks, demonstrating the similarity between the base prompts and attack prompts before the alignment layer and after the unpadded and padded alignment layers. The table shows that the alignment layer is able to effectively make attack prompts similar to the base prompts, even on attack types that were not seen in the training set and higher complexity tasks that were not trained for.

The MLP utilizing the unpadded token inputs performs extremely well, showing a significant improvement and near-complete success rate recovery in the results after being trained on the aforementioned tasks. This improvement suggests that similar attack prompts are being mapped toward the locations of the corresponding successful base embeddings. On the other hand, the MLP employing padded tokens performs significantly better than the original attacks, but to a lesser extent compared to its unpadded counterpart. The introduction of padding the input embedding with meaningless tokens might have inadvertently led to the loss of meaningful information within the model. The addition of padded tokens can obscure the significance of the original prompt, ultimately affecting the model's performance.

4.4. Evaluation on Task Complexity and Certain Attacks

We conducted an assessment of success scores for Novel Object, the hardest level of difficulty for the model. This level of difficulty incorporates new and complex textures and objects resulting in intricate language prompts. Despite the absence of direct training on this level, our model demonstrated remarkable performance, likely attributed to its training on both the placement and combinatorial levels of difficulty. This gives evidence that our system is able to perform well in new situations outside of training and is not completely overfit to our training set. Although placement and combinatorial had simpler textures, their analogous prompt structures contributed to the model's success on the challenging Novel Object-level tasks. Furthermore, our model was not trained on the simple and object attacks of the rearrangement task across all three difficulty levels, and still led to good performance. This gives evidence that our alignment layer can lead to performance improvements in similar unseen prompt attack types. Nevertheless, the model's notable performance highlights the benefits of training on similar tasks, underscoring its adaptability and robustness across varying levels of difficulty.

5. Conclusion

In this work, we seek to answer the questions raised in [45] on why prompt-based adversarial attacks work on LLM/VLM-based robotic systems, and how to develop a quick and effective defense against such attacks to return performance. Our experiments give evidence to the idea that the reason why prompt-based attacks work is due to them occupying a different position in the latent space compared to the original instruction prompts, which would be caused by a limited training set being used in the original model training. However, our experiments also show that the task definition and prompt has a strong effect too, as different tasks had different levels of performance loss compared to the cosine similarity. To recover lost performance, we added an additional alignment layer made from a 3-layer perceptron on top of the T5 LLM in VIMA and trained the alignment layer on adversarial prompts against the original prompts under the theory that making adversarial prompts have more similar embeddings to the original prompts will return performance. Our results show that this alignment layer is an effective solution to the prompt attack problem, recovering performance back to near 100% on attack prompts with minimal to no loss on the original instruction prompt performance.

5.1. Future Work

Our current model is trained on the tasks that were presented in [45]. Our work highlights the need for further investigation into understanding and defending against prompt-based attacks. We would like to investigate KnowNo and Instruct2Act's performance loss and recovery using our alignment layer system to completely address the concerns raised in [45], as well as exploring how well prompt attacks work on other LLM/VLM-based robotic systems.

Our VIMA baseline results exhibit substantial declines in success rates for attacks associated with rearrangement, noticeable decreases in attacks linked to visual manipulation, and a slight dip in attacks related to the scene understanding task. Conducting an in-depth analysis of these trends can offer valuable insights into the underlying reasons behind these variations. As a result, further development in our MLP-based model can be made to accommodate certain trends.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[2] Erik Billing, Julia Rosén, and Maurice Lamb. Language models for human-robot interaction. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 905–906, 2023. 3

[3] Sebastian G Bouschery, Vera Blazevic, and Frank T Piller. Augmenting human innovation teams with artificial intelligence: Exploring transformer-based language models. *Journal of Product Innovation Management*, 40(2):139–153, 2023. 1

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2

[5] Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. Latte: Language trajectory transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7287–7294. IEEE, 2023. 2

[6] Vishnu Sashank Dorbala, James F Mullen Jr, and Dinesh Manocha. Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation. *IEEE Robotics and Automation Letters*, 2023. 2

[7] Haolin Fan, Xuan Liu, Jerry Ying Hsi Fuh, Wen Feng Lu, and Bingbing Li. Embodied intelligence in manufacturing: leveraging large language models for autonomous industrial robotics. *Journal of Intelligent Manufacturing*, pages 1–17, 2024. 2

[8] Ed Felten, Manav Raj, and Robert Seamans. How will language modelers like chatgpt affect occupations and industries? *arXiv preprint arXiv:2303.01157*, 2023. 1

[9] Yu Fu, Yufei Li, Wen Xiao, Cong Liu, and Yue Dong. Safety alignment in nlp tasks: Weakly aligned summarization as an in-context attack. *arXiv preprint arXiv:2312.06924*, 2023. 3

[10] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90, 2023. 3

[11] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10877, 2023. 1

[12] Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv preprint arXiv:2310.05694*, 2023. 1

[13] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 2

[14] Siyuan Huang, Zhengkai Jiang, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model, 2023. 1

[15] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023. 3

[16] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. In *Fortieth International Conference on Machine Learning*, 2023. 1, 2, 3, 4

[17] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8018–8025, 2020. 3

[18] Carina Kauf, Anna A Ivanova, Giulia Rambelli, Emmanuele Chersoni, Jingyuan Selena She, Zawad Chowdhury, Evelina Fedorenko, and Alessandro Lenci. Event knowledge in large language models: the gap between the impossible and the unlikely. *Cognitive Science*, 47(11):e13386, 2023. 1

[19] Zsolt Kira. Awesome-llm-robotics, 2022. 2

[20] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023. 3

[21] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. 1

[22] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*, 2020. 3

[23] Lei Li, Yongfeng Zhang, and Li Chen. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1348–1357, 2023. 1

[24] Jing Liang, Peng Gao, Xuesu Xiao, Adarsh Jagan Sathyamoorthy, Mohamed Elnoor, Ming Lin, and Dinesh Manocha. Mtg: Mapless trajectory generator with traversability coverage for outdoor navigation. *arXiv preprint arXiv:2309.08214*, 2023. 2

[25] Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*, 2023. 1, 3

[26] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 3

[27] Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. *arXiv preprint arXiv:2311.10774*, 2023. 2

[28] Fuxiao Liu, Yaser Yacoob, and Abhinav Shrivastava. Covid-vts: Fact extraction and verification on short video platforms. *arXiv preprint arXiv:2302.07919*, 2023. 3

[29] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 259–274. Springer, 2020. 2

[30] Ariana Martino, Michael Iannelli, and Coleen Truong. Knowledge injection to counter large language model (llm) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer, 2023. 2

[31] Andriy Mulyar, Ozlem Uzuner, and Bridget McInnes. Mt-clinical bert: scaling clinical information extraction with multitask learning. *Journal of the American Medical Informatics Association*, 28(10):2108–2115, 2021. 1

[32] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 2

[33] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *international conference on machine learning*, pages 17359–17371. PMLR, 2022. 2

[34] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*, 2019. 3

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 3

[36] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023. 2

[37] Jacob Rintamaki. Everything-llms-and-robotics, 2023. 2

[38] Christian Schlarmann and Matthias Hein. On the adversarial robustness of multi-modal foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3677–3685, 2023. 3

[39] Lorenzo Serina, Luca Putelli, Alfonso Emilio Gerevini, and Ivan Serina. Synonyms, antonyms and factual knowledge in bert heads. *Future Internet*, 15(7):230, 2023. 1

[40] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023. 2

[41] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 2

[42] Ka Chun Shum, Jaeyeon Kim, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Language-driven object fusion into neural radiance fields with pose-conditioned dataset updates. *arXiv preprint arXiv:2309.11281*, 2023. 2

[43] Javier Villena Toro and Mehdi Tarkian. Model architecture exploration using chatgpt for specific manufacturing applications. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, page V002T02A091. American Society of Mechanical Engineers, 2023. 1

[44] Xingzhi Wang, Nabil Anwer, Yun Dai, and Ang Liu. Chatgpt for design, manufacturing, and education. *Procedia CIRP*, 119:7–14, 2023. 1

[45] Xiyang Wu, Ruiqi Xian, Tianrui Guan, Jing Liang, Souradip Chakraborty, Fuxiao Liu, Brian Sadler, Dinesh Manocha, and Amrit Singh Bedi. On the safety concerns of deploying llms/vlms in robotics: Highlighting the risks and vulnerabilities. *arXiv preprint arXiv:2402.10340*, 2024. 1, 3, 7, 8

[46] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023. 2

[47] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*, 2019. 3

[48] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. 3