WordArt Designer: User-Driven Artistic Typography Synthesis using Large Language Models

Jun-Yan He[1] **Zhi-Qi Cheng**[2*] **Chenyang Li**[1] **Jingdong Sun**[2] **Wangmeng Xiang**[1]
Xianhui Lin[1] **Xiaoyang Kang**[1] **Zengke Jin**[3,4] **Yusen Hu**[2,5]
Bin Luo[1] **Yifeng Geng**[1] **Xuansong Xie**[1]
[1]Alibaba DAMO Academy [2]Carnegie Mellon University [3]Zhejiang Sci-Tech University
[4]Royal College of Art [5]Imperial College London

Abstract

This paper introduces *WordArt Designer*, a user-driven framework for artistic typography synthesis, relying on the Large Language Model (LLM). The system incorporates four key modules: the *LLM Engine*, *SemTypo*, *StyTypo*, and *TexTypo* modules. 1) The *LLM Engine*, empowered by the LLM (e.g. GPT-3.5), interprets user inputs and generates actionable prompts for the other modules, thereby transforming abstract concepts into tangible designs. 2) The *SemTypo module* optimizes font designs using semantic concepts, striking a balance between artistic transformation and readability. 3) Building on the semantic layout provided by the *SemTypo module*, the *StyTypo module* creates smooth, refined images. 4) The *TexTypo module* further enhances the design's aesthetics through texture rendering, enabling the generation of inventive textured fonts. Notably, *WordArt Designer* highlights the fusion of generative AI with artistic typography. Experience its capabilities on ModelScope: https://www.modelscope.cn/studios/WordArt/WordArt.

1 Introduction

Typography, a critical intersection of language and design, finds extensive applications across various domains like advertising (Cheng et al., 2016, 2017a,b; Sun et al., 2018), early childhood education (Vungthong et al., 2017), and historical tourism (Amar et al., 2017). Despite its widespread relevance, the mastery of typography design remains an intricate task for non-professional designers. Although attempts have been made to bridge this gap between amateur designers and typography (Iluz et al., 2023; Tanveer et al., 2023), existing solutions mainly generate semantically coherent and visually pleasing typography within predefined concepts. These solutions often lack adaptability, creativity, and computational efficiency.

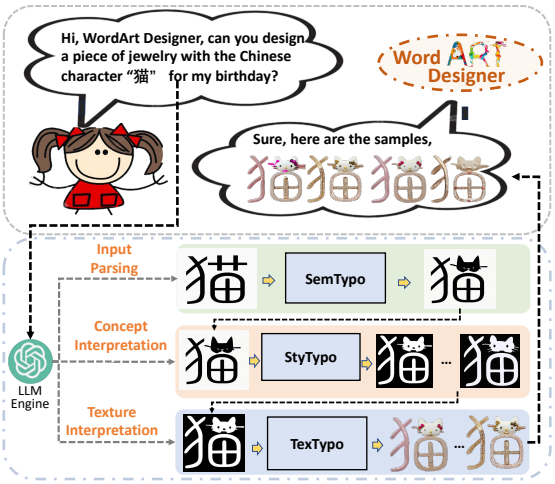

Figure 1: Demonstration of WordArt Designer: Leveraging the power of the LLM (e.g. GPT-3.5), it integrates four modules (LLM Engine, SemTypo, StyTypo, TexTypo) to transform user inputs into visually striking and semantically rich multilingual typographic designs. It democratizes the art of typography design, enabling non-professionals to realize their creative visions.

To overcome these limitations, we introduce WordArt Designer (Fig. 1), a system composed of four primary modules: the LLM Engine, SemTypo Module, and StyTypo Module, supplemented by the TexTypo Module for texture rendering. This user-focused system allows users to define their design needs, including design concepts and domains. The system consists of:

1. **LLM Engine**: Based on the power of the LLM (e.g. GPT-3.5), this engine interprets user input and produces prompts for the SemTypo, StyTypo, and TexTypo modules.

2. **SemTypo Module**: The SemTypo Module transforms typography based on a provided semantic concept. It involves a three-step process, including Character Extraction and Parameterization, Region Selection for Transformation, and Semantic Transformation & Differentiable Rasterization.

*Corresponding author

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 223–232
December 6-10, 2023

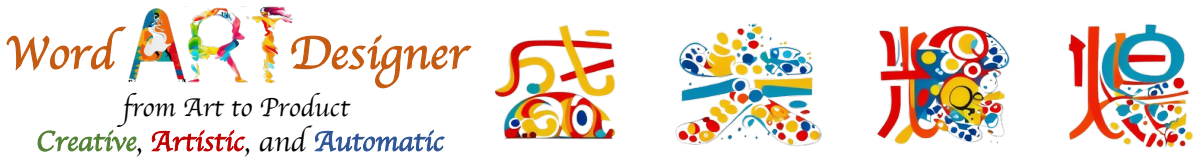

Figure 2: Examples of artistic typography generated by WordArt Designer. These instances demonstrate the system's ability to produce aesthetically pleasing, semantically coherent, and stylistically diverse typographic designs.

3. **StyTypo Module**: The StyTypo Module generates smoother, more detailed images based on the semantic layout image provided by the SemTypo module.

4. **TexTypo Module**: The TexTypo Module modifies ControlNet for texture rendering, ensuring creativity while preserving legibility.

The workflow, illustrated in Fig. 1, begins with the LLM module interpreting user input. The output of each module serves as the input for the next, with the final design decision made by the TexTypo module. This sequence ensures the final design aligns with the user's intent and maintains a unique aesthetic appeal.

This design process is iterative, involving constant interaction between the user's input and the system's modules. This user-centered approach guarantees the creation of high-quality WordArt designs (See Fig. 2), making it an effective tool in creative design-dependent industries, such as food and jewelry.

Extensive experiments on WordArt Designer have validated its creativity, artistic expression, and expandability to different languages. The inclusion of a ranking model significantly improves the success rate and overall quality of stylized images, ensuring the production of high-quality WordArt designs.

In essence, WordArt Designer provides a creative, artistic, and fully automated solution for generating word art. Our research not only lays the groundwork for future text synthesis studies but also introduces numerous practical applications. WordArt Designer can be employed in various areas, including media propaganda and product design, enhancing the efficiency and effectiveness of these systems, thereby making them more practical for everyday use.

2 Related work

LLM and their Apps. Large Language Model (LLM) has been progressively improved and utilized in a wide range of applications (Anil et al., 2023; Raffel et al., 2020; Shoeybi et al., 2019; Rajbhandari et al., 2020; Devlin et al., 2019; Cheng et al., 2023). The outstanding performances exhibited by the ChatGPT and GPT series (Radford et al., 2018; Brown et al., 2020; OpenAI, 2023) have stimulated the widespread use of the LLM. These models are adept at learning context from simple prompts, leading to their increasing use as the controlling component in intelligent systems (Wu et al., 2023; Shen et al., 2023). Building on these insights, WordArt Designer incorporates the LLM to enhance system creativity and diversity.

Text Synthesis. While significant progress has been made in image synthesis, integrating legible text into images remains challenging (Rombach et al., 2022; Saharia et al., 2022). Some solutions, such as eDiff-I (Balaji et al., 2022) and DeepFloyd (Lab, 2023), employ robust LLMs, such as T5 (Raffel et al., 2020), for improved visual text generation. Recent studies (Yang et al., 2023; Ma et al., 2023) have also looked into using glyph images as extra control conditions, while others like DS-Fusion (Tanveer et al., 2023) introduce additional constraints to synthesize more complex text forms, such as hieroglyphics.

Image Synthesis. The surge in demand for personalized image synthesis has spurred advances in interactive image editing (Meng et al., 2022; Gal et al., 2023; Brooks et al., 2022; Zhao et al., 2018) and techniques incorporating additional conditions, such as masks and depth maps (Rombach et al., 2022; Huang et al., 2020). New research (Zhang and Agrawala, 2023; Mou et al., 2023; Huang et al., 2023) is exploring multi-condition controllable synthesis. For instance, ControlNet (Zhang and Agrawala, 2023) learns task-specific conditions end-to-end, providing more nuanced control over the synthesis process.

Text-to-Image Synthesis. Significant strides in denoising diffusion probabilistic models have substantially enhanced text-to-image synthesis (Ho et al., 2020; Ramesh et al., 2021; Song et al., 2021; Dhariwal and Nichol, 2021; Nichol and Dhariwal, 2021;

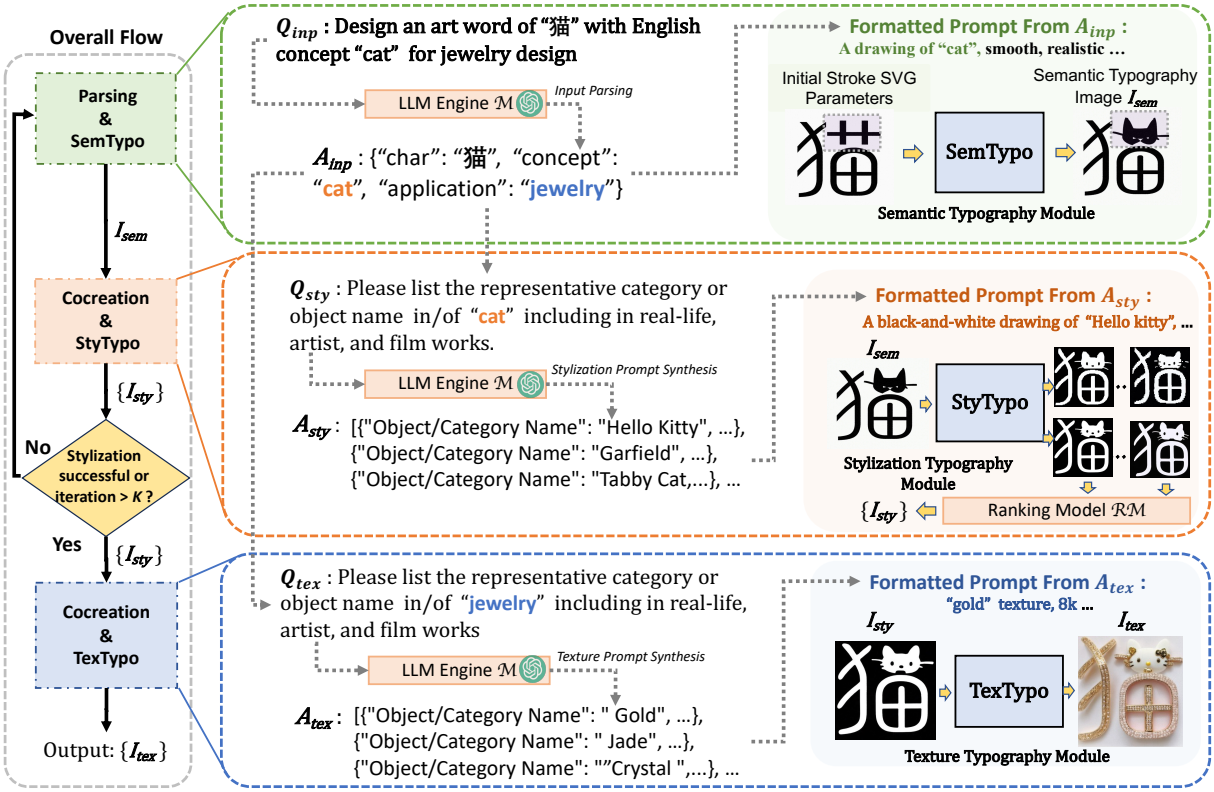

Figure 3: The architectural framework of the proposed WordArt Designer system. This structure involves an LLM engine, the SemTypo module for Semantic Typography, the StyTypo module for Stylization Typography, and the TexTypo module for Texture Typography. These modules operate coherently, guided by a preset control flow, to facilitate a seamless and innovative transformation of text into artistic typography.

Saharia et al., 2022; Ramesh et al., 2022; Rombach et al., 2022). Notable examples of these advancements are latent diffusion models such as Imagen (Saharia et al., 2022), DALLE-2 (Ramesh et al., 2022) and LDM (Rombach et al., 2022), which have enabled high-quality image generation.

3 WordArt Designer

The WordArt Designer system utilizes an assortment of typography synthesis modules, propelled by a Large Language Model (LLM) such as GPT-3.5), facilitating an interactive, user-centered design process. As illustrated in Fig. 3, users define their design needs, including design concepts and domains, e.g., "A cat in jewelry design." The LLM engine interprets the input, generating prompts to guide SemTypo, StyTypo, and TexTypo modules, thus executing the user's design vision.

To achieve automated WordArt design, we introduce a quality assessment feedback mechanism, which is vital for successful synthesis. The output from the ranking model is evaluated by the LLM engine to validate the quality of the synthesized image, ensuring the creation of at least K qualified transformations. If this criterion is not met, the LLM engine, along with the SemTypo and StyTypo modules and format directives, are restarted for another design iteration. Subsequent sections will delve into the details of each module's functionality and operation.

3.1 LLM Engine

The Large Language Model (LLM) engine is a crucial component for the WordArt designer. It serves as a knowledge engine and concretizes abstract notions, like "vegetables" and "fruit", into texture prompts in the context of food, for the eventual synthesis of the artistic text. For most concrete nouns, such as "cat", "dog", "flower", etc., semantic typography can be successfully generated. However, for words like abstract nouns and verbs, such as "winter", "hit", etc., users often fail to provide desired descriptions. The reason is that images compose highly complex scenes for abstract concepts, which is not suitable for our WordArt designer system.

To address this issue, we employ the LLM to render abstract concepts into representative objects that can be easily converted. Specifically, we can build our LLM engine using models like GPT-3.5 and

other LLMs, all of which have context-learning capabilities. The prompts for input parsing, stylization, and texture rendering are generated as:

$$A_{inp} = \mathcal{M}(Q_{inp}), A_{sty} = \mathcal{M}(Q_{sty}), A_{tex} = \mathcal{M}(Q_{tex}) \tag{1}$$

Where Q_{inp}, Q_{sty}, and Q_{tex} represent the standard prompts for input parsing, stylization, and texture rendering respectively. Q_{sty} and Q_{tex} are built using formatted prompt templates with concepts derived from the input parsing. LLM engine has ample capabilities to imbue our system with a creative and engaging "soul", ensuring the quality of artistic text synthesis. We provide detailed templates and full examples of prompts in Appendix A.

3.2 SemTypo Module

The Semantic Typography (SemTypo) module alters typographies based on a given semantic concept. It unfolds in three stages: (1) Character Extraction and Parameterization, (2) Region Selection for Transformation, and (3) Semantic Transformation and Differentiable Rasterization.

Character Parameterization. The first stage, as displayed in Fig. 3, starts by transforming the natural language input into a JSON format, specifying the characters to modify, the semantic concept, and the application domain. The FreeType font library (David Turner et al., 1996) is then employed to extract character contours and convert them into cubic Bézier curves characterized by a trainable set of parameters. For characters with surplus control points, a subdivision routine fine-tunes the control points θ, using a differentiable vector graphic rasterization scheme (Iluz et al., 2023).

Region Selection. Our unique contribution is the region-based transformation method, the second stage of the SemTypo module. This approach facilitates the selective transformation of certain character segments, effectively reducing distortions that typically affect typography generation in languages with single-character words. We choose to transform a random contiguous subset of control points within a character, instead of the entire character. We establish a splitting threshold of 20 pixels, with the set of control points randomly determined within the range [500, min(1000/control point count)], initiating from a random point.

In contrast to previous methods, such as the one by Iluz et al. (Iluz et al., 2023), which used extra loss terms with inadequate success to maintain legibility of the synthesized typography, our method only involves loss computation from the transformed sections of the characters. This approach increases efficiency and guarantees careful manipulation of character shapes, thus improving transformation quality.

Transformation and Rasterization. In the final stage, the parameters are transformed and rasterized through the Differentiable Vector Graphics (DiffVG) scheme (Li et al., 2020). As shown in Fig. 4, the transformed glyph image I_{sem} is created from the trainable parameters θ of the SVG-format character input, using DiffVG $\boldsymbol{\phi}(\cdot)$. A segment of the chosen character x is optimized and cropped to yield an enhanced image batch X_{aug} (Frans et al., 2022). The semantic concept S and the augmented image batch X_{aug} are both input into a vision-language backbone model to compute the loss for parameter optimization. The Score Distillation Sampling (SDS) loss is applied in the latent space code z, as per the DreamFusion method (Poole et al., 2023):

$$\nabla_\theta \mathcal{L}_{SDS} = \mathcal{E}_{t,\epsilon}[w(t)(\hat{\epsilon_\phi}(a_t z_t + \sigma_t \epsilon, y) - \epsilon)\frac{\partial z}{\partial X_{aug}}\frac{\partial X_{aug}}{\partial \theta}] \tag{2}$$

Here, $t \in \{1, 2, \ldots, T\}$ is uniformly sampled to define a noise latent code $z_t = a_t z_t + \sigma_t \epsilon$, with $\epsilon N \sim (0, 1)$, and a_t, σ_t act as noise schedule regulators at time t. The multiplier $w(t)$ is a constant, contingent on a_t. This revised process refines expression and amplifies the variety of output.

3.3 StyTypo Module

The Stylization Typography (StyTypo) module's main purpose is to generate smoother and more detailed images, enhancing the semantic layout image I_{sem}. To speed up I_{sty} generation, we use short iteration settings. However, this approach might lead to a lack of smoothness and details. To overcome these potential drawbacks, the StyTypo module introduces two main components: (1) stylized image generation, and (2) stylized image ranking and selection.

Stylized Images Generation. The Latent Diffusion Model (LDM) (Rombach et al., 2022) has gained attention for its ability to generate images based on given input shapes. Therefore, we employ the LDM's depth2image methodology to stylize typographic layouts, enhancing smoothness and infusing additional detail to create a unique "sketch" for texture rendering. Fig. 5 illustrates this, where

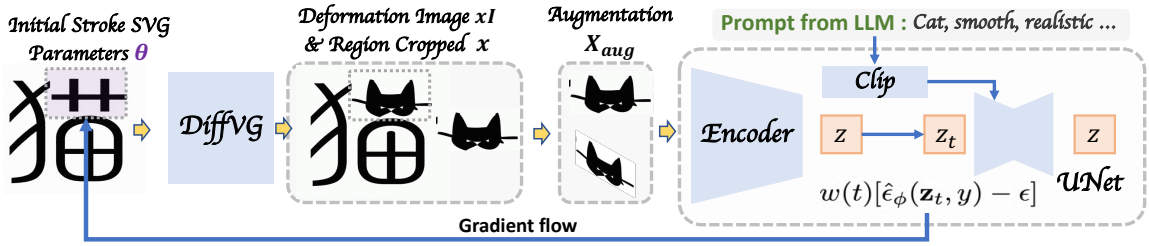

Figure 4: Differential rasterization scheme of semantic typography. The character stroke inside the purple box is the selected part for optimization.

the top row images generated by the SemTypo module, despite lacking smoothness and detail, provide a comprehensive object representation. After being processed by the StyTypo module, the stylized images on the lower row display an abundance of detail and inventive renderings for each semantic image input.

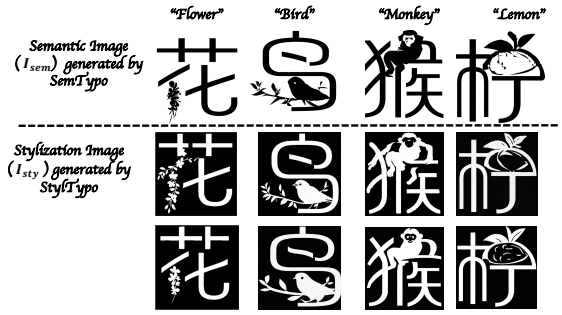

Figure 5: Comparison of the semantic and stylization images. Stylization images contain more details.

Formally, given a semantic typography image I_{sem} from the SemTypo module, and a stylization prompt A_{sty} synthesized by the LLM Engine $\mathcal{M}$, we can create the stylization image I_{sty} as:

$$I_{sty} = \text{StyTypo}(I_{sem}, A_{sty}) \quad (3)$$

where StyTypo utilizes the depth2image pipeline derived from the LDM (Rombach et al., 2022) to carry out the stylization.

Ranking and Selection. To augment the StyTypo module's efficiency, we introduce a ranking model that orders and filters the generated results. Specifically, we establish a quality evaluation dataset consisting of stylized characters classified into two groups: (1) Successful Stylization, and (2) Failed Stylization. The dataset encompasses 141 single-character Chinese characters and 5,814 stylized typographic images. We leverage the ResNet18 classification model (He et al., 2016) to learn the quality distribution of the stylization images. During the filtering stage, the trained model serves as a ranking model, providing ranking scores. Based on these scores, the top 'x' results are selected.

3.4 TexTypo Module

To advance the styling capacities of the Stylization Typography (StyTypo) module, we adapted ControlNet (Zhang and Agrawala, 2023) for the purpose of texture rendering, resulting in the creation of the Texture Typography (TexTypo) module.

As can be seen in Fig. 6, ControlNet's original control conditions relied heavily on the Canny Edge and Depth data. This constraint tended to produce fonts that were lacking in creativity and artistic flair. To counter this, we introduced Scribble conditions as an alternate control condition into ControlNet, which encourage the generation of more creatively textured fonts without compromising on readability.

Figure 6: Comparison between Canny Edge and Scribble conditions for ControlNet texture rendering. The first row is generated using the Canny Edge condition, while the rest are from the Scribble condition.

Furthermore, to cater to a range of industrial sectors, we have reconfigured ControlNet to incorporate pre-trained stable diffusion models that are relevant to different fields. These include, but are not limited to, commercial advertising, fashion design, gaming interfaces, tech products, and artistic creations.

Technically, we provide the ControlNet parameters with conditions Canny Edge, Depth, Scribble, as well as original font images. The TexTypo model receives these parameters and generates the tex-

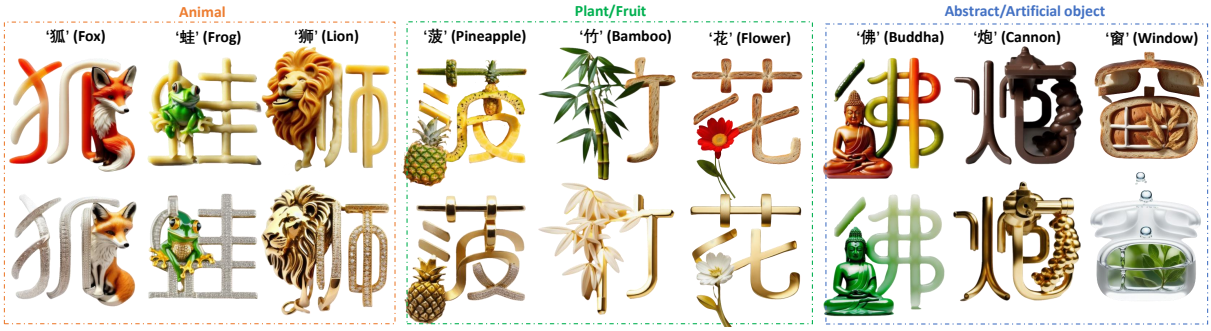

Figure 7: Results showcasing the adaptability of the WordArt Designer. The first row targets the concept of "food", which is further specified to "candy", "pasta", "cheese", "fruits", "bread", "vegetables" or "chocolate". The second row targets "jewelry", concretized to "jewels", "gold" or "jade". The variety of styles highlighted underscores WordArt Designer's versatility in creating unique artistic typography, pushing past traditional design boundaries.

tured font image as,

$$I_{tex} = \text{TexTypo}(I_{sty}, A_{tex}, P_{cond}), \quad (4)$$

where A_{tex} represents the prompts synthesized by the LLM engine $\mathcal{M}$, and P_{cond} stands for the control parameters, resulting in a creatively rendered textured font as the output.

4 Deployment Details

WordArt Designer tool has successfully been integrated into ModelScope, utilizing the TongYi QianWen 14b model as the LLM engine. In terms of deployment, the StyTypo and TexTypo Modules are hosted in Docker containers, ensuring both flexibility and scalability in deployment. StyTypo is powered by a Linux platform with 48 cores, 384GB RAM, and 4 Nvidia V100 32GB GPUs, taking roughly 32 seconds to generate 4 images. In contrast, TexTypo operates within a similar Linux environment but with 24 cores, 64GB RAM, and a single Nvidia V100 32GB GPU, and typically produces 4 images within a span of 5 to 10 seconds. For the Ranking Model, the `mmpretrain` (Contributors, 2023) is used to train a ResNet 18 model (He et al., 2016), with a total of 100 epoches at a batch size of 32. The SGD optimizer was used with a learning rate of 0.01. The training ran on a single Nvidia V100 16GB GPU.

5 Experiments

Creativity & Artistic Ability. We operationalize the concept of texture rendering to evaluate the Creativity and Artistic Ability of the WordArt Designer. The outcomes are demonstrated in Fig. 7. The first row of art words is generated by embodying the concept "food", which is further specified to "candy", "pasta", "cheese", "fruits", "bread", "vegetables" or "chocolate". The second row represents the concept "jewelry", concretized to "jewels", "gold" or "jade". The smart and reasonable texture rendering contributes to the creativity and artistic appeal of the output.

Expandability to Different Languages. Our SemTypo module, grounded on differentiable rasterization, is theoretically compatible with all types of languages. Beyond Chinese (i.e., hieroglyphs), we explore the expandability of WordArt Designer with the representative language, English (i.e., the Latin alphabet). Fig. 8 presents a collection of rendered results for Chinese characters and corresponding English words, substantiating that WordArt Designer effectively accommodates different languages.

Effect of Ranking Model. To determine the effectiveness of the ranking model, we divide the aforementioned character dataset into a training and validation set by randomly selecting 20 characters for validation. We use precision and recall to measure the model's ability to classify individual images as successfully stylized or not. In addition, we compare WordART Designer's overall success rate on transforming a character using the Ranking Model and a Random Model (a character is deemed successfully transformed if at least one of the output images is successfully stylized). As shown in Table 1, our ranking model significantly outperforms the random model, indicating its efficacy. When top-10 images are selected, we guarantee that each character has at least one successfully stylized image. To balance precision and recall, the number of selected images should ideally range from 2 to 5.

Table 1: Ablation study of the ranking model on the validation set. 'p', 'r', and 's' stand for precision, recall, and success rate, respectively. 'x' in 'TopX' indicates the number of stylized images retained. In the ranking-based method, 'TopX' are selected based on ranking scores, while for the random-based method, 'TopX' are selected randomly. Results of the random-based method are obtained by averaging over 10,000 iterations. Increased values are indicated in blue.

Methods	Metric	Top1	Top2	Top5	Top10
Random	p	18.3	18.1	18.2	18.2
	r	4.5	8.9	22.4	44.8
	s	18.3	33.1	63.4	86.5
Ranking	p	**60.0**↑41.7	**62.5**↑44.4	**46.0**↑27.8	**32.0**↑13.8
	r	**14.6**↑10.1	**30.8**↑21.9	**56.3**↑33.9	**78.8**↑34.0
	s	**60.0**↑41.7	**80.0**↑46.9	**85.0**↑21.6	**100.0**↑13.5

Figure 8: Chinese Characters and their corresponding English art words.

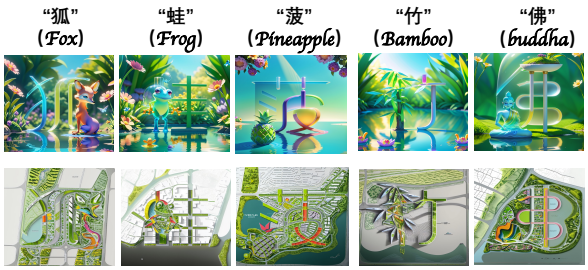

Figure 9: Various notable applications of our WordArt Designer, including art word poster creation (row 1) and urban master plan design (row 2). Note that reVAnimated is used as the base LDM (Rombach et al., 2022). For rows 1-2, we further apply the Lora models Blindbox and MasterPlan respectively.

5.1 Application

WordArt Image. We experiment with various application possibilities for WordArt Designer. The results, exhibited in Fig. 9, are representative and not cherry-picked. WordArt Designer exhibits promising potential in areas like art word poster design and even city planning. We are confident that WordArt Designer will bring innovative inspiration to professional designers.

WordArt Animation. We also utilize ControlVideo (Zhang et al., 2023) to synthesize art word videos, illustrating the transformation of the word/character. The Chinese characters for Bamboo" and Flower" are used in the video generation process, with the "Van Gogh's painting" style applied to the animations, proving useful for Chinese education. Please refer to Fig. 10 for additional animations.

(a) Bamboo (van Gogh) (b) Follower (van Gogh)

Figure 10: **Art word animations** derived from the SemTypo optimization process. CLICK the image to PLAY ANIMATION! Best viewed with Adobe Acrobat DC.

6 Ethical Considerations

Potential ethical concerns include perpetuating cultural stereotypes due to the use of certain imagery or symbols in the process of artistic transformations, or introducing bias against under-represented cultures. Another issue could be the potential inclusion of copyrighted graphics. Users need to pay attention to these issues to ensure responsible and respectful use of the system.

7 Conclusion

This paper presents WordArt Designer, a framework that harnesses Large Language Models (LLM), such as GPT-3.5, to automatically generate multilingual artistic typography. This system uses an LLM engine to parse and translates user input into directives, guiding three modules, each accountable for different aspects of the typographic design. The superior performance of WordArt Designer highlights the potential of AI to augment artistic typography. Future work aims to further explore the possibilities of integrating this technology into other aspects of design, such as graphics and interactive media.

Acknowledgments

The contributions of Zhi-Qi Cheng in this project were supported by the Army Research Laboratory (W911NF-17-5-0003), the Air Force Research Laboratory (FA8750-19-2-0200), the U.S. Department of Commerce, National Institute of Standards and Technology (60NANB17D156), the Intelligence Advanced Research Projects Activity (D17PC00340), and the US Department of Transportation (69A3551747111). Intel and IBM Fellowships also provided additional support for Zhi-Qi Cheng's research work.

References

Jennifer Amar, Olivier Droulers, and Patrick Legohérel. 2017. Typography in destination advertising: An exploratory study and research perspectives. *Tourism Management*, 63:77–86.

Rohan Anil, Andrew M. Dai, Orhan Firat, et al. 2023. Palm 2 technical report. *arXiv preprint*, abs/2305.10403.

Yogesh Balaji, Seungjun Nah, Xun Huang, et al. 2022. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint*, abs/2211.01324.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2022. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint*, abs/2211.09800.

Tom B. Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Zhi-Qi Cheng, Qi Dai, Siyao Li, et al. 2023. Chartreader: A unified framework for chart derendering and comprehension without heuristic rules. In *Proceedings of the IEEE/CVF international conference on computer vision*.

Zhi-Qi Cheng, Yang Liu, Xiao Wu, and Xian-Sheng Hua. 2016. Video ecommerce: Towards online video advertising. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1365–1374.

Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua. 2017a. Video ecommerce++: Toward large scale online video advertising. *IEEE transactions on multimedia*, 19(6):1170–1183.

Zhi-Qi Cheng, Xiao Wu, Yang Liu, and Xian-Sheng Hua. 2017b. Video2shop: Exact matching clothes in videos to online shopping images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4048–4056.

MMPreTrain Contributors. 2023. Openmmlab's pretraining toolbox and benchmark. `https://github.com/open-mmlab/mmpretrain`.

David Turner, Robert Wilhelm, and Werner Lemberg. 1996. FreeType 2.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794.

Kevin Frans, Lisa B. Soros, and Olaf Witkowski. 2022. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. In *NeurIPS*.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, et al. 2023. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Lianghua Huang, Di Chen, Yu Liu, et al. 2023. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint*, abs/2302.09778.

Siyu Huang, Haoyi Xiong, Zhi-Qi Cheng, et al. 2020. Generating person images with appearance-aware pose stylizer. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

Shir Iluz, Yael Vinker, Amir Hertz, et al. 2023. Word-as-image for semantic typography. *SIGGRAPH*.

DeepFloyd Lab. 2023. Deepfloyd if. *https://github.com/deep-floyd/IF*.

Tzu-Mao Li, Michal Lukác, Michaël Gharbi, and Jonathan Ragan-Kelley. 2020. Differentiable vector graphics rasterization for editing and learning. *SIGGRAPH*, 39(6):193:1–193:15.

Jian Ma, Mingjun Zhao, Chen Chen, et al. 2023. Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. *arXiv preprint*, abs/2303.17870.

Chenlin Meng, Yutong He, Yang Song, et al. 2022. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*.

Chong Mou, Xintao Wang, Liangbin Xie, et al. 2023. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint*, abs/2302.08453.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *ICML*, volume 139, pages 8162–8171.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint*, abs/2303.08774.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OPENAI, https://openai.com/research/language-unsupervised*.

Colin Raffel, Noam Shazeer, Adam Roberts, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:140:1–140:67.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *SC*, page 20.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, et al. 2022. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint*, abs/2204.06125.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, et al. 2021. Zero-shot text-to-image generation. In *ICML*, volume 139, pages 8821–8831. PMLR.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*.

Yongliang Shen, Kaitao Song, Xu Tan, et al. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint*, abs/2303.17580.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, et al. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint*, abs/1909.08053.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, et al. 2021. Score-based generative modeling through stochastic differential equations. In *ICLR*.

Guang-Lu Sun, Zhi-Qi Cheng, Xiao Wu, and Qiang Peng. 2018. Personalized clothing recommendation combining user social circle and fashion style consistency. *Multimedia Tools and Applications*, 77:17731–17754.

Maham Tanveer, Yizhi Wang, Ali Mahdavi-Amiri, and Hao Zhang. 2023. Ds-fusion: Artistic typography via discriminated and stylized diffusion. *arXiv preprint*, abs/2303.09604.

Sompatu Vungthong, Emilia Djonov, and Jane Torr. 2017. Images as a resource for supporting vocabulary learning: A multimodal analysis of thai efl tablet apps for primary school children. *TESOL Quarterly*, 51(1):32–58.

Chenfei Wu, Shengming Yin, Weizhen Qi, et al. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint*, abs/2303.04671.

Yukang Yang, Dongnan Gui, Yuhui Yuan, et al. 2023. Glyphcontrol: Glyph conditional control for visual text generation. *arXiv preprint*, abs/2305.18259.

Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint*, abs/2302.05543.

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, et al. 2023. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint*, abs/2305.13077.

Bo Zhao, Xiao Wu, Zhi-Qi Cheng, et al. 2018. Multi-view image generation from a single-view. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 383–391.

A Prompt Examples

Prompt for GPT: "*Now you are a creative and active explainer that helps people to understand the abstract concept, and concrete the abstract concepts, tell them the representative object in the abstract concept. All the question is in a standard format "Please list the representative category or object name in/of <CONCEPT>, including in real-life, artist, and film works." And the output must be in a strict JSON format: {"Object/Category Name": "Name", "description": "less than five critical words to describe", "reason": "your detailed reason for the choice"}. Example: " Please list the representative category or object name in/of cat including in real-life, artist, and film works." The response should be {"Object/Category Name": "Hellokitty", "description": "cute, happiness" "less than five critical words to describe", "reason": "famous for the cartoon"}.* "

Stylization and texture examples as follow:

Q_s: "Please list the representative category or object name in/of spring, including in real-life, artist, and film works."

A_s: "*{"Object/Category Name": "Rainbow", "description": "colorful, natural", "reason": "Rainbows are a natural phenomenon that occurs after rain showers during spring. They are often depicted in artwork and films as a symbol of hope, joy, and promise. Additionally, rainbows are often used in fashion and design to represent spring and its vibrant colors."}*"

Q_t: "Please list the representative category or object name in/of food, including in real-life, artist, and film works."

A_t: "*{"Object/Category Name": "Pizza", "description": "delicious, versatile", "reason": "Pizza is a popular food that is loved by many people around the world. It is a versatile food that can be customized with a variety of toppings to suit different tastes and preferences. Pizza is often featured in films, TV shows, and commercials, and it is a staple food in many countries, including Italy and the United States."}*"

The "Object/Category Name" and the "description" are utilized to build the prompt for the StyTypo and TexTypo modules, and the "reason" information can be applied to analyze the quality of the prompt.

B Addtional Results

Figure 11: Diversity of results. The LLM engine generates the texture prompt that can be explained in various concretion objects/concepts. The 1st and 2nd rows are related to the concept "jewelry" that is concrete to "gold" or "jade", respectively. The 3rd row is the concept "food" that is concrete to bread. It is worth noticing that the texture rendering is "smart" and "reasonable" which leads to creativity and artistry.