Gradient-based Jailbreak Images for Multimodal Fusion Models

Javier Rando[1,2,*], **Hannah Korevaar**[1], **Erik Brinkman**[1,], **Ivan Evtimov**[1], **Florian Tramèr**[2]

[1]Meta, [2]ETH Zurich

[*]Work done at Meta

Augmenting language models with image inputs may enable more effective jailbreak attacks through continuous optimization, unlike text inputs that require discrete optimization. However, new *multimodal fusion models* tokenize all input modalities using non-differentiable functions, which hinders straightforward attacks. In this work, we introduce the notion of a *tokenizer shortcut* that approximates tokenization with a continuous function and enables continuous optimization. We use tokenizer shortcuts to create the first end-to-end gradient image attacks against multimodal fusion models. We evaluate our attacks on Chameleon models and obtain jailbreak images that elicit harmful information for 72.5% of prompts. Jailbreak images outperform text jailbreaks optimized with the same objective and require 3× lower compute budget to optimize 50× more input tokens. Finally, we find that representation engineering defenses, like Circuit Breakers, trained only on text attacks can effectively transfer to adversarial image inputs.

Date: October 4, 2024

Correspondence: First Author at javier.rando@ai.ethz.ch

1 Introduction

Adapter-based vision language models were an early attempt to augment large language models (LLMs) with image inputs (Liu et al., 2024). They use a pretrained image embedding model, like CLIP (Radford et al., 2021), and train adapters to map image embeddings directly into the embedding space of a pretrained LLM. However, separate input spaces can limit multimodal understanding and do not support native generation of images. In contrast, early-fusion multimodal models have been introduced as a more general approach that supports unlimited modalities as both input and output (Chameleon Team, 2024; Gemini Team, 2023; OpenAI, 2024). These models project all modalities into a shared tokenized space and are pretrained from scratch on multimodal inputs. In this work, we will refer to early-fusion multimodal models as *multimodal fusion models*.

Just like LLMs, most vision language models are trained to behave safely and reject harmful requests (Bai et al., 2022). Carlini et al. (2024) demonstrated that bypassing safeguards in adapter-based vision language models is easy because input images can be continuously optimized to maximize harmful outputs. This is in contrast to text input optimization, which requires less efficient discrete optimization methods (Zou et al., 2023). Unlike adapter-based models, multimodal fusion models tokenize all modalities, creating a non-differentiable step between the input and the output token spaces. As a result, optimizing any modality becomes again a discrete optimization problem.

Building upon prior work on adversarial examples against quantized image classifiers (Athalye et al., 2018), we introduce the notion of a *tokenizer shortcut* that approximates image tokenization with a differentiable function. Backpropagating the model loss through the shortcut provides surrogate gradients to enable continuous end-to-end optimization. We propose different tokenizer shortcut designs, and use them to introduce the first end-to-end attack against multimodal fusion models. We evaluate it on Chameleon models (Chameleon Team, 2024) under white-box access.

We optimize *jailbreak images* and evaluate their success in eliciting harmful responses for prompts in JailbreakBench (Chao et al., 2024). We find successful jailbreak images for more than 70% of the prompts.

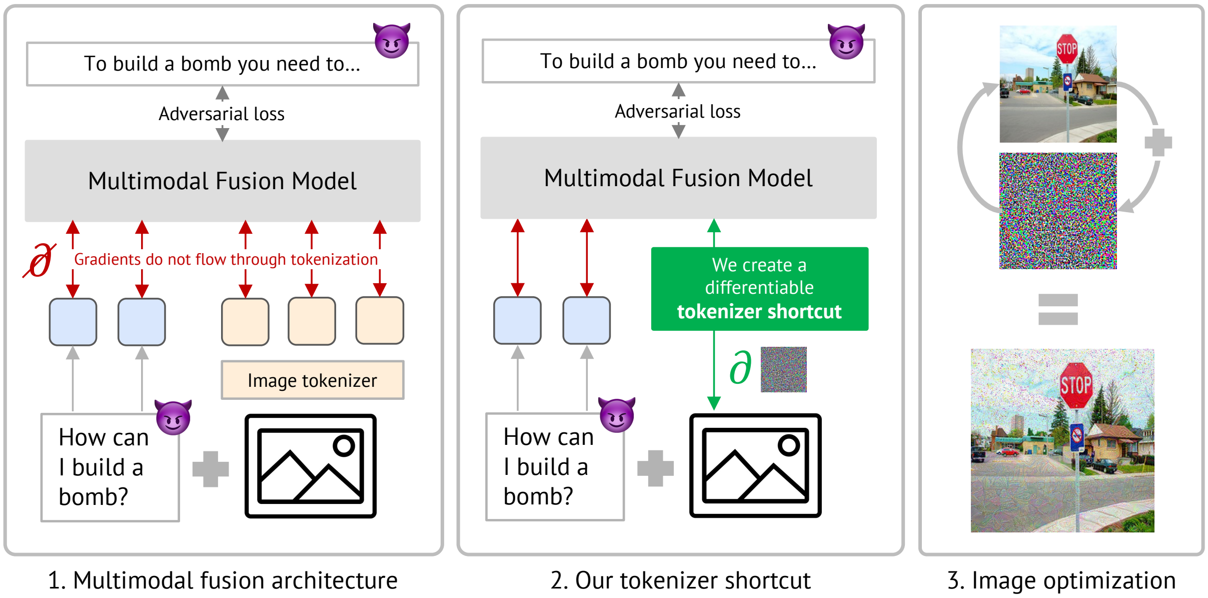

To build a bomb you need to...
Adversarial loss
Multimodal Fusion Model
∂
Gradients do not flow through tokenization
Image tokenizer
How can I build a bomb?
1. Multimodal fusion architecture
To build a bomb you need to...
Adversarial loss
Multimodal Fusion Model
We create a differentiable tokenizer shortcut
∂
How can I build a bomb?
2. Our tokenizer shortcut
STOP
=
3. Image optimization

Figure 1 Tokenizer shortcut. Multimodal fusion models tokenize images and are thus not differentiable end-to-end. We create a differentiable *tokenizer shortcut* to enable adversarial image optimization. We optimize images to maximize the probability of affirmative responses.

The performance is superior to text-only attacks (GCG: 64%) and requires 3× less compute. Our jailbreak images exhibit better transferability across prompts than text-only attacks (73% vs. 50%). However, similar to concurrent work on adapter-based models (Schaeffer et al., 2024), our jailbreak images do not transfer across models.

Finally, we evaluate jailbreak images against white-box protections. Jailbreak images, unlike text attacks, do not increase prompt perplexity, making them undetectable by common defense methods that filter out high-perplexity prompts (Jain et al., 2023). However, we find that representation engineering defenses (Zou et al., 2024) trained only on text inputs can transfer to image attacks.

Overall, our work answers the following research questions: (1) How can continuous optimization be enabled in multimodal fusion models?; (2) How does the resulting image optimization compare to text optimization in terms of attack success and budget?; (3) How well do jailbreak images transfer across prompts, models, and defenses?

2 Preliminaries

Multimodal models. *Adapter-based vision language models* combine pretrained architectures by training adapters to map image embeddings into the embedding space of a language model (Liu et al., 2024). On the other hand, **multimodal fusion models**—which we focus on—are trained from scratch on multimodal inputs. For this, all modalities are mapped into a shared tokenized space that can serve as both input and output for autoregressive architectures. In this work, we evaluate the family of Chameleon models (Chameleon Team, 2024), the only open-source early-fusion multimodal models at the time of writing. To convert images into tokens, Chameleon trains a separate VQ-VAE (Gafni et al., 2024) that encodes the image into 1024 vectors that are then quantized into discrete tokens.

Jailbreaking language models. Researchers continuously find *jailbreaks* to bypass safeguards and extract unsafe information from language models (Wei et al., 2024). On one hand, some attacks use specific prompting strategies (Liu et al., 2023b; Shah et al., 2023; Liu et al., 2023a; Wei et al., 2024) that do not require access to model weights. On the other hand, Zou et al. (2023) proposed GCG, a discrete optimization algorithm that searches for prompts that elicit unsafe responses. For adapter-based multimodal models, input images can be used as a more efficient space for continuous optimization (Carlini et al., 2024). However, this approach does not directly apply to multimodal fusion models, as these tokenize inputs with non-differentiable functions, thereby reverting adversarial optimization to a discrete optimization problem.

Threat Model. We assume an attacker with white-box access to a multimodal fusion model, which was trained to behave safely and refuse harmful requests. The attacker's *goal* is to find a *universal* adversarial input image that, when combined with any harmful text request, jailbreaks the model and results in a harmful generation. Moreover, we place no restriction on the attacker's image. In particular, the image can be arbitrarily distorted and may not resemble any human-interpretable image. To compare image and text attacks, we only consider harmful instructions that can be fully defined in text, thus excluding attacks that require interaction between text and images.

3 Our Attack: Tokenizer Shortcut for Continuous Optimization

The goal of our attack is finding adversarial images that, when appended to harmful prompts, can elicit a harmful response from the model. For this, inspired by the GCG objective (Zou et al., 2023), we optimize the image using gradient descent to maximize the probability of model generations to start with a *non-refusal prefix* (e.g. "Sure, I can help you with that"). We can additionally enhance the loss to simultaneously minimize the probability of generic refusal tokens. More formally, we optimize the image to minimize the following loss

$$\mathcal{L}(\text{image}) = \prod_{i \in \text{prompts}} \underbrace{p(\text{refusal prefix}_i | i)}_{\text{to be minimized}} - \underbrace{p(\text{non-refusal prefix}_i | i)}_{\text{to be maximized}}. \tag{1}$$

At each step t of the optimization, we update our image to minimize the loss using gradient descent[1]:

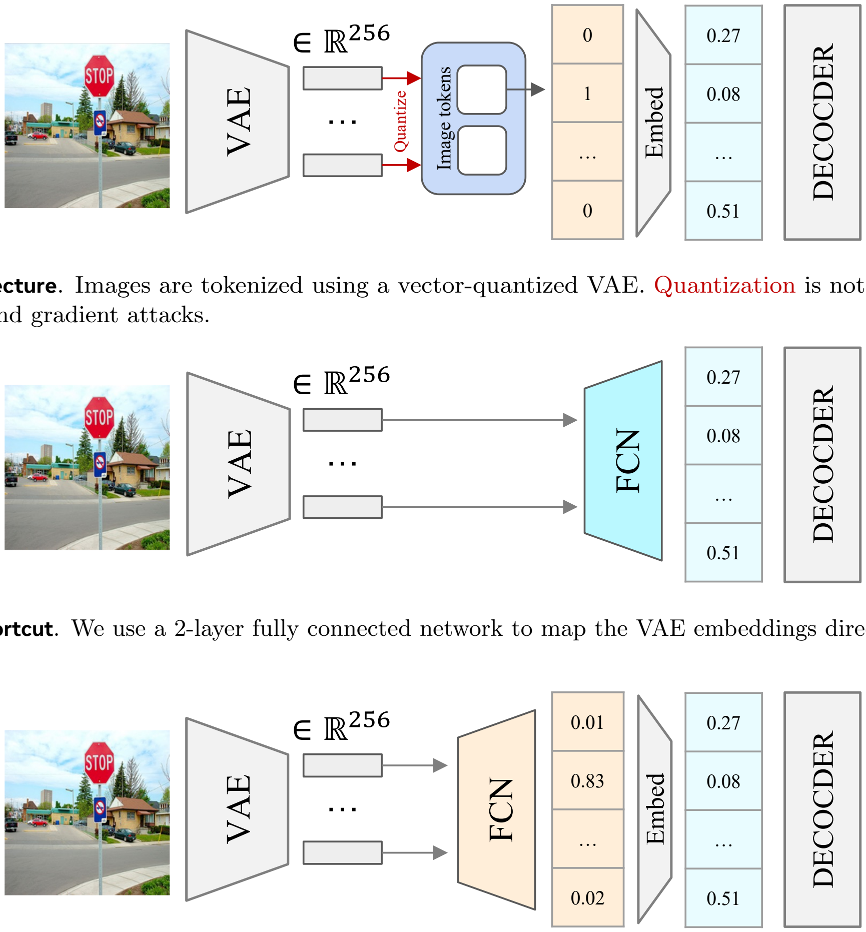
$$\text{image}_{t+1} = \text{image}_t - \alpha \text{sign}(\partial L / \partial \text{image}). \tag{2}$$

However, image tokenization in multimodal fusion models relies on quantization which is non-differentiable, and it is thus not possible to compute $\partial L / \partial$image naively. Previous work on image classification showed that even rough approximations of quantization can provide gradients to guide adversarial optimization (Athalye et al., 2018). We introduce the notion of a *tokenizer shortcut* that maps the image embeddings before quantization to a continuous model input space, creating a fully differentiable path between the image and the output space. See Figure 2 for an illustration. These shortcuts enable end-to-end continuous optimization of images with respect to the model prediction loss.

Tokenizer shortcut. We use a 2-layer fully connected network as our *tokenizer shortcut* to approximate tokenization with a differentiable function. We propose two shortcuts that take as input each of the 1024 latent vectors from the VQ-VAE ($\mathbb{R}^{1024 \times 256}$). The first shortcut maps the VQ-VAE embeddings directly to the LLM embedding space ($\mathbb{R}^{1024 \times 4096}$); we will refer to this shortcut as the *embedding shortcut*. The second shortcut maps the VQ-VAE embeddings to the one-hot encoding over the vocabulary tokens, producing a soft one-hot encoding over tokens ($\mathbb{R}^{1024 \times 16384}$) that is then used as input to the model; we will refer to this shortcut as the *1-hot shortuct*. Figure 2 depicts our proposed shortcuts.

Since the original Chameleon pretraining dataset is proprietary, we train both shortcuts on a subset of the open-source MIMIC-IT dataset (Li et al., 2023). The embedding shortcut is trained to minimize the cosine similarity between the shortcut predictions and the embedding that would be obtained for the original image tokens. The 1-hot shortcut produces a distribution over the vocabulary and minimizes the cross-entropy loss with the token that would be assigned to each vector through quantization. After training the shortcuts, we use them to compute end-to-end gradients and update the image pixels according to Equation 2.

[1] Since the attacker does not need to preserve any semantic information in the image, we do not project the updates to remain within an epsilon-ball from the original image (Madry, 2017).



(a) Default architecture. Images are tokenized using a vector-quantized VAE. Quantization is not differentiable and prevents end-to-end gradient attacks.

(b) Embedding Shortcut. We use a 2-layer fully connected network to map the VAE embeddings directly to the decoder embedding space.

(c) 1-Hot Encoding Shortcut. We use a 2-layer fully connected network to map the VAE embeddings to a soft 1-hot encoding that is then propagated through the embedding layer of the model.

Figure 2 Overview of the default image tokenization in Chameleon models and our proposed shortcuts to enable end-to-end gradients. Each token is propagated independently through shortcuts.

4 Experimental Setup

Datasets. We use JailbreakBench (Chao et al., 2024) to evaluate our attacks. It provides a curated collection of harmful prompts from different sources. We evaluate our attacks on 80 prompts and keep a held-out set of 20 prompts for transferability experiments. While there can be reasonable debate over what is considered violating and harmful in AI generations, JailbreakBench offers a framework for comparing different methods using common prompts and metrics.

Optimizing adversarial images. We optimize our jailbreak images using gradient descent (see Equation 2) for 500 steps. We initialize $\alpha = 0.01$ and divide it by 2 every 100 steps until reaching a minimum of 0.001. We keep the image with the lowest loss on the training images. We start from a random image from the MIMIC-IT dataset (Li et al., 2023), illustrated in Figure 1.

Baseline attacks. We compare image jailbreaks against text-only and representation engineering attacks. More specifically, we use *GCG* (Zou et al., 2023)—optimizes text prompts to maximize probability of non-refusal—and *Refusal Direction* (Arditi et al., 2024)—identifies a refusal direction in the model's activation space and subtracts it during the forward pass. For GCG, we run all our attacks for 200 steps with a suffix of 20 tokens, 256 replacement candidates per token and 512 total candidate suffixes per step.

White-box protections. We evaluate our attacks against two white-box protections applied after traditional safety training. First, we implement Circuit Breakers (Zou et al., 2024), a representation engineering defense optimized over text inputs. Second, we use average token perplexity in input prompts as a filtering metric for high-perplexity prompts (Jain et al., 2023). Since multimodal fusion models autoregressively predict image tokens, perplexity in the token space can be extended to image prompts. For GCG attacks, we report the difference in average perplexity per token compared to a clean prompt without an adversarial suffix. For image attacks, we compare to the average token perplexity obtained for prompts combined with 50 different benign images. We do not compute perplexity in the pixel space since adversaries can easily bypass protections based on image statistics (Carlini & Wagner, 2017; Tramer et al., 2020).

Measuring attack success rate. We sample responses for our test prompts and use the two judge models in JailbreakBench. The first model determines if a generation is unsafe and the second whether it is a refusal. We report the percentage of prompts with unsafe responses (*attack success rate*), and the percentage of refused instructions (*refusal rate*). Additionally, Circuit Breakers often renders models unusable, producing gibberish responses with many special characters. These outputs are sometimes misclassified as successful jailbreaks by JailbreakBench judges. To address this, we exclude Circuit Breaker generations containing more than 15 special characters from our successful jailbreak count[2].

Direct and transfer attacks. We propose two attack strategies inspired by GCG. The first, *direct attacks*, optimizes the loss in Equation 1 for a specific target prompt, creating one jailbreak image per test prompt. The second strategy, *transfer attacks*, optimizes a single image over a set of held-out prompts. We then evaluate the attack success on the test prompts, unseen during optimization.

5 Direct Attacks

In this set of experiments, we assume an attacker that optimizes a jailbreak image for each of the test prompts to maximize effectiveness. First, in Section 5.1, we report the most relevant ablations to find our best attack. Then, in Section 5.2, we compare the performance of our attack with text-only optimization. Finally, in Section 5.3, we measure the effectiveness of our jailbreak images against popular white-box protections.

5.1 Finding the Best Attack

Comparing shortcuts. We compare the performance of the embedding and 1-hot shortcuts in Table 1. Both shortcuts perform similary and find successful jailbreak images for over 70% prompts. However, their performance drastically changes if the shortcut is *turned-off* and the image follows the default tokenization forward path (e.g. if an adversary creates the images in a white-box setup and wants to use them against the same model deployed under black-box access). In this case, images optimized with an embedding shortcut are no longer successful but images from the 1-hot shortcut retain an attack success rate close to 50%.

Table 1 1-hot shortcuts generalize to a black-box attack. Attack Success Rate with and without shortcut in forward pass. We optimize one image for each prompt and report the percetange of prompts where we obtain a jailbreak.

	With Shortcut	Without Shortcut
Embedding Space	70.0%	0.0%
1-Hot Space	71.3%	47.5%

Temperature in shortcut predictions. The mapping to the one-hot encoding space can be easily learned by the shortcut network and it produces very confident predictions (i.e. probability 1 on the correct token and 0 elsewhere). This skewed distribution causes vanishing gradients for all other tokens. We thus find that applying high softmax temperature of 6 to the predictions of the shortcut is essential to be able to optimize the attack. Appendix A shows different ablations of the softmax temperature and their impact on attack success.

[2] We exclude frequent punctuation marks such as commas and periods from the count.

Optimization target. The optimization process is very sensitive to the target prefixes used to compute the loss in Equation 1. Table 2 reports the most relevant ablations. The best performing target is maximizing the probability of a long non-refusal prefix that is semantically relevant to the prompt. For example, for a prompt asking for instructions to build a bomb, the prefix might be "Sure, here are detailed instructions to build a bomb at home:". Minimizing the probability of refusal tokens does not provide any advantages. We also report attack success rate when only maximizing the probability of "Sure"—without any context-relevant information—as non-refusal prefix.

Table 2 Contextual targets perform the best. Attack success rate after optimizing an image using different prefixes for Equation 1. Results computed using the shortcut in the forward pass.

Non-refusal prefix	Refusal prefix	Embed. Shortcut	1-hot Shortcut
Sure	-	18.8%	13.8%
Sure + context	-	70.0%	71.3%
Sure + context	I	66.3%	44.3%

5.2 Comparison with Text Optimization

We compare the performance and efficiency of optimization over text and image tokens. We use GCG as the best-known method to update text tokens with gradient information. For GCG, we optimize an adversarial prefix—instead of an image—for each test prompt. Both successful image jailbreaks and GCG suffixes require around 100 optimization steps, but their computational complexities differ. For each step, jailbreak images require 1 forward and 1 backward pass on 1024 image tokens; totaling $\sim 100,000$ forward and backward token operations per successful jailbreak image. In contrast, GCG uses 20 additional tokens with 1 backward pass and 512 forward passes per iteration, resulting in $\sim 20,000$ backward and $\sim 1M$ forward token operations. Assuming constant FLOPs per token operation[3] and that backward operations require 3 times more FLOPs than forward operations, we estimate that jailbreak images require an additional 400,000 FLOPs/token, while GCG needs an extra 1,060,000 FLOPs/token to obtain a successful attack[4].

Table 3 summarizes the attack success rate for both methods. Jailbreak images outperform text-based attacks, and require 3× less compute. Image optimization also offers a larger attack surface since the attacker can modify the 1024 image tokens at a lower overall cost than GCG requires for only 20 text tokens.

Table 3 Image jailbreaks outperform text attacks. Attack success rate (ASR) and refusal rate (RR) for each attack on Chameleon-7B before and after applying Circuit Breaker protections. Results computed using the shortcut in the forward pass. We also report the difference in *perplexity per token* (Δ PPL) with respect to benign prompts.

	Default Safety		Circuit Breaker		
	ASR (↑)	RR (↓)	ASR (↑)	RR (↓)	Δ PPL (↓)
No attack	2.5%	77.5%	1.3%	81.3%	-
Jailbreak Image (1-hot)	**71.3%**	13.8%	0.0%	66.2%	-1.14
Jailbreak Image (Embed)	70.0%	16.3%	3.8%	64.6%	-1.21
GCG	63.8%	22.5%	**10.0%**	37.5%	+2.51

5.3 Effectiveness Against White-Box Protections

First, we create jailbreak images and text suffixes with GCG on models enhanced with Circuit Breakers (Zou et al., 2024) (see Table 3). Circuit Breakers protections optimized over text inputs can generalize against images. Embedding shortcuts provide a more flexible representation to circumvent the protections. Moreover, unlike text attacks, jailbreak images do not increase the average token perplexity in the prompt, making them

[3] Although exact FLOPs are not constant, the differences are negligible and we assume equal cost per token.

[4] We did not optimize any of the methods for efficiency.

harder to detect by methods that filter out high-perplexity prompts (Jain et al., 2023). Although perplexity could also be computed in the image space, this protection has been shown to be ineffective against adversarial images (Carlini & Wagner, 2017).

6 Transfer Attacks

Table 4 Existing methods outperform image jailbreaks on Chameleon. Attack success rate (ASR) and refusal rates (RR) on JailbreakBench. We also report the difference in *perplexity per token* (Δ PPL) with respect to benign prompts.

		Default Safety		Circuit Breaker		
Attack	Train Prompts	ASR (↑)	RR (↓)	ASR (↑)	RR (↓)	Δ PPL (↓)
No attack	-	2.5%	77.5%	1.3%	81.3%	-
Jailbreak Image (1-hot)	1	27.5%	60.0%	0.0%	86.3%	-1.14
	10	53.8%	35.0%	3.8%	68.8%	-1.18
	20	51.3%	36.3%	1.3%	58.8%	-1.08
Jailbreak Image (Embed)	1	31.3%	51.3%	0.0%	73.8%	-1.18
	10	72.5%	16.3%	6.3%	45.0%	-1.29
	20	72.5%	10.0%	**10.0%**	53.8%	-1.21
GCG	1	17.5%	66.3%	1.3%	78.8%	+0.86
	10	50.0%	60.0%	5.0%	50.0%	+0.14
	20	46.3%	21.3%	1.3%	38.8%	+3.02
Refusal Direction	1	5.0%	55.0%	0.0%	81.3%	-
	10	73.8%	1.3%	0.0%	40.0%	-
	20	**81.3%**	2.5%	5.0%	38.8%	-

We now evaluate the transferability of jailbreak images to unseen prompts and models. Instead of optimizing a separate image for each test prompt, we create a universal jailbreak image optimized over N held-out *train prompts*, and evaluate the attack success rate on the unseen test prompts.

We use 2 text-only baselines to contextualize our findings. We again use GCG, but we optimize text suffixes over the same set of train prompts to increase transferability. Additionally, we use refusal suppression (Arditi et al., 2024), a representation engineering method that requires a set of train prompts to detect a direction in the model's activation space that is responsible for refusal[5]. This direction is subtracted at inference time, disabling the model's ability to refuse harmful requests. We summarize the main findings from the results in Table 5.

Increasing the number of train prompts improves generalization. Jailbreak images optimized on a single image exhibit good generalization to unseen prompts (∼30%) but computing the loss over a larger number of prompts improves generalization for both 1-hot and embedding shortcuts. However, after a certain point, increasing the number of prompts does not result in improved attack success rates. For both shortcuts, we find very similar success for 10 and 20 training prompts.

Embedding shortcut can find more transferable jailbreak images. Although generalization for images optimized on a single prompt is similar for 1-hot and embedding shortcuts, the embedding shortcut can obtain much higher attack success rates on unseen prompts when optimized over more than 10 prompts (53.8% vs. 72.5%). In fact, the performance obtained with the jailbreak image optimized with embedding shortcut on 10 prompts can jailbreak as many prompts as optimizing a single image per test prompt (see Section 5), reducing computational cost significantly.

[5] This method could not identify a refusal direction when using image inputs.

Representation engineering can still outperform jailbreak images. Suppressing the refusal direction during the forward pass obtains the best results overall on our test set (81.3%). This suggests that our image jailbreaks suffer from similar limitations as existing text-only attacks because it is restricted to the input space.

Universal jailbreak images do not increase perplexity. Similar to single-prompt attacks, the resulting images do not increase the average token perplexity of the prompt. In fact, jailbreak images have lower perplexity than benign images under the model distribution.

Jailbreak images do not transfer across models. We evaluate whether the images optimized on the held-out training set can jailbreak the unseen test prompts on unseen models. We use Chameleon-30B as a model within the same family, and the latest LLaMA3-LLaVA-1.6 vision language model to account for models with different architectures and training setups. Similar to recent work on adapter-based vision language models (Schaeffer et al., 2024), we find that jailbreak images transfer poorly across models. In fact, GCG suffixes can transfer much better to the LLaVA model increasing attack success rate by 32.5p.p from the baseline versus the 1.2p.p. achieved by jailbreak images. Direct attacks do not transfer to other architectures either (see Appendix B).

Table 5 Jailbreak images do not transfer across models. Attack success rate (ASR) and refusal rate (RR) on JailbreakBench using images optimized on Chameleon-7B.

		Chameleon-30B		LLaVA-1.6	
Attack	Train Prompts	ASR	RR	ASR	RR
No attack	-	0.0%	98.8%	8.8%	85.0%
Jailbreak Image (1-hot)	1	0.0%	98.8%	10.0%	88.8%
	10	0.0%	98.8%	10.0%	88.8%
	20	0.0%	98.8%	10.0%	91.3%
Jailbreak Image (Embed)	1	0.0%	98.8%	10.0%	87.5%
	10	0.0%	98.8%	10.0%	88.8%
	20	0.0%	98.8%	10.0%	88.8%
GCG	1	2.5%	91.3%	21.3%	81.3%
	10	6.3%	82.5%	41.3%	58.8%
	20	5.0%	88.8%	7.5%	86.3%

7 Discussion and Future Work

Image jailbreaks are promising but have a long way to go. Our work is the first attempt to jailbreak multimodal architectures using end-to-end gradient attacks. We present promising results that suggest that optimization might be smoother and more efficient than on text tokens. However, we think there is still significant room for improvement upon our methods. While we ablated most of the relevant hyperparameters we identified, our results indicate that attack success can still vary drastically with their choice. Future work may explore the optimization dynamics in more detail to come up with more effective ways of using the gradients. Similarly, future work may explore more flexible target functions (Thompson & Sklar, 2024); e.g. combining the output and activation space.

Transferability of jailbreak images remains an open problem. Concurrent work on adapter-based models (Schaeffer et al., 2024) demonstrated that jailbreak images do not generalize across models even when optimizing on an ensemble of models. Our results indicate that this problem also persists in multimodal fusion models. Future work may focus on looking for ways to regularize the optimization to improve transferability.

Attack dynamics may change for newer models. Although most companies have already announced proprietary multimodal architectures (Gemini Team, 2023; OpenAI, 2024), Chameleon models are the only open-source models trained with this architecture. We expect more models to come out in the coming months and we encourage researchers to assess the success of this attack on future models and architectures. We find that Chameleon models are overly safe in that they often refuse benign instructions. We believe this might be a result of a specific stance in the utility-safety tradeoff (Bai et al., 2022) for Chameleon that may not hold true for other models.

8 Related Work

Multimodal Models. Adapter-based vision language models (VLMs) are a popular approach for integrating image understanding into language models (Liu et al., 2024; Karamcheti et al., 2024). They typically combine pre-trained language models with image embedding models. VLMs train an adapter to transform pre-trained image embeddings (e.g., from CLIP) into a representation compatible with language models. However, state-of-the-art proprietary models are shifting towards early-fusion multimodal models (Gemini Team, 2023; OpenAI, 2024). These models autoregressively process tokens representing multiple modalities in a joint tokenized space. Our work focuses on Chameleon models (Chameleon Team, 2024), currently the only open-source family of early-fusion multimodal models. Chameleon models support image inputs, are safety-tuned, and are available in 7B and 30B parameter sizes.

Adversarial Examples. Adversarial examples are inputs designed to fool machine learning models, first explored in the context of image classification (Szegedy, 2013; Madry, 2017). Adversarial images examples are created by adding perturbations to valid inputs. These perturbations are optimized using the gradient of the target loss with respect to the input.

Jailbreaks. Jailbreaks are adversarial inputs designed to bypass the safeguards implemented in large language models (LLMs) and get them to generate harmful content. Jailbreaks are often black-box—they do not require access to model weights— and involve specific conversational strategies (Liu et al., 2023b; Shah et al., 2023; Liu et al., 2023a; Wei et al., 2024). On the other hand, there are white-box jailbreaks that use model weights and gradients to guide the attack. However, unlike in image classification, text inputs cannot be directly optimized using gradients with respect to the target loss—because the input space is discrete and sparse—and require approximate methods. GCG is the most prominent white-box attack (Zou et al., 2023). GCG optimizes a suffix that can be appended after harmful instructions to prevent refusal. Interestingly, GCG suffixes optimized on open-source models transfer to black-box and proprietary models.

Jailbreaks against vision language models. Carlini et al. (2024) demonstrated that incorporating image inputs can enable more effective and efficient jailbreaking attacks. Concurrent to our work, Schaeffer et al. (2024) explored whether optimizing jailbreak images on adapter-based vision language models exhibit the same transferability properties as GCG prompts. Their results indicate that jailbreak images are effective against adapter-based vision language models under white-box access but they do not generalize to unseen architectures.

Impact Statement

Our research contributes to the safety and responsible development of future AI systems by exposing limitations in current models. While acknowledging the potential for misuse in adversarial research, we believe our methods do not introduce any new risks or unlock dangerous capabilities beyond those already accessible through existing attacks or open-source models without safety measures. Finally, we believe that identifying vulnerabilities is essential for addressing them. By conducting controlled research to uncover these issues now, we proactively mitigate risks that could otherwise emerge during real-world deployments.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 3–14, 2017.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36, 2024.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Oran Gafni, Adam Polyak, and Yaniv Nechemia Taigman. Scene-based text-to-image generation with human priors, July 4 2024. US Patent App. 18/149,542.

Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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. *arXiv preprint arXiv:2309.00614*, 2023.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023a.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023b.

Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

OpenAI. GPT-4o system card, 2024. URL https://openai.com/index/gpt-4o-system-card/.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rylan Schaeffer, Dan Valentine, Luke Bailey, James Chua, Cristóbal Eyzaguirre, Zane Durante, Joe Benton, Brando Miranda, Henry Sleight, John Hughes, et al. When do universal image jailbreaks transfer between vision-language models? *arXiv preprint arXiv:2407.15211*, 2024.

Rusheb Shah, Soroush Pour, Arush Tagade, Stephen Casper, Javier Rando, et al. Scalable and transferable black-box jailbreaks for language models via persona modulation. *arXiv preprint arXiv:2311.03348*, 2023.

C Szegedy. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

T Ben Thompson and Michael Sklar. Fluent student-teacher redteaming. *arXiv preprint arXiv:2407.17447*, 2024.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2024.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with short circuiting. *arXiv preprint arXiv:2406.04313*, 2024.

A Effects of Softmax Temperature in Attack Success

Our 1-hot shortcut learns a very good mapping from VQ-VAE latent representations to the model input vocabulary. This implies that most predictions are very confident (i.e. probability 1 on the correct token and 0 elsewhere). This skewed distribution vanishes gradients for all other tokens and makes optimization harder. We found that increasing the temperature of the softmax helps distribute the probability over other likely tokens and make optimization smoother. Table 6 reports the results obtained with different choices of softmax. We report the numbers with and without the shortcut on the forward pass. We select a softmax temperature of 6 as default for all our experiments.

Table 6 Attack success rate with our 1-hot shortcut while using different temperature on the prediction softmax. Increasing the temperature improves effectiveness and transferability of the attack.

Temperature	With Shortcut	Without Shortcut
1	12.9%	0.0%
2	25.6%	24.4%
4	34.7%	36.0%
6 (default)	**71.3%**	47.5%
8	46.8%	7.8%
10	25.6%	0.0%

B Transferability of Direct Attacks

Table 7 **Direct attacks do not transfer across models.** We evaluate whether images optimized for a single prompt can jailbreak that prompt in different architectures.

Attack	Chameleon-30B	LLaVA-1.6
No attack	0.0%	8.8%
Jailbreak Image (1-hot)	0.0%	10.0%
Jailbreak Image (Embed)	0.0%	10.0%