Typémon: Parameter-Based Pokemon Generation

Pranav Senthilvel*
pranav.senthilvel@yale.edu
Yale University Department of Computer Science
New Haven, Connecticut, USA

Ethan Coyle*
ethan.coyle@yale.edu
Yale University Department of Computer Science
New Haven, Connecticut, USA

Figure 1. Pokémon generated by our model, left to right: Ice Type, Electric Dragon Type, Grass Rock Type, Fire Flying Type

Abstract

The Pokémon franchise is one of the most successful intellectual properties of all time. With over 900 unique Pokémon already in publication, creating new, fresh, and canonically plausible Pokémon designs is a challenge for designers and Pokémon hobbyists. In this paper, we set out to create a tool to automatically generate novel Pokémon using a descriptive parameter-based text input. To accomplish this, we finetune a neural network based on Stable Diffusion and train it on a labeled Pokémon dataset which we constructed. We then test the results of our work by conducting a user study. In the user study we conducted, our model outperformed other Pokémon generators in plausibility and creativity, showing the effectiveness of our approach.

Keywords: content generation, pokémon, stable diffusion, parameter-based generation

1 Introduction

The Pokémon franchise, starting in 1996, is the second-largest game franchise in the world, selling over 440 million copies worldwide[3]. Since the release of the original Pokémon Red and Green games, the franchise has continually expanded the reach of Pokémon lore by introducing new geographical regions, characters, and arguably most importantly, new Pokémon. The original set of Pokémon games released with 151 Pokémon, but over the past two decades has expanded to include 905 official Pokémon, not counting variations and special edition characters[2].

As the collection of Pokémon available to be interacted with in games continues to increase, it has become a pastime of enthusiasts to create their own Pokémon designs, either based on existing Pokémon or from entirely novel ideas. While Nintendo has stated that they do not consider fan-made designs when developing new Pokémon, fans continue to generate new designs as part of the larger franchise community[10].

While Pokémon designs come in many varieties and styles, all of them are classified into a set of 18 types meant to specify each Pokémon's strengths, weaknesses, and special abilities [2]. These types, such as Fire, Water, Fairy, or Dark, often inform the style decisions behind the character designs themselves. For example, most bug-type Pokémon resemble a bug in the real world such as a butterfly (Butterfree, Vivillon), a bee (Beedrill), or even a cocoon (Metapod, Kakuna). Additionally, the Pokémon franchise has defined its own artistic style, one that can be replicated and recognized. Even those unfamiliar with the franchise can identify designs that "look like Pokémon".

These types are also important to game designers when developing new versions of the franchise, since each type is powerful against one type and weak against another. This means that when developing new Pokémon, it is important to consider attributes such as type in order to create a "balanced" generation of new Pokémon that isn't overpowered for one given type.

To facilitate enthusiasts in the creation of new Pokémon, many have developed tools that can be used to fuse together existing Pokémon or generate new ones[9][4][6]. These tools have further increased the popularity of the game and have introduced a new genre of fan content built using tools such as Japeal's builder[6].

However, each of these models has some constraints. Models that rely on the fusion of existing Pokémon often produce

*Both authors contributed equally to this research.

Figure 2. Pokémon generated by the Japeal Fakémon generator[6], fusing the Pokémon Swellow and Litten, via https://japeal.com/fmm/

very similar-looking results, with the newly generated creatures often resembling existing Pokémon too closely, as in Figure 2. On the other hand, AI-based models often generate novel-looking Pokémon, but struggle to create creatures that could plausibly be present in an actual Pokémon game. To our knowledge, existing models for generating new Pokémon face a tradeoff between creativity and plausibility.

In this paper, we propose a method of generating new Pokémon using a stable diffusion approach based on the type parameters for the desired Pokémon. By including the parameters of various Pokémon in prompts passed in for stable diffusion finetuning, we are able to generate a model capable of producing novel, plausible looking Pokémon.

We evaluate this model in comparison to existing Pokémon generation models. We compare results obtained when passing in complex parameters, such as multiple types, and the lack of parameters. We also test the creativity and plausibility of these various approaches in a user study.

In summary, this paper consists of three main contributions:

- We propose a stable diffusion-based model for generating new Pokémon given a desired type or type combination.
- We provide a dataset consisting of 798 Pokémon image-prompt pairs labeling each Pokémon's type as a stable diffusion prompt.
- We evaluate various models' success at producing creative and plausible Pokémon designs with a blind user study.

In the remainder of this paper, we will first summarize some related works in content generation and neural-network based parameter controlled image generation. In our method, we describe our approach to creating the dataset and finetuning an existing Stable Diffusion model to generate Pokémon. Experiments will show the evaluation and comparison of our model with other models, including the results of our user study. Finally, we conclude with a discussion of the project, future steps, and considerations.

2 Related Work

The generation of new Pokémon inspired by existing ones has long been an area of interest for both Pokémon enthusiasts and animation researchers. More broadly, the idea of creating new content using artificial intelligence has been one that has been heavily explored. Parameter-based content generation specifically has been a compelling area of study. Researchers have utilized generative AI models to produce high-quality images to solve various tasks, like colorization, style transfer, denoising, etc. These algorithms are relevant to this research and helped inform our strategies for creating an effective model.

2.1 Pokémon Generators

Fans of the Pokémon franchise have been responsible for the generation of many different new Pokémon generators [6][4][9].

Some models [6][9] are based on the grammar-based concept of merging individual components of existing components. This enables the creation of hybrid Pokémon that resemble characteristics of multiple existing creatures. In Japeal's Fakémon generator[6], users are even able to specify which body parts they would like to inherit from which Pokémon, allowing users to have strong agency in the designs they create. Users are capable of using this approach to create Frankenstein-esque designs. However, this approach is limited to the existing prototypes of each Pokémon, limiting the ability of a user to create an entirely new Pokémon free from the influence of existing designs.

Other models use GAN-based approaches to generate new Pokémon images given the desired color[4]. This produces novel Pokémon that go beyond the hybridization of existing designs. However, this approach also has its own problems, due to the fact that many generated designs do not look like anything at all and often do not adhere to the criteria specified by the user.

2.2 Image Generation Methods

2.2.1 Generative Adversarial Networks (GANs). One popular framework is the Generative Adversarial Network (GAN), in which two neural networks, a discriminator and a generator, compete in a zero-sum game. GAN variants, like CycleGANs[15], Conditional GANs[7], Explicitly-Controllable GANs[13], etc., have been optimized to solve specific tasks.

Specifically, explicitly-controllable GANs are a method of image generation that is highly related to our project. Explicitly-controllable GANs allow for generated images to adhere to specified feature parameters by creating a loss function for each feature that will influence the final image. For example, if the goal of a researcher was to generate an algorithm capable of producing realistic human faces, they may develop a loss calculation for specifying hair color, age, or pose to finetune the output image[13]. Our project explored the possibility of using an explicitly-controllable GAN to generate Pokemon images, but ultimately decided to explore a different method of image generation due to the complexity of generating such loss functions and fitting them to a GAN model.

2.3 Latent Diffusion Models

Recently, Latent diffusion models have become a popular way to perform text-to-image content generation. These models function by attempting to learn the latent structure of a dataset by analyzing the overall latent space. Models like DALL-E[11] have gone viral online, igniting a flurry of online discourse about AI art and the ethics of using Artificial Intelligence to perform creative tasks. Latent diffusion models prove to be extremely useful in prompt-based image generation as it allows for a transformer to create associations between a text prompt and the features to generate. These algorithms are also trained over millions of images, allowing them to scale broadly to many different art styles, image types, and applications. We fine-tune a pre-trained stable diffusion model in order to achieve the results demonstrated in this paper.

3 Methods

3.1 Stable Diffusion Finetuning

In this section, we discuss the method we used to finetune the publicly available Stable Diffusion model to create images in the style of Pokémon given type parameters.

3.1.1 Algorithm Setup. To finetune the model, we initialized the Stable Diffusion model using Stability AI's guide for finetuning from Hugging Face [14]. We also chose to use Lambda Lab's model as the base for our algorithm as it was already finetuned on CLIP image embeddings to create image variations[8]. This was beneficial for our project because our prompts are not meant to generate a specific design, but instead, they serve as a broad guideline to structure the output's representation. We then trained our model using Lambda Lab's environment setup, allowing us to provide an image-prompt pair dataset and train the Stable Diffusion model.

3.1.2 Training Details. We found that our model produced optimal results after about 15,000 training steps, spread across 90 epochs. Any less and the model seemed to produce too many artifacts that made images mostly indiscernible, and checkpoints past 90 epochs produced images that overfitted to ones provided in the training dataset, meaning that prompts had much less of an influence on the final image as the algorithm was fixated on the images provided. The algorithm was trained with a batch size of 4, 1 NVIDIA A100 GPU with 40GB of VRAM, and gradient accumulation with 2 batches. It took about 5.5 hours to reach 15,000 steps. Training can be replicated with more GPUs by reducing gradient accumulation and increasing batch size, but requires at minimum one GPU with a high amount of VRAM.

3.1.3 Prompt Engineering. In order to get new results that emulated the type distinctions we specified, we fed in prompts to the stable diffusion model that match the prompts generated in the training set (see section 3.2.2 below). We attempted other prompt arrangements but this one seemed to produce the most promising results. Testing new prompts may be an area for further exploration.

3.2 Dataset

In this section, we will discuss the method of creating the training dataset that was used to finetune our model. The dataset is publicly available at https://huggingface.co/datasets/pranavs28/pokemon-types.

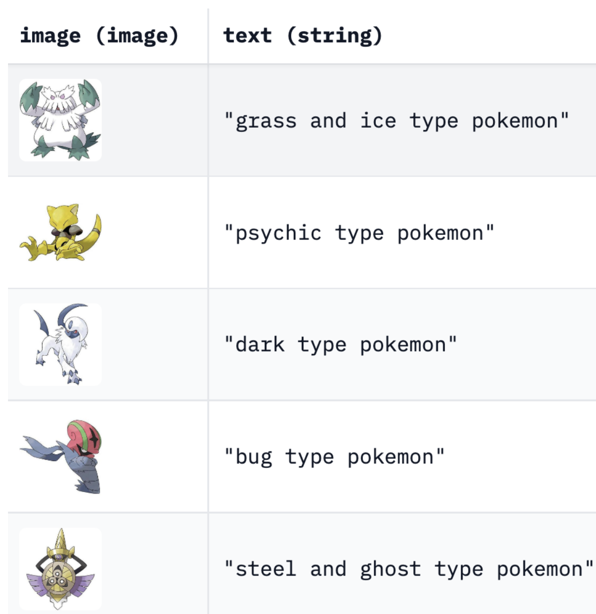

image (image)	text (string)
	"grass and ice type pokemon"
	"psychic type pokemon"
	"dark type pokemon"
	"bug type pokemon"
	"steel and ghost type pokemon"

Figure 3. Pokémon types dataset sample entries

3.2.1 Data Sourcing. The images used in our dataset were retrieved from the "Complete Pokémon Image Dataset" on Kaggle[5]. This dataset originally contained images for the 905 images in Generations 1-8. These images were stored in folders corresponding to the names of each Pokémon, with each Pokémon having 1-8 photos available in the dataset. All of these images feature a full-resolution picture of the Pokémon (no sprites) on a plain white backdrop. To obtain the type information of each Pokémon, we sourced data from the "Complete Pokemon Dataset" on Kaggle[1]. This dataset features all attribute information for each Pokémon in Generations 1-7. This data is stored in a csv file containing names, corresponding types, abilities, weaknesses, strengths, and other attributes that can be found in the official Pokédex.

3.2.2 Dataset Aggregation. To create this dataset, we first stripped the image set to only include one image for each Pokémon (even if the Pokémon in question has multiple forms). This was to prevent overfitting on a particular Pokémon since the dataset is not very large to begin with. Once these images were created, we linked each image to the Pokémon's types in the Complete Pokémon dataset. However, this dataset only stored information for Generations 1-7, meaning we had to remove Generation 8 Pokémon from the dataset. After processing the data and creating pairs, we arrived at 798 images. To convert the types into prompts that could be paired for Stable Diffusion, we wrote a simple script to turn these types into bare-bones prompts describing the Pokémon. Pokémon with one type were assigned the prompt: "[type] type pokémon" and Pokémon with two types were assigned the prompt: "[type1] and [type2] type pokémon". The dataset was then uploaded to Hugging Face for public use. See Figure 3 for samples from the set.

3.3 User Study

3.3.1 Subjects. In order to evaluate how the results from our model compare to other existing Pokémon generation tools, we conducted a user study with 20 Yale College students, half of which are familiar with the Pokémon franchise (defined by those who have played at least one game in the franchise, watched a version of the TV series, or read a version of the manga). Participants were given access to the official Pokédex[2] and were given 20 minutes prior to the study to familiarize themselves with the Pokémon in it.

3.3.2 Data. 40 images were generated from our model for four categories (Fire Type, Dark Type, Water and Electric Type, Poison and Dragon Type). The Fire Type category was selected because Fire Type Pokémon are the most recognizable Pokémon, besides Pikachu. The Dark Type category was selected because it does not represent a specific real world element (such as water, electricity, or rock) and therefore provides a different challenge to various models. We also included two combo types in order to evaluate the model's ability to create hybrid type Pokémon.

In order to retrieve data from other models, we randomly generated 40 Pokémon that meet each category criteria. For the Nokémon generator[4], we selected the primary type and generated 40 images each. We also paid for premium tokens to generate Pokémon with hybrid types for the latter two categories. To generate images for the Japeal generator[6], we randomly generated Pokémon for each category with the only input being the types.

3.3.3 Questions. For each subject in the user study, we selected 4 images at random for each category from each model. The user was not told which model produced which result, and was also not informed about which model was one that we generated. For each set of four images, the users were asked to rate the plausibility, defined as the likelihood they could see the designs generated to be included in the game, on a scale of 1-10. They were also asked to rate the creativity, defined as the uniqueness of the Pokémon in comparison to existing designs, on a scale of 1-10. These values were collected for each model and each category.

4 Results

In this section, we will show selected results of the model for producing Pokémon. To see more sample output images, please visit the Github page for this project.

4.1 Single Type Pokémon Generation

The results from our model produced promising images for single-type Pokémon. Images generated when randomizing seeds were unique and did not overfit to a particular Pokémon, except in one case (Vivillon for bug types). Figure 4 shows the breadth of images that were generated across various types. One important detail to note is that despite our finetuning data containing images only with white backgrounds, the model seems to have generated backgrounds for some images produced. This is an indicator suggesting that while the finetuning data does influence the model's generated images, it is not an absolute influencer.

Images produced seemed to be created in various poses, and maintaining the same seed across prediction steps allowed us to see that the poses were largely dependent on the seed itself and not the model fitting for a specific type.

4.2 Multi-Type Pokémon Generation

The results for multi-type Pokémon generation varied greatly depending on the type combinations. This is likely due to the lack of visual similarity within some types, meaning that in combinations, some types overpower others. In Figure 5, the Fighting/Psychic Type Pokémon demonstrates this as neither the Fighting nor Psychic type have a particular cohesive style, making it difficult for the model to create a Pokémon with a unique style. However, in general, the model still generates images that are relatively clean despite a few artifacts that best represent the type combinations.

4.3 Artifacting

In some cases, the model fails to generate Pokémon images without significant artifacting (see Figure 6). This seems to be a common problem with the Stable Diffusion model generally and can be seen in other applications that attempt to finetune Stable Diffusion for other purposes. This happens to a certain extent in about 1 in every 4 images generated, which constrains the efficacy of this model producing single sample results that are accurate every time.

Figure 4. Pokémon generated with single type classifications, left to right: Dragon Type, Psychic Type, Steel Type, Ice Type

Figure 5. Pokémon generated with multi-type classifications, left to right: Electric Dragon Type, Fighting Psychic Type, Grass Rock Type, Fire Flying Type

Figure 6. Pokémon generations that produced artifacts - Left: Ghost Type, Right: Poison Type

4.4 User Study

The results of the user study showed that our model performed better than the two prominent existing Pokémon generation tools in both plausibility and creativity (see Table 1). While our model is not near plausibility or creativity necessary to be synonymous with Pokémon design, it shows that the Stable Diffusion finetuning method can demonstrate unique and realistic designs that model existing Pokémon while adding new design elements.

Table 1. User Sentiments on Pokémon Generation Models

Model	Avg. Plausibility	Avg. Creativity
Japeal	5.05	4.74
Fakémon	4.94	7.00
Our Model	7.48	8.15

4.5 Runtime

After training the algorithm, it is possible to generate a set of 4 images for a given prompt with 50 sampling steps in about 11 seconds. This is on par with the base Stable Diffusion model. Increasing the amount of sampling steps to 75 increases the runtime by about 4 seconds, but could be useful if experiencing significant artifacting in outputted images. While the only way to run the current version of the model is through the command line or a Jupyter notebook, it is possible to build a frontend model using Gradio or Hugging Face spaces so users can try the model out without accessing any code. However, this requires a server to be hosting and processing the commands.

5 Discussion and Conclusions

This research shows the wide applications of latent diffusion and how it can be applied to create parameter-based models for Pokémon generation. The model that was trained to produce these images was based on the idea that by providing carefully selected prompts paired with consistent images could create a style that could be replicated with future prompts tailored to the same specification.

In the case of Pokémon, it can be seen that the distinct style behind the designs of characters can be translated through text-to-image generators, and can be done with a relatively small finetuning dataset.

While other existing Pokémon generation models struggle to create plausible and creative designs, our model generates images that users experience to be more plausible and creative than our generators. We also provide a dataset that labels Pokémon with prompts for use in other algorithms training and can be used to further research in the space.

6 Limitations and Societal Impact

Recent advancements in AI models used for image generation have sparked a debate about the impact that technology like this will have on those in creative professions. People fear that companies won't hire artists if they can use cheap AI-generated images instead. Copyright concerns are raised since AI is trained on the work of past artists. AI art has already advanced to the point where it can even be indistinguishable from human-made art - a piece made by AI even won an art competition [12]. The images made by our model may not be production ready, but we believe that tools like PokeGAN can be invaluable tools for designers and hobbyists. AI models can be used in the early stages of the design pipeline for inspiration, style checking, and as a low-barrier way for amateurs to get their start in designing Pokemon.

7 Future Work

There are many potential research areas to expand upon this research.

In the space of latent diffusion, various experiments can be done using different sets of data to see if dataset size, types of images used, or other factors influence the resulting images generated by the finetuned model. The Stable Diffusion model can also be trained on images from other digital game franchises, potentially allowing for the generation of new characters from other digital universes. Within the Pokémon universe, variations in representations of Pokémon images can be explored, such as testing with sprites or 3D representations of characters.

This system could also be expanded upon by including more/different parameters in the training phase to see if more specific Pokémon can be generated in this model. This could be done by adding evolution stage, legendary status, potential moves, and other attributes to the original prompts in the Pokémon types dataset. While we were unable to accomplish this due to computing constraints, it would be worthwhile to see if adding too many parameters reduces the quality of outputted images.

To make this tool more user-friendly, a frontend interface can be built in Gradio or Hugging Face Spaces to allow any user to simply select their parameters and generate a corresponding image. This would involve hosting the script on a server in a cloud platform like AWS that would receive calls from the frontend to generate images. While this would likely slow down image generation times, it would prove useful for facilitating user creation and ease of access.

Alternative machine learning models could also be explored. When conducting research for this project, we tried to utilize various different approaches, such as using explicitly-controllable or Conditional GANS. We believe some of these methods show promise for automatic Pokémon generation. Future work could explore these avenues and compare the quality of generated images across these methods.

Acknowledgments

We would like to thank Professor Julie Dorsey and Alvin Shi for guiding us throughout the development of our project. As undergraduate students with limited prior experience in machine learning, this project was a unique challenge with a steep learning curve, and it would not have been possible to accomplish what we did without their support. We would also like to thank our fellow students in CPSC 479/579 for providing helpful feedback, encouragement, and ideas as we worked on our project.

References

[1] Rounak Banik. 2017. *The Complete Pokemon Dataset*. Retrieved December 19, 2022 from https://www.kaggle.com/datasets/rounakbanik/pokemon

[2] The Pokémon Company. 2022. *Pokédex*. Retrieved December 19, 2022 from https://www.pokemon.com/us/pokedex

[3] The Pokémon Company. 2022. *Pokémon in Figures*. Retrieved December 19, 2022 from https://corporate.pokemon.co.jp/en/aboutus/figures/

[4] Liam Eloie. 2022. *Nokemon*. Retrieved December 19, 2022 from https://nokemon.eloie.tech/

[5] hlrhegemony. 2020. *Complete Pokemon Image Dataset*. Retrieved December 19, 2022 from https://www.kaggle.com/datasets/hlrhegemony/pokemon-image-dataset

[6] Japeal. 2013. *Nokémon Generator*. Retrieved December 19, 2022 from https://japeal.com/fmm/

[7] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784 http://arxiv.org/abs/1411.1784

[8] Lambda Labs ML. 2022. *Lambda Diffusers.* Retrieved December 19, 2022 from https://github.com/LambdaLabsML/lambda-diffusers

[9] Alex Onsager. 2013. *Pokemon Fusion.* Retrieved December 19, 2022 from https://pokemon.alexonsager.net/

[10] PKMNBWNET. 2012. *Fakemon Design Contest.* Retrieved December 19, 2022 from https://www.deviantart.com/pkmnbwnet/journal/Fakemon-Design-Contest-Winners-announced-279875442

[11] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. *CoRR* abs/2102.12092 (2021). arXiv:2102.12092 https://arxiv.org/abs/2102.12092

[12] Kevin Roose. 2022. *An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy.* Retrieved December 20, 2022 from https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html

[13] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gérard G. Medioni. 2021. GAN-Control: Explicitly Controllable GANs. *CoRR* abs/2101.02477 (2021). arXiv:2101.02477 https://arxiv.org/abs/2101.02477

[14] StabilityAI. 2022. *Stable Diffusion 2.1.* Retrieved December 19, 2022 from https://huggingface.co/docs/diffusers/training/text2image

[15] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *ICCV 2017* (2017). https://junyanz.github.io/CycleGAN/

A Online Resources

The GitHub repository for this project can be found at this link: https://github.com/pranavs28/cpsc479-project-final

The dataset created for this project can be found at this link: https://huggingface.co/datasets/pranavs28/pokemon-types.

To view sample images produced by this model, download the .zip file here: https://drive.google.com/file/d/1rtGXY-fV-szUVmNEv9bSo2JwEHsqCYWQ/view?usp=share_link