Giraffe: A Mapping Architecture from Hidden Text Representations to Visual Embeddings for Efficient Graphic Design

Nejla Ghaboosi
Canva Research
nejla@canva.com

Abstract

Multimodal large language models (MLLMs) have made significant progress in understanding and interpreting multimedia content. However, their ability to generate media remains limited. Recent approaches have attempted to bridge this gap by translating the hidden representations of token sequences into the embedding space of visual models or directly into raw image data. However, these methods often represent each image using multiple specialised tokens which significantly increases the input length. This becomes a major limitation for tasks such as graphic design generation where the output typically involves a seamless blend of thousands of tokens across text, multiple images, and layout information. To address this challenge, a novel architecture is proposed that maps hidden token representations to the embedding space of visual models, such as CLIP ViT-L/14, using a single `[IMG]` token per image. The architecture employs two shallow MLP blocks, each with a separate compression module followed by a shared expansion module, trained with six distinct loss functions. One block aids the other during training and is omitted during inference, resulting in a lightweight solution. Strong performance is demonstrated in both image-to-design and text-to-design generation tasks.

1. Introduction

Multimodal large language models (MLLMs) have made remarkable advancements in tasks involving visual comprehension and understanding such as visual question answering (VQA), image captioning, and object grounding [2, 12, 13, 21, 22, 24]. However, these methods primarily focus on processing multimodal inputs but only generating textual outputs. This restricts their effectiveness in tasks that require generating visual content alongside text, such as graphic design generation. Graphic designs are composed of various multimodal elements including text, images, SVG shapes, colours, and spatial layout, all working together to convey meaning and visual appeal.

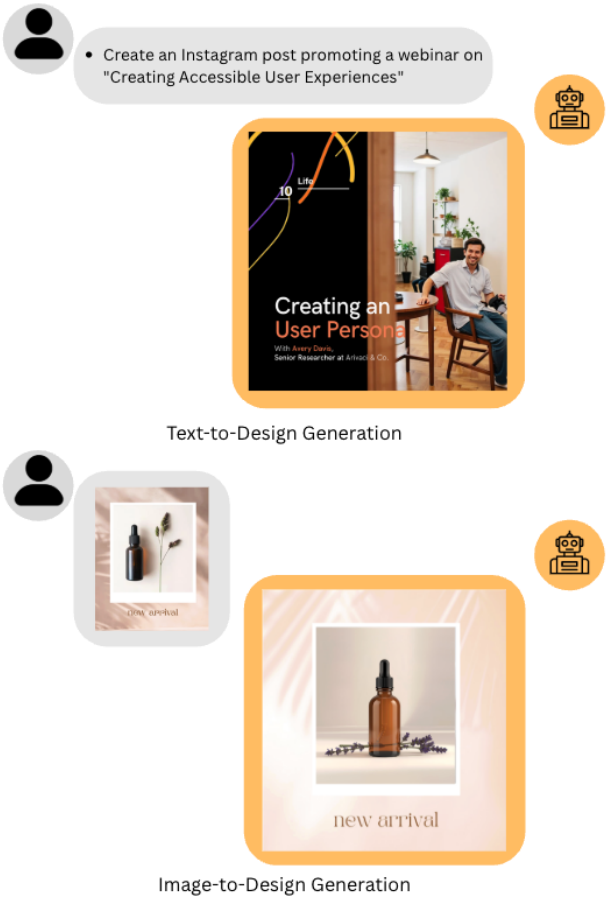


Figure 1. Example results from the proposed architecture. The model generates visually coherent and stylistically consistent graphic designs from either text or image inputs.

Recent approaches have explored the expansion of language models to accommodate the generation of multimodal outputs. In FROMAGe [10], image retrieval capabilities are introduced by augmenting a frozen language

model with two trainable linear layers. The first linear layer maps the hidden representation of a [RET] token whilst the second maps the visual embedding of the corresponding image. These two embeddings are aligned through contrastive learning [14], allowing the model to associate textual queries with relevant visual content. Nonetheless, this approach falls short when a design requires an image with a specific subject, style, or colour theme that is not available in the media library, or when user lacks access to a sufficiently large media collection.

GILL [9] extends the capabilities of FROMAGe by enabling media generation rather than just retrieval. This is achieved through a mapping network called GILLMapper - a transformer with both encoder and decoder components. GILLMapper translates the hidden representations of r [IMG] tokens, generated by the language model, into the embedding space of Stable Diffusion's [17] text encoder. The alignment between the two embedding spaces is learned using a mean squared error (MSE) loss. During inference, the images are generated by passing the mapped embeddings through the diffusion model.

Similarly, Emu [18] enables media generation by augmenting a language model with a module called Causal Transformer that transforms visual inputs into N [IMG] visual tokens. The model is trained to autoregressively predict these visual tokens, using an MSE loss between the predicted embeddings and the ground-truth tokens generated by the Causal Transformer. During inference, the generated visual tokens are decoded into images using a Stable Diffusion model that has been fine-tuned to explicitly condition on generated embeddings.

In Chameleon [19], images are represented using 1024 discrete tokens derived from a codebook of size 8192, produced by a learned image tokenizer. These image tokens, along with text tokens, are used to train the model autoregressively. During inference, a learned de-tokenizer is used to reconstruct images from generated image tokens.

Across all of these methods, a shared characteristic is the use of multiple visual tokens to represent images. In fact, [9] further investigates the impact of token count and finds that reducing the number of tokens generally leads to performance degradation as it produces shorter and less expressive inputs for the mapping network.This issue becomes particularly significant in graphic design generation tasks, where a single design may contain multiple images and is typically represented as a long, interleaved sequence of text and image tokens to capture elements such as text, images, SVG shapes, layout, and visual attributes. Consequently, representing each image with multiple tokens can cause the overall token sequence to become extremely long, often approaching or exceeding the model's maximum token capacity. Moreover, the limited attention span of transformer-based architectures makes it challenging to capture long-range dependencies across such extended sequences, thereby constraining the model's ability to produce coherent and consistent designs.

Some approaches have explored generating images directly rather than relying solely on embeddings. For instance, Transfusion [26] leverages a VAE [8] to encode images into latent representations and then applies a diffusion process within this latent space. In this setup, text is modelled autoregressively using a transformer while image generation is guided by a diffusion-based objective. DreamLLM [3], on the other hand, enables direct image generation by using a frozen Stable Diffusion decoder that is guided by semantic queries produced by the language model. These queries are trained using Score Distillation Sampling (SDS) [15], allowing the model to align textual prompts with image generation targets. Like the aforementioned methods, both these approaches still rely on multi-vector or multi-token representations per image, which may limit their effectiveness in design generation tasks.

This paper proposes a novel architecture that facilitates mapping from hidden token representations of a language model to the embedding space of a visual model, like CLIP ViT-L/14 [16], using just a single [IMG] token per image. This enables the language model to create coherent graphic designs by integrating text and image tokens within long sequences. The proposed mapping architecture, termed Giraffe due to its L-shaped structure, is composed of two shallow MLP blocks. Each block features a unique compression module, followed by a shared expansion module that links the two. One block is mainly responsible to handle the mapping, while the other assists the main block during the training phase, playing an essential role in helping the primary block to reach its mapping goal. When it comes to inference, the assisting block is removed, leading to a more lightweight solution. Since the generated embeddings follow a known distribution, there is no need to fine-tune a diffusion model. An existing pretrained model such as FLUX [11] with CLIP VIT-L/14 IP Adapter [23] can be used directly to generate images.

Extensive experiments highlight its strong performance in both text-to-design generation and image-to-design tasks. In the text-to-design generation task, it can be seen that the produced images align with the provided prompt while preserving the overall style, color palette, and semantics of the graphic design, leading to an aesthetically pleasing outcome. Likewise, in the image-to-design task, the generated graphic design closely mirrors the reference image. Collectively, the observed behavior indicates that the proposed architecture effectively captures all image-related information within a single [IMG] token. Fig. 1 shows an example for each task. It is important to note that while this paper focuses primarily on images, the proposed architecture is flexible and can be adapted for various media, including au-

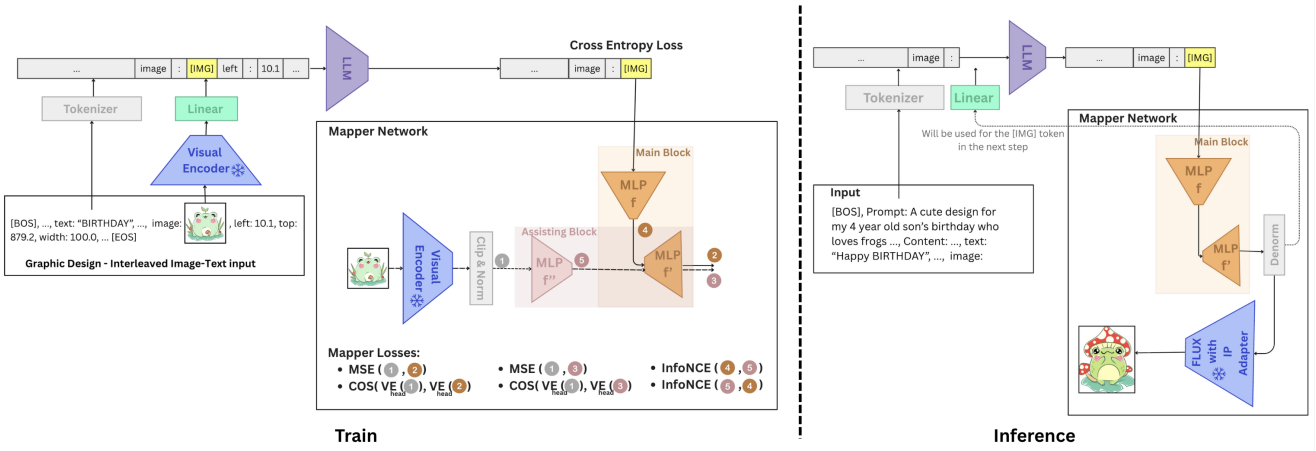


Figure 2. The proposed Giraffe architecture overview. It enables mapping from hidden token representations to the embedding space of visual models, such as CLIP ViT-L/14, using only a single `[IMG]` token. This is achieved using two shallow MLP blocks where each includes a seperate compression module followed by a shared expansion module, trained with six distinct loss functions (left). The assisting MLP block aids the main block in learning the mapping and is ignored during inference (right), leading to a more lightweight solution.

dio and video.

2. Method

This section describes the graphic design representation used by the model, the proposed Giraffe architecture, the training objectives, and the inference process.

2.1. Graphic Design Structure

A graphic design is composed of various elements, including text, images, and SVG shapes, which all work together to convey ideas and create visual harmony. Each element is defined by a set of properties that determine its structure and appearance. These include positional attributes such as top, left, width, height, and layering order, as well as visual characteristics like colour, transparency, and rotation. Certain attributes are also specific to particular element types. For example, text elements may include the textual content itself, font family, font size, and styling options such as bold or italic. By representing these element-specific properties in textual form, a graphic design can be represented as a long sequential structure of interleaved text and image components, where each x_t and x_v denotes a single text or image component, respectively. The inclusion of separate image components is necessary, as visual content cannot be fully captured through text alone.

2.2. Giraffe Architecture

Following prior work [4, 5, 13, 21, 25], the visual embedding of each image x_v is extracted and projected into the same dimensional space as the word embeddings, using the following transformations:

$$\begin{aligned} z_v &= v_\phi(x_v) \in \mathbb{R}^d, \\ u_v &= W z_v \in \mathbb{R}^m \end{aligned} \tag{1}$$

, where W is a learnable linear mapping, $v_\phi(.)$ is the pre-trained and frozen visual encoder (ViT-L/14) prior to its projection head and m represents the input embedding dimension of the language model. Each x_v in the input sequence is then replaced with just one special `[IMG]` token and its corresponding word embedding is substituted with u_v.

To train the model to understand and generate multimodal content, it is optimised with the standard next-token prediction objective. Specifically, the model is fine-tuned to predict each next token using the cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{M} \sum_{m=1}^{M} \sum_{t=1}^{T_m} \log P_{\psi,W} \left(w_t^{(m)} \mid w_{<t}^{(m)} \right) \tag{2}$$

, where M is the batch size, T_m is the length of the mth input sequence, and ψ denotes the language model parameters. Each token w_t may be either a text token or the special `[IMG]` token.

Finally, to enable image generation, a mapping network is integrated with the language model to decode each `[IMG]` token into its corresponding normalised visual embedding $\tilde{z}_v$. $\tilde{z}_v$ is obtained by first clipping the original visual embedding z_v to remove extreme values, followed by applying the min-max-normalisation technique to rescale the values into the range [-1, 1]. This mapping is learned using a multilayer perceptron (MLP) block, which comprises of a

compression module f_{θ_1} followed by an expansion module f'_{θ_s} with SiLU activation function applied throughout both modules. However, the last layer in f'_{θ_s} uses Tanh as its activation function. The parameters of this block are optimised by minimising the mean square error (MSE) loss between its output and $\tilde{z}_v$:

$$\mathcal{L}_{\text{b1-MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\tilde{z}}_{\text{v}_i} - \tilde{z}_{\text{v}_i}\right)^2, \quad \hat{\tilde{z}}_{\text{v}_i} = f'_{\theta_s}(f_{\theta_1}(h_\psi([\text{IMG}]_i))) \tag{3}$$

, where $h_\psi([IMG]_i)$ denotes the representation of the ith `[IMG]` token from the final hidden layer of the language model and N represents the number of `[IMG]` tokens in the batch.

Additionally, a cosine similarity loss is used to promote directional alignment between the predicted embeddings and the ground truth:

$$\mathcal{L}_{\text{b1-COS}} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{g_\varphi(\hat{\tilde{z}}_{\text{v}_i})\cdot g_\varphi(\tilde{z}_{\text{v}_i})}{\|g_\varphi(\hat{\tilde{z}}_{\text{v}_i})\|\|g_\varphi(\tilde{z}_{\text{v}_i})\|}\right) \tag{4}$$

, where $g_\varphi(.)$ denotes the projection head of the pretrained and frozen ViT-L/14 visual encoder. $g_\varphi(.)$ is applied to both predicted and target embeddings to ensure compatibility in the representation space.

While the proposed MLP block establishes the baseline mapping, it often struggles to learn the mapping effectively. To enhance this block's ability to fully capture the transformation, an assisting MLP block is introduced, forming the Giraffe architecture named for its L-shaped structure. The assisting block features its own compression module f''_{θ_2} with SiLU as activation function but shares the expansion module f'_{θ_s} with the main MLP block. Functionally, it operates as an autoencoder where the input is the normalised visual embedding $\tilde{z}_v$ and the output is its approximation $\bar{z}_{\text{v}_i}$. Similarly, it is trained with two losses:

$$\mathcal{L}_{\text{b2-MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\bar{z}_{\text{v}_i} - \tilde{z}_{\text{v}_i}\right)^2, \quad \bar{z}_{\text{v}_i} = f'_{\theta_s}(f''_{\theta_2}(\tilde{z}_{\text{v}_i})) \tag{5}$$

and,

$$\mathcal{L}_{\text{b2-COS}} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{g_\varphi(\bar{z}_{\text{v}_i})\cdot g_\varphi(\tilde{z}_{\text{v}_i})}{\|g_\varphi(\bar{z}_{\text{v}_i})\|.\|g_\varphi(\tilde{z}_{\text{v}_i}\|}\right). \tag{6}$$

To further assist f_{θ_1} in learning the mapping transformation, two infoNCE losses [14] are applied on the bottlenecks of the two MLP blocks:

$$\mathcal{L}_{\text{NCE}_1} = -\frac{1}{N}\sum_{i=1}^{N}\left(log\frac{e^{sim\left(f_{\theta_1}(\tilde{z}_{\text{v}_i}), f''_{\theta_2}(\tilde{z}_{\text{v}_i})\right)/\tau}}{\frac{1}{M}\sum_{j=1}^{M} e^{sim\left(f_{\theta_1}(\tilde{z}_{\text{v}_i}), f''_{\theta_2}(\tilde{z}_{\text{v}_j})\right)/\tau}}\right),$$
$$\mathcal{L}_{\text{NCE}_2} = -\frac{1}{N}\sum_{i=1}^{N}\left(log\frac{e^{sim\left(f_{\theta_1}(\tilde{z}_{\text{v}_i}), f''_{\theta_2}(\tilde{z}_{\text{v}_i})\right)/\tau}}{\frac{1}{M}\sum_{j=1}^{M} e^{sim\left(f_{\theta_1}(\tilde{z}_{\text{v}_j}), f''_{\theta_2}(\tilde{z}_{\text{v}_i})\right)/\tau}}\right), \tag{7}$$

where τ is a learnable temperature parameter.

The total loss is calculated as:

$$\begin{aligned}\mathcal{L}_{\text{total}} = {} & \lambda_1\mathcal{L}_{\text{b1-MSE}} + \lambda_2\mathcal{L}_{\text{b1-COS}} \\ & + \lambda_3\mathcal{L}_{\text{b2-MSE}} + \lambda_4\mathcal{L}_{\text{b2-COS}} \\ & + \lambda_5(\mathcal{L}_{\text{NCE}_1} + \mathcal{L}_{\text{NCE}_2}) + \lambda_6\mathcal{L}_{\text{CE}},\end{aligned} \tag{8}$$

where λ_1, λ_2, λ_3, λ_4, λ_5 and λ_6 are hyperparameters representing loss weights. The optimisation is carried out over parameters W, θ_1, θ_2, θ_s, and ψ. An overview of the architecture is shown in Fig. 2.

2.3. Inference

At inference time, f''_{θ_2} is removed, resulting in a lightweight mapping network. Each predicted `[IMG]` token is mapped into its normalised visual embedding as follows:

$$\hat{\tilde{z}}_{\text{v}} = f'_{\theta_s}(f_{\theta_1}(h_\psi([\text{IMG}]))), \tag{9}$$

which is then denormalised to recover the predicted visual embedding $\hat{z}_{\text{v}}$. During the autoregressive process, $\hat{z}_{\text{v}}$ is used directly instead of recomputing z_v via the visual encoder $v_\phi(.)$ for each generated `[IMG]` token, enabling faster token generation.

Finally, real images are produced by passing $\hat{z}_{\text{v}}$ to FLUX with CLIP VIT-L/14 IP Adapter. The generated images are then placed within the specified bounding boxes to create the final graphic design.

3. Experiments

To evaluate the proposed method, two experiments are carried out on text-to-design and image-to-design generation tasks. The model is trained on a proprietary dataset of 1,800,000 professionally created graphic designs. These graphic designs span a wide range of formats, including social media posts, banners, flyers, business cards, and logos. The dataset is split into training and validation subsets, with 90% used for training and 10% for validation.

In the mapping network, $f_{\theta_1}(.)$ consists of layers with dimensions $1024-768-512$, while $f''_{\theta_2}(.)$ has dimensions $1024-768-512-512$, with the final layer in both serving as the bottleneck. $f'_{\theta_s}(.)$ is composed of layers with dimensions $768-1024-1024$. The hyperparameters λ_1, λ_2, λ_3, λ_4, λ_5 and λ_6 are set to 1, 1, 1, 1, 0.1, and 1, respectively.

1: collage style instagram post celebrating friend's birthday (4)

2: Instagram post celebrating my friend group with a scrapbook photo collage (4)

3: A whimsical birthday party invitation for kids, featuring cute illustrations and playful typography (3)

4: A heartwarming instagram post wishing a happy birthday (6)

5: An Instagram post for handmade wooden toys as perfect hanukkah gifts (1)

6: A vibrant Instagram post announcing a summer smoothie sale with fresh fruit illustrations and bright colours (1)

7: A fun flyer promoting a kids' art workshop with playful fonts, bright colours, and doodle-style graphics (2)

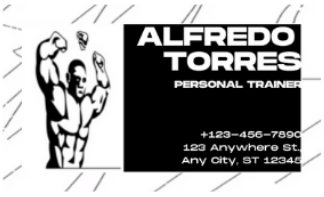
8: A vintage-style flyer for a farmers' market, incorporating rustic textures and classic typography (2)

9: Solar panel installation company advertising instagram post (8)

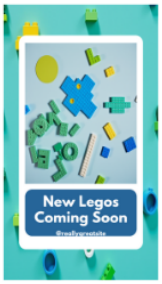
10: Art exhibition abstract Instagram post (1)

11: Children's storytime library whimsical Instagram post (1)

12: A heartwarming instagram post wishing a happy birthday (9)



13: A bold, high-contrast business card for a personal trainer, using striking typography and dynamic imagery (2)



14: A playful Instagram post showcasing new Lego products with a blue and green background (2)

15: Visually striking poster for an outdoor yoga retreat, featuring soft greens, serene imagery, and calming fonts (2)

16: An elegant business card for a wedding planner, incorporating soft pastels, floral patterns, and elegant fonts (2)

Figure 3. Qualitative results for the text-to-design task. The generated graphic designs closely follow the prompts while remaining visually coherent, stylistically consistent, and of high overall quality. The model often produces complex designs with multiple images that interact harmoniously. Numbers in parentheses indicate the number of images used in each design.

1: collage style instagram post celebrating friend's birthday (2)

2: Instagram post celebrating my friend group with a scrapbook photo collage (2)

3: A whimsical birthday party invitation for kids, featuring cute illustrations and playful typography (3)

4: A heartwarming instagram post wishing a happy birthday (2)

5: An Instagram post for handmade wooden toys as perfect hanukkah gifts (1)

6: A vibrant Instagram post announcing a summer smoothie sale with fresh fruit illustrations and bright colours (1)

7: A fun flyer promoting a kids' art workshop with playful fonts, bright colours, and doodle-style graphics (4)

8: A vintage-style flyer for a farmers' market, incorporating rustic textures and classic typography (1)

9: Solar panel installation company advertising instagram post (1)

10: Art exhibition abstract Instagram post (1)

11: Children's storytime library whimsical Instagram post (1)

12: A heartwarming instagram post wishing a happy birthday (1)

13: A bold, high-contrast business card for a personal trainer, using striking typography and dynamic imagery (1)

14: A playful Instagram post showcasing new Lego products with a blue and green background (1)

15: Visually striking poster for an outdoor yoga retreat, featuring soft greens, serene imagery, and calming fonts (1)

16: An elegant business card for a wedding planner, incorporating soft pastels, floral patterns, and elegant fonts (2)

Figure 4. Qualitative results for the baseline model. Generated results have limited layout variation and less consistent visual style. Numbers in parentheses indicate the number of images used in each design.

Table 1. FID score and CLIP cosine similarity for text-to-design and image-to-design tasks, respectively.

Model	Text-to-Design (FID)	Image-to-Design (Cosine)
Proposed	66.85	0.85 ± 0.05
Baseline	81.61	-

3.1. Text-to-design generation

In this experiment, the Gemma3 4B model [20] is used as the language model. Training is carried out in mixed precision using the bfloat16 [1] format for 56,000 steps. Evaluation is conducted on a separate set of curated prompts, chosen to cover a wide range of cases. Each design sample is represented in JSON format, with floating-point bounding box coordinates rounded to the nearest integer. Furthermore, each design is accompanied by a textual description that captures its style, theme, and key details, for example, "A vibrant Instagram post announcing a summer smoothie sale with fresh fruit illustrations and bright colours". For benchmarking, a baseline model is also trained without the mapping network, in which textual description of each image is generated instead and subsequently passed to FLUX for image generation.

To evaluate the performance of the model, a comprehensive qualitative analysis is performed on the generated results. Figure 3 and Fig. 4 show sample outputs generated by the proposed architecture and the baseline model, respectively, for the same set of prompts. The number of images in each design is indicated in parentheses. As shown in these figures, the proposed method generates designs that are more visually appealing, with greater layout diversity and stronger stylistic and thematic coherence. Many of the generated designs include multiple images that work harmoniously in terms of color theme and style, as seen in prompts 3, 4, 9, and 12. In contrast, the baseline model often produces designs with minimal layout variation, featuring a background image, a central shape, and overlaid text. When the baseline generates designs containing multiple images, the images usually lack color harmony and appear visually not very relevant to the overall design. For instance, in prompt 7, the images generated by the baseline model do not exhibit consistent color or style, whereas the corresponding design produced by the proposed method maintains a coherent theme, with images that complement each other and enhance the overall design coherence. These limitations in the baseline model likely arise from representing each image solely through textual descriptions. The resulting increase in token length, together with the limited attention span of transformer-based architectures, makes it difficult to capture long-range dependencies and constrains the model's ability to generate more complex designs. In addition, representing images as text alone makes it challenging to accurately capture their style and color theme. In contrast, the `[IMG]` token represents each image with a single token, allowing the model to focus on finer-grained details within its attention span. Moreover, the `[IMG]` token preserves the style and thematic information of an image more faithfully, enabling the proposed model to generate outputs that are more complex, visually coherent, stylistically consistent, and better aligned with the intended design.

To further evaluate the performance of the proposed mapping network, the FID score [6] is computed by comparing the statistical distributions of the generated designs with those of real, professionally created designs from the training set. As shown in Tab. 1, the proposed method achieves a considerably lower FID score than the baseline model, indicating that it generates designs of higher quality and greater diversity, consistent with the findings of the qualitative study.

3.2. Image-to-design generation

In this experiment, the language model is replaced with a transformer with both encoder and decoder components, each consisting of 10 layers. The architecture has 500 million learnable parameters. Each design in the training set is first rendered, and the resulting image is divided into patches. These patch values are then concatenated with the CLIP embedding [16] of the rendered image and fed into the transformer's encoder. The model is trained to generate the corresponding design as a sequence of text interleaved with `[IMG]` tokens, following the approach used in the previous experiment. However, since replacing the language model with a plain transformer and using a relatively small training set makes learning natural language from scratch challenging, the content of each textbox is replaced with the word "TEXT" to simplify the task. Training is performed in float32 precision for 187,000 steps. Evaluation is conducted on a small set of graphic designs generated from curated prompts using GPT-4o [7].

To evaluate the model's performance, both quantitative and qualitative studies are conducted. Cosine similarity is computed between the CLIP embedding of each original input image and the generated design to measure how closely the generated design matches the input. As shown in Tab. 1, the cosine similarity of CLIP embeddings is 0.85 ± 0.05, indicating strong adherence to the original images. Figure 5 illustrates qualitative examples produced by the trained model. In each pair, the left image is the GPT-4o generated input, and the right image is the model's output. For easier assessment, the original text content is manually added to the generated text boxes. While minor inconsistencies in details are observed, the outputs generally align closely with the inputs. These results demonstrate that a single `[IMG]` token per image captures substantial seman-

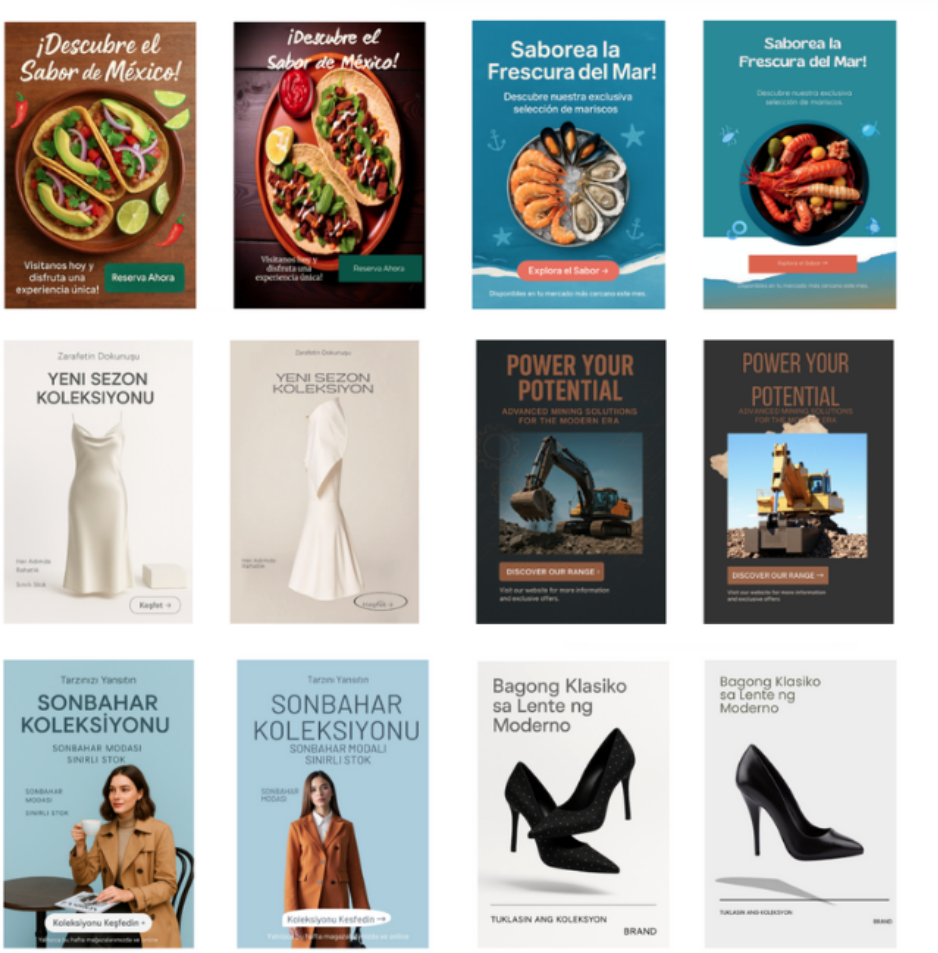


Figure 5. Qualitative results for the image-to-design task. In each pair, the left image shows the input generated by GPT-4o, and the right image shows the model's output.

tic, stylistic, and color information. They also suggest that knowledge transferred from a pretrained language model has minimal effect on the model's ability to map `[IMG]` tokens to CLIP embeddings. However, a pretrained language model is still required for generating the text content.

4. Conclusion

This paper introduces Giraffe, a novel architecture that maps hidden token representations from a language model to the embedding space of a visual model using a single `[IMG]` token per image. This approach allows language models to generate graphic designs, which would normally require very long sequences of interleaved text and image tokens. The mapping architecture consists of two shallow MLP blocks, each with a unique compression module followed by a shared expansion module that connects the blocks. The main block performs the mapping, while the second block assists in training to help the main block learn the mapping more effectively. During inference, the assisting block is removed, resulting in a lightweight solution. Experiments on a text-to-design task show that the architecture effectively captures the semantics, style, and color of each image, producing more complex designs with greater visual harmony than when images are represented solely through textual descriptions. The results also emphasize the importance of using a single token per image to achieve higher diversity and complexity in the generated graphic designs. Similarly, experiments on an image-to-design task demonstrate that the proposed architecture can capture semantics, style, and colour information with just one `[IMG]` token per image.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow:

Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 7

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 1

[3] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023. 2

[4] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. Magma–multimodal augmentation of generative models through adapter-based finetuning. In *Findings of the association for computational linguistics: EMNLP 2022*, pages 2416–2428, 2022. 3

[5] Hao Fei, Shengqiong Wu, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. *Advances in neural information processing systems*, 37:57207–57239, 2024. 3

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[7] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 7

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[9] Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36:21487–21506, 2023. 2

[10] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *International Conference on Machine Learning*, pages 17283–17300. PMLR, 2023. 1

[11] Black Forest Labs. Flux.1-dev, 2024. Accessed: 2025-06-10. 2

[12] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1, 3

[14] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 4

[15] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[16] 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*, pages 8748–8763. PmLR, 2021. 2, 7

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[18] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023. 2

[19] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 2

[20] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 7

[21] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 1, 3

[22] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International conference on machine learning*, pages 23318–23340. PMLR, 2022. 1

[23] XLabs-AI. flux-ip-adapter-v2, 2025. Accessed: 2025-06-10. 2

[24] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer, 2022. 1

[25] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Towards perceiving small visual details in zero-shot visual question answering with multimodal llms. *arXiv preprint arXiv:2310.16033*, 2023. 3

[26] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. 2