FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Generative models for style-coherent, parameterizable, and expedited 2D game asset generation

Rafael Soares Ribeiro

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Alexandre Valle Carvalho

Second Supervisor: MSc. Nelson Bilber Rodrigues

July 18, 2024

Generative models for style-coherent, parameterizable, and expedited 2D game asset generation

Rafael Soares Ribeiro

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Daniel Filipe Martins Tavares Mendes

Referee: Prof. Hugo Alexandre Paredes Guedes da Silva

July 18, 2024

Abstract

Modern casino game development is a dynamic field combining advanced technology and intricate design to create engaging and immersive experiences for players. With the growing demand for innovation and rapid production cycles, developers face the challenge of efficiently creating numerous high-quality game assets. This work studies the application of generative AI techniques, namely probabilistic diffusion models, to create 2D icons for casino games. Addressing the need to generate multiple visually coherent assets quickly, the research leverages diffusion models enhanced with novel conditioning methods, aiming to improve the efficiency of game asset development while maintaining high aesthetic standards. The proposed solution employs a state-of-the-art style transfer method combined with a novel noise manipulation technique, "Masked Inverted Latents," to ensure consistency, flexibility, and style coherency in content generation. Empirical validation of the solution's effectiveness was demonstrated through a user study, where participants ranked generated results on four video game icon effectiveness metrics. The impact each architecture component has on the quality metrics was assessed through statistical significance analysis methods. The results showed that the diffusion model backbone heavily influences the perception of quality—27 of the 31 (87.1%) statistically significant analyzed combinations included IPAdapter, revealing it influences participants' opinion of quality stronger than the tested model's pre-frozen weights. The novel method for initial latent creation generated significantly more impactful results than the traditional random noise approach, representing 22 of the 31 (71.0%) statistically significant combinations. To thoroughly and fairly validate the solution's quality, a qualitative analysis of the generated images was also carried out, supported by company artists' expertise in the work's requirements.

Keywords: Generative AI, video games, game asset development, diffusion models, style transfer, noise manipulation

Acknowledgements

I want to express my deepest gratitude to my supervisor, Professor Alexandre Valle de Carvalho, and my co-supervisor, Professor Nelson Bilber Rodrigues, for their guidance and trust and for all the time and effort they dedicated throughout this year to helping me along this journey.

I am also incredibly grateful to Sebastião Oliveira for sprouting the opportunity to develop this dissertation and to Luís Fonseca, João Paulo, and Sara Carvalho for welcoming me back so tenderly. Thank you to everyone else at Fabamaq, for all the day-to-day laughs.

To my mom, my dad, my brother, and my sister, I am forever thankful for all the kind words, support, and unconditional love you have given me all my life. You have made me the person I am today, and I owe it all to you.

To my girlfriend, my home and safe haven, my muse and endless well of joy, thank you for always being there to pull me up when I sank too deep. You are my world.

To my friends, thank you for all the good times and tough talks. I would not have held it together without you.

To everyone with whom I have crossed paths with, who has helped, through words or song, shape me into who I am today, I thank you.

Rafael Ribeiro

"I must not fear.
Fear is the mind-killer.
Fear is the little-death that brings total obliteration.
I will face my fear. I will permit it to pass over me and through me.
And when it has gone past I will turn the inner eye to see its path.
Where the fear has gone there will be nothing.
Only I will remain."

Frank Herbert, Dune

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Problem . . . 2
- 1.3 Hypothesis . . . 2
- 1.4 Dissertation Structure . . . 2

2 Background **4**
- 2.1 Image Generation . . . 4
- 2.2 Diffusion Models . . . 5
  - 2.2.1 Model Training . . . 6
- 2.3 Pixel & Latent Space . . . 7
- 2.4 Latent Diffusion Models . . . 8
  - 2.4.1 Diffusion Inference . . . 9
  - 2.4.2 Inference Conditioning . . . 11
- 2.5 Summary . . . 15

3 Solution **17**
- 3.1 Problem Scope . . . 17
- 3.2 Solution Scope . . . 17
- 3.3 Solution Requirements . . . 18
- 3.4 "Masked Inverted Reference" . . . 19
  - 3.4.1 Architecture . . . 19
  - 3.4.2 Reference Images . . . 22
  - 3.4.3 Content through Initial Noise . . . 22
  - 3.4.4 Style through Attention . . . 32
- 3.5 Summary . . . 32

4 Prototype Evaluation **36**
- 4.1 User Study . . . 36
  - 4.1.1 Test Variables . . . 37
  - 4.1.2 Questionnaire . . . 38
  - 4.1.3 Statistical Analysis . . . 39
- 4.2 Qualitative Comparison . . . 45
- 4.3 Summary . . . 48

5 Discussion **51**
- 5.1 Possible Improvements . . . 52
- 5.2 Summary . . . 55

6 Conclusion **56**
6.1 Contributions 56
6.2 Future Work 57

References **58**

A Appendix: Pre-testing Experimentation **63**
A.1 Steps and CFG Scale 63
A.2 Appendix: Systematic Literature Review Paper 65
A.3 Appendix: User Study Questionnaire 75

List of Figures

2.1 DDPM generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right) [10]. 5
2.2 Visualization of density of example training data distribution at each noising step of the forward process [12]. 6
2.3 Illustration of the KL divergence between the prediction and target [14]. 7
2.4 The architecture of the latent diffusion model [3]. 9
2.5 The architecture of the vision transformer [23]. 10
2.6 The contrastive training and sampling of CLIP [33]. 12
2.7 LoRA decomposes the whole weights matrix into two smaller, low-rank matrices.[1] 12
2.8 The architecture of IP-Adapter with decoupled cross-attention strategy [40]. . . . 13
2.9 Controlling Stable Diffusion with various conditions without prompts [41]. The top row is input conditions, while all other rows are outputs. 14
2.10 Prompt-to-Prompt [46]'s image editing method through inversion and prompt refinement. 15
2.11 Concept map with background concepts and relationships. 16

3.1 Picture of casino machine featuring the spins game "Easy Link."1: Prize screen; 2: Reel area; 3: Bet buttons and settings area. 18
3.2 Pipeline of localized masking of inverted reference noise and random noise. . . . 20
3.3 Solution architecture illustrating the flow of data and integrating stage depicted in Fig. 3.2 (marked with a star). 20
3.4 ComfyUI screenshot showing a simplified diffusion inversion pipeline. On the top, the reference image passes through a diffusion inversion process, resulting in a noised image which, when fed to the bottom forward diffusion process, results in a re-generation of the original reference image. 21
3.5 ComfyUI interface showing the testing pipeline. 22
3.6 Company asset used as reference image. 23
3.7 Images generated from base reference image (Fig. 3.6), with five different randomly chosen seeds. 23
3.8 Results of inverting and re-generating reference image, over five random seeds. Each row used the same seed for inversion and each column for generation. . . . 24
3.9 Company asset reference image converted to .jpg. 25
3.10 Results of replacing transparency channel with different solid colors. The first row is the reference image resulting from the operation; remaining rows are corresponding generations with same seed. 26
3.11 Results of interpolation between inverted reference and noisy image, over five random seeds. Each row used the same seed for generation, and each column used the same noise ratio. All inversions used the seed '508240437744435' 27

3.12 Results of normalized interpolation between inverted reference and noisy image, over five random seeds. Each row used the same seed for generation, and each column used the same noise ratio. All inversions used the seed '508240437744435' 28
3.13 Results of inverting, noising, masking, and re-generating reference image, over five random seeds. Each row used the same seed for generation, and each column used the same noise ratio. 30
3.14 Results of combining a descriptive prompt with previous techniques over five random seeds. Each row used the same initial noise manipulation, and each column used the same generation seed. 31
3.15 Results of varying degrees of noise over reference image before IPAdapter training. Each column used the same noise ratio, each row used the same generation seed, and all generations' initial noise was random. 33
3.16 Results of combining IPAdapter with previous techniques. Each column used the same seed, and each row used a different initial noise manipulation technique. All inversions used seed '508240437744435'. 34

4.1 Example of an icon pack selected for the user study, generated using the command "[man | woman | bird | pear] video game icon" 37
4.2 Icon packs evaluated in the user study, generated using the command "[man | woman | bird | pear] video game icon", and reference image 3.6. 39
4.3 Example of a question from the user study's first section. 40
4.4 Bar plots showing answer count for age and design proficiency questions. 41
4.5 Company assets used as reference images not included in the user study. 46
4.6 Examples of IPAdapter generation failures, for different reference images (prompt: *"pear video game icon"*) 47
4.7 Examples of successful, stylistically coherent generations of different subjects (reference image: Fig. 4.5d) 48
4.8 Examples of entirely successful icon pack generation (reference image: Fig. 4.5b) 49
4.9 Examples of positive and negative text element generation (reference image: Fig. 3.6) 50
4.10 Examples of initial-noise induced generation failures (reference image: Fig. 3.6) . 50

5.1 Side-by-side comparison of results shown in user study. Generated using IPAdapter plus inverted and our masked inverted initial noise, respectively. 53
5.2 Side-by-side comparison of results generated using IPAdapter plus our masked inverted and random initial noise, respectively. 53
5.3 Inputs and results of interpolation between original reference and noisy image . . 54
5.4 Inputs and results of normalized interpolation between original reference and noisy image 54
5.5 Results of normalized interpolation between original reference and noisy image, for five random seeds. 55

List of Tables

4.1 Mean responses for each statement across the six icon packs. The highest values for each statement are in bold. 42
4.2 Paired sample skewness for each statement's results. Cells are colored green, yellow, and red for symmetric, moderate, and high skewness values. 43
4.3 Wilcoxon signed-rank test results for each statement, in black, and effect size value, in red. Grey-colored cells represent significantly different group combinations. 44

Abbreviations

Gen AI	Generative Artificial Intelligence
GAN	Generative Adversarial Network
VAE	Variational Autoencoder
DM	Diffusion Model
T2I	Text-to-Image
PCG	Procedural Content Generation
DDPM	Denoising Diffusion Probabilistic Model
DDIM	Denoising Diffusion Implicit Model
LDM	Latent Diffusion Model
ELBO	Evidence Lower Bound
KL	Kullback-Leibler
DE	Differential Equation
SDE	Stochastic Differential Equation
ODE	Ordinary Differential Equation
SD	Stable Diffusion
SDXL	Stable Diffusion XL
SDXLTurbo	Stable Diffusion XL Turbo
CNN	Convolutional Neural Network
ViT	Visual Transformer
DiT	Diffusion Transformer
SSM	State Space Model
CFG	Classifier-Free Guidance
CLIP	Contrastive Language-Image Pre-training
LoRA	Low-Rank Adaptation
IP-Adapter	Image Prompt Adapter
UX	User Experience
ANOVA	Analysis of Variance
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
IS	Inception Score
FID	Fréchet Inception Distance
PPL	Perceptual Path Length

Chapter 1

Introduction

1.1 Context

Modern game development is a dynamic and ever-evolving field that combines cutting-edge technology with intricate design and psychology to create captivating and immersive experiences for players. In today's gaming landscape, developers leverage advanced graphics, sound effects, and animations to transport players to virtual worlds where luck and skill collide. With a constant push for innovation and adaptation to changing regulations and markets, the field continues to push the boundaries of entertainment while ensuring responsible gaming practices are upkept.

The rapid expansion of the video game market, coupled with the relatively short shelf life of individual games, has created an environment of relentless demand for innovation and new products. Thus, there is a necessity for research that explores ways of improving performance in any and all segments of this development, performance which in concrete terms translates to reducing costs and increasing production speed, all the while maintaining a certain standard of quality [1].

A key aspect of game development is the artistic process of game asset creation. Game assets, resources, elements, or content are terms to refer to the various components that make up a video game. These can be graphical, such as 2D sprites, 3D models, textures, and animations, or non-graphical, such as audio effects and soundtracks. To generate these assets is to obtain, through some algorithm or random process, an original piece of content that respects a set of functional and aesthetic rules.

Generative Artificial Intelligence (Gen AI) is a field of artificial intelligence that focuses on problems in the sphere of novel content generation, be it visual, audio, or textual, as opposed to traditional analysis or acting on existing data like expert systems [2]. Focusing on visual content, the families of algorithms that have garnered popularity are Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and, more recently, Diffusion Models (DMs).

DMs are a class of generative models that use neural networks to learn representations of images through the sequential addition of Gaussian noise to the training data. With the recent success associated with Text-to-Image (T2I) solutions like Stable Diffusion [3], the use of text

to guide and control what the model generates has been popularized. However, other avenues of conditioning continue to be the subject of research. One such avenue is the manipulation of the initial noise or the random sample from the Gaussian distribution from which the diffusion process starts. Besides controlling the "what," it is also important to control the "how." Images contain inherent aesthetic characteristics that can be hard to quantify using words, and as such, techniques like style transfer have been developed to manipulate said attributes.

1.2 Problem

Creating graphical assets is complex and time-consuming, often requiring complex teams of artists with high skill and expertise. The pipeline of creating a single asset from start to finish requires multiple cyclical iterations of ideation, illustration, and modeling. With each iteration, the team's collective vision is refined and converges closer to the final product, albeit at the cost of artists' work hours, as work can be discarded when the creative direction shifts. Further details about the problem are presented in Section 3.1.

1.3 Hypothesis

Recent advances in deep learning and computer vision, namely in state-of-the-art image generation models like Midjourney[1] and Stable Diffusion[4], have demonstrated the feasibility of fast, flexible, and incredibly performant image generation solutions.

Given the current state of both game asset development and image generation models, this work formulates a hypothesis that starts from the general premise that diffusion models can positively impact the process of game asset creation:

> " Combining style transfer techniques with intentional noise manipulation of the initial latent image enables consistent style and content control over the generation of valid game assets. "

We explore the state of the art in generative probabilistic diffusion models and the family of challenges these have applied to. We then develop a solution to a narrowed-down problem within this context.

Namely, we can achieve consistent style manipulation and flexible content control by combining a style transfer technique targeting the diffusion backbone with intentional noise manipulation in select areas of the initial latent image.

1.4 Dissertation Structure

In this chapter, we presented the context and problem that motivates this work and formulated the hypothesis that guides our research.

[1] https://www.midjourney.com

The remainder of the dissertation is structured as follows:

Chapter 2 exposes the background, in a funneling manner, from historical generative models to latent diffusion models and the mechanisms that control them. Chapter 3 elaborates on the problem definition, the requirements it reveals, and the solution we developed to tackle it. Chapter 4 describes the experiment undertaken to evaluate the solution, with quantitative results and a qualitative user study. In Chapter 6, we conclude from the results and list the limitations of our work.

Chapter 2

Background

This chapter presents the background on generative diffusion models that our work entailed, starting with the broad history of image generation techniques and then focusing on diffusion models, latent diffusion models, their training, usage, and other related concepts.

2.1 Image Generation

Synthetic image generation refers to the process of creating computer-generated images using algorithms, models, or techniques within the field of computer graphics. The field has been studied for decades and has been applied to a wide range of applications, including medical imaging, virtual reality, and video games.

Procedural Content Generation (PCG), which consists of automated content creation through algorithmic means, has been applied to digital games since the early 1980s [5]. This broad term encompasses techniques such as cellular automata, fractal and procedural noise, random seed-based generation, and grammar-based generation, most commonly used to generate game levels, maps, rules, and textures. Research on the application of AI methods to the area dates back to the 1990s, although not immediately promising. Guzdial et al. [6] novelly applied machine learning methods to create new *Super Mario Bros.* levels; Karavolos et al. [7] presented a hybrid surrogate model for level PCG, enabling designers to direct the generation process of shooter-game levels, using Convolutional Neural Networks (CNNs).

VAEs, first seen in 2013, are a type of unsupervised learning generative model belonging to the family of autoencoder models that map an input to a distribution over the latent space. These consist of an encoder, which compresses input image data into the latent space, and a decoder, which reconstructs an image from a given latent vector [8].

The GAN, first published in 2014, is an artificial intelligence algorithm that consists of two supervised learning models: a generator, which learns the function that transforms unstructured noise into realistic samples, and a discriminator, which examines samples and returns some estimate of whether they are real (drawn from the training distribution) or fake (obtained by running the generator) [9]. Up until very recently, these comprised the state of the art in multiple image

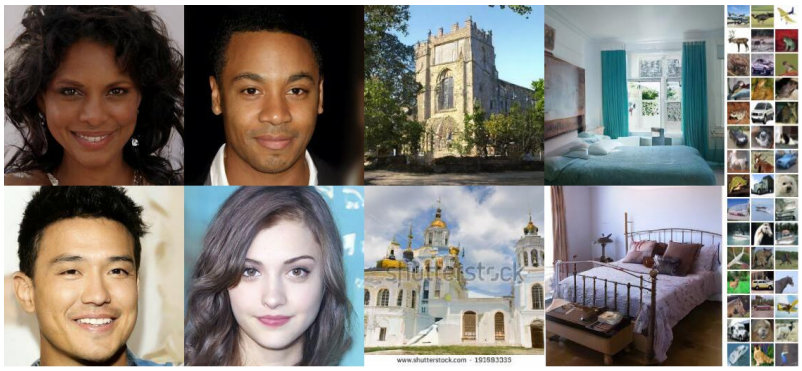


Figure 2.1: DDPM generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right) [10].

generation tasks, despite generally suffering from limitations such as mode collapse and large labeled training data requirements.

DMs, also known as diffusion probabilistic models or score-based generative models, simulate the data generation process by transforming a simple starting distribution into a more complex one. Applied to image generation, this means that the model learns to transform meaningless noise into meaningful images. First published in 2020, this method consists of two inherently opposing steps: first, an iterative forward noising process where real data is transformed into random noise, and then an iterative backward denoising process where the noise is transformed back into realistic-looking data [10]. Compared to previous approaches, these models have shown outstanding results regarding image quality and generation control but are more computationally expensive regarding training and inference. We further describe the concepts behind these models in the following Section.

2.2 Diffusion Models

Sohk-Dickstein et al. [11] proposed in 2015 a method inspired by non-equilibrium statistical physics that tractably restored structure in data from complex datasets.

Five years later, Ho et.al proposed the Denoising Diffusion Probabilistic Model (DDPM) [10], the first application of diffusion probabilistic models for "high-quality image synthesis." It was the first alternative method with results' quality rivaling those of GANs (samples from the original paper in Fig. 2.1).

The model consists of a parameterized Markov chain that learns to produce samples from data by gradually obfuscating it. In the *forward/diffusion/noising* process, noise is injected into the data, most commonly using a non-learned, manually defined variance schedule, to transform the original signal into a Gaussian distribution, as represented in Fig. 2.2. Running the *reverse/generative/denoising* process on a random, normally distributed noise sample using a trained model yields samples from the ground truth initial data distribution, that is, generates images from noise.

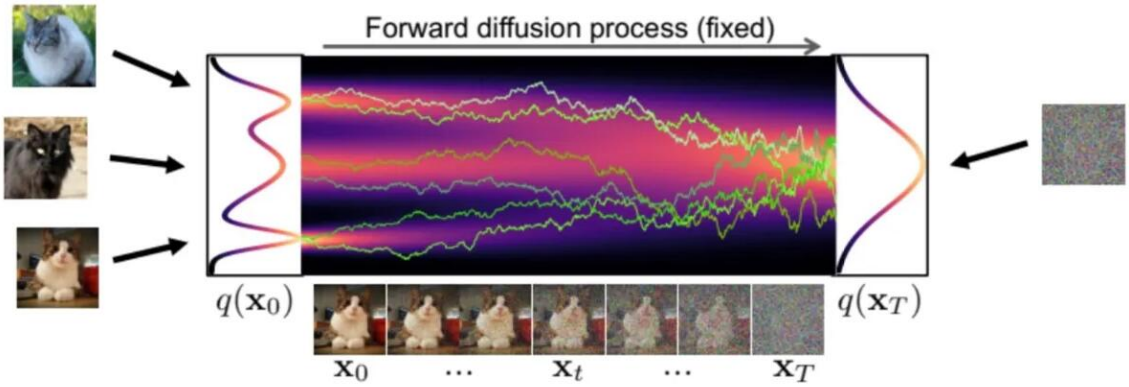


Figure 2.2: Visualization of density of example training data distribution at each noising step of the forward process [12].

2.2.1 Model Training

The objective of training a model in machine learning is maximizing the likelihood of the training data in said model's output. In DMs, this is equivalent to the reverse Markov transitions that maximize that likelihood, which in turn consists of minimizing the variational upper bound on the negative log likelihood [10]. In variational Bayesian methods, this is equivalently described as maximizing the Evidence Lower Bound (ELBO). Eq. 2.1 [10] is the score function used in that optimization, where $q(x|x_0)$ defines the *forward/diffusion/noising* process operation and $p_\theta(x)$ defines the reverse process operation. Conceptually, these represent the prior and approximate posterior of a step of the prediction architecture.

$$\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}\right] = \mathbb{E}_q\left[-\log p\left(\mathbf{x}_T\right) - \sum_{t>1} \log \frac{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}\right] =: L \tag{2.1}$$

Kullback-Leibler (KL) divergence is a measure of the difference between two probability distributions commonly used in machine learning objectives. It is not symmetric, as $D_{KL}(p||q) \neq D_{KL}(q||p)$, which is why it cannot be considered a distance metric but a divergence metric (Fig. 2.3 illustrates this fact, where the first formulation requires p to place high probability in the same values that q places high probability, and the second requires p to rarely place high probability anuwhere that q places low probability). Another interesting property is that KL divergences have a closed form (can be calculated in a finite number of operations) when obtained between two Gaussian (or Normal) distributions [13]. These are represented with notation $\mathcal{N}(\mu, \sigma^2)$, of mean μ and standard deviation σ.

Finally, the definition of the training process includes the variance schedule. It defines the variance of the noise signal added at each step and the total number of steps. These hyperparameters determine the speed and rate at which data is transformed from and into noise, which is one of the critical aspects of architecture. An aggressive noise schedule requires that the model function

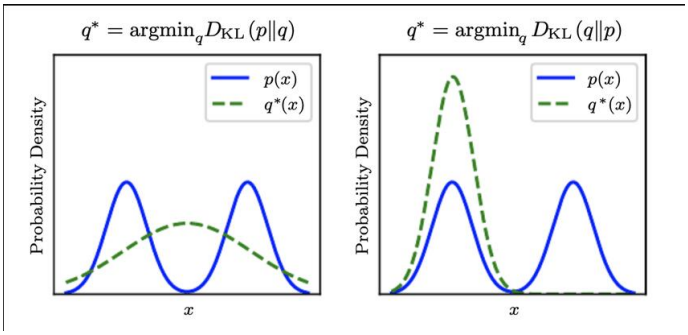


Figure 2.3: Illustration of the KL divergence between the prediction and target [14].

over more steps to denoise the data but delivers results of higher quality. In comparison, a tamer noise schedule requires fewer steps but also limits the variation in data, producing less-detailed, lower-quality results. The original DDPM [10] assumed a simple linear schedule such that the variance of x_t is constant for all timesteps. Nichol et al. [15] showed some problems that result from this simplification, like the drop to zero too early in the noising schedule, and proposed a schedule shape approximable to a cosine function.

Hand in hand with this concept goes the noise sampler, which is the algorithm used to approximately solve the Differential Equations (DEs) that are each step of predicted noise removal. The original DDPM work [10] used Euler's formula, formulating the DE as a Stochastic DE (SDE) (one of the terms is stochastic/probability-based) and solving it with the Euler-Maruyama method. The later Denoising Diffusion Implicit Models (DDIM) [16] formulated the DE as an Ordinary DE (ODE) by removing the stochastic term, enabling the "skipping" of steps to trade quality for speed. DPM [17] novelly used Heun's method, which is a 2nd-order variation of Euler's solver. It is more accurate and stable, requiring fewer steps, but is two times slower at each step.

2.3 Pixel & Latent Space

The pixel space, or RGB space, is a high-dimensional representation of visual data. Dating back to raster graphics, this representation characterizes an image as an area with a set width w and height h of pixel or color values; each split into c channels (red, green, and blue 8 *bit* channels); however, counter-intuitively, the data one contains is of $w \times h \times c$ dimensions.

A latent space, latent feature space, or embedding space defines an embedding of a set of items within a manifold in which similarity between two translates to a lower distance between them. These are commonly used in machine learning tasks that involve learning or representing data according to its inherent features. In the context of images, this translates to representing, for example, a picture of a cat as the set of its intrinsic characteristics (pointy ears, big eyes, long fur, etc.) instead of their visual manifestation in pixels.

The property that makes the latent space so useful for machine learning tasks is its dimensionality, which is much lower than that of the feature space from which its data is drawn. Lower dimensionality of data means that it is easier to handle and operate in and enables the otherwise

impossible extraction of features from unlabelled data, granted it is possible to reduce from the pixel space to the latent space and vice-versa.

A model architecture commonly used in the task of encoding to and decoding from latent space is the Autoencoder. It consists of an encoder and a decoder, which are artificial neural networks trained on the unsupervised task of dimensionality reduction. Its major limitation is the lack of regularization during training, causing the latent space to not be usable in its entirety (decoding a latent vector from an unmapped region produces meaningless data). The Variational Autoencoder [8], a probabilistic variation of the former, addresses this problem by changing the encoder output. Instead of a latent vector, it learns to return the parameters of a predefined latent space probability distribution (a mean vector and a standard deviation vector), which are then used to sample said distribution to obtain a latent vector.

2.4 Latent Diffusion Models

Unlike the previous two types of DM, which perform the noising and denoising of data in the pixel space, the Latent Diffusion Model (LDM) [3] novelly operates in the latent space, a lower-dimensional representation of data that captures the same features and semantics. By performing the reverse process on data encoded to this space of reduced computational complexity, LDMs achieve state-of-the-art execution performance and output quality on unconditional generation tasks.

Large-scale trained text-to-image models like Stable Diffusion [3] have shown outstanding performance in a series of different generation tasks and problems applied to a plethora of different domains. With the ever-changing landscape of the research area, state-of-the-art models are quickly replaced with newer, improved versions. Stable Diffusion XL (SDXL) [18], in addition to the original CLIP ViT-L text encoder, uses a more powerful second OpenCLIP ViT-bigG [19] as well as its pooled text embedding as conditioning. It was also conditioned on images of different sizes and aspect ratios, resulting in generative flexibility beyond the previously fixed 512x512 pixel outputs. Stable Diffusion XL Turbo (SDXLTurbo) [20] is an adversarial diffusion distillation model that using a teacher-student training paradigm, much like that of GANs, learns to generate results on par (image-quality and prompt-adherence-wise) with SDXL at a fraction of the diffusion steps (1 to 4 compared to 30 to 50).

The remainder of this section describes the steps involved in the creation, training, and consequent utilization of a latent diffusion model, whose architecture can be seen in Fig. 2.4. In detail, the data flow at the top shows the training process of encoding via a VAE (the light-blue trapezoid $\mathcal{E}$), pixel-space images (the purple box x) into the latent space, which are then noised over T steps and learned by the model (through the diffusion process), being henceforth internally represented as latent vectors z_T. Inversely, the data flow at the bottom shows the iterative inference process of denoising a random latent vector z_T (through the backbone's CNN architecture), with the guidance of several possible conditioning avenues (seen in the "Conditioning" box, which are all encoded, through the light-blue trapezoid τ_θ, into model-understandable embeddings), T times to obtain

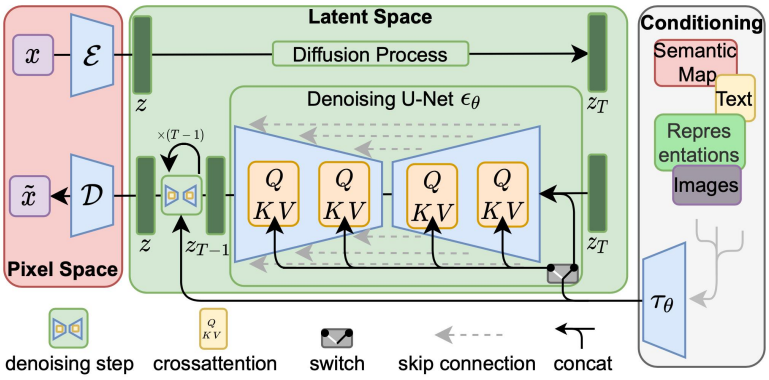


Figure 2.4: The architecture of the latent diffusion model [3].

the resulting latent image z, which is finally decoded back (through the light-blue trapezoid VAE decoder D) into a pixel-space image output $\tilde{x}$.

2.4.1 Diffusion Inference

In machine learning, inference is the process of using a trained model to make predictions on new data. These models are usually frozen; that is, training-specific operations are removed so that they are optimized for and remain unchanged during inference.

Running inference on an LDM consists of three overarching operations: obtaining the initial noise, be it by generating a noisy latent vector or by encoding a noisy image into the latent space; running the reverse diffusion process through the model, revealing features through denoising; and then decoding the resulting latent vector back into pixel space in the form of an image.

The reverse diffusion process, or the denoising approximation executed in the latent space, has been done with a series of neural networks in the past. The common choice U-Net [21] is a CNN architecture widely used in computer vision tasks like image segmentation and super-resolution. It consists of an encoder and a decoder segment, similar to the Autoencoder, that first down-scales the input data through convolutions, extracting its valuable features, and then upsamples said features back into the same dimensionality as the input, using transpose convolution, and concatenates them using skip (or shortcut) connections, a technique consisting of feeding the result of some of the intermediate layers of the network directly to the upsampling.

Transformers [22] have since also been used as a backbone for diffusion models, in the form of Vision Transformers (ViT) [23, 24] (see Fig. 2.5) and, more recently, Diffusion Transformers (DiT) [25]. Like the previously described techniques, Transformers comprise encoder and decoder segments, albeit in a series of identical stacked layers. These encoder layers are composed of a self-attention block that computes importance or dependencies between different parts of a sequence, which feeds into a simple feed-forward neural network, passing the processed output to the next layer. This sequentially processed output of the encoders is fed into the decoders, which are similar in structure except for an additional encoder-decoder attention block between the self-attention and the neural network layers. This block computes attention, or importance, between

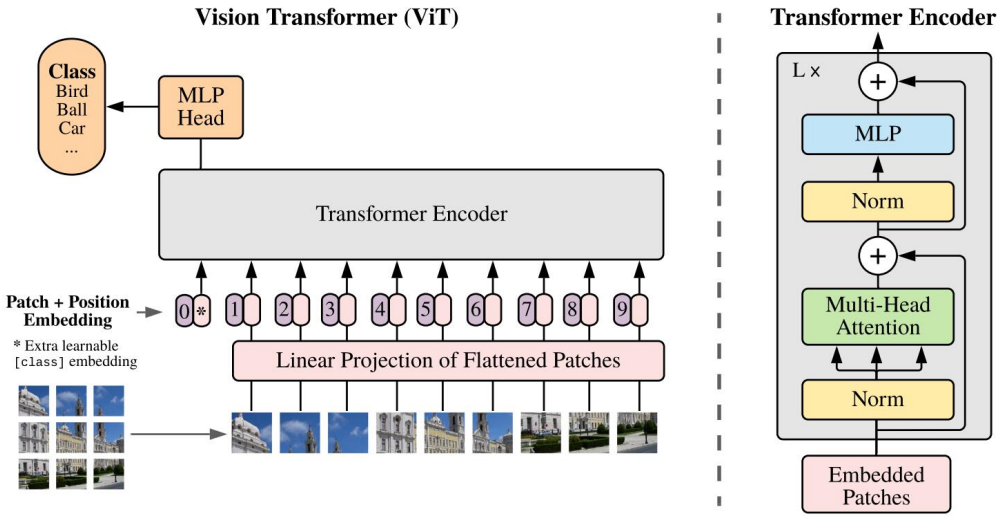


Figure 2.5: The architecture of the vision transformer [23].

two different sequences, as seen in Eq. 2.2). Q, K, and V, respectively, query, key, and value, are the components of input at each attention layer. Each of these is obtained by multiplying the learnable weight matrix $W_{Q|K|V}$ by the embedded input x.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.2}$$

Another novel aspect of this architecture is that each sequence part can be processed in parallel in a process called multi-head attention that not only brings the performance benefits of parallelization for training on extensive data but also enables longer-term dependency modeling [22]. At the image level, input data is segmented into adjacent, non-overlapping square patches, then linearly projected into one-dimensional vectors or tokens. At the self-attention layer, each vector becomes the attention-scored-weighted average of all the image vectors, enabling the modeling of the contribution of each patch to the representation of the whole.

Cross-attention [26] is a mechanism that conceptually combines the "local feature extraction capabilities of CNNs"[26] with the global attention capabilities of standard Transformers by calculating attention at the decoder using both the encoder output and the decoder input. It has shown promising results in multi-modal attention tasks, or tasks where attention on more than one modality of input is required, i.e., image captioning, visual question answering, and conditional image generation. In the latter, this is achieved by making K and V projections of the conditional information and only Q a projection of the noisy image features.

Recently, the field of LLMs shed light on the effectiveness of the State Space Model (SSM) architecture [27] as a response to the limitations of previous architectures (black-box nature, training data hungriness, and limited long-range modeling capabilities for CNNs; quadratic computational complexity bottleneck and data specificity for Transformers), which are starting to see application to the field of denoising diffusion models [28].

2.4.2 Inference Conditioning

Conditioning, in the realm of image generation, is synonymous with "trading off diversity for fidelity" [29], that is, to introduce mechanisms that narrow down the target sampling distribution. These mechanisms can be inputs of various natures, from text or concept embeddings to layout/semantic maps or style/content-reference images.

2.4.2.1 Classifier Guidance

Dhariwal et al. [30] proposed Classifier Guidance to explore the hypothesis proposed in the original diffusion paper [11] of using the gradients of a classifier model to condition sampling on a pre-trained DM, showing the feasibility of a class-conditional DM. However, this approach leveraged the classes of the labeled training data, a solution that doesn't scale to unlabelled datasets, limits the freedom of conditioning inputs, and complicates training since the classifier model must be trained on noised images in line with the DM's, precluding its use on a pre-trained DM. Classifier-Free Guidance (CFG) [31] achieved even better results while aiming for the same effect without a classifier model. Instead of training a classifier, the diffusion process, parameterized by a score estimator $\varepsilon_\theta(z_\lambda, c)$, is jointly trained conditionally and unconditionally by setting $c = \varnothing$ with some probability p_{uncond}. Sampling is then performed using a liner combination of both conditional and unconditional score estimates, $\widetilde{\varepsilon}_\theta(z_\lambda, c) = (1+w)\varepsilon_\theta(z_\lambda, c) - w\varepsilon_\theta(z_\lambda)$, with classifier strength parameterized by w. This mechanism became the first standard in diffusion model guidance and is particularly instrumental to the Stable Diffusion architecture due to its reliance on text conditioning.

Current foundational T2I DMs have demonstrated the efficacy of the text-conditional approach, where an input natural language description of the result, or text prompt, is incorporated into the generative process. AlignDRAW [32] used recurrent neural networks to draw images in patches from natural language descriptions in the first T2I solution in the literature.

Contrastive Language-Image Pre-training (CLIP) [33] (see Fig. 2.6), a zero-shot caption-image prediction model for learning joint text and image representations in computer vision tasks, has since been used in T2I model architectures. Conceptually, it learns to maximize the dot product of image and caption embeddings by training a text encoder on massive, internet-scraped datasets. By assembling random batches of image-text pairs and jointly training the image and text classifier on them, it basically creates a bi-directional classifier for correlating visual and textual information. It was first used for zero-shot prompt-to-image generation in DALL-E [34] but continued to be used as a guiding mechanism in T2I tasks.

GLIDE [29] novelly used a CLIP model trained on noised images to generate embeddings for guidance during generation, albeit without outperforming classifier-free guidance. UnCLIP[35] achieved a higher diversity of generations without sacrificing quality by training a diffusion model to decode CLIP-encoded image embeddings directly.

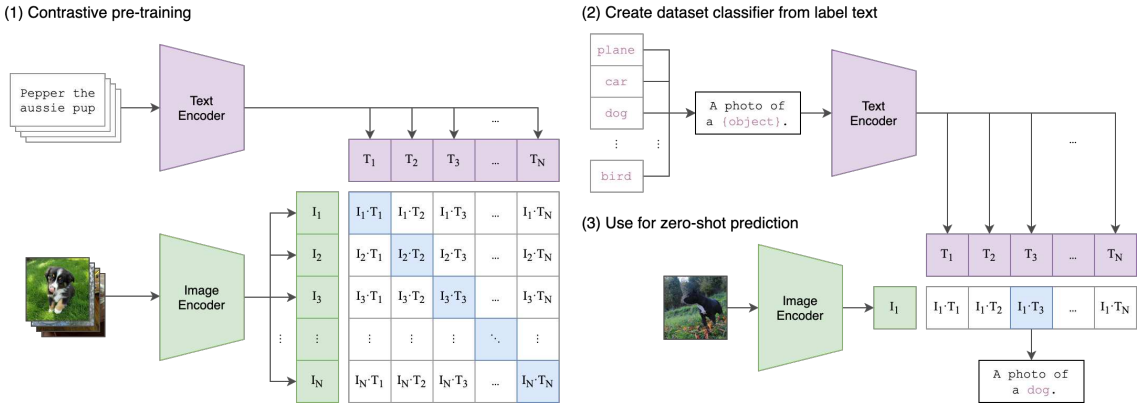


Figure 2.6: The contrastive training and sampling of CLIP [33].

2.4.2.2 Fine-tuning

In traditional machine learning tasks, fine-tuning is the most straightforward method to achieve better results for a specific domain or set of problems. As opposed to re-training a model with particular data, which may lead to overfitting or language drift [36], fine-tuning adjusts the weights of pre-trained models by leveraging additional training data to improve their performance on select tasks.

In the context of image generation, many techniques with varying flexibility and dimensions have been developed in the past three years. In Dreambooth [37], the proposed model learns a binding between a unique identifier and a specific subject through three to five input images, enabling the synthesis of a "subject in different scenes, poses, views and lighting conditions that do not appear in the references." [37] Although the concept of learning bindings of reference subjects had been previously introduced in textual inversion [38], Dreambooth fine-tunes the entire model weights instead of injecting new tokens. Despite its efficacy, the previous technique results in 2-7GB file sizes, which is expensive and leads to longer inference times. It also doesn't scale with larger models, as retraining all parameters becomes less feasible, a limitation identified by the authors of Low-Rank Adaptation (LoRA) [39]. It improved on its predecessor fine-tuning solutions by targeting the cross-attention layers of the inference model's backbone. It freezes the original, pre-trained model weights and injects, at each layer of the transformer architecture, "trainable rank decomposition matrices," which are two low-rank matrices with significantly fewer linearly independent columns, as graphically represented in Fig. 2.7, resulting in a small, scalable solution to better control generation results.

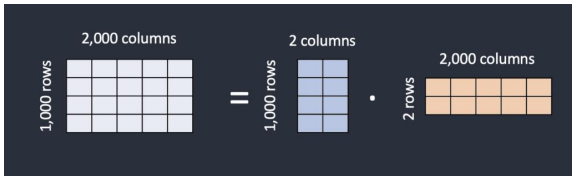


Figure 2.7: LoRA decomposes the whole weights matrix into two smaller, low-rank matrices.[1]

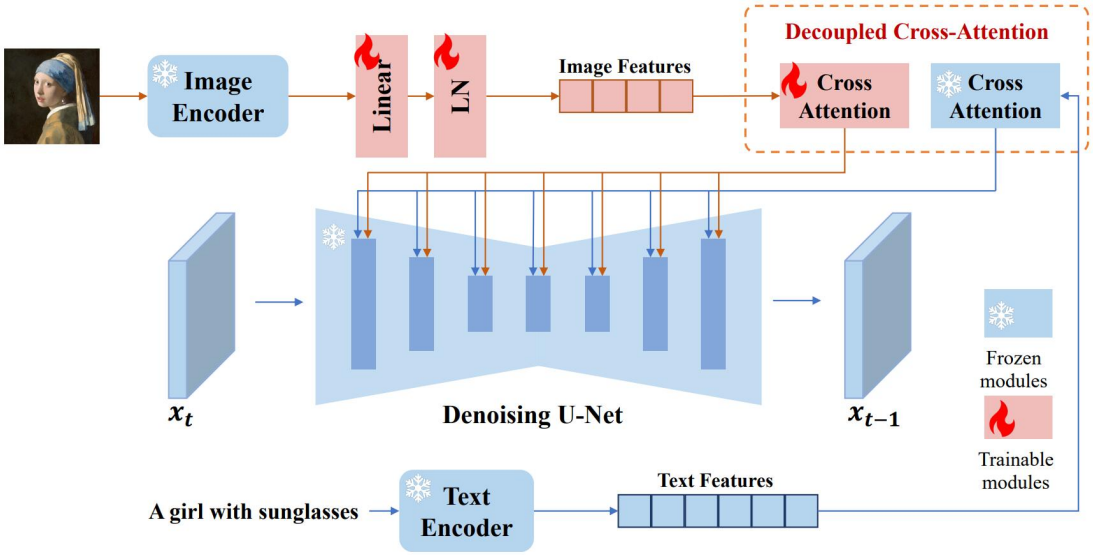


Figure 2.8: The architecture of IP-Adapter with decoupled cross-attention strategy [40].

Since LoRA, research on influencing generation by targeting the cross-attention layer has been affluent. Image Prompt Adapter (IP-Adapter) [40] similarly aims to condition diffusion inference on reference images, although it improves on the former by removing the need for training from the process, enabling what the authors call image prompt capability. Fig. 2.8 shows the architecture consisting of a pre-trained CLIP image encoder model that extracts the features from the image prompt and the cross-attention modules that feed said image features in parallel with the model's preexisting text features' cross-attention modules. As the authors tested by comparing their adapter with a simple concatenation of image and text features, the decoupling is essential for ensuring higher image quality and proper separation of image prompt concepts.

ControlNet [41] performs a form of "pseudo-finetuning" by freezing a pre-trained large DM, creating a trainable copy that will serve as a "deep, robust and strong backbone" [41] that is conditioned on an external vector c. The critical aspect of the architecture is the zero convolution layer connections between the conditioning neural network and the frozen diffusion U-Net, which are tiny convolution layers with weights initialized to zero that progressively grow during training and thus protect the pre-trained backbone from being damaged by noise in its initial phases. The conditioning vector is obtained by encoding 512×512 image inputs into 64×64 latent vectors, inputs which can take various forms, from line sketches, segmentation maps (Canny [42] edge, M-LSD [43] line, HED [44] edge), normal or depth maps and OpenPose [45] body keypoint maps, as seen in each column of Fig. 2.9, respectively.

2.4.2.3 Noise

Noisy data, colloquially known as noise, is data that is corrupted, distorted, or overall contains a large amount of meaningless information. It's generally regarded as an unwanted element in

[1] https://www.hayo.com/article/64e31448411a3dcd7da72bd9

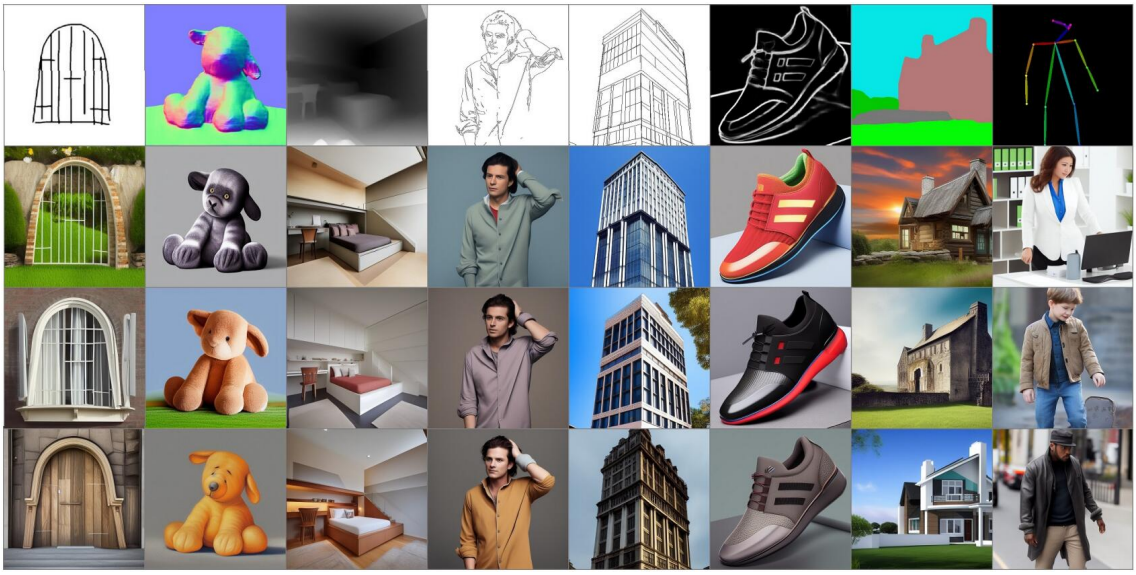


Figure 2.9: Controlling Stable Diffusion with various conditions without prompts [41]. The top row is input conditions, while all other rows are outputs.

computing tasks, except in the realm of diffusion models, where it plays a crucial role in the model's learning of generative capabilities. By having models learn through iterative addition of random noise, or in other words, by smoothing, at each step, the information contained in the training images towards Gaussian noise, it becomes the "zero" representation of data to the models. In practical terms, diffusion is the process of removing noise to uncover data, so manipulating the attributes and presence of said noise is a means of controlling the diffusion process.

This fact has been explored in the literature well since the early days of diffusion models. Research on image-to-image tasks such as image variation and localized image editing (illustrated in Fig. 2.10) novelly introduced mechanisms that leverage noisy latents to control the reverse process.

DDIM [16], an improvement to DDPM [10], reduced the required number of denoising steps with little image quality tradeoff by redefining the reverse process as non-markovian, which transforms the probabilistic (or stochastic) model into a deterministic one, allowing denoising steps to be skipped. This reduction in computational requirements played a role in the growing success of DMs in the following years. DDIM inversion, suggested in the original works [16, 30], consists in performing the diffusion process in the reverse direction $z_0 \rightarrow z_T$ instead of $z_T \rightarrow z_0$. Using the non-Markovian noise sampler to noise an image deterministically, one obtains the initial noise vector, which would generate the input image, given the same conditioning. This is based on the assumption that the denoising process is approximately invertible (x_t can be approximated from x_{t-1}, see Eq. 2.3)

$$x_t = \frac{x_{t-1} - b_t \varepsilon(x_t, t)}{a_t} \approx \frac{x_{t-1} - b_t \varepsilon(x_{t-1}, t)}{a_t} \tag{2.3}$$

However, due to local linearization assumptions introduced by its implicitness, the technique

Figure 2.10: Prompt-to-Prompt [46]'s image editing method through inversion and prompt refinement.

is unstable as error accumulates at each step [47]. The task of optimizing this inversion approximation has been studied recently. Null-Text Inversion's [48]optimization leveraged re-weighing classifier-free guidance to perform meaningful edits. Meiri et al.'s [49] Regularized Newton-Raphson Inversion (RNRI) consisted of applying a simple numerical method called fixed-point iteration to reduce the inversion error at each step, improving both reconstruction and editing quality.

Aiming to use noise as a conditioning mechanism instead, Mao et al. [50, 51] found that areas of random noise that, by definition, should contain no information hold a particular "preference for generating specific content" [50] and applied this fact to the tasks of image re-painting and layout-to-image. [51] implemented a system of collecting and classifying blocks of random noise on a set of categories. These blocks are then used to construct layout-respecting initial noise images, benefitting from the localized tendency to generate said classes.

This hypothesis is one of the key segments of our solution and, thus, is the target of study in the rest of the work.

2.5 Summary

In this chapter, we presented the background of generative DMs, starting with a historical explanation of their learning mechanism, the backbone functioning, and the training. Then, directing the focus to LDMs, we explain the concepts of the pixel and latent spaces and how this model architecture has improved the state of DMs. We then delve into the inference process, explaining

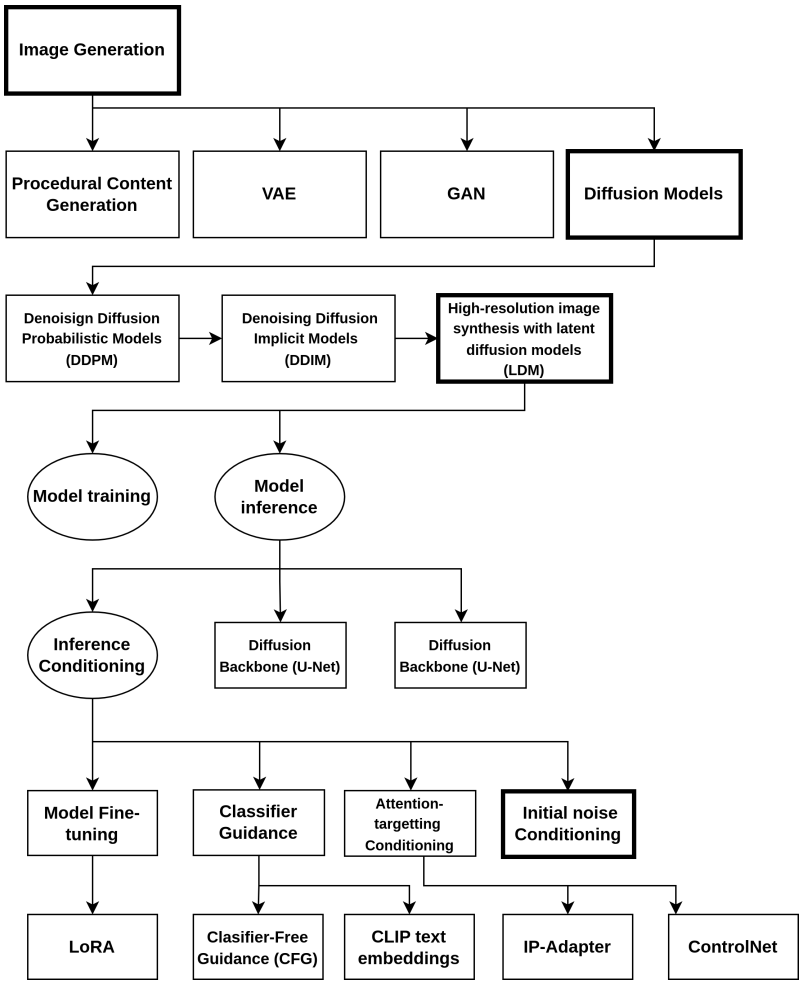


Figure 2.11: Concept map with background concepts and relationships.

what it consists of and detailing mechanisms that guide it, like CFG, text-guided CLIP conditioning, ControlNet conditioning, and LoRA fine-tuning. Finally, we explain the different roles that noise plays in LDM inference and inference conditioning. A concept map with the relationships between these concepts can be seen in Fig. 2.11.

Prior to this work, a systematic literature review on image generation within video game asset development was conducted using the PRISMA [52] framework; see Appendix A.2. It is currently under revision for publication.

Chapter 3

Solution

3.1 Problem Scope

In this Section, we further refine the problem description from Chapter 1. To develop games faster, one must develop assets faster. Assets are the digital visual or audio resources that result in a game's sensorial identity when combined. They represent the majority of the information communicated from game to player and are also essential in capturing the interest of would-be players. The difficulties associated with creating visual assets are manyfold. On the one hand, it means balancing functionality with aesthetic quality, as an asset must convey the information required by its context without that fact trumping its visual appeal. On the other hand, it must also stand by itself as a coherent piece of art, among other assets, in the context of the game section in which it lives.

Currently, generating isolated images on demand using diffusion models is easy: many public-domain trained frozen model weights with state-of-the-art performance are widely available [4, 18, 20], along with user-friendly conditioning methods, each suitable to their own family of problems (See Section 2.4.2). However, generating groups of images that share a common characteristic is a proven challenge. Due to the random starting point of the diffusion process, by definition, controlling abstract concepts such as style and theme is difficult.

3.2 Solution Scope

Due to the vastness of the problem at hand, we focus our research on the scope of a case study on the casino game maker Fabamaq [1]. Namely, we narrow visual game assets to 2D icons, which represent most of the assets used in the casino spins game genre, one of the companies' most significant products. The interface of a typical spins game can be broken down into the following elements, as illustrated in Fig. 3.1: static prizes screen (mainly composed of typography and simple button elements), the background and main screen area (typically includes the main subjects of the game or the entities that transmit the emotion of winning prizes), and the reel area.

[1] https://www.fabamaq.com/en

Figure 3.1: Picture of casino machine featuring the spins game "Easy Link."1: Prize screen; 2: Reel area; 3: Bet buttons and settings area.

The latter consists of columns of symbols that sequentially spin and stop on each round, resulting in a final combination determining the reward. Certain symbols or combinations can trigger special features, and the game's theme is usually reflected in the symbols' design. It is the area of the game screen where players will spend the most time looking at, and as such, is cardinal to the player's perception of the game's visual quality and differentiation. As such, we chose to tackle the task of automatically generating the icons that compose these reels. We call the collection of all icons composing a game the "icon pack."

3.3 Solution Requirements

The solution should accommodate the following functional requirements, shown in order of priority:

1. **Multiple assets must be able to be generated in a single run.** An essential aspect of the pack-generating nature of the solution must be the ability to generate multiple subjects with the same style and different content.

2. **Generated assets should be realistic and natural.** Coherence with real-world concepts like anatomical correctness, perspective, lighting, laws of physics and matter, etc., is paramount.

3. **Generated assets should respect the user input guidance.** A random generator serves no purpose, and a tool as such should provide meaningful avenues of conditioning. This requirement is further split into two sub-requirements:

    3.1. **Individually:** The user should be able to specifically condition the generation of every individual asset by some means in order to dictate its unique characteristics

3.2. **Collectively:** There should be a channel with which to condition the generation of all assets in a pack to inject coherency and consistency among the different subjects.

The environment and resources available for this study raise some requirements. The solution should not involve training, re-training, or other high-computational power and time-requiring operations, as we only had access to a machine with a single GeForce RTX 4070 graphics card, 32 GB of RAM, and a 13th Gen Intel(R) Core(TM) i7-13700K.

Now expanding on the hypothesis exposed in 1.3, we propose to achieve consistent style manipulation and flexible content control by combining a style transfer technique targeting the diffusion backbone with intentional noise manipulation in select areas of the initial latent image. This strategy is explained in Section 3.4.

3.4 "Masked Inverted Reference"

For style transfer, which is a problem that has been deeply studied in application to diffusion models, we decided to use IPAdapter [40], which has shown to be an effective plug-and-play method in zero-shot-transferring the style of one or a few reference images to the latent space of any trained diffusion model. In standalone use, we empirically observed that IPAdapter is capable of generating appropriately styled game icons. However, it proved unreliable in systematically transferring the stylistic features of a reference to multiple subjects, even with all other conditionings and variables fixed.

Our novel noise manipulation approach targets this segment of the process. The noise from which the diffusion process starts is known to be essential to the results' layout and content (see Section 2.4.2.3). Additionally, the success seen in text+image-to-image techniques shows the effectiveness of diffusion inversion in transferring an initial state to the diffusion process.

Our solution consists of a method that takes advantage of the reversibility of the diffusion process as a means to inject a predisposition towards a certain reference in the initial noise (See Fig. 3.2). Additionally, to ensure empty space in the reference image carries no information or predisposition, we mask said areas of the predisposed noise with interpolated, randomly generated noise. From this pipeline, we obtain a noisy image that is then encoded into the model's latent space and fed into the diffusion process. We empirically show in Chapter 4 that this approach positively impacts the transfer of attributes from the reference image without sacrificing control over the conditioning, as opposed to purely random or inverted latents.

3.4.1 Architecture

The solution architecture is illustrated in Fig. 3.3.

User inputs consist of a text prompt, where the description of each icon's content is provided (composition, expressions, poses, subject details, color, and even artist style inspirations can be given here), one or more style reference images, which will aesthetically influence the style of the resulting generations, and a noise seed, used to deterministically fix the random noise that is

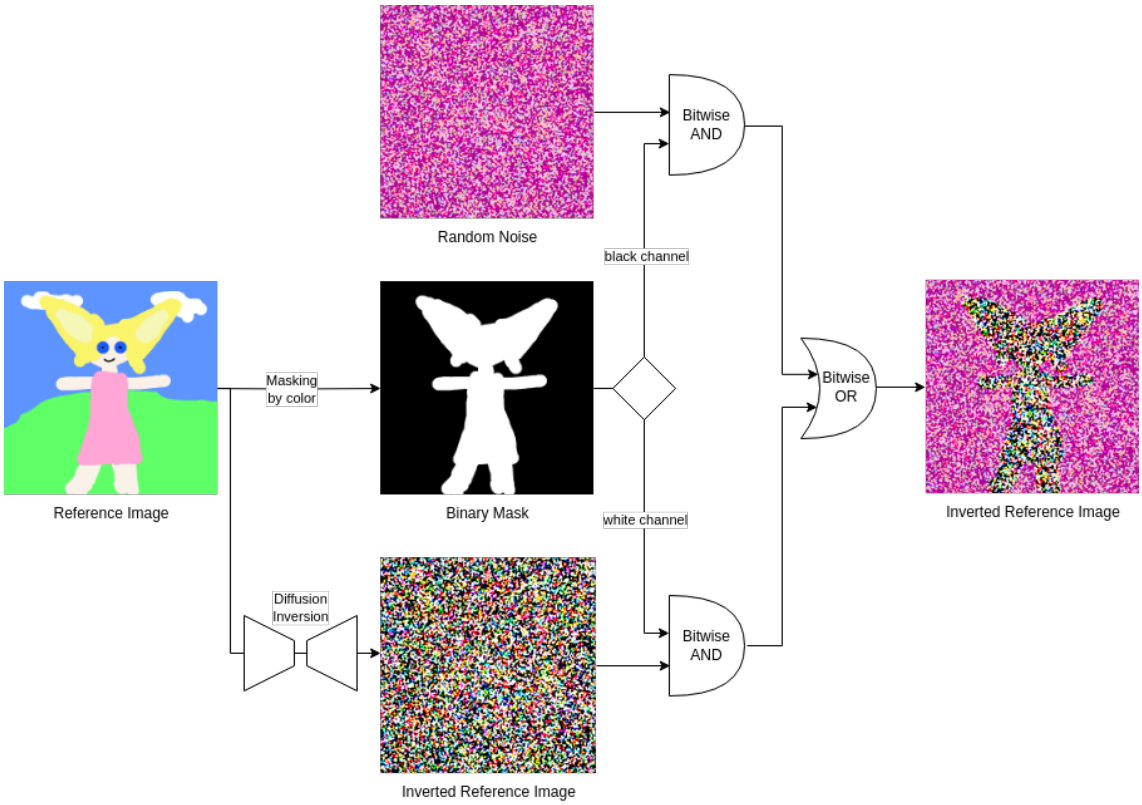


Figure 3.2: Pipeline of localized masking of inverted reference noise and random noise.

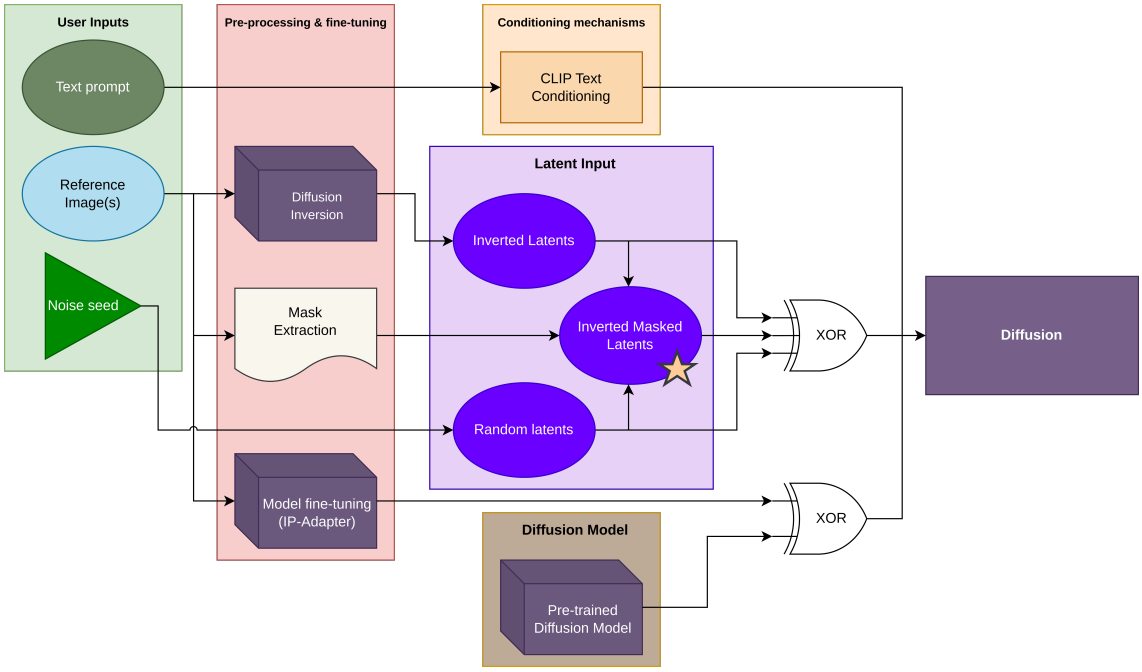


Figure 3.3: Solution architecture illustrating the flow of data and integrating stage depicted in Fig. 3.2 (marked with a star).

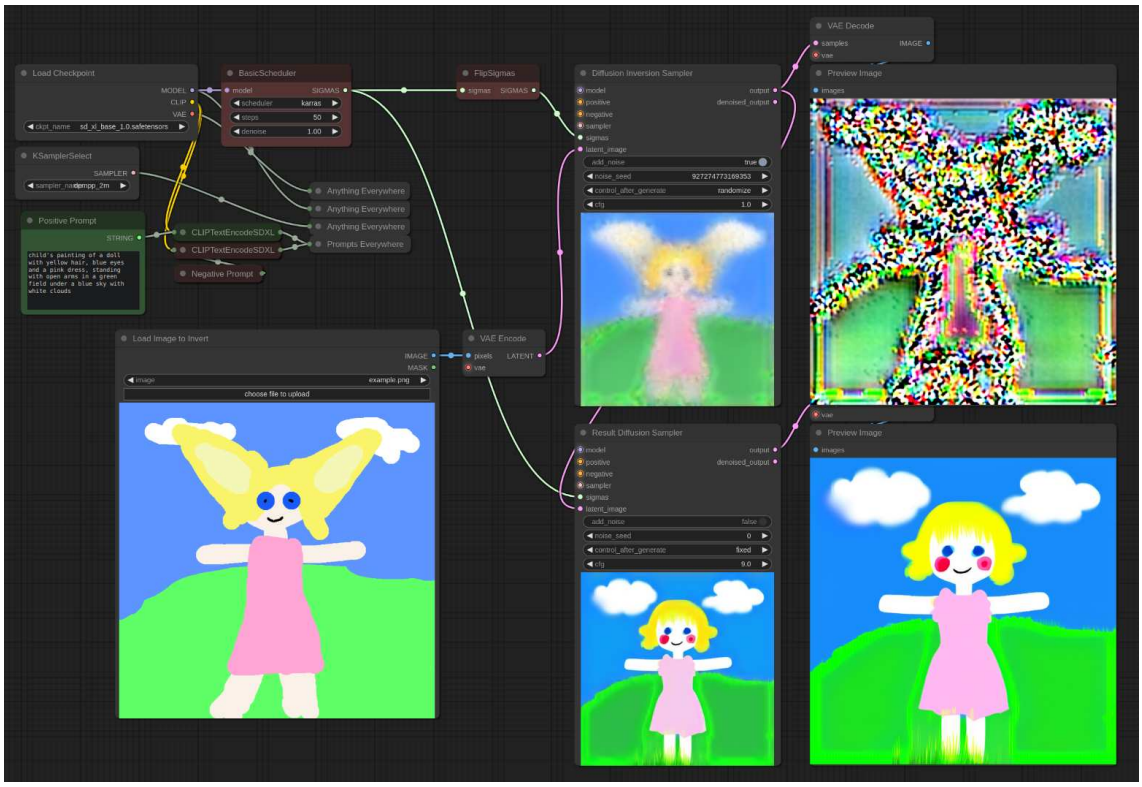


Figure 3.4: ComfyUI screenshot showing a simplified diffusion inversion pipeline. On the top, the reference image passes through a diffusion inversion process, resulting in a noised image which, when fed to the bottom forward diffusion process, results in a re-generation of the original reference image.

involved in the process. The text prompt is fed directly into a CLIP encoder to obtain a latent representation of the text, which is then used to condition the generation process. The reference image is first passed through a process of diffusion inversion (illustrated in an isolated example in Fig. 3.4), where using the same diffusion model but inverted noise sigmas (sigma-specific nodes are painted red in the figure), noise is progressively added, instead of removed, such that running the unconditioned inverse of this process on said noised image would result in the original references.

We tested multiple versions of Stable Diffusion models, from the earliest Stable Diffusion 1.5 [4] to SDXL [18] and SDXLTurbo [20], using the frozen model weights provided by HuggingFace [2]. We settled on using a community fine-tuning of the base SDXL model called Dreamshaper XL Lightning [3] as it consistently produced better results than SDXLTurbo, with the same base conditions and at the same number of steps (4 to 6), and comparable results to SDXL at less than a fifth of the number of steps (SDXL requires 30 to 50). See Appendix A.1 for results of the experimentation undertaken on the matter of the base model and its' configurations.

[2] https://huggingface.co/models?other=stable-diffusion
[3] https://huggingface.co/Lykon/dreamshaper-xl-lightning

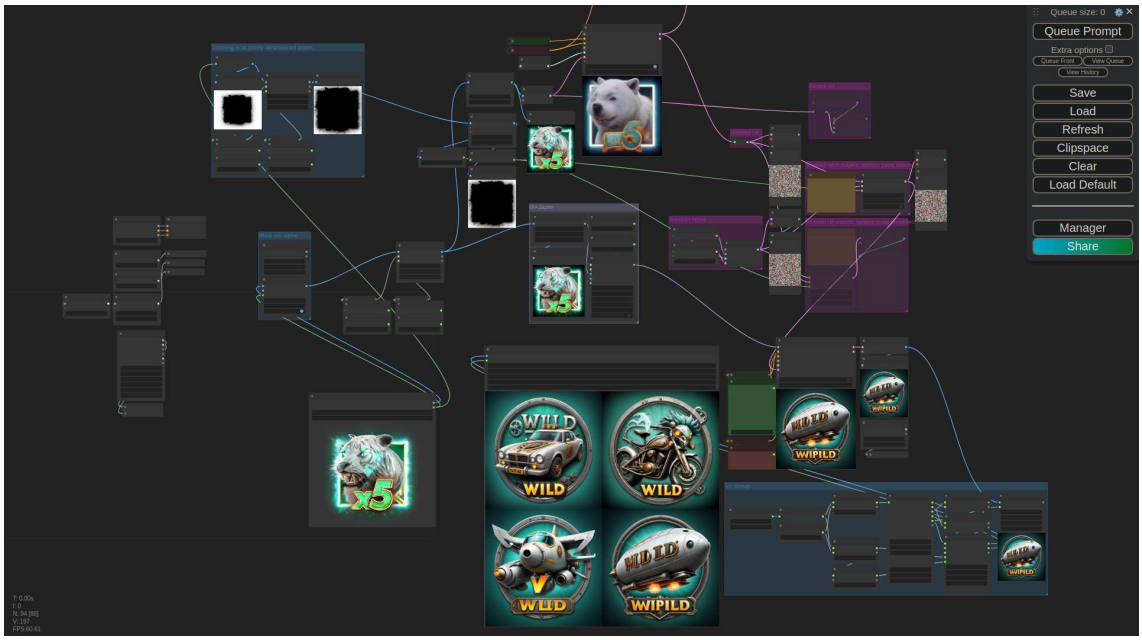


Figure 3.5: ComfyUI interface showing the testing pipeline.

The testing pipeline was created using ComfyUI [4], the open-source graph/flowchart/node-based interface officially recommended by StabilityAI. A screenshot of the interface showcasing this pipeline is illustrated in Fig. 3.5.

3.4.2 Reference Images

What we refer to as reference images are a set of one or more images that are used to influence the result of next generations in one of two conditioning methods: initial noise, through our novel Masked Inverted Reference, and style, through IPAdapter. Technically, the requirements for an effective reference image include one of the SDXL-friendly aspect ratios (square 512x512, and 1024x1024, portrait 512x640, 512x683, 512x768, and 512x896, or landscape 640x512, 683x512, 768x512, and 896x512). Additionally, the content of the image should be focused and well-placed. Cropped-out subjects or images with messy or crowded compositions won't be effective reference images. We chose to use Fig. 3.6, a company asset from the game Jin Qian Link [5], as our reference for the initial experiments and the evaluation, performed in Chapter 4.

In the following Sections, we detail each method's degree of control and the experiments undergone to determine their role in the solution design.

3.4.3 Content through Initial Noise

The initial noise, or initial latents, which represent the beginning of the diffusion process, contain stylistic, layout, and content information to some degree [50, 51], and can thus be leveraged as a

[4] https://github.com/comfyanonymous/ComfyUI

[5] https://fbmgaming.com/news/Jin-Qian-Link-the-slots-game-of-Asian-inspiration-that-has-players-on-the-edge-of-their-seats/129

Figure 3.6: Company asset used as reference image.

way of a-priori conditioning.

We tested the impact of multiple different means of feeding said noise into the diffusion process. Using the VAE provided by and trained on the Dreamshaper XL Lightning model checkpoint' [3] noise space, we encode the following variations of a single reference image into the latent space. In order to guarantee the replicability and validity of these experiments, besides seed and random noise percent, all other generation conditions were fixed, following the model creator's guidelines (5 steps, DPM++ SDE sampler with Karras scheduler, CFG = 2.0, reference image = Fig. 3.6, positive prompt = "white tiger video game icon", and no negative prompt).

3.4.3.1 Base Reference Image

The reference image is directly encoded into the latent space without noise manipulation and fed into the diffusion process. Due to the nature of the sampler used, the forward process is stochastic and thus can be seeded, which is also essential for the replicability of the experiment. The results, seen in Fig. 3.7, show the underlying inability to generate variation from a denoised image, as all images are very similar, lacking detail and proximity to the reference (Fig. 3.6).

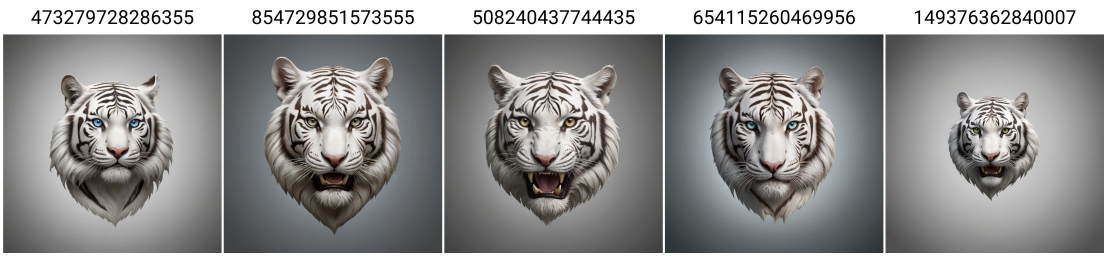


Figure 3.7: Images generated from base reference image (Fig. 3.6), with five different randomly chosen seeds.

Figure 3.8: Results of inverting and re-generating reference image, over five random seeds. Each row used the same seed for inversion and each column for generation.

From here forth, we use the same five randomly generated seeds for all experiments to ensure consistency and experiment replicability.

3.4.3.2 Inverted Reference Image

The reference image goes through the reverse process of noise addition, noise which is also seeded and fed back into the forward process. Since each of these steps is stochastic, we tested the effect of different and equal seeds, as seen in Fig. 3.8. We pass the reference image through the same model to invert it, with all conditions replicated, with the exception of noise sigmas (which are inverted) and text conditioning, which is removed. The effect evidenced by the image's diagonal, where both inverse and forward processes used the same seed, is the same as when feeding an image containing no random noise to the model, which is within reason as this results in no stochasticity added in the inversion.

Figure 3.9: Company asset reference image converted to .jpg.

For simplicity's sake, for the remainder of the tests, we chose to fix the inversion seed to '508240437744435' as it empirically yielded the most consistent result in the previous test (images in the third column are consistently closer to the reference's layout and elements).

Prompted by the fact that ComfyUI was internally converting the reference .png file (Fig. 3.6) into a .jpg file, we also tested the effect of the content with which to mask the initial noise. In Fig. 3.9 (obtained with `mogrify` [6]), we see left behind artifacts of information that were in the RGB channels but were hidden by some degree of transparency α, and were otherwise invisible to human eyes.

Since these artifacts were being fed into the inversion process unnoticed, we tested the effect of replacing the areas of the image with $\alpha \neq 0$ in the transparency channel with solid color, as seen in Fig. 3.10. Generally, saturated colors had a more significant impact, with yellow and red presenting in most results as a solid border and blue presenting in most results as a background light or effect. Empirically, we fixed this color to black for the remainder of the experiments to inject the minimum variability possible. Further experimentation on this matter was carried out post-test; see Section 5.1.

3.4.3.3 Noised Inverted Reference Image

To inject stochasticity into the inverted reference, we blend it with random noise in the latent space. This is done using a linear interpolation between both matrices I_{ref} and I_{noise} (which are the matrix representation of the reference and the random noise images, respectively), given a percentage α (which we call noise ratio), as per expression 3.1.

It is worth noting that we considered other processes for accomplishing the blending itself, namely traditional image blending techniques (multiply, overlay, color burn, linear dodge, ... [7]); however, due to implementation difficulties associated with the ComfyUI interface, these attempts

[6] https://imagemagick.org/script/mogrify.php
[7] https://helpx.adobe.com/photoshop/using/blending-modes.html

Figure 3.10: Results of replacing transparency channel with different solid colors. The first row is the reference image resulting from the operation; remaining rows are corresponding generations with same seed.

Figure 3.11: Results of interpolation between inverted reference and noisy image, over five random seeds. Each row used the same seed for generation, and each column used the same noise ratio. All inversions used the seed '508240437744435'

were abandoned.

$$I_{int} = (I_{ref} \times \alpha + I_{noise} \times (1.0 - \alpha)) \tag{3.1}$$

The results in Fig. 3.11 show that the visual variation obtained isn't proportionate to the visual amount of noise added in each column, as there is almost no perceivable difference in the results between the extremes. To combat this, as seen in expression 3.2, we normalize the matrices before interpolating (T) and re-scale the result to the original magnitude I_{scaled_int}.

$$\begin{aligned} T &= (\frac{I_{ref}}{||I_{ref}||} \times \alpha + \frac{I_{noise}}{||I_{noise}||} \times (1.0 - \alpha)) \\ I_{scaled_int} &= \frac{T}{||T||} \times (||I_{ref}|| \times \alpha + ||I_{noise}|| \times (1.0 - \alpha)) \end{aligned} \tag{3.2}$$

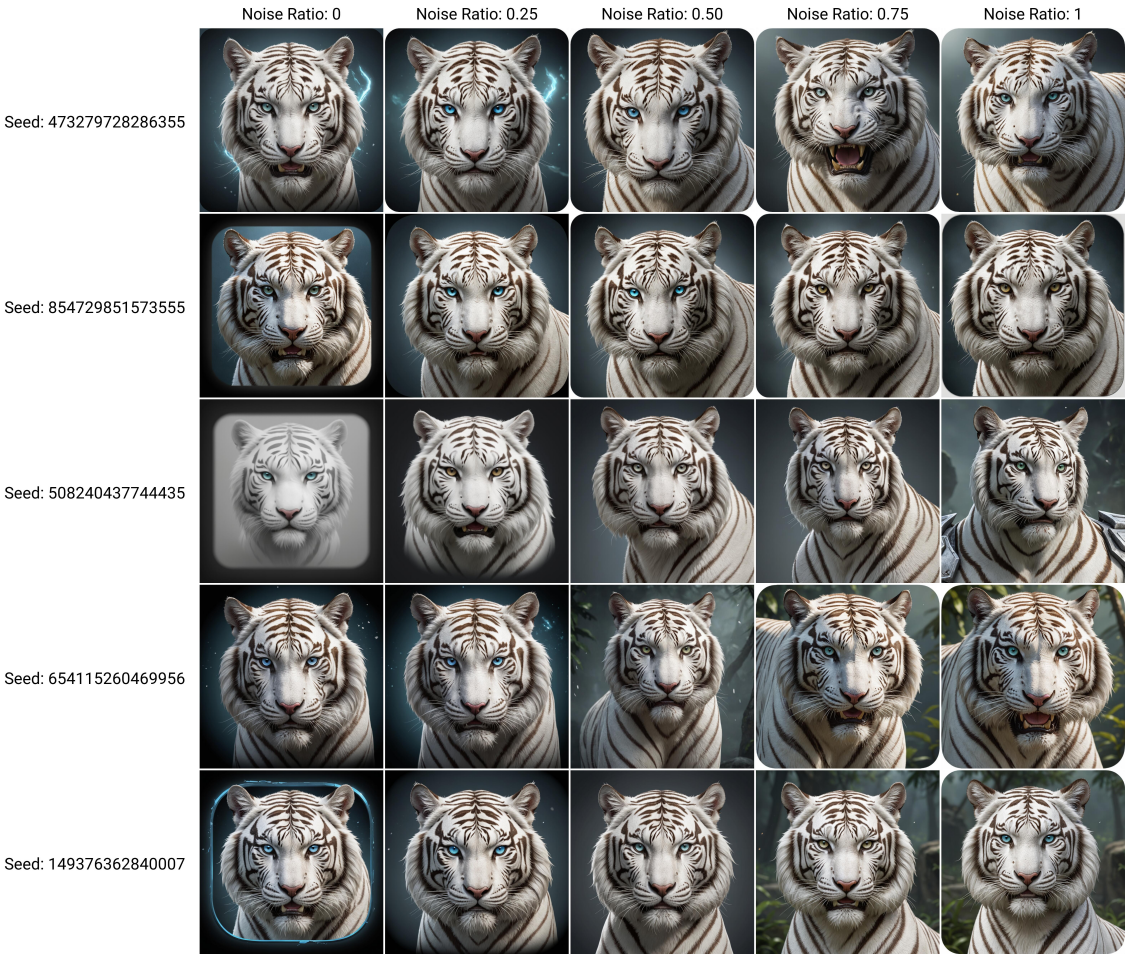


Figure 3.12: Results of normalized interpolation between inverted reference and noisy image, over five random seeds. Each row used the same seed for generation, and each column used the same noise ratio. All inversions used the seed '508240437744435'

The results, as seen in Fig. 3.12, despite showing a higher influence of the added stochasticity in the progressively more creative nature of results as noise ratio increases, still lack the desired degree of variation. Thus, for this case, we conclude that adding indiscriminate stochasticity through random noise isn't beneficial for obtaining more creative results when starting from an inverted reference. Additionally, even though no conditioning towards this was used, the image's background started diverging from the prompt (traces of jungle backgrounds are visible in the last two rows and three columns, and metal shoulder pads are visible in the third row and last column).

3.4.3.4 Our Masked Inverted Reference

In continuation of the previous test's last-mentioned flaw, we propose another way of injecting stochasticity into inverted latents. Let us consider the reference image as a composition of both important and unimportant areas. Be the latter empty space, irrelevant details, or background, these areas, once inverted, should not influence the generation process. Thus, we propose a method (better described at the beginning of Section 3.4) for controlling the areas where stochasticity is injected. Results of its application, maintaining all previous conditions, are illustrated in Fig. 3.13 (as in previous tests, all inversions used the seed '508240437744435'). For this reference image, the mask is the same as its transparency channel. We observe a higher degree of consistency and adherence in the results with a higher noise ratio (0.5 to 1). The inconsistencies found in Fig. 3.12 are no longer present, although the results arguably still lack creativity and variation.

Compounded is the inability to control style, as evidently, all but some of the images in Fig. 5.4 are in a photorealistic illustration style (which is Dreamshaper's default emerging style, by design). This is because the references' predisposition isn't directed enough to condition towards the deep-rooted features of the style. Text conditioning and the model backbone itself are the remaining possible avenues to tackle this problem, which we further test.

3.4.3.5 Text Conditioning

It is often difficult to describe the intrinsic features of a style or aesthetic in words, which is compounded by the fact that such technical attributes are rare and misrepresented in the web-crawled data these models' text encoders are trained on, resulting in generation failures [53]. We show in Fig. 3.14 the result of combining the previous techniques with a more descriptive prompt (as in previous tests, all inversions used seed '508240437744435' and noise ratio was fixed to 0.75):

"detailed smooth fur, shiny green translucent eyes and lightning, white tiger growling and showing its teeth, side profile shot, video game icon".

Despite the better adherence to the extra direction, it is still not enough to fully describe the reference's style. It also becomes clear that the most interesting reconstruction results from Fig. 3.14 are those from the second and last rows, although the remaining results, too, can be considered positive.

Figure 3.13: Results of inverting, noising, masking, and re-generating reference image, over five random seeds. Each row used the same seed for generation, and each column used the same noise ratio.

Figure 3.14: Results of combining a descriptive prompt with previous techniques over five random seeds. Each row used the same initial noise manipulation, and each column used the same generation seed.

By comparing the images generated throughout these experiments with the reference image in question 3.6, it is evident that none of the results represent a satisfactory transfer of the style of the reference, as the most prominent features of the tiger, like the rendering of the fur, the green lighting effects in the eyes and background, and the expression of the subject itself are not present in the former. This leads us to conclude that initial noise manipulation techniques, both standalone and in combination with text conditioning and seed manipulation, do not constitute the required stylistic control for our solution.

3.4.4 Style through Attention

In tackling the flaws identified in the previous experimentation, we propose using attention-targeting methods.

As described in Chapter 2, many works [54, 39, 40] have tackled the problem of style control in diffusion models through attention mechanisms. For our purposes, we chose an existing ComfyUI implementation of IPAdapter [8].

We tested the effectiveness of IPAdapter in transferring the style of a reference image to the latent space of the model. IPAdapter is a plug-and-play module that conditions the model's latent space with the style of one or more reference images. It does so by encoding the reference images into embeddings the model can understand, using a CLIP vision encoder (in our experiments, we used the OpenCLIP-ViT-bigG-14 [9], the largest checkpoint recommended by the authors), and feeding those embeddings in a novel way directly to the cross-attention layers.

The model used for the remainder of the work is the base ip-adapter_sdxl [10] adapter version, as it's the version compatible with the SDXL model checkpoint we settled on using.

Images consumed by the adapter must be of 512x512 resolution or otherwise resizable or croppable to respect the aspect ratio. Centered subjects work best.

First, we study the impact of noising the reference image before training the IPAdapter module. Following the same logic as before, noise should represent a tradeoff between loyalty to the reference and creativity in the results (see Fig. 3.15). However, the evidenced impact is minimal.

We then combined the previously proposed initial noise attainment techniques with IPAdapter conditioning, with all conditions from 3.14 replicated except the model (see Fig. 3.16). Results are, as expected, of similar composition and layout to the previous results using noise manipulation alone but now feature a flashier style and, in some cases, small corner elements and numbers, as in the reference image.

3.5 Summary

In this chapter, we detailed the problem statement, scope, and requirements of our research and proposed a solution for the problem of generating sets of casino game icons with consistent styles

[8] https://github.com/cubiq/ComfyUI_IPAdapter_plus
[9] https://huggingface.co/laion/CLIP-ViT-bigG-14-laion2B-39B-b160k
[10] https://huggingface.co/h94/IP-Adapter/blob/main/sdxl_models/ip-adapter_sdxl_vit-h.bin

Figure 3.15: Results of varying degrees of noise over reference image before IPAdapter training. Each column used the same noise ratio, each row used the same generation seed, and all generations' initial noise was random.

Figure 3.16: Results of combining IPAdapter with previous techniques. Each column used the same seed, and each row used a different initial noise manipulation technique. All inversions used seed '508240437744435'.

among each other, all the while respecting user input guidance.

Developing the prototype requires knowing, to a degree, what mechanisms impact the results and how. Methodological experimentation of each segment of the proposed architecture was undertaken, and the resulting generated icons were qualitatively analyzed in order to decide on the best configuration. The final solution consists of a novel combination of a state-of-the-art style transfer technique with manipulating the diffusion's initial noise in order to inject directed predispositions in select areas of the result.

Next, in Chapter 4, we quantitatively evaluate each initial noise manipulation technique in combination, both with and without IPAdapter style conditioning, through the application of a user study. We also qualitatively analyze handpicked results with domain expert inputs.

Chapter 4

Prototype Evaluation

Evaluating the quality of image generative models means assessing the quality of the results they produce and their usability. The former can be done qualitatively, with the help of human judgment, and quantitatively, through visual quality metrics. In contrast, the latter is commonly done through User Experience (UX) tests.

This chapter presents the process and methodology undertaken to evaluate our solution in these terms through a qualitative user study in Section 4.1, presenting and analyzing the results obtained, as well as some qualitative comparisons of results obtained throughout the research in Section 4.2, providing a detailed analysis of positive and negative aspects of some hand-picked generations. In the next Chapter 5, we discuss the findings and suggest possible improvements to the solution.

4.1 User Study

A user study was conducted on a population of 48 students from the Faculty of Engineering of the University of Porto and 23 visual artists from Fabamaq (71 total participants). Before the test, all users agreed to an informed consent form, which explained the purpose of the study, the fact that no personal identifying data besides age would be collected, and that they had the right to opt out of the study at any point in time. Its purpose was to ascertain the qualitative performance of our prototype, an image generation pipeline, in generating "effective" video game icon packs from a set of prompts. An icon pack, as exemplified in Fig. 4.1, is a set of symbols used to represent different, coexisting concepts in a game. We arbitrarily set the number of icons in a pack to four, although this number is closer to 16 or more in a real game. This results in a 2x2 grid of 1024x1024 prototype-generated casino game symbols.

The definition of an "effective" video game symbol pack, which we constructed in line with company artists, can be split into three concepts:

Realism: Each symbol, by itself, should be consistent with real-world representations of their respective prompt (e.g., human hands should have five anatomically correct fingers, flowing water should appear to obey the laws of physics, etc.). This is the first aspect by which a generated image should be deemed effective.

Figure 4.1: Example of an icon pack selected for the user study, generated using the command "[man | woman | bird | pear] video game icon"

Style continuity: Among themselves, icons should appear to belong in the same continuum stylistically (e.g., a pack containing a cat, a coin, an apple, and a car, which are subjects from very different domains, should be visually consistent among themselves). This concept should not trump the former.

Negative Space: In the realm of small game icons, especially those that exist in complex and crowded compositions like casino spin games, there must be a balance between visual appealingness and functionality. Negative space, also known as white space, is the Gestalt Principle [55] that refers to the empty or unmarked areas in a design or composition that balance and enhance the visual impact of the positive elements within it. The presence of this concept in each set of icons should be recorded, along with the impact of it on readability. Other principles' inclusion in our study was considered but later excluded as they did not apply to the nature and context of the video game icons developed by our case study's company (e.g., continuity, which states that our perception groups elements that seem to follow a continuous path in a particular direction, or proximity, which states that we group closer-together elements, separating them from those farther apart). Testing our solution in an isolated and scientific manner would be difficult if based on these metrics, which are good examples of inter-subject design principles. Thus, we narrowed our focus to intra-subject principles and reduced the list to only negative space, per visual artist expert advice.

4.1.1 Test Variables

We define the test dependent and independent variables by referring back to Section 3.4.1. Since we wish to test the impact of noise manipulation and diffusion-targeting style transfer techniques on the quality and effectiveness of generated results, these are our independent variables:

1. **Backbone:**
    1.1. *Vanilla*: Base model weights (Dreamshaper XL Lightning [3]).
    1.2. *IPAdapter*: Base model, fine-tuned on the reference image.
2. **Initial Noise:**
    2.1. *Seeded Random Noise*: Start the generative process from a random Gaussian sample.
    2.2. *Inverted Reference*: Start the generative process from the noise obtained via diffusion inversion of the reference image. This process is also seeded, as the same stochastic sampler is used in both inverse and generative processes.
    2.3. *Masked Inverted Reference*: Combining the previous two, start the generative process from a targeted composition of both noise sources.

The six combinations of these variables compose the different prototype configurations used to generate the tests' icon packs. The resulting images are shown in Fig. 4.2.

All other avenues of conditioning are fixed, including the prompt command, the inversion and generation seeds (0 and 8, respectively), and the reference image 3.6. Otherwise, independent variable influence could not be determined. Additionally, the subjects selected for the prompt command *"[man | woman | bird | pear] video game icon"* were chosen to be neutral and common enough so as not to affect the model's ability to represent them [53]; all the while being different enough amongst each other so as to provide a good range of intra-icon-pack variability.

4.1.2 Questionnaire

Respondents are first asked to agree to an informed consent form, which explains the purpose of the study, the data collected, and the respondent's rights regarding their data (although no identifying personal information is collected). Then, as part of a pre-test questionnaire, respondents' age and expertise in the design area are collected. The main questionnaire consists of 2 sections of opinion questions, where for each question, we show an image and some opinion statements and ask the user to answer on a Likert Scale from 1 (Strongly Disagree) to 5 (Strongly Agree). The first section comprises four opinion questions for each of the six generated icon packs, individually evaluating the concepts of realism, prompt adherence, negative space, and style continuity (Fig. 4.3).

The final question of the form is a multiple choice question, where upon being shown first the reference image, then all six generated icon packs, respondents are asked: "If you had to choose just one pack, which one would you put in a game, considering the aesthetic features of the reference image above?". See the full questionnaire document in Appendix A.3

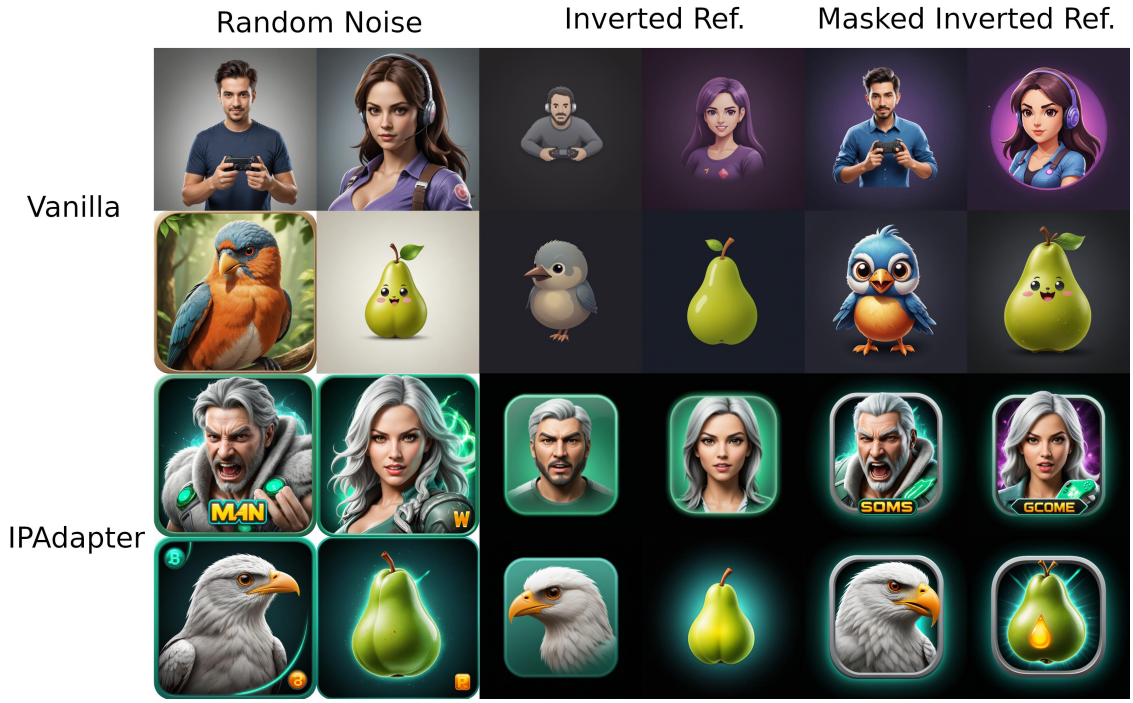


Figure 4.2: Icon packs evaluated in the user study, generated using the command "[man | woman | bird | pear] video game icon", and reference image 3.6.

4.1.3 Statistical Analysis

4.1.3.1 Demographics

We analyze and extract conclusions from 71 collected questionnaire answers in this Section. Firstly, we present the distribution of answers to two questions from the pre-test questionnaire in Fig. 4.4.

Since the participant population mainly consisted of engineering students, the age distribution (Fig. 4.4a) is skewed towards young adults, with 59% of the participants (42) aged between 18 and 25. For the same reason, we see a skewed distribution toward participants with less design proficiency (Fig. 4.4b), with 14 and 24 participants self-reported with none and low levels of design proficiency, respectively.

4.1.3.2 Opinion Questions

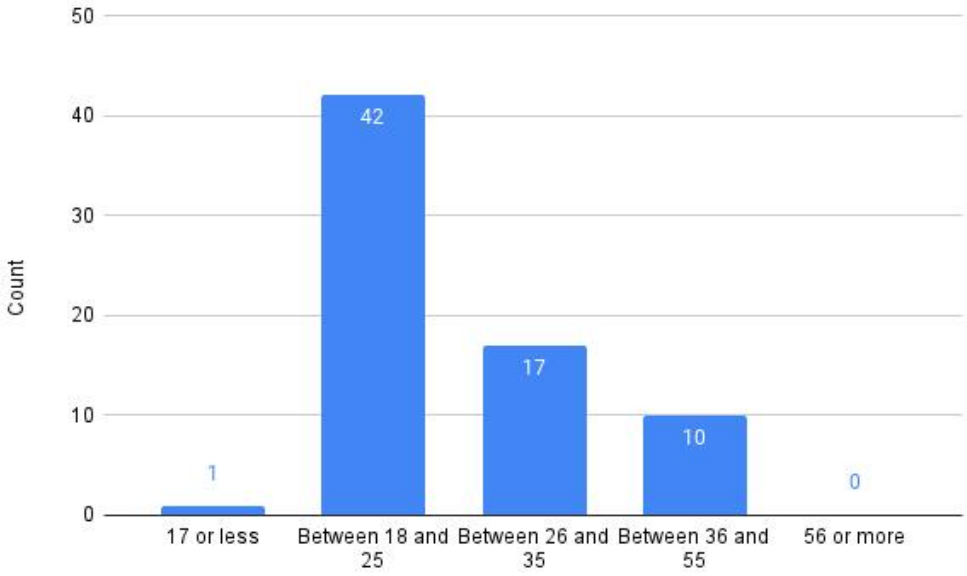
The mean of the answers to the six opinion questions, all similar to Fig. 4.3, are presented in Table 4.1. Internally, the Likert scale responses are represented from 1 ("Strongly Disagree") to 5 ("Strongly Agree").

Before performing statistical tests, one must verify the required assumptions and act accordingly. Analysis of Variance (ANOVA) tests are parametric tests that assume the data is normally distributed and has equal variance in each group that is tested. Considering the two independent variables, 'Backbone' and 'Initial Noise,' and using the Shapiro-Wilk test on the distributions of

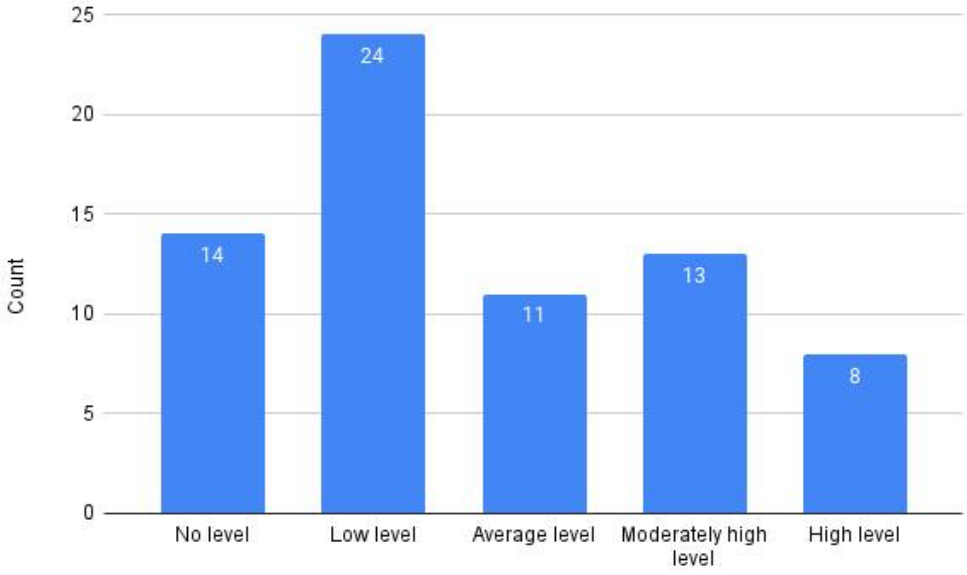
Generated Icon Pack *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Figure 4.3: Example of a question from the user study's first section.



(a) Participants' age distribution.



(b) Participants' design proficiency distribution.

Figure 4.4: Bar plots showing answer count for age and design proficiency questions.

Table 4.1: Mean responses for each statement across the six icon packs. The highest values for each statement are in bold.

		Statements			
Backbone	**Initial Noise**	*"... appear natural and realistic"*	*"... respect the text prompt"*	*"... effectively utilize negative space"*	*"... stylistically coherent ..."*
Vanilla	*Random*	2.83	3.69	3.41	2.17
	Inv.	2.23	3.49	2.90	2.46
	Masked Inv.	2.52	3.63	3.38	2.63
IPadapter	*Random*	**3.15**	**4.23**	**3.87**	**4.04**
	Inv.	2.92	4.00	3.59	3.90
	Masked Inv.	3.13	4.03	3.70	3.44

responses to each statement and variable configuration, we found that all statements show significant p-values (far below 0.05), indicating that the data is not normally distributed. Similarly, using a Leven test, we found unequal variance for backbone in the second statement and unequal variance for noise in the third statement. Since these do not hold, we must resort to non-parametric tests like the Kruskal-Wallis or Wilcoxon Signed-Rank tests. The former, however, requires inter-group independent sampling, which our user study does not have. In hindsight, our user study should have been designed around this fact, as it would have allowed for more powerful statistical tests. Testing the assumptions for the latter method, we plotted the data skewness for each statement and each pairwise combination of independent variable groups, seen in Table 4.2. Cell values are color-coded, with green indicating symmetric distributions, yellow indicating moderate skewness, and red indicating high skewness. Despite the high skewness mostly in Table 4.2b, we proceed with the Wilcoxon Signed-Rank test, as the alternative sign test is much too weak for our purposes. Thus, we acknowledge the possible incorrect conclusions that may arise, especially from the analysis of the second statement.

Significance results (after Bonferroni correction to counteract the increased likelihood of Type I error that arises from multiple comparisons) are displayed, per statement, in Tables 4.3, where significant cells (p-value < 0.005) are colored in gray. Additionally, significant pair cells show the effect size (r) in red and in parentheses, where $r = \frac{Z}{\sqrt{N}}$, Z is each pair's Wilcoxon signed-rank test statistic, and N is the number of samples (71). A value of r between 0.3 and 0.5 signifies a medium effect, and above 0.5 indicates a large effect.

In Table 4.3a, we see that the users perceived a statistically significant difference in realism between the masked inverted noise results and the random and inverted noise results between both Vanilla-IPAdapter and within IPAdapter. The difference between IPAdapter+Random and Vanilla+Inverted/Masked Inverted noise results was also significant, with similarly medium effect size values (averaging 0.497).

Users' perception is less consistent regarding prompt adherence, as seen in Table 4.3b, with spread consensus between groups. Differences within backbones were only significant for the

Table 4.2: Paired sample skewness for each statement's results. Cells are colored green, yellow, and red for symmetric, moderate, and high skewness values.

(a) "*The generated images appear natural and realistic*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	-0.03	0.95	-0.15	0.53	0.29
	Inv.		-0.36	0.93	-0.11	1.02
	Masked Inv.			0.42	-0.34	0.54
IPAdapter	*Random*				-0.33	0.17
	Inv.					0.22

(b) "*The generated images respect the text prompt*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	0.27	-1.28	0.12	-0.4	0.2
	Inv.		-1.34	0.52	0.2	0.5
	Masked Inv.			0.8	1.26	0.99
IPAdapter	*Random*				-0.59	0.42
	Inv.					1.1

(c) "*The generated images effectively utilize negative space*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	0.37	-0.2	-0.06	0.02	-0.11
	Inv.		0.37	0.54	0.32	0.07
	Masked Inv.			0.17	-0.46	-0.22
IPAdapter	*Random*				-0.39	0.23
	Inv.					-0.36

(d) "*The generated images are stylistically coherent with each other*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	0.73	0.84	-0.48	0.39	0.12
	Inv.		-0.47	-0.43	0.14	-0.43
	Masked Inv.			-0.25	0.59	-0.03
IPAdapter	*Random*				-0.33	0.06
	Inv.					0.05

Table 4.3: Wilcoxon signed-rank test results for each statement, in black, and effect size value, in red. Grey-colored cells represent significantly different group combinations.

(a) "*The generated images appear natural and realistic*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	1.000	0.217	0.621	0.563	0.011 (0.40)
	Inv.		0.387	0.010 (0.40)	1.000	0 (0.55)
	Masked Inv.			0 (0.52)	1.000	0 (0.58)
IPAdapter	*Random*				0.001 (0.47)	0.740
	Inv.					0 (0.56)

(b) "*The generated images respect the text prompt*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	0.174	0 (0.52)	1.000	0.151	1.000
	Inv.		0.106	0.007 (0.41)	1.000	0.003 (0.44)
	Masked Inv.			0 (0.50)	0.156	0 (0.56)
IPAdapter	*Random*				0.036	1.000
	Inv.					0.002 (0.46)

(c) "*The generated images effectively utilize negative space*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	1.000	0.041 (0.36)	1.000	0.824	0.016 (0.39)
	Inv.		0.607	1.000	1.000	0.001 (0.50)
	Masked Inv.			0.024 (0.37)	1.000	0 (0.50)
IPAdapter	*Random*				0.294	0.023 (0.38)
	Inv.					0.001 (0.48)

(d) "*The generated images are stylistically coherent with each other*"

		Vanilla		*IPAdapter*		
		Inv.	*Masked Inv.*	*Random*	*Inv.*	*Masked Inv.*
Vanilla	*Random*	0. (0.78)	0 (0.82)	0.004 (0.43)	0 (0.66)	0.636
	Inv.		1.000	0 (0.67)	0.065	0 (0.70)
	Masked Inv.			0 (0.72)	0.001 (0.47)	0 (0.74)
IPAdapter	*Random*				0 (0.53)	1.000
	Inv.					0 (0.59)

cases Vanilla+Random and Vanilla+Masked Inverted and for IPAdapter+Inverted and IPAdapter+Masked Inverted. Between both backbone groups, differences between both masked inverted results, Vanilla+Inverted and IPAdapter+Random/Masked Inverted, and Vanilla+Masked Inverted and IPAdaptter+Random/Masked Inverted were also significant. The effect size is lower than in the previous statement but still signifies medium impact (0.405).

In terms of negative space perceived effect, in Table 4.3c, users consensually perceived differences between IPAdapter+Masked Inverted and all

Finally, in Table 4.3d, users perceived stylistic differences between IPAdapter+Random and all other configurations except IPAdapter+Random noise. They also perceived differences between Vanilla+Random and all but IPAdapter+Masked Inverted configurations, as well as between Vanilla+Masked Inverted and between Vanilla+Inverted configurations.

4.2 Qualitative Comparison

To complement the user study's results, we present some hand-picked generation failures and successes, explaining why the former don't satisfy the problem requirements and highlighting how the latter do.

Figs. 4.5 show the company assets used as reference images for the following generations but were not included in the user study.

A common problem that arises from using IPAdapter is that the transfer isn't directed enough to control what elements of the reference should be present and to what degree. In Fig. 4.6a, which we generated while constructing the user study, the combined effect of too high a strength on the IPAdapter conditioning and too rare a target subject in the prompt can be seen. Within the model's inner representation of the world, due to the nature of the training data, some subjects are over-represented, while others are under-represented. We can infer that terms like man and woman are common occurrences in the Dreamshaper training data, while pear probably isn't as common, leading to the displayed effect of the reference tiger's features (open mouth, teeth, eyes, fur) bleeding into the generated subject.

This phenomenon may also be present when using different reference images and different rare concepts (see Figs. 4.6).

In contrast, Figs. 4.7 exemplify successful individual generations, where the model correctly transfered the reference style and shape while adhering to the prompt in the different subjects' generations, but failed to do so from the same random seed for all the four pack's subjects.

Figs. 4.8, like the ones used in the user study, represent successful pack-wise success.

Another aspect that impacts effectiveness is the presence of text elements. Since coherent text generation is a known limitation of the SDXL model architecture [18], the evaluation process did not consider this a success metric; however, malformed or incorrect text still influences the viewers's perception of quality, a fact that was confirmed in a post-test environment with some

(a) Golden dragon, from Jin Qian Link.

(b) Red phoenix, from Jin Qian Link.

(c) Asian green dragon, from Reel Strike.

(d) Red gem, from Reel Strike.

Figure 4.5: Company assets used as reference images not included in the user study.

(a) Reference image: Fig. 3.6.

(b) Reference image: Fig. 4.5a.

(c) Reference image: Fig. 4.5c.

Figure 4.6: Examples of IPAdapter generation failures, for different reference images (prompt: *"pear video game icon"*)

(a) Prompt: *"woman video game icon"*. (b) Prompt: *"pear video game icon"*.

Figure 4.7: Examples of successful, stylistically coherent generations of different subjects (reference image: Fig. 4.5d)

participants. The current state-of-the-art Stable Diffusion 3 [56] significantly improves consistency in this matter, and we believe its incorporation with our backbone-agnostic solution would greatly enhance it.

Figs. 4.9 show examples of both text generation success and failure, resulting from all the same conditions but the prompt. We did not investigate ways to alleviate or counteract these issues; we believe this could be the subject of future work.

Other than in the results, text elements also impacted our solution's effectiveness when present in the reference image. Figs. 4.10 highlight this intrinsic flaw of our solution, where unwanted, unhandled elements in the reference image (like the "x5" text seen in Fig. 3.6) can sometimes produce artifacts in that area of the result. We believe that appropriate "masking out" of these elements would decrease the likelihood of these artifacts appearing, and we also reference this as possible future research, as we could not find a reliable way to execute dynamically said masking.

4.3 Summary

In this chapter, we presented the results of our user study, which aimed to evaluate the effectiveness of our solution in generating video game icon packs, measuring the impact of different solution configurations on four problem-specific image quality metrics. We also acknowledge some limitations of our study, such as the small sample size and the non-normality of the collected data, which reduces the statistical soundness of our conclusions. We also presented qualitative comparisons of successful and failed generations on different aspects. Chapter 5 discusses our findings' implications and notes aspects that could improve the solution.

(a) Prompt: *"man video game icon"*.

(b) Prompt: *"woman video game icon"*.

(c) Prompt: *"bird video game icon"*.

(d) Prompt: *"pear video game icon"*.

Figure 4.8: Examples of entirely successful icon pack generation (reference image: Fig. 4.5b)

(a) Prompt: *"bird video game icon"*. (b) Prompt: *"woman video game icon"*.

Figure 4.9: Examples of positive and negative text element generation (reference image: Fig. 3.6)

(a) Prompt: *"pear video game icon"*. (b) Prompt: *"apple video game icon"*.

Figure 4.10: Examples of initial-noise induced generation failures (reference image: Fig. 3.6)

Chapter 5

Discussion

Firstly, we note that the applied user study has some implicit flaws. Despite the isolated application of the questionnaire to unbiased participants who had never seen any of the generated results before, using an existing company asset as the reference image may have biased the opinion of some artists (6 of the 23 participated in developing its game, but it is improbable any of them didn't see the image at least once, at some point, throughout development and after release). We also note that the questionnaire would have been better applied to a population with a more uniform distribution of design expertise (by, for example, distributing it among art and design students instead of engineering students). Additionally, we acknowledge now that the readily accessible expert opinions from company artists would have been better applied to fewer directed expert interviews instead of including them indiscriminately in the user study.

Post-hoc, we also realize that evaluating the packs as a whole, without assessing the individual images, hampered the participant's expression of their opinion and reduced the granularity of the results.

We also note that despite the introduction of the concepts before the opinion questions, our usage of the concepts "naturalness" and "realism" was unsuccessful, as most users interpreted it in the sense of "real-world" stylistic realism. This is evident in Table 4.1's third row, having the highest mean responses for the statement evaluating this concept (3.15).

Discussing the mean response results, it is evident that the IPAdapter backbone produces preferable results over all four statements. We conclude that the stylization that emerges from conditioning the model on a vibrant, colorful, stylized game icon positively impacts results on realism, prompt adherence, negative space utilization, and intra-pack stylistic coherence.

We now elaborate on the meaning of the significance test results in Table 4.3.

The IPAdapter backbone variable consistently influenced the participants' perception in all four statements: of the 31 significant configuration pairs among the four statements, 27 contained at least one IPAdapter. This does not directly translate to better results but further reinforces the conclusions extracted from Table 4.1. Additionally, 22 of the 31 significant configuration pairs included at least one Masked Inverted Noise result, which is evidence that this variable also significantly impacts the users' perception of visual quality, especially in terms of negative space

utilization and stylistic coherence. In comparison, 18 and 16 of the 31 pairs contained at least one random noise and one inverted noise configuration, respectively. This translates to a 13.5% and 20% higher rate of impactful configurations of our novel initial noise configurations' impact over the traditional alternatives. On all but the negative space statement (Table 4.3c) where there was an equal share, most significant configurations contained IPAdapter over a Vanilla backbone. Moreover, Masked Inverted Noise represented the majority of impactful configurations on all but the stylistic coherence statement (Table 4.3d), where Random Noise appeared in 9 of the 11 significant pairs.

Due to the nature of the Wilcoxon signed-rank test, we cannot infer the direction of the impact, and further testing (through another user study, namely) is needed to determine, with certainty, its superiority over the alternative initial noise configurations. A Kruskal-Wallis test would be a powerful alternative if the response data met the required assumptions.

With recourse to the expert opinion of the company artists in a post-experiment environment, we now discuss some of the results on a case-to-case basis.

Comparing the results of generating from inverted versus masked inverted noise, as seen in Fig. 5.1, we see a lack of expression and dynamic range in the former. The poses of the human characters are stiff and bland, and despite obeying the layout restrictions of the reference image's shape, the backgrounds are uninteresting and lack depth and detail. On the other hand, the former's results are much more dynamic, with colorful backgrounds, flowing hair in the case of the woman, strong expressions in the case of the man, and stronger reference image influence bleeding through, evident in the attempted text elements. Despite these strengths, the results suffer from lesser stylistic adherence, as the background purple colors and smoky effects are hallucinated (not part of any conditioning given by the user, neither in the reference image nor in the prompt). We found that generally, random noise generations suffer less from this loss, as evidenced in Fig. 5.2's colors and background effects. This style-layout tradeoff is something we further investigated in Section 5.1.

5.1 Possible Improvements

Post-user-study application, in an effort to improve the stability of the solution, we experimented with the following changes:

Noised reference image: Following the same logic of blending noise to inject variability, we tested the effect of interpolation and normalized interpolation over five generation seeds and five ratios in Fig. 5.3 and Fig. 5.4, respectively., but injected instead onto the original reference image.

Compared to Section 3.4.3.3, results seen in Fig. 5.3 and Fig. 5.4 (especially the latter), show a smoother transition between the reference and noisy image when using noise ratios between 0.5 and 0.75. Results are both creatively disparate from the reference but still feature elements from it (for example, the position of the head, the growling mouth, and the border shape). Future work should study the impact of feeding a pixel-space image into the diffusion process and measure the

Figure 5.1: Side-by-side comparison of results shown in user study. Generated using IPAdapter plus inverted and our masked inverted initial noise, respectively.

Figure 5.2: Side-by-side comparison of results generated using IPAdapter plus our masked inverted and random initial noise, respectively.

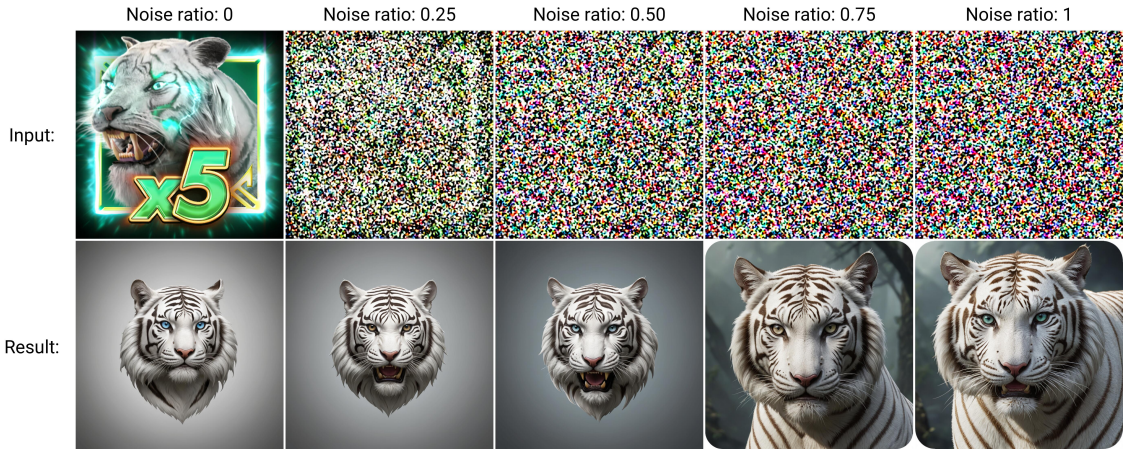


Figure 5.3: Inputs and results of interpolation between original reference and noisy image

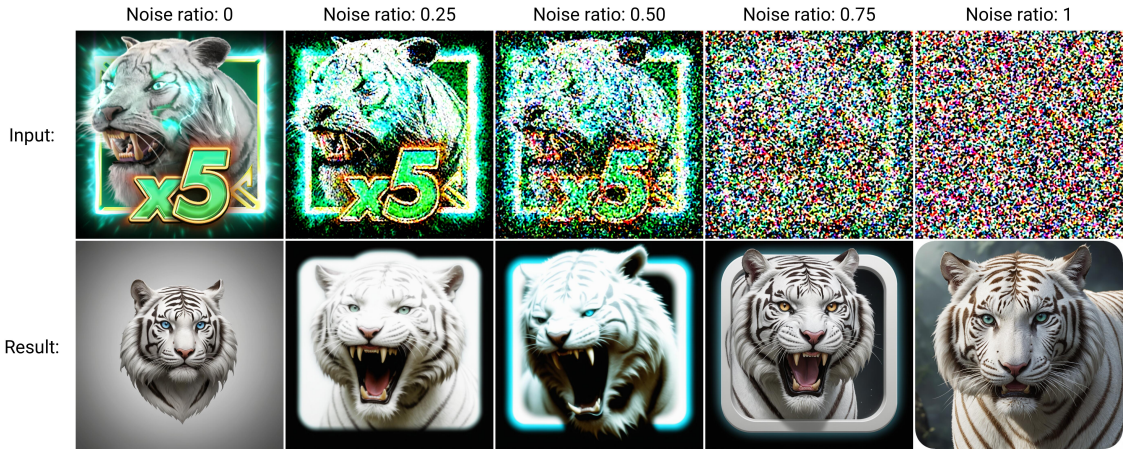


Figure 5.4: Inputs and results of normalized interpolation between original reference and noisy image

Figure 5.5: Results of normalized interpolation between original reference and noisy image, for five random seeds.

impact of noising said image. In Fig. 5.5, we solidify this claim by expanding the results of five different seeds.

5.2 Summary

This chapter discussed the findings from evaluating our game icon pack-generating solution. We found that the IPAdapter backbone consistently outperformed the Vanilla backbone in all four assessed mediums. Our novel Masked Inverted Noise source also significantly impacted the users' perception of the generated images. We also discussed the results of a post-experiment expert evaluation, which highlighted the strengths and weaknesses of the different configurations in a manner closer to the game development context. Finally, we presented possible improvements to our solution, such as using a noised reference image instead of a diffusion-inverted one, and discussed the results.

Chapter 6

Conclusion

This dissertation explored the potential of using diffusion models to generate casino digital video game icon packs. The motivation behind the research was the necessity to generate multiple visually coherent assets quickly, leveraging generative AI techniques to improve the efficiency of game asset development while maintaining high aesthetic standards. In Chapter 3, we proposed a solution combining existing state-of-the-art style transfer techniques with a novel initial noise manipulation method, which we evaluated in Chapter 4 through a user study and qualitative, expert-supported evaluation. We outlined the exhaustive experimentation that justified our solution's design choices. In Section 4.1, through a statistical impact analysis of the components of our solution, we determined that our novel method outperformed the traditional alternatives in all four assessed result quality metrics. In Section 4.2, we analyzed some handpicked successes and failures resulting from the research and experimentations, discussing domain-specific implications in close proximity with company experts. Finally, in this chapter, we present in Section 6.2 possible improvements to our solution stemming from missed opportunities during development, difficulties caused by the chosen user interface software, and inherent limitations of the generative diffusion process itself.

6.1 Contributions

Revisiting the original hypothesis in Section 1.3, we now summarize the contributions of our work.

Our proposed solution, consisting of a diffusion pipeline powered by state-of-the-art, frozen pre-trained SDXL weights, harmonizes a state-of-the-art style transfer module, IPAdapter, with a minimal initial noisy latent manipulation method. After conducting a user study on a population consisting of students and professional visual artists, we statistically proved that our novel method outperformed traditional alternatives. However, the degree of this improvement was not measurable due to limitations in the study data.

We conclude that our work successfully answered the hypothesis, with results proving its' feasibility and effectiveness in some cases. We consider that its limitations are not insurmountable and that further research is warranted.

6.2 Future Work

With generative diffusion models still in their infancy and the advent of their widespread adoption into the gaming industry, this work leads to many avenues for future research and improvement. We did not find much published work on diffusion models in 2d video game asset generation (See Appendix A.2), so we believe further research is warranted. Namely, including textual elements in icons and generating other game elements like user interfaces and screen layouts/compositions are currently feasible but, to our knowledge, have not yet been explored.

The most severe limitation of our solution is the lack of stability. The fact that fixing all but one of the conditions can yield either completely uncoherent icon sets or very coherent ones is a significant issue. This is a direct result of the unpredictable nature of noise. Even though our solution harbored predisposition characteristics of a reference image's noise to try to "limit" the resulting content layout, the introduced stochasticity was not the target of any manipulation or study. We believe work on this segment could prove fruitful (see Mao et. al.[51] for similar motivation).

As mentioned in Section 3.4.3.3, other noise-reference image blending processes should be studied, though we don't expect this to impact result flexibility or quality significantly. The exact effects of color and noisy patterns on conditioning within the initial noise vector should also be studied further.

On the other hand, improvements to the inversion process could improve the performance of content conditioning since the traditional error-prone inversion method still enabled successful results. Methods with lesser regeneration loss and better editability [57, 58] are expected to improve our solution.

We recognize that the hardware limitations of our work environment, which led us to limit our solution's area of influence to only the diffusion inference process, may have hindered the quality of the solution. Given more computational power and leveraging company data, a solution involving further training or fine-tuning of a model would significantly improve the quality of the results.

Lastly, in terms of evaluating the solution, we believe that a more systematic evaluation is required. Unlike most recent diffusion research works, ours performed no quantitative evaluation of the results' measurable visual quality nor efficacy in the realm of the problem. Developing a robust evaluation metric for the quality dimensions we studied in Section 4.1 would be a significant contribution. We note that the difficulty of this task lies in the heavily subjective and ungeneralizable definition of image quality, even more so in video game symbol images.

References

[1] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? In *Proceedings of the 36th International Conference on Software Engineering*, pages 1–11, 2014. 1

[2] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022. 1

[3] 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. vi, 1, 8, 9

[4] 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. 2, 17, 21

[5] Gillian Smith. An analog history of procedural content generation. In *FDG*. Boston, MA, 2015. 4

[6] Matthew J Guzdial and Mark O Riedl. Combinatorial creativity for procedural content generation via machine learning. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 4

[7] Daniel Karavolos, Antonios Liapis, and Georgios N Yannakakis. Using a surrogate model of gameplay for automated level design. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018. 4

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4, 8

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets, 2014. 4

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. vi, 5, 6, 7, 14

[11] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 5, 11

[12] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. vi, 6

[13] Nizar Bouhlel and Ali Dziri. Kullback–leibler divergence between multivariate generalized gaussian distributions. *IEEE Signal Processing Letters*, 26(7):1021–1025, 2019. 6

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. vi, 7

[15] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 7

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 7, 14

[17] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 7

[18] 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. 8, 17, 21, 45

[19] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, Jul 2021. 8

[20] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 8, 17, 21

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 9

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 9, 10

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. vi, 9, 10

[24] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22669–22679, 2023. 9

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 9

[26] Hezheng Lin, Xing Cheng, Xiangyu Wu, and Dong Shen. Cat: Cross attention in vision transformer. In *2022 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2022. 10

[27] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 10

[28] Jing Nathan Yan, Jiatao Gu, and Alexander M Rush. Diffusion models without attention. *arXiv preprint arXiv:2311.18257*, 2023. 10

[29] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 11

[30] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 11, 14

[31] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 11

[32] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015. 11

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. vi, 11, 12

[34] 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. 11

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. arxiv 2022. *arXiv preprint arXiv:2204.06125*, 2022. 11

[36] Jason Lee, Kyunghyun Cho, and Douwe Kiela. Countering language drift via visual grounding. *arXiv preprint arXiv:1909.04499*, 2019. 12

[37] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 12

[38] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 12

[39] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 12, 32

[40] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. vi, 13, 19, 32

[41] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. vi, 13, 14

[42] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, pages 679–698, 1986. 13

[43] Geonmo Gu, Byungsoo Ko, SeoungHyun Go, Sung-Hyun Lee, Jingeun Lee, and Minchul Shin. Towards light-weight and real-time line segment detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 726–734, 2022. 13

[44] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 13

[45] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 13

[46] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. vi, 15

[47] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023. 15

[48] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 15

[49] Barak Meiri, Dvir Samuel, Nir Darshan, Gal Chechik, Shai Avidan, and Rami Ben-Ari. Fixed-point inversion for text-to-image diffusion models. *arXiv preprint arXiv:2312.12540*, 2023. 15

[50] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5321–5329, 2023. 15, 22

[51] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Semantic-driven initial image construction for guided image synthesis in diffusion model. *arXiv preprint arXiv:2312.08872*, 2023. 15, 22, 57

[52] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *International journal of surgery*, 88:105906, 2021. 16

[53] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4695–4703, 2024. 29, 38

[54] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. *arXiv preprint arXiv:2312.09008*, 2023. 32

[55] Max Wertheimer. Investigations into the theory of form. *Gestalt Theory*, 39(1):79–89, 2017. 37

[56] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 48

[57] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. *arXiv preprint arXiv:2403.14602*, 2024. 57

[58] Barak Meiri, Dvir Samuel, Nir Darshan, Gal Chechik, Shai Avidan, and Rami Ben-Ari. Fixed-point inversion for text-to-image diffusion models. *arXiv preprint arXiv:2312.12540*, 2023. 57

Appendix A

Appendix: Pre-testing Experimentation

Before the main experiments, we exhaustively tested the model configurations to understand the impact of each setting and to decide which model checkpoint was more appropriate for testing our solution.

A.1 Steps and CFG Scale

We started by testing the Stable Diffusion XL, which was, at the time, the state-of-the-art StabilityAI diffusion model. We tested the model with different steps and CFG scales to understand the impact of these settings on the model's performance. The conditions that were not tested were fixed so as not to influence the testing (noise sampler was fixed to DPM++_SDE, noise sampler to Karras, the positive prompt was set to *"portrait photograph, beautiful girl"*, the negative prompt to an empty string, and the noise seed to 0). Besides the positive prompt, which was an arbitrarily chosen, simple example prompt, all other configurations follow the specification of the respective model authors. Despite the high quality evidenced in the 15-100 step range, for all CFG scales between 1.0-15.0, each diffusion step took an average of 2 seconds in the development environment available to us, which drove us to investigate the feasibility of faster models already available.

Within the timeframe of the beginning of the research, SDXL Turbo had just been released by StabilityAI as a model that traded off result quality for generation speed. We also tested different steps and CFG scales under the same scheduler and prompt conditions. Results for generations from 1 to 10 steps, using CGF scales of 0.8 to 3.0, showed that CFG scales under 1.0 and over 2.0 are unusable, with results being either unsaturated and full of hallucinations (foliage and flowers not in the prompt) or oversaturated and with leftover noise artifacts, respectively. By design, as a distilled diffusion model, it performs best in the 1-4 steps range, as expected. As such, the ComfyUI implementation of the model's scheduler only allows for a maximum of 10 steps. Despite the clear speed advantage, the model's best results were very inferior in quality to the SDXL model, leading us to explore community-developed models built on this architecture.

Dreamshaper XL Lightning [3] is a community-developed refinement of the previous SDXL Turbo architecture, trained on a mix of high-quality realistic and artistic image banks. We tested

the model under the same conditions as SDXL Turbo, obtaining results that show that the model was much more resistant to high CFG scales, only starting to oversaturate results at values over 5.0, an effect that even then is attenuated by increasing the diffusion steps above the ideal 4-6 range. The resulting quality is also highly superior to SDXL Turbo's and is on par with SDXL at a fraction of the steps.

A.2 Appendix: Systematic Literature Review Paper

Preprint under revision for publication in IEEE Transactions on Games Journal 2024.

IEEE TRANSACTIONS ON GAMES, VOL. 16, NO. 1, MARCH 2024 1

Image-based video game asset generation and evaluation using deep learning: a systematic review of methods and applications

Rafael Ribeiro*, Alexandre Valle de Carvalho*†, Nelson Bilber Rodrigues*†
* FEUP † INESC TEC

***Abstract*—Creating content for digital video game is an expensive segment of the development process, and many techniques have been explored to automate it. Much of the generated content is graphical, ranging from textures and sprites to typographical elements and user interfaces.**

Numerous techniques have been explored to automate the generation of these assets, with recent advancements incorporating artificial intelligence methodologies such as deep learning generative models. This study comprehensively surveys the literature from 2016 onwards, focusing on using machine learning to generate image-based assets for video game development, reviewing the deep learning approaches employed, and analyzing the specific challenges found. Namely, what deep learning approaches have been used, what problems within the domain have been tackled, and the metrics by which their results are evaluated.

The study demonstrates a knowledge gap in generative methods for some types of video game assets. Additionally, applicability and effectiveness of the most used evaluation metrics in the literature are studied. As future research prospects, with the increase in popularity of generative AI, the adoption of such techniques will be seen in automation processes.

***Index Terms*—GAN, Diffusion Models, Evaluation Metrics**

I. Introduction

THE rapid expansion of the digital video game (henceforward, video game) market, coupled with the relatively short-shelf life of individual games, has created an environment of relentless demand for innovation and new products.

Creating video game content is an intensive, expensive, and laborious task. It involves many experts, from visual artists to sound designers, all collaborating and iterating on a process to create and unify the many pieces that compose the envisioned product that is the final game. A considerable part of that process is graphical, ranging from textures and sprites to typographical elements and user interfaces.

The idea to automate segments of this process is not new. As early as the 1980s, techniques like Procedural Content Generation (PCG) have been used to leverage the generation of game levels and assets. Despite this, the majority of visual elements in games are still crafted by hand. Machine learning techniques have garnered significant attention in the past years to agile the process of asset generation, namely the use of Generative Adversarial Networks (GAN) [1] to generate game content with interesting results [2], and deep learning [3]. This systematic review, implemented on the PRISMA [4] framework, studies the problem of graphical asset generation for video games using machine learning methods, with emphasis in GANs, VAEs and GMMs. Beside the aforementioned techniques for asset generation this study incorporates works based in Diffusion Models (DM) and presents the promising results in image generation tasks. Furthermore, the contributions of this study are exposed in the following research questions:

1) What approaches have been used in the field of synthetic image generation?
2) What problems have been tackled in literature?
3) What evaluation methods have been used?

The review supports that GANs have dominated the state-of-the-art image generation in the last five years and that DMs' popularity has spiked in the past year.

On the topic of graphical game content generation problems, it identifies 3D, map or level, and image manipulation tasks as the most studied and game interface and logo generation as the least studied. It identifies the encompassing sub-problems within the image manipulation topic and organises them by input types.

Concerning the evaluation methods, it classifies the most found metrics in the literature as state-of-the-art or not and as objective, perceptive, or hybrid, that is, if they measure known statistical properties, qualitative features, or a mixture of both. The leading indicators found for objective metrics are Frechét Inception Distance (FID), Peak-Signal-to-Noise-Ratio (PSNR), and Inception Score (IS), and for perceptive metrics are Structured Similarity Indexing Method (SSIM), CLIP Score (CS), CLIP-R-Precision, and Neural Image Assessment (NIMA). Additionally, it expands upon qualitative means of evaluation, such as user studies and surveys.

The remainder of this article is as follows. Section II describes the steps to perform the systematic review based on PRISMA, and Section III details the execution of the chosen methodology. Section IV presents the review results, and Section V elaborates on the discussion about the achieved results. The article is concluded in Section VI.

II. Review Methodology

The methodology and guidelines to conduct the systematic review were the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [4]. The following steps were conducted: search in scientific digital databases, define inclusion and exclusion criteria, gather data, filter, screen, and study the eligible papers in depth.

A. Search Process

The search process to support the research questions was conducted through an exhaustive search and query process by consulting the following digital libraries of scientific literature, namely, ACM Digital Library, IEEE Xplore, Science Direct, Springer, CiNii, and arXiv. These were selected according to their relevance to the theme and the quality of publications.

A thesaurus of keywords is compiled and organized into Table I. Keywords are joined within each cell using the `OR` operator and between columns using the `AND` operator, resulting in the set of all possible, non-redundant search strings necessary to collect the state of the art in its entirety.

TABLE I
SEARCH KEYWORDS.

A	B	C
Image generation	Videogame Video game Video-game Videogames Video games Video-games	Generative Adversarial Network Generative Adversarial Networks GAN GANs
	Gaming	Diffusion
		Variational Autoencoder Variational Autoencoders VAE VAEs
		Gaussian Mixture GMM GMMs

Each search string is input into each digital library database, and search results are exported and treated. Any books, conference papers, journal and workshop articles are included. Columns A to C were chosen from broad topics such image generation narrowing down to specific techniques relevant to our study, namely, GANs, VAEs, and GMMs, having video games as the intermediate application area. Column B also narrows down to our focus application area, which is video games.

B. Inclusion & Exclusion Criteria

Inclusion and exclusion criteria are applied to the collection resulting from the previous search process to narrow down the works to screen. Besides the seminal paper about GANs appears in 2014 [5], the authors decide to search after 2016, when was possible to observe relevant number of papers applied to the subject of video game development. Table II lists the ordered applied criteria:

TABLE II
INCLUSION AND EXCLUSION CRITERIA.

Index	Criteria	Type
1	Search engine results after executing a search query	Inclusion
2	Not duplicated	Exclusion
3	Published in the English language	
4	Date of publication since 2016	
5	Full-text access	
6	Not summaries of workshops nor panels, tutorials, reviews, surveys, proceedings, or editorials	

C. Data Collection

The filtering and analysis of the collected works required extracting and processing the metadata and content-related information, as listed in Table III. The former is collected on retrieval, and the latter is extracted from the full texts in the last phase of the review.

TABLE III
FIELDS COLLECTED DURING DATA RETRIEVAL.

Category	Fields
Metadata	• Title • Authors • Publication year • Publication type • Bibtex reference
Content-related	• Application domain • Problem • Deep learning architecture • Image quality metrics

D. Data Evaluation

For each included study, an assessment composed of the questions outlined in Table IV is applied. For each question, the possible answers' meaning is Good, Average, or Poor, in this order.

TABLE IV
QUALITY ASSESSMENT QUESTIONS.

Index	Question	Answers
QA1	Did the work extend the state of the art on image generation techniques?	1) The work extends the state-of-the-art image generation techniques. 2) The work shows some new results, using existing techniques from the state of the art. 3) The work uses an existing technique and shows no new findings.
QA2	Did the work extend the state of the art on evaluating generated images?	1) The work introduces new evaluation metric(s). 2) The work applies commonly-used metrics, qualitative comparisons or user studies. 3) The work performs no explicit image evaluation.
QA3	Did the work consider the degree of control of style and content of the generated results?	1) The work provides detailed mechanisms to control both style and content (i.e. iterative guidance, natural language prompts). 2) Some considerations are given to controlling style and content (i.e. through inputs). 3) The work does not focus on providing control over the style and content.

III. Review Execution

This section describes the steps taken to apply the previously described methodology.

A. Querying

Query strings were generated from a list of keywords and aggregation rules (Table I). For each digital library, queries were executed using the resulting query strings. The resulting response page is scraped for the returned work's titles, abstracts, publication years, and publication types, which were then counted and saved in a datasheet. The pipeline of interaction with the databases' web interfaces, from inputting the query to running the search, interacting with pagination, and scraping the content, was implemented using Selenium WebDriver [1].

B. Filtering

The resulting datasheet was automatically filtered according to the inclusion and exclusion criteria described in Section II-B. This is done by checking the corresponding fields for each work and removing those that do not meet the criteria. Namely, duplicate titles, publication dates before the year 2016, citation types that include "workshop", "panel", "tutorial", "review", "survey", "proceedings", or "editorial", and title or abstracts not in English (checked using the `langdetect` [2]).

C. Screening

Titles and abstracts of works from the previous stage were individually analyzed to further trim down the pool of included works.

Reasons for exclusion in this phase included: application domain in the medical, fashion or industrial fields, traffic management, or city planning; de-noising/de-hazing, data prediction, image retrieval, steganography, optical character recognition (OCR); facial expression, emotion, speech, and music generation or identification.

D. Eligibility Assessment

Finally, the full texts of the remaining works were assessed for inclusion in the systematic review, resulting in a total of 99 included articles. Abstracts, contexts, conclusions, and experimental results sections were analyzed, in this order, to determine whether the work is relevant to answering the research questions. Otherwise, it is excluded, and the reason is annotated. To maximize the amount of information extracted from each work on the first read-through, eligibility assessment and answers to the research questions are simultaneously annotated on each work.

At the end of the process, since these were not in the search results of the scientific databases, citations for each work were collected in BibTeX format using a Google Scholar API [3]

[1]https://www.selenium.dev/
[2]https://pypi.org/project/langdetect/
[3]https://github.com/ckreibich/scholar.py

IV. Review Results

Fig. 1 shows the resulting PRISMA flowchart, with the number of papers retrieved and excluded at each stage and for what reason.

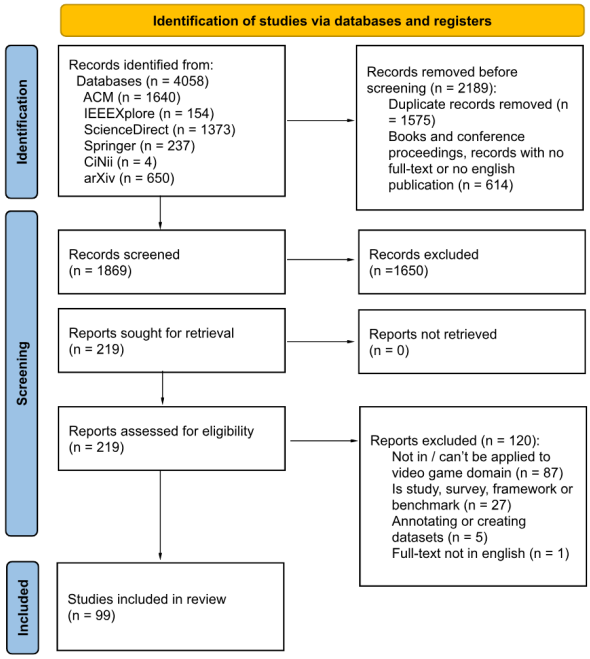


Fig. 1. PRISMA flow diagram [4].

A total of 4058 papers were first identified after querying all databases; 1869 remained after the automatic application of the inclusion and exclusion criteria, 1650 were excluded after the screening, and 120 were excluded after the eligibility assessment. The reasons for this last round of exclusion, besides the ones listed in Section III-C, are mainly false positives from the screening and pre-screening stage, as most are cases outside the video game domain or of undesired publication types, such as surveys and studies. The final number of included works is 99.

Fig. 2 illustrates the distribution of papers from each digital library at each review step. Despite representing just 16.1% (650 of 4058) of the initial retrieval, in the end, the major contributor to the final pool is arXiv, with 39.4% (39 of 99). The second major contributor is Springer, with 25.3% (25 of 99), followed by ACM, with 16.2% (16 of 99), ScienceDirect, with 15.2% (15 of 99), and IEEE, with 4.0% (4 of 99). CiNii is the only digital library that did not contribute to the final pool of included works, with 0.1% and 0.2% in the first two stages, respectively, and no remaining papers after screening.

On the quality of the findings, Fig. 3 shows the distribution of the answers to the quality assessment questionnaire.

Only 10.1% of the works present no new image generation results or techniques. 50.5% present novel techniques, and

[4]http://prisma-statement.org/prismastatement/flowdiagram.aspx

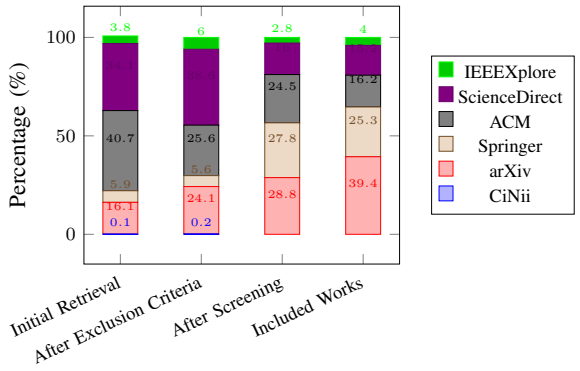


Fig. 2. Distribution of papers retrieved from digital libraries at each step of the systematic review process.

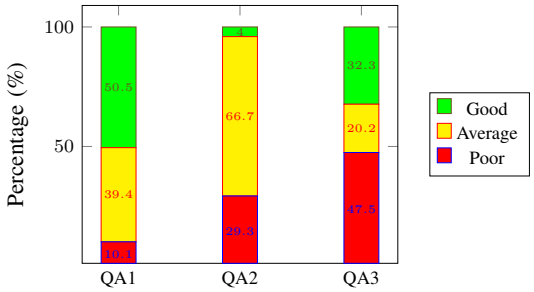


Fig. 3. Quality assessment answers distribution.

39.4% present new results using existing techniques. The latter mostly represent combinations of two or more existing approaches to a new domain or problem [6], [7].

29.3% performed no image evaluation of the results. In the majority of the cases, this is because the work itself doesn't target image quality but other aspects of video game quality like level novelty, although a small percentage of this fraction actually performed no evaluation at all. The majority of works (66.7%) performed adequate evaluations, be it through quantitative measurements, qualitative comparisons and user studies, or a combination of both. A small fraction (4%) developed new metrics for the specific problem at hand (e.g. [8] proposed "FISI, which compromises FID and SSIM to focus on both target features and retaining input content").

About half (47.5%) of the works do not focus on providing direct user control over the output's style and content, 20.2% provide some degree of control, albeit not directly through text prompts or style images, and 32.3% provide detailed mechanisms for controlling both style and content. Namely, style image inputs [9], [10], label maps/layouts [11], [12], editing instructions [13], [14], and natural language prompts [6], [7], [15], [16], [17], [18].

V. Discussion

Further analysis is devoted to the findings of the previous section by answering the research questions proposed in Section I and then by concluding with a discussion of the key findings of this study in Section V-B.

A. Answering Research Questions

1) RQ1: What approaches have been used in the field of synthetic image generation?: Of the 4333 originally retrieved documents containing mentions of image generation, only 99 were included in the final pool. These implemented a myriad of different approaches to the problem of video game content generation, from level-generating cellular automata [19] and reinforcement learning agents [20], [21] to Feedforward neural networks [22] and Generative Playing Networks [23]. However, the most prevalent solutions were those based on deep learning (99.94%).

Fig. 4 shows the chronological distribution of deep learning solutions in the included works. GANs are the most prevalent, representing 67.7% of the total works' approaches. However, in this past year, Diffusion Models (DM) have been gaining popularity. Representing just 21.2% of the full works share but also 50.0% of this past year's works, these networks have shown great promise in image generation, with quantitatively and qualitatively better results than previous architectures.

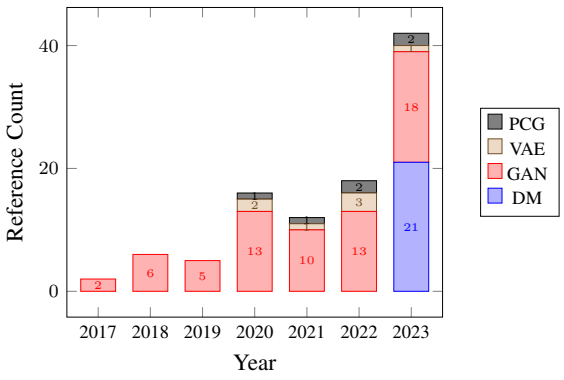


Fig. 4. Included papers of different image generation approaches for each year (2017-2023).

2) RQ2: What problems have been tackled in the literature?: Video game development pipelines contain many problems that can be solved using image generation. Included works were first categorized by their application domain, as can be seen in Fig. 5. Then, the focus is directed to the research scope, and a selected domain is analyzed to identify specific tasks.

An emphasis on Level/Map Generation is evident. The segment of these works focused on solving the task of 2D level generation in platformer [23], puzzle [88], and role-playing [16], [22], [89] games, from both game design and visual aesthetic perspectives. In contrast, another focused on the 3D generation of world terrains [24], [25], [93], [94], indoor scenes [7], [29] and urban settings [45].

Besides maps, 3D content generation is a main trend and it has been applied to model texturing [38], hair [41], asset [39], [40], [43], voxel shape [37] and human face/head [26], [27],

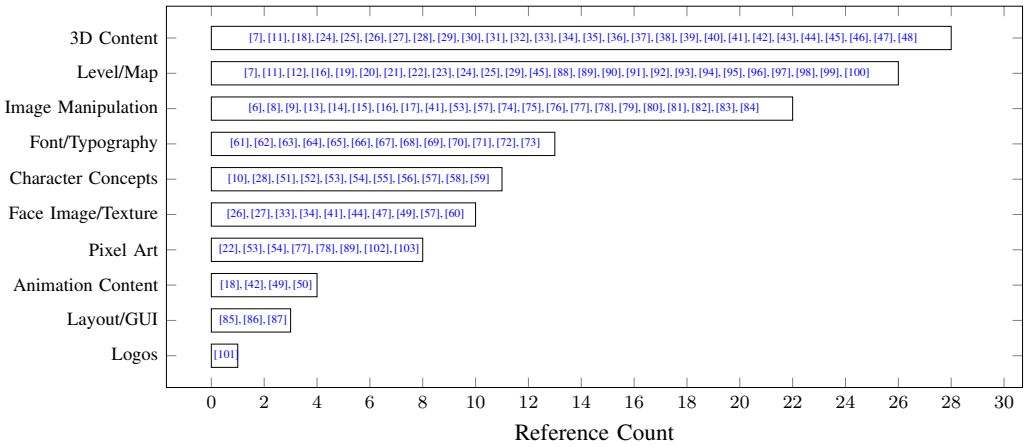


Fig. 5. Distribution of included works by application domain.

[33], [34], [44], [47] generation. Another main area of focus is Font/Typography Generation. Namely, the generation of artistic fonts from natural language descriptions [61], high-level attributes [71], handwriting [63], [66], [67], [70] and style transfer [68], [73]. Lastly, character concept generation from random noise [51], [56], [59], line drawings/sketches [52], [54] and blurred silhouettes [10]. Less explored generation problems include logos [101], pixel art characters [53], [54], [77], [89], sprite assets [22], [54], [89], map chunks [22], [89], input image pixelization [78], game effects [74] and Graphical User Interfaces (GUI) [85], [86], [87].

TABLE V
IMAGE MANIPULATION SUBPROBLEMS AND INPUTS.

Task	Inputs	References
Colorization	Sketch outline	[79]
Semantic image editing	Input image + text instruction	[13], [14]
Stylized image generation	Text prompt	[15]
	Style image + structure image	[74]
	Text + Sparse semantic map	[17]
	Content image + prompt + style image	[9]
Content image generation	Background image + foreground content text	[76]
Photograph stylization	Photograph	[8], [75], [80]
Improvements	Input text generation query	[6]

The image manipulation domain is further studied to support the style and content control aspect of image generation, previously referenced in QA3. Table V splits it into its subproblems and the inputs of each solution. Manipulation of existing images has been applied to the problem of colorization of hand-drawn sketches [79] as well as in photograph cartoonization/stylization [8], [75], [80] and image editing, be it using subject manipulation instructions (translating, rotating, enlarging, adding or removing) [13], [14], [76]. The latter are also split into stylized and content-based generations.

3) RQ3: What evaluation methods have been used?: Assessing the quality of generated images is a crucial aspect of image generation model development. Measuring the degree of certain characteristics of results is essential to determine performance and achieve progress. Various methods have been developed to objectively and subjectively measure these synthesized images' fidelity, realism, perceptual accuracy, and creativity. The literature review found a total of 48 different objective computable metrics, along with quantitative evaluation methods, and to answer this research question, these were categorized and separated by the research problem formulation.

There are multiple fields by which to characterize an evaluation metric. The metrics found in the literature are aggregated by their objectiveness (objective vs perceptive nature), i.e., if they are computed based on known properties and characteristics of images, measure image quality or features that align with human perception and judgment.

Objective Metrics

Objective metrics are quantifiable and rely on mathematical computations and direct comparisons between reference and distorted images. See Table VI.

The most common metrics referenced in the literature are Fréchet Inception Distance (FID), Peak-Signal-to-Noise-Ratio (PSNR), and Inception Score (IS). PSNR evaluates the quality of an image or video by measuring the difference between the original and compressed versions. IS is maximized when the entropy of the distribution of labels predicted by the Inceptionv3 [104] model for the generated images is minimized and when the predictions of the classification model are evenly distributed across all possible labels, indicating both "sharp" or "distinct" images and diverse outputs from the generative model. However, it evaluates only the distribution of generated images, unlike FID, which also considers real images from the training set. It is calculated by measuring the distance between

TABLE VI
OBJECTIVE EVALUATION METRICS AND REFERENCES.

Evaluation Metric	References
Fréchet Inception Distance (FID)	[8], [11], [13], [15], [17], [25], [26], [28], [29], [38], [39], [40], [41], [53], [62], [65], [66], [70], [71], [73], [75], [78], [81], [82], [85], [86], [92], [94], [101]
FID-Ori & FID-3D	[40]
Gan-Train & Gan-Test	[66]
Single Image FID (SIFID)	[30]
Inception Score (IS)	[8], [11], [13], [43], [58], [71], [73], [76], [94], [101]
Peak-Signal-to-Noise-Ratio (PSNR)	[12], [35], [44], [50], [52], [64], [79], [91], [94]
Mean Squared Error (MSE)	[44], [45], [53], [64], [91]
Root Mean Squared Error (RMSE)	[44], [49], [54], [68]
Pixel-Level Accuracy (pix-acc)	[54], [68], [71], [73]
Kernel Inception Distance (KID)	[34], [38], [78]
1-Nearest Neighbor Accuracy (1-NNA)	[86]
Conditional IS (CIS) & Conditional FID (CFID)	[62]
Chamfer Distance (CD)	[40]
Classification Error (CE)	[49]
Color Histogram (CH)	[79]
Entropy of Images (EN)	[52]
Image Euclidean Distance (IED)	[57]
Masked IS (m-IS)	[58]
Maximum Mean Discrepancy (MMD)	[37]
Mean Absolute Error (MAE)	[49]
Per-vertex reconstruction error	[42]
Zero Normalized Cross-Correlation (ZNCC)	[45]

feature vectors calculated for real and generated images.

Variations of these metrics have been implemented to better evaluate a solution on a case-by-case basis. For example, Masked IS (m-IS), a variation of IS, masks out the image backgrounds before evaluation. Kernel Inception Distance (KID), which uses the Maximum Mean Discrepancy (MMD) to measure distribution dissimilarity, and Conditional FID (CFID), which measures the spread of conditioned classes over actual classes, are variations of FID.

Perceptive Metrics

In contrast, perceptive metrics aim to mimic how humans perceive images and consider various factors such as visual saliency, color perception, texture, and overall aesthetics, and as such, often correlate better with human judgment (see Table VII).

The most referenced perceptive metrics in the performed literature review are Structured Similarity Indexing Method (SSIM), CLIP Score (CS) and CLIP-R-Precision, and Neural IMage Assessment (NIMA). SSIM is a traditional measure of the similarity between two images. It is based on the concept of structural information, incorporating critical perceptual phenomena such as luminance masking and contrast masking terms. Its modern variation, Multi-Scale Structural Similarity Indexing Measure (MS-SSIM), performs the evaluation at multiple scales through a multi-step downsampling process. CS and CLIP-R-Precision are used to evaluate the correlation between a generated caption for an image and the actual content of the image and are highly correlated with human judgment. A variation of these, Directional CLIP similarity, measures the consistency of the change between two images and two image captions. NIMA produces a distribution of ratings for any given image, assigning likelihoods to each of the possible scores, rather than classifying images as low/high quality or regressing to the mean score.

Qualitative Evaluations

Despite the variety of existing metrics, many visual qualities are hard to quantify objectively and much less compute. This fact is especially accentuated in the context of entertainment experiences such as video games, where the visual medium serves multiple purposes besides static displays of images, and so measuring their effectiveness becomes too complex a task. Many authors advocate the use of user studies [105], surveys, focus groups, or expert panels as alternative qualitative evaluations, that is, resorting to human judgement to assess the results' quality. The Table VIII list the studies and works on the survey that uses qualitative methods for evaluation.

B. Key Findings

The distribution of included papers' publication years, in Fig. 4, illustrates the growing trend of importance of the image generation research area, especially in the past two years, with a total of 25 and 41 included works published in the years 2022 and 2023, respectively. This is aligned with the increasing popularity of deep learning techniques in the past years and the growing interest within the video game industry in applying these techniques to generate content.

Works were liberally included in this study when it was believed they could be reasonably applied to the domain of video games. The decision to do so was made with the purpose of broadening the reach of this study to its maximum, and it is believed that this finding does not indicate a gap in the literature on game content generation.

The answers to the quality assessment questionnaire, in Fig. 3, specifically to QA3, reveal that natural language prompting was the most prevalent method for controlling style and content. This is associated with the recent explosion

TABLE VII
PERCEPTIVE EVALUATION METRICS AND REFERENCES.

Evaluation Metric	References
CLIP Score (CS)	[6], [11], [15], [22], [26], [34]
CLIP-R-Precision	[33], [36], [92]
Neural IMage Assessment (NIMA)	[6], [13], [76]
Learned Perceptual Image Patch Similarity (LPIPS)	[35], [71]
Aesthetic Evaluation	[55]
Directional CLIP similarity	[14]
F-score of Inception and SImilarity (FISI)	[8]
LAION-Aesthetic score (LAION-Aes)	[9]
Semantic Object Accuracy (SOA)	[92]
Structured Similarity Indexing Method (SSIM)	[8], [12], [35], [49], [50], [52], [53], [58], [64], [71], [73], [79], [91], [94]
Multi Scale Strctural Similarity Indexing Measure (MS-SSIM)	[35], [37]
Visual Information Fidelity (VIF)	[52], [64]
Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE)	[74]
Human Preference Score (HPS)	[9]
Masked SSIM (m-SSIM)	[58]
Natural Image Quality Evaluator (NIQE)	[74]
Perception based Image Quality Evaluator (PIQE)	[74]

TABLE VIII
WORKS WITH NO OR QUALITATIVE EVALUATIONS.

Evaluation Metric	References
Qualitative Comparison	[8], [15], [17], [26], [27], [31], [32], [34], [36], [47], [52], [55], [58], [59], [66], [68], [70], [72], [76], [78], [80], [87], [91], [93]
User Study	[8], [10], [16], [18], [34], [52], [60], [70], [77], [80], [86], [87], [94], [96]
None	[7], [19], [20], [21], [23], [24], [46], [51], [56], [61], [63], [67], [69], [88], [89], [90], [95], [98], [106]

in the popularity of Large Language Models (LLMs) like ChatGPT [107]. It's also predicted that this trend will continue, with the usability of generative tools evolving toward more natural interaction methods.

The distribution of included works' application domains, seen in Fig. 5, highlights less explored game elements' generation, like the GUI.

Works such as [16], [28] stood out on the topic of controlling style and content throughout the generation process. One of the main challenges identified was balancing the degree of control without sacrificing output quality.

Fig. 6. Visualization of 3D stylized avatars [28].

Dang et al [16] built a proof of concept that enabled historical editing of map tiles through region painting and text-to-image. The combination of the means of control implemented proved beneficial for real-world use. Zhang et al. [28] generated high-quality 3D avatars by combining diffusion priors for training data generation and GANs for generating 3D models. Within the problematic task of 3D model generation, their results achieved outstanding quality paired with the high degree of stylistic control that comes with textual guidance.

VI. CONCLUSION

The growing trend of using artificial intelligence to generate content for games inspired the survey, with a focus on studying the techniques used by the scientific community from 2016 to 2023.

PRISMA was the methodology chosen to conduct the systematic review. After filtering and screening, 99 papers were selected to answer the research questions related to the approaches to generating synthetic images, the associated problems and challenges, and how the evaluation metrics have been applied.

The results extracted by the literature review demonstrate that GANs and Diffusion Models are the most referenced techniques, and their popularity has increased over the years. Also, it was possible to verify that these techniques are popular in scientific literature to address problems such as 3D content, level/map generation, and image manipulation. The performed review show that the evaluation of generated content was based on qualitative and quantitative metrics. User studies, survey studies and interviews ensured the qualitative analysis. Nevertheless, the focus of this study was quantitative metrics, which were divided into three categories: objective, perceptive, and hybrid. Frechét Inception Distance (FID), Peak-Signal-to-Noise-Ratio (PSNR), and Inception Score (IS) are the leading indicators for objective metrics. For perceptive metrics, Structured Similarity Indexing Method (SSIM), CLIP Score (CS), CLIP-R-Precision, and Neural Image Assessment (NIMA) are the most referenced techniques. The hybrid metrics reference is a combination of Average Content Distance (ACD) and Cumulative Probability of Blur Detection (CPBD).

Some prospects for future research lie in segments of the game content development pipeline. Namely, layout generation and logo generation were the areas with the least included works identified. Other areas not referenced in this study

and whose generation were considered relevant include non-human 3D models (vehicles, animals, tools, etc.), special effects (SFX), particles, post-processors, shaders, and other unconventionally-rendered game content.

REFERENCES

[1] A. Tilson and C. M. Gelowitz, "Towards generating image assets through deep learning for game development," in 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 2019, pp. 1–4.

[2] S. Risi and J. Togelius, "Increasing generality in machine learning through procedural content generation," Nature Machine Intelligence, vol. 2, no. 8, pp. 428–436, 2020. [Online]. Available: https://doi.org/10.1038/s42256-020-0208-z

[3] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, "Deep learning for procedural content generation," Neural Computing and Applications, vol. 33, no. 1, pp. 19–37, 2021.

[4] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan et al., "The prisma 2020 statement: an updated guideline for reporting systematic reviews," International journal of surgery, vol. 88, p. 105906, 2021.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[6] Y. Lu, C. Guo, Y. Dou, X. Dai, and F.-Y. Wang, "Could chatgpt imagine: Content control for artistic painting generation via large language models," Journal of Intelligent & Robotic Systems, vol. 109, no. 2, pp. 1–15, 2023.

[7] Y. Lin, H. Wu, R. Wang, H. Lu, X. Lin, H. Xiong, and L. Wang, "Towards language-guided interactive 3d generation: Llms as layout interpreter with generative feedback," arXiv preprint arXiv:2305.15808, 2023.

[8] S.-T. Ho, M.-K. N. Huu, T.-D. Nguyen, N. Phan, V.-T. Nguyen, T. D. Ngo, D.-D. Le, and T. V. Nguyen, "Abstraction-perception preserving cartoon face synthesis," Multimedia Tools and Applications, pp. 1–18, 2023.

[9] J. Chen, Y. Pan, T. Yao, and T. Mei, "Controlstyle: Text-driven stylized image generation using diffusion priors," in Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 7540–7548.

[10] L. Guo and A. Bhojan, "Converting nebulous ideas to reality–a deep learning tool for conditional synthesis of character designs," in International Conference on Computer, Communication, and Signal Processing. Springer, 2022, pp. 75–89.

[11] C. Fang, X. Hu, K. Luo, and P. Tan, "Ctrl-room: Controllable text-to-3d room meshes generation with layout constraints," arXiv preprint arXiv:2310.03602, 2023.

[12] C. Liu, Z. Yang, F. Xu, and J.-H. Yong, "Image generation from bounding box-represented semantic labels," Computers & Graphics, vol. 81, pp. 32–40, 2019.

[13] R. Morita, Z. Zhang, M. M. Ho, and J. Zhou, "Interactive image manipulation with complex text instructions," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 1053–1062.

[14] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18 392–18 402.

[15] Z. Xue, G. Song, Q. Guo, B. Liu, Z. Zong, Y. Liu, and P. Luo, "Raphael: Text-to-image generation via large mixture of diffusion paths," arXiv preprint arXiv:2305.18295, 2023.

[16] H. Dang, F. Brudy, G. Fitzmaurice, and F. Anderson, "Worldsmith: Iterative and expressive prompting for world building with a generative ai," arXiv preprint arXiv:2308.13355, 2023.

[17] D. Pavllo, A. Lucchi, and T. Hofmann, "Controlling style and semantics in weakly-supervised image generation," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. Springer, 2020, pp. 482–499.

[18] Y. Huang, J. Wang, A. Zeng, H. Cao, X. Qi, Y. Shi, Z.-J. Zha, and L. Zhang, "Dreamwaltz: Make a scene with complex 3d animatable avatars," arXiv preprint arXiv:2305.12529, 2023.

[19] S. Earle, J. Snider, M. C. Fontaine, S. Nikolaidis, and J. Togelius, "Illuminating diverse neural cellular automata for level generation," in Proceedings of the Genetic and Evolutionary Computation Conference, 2022, pp. 68–76.

[20] Z. Jiang, S. Earle, M. Green, and J. Togelius, "Learning controllable 3d level generators," in Proceedings of the 17th International Conference on the Foundations of Digital Games, 2022, pp. 1–9.

[21] S. Mohaghegh, M. A. R. Dehnavi, G. Abdollahinejad, and M. Hashemi, "Pcgpt: Procedural content generation via transformers," arXiv preprint arXiv:2310.02405, 2023.

[22] T. Merino, R. Negri, D. Rajesh, M. Charity, and J. Togelius, "The five-dollar model: generating game maps and sprites from sentence embeddings," in Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 19, 2023, pp. 107–115.

[23] P. Bontrager and J. Togelius, "Learning to generate levels from nothing," in 2021 IEEE Conference on Games (CoG). IEEE, 2021, pp. 1–8.

[24] E. Panagiotou and E. Charou, "Procedural 3d terrain generation using generative adversarial networks," arXiv preprint arXiv:2010.06411, 2020.

[25] Y.-L. Huang and X.-F. Yuan, "Styleterrain: A novel disentangled generative model for controllable high-quality procedural terrain generation," Computers & Graphics, vol. 116, pp. 373–382, 2023.

[26] H. Zhang, Y. Feng, P. Kulits, Y. Wen, J. Thies, and M. J. Black, "Text-guided generation and editing of compositional 3d avatars," arXiv preprint arXiv:2309.07125, 2023.

[27] Z. Canfes, M. F. Atasoy, A. Dirik, and P. Yanardag, "Text and image guided 3d avatar generation and manipulation," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 4421–4431.

[28] C. Zhang, Y. Chen, Y. Fu, Z. Zhou, G. Yu, B. Wang, B. Fu, T. Chen, G. Lin, and C. Shen, "Styleavatar3d: Leveraging image-text diffusion models for high-fidelity 3d avatar generation," arXiv preprint arXiv:2305.19012, 2023.

[29] L. Gao, J.-M. Sun, K. Mo, Y.-K. Lai, L. J. Guibas, and J. Yang, "Scenehgn: Hierarchical graph networks for 3d indoor scene generation with fine-grained geometry," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[30] A. Abdellatif and A. H. Elsheikh, "Generating infinite-resolution texture using gans with patch-by-patch paradigm," arXiv preprint arXiv:2309.02340, 2023.

[31] T. Jakab, R. Li, S. Wu, C. Rupprecht, and A. Vedaldi, "Farm3d: Learning articulated 3d animals by distilling 2d diffusion," arXiv preprint arXiv:2304.10535, 2023.

[32] R. Chen, Y. Chen, N. Jiao, and K. Jia, "Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation," arXiv preprint arXiv:2303.13873, 2023.

[33] N. Kolotouros, T. Alldieck, A. Zanfir, E. G. Bazavan, M. Fieraru, and C. Sminchisescu, "Dreamhuman: Animatable 3d avatars from text," arXiv preprint arXiv:2306.09329, 2023.

[34] L. Zhang, Q. Qiu, H. Lin, Q. Zhang, C. Shi, W. Yang, Y. Shi, S. Yang, L. Xu, and J. Yu, "Dreamface: Progressive generation of animatable 3d faces under text guidance," arXiv preprint arXiv:2304.03117, 2023.

[35] J. Yang, Z. Cheng, Y. Duan, P. Ji, and H. Li, "Consistnet: Enforcing 3d consistency for multi-view images diffusion," arXiv preprint arXiv:2310.10343, 2023.

[36] A. W. Bergman, W. Yifan, and G. Wetzstein, "Articulated 3d head avatar generation using text-to-image diffusion models," arXiv preprint arXiv:2307.04859, 2023.

[37] K. Nakada and H. Kimata, "A study on voxel shape generation and reconstruction with vq-vae-2," in Proceedings of the 2023 7th International Conference on Graphics and Signal Processing, 2023, pp. 9–15.

[38] Y. Siddiqui, J. Thies, F. Ma, Q. Shan, M. Nießner, and A. Dai, "Texturify: Generating textures on 3d shape surfaces," in European Conference on Computer Vision. Springer, 2022, pp. 72–88.

[39] K. Dharma, C. T. Morrison, and B. Walls, "Texture generation using a graph generative adversarial network and differentiable rendering," in International Conference on Image and Vision Computing New Zealand. Springer, 2022, pp. 388–401.

[40] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler, "Get3d: A generative model of high quality 3d textured shapes learned from images," Advances In Neural Information Processing Systems, vol. 35, pp. 31 841–31 854, 2022.

[41] X. Guo, M. Kan, T. Chen, and S. Shan, "Gan with multivariate disentangling for controllable hair editing," in European Conference on Computer Vision. Springer, 2022, pp. 655–670.

[42] A. Akman, Y. Sahillioğlu, and T. M. Sezgin, "Deep generation of 3d articulated models and animations from 2d stick figures," Computers & Graphics, vol. 109, pp. 65–74, 2022.

[43] S. Kim, H.-g. Chi, and K. Ramani, "Object synthesis by learning part geometry with surface and volumetric representations," Computer-Aided Design, vol. 130, p. 102932, 2021.

[44] C. Murphy, S. Mudur, D. Holden, M.-A. Carbonneau, D. Ghafourzadeh, and A. Beauchamp, "Artist guided generation of video game production quality face textures," Computers & Graphics, vol. 98, pp. 268–279, 2021.

[45] B. H. Aithal, S. K. Das, and P. P. Subrahmanya, "Urban 3d structure reconstruction through a generative adversarial network model," Arabian Journal for Science and Engineering, vol. 45, no. 12, pp. 10 731–10 741, 2020.

[46] R. Spick, S. Demediuk, and J. Alfred Walker, "Naive mesh-to-mesh coloured model generation using 3d gans," in Proceedings of the Australasian computer science week multiconference, 2020, pp. 1–6.

[47] C. Murphy, S. Mudur, D. Holden, M.-A. Carbonneau, D. Ghafourzadeh, and A. Beauchamp, "Appearance controlled face texture generation for video game characters," in Proceedings of the 13th ACM SIGGRAPH Conference on Motion, Interaction and Games, 2020, pp. 1–11.

[48] H. Abu Alhaija, S. K. Mustikovela, A. Geiger, and C. Rother, "Geometric image synthesis," in Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI 14. Springer, 2019, pp. 85–100.

[49] J. Zhang, C. Liu, K. Xian, and Z. Cao, "Large motion anime head animation using a cascade pose transform network," Pattern Recognition, vol. 135, p. 109181, 2023.

[50] D. Loftsdottir and M. Guzdial, "Sketchbetween: Video-to-video synthesis for sprite animation via sketches," in Proceedings of the 17th International Conference on the Foundations of Digital Games, 2022, pp. 1–7.

[51] K. Scoon and K. Samara, "Synthesizing pokémon trading cards using nvidia stylegan2 from home," in Proceedings of SAI Intelligent Systems Conference. Springer, 2022, pp. 646–654.

[52] K. Fan, S. Liu, and W. Lu, "Semipainter: Learning to draw semi-realistic paintings from the manga line drawings and flat shadow," in Computer Graphics International Conference. Springer, 2022, pp. 305–317.

[53] A. Saravanan and M. Guzdial, "Pixel vq-vaes for improved pixel art representation," arXiv preprint arXiv:2203.12130, 2022.

[54] Y. R. Serpa and M. A. F. Rodrigues, "Human and machine collaboration for painting game assets with deep learning," Entertainment Computing, vol. 43, p. 100497, 2022.

[55] M. Li, J. Lv, C. Tang, J. Wang, Z. Lai, and Y. Huang, "Combination of certainty and uncertainty: Using fusiongan to create abstract paintings," Neural Networks, vol. 144, pp. 443–454, 2021.

[56] D. P. Jaiswal, S. Kumar, and Y. Badr, "Towards an artificial intelligence aided design approach: application to anime faces with generative adversarial networks," Procedia Computer Science, vol. 168, pp. 57–64, 2020.

[57] E. Ververas and S. Zafeiriou, "Slidergan: Synthesizing expressive face images by sliding 3d blendshape parameters," International Journal of Computer Vision, vol. 128, no. 10-11, pp. 2629–2650, 2020.

[58] S. R. Kurupathi, V. Dumpala, and D. Stricker, "Multi-stage conditional gan architectures for person-image generation," in International Conference on Deep Learning Theory and Applications. Springer, 2020, pp. 24–48.

[59] M. T. Mann and H. J. Hamilton, "Automatic generation of video game character images using augmented structure-and-style networks," in Advances in Artificial Intelligence: 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28–31, 2019, Proceedings 32. Springer, 2019, pp. 387–393.

[60] H. Dang, L. Mecke, and D. Buschek, "Ganslider: How users control generative models for images using multiple sliders with and without feedforward information," in Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–15.

[61] J.-Y. He, Z.-Q. Cheng, C. Li, J. Sun, W. Xiang, X. Lin, X. Kang, Z. Jin, Y. Hu, B. Luo et al., "Wordart designer: User-driven artistic typography synthesis using large language models," arXiv preprint arXiv:2310.18332, 2023.

[62] R. Kalingeri, V. Kushwaha, R. Kala, and G. Nandi, "Synthesis of human-inspired intelligent fonts using conditional-dcgan," in Computer Vision and Machine Intelligence: Proceedings of CVMI 2022. Springer, 2023, pp. 691–707.

[63] H. Shimomura and H. Miwa, "Automatic generation of handwritten style characters including untrained characters," in International Conference on Intelligent Networking and Collaborative Systems. Springer, 2023, pp. 14–23.

[64] Y. Miao, H. Jia, and K. Tang, "Artistic font generation network combining font style and glyph structure discriminators," Multimedia Tools and Applications, pp. 1–21, 2023.

[65] Y. Tuo, W. Xiang, J.-Y. He, Y. Geng, and X. Xie, "Anytext: Multilingual visual text generation and editing," arXiv preprint arXiv:2311.03054, 2023.

[66] J. Zdenek and H. Nakayama, "Jokergan: Memory-efficient model for handwritten text generation with text line awareness," in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 5655–5663.

[67] D. Chang, W. Xie, and Z. Yu, "Handwriting image generation by adversarial training," in 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). IEEE, 2021, pp. 658–661.

[68] C. Wang, Y. Zhu, Z. Shen, D. Wang, G. Wu, and Y. Yao, "Font transfer based on parallel auto-encoder for glyph perturbation via strokes moving," in International Conference on Algorithms and Architectures for Parallel Processing. Springer, 2021, pp. 586–602.

[69] M. Liao, B. Song, S. Long, M. He, C. Yao, and X. Bai, "Synthtext3d: synthesizing scene text images from 3d virtual worlds," Science China Information Sciences, vol. 63, pp. 1–14, 2020.

[70] L. Kang, P. Riba, Y. Wang, M. Rusinol, A. Fornés, and M. Villegas, "Ganwriting: content-conditioned generation of styled handwritten word images," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16. Springer, 2020, pp. 273–289.

[71] Y. Wang, Y. Gao, and Z. Lian, "Attribute2font: Creating fonts you want from attributes," ACM Transactions on Graphics (TOG), vol. 39, no. 4, pp. 69–1, 2020.

[72] H. Hayashi, K. Abe, and S. Uchida, "Glyphgan: Style-consistent font generation based on generative adversarial networks," Knowledge-Based Systems, vol. 186, p. 104927, 2019.

[73] Y. Gao, Y. Guo, Z. Lian, Y. Tang, and J. Xiao, "Artistic glyph image synthesis via one-stage few-shot learning," ACM Transactions on Graphics (TOG), vol. 38, no. 6, pp. 1–12, 2019.

[74] J. Kim, K. Jin, S. Jang, S. Kang, and Y. Kim, "Game effect sprite generation with minimal data via conditional gan," Expert Systems with Applications, vol. 211, p. 118491, 2023.

[75] F. Zhang, H. Zhao, Y. Li, Y. Wu, and X. Sun, "Cba-gan: Cartoonization style transformation based on the convolutional attention module," Computers and Electrical Engineering, vol. 106, p. 108575, 2023.

[76] R. Morita, Z. Zhang, and J. Zhou, "Batinet: Background-aware text to image synthesis and manipulation network," in 2023 IEEE International Conference on Image Processing (ICIP). IEEE, 2023, pp. 765–769.

[77] P. Lei, S. Xu, and S. Zhang, "An art-oriented pixelation method for cartoon images," The Visual Computer, pp. 1–13, 2023.

[78] Z. Wu, L. Chai, N. Zhao, B. Deng, Y. Liu, Q. Wen, J. Wang, and S. He, "Make your own sprites: Aliasing-aware and cell-controllable pixelization," ACM Transactions on Graphics (TOG), vol. 41, no. 6, pp. 1–16, 2022.

[79] S. Jiang, C. Li, and C. Wang, "Copaint: Guiding sketch painting with consistent color and coherent generative adversarial networks," in Advances in Computer Graphics: 38th Computer Graphics International Conference, CGI 2021, Virtual Event, September 6–10, 2021, Proceedings 38. Springer, 2021, pp. 229–241.

[80] Y. Nakashima and Y. Bannai, "A comparison of cartoon portrait generators based on generative adversarial networks," in Human Interface and the Management of Information. Interacting with Information: Thematic Area, HIMI 2020, Held as Part of the 22nd International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part II 22. Springer, 2020, pp. 231–244.

[81] J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. Tenenbaum, and B. Freeman, "Visual object networks: Image generation with disentangled 3d representations," Advances in neural information processing systems, vol. 31, 2018.

[82] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," arXiv preprint arXiv:1807.09384, 2018.

[83] A. Feygina, D. I. Ignatov, and I. Makarov, "Realistic post-processing of rendered 3d scenes," in ACM SIGGRAPH 2018 Posters, 2018, pp. 1–2.

[84] E. Zamyatin and A. Filchenkov, "Learning to generate chairs with generative adversarial nets," Procedia Computer Science, vol. 136, pp. 200–209, 2018.

[85] C. Acornley, "Using generative adversarial networks to create graphical user interfaces for video games," in Proceedings of the 2021

International Conference on Multimodal Interaction, 2021, pp. 802–806.

[86] T. Zhao, C. Chen, Y. Liu, and X. Zhu, "Guigan: Learning to generate gui designs using generative adversarial networks," in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). IEEE, 2021, pp. 748–760.

[87] M. Ueno and S. Satoh, "Continuous and gradual style changes of graphic designs with generative model," in 26th International Conference on Intelligent User Interfaces, 2021, pp. 280–289.

[88] F. Abraham and M. Stephenson, "Utilizing generative adversarial networks for stable structure generation in angry birds," in Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 19, 2023, pp. 2–12.

[89] A. Shams, D. Summers-Stay, V. Metelsky, A. Tripathi, and K. Malhotra, "Towards a holodeck-style simulation game," arXiv preprint arXiv:2308.13548, 2023.

[90] G. Todd, S. Earle, M. U. Nasir, M. C. Green, and J. Togelius, "Level generation through large language models," in Proceedings of the 18th International Conference on the Foundations of Digital Games, 2023, pp. 1–8.

[91] F. Abbas and M. C. Babahenini, "Forest fog rendering using generative adversarial networks," The Visual Computer, vol. 39, no. 3, pp. 943–952, 2023.

[92] Z. Chen, Z. Mao, S. Fang, and B. Hu, "Background layout generation and object knowledge transfer for text-to-image generation," in Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 4327–4335.

[93] J. Zhang, C. Li, P. Zhou, C. Wang, G. He, and H. Qin, "Authoring multi-style terrain with global-to-local control," Graphical Models, vol. 119, p. 101122, 2022.

[94] G. Siracusa, D. Seychell, and M. Bugeja, "An approach towards architecture-independent output for generative networks: Texturing aerial town maps for roleplaying games," in The International Conference on Intelligent Systems & Networks. Springer, 2021, pp. 193–219.

[95] S. Snodgrass and A. Sarkar, "Multi-domain level generation and blending with sketches via example-driven bsp and variational autoencoders," in Proceedings of the 15th international conference on the foundations of digital games, 2020, pp. 1–11.

[96] J. Schrum, J. Gutierrez, V. Volz, J. Liu, S. Lucas, and S. Risi, "Interactive evolution and exploration within latent level-design space of generative adversarial networks," in Proceedings of the 2020 Genetic and Evolutionary Computation Conference, 2020, pp. 148–156.

[97] A. Sarkar, Z. Yang, and S. Cooper, "Conditional level generation and game blending," arXiv preprint arXiv:2010.07735, 2020.

[98] R. R. Torrado, A. Khalifa, M. C. Green, N. Justesen, S. Risi, and J. Togelius, "Bootstrapping conditional gans for video game level generation," in 2020 IEEE Conference on Games (CoG). IEEE, 2020, pp. 41–48.

[99] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi, "Evolving mario levels in the latent space of a deep convolutional generative adversarial network," in Proceedings of the genetic and evolutionary computation conference, 2018, pp. 221–228.

[100] É. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, and B. Martinez, "Interactive example-based terrain authoring with conditional generative adversarial networks." ACM Trans. Graph., vol. 36, no. 6, pp. 228–1, 2017.

[101] L. Dong, Y. Zhou, and J. Jiang, "Generative synthesis of logos across dct domain," Neurocomputing, vol. 467, pp. 163–172, 2022.

[102] D. Lee and H.-j. Choi, "Text to game characterization: a starting point for generative adversarial video composition," in 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2018, pp. 717–720.

[103] L. Horsley and D. Perez-Liebana, "Building an automatic sprite generator with deep convolutional generative adversarial networks," in 2017 IEEE Conference on Computational Intelligence and Games (CIG). IEEE, 2017, pp. 134–141.

[104] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[105] R. Kosara, C. G. Healey, V. Interrante, D. H. Laidlaw, and C. Ware, "Thoughts on user studies: Why, how, and when," IEEE Computer Graphics and Applications, vol. 23, no. 4, pp. 20–25, 2003.

[106] Y. Sun, Z. Li, K. Fang, C. H. Lee, and A. Asadipour, "Language as reality: a co-creative storytelling game experience in 1001 nights using generative ai," in Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 19, 2023, pp. 425–434.

[107] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," Internet of Things and Cyber-Physical Systems, 2023.

A.3 Appendix: User Study Questionnaire

Casino Game Icon Generator

This questionnaire is part of the Master's Dissertation in Informatics and Computing Engineering at the Faculty of Engineering of the University of Porto (FEUP).

This study aims to evaluate the quality of an icon generation solution for casino video games based on a stylistic reference and a text command.

At the beginning of the survey, you will be asked for some personal information (age and level of proficiency in the design area). **No information that could identify you will be requested or collected, and the data collected will be treated confidentially, anonymously, and used only for the purposes of this study by the person responsible.**

Participation in this study is **strictly voluntary**, so you may interrupt your participation or withdraw your consent for treatment of your data at any time during or after completing the form.

I thank you in advance for your collaboration as it is essential for the development of this study. For any suggestions, questions or clarifications, please contact the person responsible for the study:

Rafael Ribeiro, up201806330@fe.up.pt.

By proceeding, you are agreeing to all the terms mentioned above.

* Indicates required question

Pre-Test Questionnaire

How old are you? *

- ◯ 17 or less
- ◯ Between 18 and 25
- ◯ Between 26 and 35
- ◯ Between 36 and 55
- ◯ 56 or more

How do you rank your proficiency in the area of design *

- ◯ No level of proficiency – no experience in the area
- ◯ Low level of proficiency – little experience in the area
- ◯ Average level of proficiency – some experience in the area
- ◯ Moderately high level of proficiency - good experience in the area
- ◯ High level of proficiency - extensive experience in the area

Introductory Concepts

Naturalness and realism: the quality of representing something in an accurate and faithful way to real life. In this context, we use this term **separately from stylistic realism**. Deformed or anatomically incorrect hands and objects that disobey the laws of physics are examples of a low degree of realism.

Prompt alignment: the degree to which a generated image respects the prompt, or command, used to generate it.

Negative space: When it comes to small game icons, especially those that live in complex and crowded compositions like casino games, there must be a balance between attractive and functional. The concept of negative space, also known as "white" space, refers to the empty or unmarked areas in a drawing or composition that serve to balance and enhance the visual impact of the positive elements within it.

Considering these concepts, we will now ask you to evaluate a total of six sets of images generated using our prototype.

Opinion Questions

Look at the following icon packs. An icon pack is made up of 4 individually generated images, organized in a grid.
Indicate your level of agreement with the statements on a scale from **1 (strongly disagree) to 5 (strongly agree)**.

If you feel the need to, you may zoom in and out of the webpage (**CTRL +/-**, **CTRL MouseWheelUp/Down**), or open the images in full size in a new tab (**RightClick, Open image in new tab**)

Opinion Questions

The images for the following icon pack were generated with the prompt:

"[man || woman || bird || pear] video game icon", where each image is one of the terms within brackets:
(1) man video game icon

(2) woman video game icon

(3) bird video game icon

(4) pear video game icon

Generated Icon Pack 1 *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Opinion Questions

Indicate your level of agreement with the statements on a scale from **1 (strongly disagree) to 5 (strongly agree)**.

The images for the following **5 icon packs** were generated with the same previous prompts (in short: **"[man || woman || bird || pear] video game icon"**), **but, additionally, the following image was given as reference :**

Generated Icon Pack 2 *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Generated Icon Pack 3 *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Generated Icon Pack 4 *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Generated Icon Pack 5 *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Generated Icon Pack 6 *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The generated images appear natural and realistic	○	○	○	○	○
The generated images respect the text prompt	○	○	○	○	○
The generated images effectively utilize negative space	○	○	○	○	○
The generated images are stylistically coherent with each other	○	○	○	○	○

Consider now both the text prompt and the reference image used to generate the previous six icon packs:

"[man || woman || bird || pear] video game icon"

If you had to choose just one pack, which one would you put in a game, considering the aesthetic features of the reference image above? *

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5
- ◯ 6

Get link

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms