HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION COMMUNICATION TECHNOLOGY

PROJECT 3

BOVAGAU MEME GENERATOR

Advanced Personalization of Generative Models using Pivotal Tuning

Instructor: Prof. Tran Viet Trung
Students: Le Gia Huy - 20225498
Trinh Hoang Anh - 20225470

January 15, 2026

Contents

Abstract

This report presents *BovaGau Meme Gen*, a domain-specific generative AI system for automating branded meme creation featuring the proprietary mascots "Bo" (Bull) and "Gau" (Bear). The system addresses the challenge of adapting generic meme templates to brand-specific characters while preserving strict identity fidelity under conditions of limited training data.

To mitigate identity degradation caused by sparse datasets, we adopt a **Pivotal Tuning** strategy on Stable Diffusion v1.5 that combines **Textual Inversion** for encoding character identity with **Low-Rank Adaptation (LoRA)** for controlled stylistic adaptation. To ensure faithful transfer of meme layout and context, the pipeline further integrates **ControlNet**, which conditions generation on structural cues extracted from reference meme images. This design decouples character preservation from contextual guidance, enabling reliable identity retention while mimicking diverse meme formats.

The report also describes a custom data collecting and preprocessing pipeline, along with CLIP-based evaluation metrics for assessing identity consistency and prompt alignment. Together, these components form an end-to-end system for scalable, high-fidelity branded meme generation.

1 Introduction

1.1 Project Context

In the contemporary digital marketing landscape, visual memes have emerged as a dominant medium for online engagement due to their high shareability, rapid production cycles, and strong cultural relevance. Brands increasingly rely on meme-based content to communicate informally with audiences, reinforce brand personality, and maintain visibility across social media platforms.

Despite their popularity, adapting generic meme templates to feature proprietary brand mascots remains a labor-intensive and repetitive task. This process typically requires skilled designers to manually redraw or edit characters to fit each new meme context. For the "BovaGau" brand, whose identity is closely tied to the consistent visual representation of its mascots "Bo" and "Gau," even minor deviations in appearance—such as horn shape, facial proportions, or color tone—can weaken brand recognition and visual coherence. As a result, full automation using generic image generation models is not viable without specialized adaptation.

1.2 Objectives

The primary objective of this project is to develop an automated generative pipeline that accepts a reference meme image and a textual prompt, and produces a high-fidelity branded meme image. The system is designed to satisfy the following objectives:

- **Preserve Character Identity:** Through Textual Inversion and LoRA-based fine-tuning, the generated characters must remain unmistakably identifiable as "Bo" or "Gau," consistently retaining defining traits such as horn shape, fur color, facial proportions, and overall silhouette.

- **Transfer Meme Context and Structure:** Using ControlNet, the model must accurately capture and reproduce the structural layout, pose, and compositional context of the input meme template, enabling faithful mimicry of popular meme formats.

- **Enable Contextual Flexibility:** The system should support diverse meme scenarios by combining textual prompts with ControlNet conditioning, allowing the mascots to be placed into new situations while respecting both semantic intent and visual structure.

By achieving these objectives, the project seeks to demonstrate that domain-specific fine-tuning of diffusion-based generative models can serve as a practical and scalable solution for branded content generation under real-world data constraints.

2 Data Engineering Pipeline

High-quality, well-curated data is a critical prerequisite for successful fine-tuning of diffusion-based generative models, particularly under conditions of extreme data scarcity. Given the limited availability of proprietary visual assets for the *BovaGau* mascots, we designed a custom data engineering pipeline that prioritizes identity fidelity, annotation consistency, and contextual diversity over raw dataset size.

2.1 Data Collection & Composition

The final training dataset was constructed from two complementary sources: high-fidelity internal design assets provided by the client and a curated set of legacy illustrations collected via automated web scraping. This hybrid strategy balances anatomical accuracy with contextual variability.

2.1.1 Internal Design Assets (Provided by Client)

Client-provided design materials serve as the authoritative visual reference for the mascots and function as ground-truth identity anchors during training. These assets exhibit clean line work, consistent proportions, and canonical color palettes, making them particularly valuable for learning stable character representations.

The internal dataset consists of:

- **Expression Sheets:** 5 images each for "Bo" and "Gau," depicting a range of facial expressions and emotional states.

- **Pose References:** 11 images each for "Bo" and "Gau," featuring full-body illustrations across diverse stances and viewpoints.

- **Interaction Data:** A single high-quality image containing both characters in close proximity, used to reinforce inter-character scale and relational consistency.

2.1.2 Web Scraping via Custom Scrape UI

To increase pose, composition, and contextual diversity, we developed a custom scraping tool to collect legacy *BovaGau* illustrations from the official brand website (`https://bovagau.vn`). Unlike generic web scraping, this tool was specifically designed to support downstream character-level extraction required for Textual Inversion training.

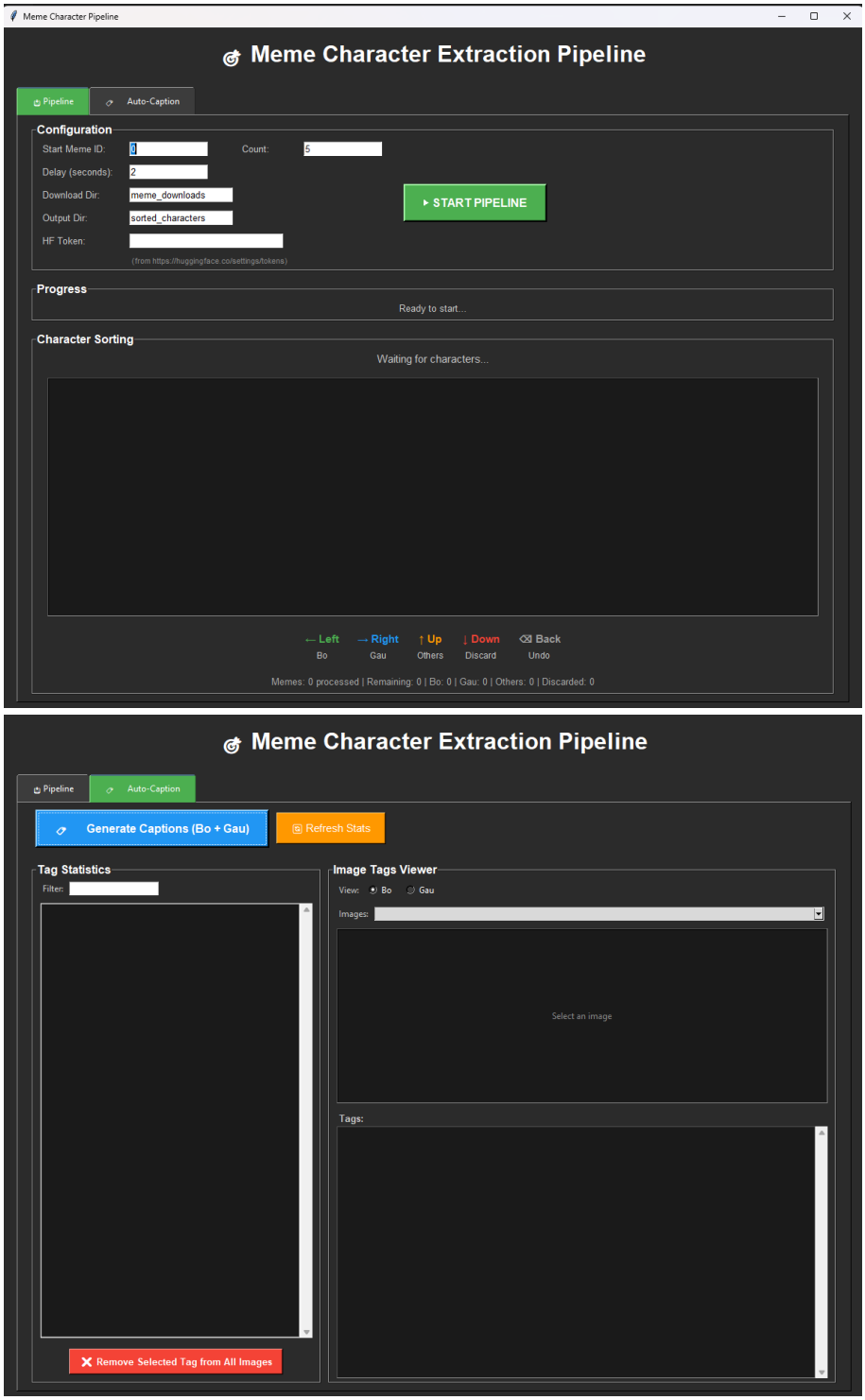

Figure 1: Screenshot of the custom scraping, segmentation and captioning UI.

Using Selenium, the scraper programmatically navigates URLs of the form:

`https://bovagau.vn/meme/{meme_id}`

where `meme_id` is iterated over a user-defined range. For each page, the scraper searches for a download button identified by the CSS class `‘‘Download Meme’’`. Pages without a valid download element are skipped automatically. Valid images are downloaded and stored in a predefined directory for further processing.s

Since many scraped illustrations contain multiple characters within a single image, direct usage is unsuitable for character-specific fine-tuning. To address this, we employ Meta's **Segment Anything Model 3 (SAM 3)** [1] to perform automated character segmentation. Released in late 2025, SAM 3 introduces *Promptable Concept Segmentation (PCS)*, allowing natural language prompts to trigger exhaustive instance-level segmentation. By prompting with a generic concept such as "character," the model is able to detect and isolate all character instances in a single zero-shot pass, without requiring manual point clicks or bounding boxes.

Each segmented instance is cropped using its bounding box and displayed within an interactive sorting interface. To separate images of "Bo" and "Gau," we designed a lightweight real-time UI that allows the operator to classify segments using keyboard controls (left, right, up, down). This human-in-the-loop step ensures high precision in character labeling while remaining significantly more efficient than manual cropping.

Following filtering and quality control, the retained web-scraped data consists of:

- **Bo (Bull):** 92 clean, single-character images.
- **Gau (Bear):** 39 clean, single-character images.

2.1.3 Final Training Set Statistics

Table 1 summarizes the composition of the final training dataset after all filtering and preprocessing steps.

Category	Bo (Bull)	Gau (Bear)	Description
Expressions (Internal)	5	5	Facial details and emotions
Poses (Internal)	11	11	Full-body structural references
Scraped Data (Web)	92	39	Diverse contexts and viewpoints
Total Images	**108**	**55**	*Exclusive of joint image*

Table 1: Detailed breakdown of the final training dataset composition.

2.2 Preprocessing Workflow

Raw collected images often contain artifacts—such as watermarks, overlaid meme text, and inconsistent resolutions—that can negatively impact diffusion model training. To mitigate these issues, we designed a structured preprocessing workflow aimed at standardization and semantic clarity.

2.2.1 Restoration and Cleaning

- **Super-Resolution:** All images were upscaled and normalized to a resolution of 512×512 pixels using AI-based super-resolution models to ensure consistency with Stable Diffusion training requirements.
- **Artifact Removal:** Original meme text, dialogue bubbles, and watermarks were removed using content-aware object removal tools (e.g., Photoshop Content-Aware Fill) to prevent the model from learning spurious visual correlations.

2.2.2 Semantic Tagging and Captioning

Automatic caption generation was performed using the `wd14-vit-v2` image tagger [2] to provide descriptive conditioning signals during training. Rather than directly using all generated tags, we applied a targeted **negative filtering** strategy.

- *Filtering Strategy:* Explicit anatomical tags such as "horns," "animal ears," or "fur" were manually removed from captions.
- *Rationale:* By excluding these descriptors, we encourage the model to associate defining anatomical features directly with the learned identity tokens (e.g., `<Bo>`, `<Gau>`), rather than treating them as generic attributes. As a result, invoking a character token implicitly activates its defining features without requiring explicit prompting.

3 Theoretical Framework

This section explores the mathematical foundations of the selected models and evaluation metrics.

3.1 Latent Diffusion Models (LDMs)

Stable Diffusion is built upon the architecture of Latent Diffusion Models (LDMs), which were designed to overcome the high computational costs of traditional pixel-space diffusion models [3]. By performing the diffusion process in a compressed latent space rather than the high-dimensional pixel space, LDMs achieve a significant reduction in training and inference requirements while maintaining high visual fidelity.

1. **Perceptual Compression (VAE):** The first stage involves a Variational Autoencoder (VAE) that learns to compress the image $x \in \mathbb{R}^{H \times W \times 3}$ into a lower-dimensional latent representation $z = \mathcal{E}(x)$, where $z \in \mathbb{R}^{h \times w \times c}$. This compression removes high-frequency details that are imperceptible to the human eye, allowing the generative model to focus on the semantic composition of the image. After the diffusion process is complete, a decoder $x \approx \mathcal{D}(z)$ maps the latent back into the pixel space.
2. **Latent Diffusion Process:** Unlike standard diffusion models that operate on pixels, LDMs apply the forward diffusion process to the latent z. Noise is iteratively added according to a variance schedule until the signal is transformed into near-white Gaussian noise z_T. This transition allows the model to learn the underlying distribution of the latent manifold rather than the raw pixel grid.

3. **Denoising U-Net with Conditioning:** The core of the LDM is a time-conditional U-Net $\epsilon_\theta(z_t, t, \tau_\theta(y))$. At each timestep t, the U-Net is trained to predict the noise component of the noisy latent z_t. To enable text-to-image generation, the model utilizes a **Conditioning Mechanism** (typically a CLIP text encoder). The encoded prompt $\tau_\theta(y)$ is injected into the U-Net's intermediate layers via **cross-attention** mechanisms, guiding the denoising process to align the resulting image with the textual description y.

By decoupling the image generation task into a compression stage (VAE) and a generative stage (U-Net), LDMs strike an efficient balance between computational efficiency and the ability to model complex, high-resolution data.

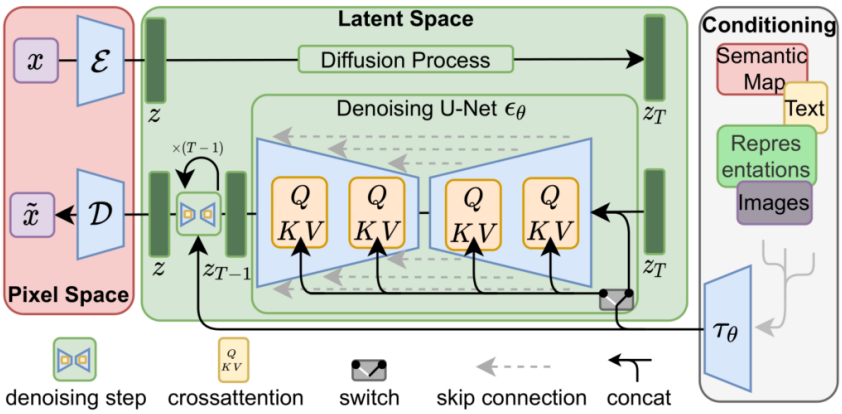

Figure 2: Latent Diffusion Architecture.

3.2 Fine-tuning Strategies

3.2.1 Textual Inversion (TI)

Textual Inversion[4] does not alter the model's weights. Instead, it optimizes a new embedding vector v_* in the text encoder's embedding space. The optimization objective is:

$$v_* = \arg\min_v \mathbb{E}_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y, v))\|_2^2\right] \tag{1}$$

where y is a template prompt like "A photo of S_*". This allows the model to reconstruct the concept using a new "word" without catastrophic forgetting.

3.2.2 Low-Rank Adaptation (LoRA)

LoRA hypothesizes that the change in weights during fine-tuning has a low intrinsic rank. Instead of updating the full weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA injects trainable rank decomposition matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ where $r \ll \min(d, k)$.

$$W' = W + \Delta W = W + BA \tag{2}$$

This significantly reduces the number of trainable parameters while allowing the model to learn complex stylistic shifts that TI cannot capture.

3.3 ControlNet

ControlNet [5] is a neural network architecture designed to add spatial conditioning controls to large, pre-trained text-to-image diffusion models. Unlike fine-tuning methods like LoRA or Textual Inversion, which primarily adapt style or concepts, ControlNet focuses on structural guidance (e.g., edges, depth maps, segmentation masks, or human pose skeletons).

3.3.1 Architecture

The core innovation of ControlNet is its ability to reuse the robust feature extraction of a pre-trained model while avoiding "catastrophic forgetting." It achieves this through a dual-stream architecture:

1. **Locked Copy:** A copy of the original diffusion model's encoder blocks is kept frozen to preserve the base model's knowledge.
2. **Trainable Copy:** A duplicate of the encoder blocks is created and trained with the specific conditioning input (e.g., Canny edge map or OpenPose skeleton).

These two streams are connected via **Zero Convolutions**—convolutional layers initialized with both weights and biases set to zero.

$$y = \mathcal{F}(x; \Theta) + \mathcal{Z}(\mathcal{F}(x + c; \Theta_c); \Theta_z) \tag{3}$$

where $\mathcal{F}$ represents the neural network block, Θ are the locked weights, Θ_c are the trainable copy weights, $\mathcal{Z}$ denotes the zero convolution, and c is the extra conditioning vector.

Because the zero convolutions are initialized to zero, the ControlNet initially behaves exactly like the original model. As training progresses, the zero convolutions learn to inject the control signal into the deep features of the UNet, allowing for precise structural adherence without degrading the generated image quality.

3.4 Evaluation Metric: CLIP Score

To quantitatively evaluate prompt adherence, we utilize the **CLIP (Contrastive Language-Image Pre-training)** Score.

CLIP consists of two encoders: an Image Encoder (E_I) and a Text Encoder (E_T). It is trained on 400 million pairs to maximize the cosine similarity between the embeddings of matched image-text pairs.

$$\text{Score}(I, T) = \cos(E_I(I), E_T(T)) = \frac{E_I(I) \cdot E_T(T)}{\|E_I(I)\| \|E_T(T)\|} \tag{4}$$

A higher CLIP score indicates that the generated image I is semantically closer to the input prompt T.

4 System Implementation

We implemented a **Pivotal Tuning** architecture, combining the strengths of TI and LoRA.

4.1 Phase 1: Textual Inversion (Concept Anchoring)

The goal was to establish a semantic anchor.

- **Initializer:** "cow" (for Bo) and "bear" (for Gau). This provides a strong structural prior.
- **Vectors:** 3 vectors per token. This was found to be the optimal balance between expressiveness and overfitting.

4.2 Phase 2: LoRA Training (Stylistic Refinement)

We merged the TI embedding into the base model and trained LoRA adapters on the U-Net to learn the specific artistic style.

Custom Data Loader: A specialized 'LoRADataset' class was implemented to handle nested directories and merge text captions dynamically.

```
class LoRADataset(Dataset):
    def __init__(self, data_root, tokenizer, placeholder_token, ...):
        # Recursive directory walking
        for root, dirs, files in os.walk(data_root):
            for file in files:
                if file.lower().endswith(('.png', '.jpg')):
                    self.image_paths.append(os.path.join(root, file))

    def __getitem__(self, index):
        # Dynamic Prompt Construction
        if os.path.exists(txt_path):
            with open(txt_path, "r") as f:
                caption = f.read().strip()
            prompt = f"a photo of {self.placeholder_token}, {caption}"
        else:
            prompt = f"a photo of {self.placeholder_token}"
        return example
```

Listing 1: Custom LoRADataset Class Implementation

4.3 Phase 3: Optional ControlNet Integration (Inference Only)

To address the requirement of context preservation in meme generation, we integrated **ControlNet** as a modular component in the inference pipeline. While LoRA handles the character identity and artistic style, ControlNet allows the system to lock the spatial structure of the input meme.

- **Preprocessor Mechanism:** We specifically utilize **DWPose** (a more robust alternative to OpenPose) [6] to extract precise skeletal structures from reference memes.
- **Model Usage:** We employ pre-trained ControlNet models directly for inference (e.g., `control_v11p_sd15_openpose`). **No fine-tuning** of the ControlNet layers was performed; the system leverages the robust generalization capabilities of existing models to transfer poses to our custom characters.
- **Application:** This module is optional and is activated specifically when the user provides a reference image, ensuring that the generated "Bo" or "Gau" mimics the

exact posture of the original meme subject without requiring additional training costs.

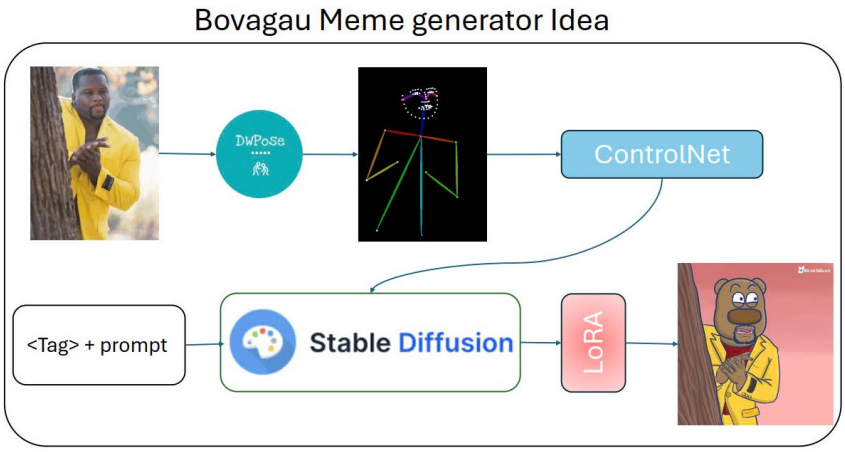

Figure 3: Controlnet Pipeline Diagram with input and expected output

4.4 Training Configuration

The following hyperparameters were determined to be optimal for our specific dataset constraints after extensive experimentation.

Hyperparameter	**Textual Inversion (Phase 1)**	**LoRA Fine-tuning (Phase 2)**
Base Model	Stable Diffusion v1.5	SD v1.5 (with TI embedding)
Resolution	512×512	512×512
Batch Size	2	1
Gradient Accumulation	4	4
Effective Batch Size	8	4
Learning Rate	1.0×10^{-4}	1.0×10^{-4}
LR Scheduler	Constant	Constant
Optimizer	AdamW	AdamW
Steps	3000	1500
Rank / Vectors	3 vectors	Rank 32
Mixed Precision	fp16	fp16

Table 2: Optimal Hyperparameters for Pivotal Tuning

4.5 Experimental Constraints & Alternative Approaches

Prior to finalizing the architecture described above, several alternative methodologies were evaluated but eventually rejected due to performance or resource limitations.

4.5.1 IP Adapter Pipeline

Initial experiments explored an identity-preserving (IP) Adapter pipeline [7] combined with ControlNet to transfer character features directly from reference images without fine-tuning. While this approach offered the advantage of a zero-shot workflow, it failed to

maintain the strict identity fidelity required for the brand. The generated characters frequently exhibited generic traits, losing critical brand-specific details—such as exact horn shapes and facial proportions—that distinguish "Bo" from "Gau."

4.5.2 SDXL Fine-tuning (AutismMix)

We also attempted to leverage **AutismMix SDXL** [8], a fine-tuned version of Stable Diffusion XL specifically optimized for cartoon and anime aesthetics. Despite its superior native resolution and stylistic alignment, this model proved computationally prohibitive given our reliance on the Kaggle platform for training. A single training epoch for SDXL required over 12 hours to complete, exceeding the session runtime limits and making the iterative hyperparameter tuning necessary for high-quality adaptation impossible. Consequently, we decide to stick with the lighter Stable Diffusion v1.5 architecture, which enabled rapid iteration and successful convergence within the available computational budget.

5 Results & Discussion

5.1 Qualitative Results

The TI + LoRa model demonstrates a strong ability to reproduce the mascots in diverse contexts.

(a) Final Result (High Fidelity) (b) Early Stage (Overfitting/Burnt)

Figure 4: Visual comparison of training outcomes.

Observations:

- The distinct horn shape of Bo is preserved across different angles (Figure 4a).
- Early iterations (Figure 5b) showed signs of overfitting (high contrast/burn-in), which were resolved by reducing the Learning Rate.

5.2 Style Transfer Capabilities

We tested the full pipeline ability to adapt the character to different meme formats.

(a) Input Image (b) Output Image

Figure 5: Examples of Meme Style Transfer Capabilities

As shown in the two images above, the character's pose from the original meme has been successfully transferred to our "bo" character. The overall features of "bo" remain clearly recognizable. However, there are some issues with the hands—specifically the rubbing-together gesture in the original meme—which is a known challenge, particularly for SD-based models. In addition, the beard is not fully transferred and appears only as a faint outline.

5.3 Quantitative Evaluation (CLIP Score)

We monitored the CLIP Score throughout the training process. The score stabilized around 29.5, indicating strong semantic alignment.

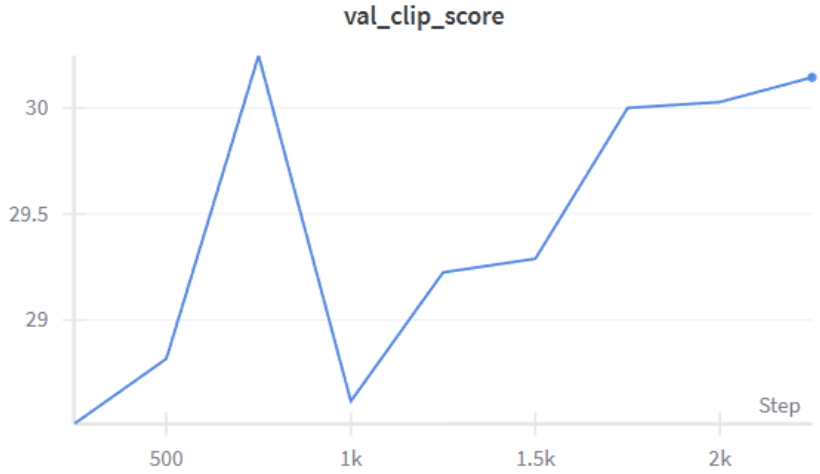

Figure 6: CLIP Score Chart

6 Future works

To further enhance the capability and usability of the BovaGau Meme Gen system, we propose the following development roadmap:

- **Improve pipeline:** The current pipeline with Controlnet successfully capture the character's pose from the source meme and generate our character in that pose. But to fully embed the character into the meme, we still need to transfer the scene — including lighting, perspective, and background elements — and render those elements in our art style so the character appears naturally integrated into the environment.

- **Model Scalability (SDXL / Flux.1):** If more computational resources allow, we will aim to transition the diffusion backbone to larger architectures like **Stable Diffusion XL (SDXL)** or **Flux.1**. These models offer significantly improved text understanding and native resolution, which would reduce the need for aggressive upscaling in the post-processing phase.

- **Automated Content Pipeline:** We plan to integrate Large Language Models (LLMs) such as GPT-4 or Llama 3 into the workflow. The LLM would act as a creative agent, automatically analyzing trending topics and generating humorous captions and prompts that are then fed into the BovaGau image generator, creating a fully autonomous "Meme Factory."

7 Conclusion

The project successfully delivered a specialized meme generation model. By implementing a custom data pipeline and a hybrid **Pivotal Tuning** strategy, we overcame the limitations of small datasets. The theoretical application of CLIP scoring provided a reliable metric for automated evaluation, ensuring the final model met both identity and semantic requirements.

References

[1] N. Carion et al., *Sam 3: Segment anything with concepts*, 2025. arXiv: 2511.16719 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2511.16719.

[2] SmilingWolf, *Wd 1.4 vit tagger v2*, https://huggingface.co/SmilingWolf/wd-v1-4-vit-tagger-v2, 2022.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, 2022. arXiv: 2112.10752 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2112.10752.

[4] R. Gal et al., *An image is worth one word: Personalizing text-to-image generation using textual inversion*, 2022. arXiv: 2208.01618 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2208.01618.

[5] L. Zhang, A. Rao, and M. Agrawala, *Adding conditional control to text-to-image diffusion models*, 2023. arXiv: 2302.05543 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2302.05543.

[6] Z. Yang, A. Zeng, C. Yuan, and Y. Li, *Effective whole-body pose estimation with two-stages distillation*, 2023. arXiv: 2307.15880 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2307.15880.

[7] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models," 2023. arXiv: 2308.06721 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2308.06721.

[8] Autismix_anon, *Autismmix SDXL*, version AutismMix_pony, Feb. 1, 2024. Accessed: Jan. 12, 2026. [Online]. Available: https://civitai.com/models/288584/autismmix-sdxl.