

Grounded Generation of Embellished Bar Chart Ensuring Chart Integrity

Seon Gyeom Kim^{*†}
KAIST

Jae Young Choi^{*‡}
KAIST
Jihyung Kil
Adobe Research

Phillip Y. Lee
KAIST
Eunyee Koh
Adobe Research

Jaeryung Chung
KAIST
Tak Yeon Lee[§]
KAIST

Ryan Rossi
Adobe Research

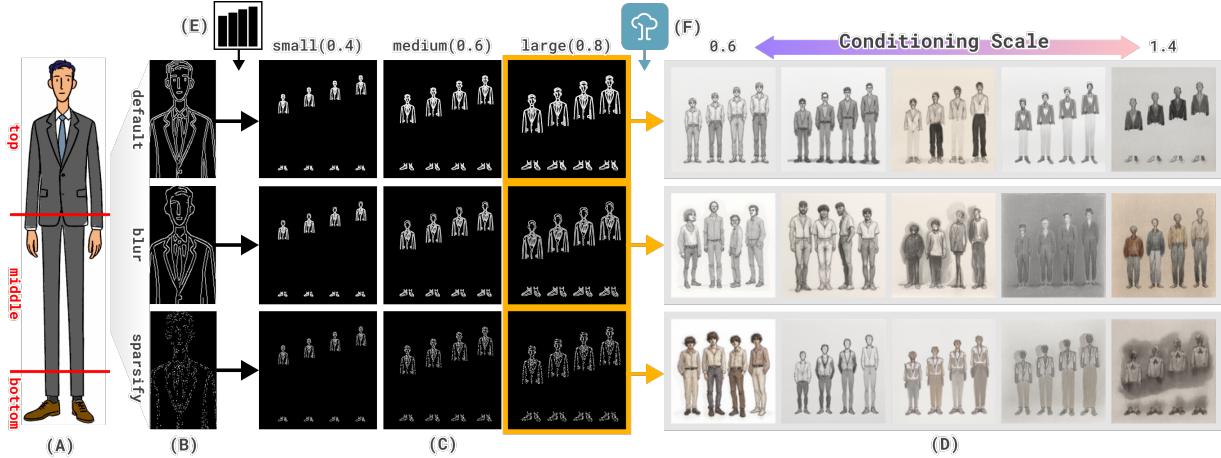


Figure 1: Overview of our embellished chart generation pipeline. From (A) an object image having top / middle / bottom parts, it can generate (B) three types of Canny edges (default / blur / sparsify). By arranging the edges for (E) the original bar chart, it can generate (C) small-to-large scale images for controlling the model. Lastly, (F) the ControlNet model can generate embellished bar charts with a degree of conditioning scales.

ABSTRACT

Grounded image generation enables precise spatial control over pre-trained diffusion models, making it possible to use chart images as visual guides during the image generation process. This paper presents a novel approach that generates cohesive and natural illustrations of vertical bar charts by integrating real-world object images as visual embellishments. The proposed pipeline takes an object image and a reference bar chart as input and produces an embellished bar chart that follows the structure of the input chart. To preserve chart integrity by maintaining the count, position, size, and order of data values, we introduce a strategy that anchors the top and bottom parts of the object image to the top and bottom of each bar while allowing the middle section to be filled by the generation model. We demonstrate the efficacy of the pipeline through the generation of 4,725 chart images followed by evaluation based on three integrity metrics. The results show that generation success rate is affected by various factors. Finally, we discuss future directions for generalization and better usability of our pipeline, and limitations of evaluation used in our approach.

Index Terms: Chart Embellishment, Grounded Image Generation.

1 INTRODUCTION

Basic charts are commonly used to represent data through abstract and geometric marks [10]. Their structured design makes them effective tools for conveying data, trends, and analytical insights.

As a result, charts are frequently employed in visual materials intended for diverse audiences, such as those found in journalism or educational content. In order to improve the communicative effectiveness of charts, prior research has examined methods for enhancing visual engagement [2]. These methods, often referred to as chart embellishments or chart junk, aim to capture attention without compromising clarity or interpretability [4, 6]. A representative approach involves the use of semantic illustrations that reflect the subject matter of the data or the intended message. Empirical studies have demonstrated that such illustrations can increase engagement without introducing significant interpretive drawbacks [9, 5]. To address this challenge computationally, systems such as ChartEditor [29] and ChartSpark [27] apply generative models to automatically produce semantic embellishments. These systems preserve the integrity of the original data by using chart marks as spatial references or masks to guide the placement of generated elements.

However, our study employs a different strategy which guides image generation with visual inputs for matching the position and dimension of a particular object image to chart marks. This is made possible by facilitating grounded image generation models, which enable controlling output images using visual information such as edge maps, depth maps, or bounding boxes during the generation process [17, 31]. In addition, controllability parameters allow users to specify how closely the model should adhere to the provided input. This generation method produces visually coherent outputs without the need to separately compose icons, backgrounds, or other embellishments.

To explore the applicability of this approach, Section 3 presents our embellished bar chart generation pipeline using the grounded image generation model “ControlNet checkpoint for Flux.1-dev” [3, 28]. We used a Canny edge image as the controlling input, and the image generation model is combined with the “Sketch Smudge” LoRA to consistently produce clear 2D chart sketches with simple backgrounds rather than photorealistic images. Sec-

^{*}Equal contributions.

[†]e-mail: ksg_0320@kaist.ac.kr

[‡]e-mail: jaeyoungchoi@kaist.ac.kr

[§]e-mail: takyeonlee@kaist.ac.kr

tion 4 explains which variables and their values were chosen for exploratory generation of embellished bar chart using the pipeline. Subsequently, we evaluated these generated outputs in terms of chart integrity and identified embellished bar charts that preserve it in Section 5. Lastly, we discussed the insights and considerations on creating embellished bar charts using grounded generation models and the future research directions in Section 6.

2 RELATED WORK

Grounded Image Generation. Recent progress in text-to-image (T2I) models increasingly explores ways to enhance spatial controllability not only through text prompts but also via image inputs. These inputs vary in form and provide different types of controllability. For example, skeletons can guide object pose, while depth maps can influence both the silhouette and layout of a scene. To enable such controls, T2I models must be trained with specific input types, which depend on the base model. Architectures like GLIGEN [17] and InstanceDiffusion [26] enable bounding boxes paired with corresponding textual captions to guide image generation. On the other hand, ControlNet [31] and UniControl [20] provide general-purpose conditioning capabilities, leveraging spatial inputs like segmentation maps, Canny edges, or skeletons. However, the availability of such controllable checkpoints is constrained by the architecture, and the conditioning effectiveness may vary depending on the training dataset and parameters. As a result, publicly available models support a limited number of input types, for instance, the FLUX.1-dev model currently provides checkpoints only for Canny edge, HED, and depth map inputs.

Embellished Chart. A variety of terms, such as chart junk and embellishment [16, 23, 2], pictorial charts [30, 22], infographics [8], and anthropographics [19] have been widely used to describe charts that adopt a rich and visually elaborate style. While data visualization research has traditionally focused on optimizing data perception for analytic tasks, stylized charts have been criticized for compromising chart integrity. However, recent studies have challenged this negative aspect of stylized charts, showing that they can serve communicative functions beyond analytic precision. In particular, research has shown that stylized charts can effectively provoke emotions [1, 14] and enhance long-term memory retention [2, 24]. These effects become especially pronounced in narrative integrated data visualizations, where the author’s communicative intent is strongly reflected. In such contexts, emotions such as engagement, surprise, awe, sadness, or anger play a crucial role, highlighting the communicative and affective role of data visualizations [13, 12, 15]. Building on these findings, some researchers have attempted to classify embellishment strategies along structural and functional dimensions. For example, Chen et al. [7] proposed a design space derived from 361 embellished visualizations, which organizes common techniques according to their roles in supporting three communicative goals: exploring data insights, integrating contextual meaning and emotion, and expressing stylistic aesthetics. Their work also highlights the diverse visual components that can be embellished, such as marks, labels, axes, titles, and layout.

3 PIPELINE FOR CHART EMBELLISHMENT

The pipeline takes two user-provided inputs: an object image (Fig. 1A) and a vertical bar chart (Fig. 1E). The objective is to generate images featuring multiple instances of the object image arranged in a way that their positions and heights correspond to the data in the bar chart. To preserve the integrity of the chart (e.g., the number and heights of data points) during generation, we used grounded image generation supported by ControlNet, a neural network structure that enables precise control over pre-trained diffusion models [31]. In particular, we used Flux.1-dev [3], a state-of-the-art image generation model.

For the input bar chart, we used chart images in SVG format. Using rasterized charts as input introduces issues such as data extraction, which are beyond the scope of this work. Among the chart components, only the bar marks are used to anchor object image segments, and other elements are ignored.

The input object image has a fixed aspect ratio, which makes it challenging to visually represent bars of varying heights without distortion. To address this, we used only the top and bottom parts of the object image, placing them at the corresponding ends of each bar, and left the middle part blank to allow the generation model to fill it in. As shown in Fig. 1C, this ensures that the top and bottom segments align precisely with the original bar positions. Among the various control options (e.g., HED, depth map, Canny edge), we used Canny edge as an input (Fig. 1B). Since Canny edge is a simple edge detection technique and we can convey minimal information while retaining key visual characteristics of the object, this would allow the model greater flexibility in generation.

During the generation, we used the prompt “*{Data Count} {Object Image Name}*” with different heights”, where *Data Count* and *Object Image Name* refer to the variables listed in Section 4. The resulting chart output is a rasterized image (Fig. 1D), composed only of stylized marks without other chart elements.

4 EXPLORATORY GENERATION USING THE PIPELINE

To examine the capability and limitations of the pipeline, we generated three images for each combination of the variables listed in Table 1, producing a total of 4,725 output images.

Category	Values
Object Images	human, bottle, palm tree, castle tower, balloon, cactus, stack of coins
Edge Processing Styles	default, blur, sparsify
Canny Size	small, medium, large
Data Count	3, 4, 5, 6, 7
Conditioning Scale	0.6, 0.8, 1.0, 1.2, 1.4

Table 1: Predefined variables used in bar chart generation.

Object Images have a wide range of visual details that can significantly impact the performance of the pipeline. For example, the human figure (Fig. 1A) has details such as hands or waists in the middle region, which can make it difficult for image models to fill. In contrast, other objects (e.g., palm tree) tend to have repetitive middle sections, making the image generation easier, but their wider top sections can make bars overlap each other. We thus selected seven objects that have a wide range of visual and structural complexity as listed in Table 1.

Edge Processing Styles control the amount of edge information that guides the image generation process. For example, the *blur* technique applies Gaussian blur before detecting edges in the object image. The *sparsify* technique randomly removes 30% of the detected edge lines. As a result, the *blur* and *sparsify* edges in Fig. 1 contain fewer details of the face of the original human figure.

Canny Size Factors. Before applying Canny edge detection, we first resized all object images to a uniform height of 512 pixels, which is the height of the final output image. To explore the impact of image size factors, we set three object size settings, which are *small* (0.4 of the original object), *medium* (0.6), and *large* (0.8).

Data Count. Real-world bar charts have varying numbers of data points. However, including many data points can make the result resemble a dense histogram or an area chart rather than a bar chart due to low resolution. Also, many data points can lead to excessive overlap similar to the case of large object size. Therefore, we predefined seven values that follow a logarithmic trend. Then, we used only a subset taken from the end of this sequence for each generated image. The number of data points ranged from three to seven.

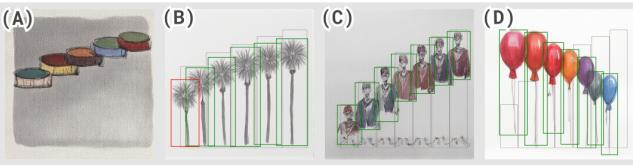


Figure 2: Failure cases occurred in data integrity evaluation. (A) Missed detections due to weak alignment between prompt (stack of coins) and object. (B) Incorrect detection count despite visible objects. (C) Bar bottom y -coordinate misaligned beyond threshold. (D) Inconsistent bar height ranking versus reference bars.

Conditioning Scale. When using the Flux-based ControlNet, a control scale parameter determines how strongly the control signal affects the output. The default value is set to 1.0, and the impact of different weights was difficult to predict. Therefore, we used an arbitrary gap of 0.2 and tested five values ranging from 0.6 to 1.4.

5 FILTERING OUTPUTS BASED ON CHART INTEGRITY

In this section, we filter total 4,725 generated images to identify embellished bar charts which ensure chart integrity. We applied object detection to each image to analyze the location and dimensions of generated objects. The detected objects were then compared to the reference bars in the original chart to determine whether the generated image accurately followed the spatial constraints.

5.1 Object Detection and Matching

In order to detect the objects, OWLv2-base model [18] was utilized, which supports zero-shot text-conditioned object detection. For each image, the corresponding object name (e.g., “balloon”, “palm tree”) was passed to the model as a text query, allowing the model to return bounding boxes for detected objects in the image.

To assess the alignment between the detected bars and reference bars, we matched them using the Hungarian algorithm [11] with a cost matrix $C \in R^{m \times n}$, where each element is defined as:

$$C_{ij} = -\text{IoU}(r_i, d_j) \quad (1)$$

Here, r_i and d_j denote the i -th reference bar and j -th detected bounding box, respectively. The Intersection over Union (IoU), defined as the ratio of intersection area to union area between two boxes, is computed for each pair. The Hungarian algorithm then finds an optimal one-to-one matching by minimizing the negative IoU, effectively maximizing IoU across matched pairs.

5.2 Chart Integrity Evaluation

We examined whether each generated chart faithfully preserved the data by comparing detected bounding boxes with corresponding reference bars. Specifically, we defined three criteria as follows:

- **Match:** whether the detected boxes match the reference bars in both total count and individual positions
- **Align:** whether all detected boxes are correctly aligned in terms of their top and bottom y -coordinates with the reference bars
- **Rank:** whether the rank order of the top y -coordinates of the detected boxes is identical to that of the reference bars

First, in the **match** criteria, we identified 2,573 matching failure cases over total 4,725 images where the detected boxes did not properly match the reference bars. In this metric, accurate object detection is critical for evaluation. Representative failure cases are caused by weak semantic similarity between the query prompt and the generated object appearance, resulting in completely missed detections (Fig. 2A); notably, 1,080 cases showed no detection at all. Additionally, even when objects were detected, mismatches still occurred. Some cases had either fewer or more detected boxes than the expected number of reference bars, despite clear object visibility (Fig. 2B). For the result, this step ensured that 2,152 images were generated with matching the reference bars correctly.

Category	Value	Match (%)	Align (%)	Rank (%)	Success (%)
Object Images	bottle	64.0	35.1	40.1	34.5
	human	75.3	25.3	57.3	25.2
	palm tree	46.7	22.1	32.1	21.2
	balloon	49.3	17.0	30.8	17.0
	cactus	33.8	8.1	15.7	8.1
	castle tower	49.8	6.1	40.4	5.8
Edge Processing Styles	stack of coins	0.0	0.0	0.0	0.0
	default	57.0	28.9	51.1	28.8
	blur	40.9	12.9	24.7	12.7
Canny Size	sparsify	38.7	7.0	17.0	6.5
	large	53.3	23.1	43.7	22.9
	medium	47.3	17.0	32.5	16.6
Data Count	small	36.0	8.6	16.6	8.5
	3	67.3	24.9	38.9	24.4
	4	58.1	23.5	37.5	23.1
	5	47.0	17.7	32.0	17.2
	6	30.8	10.8	25.8	10.7
Conditioning Scale	7	24.6	4.4	20.5	4.4
	0.6	46.7	7.7	17.0	7.4
	0.8	49.4	16.0	29.2	15.8
	1.0	48.6	18.4	34.3	18.1
	1.2	45.2	22.1	38.3	21.8
	1.4	37.9	17.0	35.9	16.8

Table 2: Evaluation results for the generated images. The Match, Align, and Rank columns show the percentage of images meeting each criterion, and Success indicates those satisfying all three.

Next, for 2,152 images which passed **match**, we checked both **align** and **rank** metrics. In the **align**, we measured if the detected boxes accurately represent the original data by comparing the top and bottom y -coordinates of each matched pair (r_i, d_i) . Let

$$e_i^{\text{top}} = |y_{r_i}^{\text{top}} - y_{d_i}^{\text{top}}|, \quad e_i^{\text{bot}} = |y_{r_i}^{\text{bot}} - y_{d_i}^{\text{bot}}| \quad (2)$$

be the errors for the top and bottom edges of the bar, where d_i and r_i denote i -th detected box and reference bar, respectively. In order to keep the permitted error independent of each bar’s height, we applied the fixed absolute displacement tolerance of 5% of the total chart height. We chose the value 0.05 since it is a commonly used threshold such as in Tuft’s Lie Factor [25] and chart-data extraction evaluations [21]. A bar was considered misaligned if either e_i^{top} or e_i^{bot} exceeded this tolerance. The images were labeled as **align** failures if at least one of their bars was misaligned, leading to 1,384 failures identified through the evaluation (Fig. 2C).

Lastly, for the **rank**, we examined whether the detected boxes preserved the intended visual trend in matched images. This involved checking the rank ordering of detected box heights based on their top y -coordinates against the corresponding reference bar ranks. This evaluation revealed 690 **rank** failure cases over 2,152 matched images, with an example shown in Fig. 2D.

5.3 Evaluation Result

Table 2 shows detailed evaluation results for generated images across experimental conditions. Of the total 4,725 images, 755 were successfully identified as bar charts. In our case, only 13 images passed the **align** criterion but failed the **rank** criterion.

Across object images, success rate varied considerably, with *bottle* achieving the highest ratio of validated charts (34.5%), followed by *human* (25.2%), *palm tree* (21.2%), and *balloon* (17.0%). Lower success rates are observed for *cactus* (8.1%), and *castle tower* (5.8%), whereas *stack of coins* recorded no successes at all.

For edge processing styles, the *default* (28.8%) outperformed *blur* (12.7%) and *sparsify* (6.5%). And using *large* Canny size yielded the highest successes (22.9%), with performance dropping as the Canny size became smaller (*medium*: 16.6%, *small*: 8.5%).

Finally, our results show a decreasing trend in performance as the data count increases from three to seven. For instance, charts with three bars yielded 24.4% success rate, while those with seven bars achieved only 4.4%. And for conditioning scale, performance improved as the scale increased from 0.6 to 1.2, with the highest success rate observed at scale 1.2 (21.8%). However, performance declined at 1.4 (16.8%).

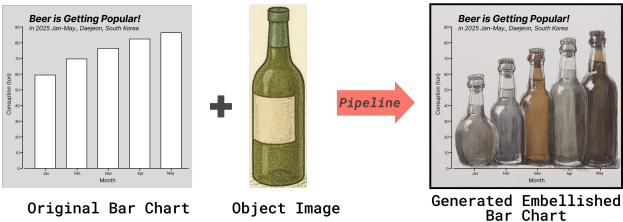


Figure 3: Sample use scenario of our pipeline. Authors can replace bar marks with bottle images while retaining all other chart elements.

6 DISCUSSION

Research Opportunities for the Usability. In our pipeline, chart images are generated as a single, unified image rather than composing individual chart elements using multiple layers. By adopting such end-to-end image generation, our method produces more cohesive and natural outputs. We present an example of a bar chart embellished in such ways in Fig. 3. In this example, the positions and heights of generated bar images are accurately maintained so they could integrate seamlessly with the axes, titles, ticks, and labels from the original chart. Also, they exhibit a consistent visual style as a group but individually distinct, complete with realistic details such as shadows and color blending between overlapping bottles. This level of visual integration would be difficult to achieve by just scaling and placing pre-generated assets.

However, an important note about Fig. 3 is that we did not explicitly specify details such as the shape or color of each bottle. While it is promising that the model can implicitly handle these undefined characteristics, usability challenges remain for practical adoption of our pipeline. For instance, future work could examine whether the model reliably responds to element-level prompting (e.g., “Render the third bottle from the left in orange”) or whether editing after generation of individual elements is readily feasible. Regarding this, implicit integration with high-level semantics like dataset context or author intention may reduce the manual workload required for such tasks. One example would be a chart where the x-axis represents age ranges and the chart highlights gaps in y-axis values between different age groups. In this case, the pipeline could automatically generate children images for younger age groups and elders for older age groups without explicit instruction. This would reduce the effort needed to prepare appropriate images for each bar, allowing authors to easily create contextually meaningful output.

Generalizability of the Pipeline from Design Perspectives. We provided visual cues at data positions while leaving other areas open, ensuring chart integrity and supporting diverse generations. However, designing effective visual cues remains challenging as generative models may interpret the same cues differently. For example, we selected “Sketch Smudge” as it generates the desired objects with simple backgrounds, whereas the default or other LoRAs often treated visual cues as part of a complex scene, adding auxiliary objects and detailed backgrounds. Thus, we believe integrating selective removal of such elements or focusing generation solely on the target objects could enable easier exploration of diverse styles.

Similar to the LLMs’ temperature for output diversity, image generation models have a conditioning scale that defines how strictly they adhere to the provided visual guidance, resulting in varied interpretations of visual cues. For example, as shown in Fig. 1C, the input to the model would have legs that are unnaturally short or long relative to the upper body and feet depending on the height of the bars. When the conditioning scale is large, the model may faithfully draw the given upper body and feet but fail to connect them naturally. This suggests that finding a conditioning scale for expressive and faithful output is critical, but it seems challenging because the optimal scale may vary depending on the type of object.

In addition, although we explored only vertical bar charts, this approach could extend to other chart types by designing visual cues

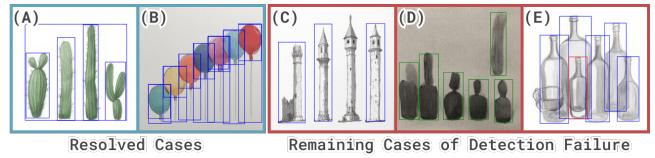


Figure 4: Resolved and remaining detection failure cases. (A) Boxes over 60% of chart width. (B) Partial boxes inside larger boxes. (C) Missed valid objects. (D) Unwanted detections. (E) Removed valid objects due to overlap with retained boxes.

and blank segments suited to their structures. For example, horizontal bar charts may use flat objects segmented left and right instead of top and bottom. For scatterplots, selecting standalone objects placed at individual data points could reduce unintended visual connections. Line charts may require stronger cues for accurate point placement and coherent visual links. Area charts, as an extension of line charts, would need cues that clearly indicate which regions should be filled, possibly using object density or scene composition.

Limitations of Detection-Based Evaluation. Since the width of a bar in the chart cannot exceed half of the image width, we removed bounding boxes with widths exceeding 60% of the image width to account for detection noise (Fig. 4A). To address cases where multiple parts of the same object were detected separately, we examined pairwise containment and discarded boxes more than 90% contained within another, retaining only the larger one (Fig. 4B).

Despite these filtering procedures, semantically correct assets were still often missed when their detection scores fell below the confidence threshold (Fig. 4C). The detector also returned spurious boxes around visually similar but undesired objects (Fig. 4D). Moreover, in crowded scenes where many objects overlap, multiple objects may be merged into one bounding box, resulting in fewer detected objects than actually present (Fig. 4E).

These issues highlight challenges in using object detection models to evaluate generated chart images. Limitations in the detection pipeline can lead to inaccuracies in downstream assessments, introducing the need for careful object detection setup to ensure chart integrity and to minimize misinterpretations from detection failures.

7 CONCLUSION

This study presents a generative pipeline that transforms vertical bar charts into embellished bar charts using real-world object images for visual guidance. To preserve the data-related structure of the input chart, we proposed a strategy that anchors object segments at key positions to leverage the precise spatial controlling ability of the grounded image generation model. We generated 4,725 images and evaluated them using three data integrity metrics based on detection results. Our findings show that chart integrity can be maintained under specific parameter settings, and the visual factors can affect the success rate of generated outputs. Further, our pipeline revealed several limitations, suggesting directions for future research. Specifically, end-to-end image generation in our pipeline offers cohesive and visually integrated outputs, but future work is needed to improve element-level control and explore semantic-aware generation that can automatically align with data context or author intention. The generalizability of our approach across chart types and visual styles depends on designing appropriate visual cues and carefully tuning the conditioning scale, which controls the model’s adherence to input guidance. Moreover, limitations in our detection-based evaluation highlight the need for more reliable object detection strategies to ensure accurate assessment of chart integrity.

SUPPLEMENTAL MATERIALS

The supplemental materials¹ include the 4,725 generated images from Section 4, as well as versions of each image with detected boxes overlaid according to the detection step outlined in Section 5.

¹available at <https://groundedchartgeneration.github.io>

REFERENCES

- [1] T. Andry, C. Hurter, F. Lambotte, P. Fastrez, and A. Telea. Interpreting the effect of embellishment on chart visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021. doi: 10.1145/3411764.3445739 [1](#)
- [2] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk? the effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 2573–2582, 2010. doi: 10.1145/1753326.1753716 [1, 2](#)
- [3] Black Forest Labs. Flux.1-dev. https://huggingface.co/black_forest_labs/FLUX.1-dev. Accessed: 2025-05-01. [1, 2](#)
- [4] R. Borgo, A. Abdul-Rahman, F. Mohamed, P. W. Grant, I. Reppa, L. Floridi, and M. Chen. An empirical study on using visual embellishments in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2759–2768, 2012. doi: 10.1109/TVCG.2012.197 [1](#)
- [5] M. A. Borkin, Z. Bylinskii, N. W. Kim, C. M. Bainbridge, C. S. Yeh, D. Borkin, H. Pfister, and A. Oliva. Beyond memorability: Visualization recognition and recall. *IEEE transactions on visualization and computer graphics*, 22(1):519–528, 2015. doi: 10.1109/TVCG.2015.2467732 [1](#)
- [6] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE transactions on visualization and computer graphics*, 19(12):2306–2315, 2013. doi: 10.1109/TVCG.2013.234 [1](#)
- [7] Q. Chen, Z. Liu, C. Wang, X. Lan, Y. Chen, S. Chen, and N. Cao. Vizbelie: A design space of embellishments for data visualization. *arXiv preprint arXiv:2209.03642*, 2022. doi: 10.48550/arXiv.2209.03642 [2](#)
- [8] I. Damyanov and N. Tsankov. The role of infographics for the development of skills for cognitive modeling in education. *International Journal of Emerging Technologies in Learning (iJET)*, 13(1):82–92, 2018. doi: 10.3991/ijet.v13i01.7541 [2](#)
- [9] S. Haroz, R. Kosara, and S. L. Franconeri. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 1191–1200, 2015. doi: 10.1145/2702123.2702275 [1](#)
- [10] N. Holmes. Designer’s guide to creating charts & diagrams. (*No Title*), 1984. [1](#)
- [11] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. [3](#)
- [12] X. Lan, Y. Shi, Y. Wu, X. Jiao, and N. Cao. Kineticcharts: Augmenting affective expressiveness of charts in data stories with animation design. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):933–943, 2021. doi: 10.1109/TVCG.2021.3114775 [2](#)
- [13] X. Lan, Y. Wu, and N. Cao. Affective visualization design: Leveraging the emotional impact of data. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1–11, 2023. doi: 10.1109/TVCG.2023.3327385 [2](#)
- [14] X. Lan, Y. Wu, Q. Chen, and N. Cao. The chart excites me! exploring how data visualization design influences affective arousal. *arXiv preprint arXiv:2211.03296*, 2022. doi: 10.48550/arXiv.2211.03296 [2](#)
- [15] E. Lee-Robbins and E. Adar. Affective learning objectives for communicative visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1–11, 2022. doi: 10.1109/TVCG.2022.3209500 [2](#)
- [16] H. Li and N. Moacdieh. Is “chart junk” useful? an extended examination of visual embellishment. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, pp. 1516–1520. Sage Publications Sage CA: Los Angeles, CA, 2014. doi: 10.1177/1541931214581316 [2](#)
- [17] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22511–22521, 2023. doi: 10.48550/arXiv.2301.07093 [1, 2](#)
- [18] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection, 2023. [3](#)
- [19] L. Morais, Y. Jansen, N. Andrade, and P. Dragicevic. Showing data about people: A design space of anthropographics. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1661–1679, 2020. doi: 10.1109/TVCG.2020.3023013 [2](#)
- [20] C. Qin, S. Zhang, N. Yu, Y. Feng, X. Yang, Y. Zhou, H. Wang, J. C. Niebles, C. Xiong, S. Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147*, 2023. doi: 10.48550/arXiv.2305.11147 [2](#)
- [21] C. Rane, S. M. Subramanya, D. S. Endluri, J. Wu, and C. L. Giles. Chartreader: Automatic parsing of bar-plots. In *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 318–325. IEEE, 2021. doi: 10.1109/IRI51335.2021.00050 [3](#)
- [22] Y. Shi, P. Liu, S. Chen, M. Sun, and N. Cao. Supporting expressive and faithful pictorial visualization design with visual style transfer. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):236–246, 2022. doi: 10.1109/TVCG.2022.3209486 [2](#)
- [23] D. Skau, L. Harrison, and R. Kosara. An evaluation of the impact of visual embellishments in bar charts. In *Computer Graphics Forum*, vol. 34, pp. 221–230. Wiley Online Library, 2015. doi: 10.1111/cgf.12634 [2](#)
- [24] U. H. Syeda, L. South, J. Raynor, L. Panavas, D. Saffo, T. Morriss, C. Dunne, and M. Borkin. More useful junk? replicating the effects of visual embellishment on description and recall of charts. 2023. [2](#)
- [25] E. R. Tufte and P. R. Graves-Morris. *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 1983. [3](#)
- [26] X. Wang, T. Darrell, S. S. Rambhatla, R. Girdhar, and I. Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6232–6242, 2024. doi: 10.1109/CVPR52733.2024.00596 [2](#)
- [27] S. Xiao, S. Huang, Y. Lin, Y. Ye, and W. Zeng. Let the chart spark: Embedding semantic context into chart with text-to-image generative model. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):284–294, 2023. doi: 10.1109/TVCG.2023.3326913 [1](#)
- [28] XLabs-AI. <https://huggingface.co/XLabs-AI/flux-controlnet-canny-diffusers>. Accessed: 2025-05-01. [1](#)
- [29] S. Yan, T. Liu, W. Yang, N. Tang, and Y. Luo. Charteditor: A human-ai paired tool for authoring pictorial charts. *arXiv preprint arXiv:2501.07320*, 2025. doi: 10.48550/arXiv.2501.07320 [1](#)
- [30] J. E. Zhang, N. Sultanum, A. Bezerianos, and F. Chevalier. Dataquilt: Extracting visual elements from images to craft pictorial visualizations. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pp. 1–13, 2020. doi: 10.1145/3313831.3376172 [2](#)
- [31] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023. doi: 10.48550/arXiv.2302.05543 [1, 2](#)