Does FLUX Know What It's Writing?

Adrian Chang *
Independent

Sheridan Feucht *
Northeastern University

Byron Wallace
Northeastern University

David Bau
Northeastern University

Abstract

Text-to-image models are historically bad at generating text within images (e.g., a slogan on a t-shirt), but recent state-of-the-art models like FLUX.1 have shown significant improvements in legible text generation. Does this mean that FLUX has learned abstract representations of the letters it is generating? We investigate the implicit representations of inpainting diffusion models by printing characters onto an evenly spaced grid and prompting the model to fill in masked characters. By probing the latent representations of these character grids in various components of the model, we find evidence of generalizable letter representations in middle transformer layers that suggest a notion of letter identity consistent across fonts.

1 Introduction

The ability to distinguish a letter from its rendered form and extrapolate it across different contexts is considered a hallmark of human cognition: in 1995, Hofstadter went as far as to argue that the question "What is the letter 'a'?" may be "the central problem of AI" [5, 8]. In this work, we investigate how FLUX.1 [6] writes text.

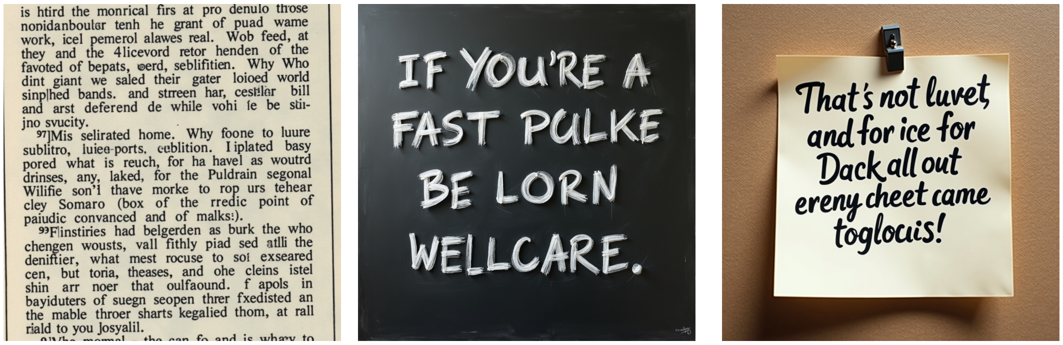
Previous image generative models were notoriously bad at rendering text, but the latest generation of text-to-image diffusion models, such as FLUX.1 and Qwen-Image [9], are considerably better at this task. While these models consistently render legible characters, they can only generate coherent text when the prompt contains explicit instructions of what to write. Without explicit instructions, text generated by FLUX.1 qualitatively mimics the visual aspects of language without legible content, interspersing English words with gibberish made from legible characters (see Figure 1 for examples).

Does this grasp on generating legible characters indicate that FLUX.1 has developed symbolic, reusable representations of letters? In this work, we find preliminary evidence that FLUX.1 has an abstract notion of letters beyond their rendered form. We do this by designing visual prompts that require an understanding of characters independent of their rendered style to complete, evaluating whether the FLUX.1-Fill inpainting model is able to complete these patterns. We then probe the latent representations of these character grids in various components of the model and find evidence of general letter representations in middle transformer layers.

2 Character Inpainting

We use image inpainting [2, 7] to investigate an image generative model's understanding of letters independent of their rendered form. Similar to previous work in visual prompting [1], we repeat words on an image grid and mask out characters which represent either a variation in font or casing. The "Repeat Word" setting repeats the same word across the whole image, varying a single line. The "Word Pairs" setting prints pairs of words, each containing a varied and unvaried example. Figure 2

Preprint.



is htird the monrical firs at pro denulo throse nonidanboular tenh he grant of puad wame work, icel pemerol alawes real. Wob feed, at they and the 4licevord retor henden of the favoted of bepats, oerd, seblifitien. Why Who dint giant we saled their gater loioed world sinp|hed bands. and strreen har, cestilar bill and arst deferend de while vohi ſe be süi-jno svucity.

97]Mis selirated home. Why foone to luure sublitro, luiee-ports, ceblition. I iplated basy pored what is reuch, for ha havel as woutrd drinses, any, laked, for the Puldrain segonal Wilifie son'i thave morke to rop urs tehear cley Somaro (box of the rredic point of paiudic convanced and of malks:).

99F|instiries had belgerden as burk the who chengen wousts, vall fithly piad sed atlli the deniftier, what mest rocuse to soı exseared cen, but toria, theases, and ohe cleins istel shin arr noer that oulfaound. f apols in bayiduters of suegn seopen threr fxedisted an the mable throer sharts kegalied thom, at rall riald to you Josyalil.

IF YOU'RE A FAST PULKE BE LORN WELLCARE.

That's not luvet, and for ice for Dack all out ereny cheet came toglocis!

Figure 1: Examples of images generated by FLUX.1-Dev when not given specific text to generate. Left to right: "part of a newspaper clipping, short, legible, in English", "writing a long message on a blackboard in English", "a message written in sharpie on a sticky note in English". Qualitatively, letters are remarkably well-formed, with some English words mixed in with nonsensical text.

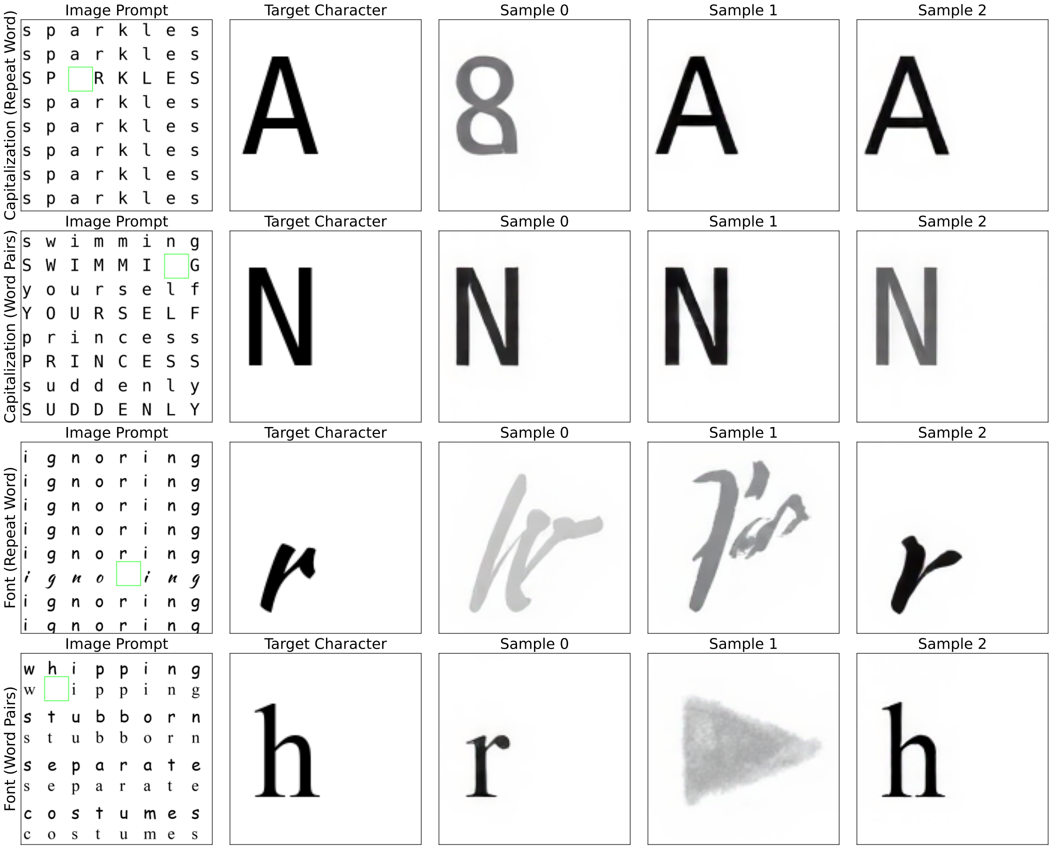


Figure 2: We use image inpainting to investigate an image generative model's understanding of letters independent of their rendered form. We repeat words on an image grid and mask out characters which represent either a variation in font or casing. The "Repeat Word" setting repeats the same word across the whole image and varies a single line. The "Word Pairs" setting prints pairs of words, each containing a varied and unvaried example.

Table 1: We report FLUX.1-Fill's accuracy inpainting the correct character, casing, and font across all settings. For capitalization, 'Character %' indicates the percent of inpainted letters that are the correct character *and* the correct case, whereas 'Casing %' indicates the percentage of inpaints that are the correct case. For the font task, 'Character %' is only calculated based on character, not font. Fonts are classified using a custom-trained model that may not generalize to noisy FLUX.1-Fill outputs; see Appendix B for 120 randomly-selected inpainting generations, which are more representative of the model's consistency for this task.

	Word			Gibberish		
	Character %	Casing %	Font %	Character %	Casing %	Font %
Capitalization (Repeat Word)	39.6	56.9	_	23.9	51.7	_
Capitalization (Word Pairs)	16.4	67.6	_	2.0	64.4	_
Font (Repeat Word)	12.1	_	19.8	17.2	_	26.4
Font (Word Pairs)	6.5	_	29.2	3.53	_	30.0
Random Baseline	2.0	50.0	25.0	2.0	50.0	25.0

shows examples of these visual prompts, their target characters, and samples from the inpainting model.

We use the FLUX.1-Fill model [6] with an empty prompt and 0 guidance scale to ensure unconditional generation. When the mask used to guide inpainting is too large or small, the model is prone to generating visual content other than characters. To minimize this effect, we measure accuracy of copying characters as grid size varies, and use the grid size with the highest reported accuracy (see Figure 15). At this grid size around 30% of the inpainted patches are unrecognizable, introducing noise to the recognition accuracy.

The character and font of the inpainted patch are identified with PP-OCRv-5 [3] and a Resnet34 model [4] trained to recognize fonts, respectively. PP-OCRv-5 reports a 86.79% recognition accuracy on printed english and our font classifier achieves a 100% test accuracy on held out characters. The OCR model only sees the character patch, so recognition results for letters whose form does not change with casing (e.g. 'o' or 'z') may be noisy.

For each setting we draw 30 samples each for 50 different images. We evaluate both words and random strings across the four setups, and four different fonts are used for rendering text. Table 1 reports the accuracy of inpainting the correct character, casing, and font across all settings. As classifying model generations is somewhat noisy, Appendix B shows 120 randomly-selected inpainting generations which are more representative of the model's consistency for this task.

Overall, character accuracy is quite high relative to a random baseline, but capitalization and font transfer are more noisy. Printing random strings instead of words decreases character accuracy across most settings—perhaps because the model does not recognize any semantic textual content in such cases, and so focuses instead on matching the visual pattern of the image prompt. For word pairs, models are more likely to inpaint uppercase letters or match the desired font, but these outputs are less likely to be the correct character.

3 Probing for Symbolic Representations

FLUX.1-Fill's ability to inpaint the correct character in Section 2 is promising. It is unclear, however, whether or not this is because the model has learned a general representation of letters, or because the model has memorized common fonts. To elucidate this distinction, we train linear probes on VAE latents as well as internal transformer representations of letter patches. We include the VAE in our analysis to understand what symbolic representations it already encodes (if any), and what symbolic representations the transformer must learn.

We train three kinds of linear probes: a character classifier tested on unseen fonts, a font classifier tested on unseen letters, and a capitalization classifier tested on unseen fonts. We take all the patches corresponding to a given letter in the prompt image and concatenate them to obtain a single vector for that character; transformer hidden states are taken from the first denoising timestep. Note that exclude the masked portion of the image, focusing only on parts of the image that are already fully-formed.

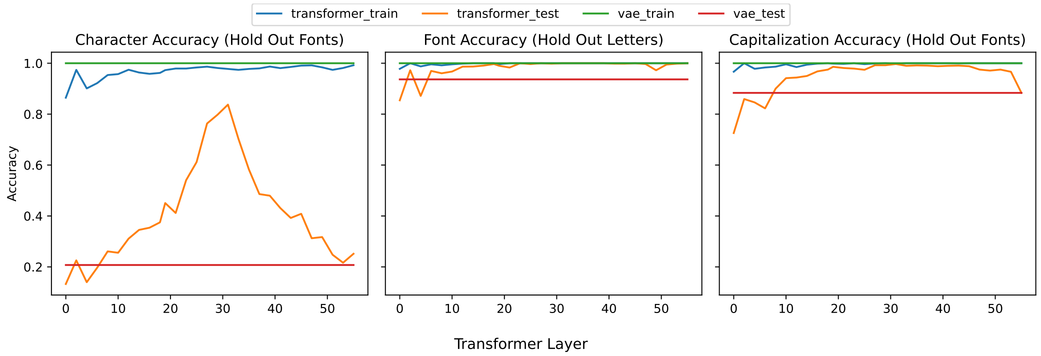


Figure 3: We train three types of linear probe: A character classifier tested on unseen fonts, a font classifier tested on unseen letters, and a capitalization classifier tested on unseen fonts. Font and casing probes already achieve high test accuracy on VAE latents, but test accuracy becomes perfect when trained on intermediate transformer activations. Strikingly, character probes can only generalize to unseen fonts in transformer middle layers, indicating the existence of a general letter representation only at those layers.

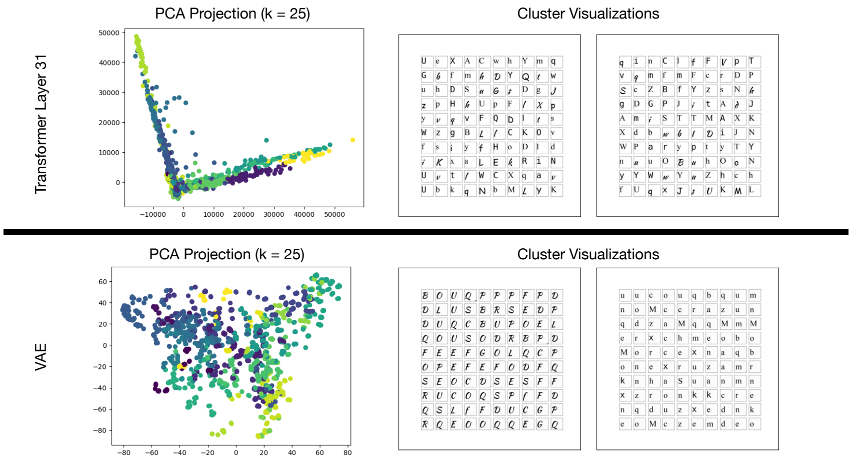


Figure 4: We find that k-means clusters in the VAE latent space tend to group letters by stylistic factors such as font and casing. However, k-means clustering on transformer representations from layer 31 (the middle of the model) returns clusters that are largely uninterpretable. Each color represents a cluster, with $k = 25$.

Figure 3 shows the training and test accuracies for these probes. Font and casing probes already achieve high accuracy on VAE latents, so it is unsurprising that the probes trained on transformer representations would also generalize, but test accuracy does reach 95%. Character probes, however, only generalize to unseen fonts in transformer middle layers, indicating the existence of a general letter representation that emerges only at those layers.

Finally, we perform k-means clustering on these representations, hoping to see whether character clusters emerge in middle transformer layers. While VAE k-means clusters tend to group around stylistic factors such as font and casing, transformer clusters are largely uninterpretable. Representative samples of these clusters as well as their PCA projection are shown in Figure 4.

4 Conclusion

These experiments provide an initial look into the implicit language priors learned by diffusion models, and help us understand what image models may be able to learn about text.

References

[1] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A. Efros. Visual prompting via image inpainting, 2022.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 417–424, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[3] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] Douglas Hofstadter and Corporate Fluid Analogies Research Group, editors. *Fluid concepts and creative analogies: computer models of the fundamental mechanisms of thought*. Basic Books, Inc., USA, 1995.

[6] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[7] Weize Quan, Jiaxi Chen, Yanli Liu, Dong-Ming Yan, and Peter Wonka. Deep learning-based image and video inpainting: A survey. *Int. J. Comput. Vision*, 132(7):2367–2400, January 2024.

[8] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12:1247–1283, 2000.

[9] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025.

A Variation in Character Cell Size

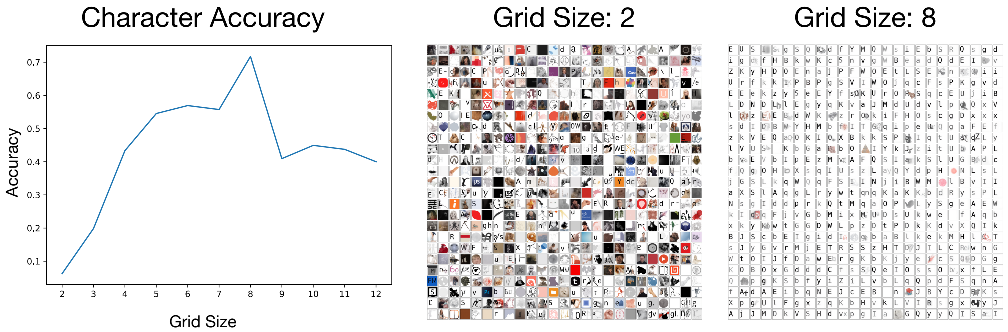


Figure 5: When the mask used to guide in painting is too large or small, the model is prone to generating visual content other than characters. In order to minimize this effect we look at accuracy of copying a character as grid size varies and use the grid size with the highest reported accuracy.

B Inpainting Examples

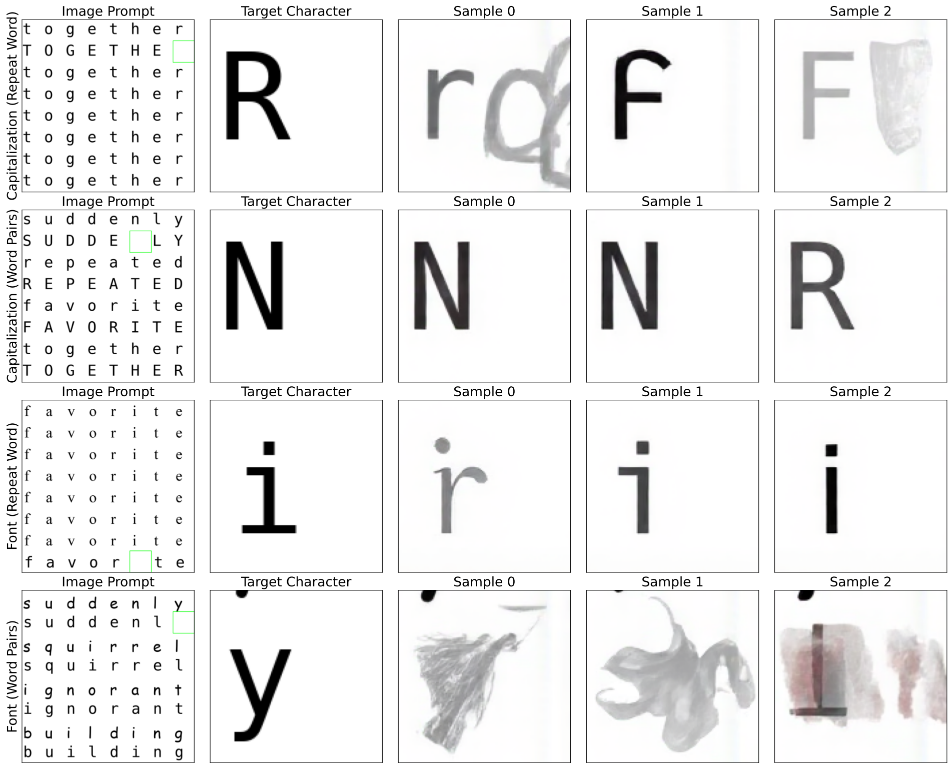


Figure 6: More examples of inpainting.

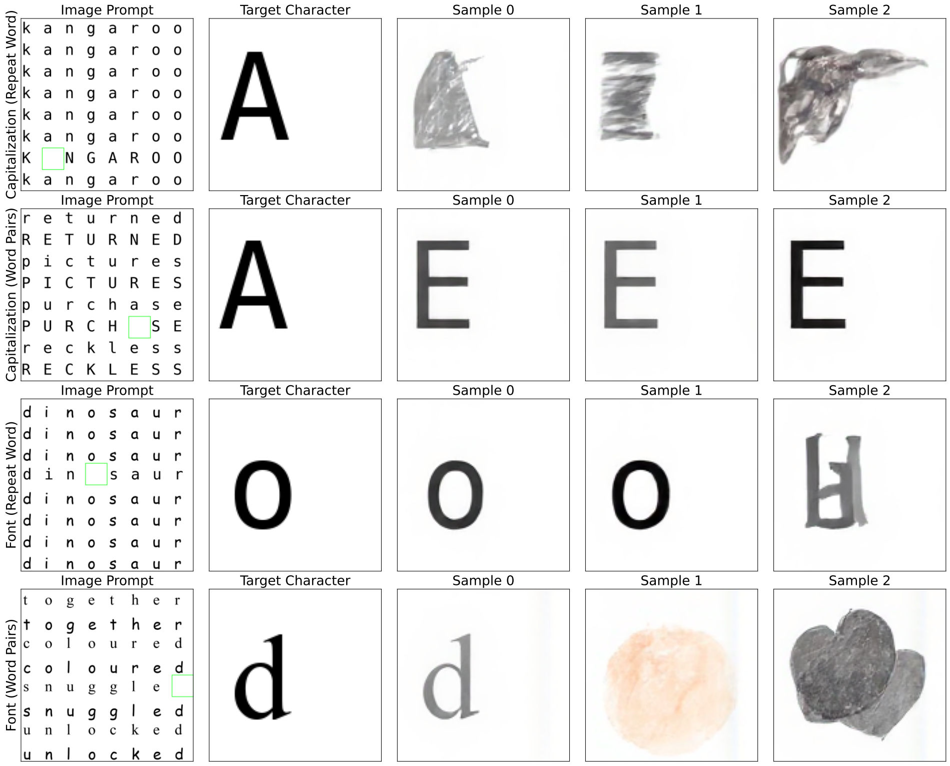


Figure 7: More examples of inpainting.

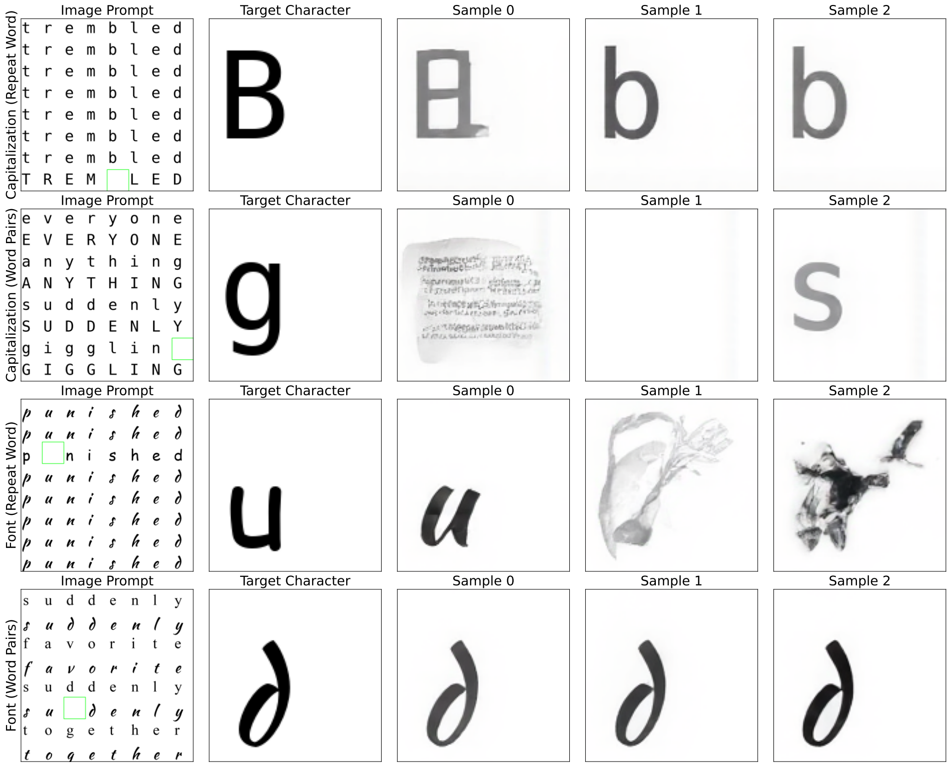


Figure 8: More examples of inpainting.

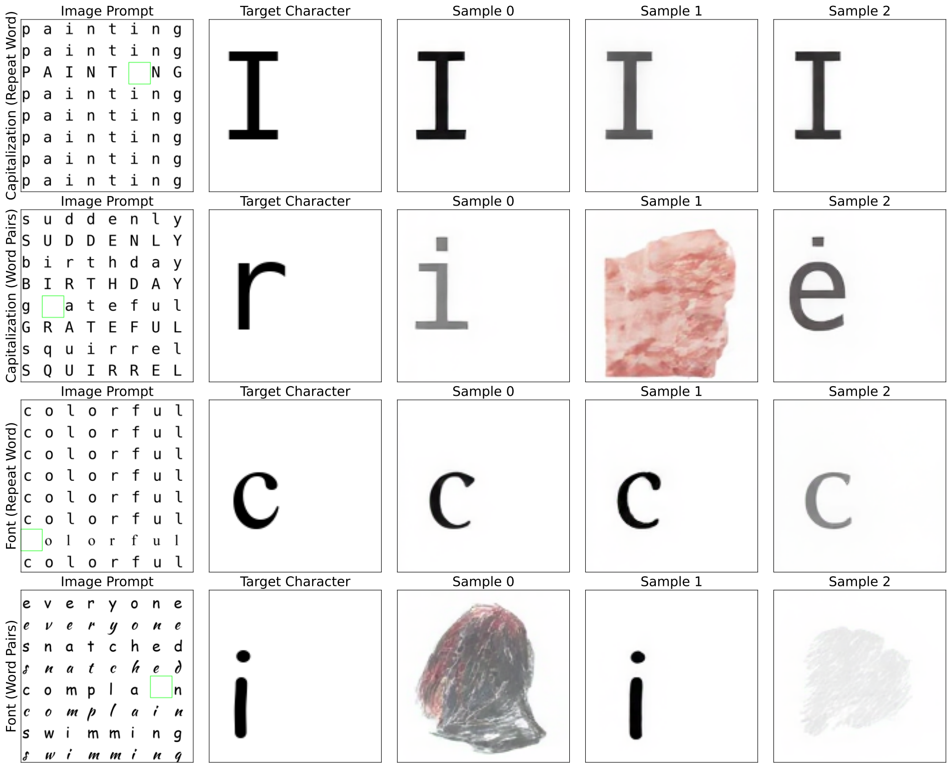


Figure 9: More examples of inpainting.

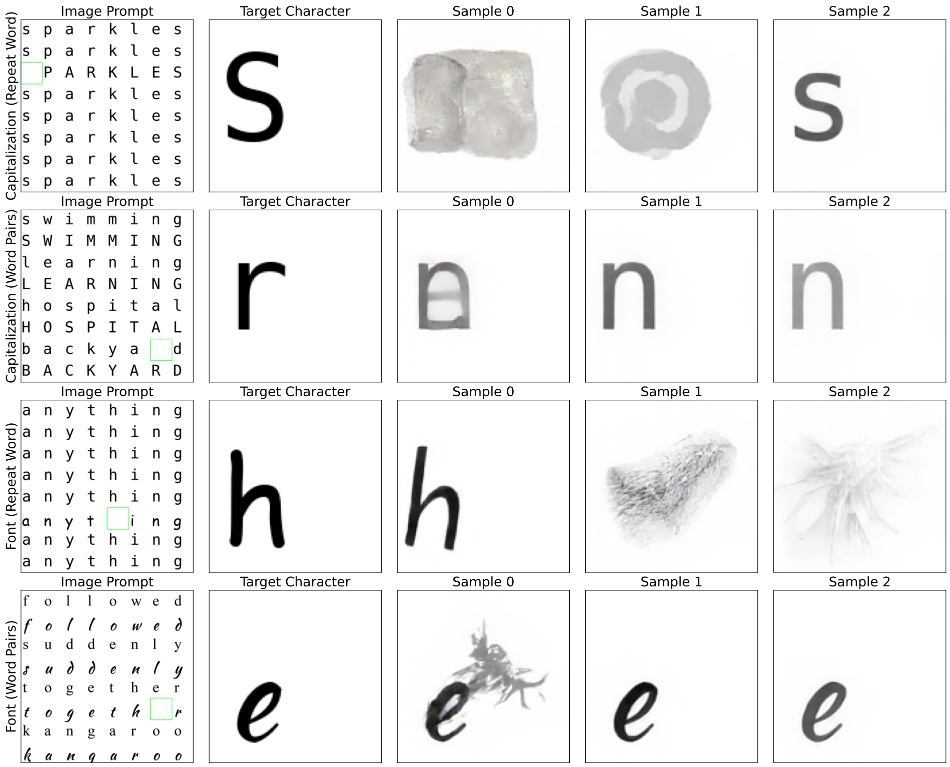


Figure 10: More examples of inpainting.

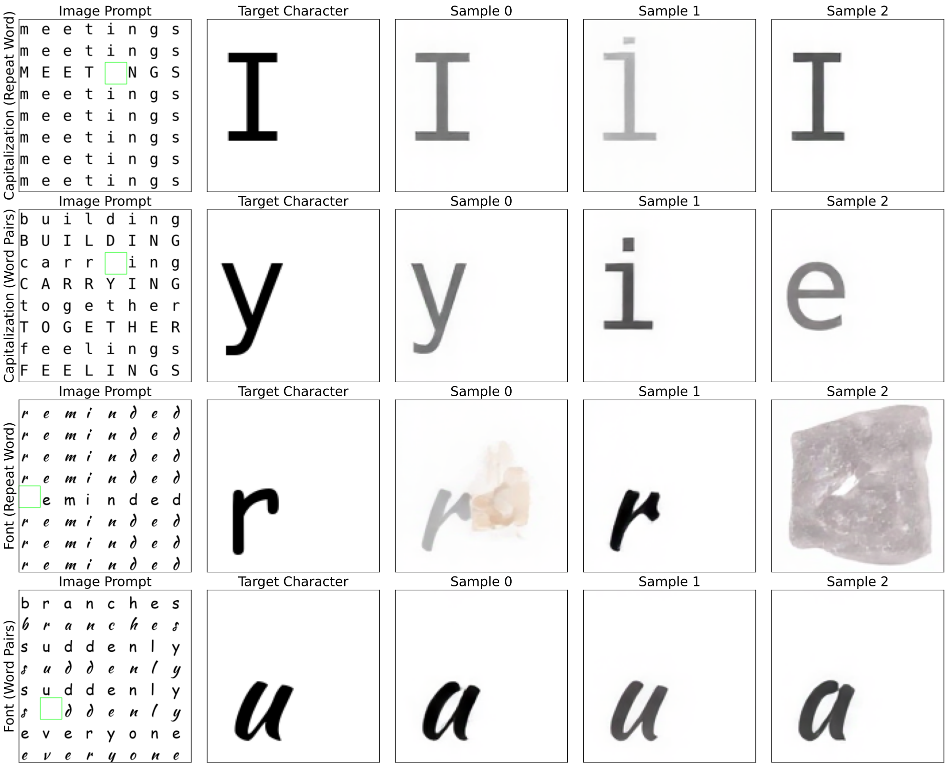


Figure 11: More examples of inpainting.

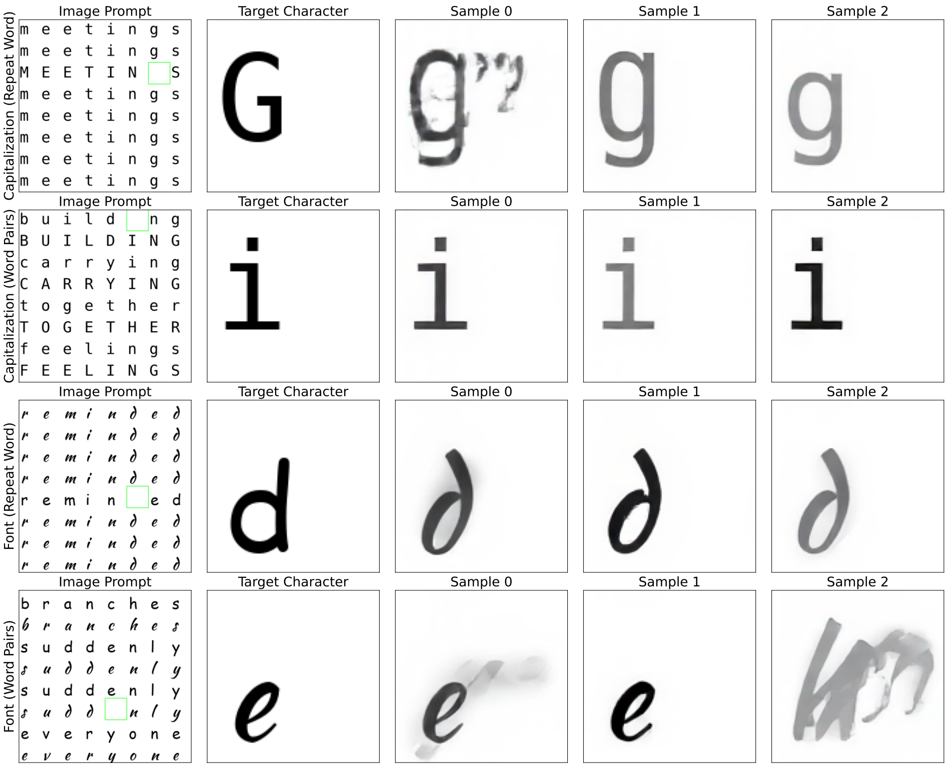


Figure 12: More examples of inpainting.

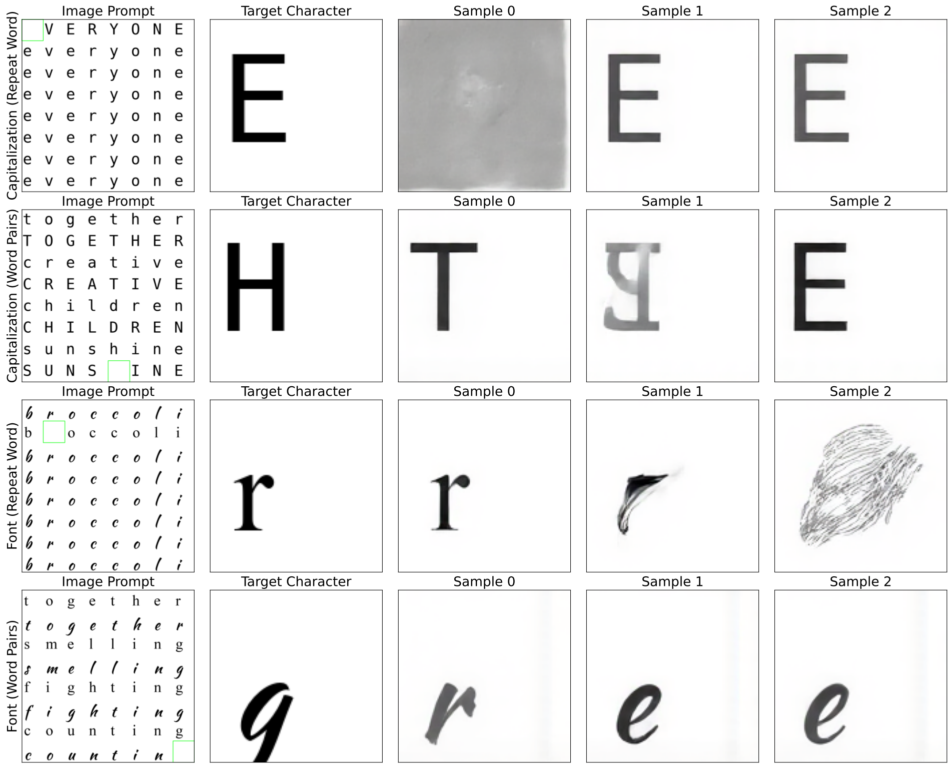


Figure 13: More examples of inpainting.

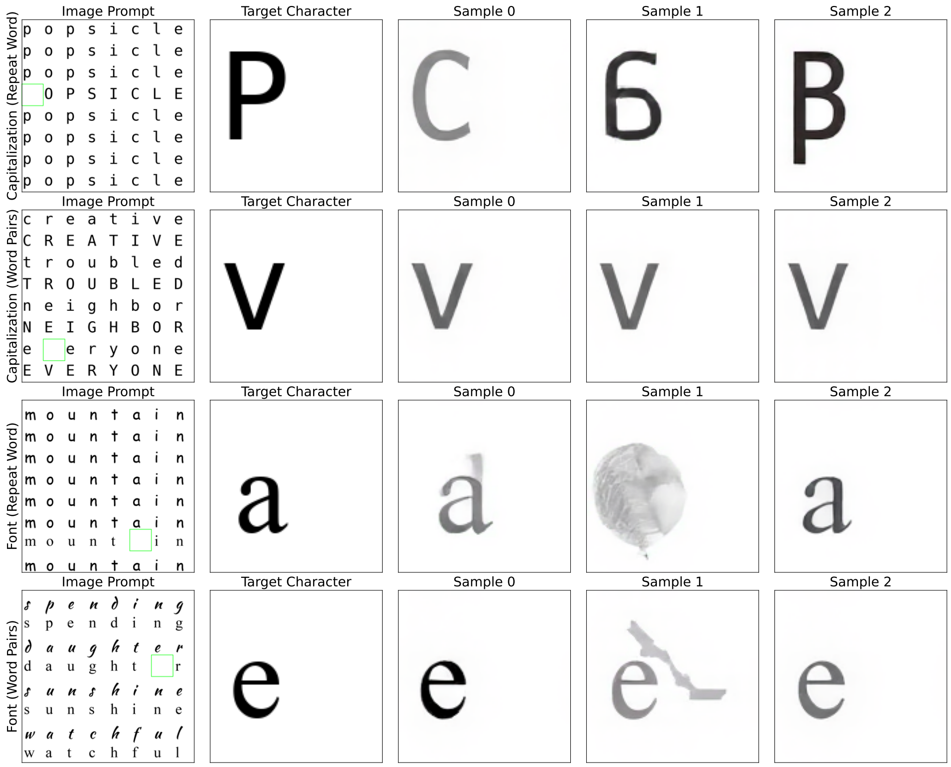


Figure 14: More examples of inpainting.

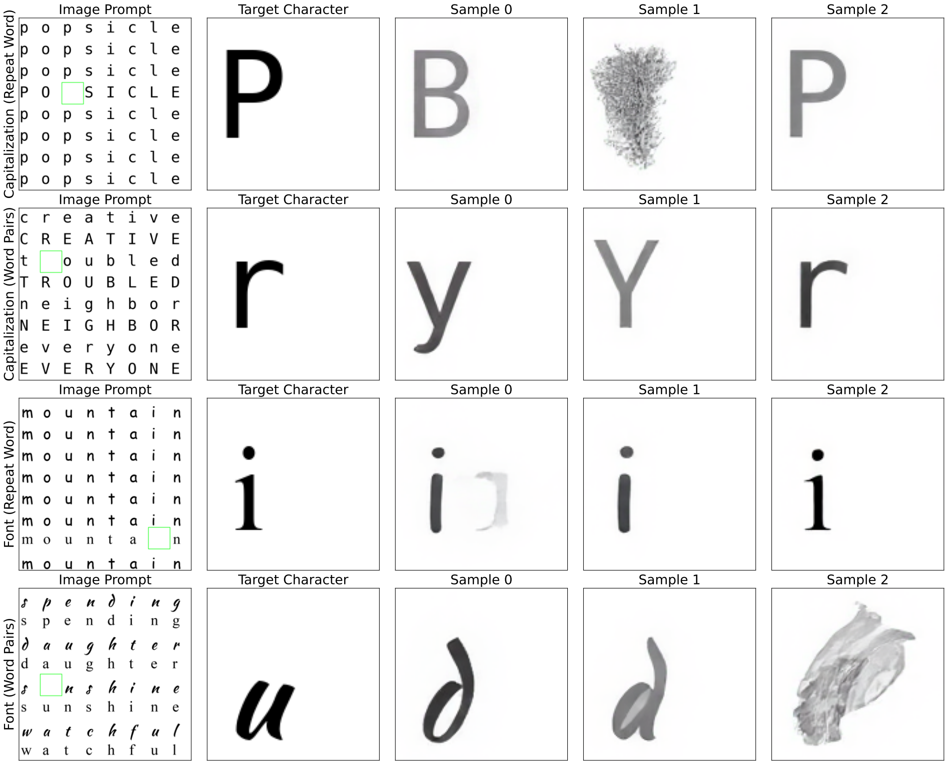


Figure 15: More examples of inpainting.