FigCode Draft v2

```
%%%%%%%%%%%%%%%%%%%%%%%%%方法部分第一段%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%我们的研究目标， 这个研究问题的难点是什么（2-3句话）， 以前工作时怎么解决这个问题的（2-3句话） 他们的主要不足在哪里（如果有）。 针对这个问题我们的方法阐述，详细介绍fig1，这么做的优势在哪里。

%%%%%%%%%%%%%%%%%%%%%%%%%方法部分第二段%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%任务定义 文字+公式（1段）

%%%%%%%%%%%%%%%%%%%%%%%%%方法部分第三段%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 我们的方法 Pictorial Decomposed-word

%%%%%%%%%%%%%%%%%%%%%%%%%方法部分第四段%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 我们的方法 Disguised Code Instruction
```

方法部分第一段

%我们的研究目标， 这个研究问题的难点是什么（2-3句话）， 以前工作时怎么解决这个问题的（2-3句话） 他们的主要不足在哪里（如果有）。 针对这个问题我们的方法阐述，详细展开介绍fig1，我们优势在哪里。

% 根据以前的研究\cite{gong2023figstep}，我们可以发现，视觉模态可能会为大型多模态语言模型（MLLMs）带来更多的脆弱性，因为图像的解析需要处理颜色、形状、纹理、背景等多方面信息，而这些信息之间的关联性和变化性非常高。然而，手动制作图像以绕过MLLMs的检测并不容易。而单独的image在目前的模型中都已经被对齐技巧部分解决。

Drawing upon prior research \cite{gong2023figstep}, it becomes evident that the visual modality can engender additional vulnerabilities in large multimodal language models (MLLMs). This phenomenon arises from the intricate nature of image parsing, which necessitates the processing of diverse elements such as color, shape, texture, and background—each characterized by significant variability and complex interdependencies. Nevertheless, the manual fabrication of images designed to circumvent the detection mechanisms of MLLMs proves to be a formidable challenge. Furthermore, contemporary models have partially mitigated the vulnerabilities of individual images through sophisticated alignment techniques.

% 目前主流的MLLMs攻击方法可以分为文本攻击（text-attack）和图像攻击（image attack）。文本攻击的方法多种多样，包括基于遗传算法的方法

\cite{liu2023autodan}，以及基于梯度的token变换方法\cite{zou2023universal}。图像攻击的研究大多依赖于访问模型的梯度\cite{anonymous2023on}，或者采用某种方式模拟梯度进行对抗攻击\cite{maus2023black, mehrotra2024tree}。

Contemporary strategies for breaching Massive Language and Learning Models (MLLMs) predominantly bifurcate into two domains: textual attack and visual attack. Text attacks encompass a variety of approaches, including genetic algorithm-based methods \cite{liu2023autodan} and gradient-based token transformation methods \cite{zou2023universal}. Most Researches on image attacks predominantly relies on accessing the model's gradients \cite{anonymous2023on} or using techniques to simulate gradients for adversarial attacks \cite{maus2023black, mehrotra2024tree}.

% 尽管这些方法在攻破MLLMs方面取得了一定的成效，但也存在一些问题。首先，大多数方法需要进行白盒攻击，即需要访问到模型的梯度信息。其次，这些方法不够简单通用且复杂，往往需要针对特定模型进行优化，或需要执行多个步骤才能完成攻击。因此，开发一种无需模型梯度信息、能够自动化并通用于各种MLLMs的攻击算法，显得尤为迫切。

Despite the efficacy exhibited by these methodologies in breaching MLLMs, they are not without their shortcomings. Primarily, the majority of these techniques necessitate white-box attacks, implying access to the model's gradient information. Moreover, these approaches lack universality and often entail complexity, requiring optimization tailored to specific models or the execution of multiple steps to accomplish an attack. Thus, the development of an attack algorithm that obviates the necessity for model gradient information, while being automated and universally applicable to diverse MLLMs, emerges as an imperative endeavor.

% 基于此，我们提出了PiCo算法，从\ref{fig:intro_image} 中可以看到，简单的文本攻击以及图片+文本的攻击已经被目前的大模型通过instruction tunning或SFT解决的较好。而我们的模型能够直接跳过MLLMs的输入端和输出端的filter，输出有害内容。

我们的研究目标是探索大型多模态模型在代码场景中的对齐效果，特别是当有害文本被转换为图像时，这些模型的脆弱性。

方法部分第二段

任务定义 文字+公式（1段）

Jailbreak Attack: Jailbreak攻击是一种针对大型语言模型（LLM）的攻击方法，其目的是破坏模型预设的对齐值和其他约束，迫使模型对恶意问题作出回答。黑客设计一组恶意问题，并通过加入特定的提示，诱导模型生成符合攻击者预期的回答，而不是拒绝回答。

考虑一组恶意问题Q = {Q1, Q2, ..., Qn}，黑客将这些问题与jailbreak提示P结合，形成组合输入集M = {Mi = <P, Qi>}i=1,2,...,n。当将输入集M呈现给目标MLLM时，该模型生成一组响应R = {R1, R2, ..., Rn}。Jailbreak攻击成功会导致R中的响应主要是与恶意问题Q紧密相关的答案，而不是与人类价值观对齐的拒绝消息。

Research 目标：本研究的主要目的是探索和解决大型多模态模型中的对齐漏洞，特别是在将有害文本转换为图像的情况下。此探索旨在了解这些模型在处理与代码相关的上下文时的对齐有效性，并制定可靠的策略来规避这些漏洞。

算法应当：

1. 确保其独立于模型梯度运行，并可普遍应用于不同的 MLLM。
2. 算法自动化，最大限度地减少对特定模型的手动干预和优化需求

Algorithm Development:

算法开发：

- Develop the PiCo algorithm, ensuring it operates independently of model gradients and can be applied universally across different MLLMs.开发 PiCo 算法，确保其独立于模型梯度运行，并可普遍应用于不同的 MLLM。
- Ensure the algorithm is automated, minimizing the need for manual intervention and optimization for specific models.确保算法自动化，最大限度地减少对特定模型的手动干预和优化需求。

我们的任务是对大型多模态模型（MLLMs）进行攻击，目的是探索模型在处理具有代码场景的图像时的脆弱性。具体来说，我们将有害文本转换为图像形式的代码，并嵌入到图像中，然后让MLLMs对这些图像进行解析。我们的目标是使模型产生预期外的输出，即输出原本被模型检测机制过滤掉的有害内容。换句话说，我们的任务是找出模型处理图像形式的代码时的漏洞，并利用这些漏洞绕过模型的检测机制。

我们的任务是设计和实现一个强大的多模态学习模型，该模型能够有效地在代码场景中对齐文本和图像。这个任务涉及处理和理解两种不同的数据类型：一种是文本数据，包括有害的和无害的文本；另一种是图像数据，包括代码和非代码图像。我们的目标是确保模型能够有效地识别和处理这些不同类型的数据，以提高其在面对复杂和多变的数据时的稳定性和鲁棒性。

方法部分第三段
我们的方法 Pictorial Decomposed-word

方法部分第四段
我们的方法 Disguised Code Instruction

注意：

1） 我们未进入模型内部，尽量少提token-level。我们只处于prompt-level
2） 每一句都必须有用且客观严谨，不能简单翻译中文，堆字数。 不能有废话，无意义的话

根据以前的研究，我们可以发现，视觉模态可能会为MLLMs带来更多的脆弱性，因为图像的解析需要处理颜色、形状、纹理、背景等多方面信息，而这些信息之间的关联性和变化性非常高。However， it is not easy to manually craft the image to jailbreak the MLLMs.

我们的研究发现，在大型多模态模型中，代码场景的存在显著降低了模型的对齐效果，使其更易受到攻击。我们通过将有害文本转换为图像来增加其毒性，成功地破坏了代码场景中的模型对齐效果。具体而言，我们首先配置大型视觉模型（LVM）通过文本提示生成代码，然后将有害文本转换为代码形式，并将此代码嵌入图像中。由于模型在处理图像时的对齐效果显著降低，这一方法使我们能够绕过模型的正常检测机制，成功进行攻击。

基于此，我们考虑尝试将视觉模态和代码场景结合起来。代码调试和解释中的场景下，利用图像和代码的不同特性来增强模型的理解能力。可以更加让模型更加曲折但直观的理解代码的功能和行为以达成我们想要的目的。此外，视觉数据和代码数据可以相互补充，验证模型的鲁棒性。

Both Gemini Pro and GPT-4o models have shown significant improvements when switching from the Text-only setting to the PtCo setting. For instance, Gemini Pro's toxicity rating in the Animal category increased from 1.22 in the Text-only setting to 3.09 in the PtCo setting, but its helpfulness rating improved more substantially, from 1.84 to 3.12. This resulted in an overall higher THS (0.63) in the PtCo setting compared to the Text-only setting (0.36), indicating better overall performance.

Similarly, GPT-4o showed lower toxicity ratings across all categories in the Text-only setting, with an Animal category rating of 1.02, but had a higher helpfulness rating of 2.72. However, in the PtCo setting, while the toxicity rating increased, the helpfulness rating showed a more significant increase, such as in the Financial category where the toxicity increased to 4.50 but the helpfulness increased to 1.71. Overall, GPT-4o's THS in the PtCo setting (3.25) was significantly higher than in the Text-only setting (1.52), manifesting the marked advantage of the PtCo setting.

Gemini Pro 和 GPT-4o 模型在从纯文本设置切换到 PtCo 设置时都显示出了显著的改进。例如，Gemini Pro 的 Animal类别的毒性为1.22，帮助性为1.84。而在PtCo设置下，毒性评分上升到了3.09，帮助性上升到了3.12。This resulted in an overall higher THS (0.63) in the PtCo setting compared to the Text-only setting (0.36), indicating better overall performance. Similarly, GPT-4o showed lower toxicity ratings across all categories in the Text-only setting, with an Animal category rating of 1.02 in toxicity. 在 PtCo 设置中的GPT4o，毒性评级增加了，有用性评级也显示出更显著的增加

GPT 4-turbo模型在Text-only设置下，各类别的毒性评分较低，但帮助性评分同样不高。例如Animal类别的毒性为1.06，帮助性为2.32。而在PtCo设置下，毒性评分和帮助性评分均有所上升，如Animal类别的毒性为2.44，帮助性为2.89。总体来看，GPT 4-turbo在PtCo设置下的THS（3.01）高于Text-only设置（1.39），同样显示出PtCo设置的优势。

从整体结果来看，PtCo 设置下所有模型的 THS 均显著高于 Text-only 设置，说明 PtCo 设置处理敏感内容时能够更有效地平衡毒性与帮助性。此外，GPT-4o 模型在 PtCo 设置下表现最佳，THS 达到 3.25，展现出较强的综合处理敏感内容的能力。

GPT 4-turbo模型在Text-only设置下，各类别的毒性评分较低，但帮助性评分同样不高。例如Animal类别的毒性为1.06，帮助性为2.32。而在PtCo设置下，毒性评分和帮助性评分均有所上升，如Animal类别的毒性为2.44，帮助性为2.89。总体来看，GPT 4-turbo在PtCo设置下的THS（3.01）高于Text-only设置（1.39），同样显示出PtCo设置的优势。

从整体结果来看，所有模型在PtCo设置下的THS均显著高于Text-only设置，这表明在处理敏感内容时，PtCo设置能够更有效地平衡毒性和帮助性。此外，GPT-4o模型在PtCo设置下的表现最佳，其THS达到了3.25，显示出在处理敏感内容时具有较强的综合能力。