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[1], Yusen Hu[2], Bin Luo[1], Yifeng Geng[1], Xuansong Xie[1], Jingren Zhou[1]

[1]Alibaba DAMO Academy [2]Carnegie Mellon University

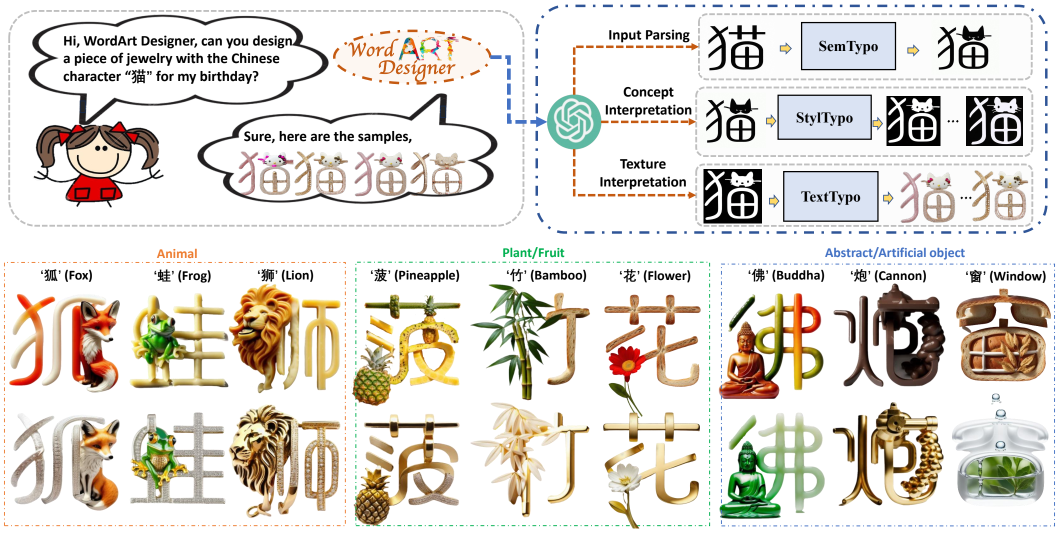

Figure 1: **Top**: Showcasing the WordArt Designer: Driven by GPT-3.5-turbo and a suite of specialized visual models, WordArt Designer seamlessly transmutes user text prompts into stunning, semantically resonant multilingual typographic masterpieces. With user-friendly interactions, it democratizes the artistic process, inviting all, irrespective of their design expertise, to realize their creative dreams. **Bottom**: Examples of artistic typography generated by WordArt Designer, in the Animal, Plant, and Abstract categories. We welcome you to explore the online demo here: https://www.modelscope.cn/studios/WordArt/WordArt

1 Introduction

Typography, a critical intersection of language and design, finds extensive applications across various domains including advertising [5], early childhood education [11], and historical tourism [1]. Despite its relevance, the mastery of artistic typography design remains an arduous task for non-professionals. Some attempts have been made to improve typography's accessibility to amateur designers [6, 10], but they are often constrained to predefined concepts, lack adaptability, creativity, and are computationally inefficient.

We introduce WordArt Designer (Fig. 1), a user-driven framework for artistic typography synthesis using Large Language Models (LLMs). The workflow, illustrated in the top-half of Fig. 1, begins with the LLM module interpreting user input. This generation process is iterative, in that a quality assessment feedback system automatically retries designs until at least a set number of successful transformations is produced, ensuring the creation of high-quality WordArt designs.

*Correspondence author

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, advertisement, and product design, enhancing the efficiency and effectiveness of these systems, making them more practical for everyday use.

Examples of artistic typographies generated by WordART are presented in the bottom-half of Fig. 1. We welcome you to further explore WordART's capabilities at https://www.modelscope.cn/studios/WordArt/WordArt.

2 Method

The WordArt Designer system utilizes an assortment of typography synthesis modules, propelled by a Large Language Model (LLM) such as GPT-3.5-turbo, facilitating an interactive, user-centered design process. As illustrated in Fig. 2, 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, StylTypo, and TextTypo 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 StylTypo modules and format directives, are restarted for another design iteration.

LLM Engine. Concrete nouns such as "cat" or "flower" correspond to well-defined imagery, while abstract nouns (e.g. "winter") often fail to pinpoint a specific visual outcome. The Large Language Model (LLM) Engine interprets the user's input and concretizes any abstract notions to provide specific instructions for subsequent modules. For more details, see Appendix B.

SemTypo Module. The Semantic Typography module (SemTypo) 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. For more details, see Appendix C.

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

TextTypo Module. To advance the styling capacities of the Stylization Typography (StylTypo) module, we adapted ControlNet [12] for the purpose of texture rendering, resulting in the creation of the Texture Typography (TextTypo) module, which generates the final textured font image. For more details, see Appendix E.

3 Evaluation

Creativity & Artistic Abilities. WordART Designer's capacity to generate rich and visually pleasing typographies are recognized, and is already deployed onto production platforms such as ModelScope, which is also used internally by the Alibaba Group. Selected outcomes are shown in the bottom-half of Fig. 1, demonstrating the different textures (e.g. organic vs metallic for food and jewelry) that WordART Designer utilizes when generating typographies for different contexts.

Expandability to Different Languages. Using differentiable rasterization, WordART Designer is theoretically compatible with all types of languages. Fig. 3 presents a collection of rendered results for Chinese characters and corresponding English words, substantiating that WordArt Designer works effectively with languages beyond Chinese.

Effect of Ranking Model. The ranking model in WordART Designer's feedback system significantly improves the likelihood of producing successful transformations. When the top-10 images are selected, we guarantee that each character has at least one successfully stylized image. For more details, see Appendix F.

4 Ethical Implications

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.

References

[1] Jennifer Amar, Olivier Droulers, and Patrick Legohérel. "Typography in destination advertising: An exploratory study and research perspectives". In: *Tourism Management* 63 (2017), pp. 77–86. ISSN: 0261-5177. DOI: https://doi.org/10.1016/j.tourman.2017.06.002. URL: https://www.sciencedirect.com/science/article/pii/S0261517717301243.

[2] David Turner, Robert Wilhelm, and Werner Lemberg. *FreeType 2*. FreeType, 1996. URL: https://freetype.org/index.html (visited on 07/24/2023).

[3] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. "CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders". In: *NeurIPS*. 2022.

[4] Kaiming He et al. "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*. 2016, pp. 770–778.

[5] Xianzhong He. "Transitivity of kinetic typography: Theory and application to a case study of a public service advertisement". In: *Visual Communication* 16.2 (2017), pp. 165–194.

[6] Shir Iluz et al. "Word-As-Image for Semantic Typography". In: *SIGGRAPH* (2023).

[7] Tzu-Mao Li et al. "Differentiable vector graphics rasterization for editing and learning". In: *SIGGRAPH* 39.6 (2020), 193:1–193:15.

[8] Ben Poole et al. "DreamFusion: Text-to-3D using 2D Diffusion". In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. 2023.

[9] Robin Rombach et al. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *CVPR*. June 2022, pp. 10684–10695.

[10] Maham Tanveer et al. "DS-Fusion: Artistic Typography via Discriminated and Stylized Diffusion". In: *arXiv preprint* abs/2303.09604 (2023).

[11] Sompatu Vungthong, Emilia Djonov, and Jane Torr. "Images as a Resource for Supporting Vocabulary Learning: A Multimodal Analysis of Thai EFL Tablet Apps for Primary School Children". In: *TESOL Quarterly* 51.1 (2017), pp. 32–58. DOI: https://doi.org/10.1002/tesq.274. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/tesq.274. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/tesq.274.

[12] Lvmin Zhang and Maneesh Agrawala. "Adding Conditional Control to Text-to-Image Diffusion Models". In: *arXiv preprint* abs/2302.05543 (2023).

A Figures

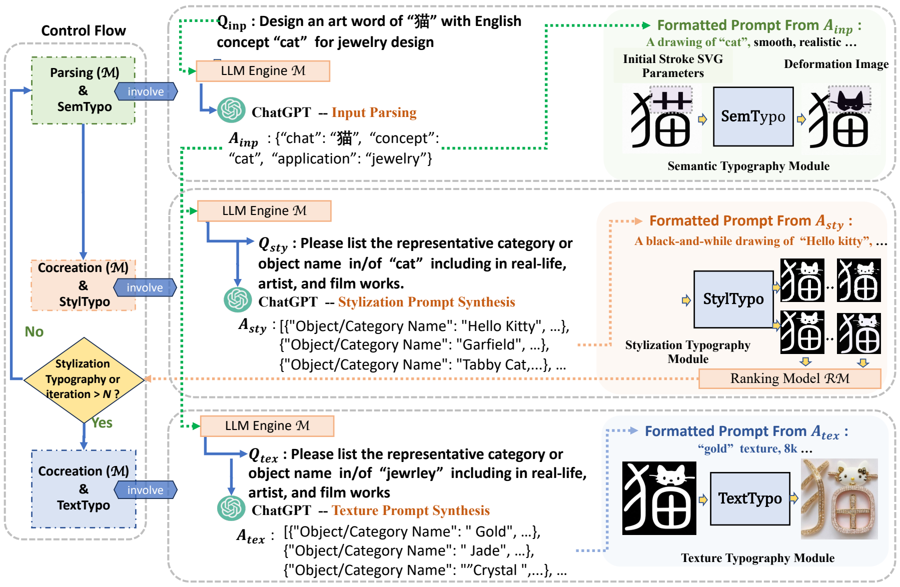
Control Flow
Parsing (ℳ) & SemTypo
involve
Cocreation (ℳ) & StylTypo
No
Stylization Typography or iteration > N ?
Yes
Cocreation (ℳ) & TextTypo
Q_{inp} : Design an art word of "猫" with English concept "cat" for jewelry design
LLM Engine ℳ
ChatGPT -- Input Parsing
A_{inp} : {"chat": "猫", "concept": "cat", "application": "jewelry"}
Formatted Prompt From A_{inp} :
A drawing of "cat", smooth, realistic ...
Initial Stroke SVG Parameters
Deformation Image
SemTypo
Semantic Typography Module
Q_{sty} : Please list the representative category or object name in/of "cat" including in real-life, artist, and film works.
ChatGPT -- Stylization Prompt Synthesis
A_{sty} : [{"Object/Category Name": "Hello Kitty", ...}, {"Object/Category Name": "Garfield", ...}, {"Object/Category Name": "Tabby Cat,...}, ...
Formatted Prompt From A_{sty} :
A black-and-while drawing of "Hello kitty", ...
StylTypo
Stylization Typography Module
Ranking Model ℛℳ
Q_{tex} : Please list the representative category or object name in/of "jewrley" including in real-life, artist, and film works
ChatGPT -- Texture Prompt Synthesis
A_{tex} : [{"Object/Category Name": " Gold", ...}, {"Object/Category Name": " Jade", ...}, {"Object/Category Name": ""Crystal ",...}, ...
Formatted Prompt From A_{tex} :
"gold" texture, 8k ...
TextTypo
Texture Typography Module

Figure 2: The architecture of the WordArt Designer system.

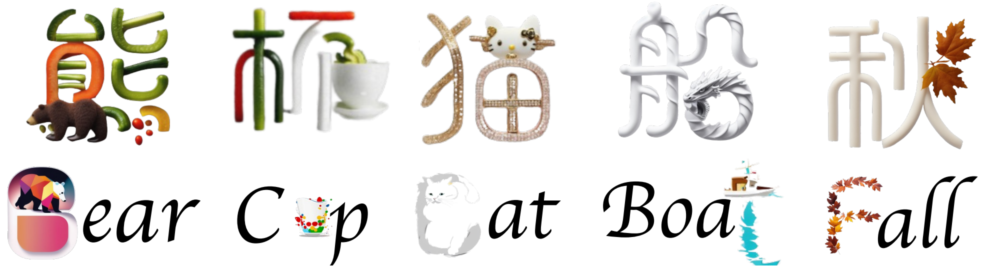

Figure 3: Chinese characters and their corresponding English art words.

B *LLM Engine* Details

We employ the LLM to render abstract concepts into representative objects that can be easily converted. Specifically, we build our LLM engine using gpt-3.5-turbo, which has 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.

Prompt for ChatGPT3.5: "*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 StylTypo and TextTypo modules, and the "reason" information can be applied to analyze the quality of the prompt.

C *SemTypo* Module Details

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. 2, 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 [2] 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 [6].

Region Selection. Our unique contribution is the region-based transformation method, the second stage of the SemTypo module. This approach facilitates 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. [6], 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 [7]. 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} [3]. 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 [8]:

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

Here, $t \in \{1, 2, \dots, T\}$ 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.

D *StylTypo* Module Details

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

Stylized Images Generation. The Latent Diffusion Model (LDM) [9] 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.

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:

$$I_{sty} = \text{StylTypo}(I_{sem}, A_{sty}) \tag{3}$$

where StylTypo utilizes the depth2image pipeline derived from the LDM [9] to carry out the stylization.

Ranking and Selection. To augment the StylTypo 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 [4] 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.

E *TextTypo* Module Details

To advance the styling capacities of the Stylization Typography (StylTypo) module, we adapted ControlNet [12] for the purpose of texture rendering, resulting in the creation of the Texture Typography (TextTypo) module.

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, which encourage the generation of more creatively textured fonts without compromising on readability.

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 TextTypo model receives these parameters and generates the textured font image as,

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

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.

F Ablation Study on Ranking Model

To perform an ablation study of the ranking model, we divide the character dataset into a training set and a validation set by randomly selecting 20 characters for validation. Precision, recall, and success rate are employed as evaluation metrics. Among them, precision and recall measure image-level performance, while the success rate is a character-level metric (a character is deemed successful if at least one of its images is successfully stylized.). 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