Harnessing Layered Graphic Designs with Real Intentions for Text-to-Design Generation

Xinya Song Bo Yang Ying Cao*
ShanghaiTech University

Abstract

Text-to-design generation, which synthesizes plausible and diverse graphic designs from textual design intentions, has recently emerged as a research area that gains growing interest. However, further process in this area is hindered by the absence of public, high-quality paired intention-design data. To mitigate this issue, we introduce LADEREIN, a new benchmark of layered designs with real intentions for training and evaluating text-to-design models. As opposed to synthetic short intention prompts in prior datasets, the intentions of our dataset are real, long and complex, spanning various factors such as design purpose, visual style, feeling, and expected audience behavior. Such real intentions allow us to train text-to-design models that generalize to real design scenarios, and make our dataset a promising ground for validating progress in text-to-design generation. For objective assessment of model performance in intention-following, we develop, DesignCLIP, a new text-design alignment metric. Moreover, we build a simple yet effective text-to-design generator, DesignDiff, as a baseline on our benchmark. We show that: 1) our DesignCLIP outperforms GPT-4V in judging the alignment of graphic designs and textual design intentions; 2) our LADEREIN dataset enhances the capabilities of text-to-design models in following complex user intentions accurately; 3) our DesignDiff is able to generate high-quality designs of great text alignment.

1. Introduction

Graphic design serves as an important medium for visual communication, conveying intended messages through the selection and composition of multimodal elements, such as images and texts. However, creating effective graphic designs is challenging, requiring a lot of design expertise, experience, and creativity. This challenge has stimulated tremendous interest in developing generative models to automate some design sub-tasks or the entire design generation process [2, 7, 10, 13, 15, 18, 25, 31] .

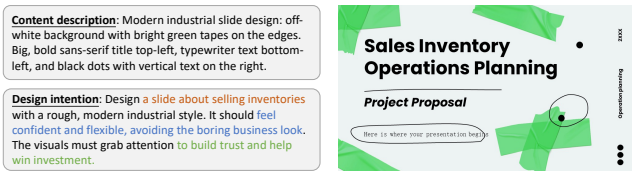

Figure 1. Comparison between a content-focused textual description (top left) and a design intention (bottom left) for a graphic design (right). For the design intention, we highlight the design purpose (orange), feelings to convey (blue), and expected audience actions (green).

Motivated by the success of recent text-to-image models [20, 23, 24], several attempts [14, 15] have been made to address *text-to-design* generation, which aims to automatically translate textual design intentions into specific graphic designs. Compared to text-to-image generation, text-to-design generation faces several unique difficulties. First, common image descriptions that focus on what are contained in the generated images (e.g., objects, their attributes and relationships). However, as shown in Figure 1, the textual inputs to text-to-design models are *vague design intentions* that specify information such as design purposes, visual styles, feelings to convey, and actions that audiences are expected to take, without explicitly mentioning concrete objects in the generated designs. Thus, the models should be able to understand abstract concepts in the input intentions, and map them into specific designs. Second, unlike images that are 2D grids of pixels, graphic designs are essentially a composition of multimodal elements with rich attributes (e.g., category, position, and font color) that can be easily edited for exploring a variety of variations. Hence, for the models to be practically useful, they should generate *highly editable outputs* to support iterative refinement and customization during the real design process.

Despite encouraging results from the recent works, an obstacle to further advancing text-to-design generation exists—lack of high-quality datasets of text-design pairs. Existing *public* layered graphic design datasets such as

*Corresponding author.

Crello [29] do *not* contain design intentions. Previous text-to-design works bypass this issue by leveraging pretrained large language models (e.g., GPT) to generate *synthetic* intentions for model training [14, 15]. However, due to the gap between synthetic and real intentions, the models trained on the synthetic data may have limited generalizability to complex design intentions encountered in real-world design tasks, which would compromise the practical applicability of the models. In view of this problem, we introduce **LADEREIN**, a new graphic design dataset especially for text-to-design generation, which encompasses **LA**yered **DE**signs with **RE**al **IN**tensions. LADEREIN consists of 13,635 *layered* graphic designs (PowerPoint templates), each of which is represented by a structured document encompassing a rich set of element attributes. Furthermore, each design is paired with a *real, detailed* design intention description. To the best of our knowledge, LADEREIN is the first public graphic design dataset that has real, long user intention descriptions. The intentions in our dataset are regarded by design experts as being more aligned with the corresponding designs, compared to synthetic ones. These real design intentions enables training text-to-design models with better generalizability to real use cases, and allows our dataset to better represent real-world design tasks, which makes it a promising ground for testing text-to-design models under the realistic setting. We show that LADEREIN enables text-to-design models to have strong capabilities to comprehend and follow complex and diverse design intentions.

To evaluate the intention-following capabilities of text-to-design models, prior works largely rely on pretrained large multimodal models (LMMs) [15]. It remains unclear, however, whether the pretrained LMMs can well understand graphic designs and design intentions, and connect them in a reliable manner. To address this issue, we introduce *DesignCLIP*, a specialized metric for evaluating text-design alignment, and demonstrate that DesignCLIP shows higher agreement with human judgments compared to the LMM-based evaluation. We additionally offer a simple text-to-layered-design model as a baseline on our dataset, which future works can be easily compared with.

The main contributions of this paper are as follows:

- *LADEREIN.* We present a new dataset that, for the first time, contains real, long design intentions, along with the corresponding graphic designs in layered format.
- *DesignCLIP.* We introduce a new quantitative metric to evaluate the intention-following abilities of text-to-design models more reliably than the commonly used LMM-based metric.
- *Open-source assets.* We make our dataset, metric and baseline publicly available to facilitate future work in the field of text-to-design generation.[1]

[1] https://songxyjoy.github.io/LADEREIN/

2. Related Work

2.1. Graphic Design Generation

Several general approaches have been explored to generate layered graphic designs that are fully editable. CanvasVAE [29] treats a graphic design as a set of canvas and element attributes and trains a VAE to learn a distribution over the attribute-based representation space. GOL [30] uses the learned element order to train generative models that generates graphic designs as sequences of element attribute tokens. Recently, a number of works emerge to automatically compose a set of multimodal elements into a cohesive design, by leveraging large multimodal models (LMMs) [5, 18]. However, none of the above works consider textual design intentions as inputs.

Text-to-design generation has received considerable interest in the past few years. Some methods try to fine-tune existing text-to-image diffusion models to generate highly aesthetic graphic design images [4, 27], but the generated designs in pixel space prevents editing elements independently. COLE [15] and OpenCOLE [14] build cascaded pipelines of a large language model (LLM), a LMM and text-to-image diffusion models to generate layered graphic designs from brief design intentions. ART [21] generates graphic designs with a large number of transparent element layers that can be edited separately. Several recent works decompose a generated design image into a sequence of layers through progressive top-layer object extraction that iteratively applies large vision models (such as segmentation and inpainting models) [3, 26] Motivated by recent advances in multimodal creation in LMM, IGD [22] finetunes a LLM alongside a diffusion model to generate layered graphic designs as mixed text-image data. In this work, we implement a simple text-to-design diffusion model as an effective baseline on our benchmark.

2.2. Graphic Design Datasets

With a rising interest in building generative models for graphic design, various public graphic design datasets has recently emerged. Some of them are for layout generation [6, 10, 28, 31, 33–35], containing graphic design images with layout annotations that mostly include element categories (e.g., text and logo) and bounding box coordinates, but lacking other element attributes (e.g., font face and color) that are necessary for complete design rendering. Crello [29] consists of vector graphic designs with rich element attributes, but with synthetic, short design intentions. In contrast, our dataset contains real, complex design intentions.

2.3. Evaluation of Design Generation Models

Finding reliable quantitative metrics for measuring the performance of text-to-design models is still an open problem.

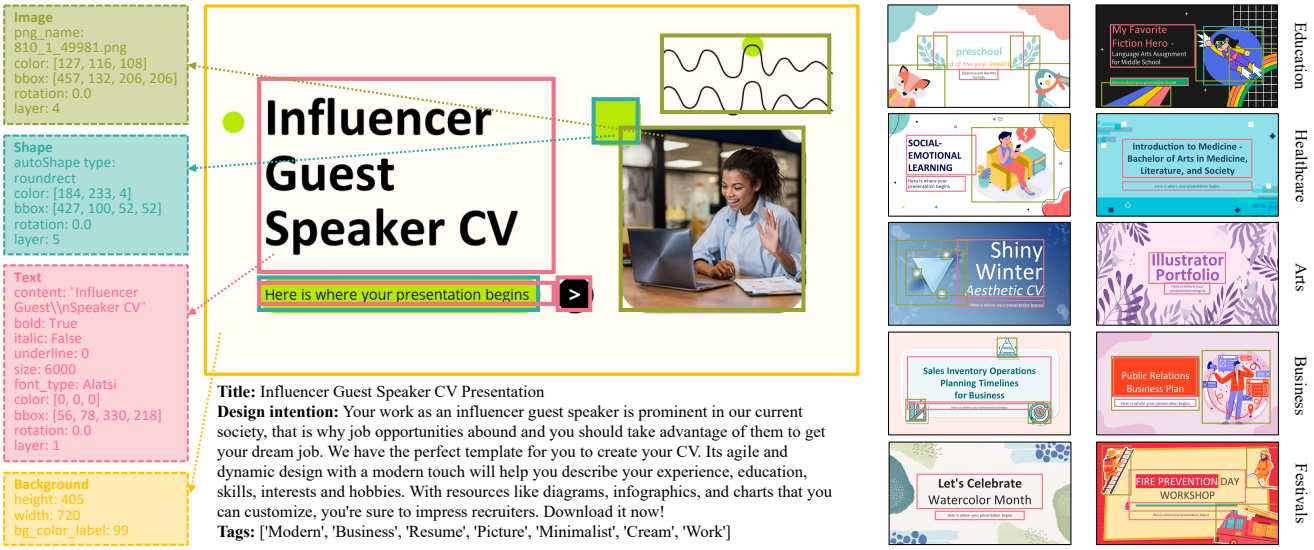

Figure 2. *Left*: Each design in our dataset is composed of basic elements of four different types: text, image, shape, and background. Different element types are highlighted with bounding boxes in different colors. Each element has a set of attributes that define its content and control its appearance and composition on the design. The detailed attributes of one element per type are shown in the dashed rectangle. Each design is accompanied by a detailed design intention alongside with the title and tags. *Right:* The dataset encompasses various themes such as education, healthcare, arts, business, and festivals.

Previous works [14, 15] reply on LMMs such as GPT-4V to assess the quality of generated designs across several important dimensions. However, the accuracy and reliability of such LMM-based evaluation has not yet been studied thoroughly. A recent study shows that GPT-4o can reliably evaluate graphic designs based on some common graphic design principles such as alignment and white space [9] , but there is no empirical evidence showing that LMMs can provide a reliable measure of how well graphic designs align with design intentions. In this work, we develop an intention-design alignment metric that are more consistent with human judgments than LMMs.

3. Our Dataset

Our goal is to train text-to-design generation models that can 1) effectively understand and follow complex text descriptions, aligned with user design intention, and 2) generate outputs that naturally support element-level editing. To this end, we construct a new dataset, termed as *LADEREIN*, which comprises graphic designs in a layered format, captioned with real, complex textual design intentions.

Dataset Construction. Graphic designs in our LADEREIN dataset are obtained from an online PowerPoint (PPT) template repository, *Slidesgo.com*. To construct the dataset, we first downloaded a large number of PPT templates in *PPTX* format and associated metadata stored in *JSON* format. Since a PPT template contains a sequence of slides, we choose the first slide, whose design is often representative of that of all the slides, as a design in our dataset.

To obtain the layered representation of designs, we identified elements in each design and extracted their detailed attributes from the corresponding template file, by employing the open-source library *python-pptx* [1] and a HTML parser. After filtering out element types that rarely appear across all the designs (e.g., line and table) and consolidating functionally similar element types (e.g., some placeholders as text), we consider four types of elements: *image*, *text*, *shape*, *background*. While all element types share a set of common attributes including type, layer, position and size, different element types have their unique, type-specific attributes. For example, an image element has an image attribute (i.e., the image content); a text element has a text attribute (i.e., the text content) as well as styling parameters (e.g., font family and color). There are a total of 26 distinct attributes in our dataset. After the element and attribute extraction, we create the layered representation $\mathbf{X}$ of each design as a set of elements, each of which is described by its attributes. We additionally render the image of the design as its raster representation $\mathbf{I}$.

For each design, we also extract information, including a title l, tags $\mathbf{g}$, a design intention $\mathcal{T}$, from the metadata JSON file. There are 892 unique tags in total, which specify a range of design themes (e.g., "education", "healthcare", "arts"), and a variety of design styles and usage contexts. We end up with a dataset $\mathcal{D}_{\text{LADEREIN}} = \{(\mathbf{X}, \mathcal{T}, \mathbf{I}, l, \mathbf{g})\}$, which consists of 13,635 text-design pairs $(\mathbf{X}, \mathcal{T})$. We partition the dataset into 11,589, 1,364, 682 samples for train-

Table 1. Dataset comparison.

Datasets	Size	Design		Text	
		Layered?	Attributes	Design Intention?	Avg. Length
RICO2.5K	2,412	No	Type, Bounding box	No	53.00±17.12
Web-design	∼40,000	No	Type, Bounding box	No	29.21±14.08
Crello	23,182	Yes	Type, Bounding box, Image, Font Family, Font Size, Text content, Color, Rotation	Yes, Synthetic	37.52±25.21
LADEREIN (Ours)	13,635	Yes	Type, Bounding box, Image, Font Family, Font Size, Text content, Color, Rotation, Bold, Italic, Underline, Auto Shape Info, etc.	Yes, **Real**	**81.79**±16.89

ing, test, and validation splits, respectively. Figure 2 shows one sample in our dataset, and some designs of different themes.

Comparison to Prior Datasets. Table 1 shows a detailed comparison of our dataset to some existing public graphic design datasets *with textual descriptions*, including RICO2.5K [17], Web-design [28] and Crello [29]. RICO2.5K (a dataset of mobile UIs) and Web-design (a dataset of web banners) mainly contain layout information (i.e., element types and bounding box coordinates), lacking other element attributes necessary to render complete designs. More importantly, the text descriptions on these datasets are *not* design intentions: RICO2.5K's text annotations are for element layouts, while the text captions in Web-design primarily describe the products being advertised.

Similar to our dataset, Crello [29] encompasses layered graphic designs with complete element attributes. The original Crello dataset comes without text descriptions. Later works on text-to-design generation [14, 15] extends it with GPT-generated textual design intentions. However, these synthetic intentions are short and, more importantly, there is inevitably a gap between synthetic and real intentions. In contrast, the intentions in our dataset are real, and significantly more complex with an average text length of 81.79 (vs. 37.52 for Crello).

We also analyze how well text annotations in both datasets cover fundamental factors that are critical for expressing design intentions, including design purpose, feeling to convey and expected audience action. For this end, we prompt a large language model (GPT) to extract text segments relevant to each factor from the design intentions of Crello and LADEREIN. Table 2 reports the average lengths and null rates of extracted texts for each factor on both datasets. The average text lengths of LADEREIN are longer than those of Crello across all the factors, suggesting that LADEREIN contains richer text information that is imperative for design intention, compared to Crello. Furthermore, For all the factors, LADEREIN exhibits lower null rates than Crello, whereas about half of the design intentions in Crello do not contain any information about feeling to convey and expected audience action.

Table 2. Design intention comparison between Crello and LADEREIN in terms of three factors: design purpose, feeling to convey, expected audience action.

Factor	Avg. Text Length↑		Null Rate↓	
	Crello	LADEREIN	Crello	LADEREIN
Purpose	11.93	**12.01**	≈ 0	≈ 0
Feeling	3.196	**4.010**	42.63%	**11.95%**
Action	3.509	**7.664**	55.91%	**11.73%**

Figure 3 show some example text-design pairs sampled from LADEREIN and Crello under three themes: "winter holiday", "business", "back to school". It is evident that, as compared to Crello, LADEREIN offers more detailed, diverse and expressive descriptions of design intention, which are better in line with how human designers would express their intent in practice. In contrast, the textual intentions in Crello are short and simple. In addition, we note that LADEREIN's intentions are generally vague and abstract, whereas those from Crello usually mention specific contents that appear in the designs (e.g., "a cute fox wearing eyeglasses" in the bottom row of Figure 3).

4. DesignCLIP

To reliably measure the consistency between graphic designs and textual design intentions, we develop a metric, *DesignCLIP* or DCLIP for short. Our DCLIP follows the standard contrastive training scheme of the CLIP to jointly train a text encoder and a design encoder that project textual intentions and graphic designs into a joint embedding space, and computes cosine similarity between intention and design embeddings as an alignment score.

Design Image Encoder. Our DCLIP relies on a design image encoder, which maps a design in pixel space to an

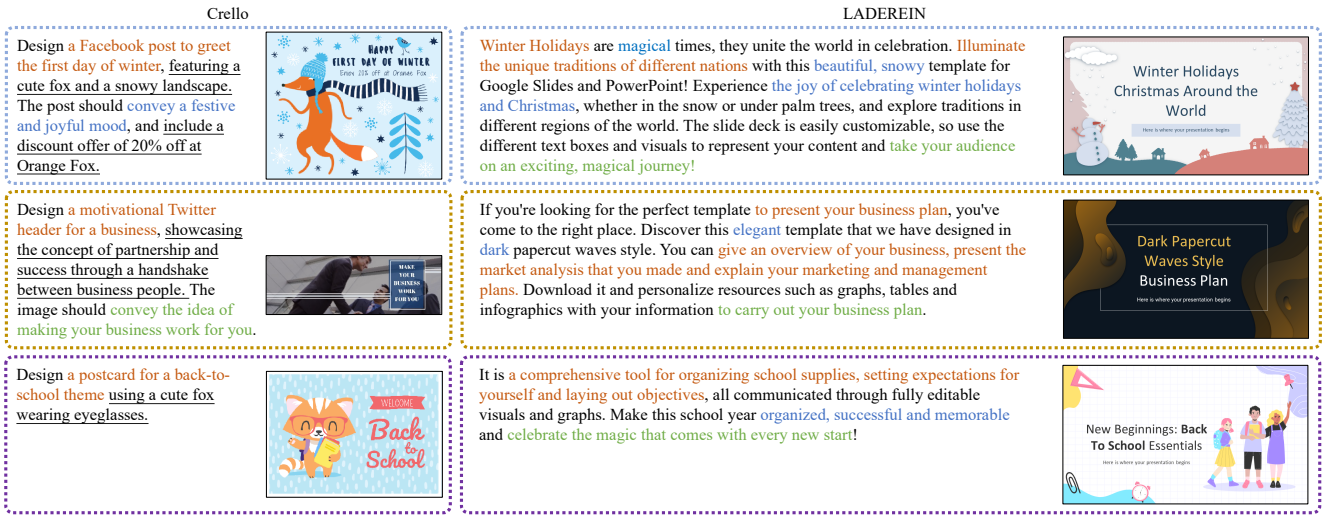

Figure 3. Comparison of textual design intentions from Crello and LADEREIN under three themes: "winter holiday" (top row), "business" (middle row), "back to school" (bottom row). We highlight some fundamental elements in a design intention, including the design purpose (orange), feelings to convey (blue), and actions that audiences are expected to take (green). The parts of Crello's intentions that refer to specific design contents are underlined.

embedding. We directly use the pretrained CLIP image encoder for the design image encoder, which is fine-tuned with contrastive training on paired intention-design data.

Text Encoder. Considering that design intentions in our dataset are long and complex, we adopt the Long-CLIP text encoder [32] as our text encoder due to its capabilities to process long text inputs. Our text encoder is initialized from the Long-CLIP's pretrained weights, and fine-tuned on intention-design pairs using contrastive training.

5. DesignDiff

To generate designs that can be edited at the level of elements, we build a simple diffusion model, *DesignDiff*, that generates design sequences.

Design Sequence. Following prior work on layout generation [8, 16] and graphic design generation [30], we serialize a design as a sequence of discrete tokens. Specifically, a graphic design element $\mathbf{e}$ is represented as a set of attributes:

$$(a_{\mathrm{c}}, a_{\mathrm{clr}}, a_{\mathrm{img}}, a_{\mathrm{ff}}, a_{\mathrm{fs}}, a_{\mathrm{shp}}, a_{\mathrm{bkg}}, a_{\mathrm{lyr}}, a_{\mathrm{x}}, a_{\mathrm{y}}, a_{\mathrm{w}}, a_{\mathrm{h}}). \quad (1)$$

The element of category a_{c} at layer a_{lyr} has a bounding box with top-left position $(a_{\mathrm{x}}, a_{\mathrm{y}})$, width a_{w} and height a_{h}. a_{clr} denotes the element color, and a_{ff} and a_{fs} represent font family and color (exclusive for text elements). a_{shp} is the shape type (exclusive for shape elements). a_{img} represents the image (exclusive for image elements), and a_{bkg} represents the background image (exclusive for background elements). For a_{img} and a_{bkg}, we encode the image and background image compactly as low-dimensional continuous image embeddings extracted by the pretrained CLIP image encoder. After discretizing the continuous attributes including $a_{\mathrm{x}}, a_{\mathrm{y}}, a_{\mathrm{w}}, a_{\mathrm{h}}, a_{\mathrm{clr}}, a_{\mathrm{fs}}, a_{\mathrm{img}}, a_{\mathrm{bkg}}$ into bins using the k-means clustering, we turn each element into a sequence of attribute tokens, and concatenate them into a single design sequence. To facilitate mini-batch training, we fix element sequence length to the total number of attributes defined in Equation 1, and fill the missing attributes of an element (e.g., image attribute for a text element) with a `[NULL]` token.

Model. Specifically, we adapt the discrete diffusion model of [12], which is originally proposed for layout generation, to design generation by training the model on sequences of discrete attribute tokens. For conditioning on design intentions, we encode the text inputs using the Long-CLIP text encoder and incorporate the output text embeddings into the Transformer-based denoising network through cross-attention. The Long-CLIP text encoder here is fine-tuned on our dataset, and is frozen during the training of the DesignDiff. For fine-tuning the Long-CLIP text encoder, we first train an autoencoder with a Transformer-based design sequence encoder, and then fine-tune the Long-CLIP text encoder and the design sequence encoder using CLIP-like contrastive learning (as in Section 4).

Design Rendering. To convert a generated design sequence into the final design, we retrieve images for image elements from our dataset using the predicted image embeddings, and prompt GPT-4o [19] to generate text contents for text elements based on the input design intention and their predicted attributes. Finally, the retrieved images and generated texts are composed into a design, based on the attributes in the design sequence.

6. Experiment

6.1. Implementation Details

For *DCLIP*, we fine-tune Long-CLIP using AdamW optimizer with a learning rate of 5×10^{-7}. For *DesignDiff*, we use a Transformer encoder with 4 blocks, 8 attention heads, embedding dimensionality 512, and hidden dimensionality 2048. We use the training hyperparameters from [12]. Fine-tuning the *SDXL* [20] model for the sake of comparison uses LoRA [11] with a learning rate set of 8×10^{-7}.

6.2. Evaluation of DesignCLIP

We evaluate the effectiveness of DesignCLIP (DCLIP) in connecting textual design intentions and graphic designs under two settings: cross-modal retrieval and human-metric agreement.

Cross-modal Retrieval. We consider training DCLIP on three different datasets: Crello (with synthetic short intentions), LADEREIN (with real long intentions), and a mixture of Crello and LADEREIN, and compare the performance of the three variants and the pretrained Long-CLIP for text-to-image and image-to-text retrieval tasks on the Crello and LADEREIN test sets.

We report top-1 accuracy results in Table 3. All the variants of DCLIP outperform Long-CLIP across the two test datasets, suggesting the usefulness of fine-tuning on a paired intention-design dataset. Furthermore, we find that training on one dataset has limited generalizability to the other dataset. The variant trained on the mixture dataset can yield promising performance on both datasets, indicating that it has good abilities to align both short and long intentions with graphic designs. Therefore, we use DCLIP trained on the mixture of Crello and LADEREIN as our intention-design alignment metric, and use it in subsequent evaluations.

Table 3. Top-1 accuracy of text-image retrieval on Crello and LADEREIN with Long-CLIP and DCLIP. I2T denotes image-to-text retrieval, and T2I denotes text-to-image retrieval. The best and second best results are in bold and underlined, respectively.

Method	Crello		LADEREIN	
	I2T	T2I	I2T	T2I
Long-CLIP	38.27	45.43	44.39	70.51
DCLIP (Crello)	**51.45**	**55.24**	57.01	72.78
DCLIP (LADEREIN)	40.29	46.61	**80.19**	**82.83**
DCLIP (Mixture)	<u>51.24</u>	<u>50.44</u>	<u>77.40</u>	<u>77.84</u>

Human-metric Alignment. We also evaluate how well our DCLIP aligns with human judgments. To this end, we first construct a test dataset of 4,592 triplets $\{(y, x, \tilde{x})\}$ using the LADEREIN test set, where y is a design intention and x is its corresponding design. $\tilde{x}$ is a randomly selected design that has the same scheme as x. The test set serves as a proxy for human judgments that x aligns better with y than $\tilde{x}$. The test set includes 500 "challenging" triplets where x and $\tilde{x}$ are similar (i.e., close in the embedding space of the image encoder of our DCLIP). On this test set, we compare DCLIP with GPT-4V that is commonly used for quantitative evaluation in the recent text-to-design literature [14, 15]. For this experiment, GPT-4V is asked to give a score from 1 to 10, by being prompted with the "content relevance and effectiveness" grading criteria of the "quality assurance prompt" provided in COLE [15]. Given a design and a design intention, we prompt GPT-4V to evaluate the their alignment and give a score on a scale of 1 to 10. For a candidate metric (DCLIP or GPT-4V), let $s(\cdot, \cdot)$ be the alignment score under the metric. Given a triplet $(y, x, \tilde{x})$, the metric decides that x is more aligned with y, only if $s(y, x) > s(y, \tilde{x})$.

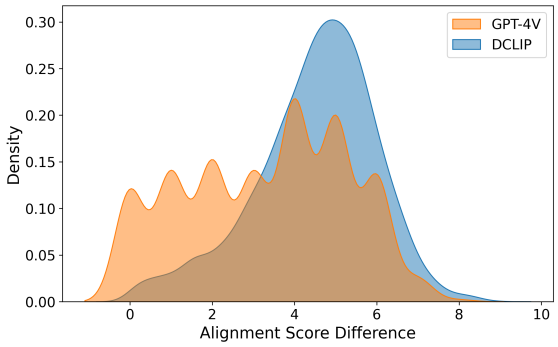

Figure 4. Distributions of alignment score differences for GPT-4V and DCLIP.

DCLIP achieves 99.39% accuracy in predicting human judgments on the test set, while the accuracy of GPT-4V is 87.51%. This suggests that our DCLIP is more consistent with human judgments than GPT-4V. On the challenging cases, our DCLIP still achieves strong performance (95.60% accuracy), while the performance of GPT-4V greatly degrades (66.00% accuracy). This implies that our DCLIP can better discriminate between similar designs.

In Figure 4, we plot the distributions of alignment score differences $s(y, x) - s(y, \tilde{x})$ for DCLIP and GPT-4V. The distribution of DCLIP is right-skewed and has higher density on significant score margins around 5. This suggest that DCLIP can effectively distinguish aligned designs from unaligned ones with high confidence. In contrast, the distribution of GPT-4V is left-skewed and has higher density on low score margins than DCLIP. This suggests that GPT-4V struggles to clearly differentiate between aligned and unaligned designs.

6.3. Evaluation of Our Dataset

An important characteristic of our dataset, relative to existing similar datasets, is the inclusion of real and detailed

design intention descriptions. Prior work on text-to-design generation uses GPT-generated intentions on Crello [29]. Thus, we evaluate the effectiveness of our dataset by testing the quality and necessity of our real intentions, and investigating if our dataset can help improve the performance of text-to-design models compared to Crello (with synthetic intentions).

Real Intention vs. Synthetic Intention. We first conduct a user study comparing real intentions and synthetic intentions. Our study involves 35 subjects, 12 of whom are professional designers and the remaining ones have little or no design experience. We select 30 designs from our LADEREIN dataset. For each of the selected design, besides its corresponding real intention, we synthesize an intention using the same method and prompts of COLE [15]. The subjects were presented with a design alongside two intentions, one real and one generated, and were asked to select which one better matches the design. The results show that the real intentions are preferred 80.76% of the time, indicating their superiority to the synthetic ones. Moreover, there is an 85.00% probability of the *professional* designers preferring the real intentions, which is higher than 78.55% for the *non-professional* subjects. This indicates that the real intentions can more accurately and comprehensively express underlying design requirements from the designer perspective, which further highlights their benefits.

We further generate intentions for the entire LADEREIN dataset, and train DesignDiff on LADEREIN with the synthetic intentions as *DesignDiff-Syn*. We then compare DesignDiff-Syn with DesignDiff trained on the original LADEREIN with real intentions (*DesignDiff-Real*). DesignDiff-Real outperforms DesignDiff-Syn (FID↓: 1.468 vs. 3.425; DCLIP↑: 18.97 vs. 18.16), suggesting the real intentions are necessary for training models to handle complex and diverse design intentions, and generate high-quality designs.

LADEREIN vs. Crello. To gain an understanding of how LADEREIN and Crello affect text-to-design models, We fine-tune SDXL [20] for text-to-design generation, on the training sets of Crello and LADEREIN, respectively, resulting in two models: *SDXL-Crello*, *SDXL-LADEREIN*. The two models are then evaluated on the Crello and LADEREIN test sets. Since design aspect ratios varies between LADEREIN and Crello, for a comparison, we consider three output resolutions separately: 1024×1024 (common output resolution of SDXL), 960×512 (common resolution in LADEREIN), and 256×512 (common resolution in Crello). For each resolution, we restrict the two models to only output images at the resolution, on which various metrics are computed. The results are reported in Table 4. SDXL-LADEREIN achieves consistent performance improvements, compared to SDXL-Crello, across all the resolutions and metrics on the LADEREIN test set, which contains real intentions. This suggests that LADEREIN is effective in enhancing the model's ability of understanding and following complex, challenging intentions to generate high-quality designs. In addition, we observe that SDXL-LADEREIN's DCLIP score is only slightly lower than that of SDXL-Crello only at the 256×512 resolution on the Crello test set that contains designs from various domains (e.g., poster, blog header and web banner). It implies that the designs of LADEREIN, despite being in PPT domain, span a range of representative design patterns that are applied in different graphic design domains.

Table 4. Quantitative results of fine-tuning SDXL on the Crello and LADEREIN training sets, and testing them on the test splits of the two datasets. For each resolution, the two models are only allowed to generate images of the resolution for computing the quantitative scores.

Resolution	Model	Crello		LADEREIN	
		FID↓	DCLIP↑	FID↓	DCLIP↑
1024×1024	SDXL-Crello	8.655	20.92	13.81	17.45
	SDXL-LADEREIN	**8.139**	**20.97**	**10.49**	**17.97**
960×512	SDXL-Crello	8.803	20.50	11.21	17.13
	SDXL-LADEREIN	**7.143**	**20.66**	**8.487**	**17.89**
256×512	SDXL-Crello	18.76	**19.53**	15.44	16.67
	SDXL-LADEREIN	**7.961**	19.34	**9.203**	**16.88**

6.4. Evaluation of DesignDiff

Baselines. We compare DesignDiff against the following two methods: a text-to-image generation model, *SDXL* [20], a recent cascaded model for text-to-design, *OpenCOLE* [14]. We also train an autoregressive model, *DesignAR*, as an additional baseline, which predicts sequences of element attribute tokens in the same format used by DesignDiff. For fair comparison, all the models are trained on our LADEREIN training split. DesignDiff and DesignAR output design sequences, while SDXL generates images, and OpenCOLE produces a layered design with an image layer and a text layer (with text elements that can be modified individually).

Metrics. We use *FID* and *DCLIP* to evaluate the visual quality and intention-alignment of rendered design images. We additionally reply on GPT-4V to assess the design images across five dimensions as in COLE[15].

Results. As reported in Table 5. DesignDiff outperforms DesignAR, SDXL and OpenCOLE in terms of both FID and DCLIP. For GPT-4V assessment, DesignDiff achieves the best scores in the dimensions "content relevance" and "typography and color", suggesting that DesignDiff has an excellent intention-following ability and that it has strong capabilities to set proper typographic attributes and select the right colors. On the other three dimensions, the performance of DesignDiff slightly lags behind that of Open-

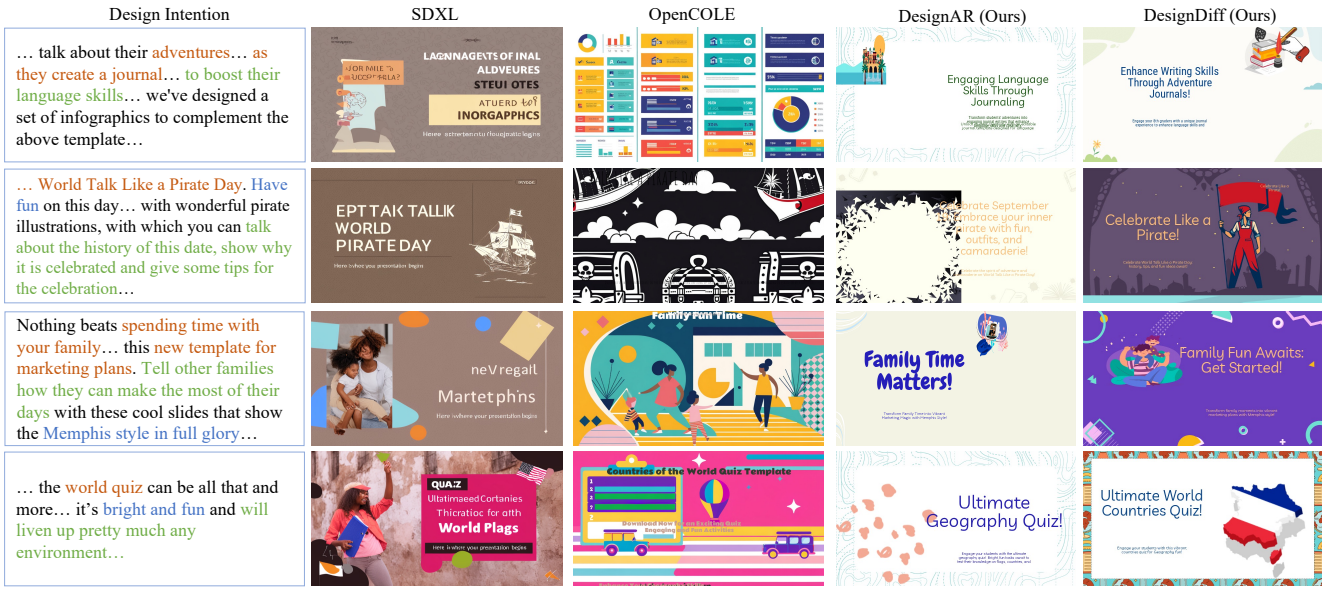

Figure 5. Comparison of designs generated by different models from the input design intentions (on the leftmost). In each intention, we highlight the design purpose (orange), feelings to convey (blue), and expected audience actions (green).

Table 5. Quantitative evaluation of different text-to-design generation models on the LADEREIN test set. GPT-4V rate generated designs for the following dimensions: (i) design and layout, (ii) content relevance, (iii) typography and color, (iv) graphics and images, (v) innovation. The best and second best results are in bold and underlined, respectively. The scores of real samples (GT) are reported for reference.

Model	FID↓	DCLIP↑	GPT-4V Rating↑				
			(i)	(ii)	(iii)	(iv)	(v)
SDXL	8.487	17.89	4.997	5.595	4.679	5.725	5.600
OpenCOLE	5.641	18.44	**6.916**	<u>7.497</u>	<u>6.730</u>	**7.176**	**6.567**
DesignAR (Ours)	<u>2.778</u>	<u>18.81</u>	6.239	7.136	6.444	5.650	6.067
DesignDiff (Ours)	**1.468**	**18.97**	<u>6.796</u>	**7.628**	**6.882**	<u>6.255</u>	<u>6.246</u>
GT	-	19.28	7.542	8.273	7.573	7.284	6.814

COLE, but the gaps are small. Notably, DesginDiff is much simpler than OpenCOLE and, importantly, only text elements in the outputs of OpenCOLE can be modified, whereas DesignDiff's outputs support editing of *all* elements.

Figure 5 visually compares the generated designs by different models. SDXL, which directly generates designs in pixel space, suffers from some obvious issues, such as texts of low legibility and misaligned elements (partially due to its inability to accurately capture spatial relationships between elements). OpenCOLE fails to understand and express abstract concepts in input intentions, generating designs that mainly focus on depicting concrete objects mentioned in the input text. For example, in the first row, OpenCOLE fills the generated design with only a group of charts since the input text contains "a set of infographics", which can not convey other important concepts, such as "language skills" . Moreover, the text placement and colors of OpenCOLE are sometimes inappropriate, which degrades text readability (e.g., 2nd and 4th rows). In the results of DesignAR, undesired overlap between elements often occurs (e.g., 1st and 2nd rows), and the graphic elements are not related to the input design (e.g., 3rd row). In contrast to the other models, DesignDiff is capable of generating high-quality and diverse designs, while enjoying great alignment with the input intentions. For example, in the 1st row, the result of DesignDiff uses an illustration containing books, a speech balloon and a writing hand, which emphasizes the concept "language". In the 4th row, DesignDiff puts a large pure white rectangle against a colorful background with repetitive patterns, to make the entire design look "bright" and "fun".

7. Conclusion

In this paper, we introduce a layered graphic design dataset with real and complex textual design intentions. We also develop a quantitative metric that can reliably measure intention-design alignment, and build a simple yet effective diffusion-based baseline model for text-to-design generation. We hope that our dataset, along with the evaluation metric and the baseline model, can promote research in the area of graphic design generation and inspire future work in building more capable text-to-design models.

References

[1] Steve Canny, karthik1024, alexeybaj, Gunnlaugur Thor Briem, Jozef Leskovec, Kevin Gu, and Matthew Hoopes. *scanny/python-pptx*. 2024. 3

[2] Shang Chai, Liansheng Zhuang, and Fengying Yan. Layoutdm: Transformer-based diffusion model for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18349–18358, 2023. 1

[3] Jingye Chen, Zhaowen Wang, Nanxuan Zhao, Li Zhang, Difan Liu, Jimei Yang, and Qifeng Chen. Rethinking layered graphic design generation with a top-down approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16861–16870, 2025. 2

[4] SiXiang Chen, Jianyu Lai, Jialin Gao, Tian Ye, Haoyu Chen, Hengyu Shi, Shitong Shao, Yunlong Lin, Song Fei, Zhaohu Xing, et al. Postercraft: Rethinking high-quality aesthetic poster generation in a unified framework. *arXiv preprint arXiv:2506.10741*, 2025. 2

[5] Yutao Cheng, Zhao Zhang, Maoke Yang, Hui Nie, Chunyuan Li, Xinglong Wu, and Jie Shao. Graphic design with large multimodal model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2473–2481, 2025. 2

[6] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854, 2017. 2

[7] Yifan Gao, Jinpeng Lin, Min Zhou, Chuanbin Liu, Hongtao Xie, Tiezheng Ge, and Yuning Jiang. Textpainter: Multimodal text image generation with visual-harmony and text-comprehension for poster design. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7236–7246, 2023. 1

[8] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021. 5

[9] Daichi Haraguchi, Naoto Inoue, Wataru Shimoda, Hayato Mitani, Seiichi Uchida, and Kota Yamaguchi. Can gpts evaluate graphic design based on design principles? In *SIGGRAPH Asia 2024 Technical Communications*, pages 1–4. 2024. 3

[10] Hsiao Yuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6018–6026, 2023. 1, 2

[11] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. 6

[12] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023. 5, 6

[13] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14287–14296, 2023. 1

[14] Naoto Inoue, Kento Masui, Wataru Shimoda, and Kota Yamaguchi. Opencole: Towards reproducible automatic graphic design generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8131–8135, 2024. 1, 2, 3, 4, 6, 7

[15] Peidong Jia, Chenxuan Li, Yuhui Yuan, Zeyu Liu, Yichao Shen, Bohan Chen, Xingru Chen, Yinglin Zheng, Dong Chen, Ji Li, et al. Cole: A hierarchical generation framework for multi-layered and editable graphic design. *arXiv preprint arXiv:2311.16974*, 2023. 1, 2, 3, 4, 6, 7

[16] Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutformer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18403–18412, 2023. 5

[17] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Weijiang Xu, Ting Liu, Jian-Guang Lou, and Dongmei Zhang. A parse-then-place approach for generating graphic layouts from textual descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23622–23631, 2023. 4

[18] Jiawei Lin, Shizhao Sun, Danqing Huang, Ting Liu, Ji Li, and Jiang Bian. From elements to design: A layered approach for automatic graphic design composition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8128–8137, 2025. 1, 2

[19] OpenAI Canva. Canvagpt: Effortlessly design anything: presentations, logos, social media posts and more. https://www.canva.com/canvagpt/, 2023. Accessed: September 2023. 5

[20] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 6, 7

[21] Yifan Pu, Yiming Zhao, Zhicong Tang, Ruihong Yin, Haoxing Ye, Yuhui Yuan, Dong Chen, Jianmin Bao, Sirui Zhang, Yanbin Wang, et al. Art: Anonymous region transformer for variable multi-layer transparent image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7952–7962, 2025. 2

[22] Yadong Qu, Shancheng Fang, Yuxin Wang, Xiaorui Wang, Zhineng Chen, Hongtao Xie, and Yongdong Zhang. Igd: Instructional graphic design with multimodal layer generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18218–18228, 2025. 2

[23] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever.

Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 1

[24] 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. 1

[25] Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of langauge model to contents-aware layout generation. *arXiv preprint arXiv:2404.00995*, 2024. 1

[26] Tomoyuki Suzuki, Kang-Jun Liu, Naoto Inoue, and Kota Yamaguchi. Layerd: Decomposing raster graphic designs into layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17783–17792, 2025. 2

[27] Zhendong Wang, Jianmin Bao, Shuyang Gu, Dong Chen, Wengang Zhou, and Houqiang Li. Designdiffusion: High-quality text-to-design image generation with diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20906–20915, 2025. 2

[28] Haohan Weng, Danqing Huang, Yu Qiao, Zheng Hu, Chin-Yew Lin, Tong Zhang, and C. L. Philip Chen. Desigen: A pipeline for controllable design template generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12721–12732, 2024. 2, 4

[29] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. 2, 4, 7

[30] Bo Yang and Ying Cao. Order matters: Learning element ordering for graphic design generation. *ACM Transactions on Graphics (TOG)*, 44(4):1–16, 2025. 2, 5

[31] Ning Yu, Chia-Chih Chen, Zeyuan Chen, Rui Meng, Gang Wu, Paul Josel, Juan Carlos Niebles, Caiming Xiong, and Ran Xu. Layoutdetr: detection transformer is a good multimodal layout designer. In *European Conference on Computer Vision*, pages 169–187. Springer, 2024. 1, 2

[32] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*, pages 310–325. Springer, 2024. 5

[33] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 2

[34] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 1015–1022. IEEE, 2019.

[35] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. *arXiv preprint arXiv:2205.00303*, 2022. 2