Supplemental material for the paper: Improving Digital Communication with Personalized Expressive Characters in Interactive Comic Scenes

Alexander Schier[a,1], Caro Schmitz[a,1], Reinhard Klein[a]

[a] *Rheinische Friedrich-Wilhelms-Universität Bonn, Institute of Computer Science II, Friedrich-Hirzebruch-Allee 8, 53113 Bonn, Germany*

Abstract

In this supplemental material to the paper *Improving Digital Communication with Personalized Expressive Characters in Interactive Comic Scenes*, we provide additional insights into the tools used to build our chat, explain the choices and possible alternatives of software and AI models, and show some additional images of our system.

1. Used models, techniques and software

Our system uses a number of existing software and algorithms for parts that we have not reimplemented because the existing solutions have proven to work well. We give an overview of our choices, which are by no means the only viable options, and also discuss possible alternatives.

Gen-AI models

For the image model we chose, Stable Diffusion 1.5 with the main motivation being performance, especially when combined with an LCM [6], and good support for ControlNets [8]. There is also a large user community for Stable Diffusion 1.5, although it is shifting towards SDXL, with parts of the community even shifting towards Flux [2] as people get more powerful graphics cards and the ControlNets for SDXL, which for a long time were worse than those for SD 1.5, are getting better. SDXL is also the most likely image model upgrade in the near future, as the newer models like Flux are not fast enough for an interactive system on current hardware. While we motivated the use of an anime-style model, the specific choice of the *Cartunafied* model is arbitrary, and many other models would be possible alternatives, and we plan to create our own fine-tuned model with a unique style.

For the large language model we chose *Mistral 7B* [4], which was a recent model when we integrated LLM into our system. For our purposes it does not necessarily need to be replaced any time soon, but as there are many newer open weight language models, we may still upgrade to a newer model. The most important qualities a model must have in our system are the ability to transform a textual description into image tags and to reliably generate structured output such as JSON for easy parsing. More powerful models would be interesting for our application if they are able to extract tags from conversations that both match the conversation topic and describe only visual aspects, which current mid-range (7-30B) models cannot reliably do.

ControlNets and LoRA

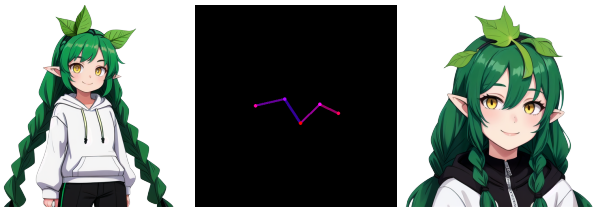

Figure 1: Even when using the `portrait` and `face` tags, we may still get an upper body image (left). By using the OpenPose ControlNet with the pose shown in the middle, we can be sure to get the portrait image we wanted (right). Both images have the same prompt and random seed.

The OpenPose ControlNet is an integral part of the character generation process. Without it, it would be difficult to create convincing character interactions, or even to ensure that character images are in the desired "cowboy shot" pose, and that character avatars show only the face. Figure 1 shows an example of how a portrait image can contain more than just the face if the OpenPose ControlNet is not used. The other ControlNets we use are optional, and could be omitted without major inconvenience. Without the Lineart ControlNet, some meme images would be less faithful to what they should look like, but would still work most of the time. The Tile ControlNet is only used optionally to improve the images during idle times, but is by no means necessary.

We use LoRA models for memes, the backgrounds of some themed rooms, effects like pixelation filters, and to allow users to have a personalized character. While these are fun features, they are not integral to the chat and could be removed without losing important functionality. We also use a LoRA model for the LCM, but there are Sta-

[1] Equal contribution.

ble Diffusion models that already include an LCM, which could be used to avoid having to load it as a LoRA model.

Backend software

For drawing operations, we use the Pillow library, which is a common choice for image processing in Python, and implemented the additional drawing primitives for the splines and extended text wrapping for speech bubbles ourselves. In the future, we plan to switch to more advanced libraries that provide features like anti-aliased drawing, more sophisticated text rendering, and colored emoji.

For image generation we currently use the Stable Diffusion webui [1] as backend, which provides a rich UI for interactive image generation, but the node-based workflow of ComfyUI [3] may be a more flexible alternative for our use case. To run LLMs for our prototype, we use the Text generation web UI [7] with the OpenAI API plugin, which provides a decent web interface for experimenting with LLMs, but for production setups, high-throughput backends like vLLM [5] will be a better choice. Another interesting feature that some alternative backends offer is structured generation, which constrains the token selection to follow a formal grammar, e.g. to ensure that the result is valid JSON data.

For the chat backend, we use the Django web framework, which provides simple model-view-controller logic with templates, URL routes, and an ORM that simplifies database access, and for executing long-running background tasks like rendering jobs, we use the Celery task queue. There is no requirement to use exactly this software, and the backend could also be implemented using other frameworks. The web-based frontend is implemented using HTML5 and JavaScript, so users do not need to install any additional software to use the chat, and we do not need to develop different clients for different operating systems. Using a reactive layout, it can also wrap the panels differently depending on the screen size.

2. Additional images

Here we show a few additional images from our comic chat:

- Figure 2 shows how the previews in the emotion selector are personalized for each character.
- Figure 3 shows how indoor scenes do not work well with characters placed in front of the scene.
- Figure 4 shows some of the perspectives tagged in the Danbooru dataset.
- Figure 5 is an exported comic of a our chat tutorial.
- Figure 6 shows screenshots of the character creation tools.

Figure 2: The emotion and meme previews are personalized to the character of the user. Some memes have a non-square aspect ratio and are centered within a transparent square.

Figure 3: Many indoor scenes do not work well because the size relationships between characters and scene objects do not match, which is especially bad when using our zoom approach for moving around the scene, and one would expect the characters to be standing in the scene instead of in front of it.

Figure 4: A few of the perspectives tagged in the Danbooru dataset. From left to right: `full body, cowboy shot, upper body, portrait`

References

[1] AUTOMATIC1111, and others. Stable Diffusion web UI, 2024.

[2] Black Forerst Labs. Announcing black forest labs, August 2024.

[3] comfyanonymous, and others. Comfyui, 2024.

[4] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[5] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[6] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

[7] oobabooga, and others. Text generation web UI, 2024.

[8] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824. IEEE, 2023.

Figure 5: In the tutorial room, a chatbot interactively introduces users to the various chat features.

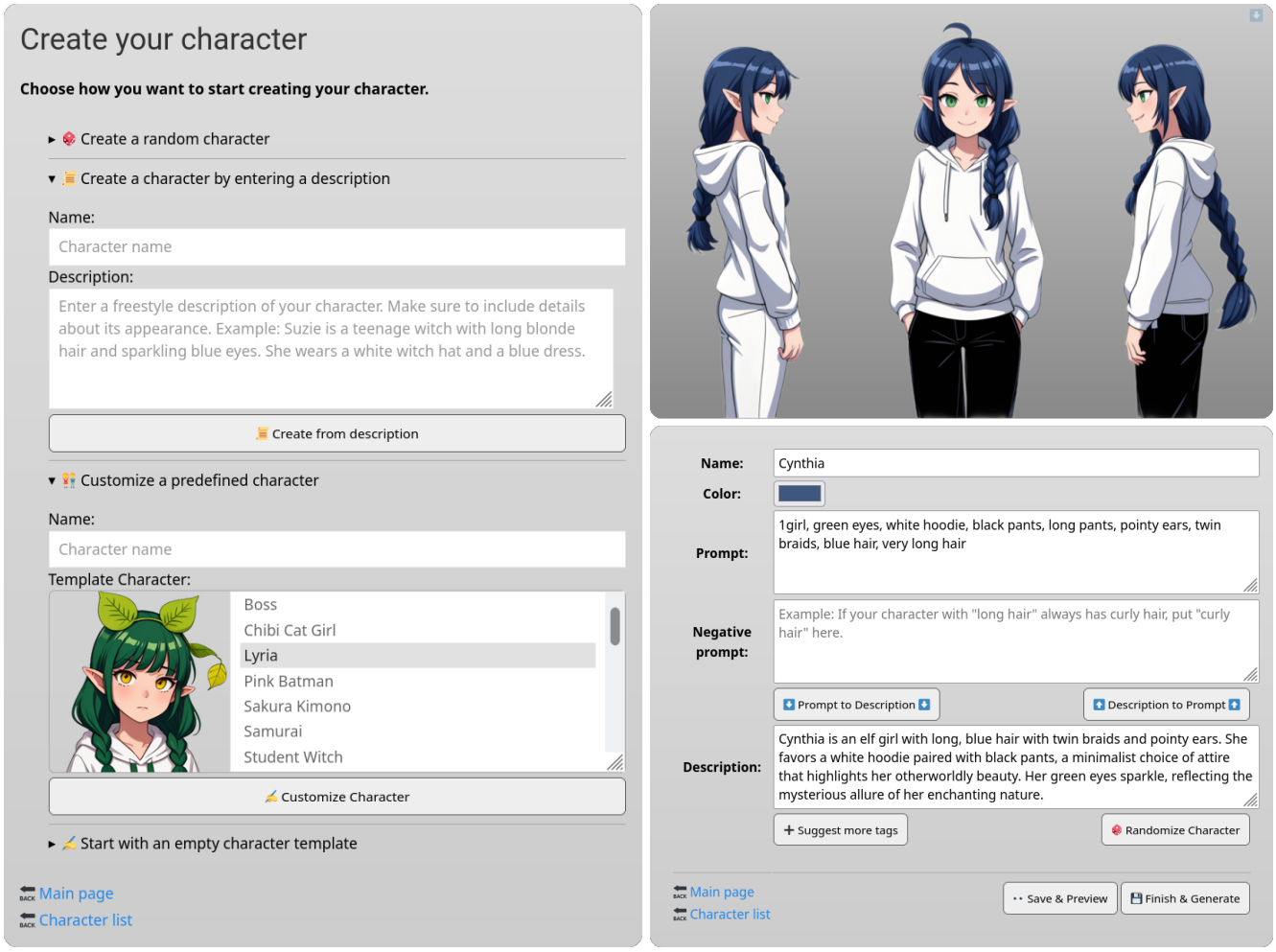

Figure 6: Left: We offer several options for users to create new characters. Right: Once the new character is created, it can be further customized in the tag-based editor.