

# Prompt to picture using stable diffusion and LLM

## Abstract

This project explores a novel approach to text-to-image generation by leveraging Large Language Models (LLMs) and Stable Diffusion to automatically determine and apply semantic blending between two textual prompts. Unlike traditional prompt-to-image systems that rely on user-defined weights for controlling the influence of different concepts, this project automates the weighting process by utilizing an LLM to interpret and emphasize the most relevant subjects within a prompt. The core methodology involves two distinct prompts and blending them using a series of weighted interpolations controlled by an alpha parameter (ranging from 0 to 1). For each alpha value, a blended prompt embedding is computed using CLIP (Contrastive Language–Image Pretraining) embeddings. These embeddings are then passed through a Stable Diffusion pipeline to generate corresponding images. To identify the most semantically aligned result, the generated images are compared using cosine similarity with the original prompts' embeddings. The image with the highest similarity score is selected as the best representation of the blended concept.

## Contents

### 1. Introduction

- 1.1 Background
- 1.2 Objectives
- 1.3 Scope
- 1.4 Tools & Libraries

### 2. System Overview

- 2.1 High-Level Architecture
- 2.2 Workflow Description

### **3. Prompt Processing and Weight Assignment**

- 3.1 Input Prompt Parsing
- 3.2 LLM Use for Subject Weighting
- 3.3 Examples

### **4. Prompt Embedding & Blending**

- 4.1 Encoding with CLIP
- 4.2 Alpha Interpolation
- 4.3 Generation of Interpolated Prompts
- 4.4 Prompt Sampling Strategy

### **5. Image Generation using Stable Diffusion**

- 5.1 Model Details
- 5.2 Inference Settings
- 5.3 Batch Generation
- 5.4 Challenges & Observations

### **6. Evaluation & Best Image Selection**

- 6.1 Feature Extraction via CLIP
- 6.2 Similarity Computation
- 6.3 Best Match Selection
- 6.4 Example Comparisons

## **1. Introduction**

### **1.1 Background**

Prompt-based image generation has gained prominence with diffusion models like Stable Diffusion, enabling the synthesis of high-quality images from natural language descriptions. Traditionally, users manually craft

prompts and iterate to get the desired output. However, this becomes inefficient when dealing with composite ideas or abstract themes.

## 1.2 Objectives

- Use LLM to combine two prompts using different values of alpha.
- Use a Large Language Model (LLM) to assign weights to different subjects in prompts.
- Create an embedding from the blended prompt.
- Synthesize images using Stable Diffusion from the embedding.
- Automatically select the best image using cosine similarity with original prompt embeddings.

## 1.4 Scope

This project focuses on automatic prompt blending and image generation using LLM model without allowing user-defined weights.

## 1.5 Tools & Libraries

Hugging Face Transformers (for LLM and CLIP)

OpenAI CLIP model

Stable Diffusion (Hugging Face)

NumPy

# 2. System Overview

## 2.1 High-Level Architecture

- **Input:** A user provides any two prompts (Prompt A and Prompt B) for which he wants to generate an image that includes the concepts of both Prompt A and Prompt B
- **LLM:** It first creates a blended prompt (using Prompt A and Prompt B) using different values of alpha ranging from 0.4 to 0.6 which means each prompt will share a different level of significance in the final blended prompt. Also it assigns different weights to the main subjects, main objects, secondary subjects and features according to their respective importance.

- **CLIP Encoder:** Generates embeddings for the final blended prompt which is the weighted prompt(all the weights are applied to the subjects,objects,etc)
- **Stable Diffusion:** Synthesizes images from the final prompts which is a dictionary of 3 prompts - each prompt from a different value of alpha
- **CLIP Similarity:** Selects a best image on the basis of cosine similarity from a set of 9 images (3 images from each prompt using different seeds)

## 2.2 Workflow Description

- Use LLM to blend the provided prompts using alpha
- Use LLM to weigh the importance of each subject.
- Generate CLIP embeddings of each prompt[alpha].
- Generate images using Stable Diffusion using different seeds.
- Re-encode generated images with CLIP.
- Compare image embeddings with original prompts.
- Select the image with the highest cosine similarity.

## 3. Prompt Processing and Weight Assignment

### 3.1 Input Prompt Parsing

Prompts are decomposed into noun phrases, adjectives, and modifiers using syntactic parsing.

### 3.2 LLM use for subject weighting

The LLM creates a final prompt by combining the provided prompts using alpha and weighs different elements in the final prompt

### 3.3 Examples

Prompt A: "A man watching TV"

Prompt B: "A star in galaxy"

LLM combines them using alpha[0.4 to 0.6] in the following way:

**Prompt\_0.4:** A man(1.7) sitting on a couch(1.2), watching a documentary about stars on TV(1.7), with a massive galaxy(1.7) star projected on the screen.

**Prompt\_0.5:** A man(1.7) in a modern living room, sitting on a couch(1.2), gazes at a TV(1.7) showing a documentary about a stunning star in a galaxy(1.7).

**Prompt\_0.6:** 'An astronaut in a spaceship, sitting on a couch(1.2), watching a TV(1.7) broadcasting a live galaxy(1.7) view, with a massive star(1.2) glowing outside the window.

## 4. Prompt Embedding

Each prompt[alpha] is embedded using the CLIP text encoder and then corresponding images are generated

## 5. Image Generation using Stable Diffusion

### 5.1 Model Details

Stable Diffusion v1.5 with pre-trained weights.

### 5.2 Inference Settings

- Resolution: 512x512
- Guidance scale: 7.5
- Steps: 50
- Num\_variants=5
- Random seed fixed for reproducibility

## 5.3 Batch Generation

Embedding with different seeds produces one image.

## 5.4 Observations

Intermediate alphas often yield more balanced images combining features from both prompts.

# 6. Evaluation & Image Selection

## 6.1 Re-Embedding Images with CLIP

Generated images are encoded using CLIP's image encoder.

## 6.2 Cosine Similarity

First, the similarity between embedding of the image and both prompts is calculated. Then similarity between the blended prompt and embedding of the image is calculated and the final score is calculated using

```
final_score = lambda_ * sim_blend + (1 - lambda_) * min(sim_A, sim_B)
```

## 6.3 Selection Criteria

The image with the highest cosine similarity is selected as the final output.

## 6.4 Examples

Alpha = 0.4 may yield an image with a robot wearing medieval armor in a sandy forest, selected as best.

# 7. Results

Here are some of the results of the prompt to image model

# 1. A skeleton, a man eating food

### Prompt Blending Image Generator

Enter two prompts. The system will blend them, generate multiple images, and return the best one.

Prompt1

a skeleton

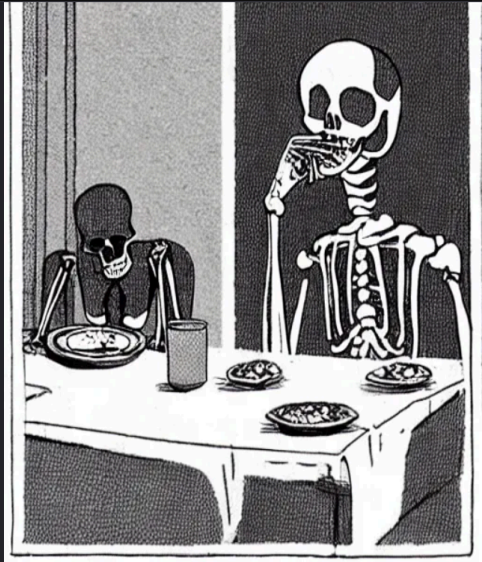
Prompt2

a man eating food

Clear

Submit

output



Flag

## 2. An eagle flying , tall buildings

### Prompt Blending Image Generator

Enter two prompts. The system will blend them, generate multiple images, and return the best one.

Prompt 1

an eagle flying


Prompt 2

tall buildings

Clear

Submit

output



Flag



3. A vintage car parked on a quiet street in Paris, 1960s aesthetic, A serene lake surrounded by autumn trees, high-resolution landscape.

Prompt Blending Image Generator

Enter two prompts. The system will blend them, generate multiple images, and return the best one.

Prompt 1

A vintage car parked on a quiet street in Paris, 1960s aesthetic

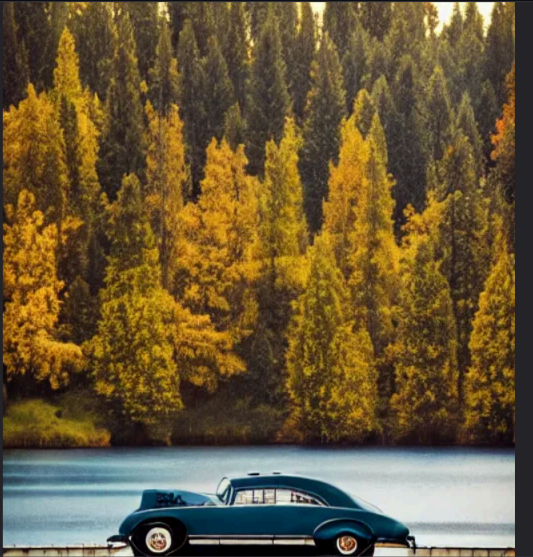
Prompt 2

A serene lake surrounded by autumn trees, high-resolution landscape

Clear

Submit

output



Flag