Published: 19 July 2024

RESEARCH-ARTICLE

Creating LEGO Figurines from Single Images

JIAHAO GE, Chinese University of Hong Kong, Hong Kong, Hong Kong

MINGJUN ZHOU, Chinese University of Hong Kong, Hong Kong, Hong Kong

WENRUI BAO, Chinese University of Hong Kong, Hong Kong, Hong Kong

HAO XU

CHI WING FU, Chinese University of Hong Kong, Hong Kong, Hong Kong

Creating LEGO® Figurines from Single Images

JIAHAO GE, MINGJUN ZHOU, and WENRUI BAO, The Chinese University of Hong Kong, Hong Kong, China
HAO XU, Qianzhi Technology Inc., China
CHI-WING FU, The Chinese University of Hong Kong, Hong Kong, China

Fig. 1. Physical assemblies of 3D LEGO® figurine models, designed by our computational pipeline (the input portrait photos for the left three figurine models are shown in Figure 2). Note the rich styles and also the variety of garment patterns, logos, and details on the generated figurine models. From left to right, the figurines depict *Ludvig Von Beethoven*, *a woman wearing a star-pattern sweater*, the football star *Cristiano Ronaldo*, *Paul Atreides* played by *Timothée Chalamet* in the movie *Dune*, *Barbie* played by *Margot Robbie* in the movie *Barbie*, and *Goku* in the manga *Dragon Ball*.

This paper presents a computational pipeline for creating personalized, physical LEGO®[1] figurines from user-input portrait photos. The generated figurine is an assembly of coherently-connected LEGO® bricks detailed with uv-printed decals, capturing prominent features such as hairstyle, clothing style, and garment color, and also intricate details such as logos, text, and patterns. This task is non-trivial, due to the substantial domain gap between unconstrained user photos and the stylistically-consistent LEGO® figurine models. To ensure assemble-ability by LEGO® bricks while capturing prominent features and intricate details, we design a three-stage pipeline: (i) we formulate a CLIP-guided retrieval approach to connect the domains of user photos and LEGO® figurines, then output physically-assemble-able LEGO® figurines with decals excluded; (ii) we then synthesize decals on the figurines via a symmetric U-Nets architecture conditioned on appearance features extracted from user photos; and (iii) we next reproject and uv-print the decals on associated LEGO® bricks for physical model production. We evaluate the effectiveness of our method against eight hundred expert-designed figurines, using a comprehensive set of metrics, which include a novel GPT-4V-based evaluation metric, demonstrating superior performance of our method in visual quality and resemblance to input photos. Also, we show our method's robustness by generating LEGO® figurines from diverse inputs and physically fabricating and assembling several of them.

CCS Concepts: • **Applied computing → Computer-aided manufacturing**; **Consumer products**.

Additional Key Words and Phrases: LEGO®, computational design, fabrication, assembly, appearance adaptation, image synthesis

ACM Reference Format:
Jiahao Ge, Mingjun Zhou, Wenrui Bao, Hao Xu, and Chi-Wing Fu. 2024. Creating LEGO® Figurines from Single Images. *ACM Trans. Graph.* 43, 4, Article 153 (July 2024), 16 pages. https://doi.org/10.1145/3658167

1 INTRODUCTION

For decades, researchers in computer graphics have dedicated to enabling the creation of customized or personalized objects. Significant advancements have been made, as seen in the results produced

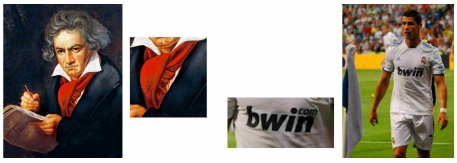

Fig. 2. The input portraits for creating the first and the third figurines shown in Figure 1 (from left to right). Prominent garment patterns in the portraits are blown-up, demonstrating the strong capability of our method to reconstruct these details; see Figure 1. The left portrait is *Beethoven with the manuscript of the Missa solemnis* by Joseph Karl Stieler, and the right portrait is *Cristiano Ronaldo* ©Jan S0L0.

[1]LEGO® is a trademark of the LEGO® Group, which does not sponsor, authorize or endorse this work. All information in this paper is collected and interpreted by its authors and does not represent the opinion of the LEGO® Group.

Authors' Contact Information: Jiahao Ge, 1155209932@link.cuhk.edu.hk; Mingjun Zhou, mingjunzhou@link.cuhk.edu.hk; Wenrui Bao, 1155157220@link.cuhk.edu.hk, The Chinese University of Hong Kong, Hong Kong, China; Hao Xu, hao.xu@maic.fun, Qianzhi Technology Inc., China; Chi-Wing Fu, cwfu@cse.cuhk.edu.hk, The Chinese University of Hong Kong, Hong Kong, China.

ACM 1557-7368/2024/7-ART153
https://doi.org/10.1145/3658167

by many innovative computational algorithms, *e.g.*, wired structures [Yang et al. 2021], compliant structures [Zhang et al. 2021], on layered LEGO® sketches [Zhou et al. 2023], *etc.* Yet, more efforts are typically required to reach out to wider audience beyond a niche group of technology-enthusiasts [Baudisch and Mueller 2017].

Personal fabrication has received much attention, motivated earlier by the immense popularity of 3D printing. Particularly, it aims not only to create unique objects physically but also to make objects that are personal, or customizable towards individuals. To realize these fundamental concerns, we need computational methods that are robust and simple-to-use, reducing or bypassing the need of professional inputs in the pipeline. Second, the production process should yield personalized objects of quality comparable to products that are massively produced, while being also cost-effective.

In this work, we design a new computational pipeline to aid the creation of personal fabrications, with a specific focus on creating personalized LEGO® figurine models, which are attractive as gifts or desk ornaments. Importantly, we aim to create a 3D object, not as a single piece, but as an assembly of standard LEGO® bricks. In this way, we can achieve high-quality surface finish, while avoiding cost-intensive material reshaping like molding. More specifically, our computational pipeline is robust, capable of designing LEGO® figurine models from user-input portraits, without specific requirements on the photo styles and shooting conditions. Also, the generated figurines are made up of coherently-connected LEGO® bricks adorned with cartoon-style decals, stylistically capturing prominent features, such as hairstyle, clothing style, and garment color, as well as intricate details, such as logos, text, and patterns, given in the user portrait; see Figures 1 and 2.

This task is non-trivial, primarily due to the substantial domain gap between the portrait photos and the LEGO® figurine models. User-taken photos typically have varying shooting conditions, while exhibiting occlusions and clothing distortions/wrinkles, whereas LEGO® figurines typically have a consistent cartoon style in the body shapes and fairly-upright decals on the model surface. Having said that, our method thus needs to carefully reproduce these varied photo elements, while conforming to the style of LEGO® figurines. Also, these elements need to be abstracted, *e.g.*, major structural elements such as ties and belts, while intricate details like logos and text should be preserved. Further, this is far beyond an image translation task. We need to ensure the results are physically realizable as LEGO® bricks assemblies. See again our results in Figure 1.

To address the challenges of physical realizability and domain gap between user portraits and LEGO® figurines, we design the three-stage computational pipeline illustrated in Figure 3. First, we devise a CLIP-guided approach to connect the two domains and formulate a retrieval-based method to ensure creating a physically-buildable LEGO® figurine, which matches the input portrait. Second, we generate decals on the figurine by designing a symmetric U-Nets architecture, equipped with an appearance adaptation strategy and conditioned by the appearance features extracted from the portrait photo. Last, we prepare essential data to aid physical production, including uv-printing decals on LEGO® bricks and producing assembly instructions, such that we can assemble the fabricated bricks into a physical LEGO® figurine. Further, to enhance our method's robustness, we leverage several pre-trained large models in our pipeline design, including CLIP [Radford et al. 2021] for extracting image semantics during the model retrieval, DINO [Oquab et al. 2023] for capturing intricate details while improving resilience to varying photo shooting conditions, and Stable Diffusion [Rombach et al. 2022] for high-quality decal generation.

To evaluate our method, we recruited expert LEGO® designers to help create LEGO® figurines and performed a quantitative analysis against six alternative approaches. Also, we introduce a comprehensive set of evaluation metrics, including one that creatively utilizes the image understanding abilities of GPT-4V (GPT-4 with vision) [OpenAI 2023], to compare figurines generated by different methods against the expert designs. The quantitative comparison results show that our method not only outperforms the alternative approaches but also rivals the quality of the expert designs in terms of visual fidelity. Further, we interviewed five human designers and collected their subjective feedback and ratings. Besides, we test our method's robustness on diverse inputs with, *e.g.* , occlusions, garment distortions, and varying styles, *e.g.* ,anime characters and oil paintings. Last, we fabricate and assemble six figurines.

To conclude, the overall contribution of our work is on the development of a computational pipeline, capable of designing personalized objects, namely the LEGO® figurines. Our work demonstrates the technical feasibility of the pipeline, which allows one to create personalized figurines from casual portrait photos, while keeping a cost-effective production process. In terms of technical contributions, we present a three-stage pipeline solution that achieves the creation of a LEGO® figurine that not only resembles the input portrait but is also physically realizable. Our pipeline includes a CLIP-guided module that leverages the CLIP space for model retrieval, a symmetric U-Nets structure with appearance adaptation for high-quality decal generation, and also a physical production process to aid the LEGO® model creation. Last, we also introduce a new GPT-4V-based metric for quantitative evaluation of results, offering a fresh perspective on developing large-scale quantitative comparisons similar to human perception.

2 RELATED WORK

In this section, we discuss assorted areas of related works.

Personal fabrication. The concept of "personal fabrication" is first comprehensively studied in Baudisch et al. [2017], which aims to enable average users to create personalized physical objects. To achieve this goal, researchers in computer graphics have been dedicated to developing advanced design algorithms and interaction techniques to meet the needs of creating various specific objects. Mitani and Suzuki [2004] transform a given 3D mesh to a papercraft by strip-based approximate unfolding. Duncan et al. [2017] take a pair of distinct 2D shapes and dissert them into compatible component pieces that can be assembled to both input shapes. Song et al. [2017] turn a wind-up toy design with part segments and motion annotations into a functional toy mechanism that can perform the desired motions. More recently, Korosteleva and Sorkine-Hornung [2023] take high-level user operations to aid the design of garment sewing patterns. In the field of architecture, assorted works [Becker et al. 2023; Panetta et al. 2019; Soriano et al. 2019] have been developed for producing deployable grid-shell structures that closely follow the

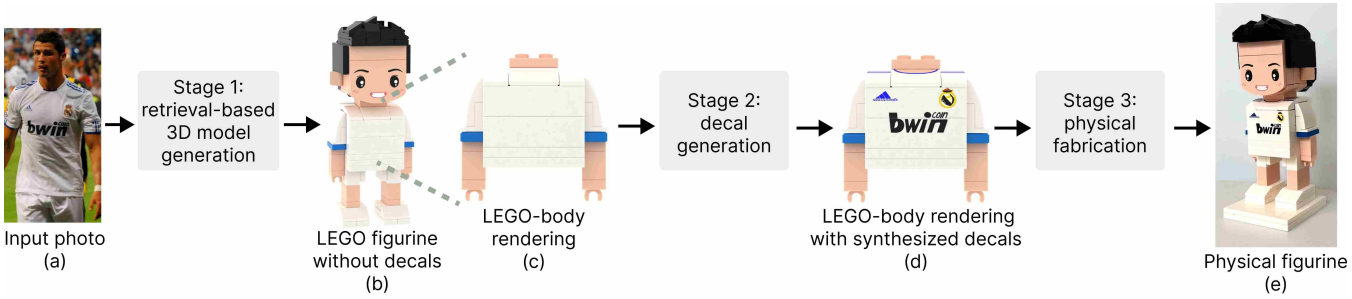

Fig. 3. Overview of our three-stage pipeline. Stage 1: we take a CLIP-guided approach to retrieve and produce a 3D LEGO® figurine model (b) that best resembles the overall appearance of the human in input (a). Stage 2: we design a decal generation module to capture intricate details and synthesize decals (d) on the LEGO® figurine body (c). Stage 3: we print the decals onto LEGO® bricks, then assemble the bricks into the LEGO® figurine (e). Image ©Jan S0L0.

given target surfaces. As there have been numerous works designed for fabricating various types of objects, we discuss mainly works on different object types rather than being exhaustive.

LEGO® generation. Our work aligns with research on LEGO® generation. The first work is by Gower et al. [1998] on automatic construction of LEGO® models with regular bricks. Subsequent studies focus on creating voxelized LEGO® models from colored 3D models. Some follow-up works include Petrovic [2001], who use an evolutionary algorithm, Winkler [2005], who exploit beam search, Testuz et al. [2013], who propose a graph-based algorithm, Stephenson [2016], who suggest a multi-phase method, and Lee et al. [2018], who develop a genetic algorithm. Considering not only target shape but also brick colors and structural stability, Luo et al. [2015] aim to construct buildable LEGO® models at larger scales. To facilitate LEGO® construction with AI assistance, Lennon et al. [2021] attempt to use neural networks to convert real-world photographs to simple voxelized LEGO® models.

More recently, various methods were designed for constructing different types of LEGO® models, *e.g.*, LEGO® houses with slope bricks [Zhou et al. 2019], LEGO® Technic models [Xu et al. 2021], layered LEGO® sketches [Zhou et al. 2023], *etc.* In this work, we aim at personal fabrication and focus on constructing 3D LEGO® figurines from portrait photos. This task has not been explored in prior works and we aim particularly to minimize the necessity of having professional inputs in the computational pipeline.

Image stylization. If we ignore the requirement of physical buildability, our task can be viewed as a unique form of stylization, in which we transform a portrait photo into a LEGO® figurine. Since the advent of GANs, there has been a surge in efforts on creating stylized portraits from photographic portraits. Various architectures, *e.g.*, those by Zhu et al. [2017] and Karras et al. [2020; 2019], were proposed for image stylization. Cao et al. [2018] transform face photos into caricatures with geometric exaggeration, whereas Song et al. [2021] generate stylistic portraits through inversion-consistent transfer learning. Later, Gal et al. [2022b] demonstrated the adaptation of StyleGAN to new artistic domains, guided by CLIP [Radford et al. 2021], achieving the task of text-driven toonification.

Following the significant achievements of diffusion models in text-to-image generation, many diffusion-based image-to-image methods emerge, as highlighted in [Wang et al. 2022] and [Saharia et al. 2022]. Inspired by Tumanyan et al. [2022], Kwon and Ye [2023] successfully transform the appearance of an object into a target domain, while maintaining the image structure. Tumanyan et al. [2023] further perform text-driven image-to-image conversions by manipulating the cross-attention layers. Motivated by Gal et al. [2022a], Zhang et al. [2023b] treat the style of a painting as a textual description that can be learned. Zhang et al. [2023a] disentangle material, style, and layout in a single image by altering the conditioning prompts at different diffusion stages.

Very recently, several research studies [Chen et al. 2023a,b; Gal et al. 2022a; Jia et al. 2023; Li et al. 2023; Ruiz et al. 2023; Shi et al. 2023] demonstrate successful synthesis of images, reconstructing a specific object from multiple reference images, leading to the development of reference-image-controlled generation. Following a similar spirit, some works reconstruct appearance in image-to-video tasks using symmetric U-Nets [Hu et al. 2023; Xu et al. 2023].

Examining their results, we can see that although they can be employed to produce images in LEGO® style, the LEGO® models in the generated images are usually not physically realizable, as demonstrated later in Figure 4. This important requirement significantly distinguish our task from the image stylization task. Also, we are required not just to abstract major structural elements such as ties and belts into a cartoon style, but also to make best effort to preserve intricate details such as logos, text, and patterns, as well as to further rectify these details on the LEGO® figurines.

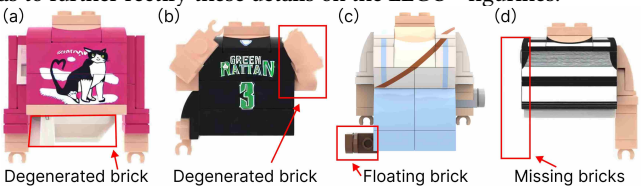

Fig. 4. Different types of artifacts in synthesized LEGO®-body images produced by existing generative models: (a,b) degenerated LEGO® brick not existing in standard LEGO® brick set, (c) floating LEGO® brick disconnected from the main model, and (d) missing bricks from the LEGO® body.

Fig. 5. We aim to produce a LEGO® figurine (b) that resembles the human in the input portrait (a), with decals specifically generated on the body (c) to abstract and preserve intricate details. Image ©Alexandra Walt.

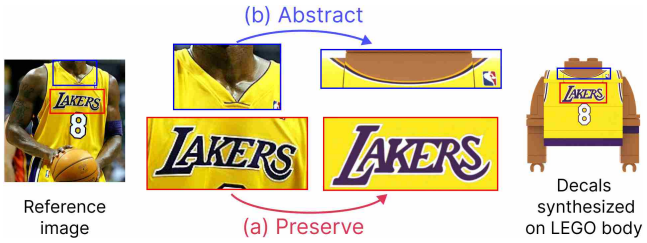

Fig. 6. Our method should *preserve* meaningful elements such as the text in (a), while *abstracting* major structural elements like the collar into a cartoon style, without wrinkles and shadings; see (b). Image ©Alexandra Walt.

3 OVERVIEW

Task definition. Given an input portrait photo, our objective is to create a physically-assemble-able LEGO® figurine model that resembles the human subject in the portrait photo. The generated LEGO® figurine should reproduce prominent features, such as hairstyles and garment styles, primarily by creating an assembly of LEGO® bricks. Besides, the figurine should capture intricate and distinctive details, such as stripes, plaids, collars, text, logos, and accessories. See Figure 5 for an illustration of the task.

Challenges. The task has three challenges, primarily stemming from the substantial domain gap between portrait photos and LEGO® figurines, while requiring the figurines to be physically realizable.

First, regarding the building of the LEGO® figurine, we have to utilize LEGO® bricks, which have limited 3D shapes, to reproduce the human subject in the portrait photo. To achieve so involves identifying prominent features, such as hairstyle, clothing style, and garment color, and carefully arranging LEGO® bricks accordingly.

Second, we need to address the domain gap between input portrait photos and target LEGO® figurines. Particularly, we need to connect the two domains to allow effective translation of garment details and structural elements from portrait photos to decals produced on the LEGO® figurines. More fundamentally, we need to *preserve* meaningful details while *abstracting* major structural elements (*e.g.*, ties and buttons) by disregarding the influence of distortion, occlusion, material, and lighting in the photos; see Figure 6 for examples.

Last, we need to ensure physical realizability of the LEGO® figurines, meaning that the generated LEGO® figurines should be physically buildable by LEGO® bricks. Though existing generative methods such as [Rombach et al. 2022] can effectively produce LEGO®-style images, various artifacts that are not physically realizable often occur, *e.g.*, the use of non-existing LEGO® bricks, floating bricks, or missing bricks, as demonstrated in Figure 4.

Our approach. To ensure physical realizability of the LEGO® figurines while reproducing prominent features and intricate details, we design a three-stage computational pipeline illustrated in Figure 3. Importantly, we first design Stage 1 with a CLIP-guided retrieval mechanism to ensure the physical realizability of the generated figurine models. Then, in Stage 2, we focus on capturing the intricate details in the portrait photos and reproducing them as decals generated on the LEGO® figurine bodies. Last, Stage 3 focuses on the physical production of the LEGO® figurine models.

Stage 1: The retrieval-based 3D model creation aims to generate a 3D LEGO® figurine model without decals that best reproduces prominent features in the input portrait photo. To do so, we first prepare a dataset of LEGO® figurine models designed by expert designers. Then, we obtain an aligned CLIP space that helps us to quantify the similarities between the portrait photos and LEGO® figurines in dataset. Next, we retrieve the most similar LEGO® figurine model from the dataset, as guided by the aligned CLIP space. Last, we modify the brick colors in some local areas (if needed), following the associated colors of the input photo. See Section 4.

Stage 2: Decal generation is dedicated to the creation of decals on the LEGO® figurine body derived from Stage 1. Here, we aim to synthesize a 2D image of the LEGO® figurine body, with decals generated through a diffusion-based symmetric U-Nets architecture. The first U-Net is the *decal generator*, which iteratively denoises a random vector to generate decals on the LEGO® figurine body. The second U-Net is the *appearance extractor*, which is for capturing intricate details such as distinctive garment features from the input photo, such that these features can then be integrated as conditions on the decal generator after a feature fusion. This scheme is specifically designed to enable the decal generator to adaptively reproduce different areas in the portrait photo. See Section 5.

Stage 3: Physical fabrication aims to prepare for the physical production of the LEGO® figurine, *e.g.*, compiling the brick set, uv-printing decals on some of the LEGO® bricks, generating the assembly instruction, *etc.* See Section 6.

4 STAGE 1: RETRIEVAL-BASED 3D MODEL GENERATION

Beginning with the input portrait, Stage 1 aims to generate a 3D LEGO® figurine model that is physically realizable, while capturing prominent features of the human subject in the input. Hence, we ignore intricate details and decals, and leave them to Stage 2.

Procedure-wise, we undertake three subtasks. First, we train two mapper networks using the pre-trained CLIP model [Radford et al. 2021]. These networks map both the input portraits and LEGO® figurine renderings into the same latent space, which we refer to as the *aligned CLIP space*. Second, we built a dataset with around

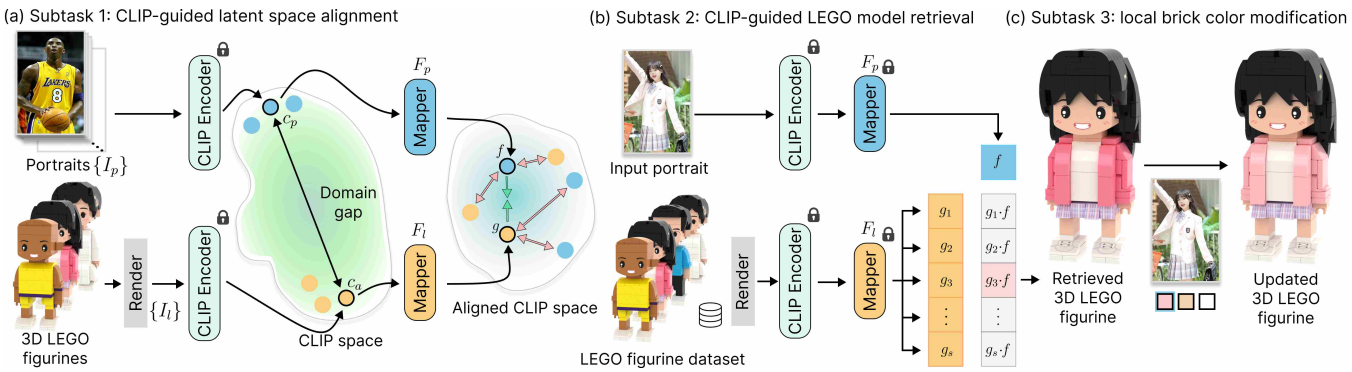

Fig. 7. Overview of the Stage 1. (a) First, using a batch of (portrait, LEGO® figurine) pairs, we train mapper networks F_p and F_a using a contrastive learning strategy to align the CLIP codes of portrait photos and LEGO® figurine renderings and to obtain an aligned CLIP space. (b) During inference, given an input portrait, we retrieve the LEGO® figurine model in the dataset that best matches the portrait by comparing feature vectors in the aligned CLIP space. (c) We replace mismatched colors with the nearest standard LEGO® colors, following the dominant colors in the portrait. Top-left corner image ©Alexandra Walt.

40k expert-designed LEGO® figurines, from which we retrieve the LEGO® figurine that best matches the input portrait in the aligned CLIP space. Note that the geometries of the LEGO® figurines have relatively small variations, *e.g.*, they have similar overall body shapes and vary mainly on hairstyles and clothing styles. Since we focus on the geometry in this stage, we find that the dataset already covers most variations. Lastly, our method automatically modifies brick colors in certain local regions to enhance the visual resemblance to the input. Figure 7 overviews the three subtasks in Stage 1.

Subtask 1: CLIP-guided latent space alignment. To capture high-level semantic features, we use the pre-trained CLIP image encoder [Radford et al. 2021] on both the input portrait and the LEGO® figurine rendering image. As shown in Figure 7(a), for each pair of portrait I_p and LEGO® figurine rendering I_l, we encode them into CLIP codes $c_p = E(I_p)$ and $c_l = E(I_l)$, where E is the pre-trained CLIP image encoder. Specifically, rather than using the final embedding $c_{final} \in \mathbb{R}^{1\times1024}$ after the max pooling layer, we opt for the CLIP code in the last hidden layer $c_{hidden} \in \mathbb{R}^{257\times1024}$, aiming to retain detailed semantic information before the information is filtered out by the pooling.

Due to the intrinsic difference between the portraits and LEGO® figurines and also the scarcity of LEGO® figurine images in the training data of CLIP, CLIP codes $\{c_p\}$ and $\{c_l\}$ can be very different in the CLIP space. In fact, Figure 8 shows the two sets of codes are highly clustered and separate from one another in the CLIP space. To mitigate this domain gap, we adopt two lightweight mapper networks, F_p and F_l, to transform CLIP codes c_p and c_l, respectively, into an aligned CLIP space; see the blue and yellow boxes in Figure 7(a). F_p and F_l have the same architecture, both comprising two layers of feed-forward neural networks and one max pooling layer, yet they do not share parameters. Using them, we can then obtain transformed CLIP codes $f = F_p(c_p)$ and $g = F_l(c_l)$. The dimensions of f and g are $\mathbb{R}^{1\times1024}$ after the max pooling. Note that we tried alternative architectures, yet found that using our CLIP+mappers architecture leads to the best performance, as it better preserves prior information and can be effectively adapted to new tasks.

Fig. 8. Visualizing the CLIP codes of a randomly-selected subset of 1,500 pairs of portraits and LEGO® figurine renderings. The CLIP codes are projected onto the 2D domain via the t-distributed stochastic neighbor embedding (t-SNE). The CLIP codes of the portraits (blue) and the figurine renderings (yellow) are highly clustered and separated from each other.

To train the mappers, we formulate a CLIP-style contrastive learning approach. Here, we consider a batch of final embeddings of LEGO® figurine rendering and portrait pairs, denoted as $\mathcal{B} = \{(f_i, g_i)\}_{i=1}^{B}$, where B is the batch size. Our goal is to optimize the mappers, such that we can maximize the similarity between the final embeddings of (f_i, g_i) in the batch and minimize the similarity between all mismatched pairs $(f_i, g_j)_{i\neq j}$; see Figure 7(a). Formally, the training loss is defined as the sum of the cross entropy for portrait-to-figurine and figurine-to-portrait pairs across the batch:

$$L = \mathbb{E}\left[H_{\mathcal{B}}\left(\mathbf{v}^{p2l}(f_i), \mathbf{d}^{p2l}(f_i)\right) + H_{\mathcal{B}}\left(\mathbf{v}^{l2p}(g_{i'}), \mathbf{d}^{l2p}(g_{i'})\right)\right],$$

where $\mathbb{E}$ denotes the expectation; $H_{\mathcal{B}}$ denotes the cross entropy for the training batch $\mathcal{B}$; $\mathbf{v}$ denotes the one-hot vector that designates the actual matching between the LEGO® figurine rendering and portrait within the training batch $\mathcal{B}$; and $\mathbf{d}$ denotes the vectors of the softmax-normalized dot products formed by one embedding in one domain and all the embeddings in both the positive and negative pairs in the other domain, *i.e.*, $\mathbf{d}^{p2l}(f_i)$ denotes the probability

distribution over the similarities of LEGO® figurines for a certain portrait, whereas $\mathbf{d}^{l2p}(g_{i'})$ denotes the analogous probabilities from LEGO® figurines to portraits. Specifically,

$$\mathbf{d}^{p2l}(f_i) = \left[\frac{\exp(f_i \cdot g_1)}{\sum_{j=1}^{B} \exp(f_i \cdot g_j)}, \cdots, \frac{\exp(f_i \cdot g_B)}{\sum_{j=1}^{B} \exp(f_i \cdot g_j)} \right]$$

$$\text{and } \mathbf{d}^{l2p}(g_{i'}) = \left[\frac{\exp(g_{i'} \cdot f_1)}{\sum_{j=1}^{B} \exp(g_{i'} \cdot f_j)}, \cdots, \frac{\exp(g_{i'} \cdot f_B)}{\sum_{j=1}^{B} \exp(g_{i'} \cdot f_j)} \right].$$

Subtask 2: CLIP-guided LEGO® model retrieval. Next, given a portrait photo, we need to retrieve a similar LEGO® figurine from the set of pre-built LEGO® figurine models. Specifically, on the one hand, we first render these models into images $\{I_{l,i}\}$ and pre-compute embeddings $g_i = F_l(E(I_{l,i}))$. On the other hand, we encode the input photo I_p as $f = F_p(E(I_p))$. Then, we can compare f against all $\{g_i\}$ in the aligned CLIP space by calculating the dot product between f and each g_i. Next, we can take the LEGO® figurine with the largest dot product value as the retrieved 3D model; see Figure 7(b) for an illustration. Also, we explored alternative approaches such as directly using the CLIP codes without the mappers or training feature extractors from scratch without CLIP; both approaches yield inferior results, due to latent-space misalignment or lack of high-level semantics; see Section 7.3 for experimental results. Note that with the aligned CLIP space, our approach is naturally scalable for retrieving in dataset with any size.

Subtask 3: Local brick color modification. The colors of the LEGO® pieces in the retrieved figurine model may not always match the colors of the related image elements in the input, so we first extract the dominant colors from the portrait photo using color histograms. Then, we register the colors shown in the surface bricks and repaint them with the closest matches from the dominant colors. To ensure the brick color is available, we replace the color with matches from the LEGO® palette [BrickLink 2024]; see Figure 7(c). In the end, the updated LEGO® body is sent to Stage 2 for synthesizing appropriate decals that match the garment details in the input photo. Then, in Stage 3, we stitch the synthesized decals back onto the 3D model to produce a physically-assemble-able LEGO® figurine.

Discussion on retrieval-based creation. Instead of directly creating the mesh of the LEGO® figurine models with neural networks, we find a retrieval-based approach more reliable and effective. Retrieving a model from the dataset and modifying the specific details can produce a satisfactory result. Our objective is to construct a model that not only visually resembles the intended portrait but is also physically assemble-able using LEGO® bricks. This necessitates ensuring that the model comprises multiple manifolds, adheres to brick connection policies, and conforms precisely to the standard of pre-fabricated LEGO® bricks. Given the current state of 3D neural network technologies, these complexities surpass their processing capabilities. We will leave this as future work.

Discussion on retrieval strategy. One alternative way to accomplish the task in Stage 1 is to find the most similar portrait photo in our dataset with respect to the input portrait and take the associated LEGO® figurine as the retrieved template. However, we did not consider this option because portrait images are highly diverse and often contain irrelevant information, *e.g.* ,background, lighting, image style, human poses, *etc.*, which are not relevant to figurine creation. Thus, the most similar portrait in the dataset with respect to the input portrait may be heavily influenced by the irrelevant information, and the corresponding LEGO® figurine model may not represent the focused features well. On the other hand, the styles of LEGO® figurine models are uniform, demonstrating design variances only in the focused features, *e.g.* , the overall shapes of the hair and the clothes. Another benefit brought by our choice of approach is the extensibility of our dataset used for retrieval. While it is challenging to manually collect pairs of (portrait, LEGO® figurine) data entries, it is relevantly easier to augment the collection of figurines by exchanging the parts among different figurine models. After the network models are trained, we can augment the figurine collection to retrieve results with more variants.

5 STAGE 2: DECAL GENERATION

The goal of Stage 2 is to generate decals onto the LEGO® figurine body output from Stage 1, such that the decals depict the planar-based details in the portrait photo, such as logos, text, patterns, and any other distinctive garment details. In particular, the generated decals should meet four requirements: (i) They should depict the major structural elements (*e.g.*, ties and buttons) in a cartoon style, characterized by clear edges, smooth color shading, and relatively simple textures. (ii) They should preserve meaningful elements, like logos, text, and unique embellishments; (iii) The decals need to be appropriately sized and positioned to align with the LEGO® bricks, as exemplified in Figure 10. (iv) The decals should accurately reflect the original appearance of the figure, unaffected by any distortion, wrinkles, lighting, or occlusions present in the input portrait.

Overall, the four requirements are summarized empirically from the figurine data. The first two intrinsically conflict with each other, since requirement (i) demands a simplified expression of the major structural elements, while requirement (ii) requires the preservation of meaningful details, see Figure 6 for examples. Like human experts, our model should adaptively reproduce different garment details in the input portrait in distinct ways. Below, we present the details of our approach, which is designed to produce decals that meet the four requirements. See also Figure 9 for the overall procedure.

Preprocessing. To direct the neural network to focus decal generation on the body area, we first preprocess the input portrait by cropping out the body area and removing the background. Similarly, we isolate the body part from the LEGO® figurine model generated from Stage 1. Please refer to Figure 9 for an example. Technically, we use YOLOv8 [Jocher et al. 2023], trained on the DeepFashion2 [Ge et al. 2019] dataset, to crop the body part from the input portrait. Following this, we segment and remove the background from the cropped image using the Segformer model [Xie et al. 2021] fine-tuned on the ATR (Annotated Human Parsing) datasets [Liang et al. 2015a,b] for clothes segmentation. The processed image, referred to as the reference image x_{ref}, is then prepared for the subsequent decal generation phase. For the figurine model, we extract its body part and produce a rendered image from the front view, denoted by x_{body}. By then, our model is trained to learn to generate decals on x_{body} according to the feature elements in x_{ref}.

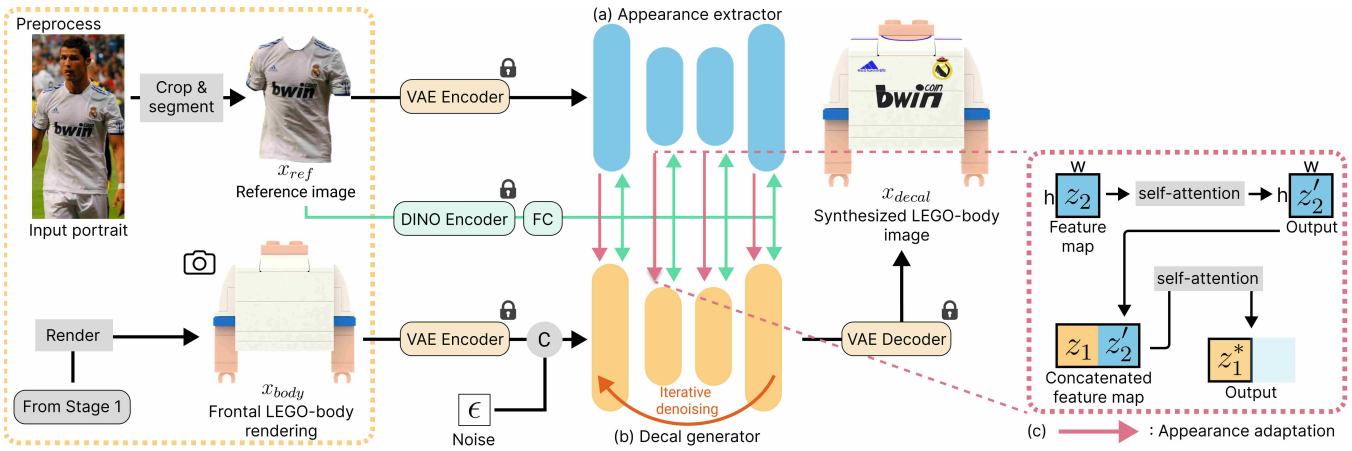

Fig. 9. Overview of the Stage 2 pipeline. In the preprocessing, we crop and segment the input portrait to form a reference image x_{ref}, and produce a frontal rendering x_{body} of LEGO® body from Stage 1. Then, we adopt a symmetric U-Nets architecture, containing (a) the appearance extractor for extracting multi-scale garment information from x_{ref} and (b) the decal generator for generating synthesized LEGO®-body images x_{decal} by iteratively denoising a random noise. The LEGO®-body rendering is encoded with a VAE encoder and concatenated with the initial noise. The features of reference image are extracted with DINO-V2 encoder and appearance extractor separately, and fed into the decal generator via cross-attention and appearance adaptation, respectively. Image ©Jan S0L0.

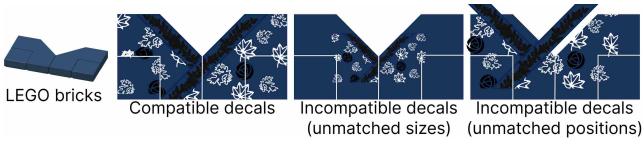

Fig. 10. Compatible decals should well align with the underlying LEGO® bricks to reproduce the intricate garment details. Incompatible decals may have unmatched sizes or positions with respect to the LEGO® bricks.

Diffusion-based decal generation. Our overall generative framework is based on the Stable Diffusion (SD) model. To extract multi-scale features from the reference image, we adopt a symmetric U-Nets architecture. First, the decal generator (Figure 9(b)) is the denoising U-Net inherited from SD. It iteratively removes random noises to yield synthesized LEGO®-body images with decals, namely x_{decal}. Second, in parallel to the decal generator, we use another U-Net called the appearance extractor to extract multi-scale garment information from the reference image. The appearance extractor has the same architecture and initial parameters as the decal generator. Yet, their parameters are independent. As Figure 9 shows, garment information extracted by the appearance extractor can then be taken as a condition on the decal generator via an appearance adaptation strategy. Besides, from the reference image, we use the DINO-V2 [Oquab et al. 2023] image encoder to extract instance-level information, which is conditioned via cross-attention. Since DINO-V2 is pre-trained on a large-scale dataset in a self-supervised manner, it can effectively map objects to a feature space invariant to augmentations such as lighting and twisting, so it is a good fit for extracting garment details under a general shooting condition.

Specifically, in each U-Net, there are 24 blocks in four resolutions (64×64, 32×32, 16×16, 8×8) with each resolution replicated three times in both down-sampling and up-sampling stages. We sequentially perform appearance adaptation and cross-attention in each block. The appearance adaptation is described as follows; see also Figure 9(c). For each block, we aim to adapt the appearance feature map $z_2 \in \mathbb{R}^{h\times w\times c}$ from the appearance extractor into the corresponding feature map $z_1 \in \mathbb{R}^{h\times w\times c}$ from the decal generator. First, we perform self-attention on feature map z_2 to refine it into feature map z'_2. Then, we concatenate feature maps z_1 and z'_2 along the w dimension and perform self-attention on $[z_1, z'_2]$. After that, we take the first half of the resultant feature map along the w dimension, *i.e.*, z^*_1, as the input for the subsequent cross-attention. For the instance-level information, we use DINO-V2 to encode the reference image x_{ref} into a global token $I_g^{1\times 1536}$ and a series of patch tokens $I_p^{256\times 1536}$. The two types of tokens are concatenated to provide more image context. Further, we use a fully connected layer to project and align these tokens with the text embedding space of Stable Diffusion (SD). We then perform cross-attention between the resultant token $I^{257\times 768}$ and the feature map from each U-Net block to fuse the instance-level embedding into the feature map.

On the other branch, we take x_{body} as another condition to the decal generator. Here, we encode x_{body} using the SD VAE encoder to the embedding z_{body} and concatenate z_{body} onto a randomly-sampled noise as the input to the decal generator.

Discussion on Symmetric U-Nets. Next, we discuss the motivation and insight of our model, illustrating why it successfully tackles the challenging requirements and generates compelling results. Fundamentally, the symmetric U-Nets architecture is inspired by the image-to-video (i2v) tasks [Hu et al. 2023; Xu et al. 2023] for handling the appearance consistency across video frames. Our task, on the other hand, requires preserving low-level information in addition to pure high-level semantics in the reference image. The symmetric U-Nets offer the capability to selectively generate stylized major-structure garment features (*e.g.*, stripes, plaids, collars,

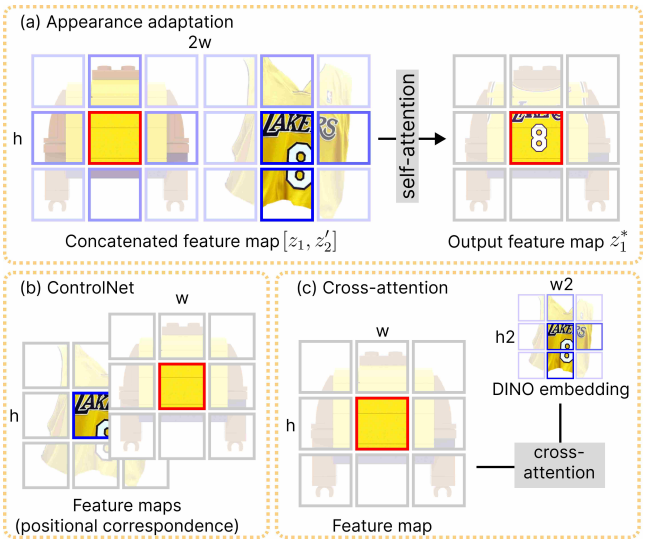

Fig. 11. An intuitive comparison of three different designs. Take the red box as an example patch; the black box denotes the patch without attention, and the intensity of the blue box denotes the degree of attention that the example patch pays. In our appearance adaptation, the example patch pays attention to both feature maps from the denoising U-Net and the appearance extractor. In ControlNet, the example patch pays attention only to the patch sharing the same position in the feature map from ControlNet. In cross-attention, the example patch pays attention only to the patches in the embedding from the DINO encoder, where the reference image is down-sampled. Image ©Alexandra Walt.

buttons, etc.) or copy-and-paste meaningful elements, such as logos and text, from the reference image. This is attributed to the fact that a specific feature map patch z_1 in the decal generator block pays attention to the feature map patches in some other positions, both in the decal generator and appearance extractor (z_1 or z_2). See the illustration in Figure 11(a). So, our model can learn stylistic correlation between the generated decals and reference images.

Further, we compare our approach with two alternatives. First is the ControlNet architecture [Zhang et al. 2023c], in which every feature map patch in its denoising U-Net receives strong condition only from its positionally-corresponded feature map patch in the standalone ControlNet; see Figure 11(b). Though it has great capability in preserving structural conditions such as edges, depths, and poses, it cannot handle our task, as there is no explicit positional correspondence between the reference image and LEGO®-body rendering. The second alternative is to directly extract information from the reference image using solely a pre-trained image encoder such as DINO via cross-attention; see Figure 11(c). Yet, doing so often fails to preserve detailed elements in the reference image due to the information loss after downsampling. Note that the input size for both the appearance extractor and ControlNet is 512×512, yet the input for the DINO-V2 encoder is down-sampled first to 224×224. Further, these encoders are not designed for extracting multi-scale information like the appearance extractor, so the resultant embeddings cannot effectively encapsulate multi-level information of the garments in the reference image; see Figure 18 for examples.

To conclude, this work is the first that adopts the symmetric U-Nets architecture to the image-to-image style transfer task, showcasing its strong capability to selectively generate or preserve various garment elements in the decal generation.

Training strategy. We initialize the models of the decal generator and appearance extractor using the pre-trained weights from the Stable Diffusion. The fully connected layer for the DINO-V2 encoder is initialized using Gaussian weights. The weights of the VAE encoder and decoder, as well as the DINO-V2 image encoder, are all fixed. We denote the decal generator as ϵ_θ. It takes the image embedding I and feature maps z_θ from appearance extractor as conditions to predict noise $\epsilon \sim \mathcal{N}([0, 1])$. The loss is formulated as

$$\mathcal{L} = \mathbb{E}_{z,t,I,z_\theta}\left[\|\epsilon - \epsilon_\theta(z_t, t, I, z_\theta)\|_2^2\right],$$

where z is the ground-truth image latent embedding; t is the diffusion timestep; and z^t is the noisy latent concatenated with the latent code z_{body} of the frontal LEGO®-body rendering x_{body}.

To effectively capture the conditions, we also adopt the classifier-free guidance (CFG) with the symmetric U-Nets. In particular, we drop the image embedding I and feature maps z'_θ at a fixed rate p during training. At the inference, the CFG is formulated as

$$\epsilon_{pred} = \epsilon_{uc} + \beta_{cfg}(\epsilon_c - \epsilon_{uc}),$$

where ϵ_{pred}, ϵ_{uc}, ϵ_c, and β_{cfg} are the model's final output, unconditional output, conditional output, and the CFG weight, respectively.

Inference. During the inference, we employ the trained models to iteratively denoise a Gaussian latent code z under the condition of the reference image x_{ref} and the frontal LEGO®-body rendering x_{body}. Then, we generate renderings of the LEGO® body with decals; see Figure 9. After that, in Stage 3, we can further reproject and uv-print the synthesized decals onto the associated LEGO® bricks for creating the physical figurine assemblies.

6 STAGE 3: PHYSICAL FABRICATION

Stage 3 aims to produce the physical LEGO® figurine for the input portrait. Figure 12 outlines the three major steps in Stage 3.

Step 1: Decals reprojection. Given the synthesized LEGO® body image x_{decal} from Stage 2, we first project the decals onto associated LEGO® bricks in the LEGO® figurine model from Stage 1. This requires establishing a UV map between the 3D model and front-view rendering. Hence, we adopt the virtual camera for rendering in Stage 2 to project each brick surface (see the colored rectangles in Figure 12(a)) of the front-facing LEGO® bricks in the 2D rendering space. Each projected surface encloses a sub-region in x_{decal}, thus defining a UV map between individual brick and the synthesized LEGO®-body image. See Figure 12(a) for an example.

Step 2: UV printing. Then, we imprint the decals onto the LEGO® bricks using a UV Flatbed printer. To do so, we designed specialized fixtures to secure the bricks and raise their heights to the same level above the printing bed, enabling simultaneous printing of multiple bricks. In a single printing batch, we designate the bricks and decals to be printed, concatenate the decals into a frame, and set the layout for the bricks that need to be printed. After loading the settings and the fixtures with the bricks into the printer, the varying bricks are

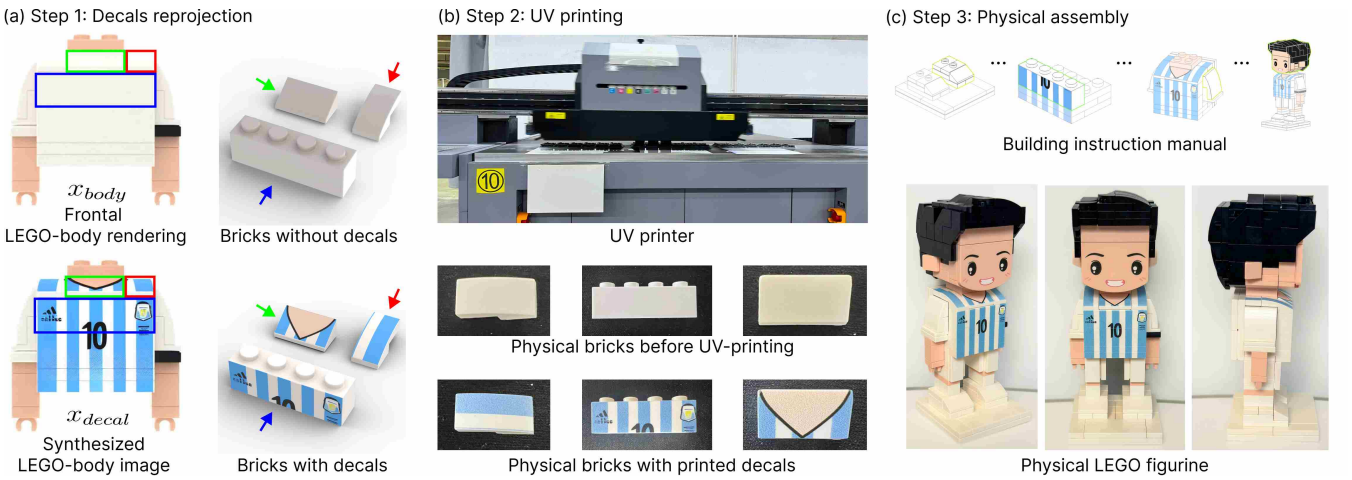

Fig. 12. Overview of the Stage 3 pipeline: (a) the reprojection of decals from the synthesized LEGO®-body image to the 3D LEGO® figurine model, (b) the image of the UV printer and imprinted LEGO® bricks, and (c) the physically-assembled LEGO® figurine. The projected surfaces of the selected bricks are bounded by the red, green, and blue rectangles on x_{body} and x_{decal}, respectively.

printed simultaneously. See Figure 12(b) for the UV printer and example LEGO® bricks uv-printed with decals.

Step 3: Physical assembly. Finally, our system automatically compiles a list of LEGO® bricks for creating the LEGO® figurine model. To aid user assembly, we also generate step-by-step assembly instructions, produced by iteratively removing bricks from the full assembly without causing collisions; see Figure 12 (c). Note that the assembly sequence is determined in Stage 1, so the decal generation stage does not affect the generated sequence. Readers can refer to Figure 1 again for physically-assembled LEGO® figurines that are meticulously created from different input portraits.

7 RESULTS AND EXPERIMENTS

7.1 Implementation Detail

In Stage 1, to train the CLIP mappers, we first pre-compute and store the CLIP codes of both the LEGO® figurine renderings and portrait photos. Then, we use four NVIDIA A100 GPUs to train the networks. This training completes in 300 steps, utilizing a batch size of 37,000 and a learning rate of 2e-4. In Stage 2, we train our generative model on the same set of GPUs for 30,000 steps, with a batch size of 48, learning rate of 1e-5, and dropout rate of 0.2. At the inference, we utilize the denoising diffusion probabilistic model sampler with 50 denoising steps and a guidance scale of β_{cfg}=6.

All the models are trained on a dataset comprising 37,000 training entries and 3,000 test entries. Each of these entries contains a real human portrait paired with an associated 3D LEGO® figurine model. The portrait photos are collected from various online sources, while the figurine models are crafted by expert designers, following the requirements detailed in Section 3. The renderings on the LEGO® figurines are produced using the Blender software [Community 2024] and are incorporated into the dataset together.

7.2 Results

To evaluate the quality of the final results generated by our three-stage pipeline, we prepare a standalone set of eight hundred portraits featuring diverse styles as the user inputs. Distinct from the dataset mentioned previously, the image styles in the set range from real human portraits to anime images and oil paintings. We then leverage our pipeline to generate LEGO® figurines for each of these photos.

Gallery. Figure 13 displays ten of the generated figurines alongside their input photos, demonstrating its robustness to handle unconstraint input photos effectively. Figure 14 presents seven additional figurines, featuring some iconic figures. This collection shows the capability of our approach to reproduce various hair and garment styles using a variety of brick shapes, ranging from the traditional rectangular pieces to those with slopes and curves; see Figures 13 (d, f) and Figure 14 (e). Furthermore, our method effectively preserves the meaningful details in the portraits, making results more identifiable; see the garment graphics in Figures 13 (f, h), brooches in Figures 13 (d), and emblems in Figures 14 (a, b). On the other hand, our method also abstracts the major structures and filters out unwanted influences from wrinkles, shading, or distortion on them; see the collars in Figures 13 (g, j) and the buttons in Figures 13 (b, c). In addition, our method demonstrates resilience to imperfect shooting conditions, reproducing the original color without being affected by the varying lighting conditions. For instance, Figures 13 (b, c) are photographed in dark environments, leading to dimmed shading on the garments, yet our method can reproduce the garment color, buttons, and other details with consistent and bright colors. Further, our method demonstrates the capability to re-organize disorganized elements in the input photo. A notable example is the realignment of buttons, as shown in Figure 13 (b). Our method effectively recovers the garment's natural, undistorted appearance and arranges the buttons in their proper direction and order.

Fig. 13. A gallery of LEGO® figurines created by our pipeline, alongside with the corresponding input portraits. Image (a) & (i) created with Outfit Anyone developed by Alibaba Group. Image (b) & (c) ©Universal Pictures. Image (d) ©Navy Petty Officer 1st Class Carlos M. Vazquez II. Image (e) ©Hao Xu. Image (f) ©Tatiana. Image (g) ©Bettmann/CORBIS. Image (h) ©All-Pro Reels. Image (j) ©jamesomalley.

Fig. 14. A gallery of LEGO® figurines created by our pipeline, reproducing a range of iconic figures, including sports stars and manga/movie characters: (a) the football star *Kylian Mbappé*, (b) the football star *Lionel Messi*, (c) *Luffy* in the manga *One Piece*, (d) *Vegeta* in the manga *Dragon Ball*, (e) *Naruto Uzumaki* in the manga *Naruto*, (f) *Forrest Gump* played by *Tom Hanks* in the movie *Forrest Gump*, and (g) *Jack Dawson* played by *Leonardo DiCaprio* in the movie *Titanic*.

Results on occluded photos. Our method possesses strong robustness in handling occluded photos, demonstrating its capability to infer and reconstruct occluded areas. This is evidenced in Figures 13 (c, d), where the figures are not entirely front-facing, leading to partial visibility of their bodies. Despite this, the generated figurines reproduce the garment decals completely.

We further evaluated our method using the iconic Abbey Road album cover by the Beatles, where the band members are positioned nearly completely side-facing, presenting serious occlusions, particularly for their hair and garments. Figure 15 displays both the input photo and the generated figurines. Our approach skillfully captures the hairstyles, clothing styles, and intricate garment details, such as the pockets on the first figure and the buttons on the third figure, from only slightly visible areas.

Results on input photos with multiple artistic styles. Our method also exhibits remarkable versatility with diverse artistic styles, such as anime and oil paintings; see Figure 16 and the teaser. It is noteworthy that our dataset comprises only real human portraits. Yet, even with this limitation, our method can transform garment elements from these varied artistic domains into stylistically-consistent LEGO® decals. Whether dealing with the unique textures of an oil painting or the uncommon costumes of anime characters, our approach effectively recognizes and maintains the essence of the garment.

7.3 Evaluation on Stage 1

We conducted experiments to evaluate our method in Stage 1, comparing its performance with two alternative approaches.

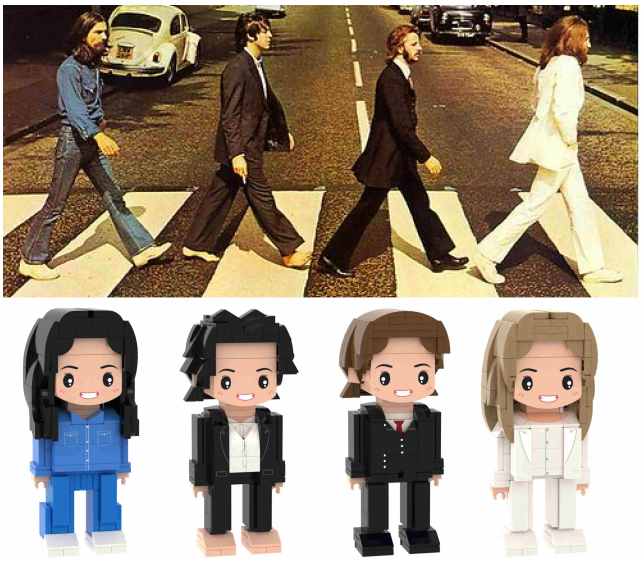

Fig. 15. Four LEGO® figurine models generated from seriously occluded photos. Image ©David Erickson.

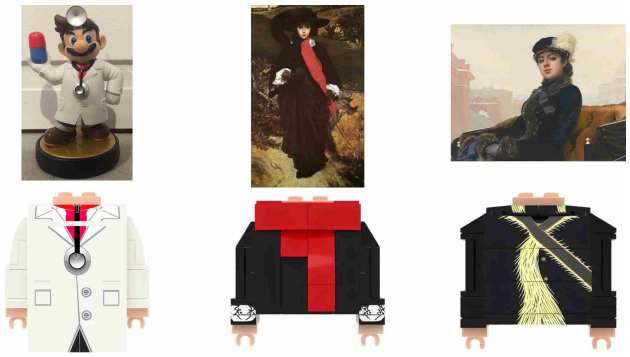

Fig. 16. Our method can handle complicated styles of art, including cartoon and oil painting, and produce neat LEGO® style decals. From left to right, the input images are *The doctor is in!* ©Ant-Man, *Portrait of May Sartoris* by *Frederic Leighton*, and *Portrait of an Unknown Woman* by *Ivan Kramskoi*.

Evaluation metrics. We developed two evaluation metrics to measure the *semantic* and *structural* similarity between renderings of LEGO® figurines. These metrics compare figurines created by our computational method with those designed by expert designers.

The first metric $\mathcal{F}_{\text{CLIP}}$ scores the semantic similarity between two renderings. Particularly, we first employ the CLIP image encoder to map the two LEGO® figurine renderings into the CLIP space, then take the cosine distance between the CLIP embeddings as the metric. Here, $\mathcal{F}_{\text{CLIP}}$ is mainly used to evaluate the capability of our approach to recognize prominent features in portraits, such as hairstyles and clothing styles. The second metric $\mathcal{F}_{\text{SSIM}}$ measures the structural similarity by calculating the SSIM [Wang et al. 2004] scores between two renderings. SSIM is widely used in measuring appearance consistency or structure consistency. Here, we leverage it to compare the brick arrangement of our results against expert-designed ones.

Table 1. Quantitative evaluation of Stage 1 with two alternative approaches. The two metrics measure the semantic ($\mathcal{F}_{\text{CLIP}}$) and structural similarity ($\mathcal{F}_{\text{SSIM}}$) between the renderings of the retrieved LEGO® figurines and the ground truth models, respectively. Compared with the two alternative approaches discussed in Section 4, our approach achieves the highest scores in both metrics.

Approach	$\mathcal{F}_{\text{CLIP}}$ ↑	$\mathcal{F}_{\text{SSIM}}$ ↑
ViT	19.80	76.89
CLIP w/o mapper	25.58	76.18
Ours	**50.37**	**90.29**

Compare with two alternative approaches. In Stage 1 evaluation, we compare our method with two alternative approaches, as discussed in Section 4. In the first approach, we obtain a shared latent space by directly training two vision transformer (ViT) networks, one for the LEGO® figurine renderings, and the other for the input portraits. Note that the networks have a comparable amount of parameters with the CLIP model and are trained in the same contrastive learning manner on our training set. In the second approach, we directly conduct the retrieval in the CLIP space without the alignment of the mapper networks (CLIP w/o mapper). We experiment on the test set and compute the similarity scores between the retrieved results and ground truths.

The results are reported in Table 1. Our approach outperforms the other two approaches in both metrics, indicating a higher degree of semantic and structural resemblance of the retrieved LEGO® figurines. As discussed in Section 4, the inferior performance of the CLIP w/o mapper is mainly attributed to a misalignment in the CLIP latent space between portraits and LEGO® figurine renderings. For ViT, the absence of a pre-trained CLIP image encoder on a large-scale image dataset leads to less representative feature extraction compared to the CLIP codes transformed by our mapper networks. Additionally, training ViT from scratch requires significantly more time (approximately 14 hours on our GPUs) compared to our method (around 1 hour).

Note that we opt for the similarity scores instead of retrieval accuracy since multiple LEGO® figurines could be perceived as representations of the same portrait. In addition, we will perform a local brick color modification to enhance the visual resemblance, which makes retrieval accuracy an incomplete indicator of our method.

7.4 Evaluation on Stage 2

In Stage 2 evaluation, we evaluate the quality of the decals generated using our methods against two types of references: the body part of the original portrait images, and the ground truth LEGO® figurines designed by human experts. We measure the similarity between the generated decals and the references using two metrics, which we will describe in the next section. To test the robustness of our methods, we generate 4,000 decals from 200 portraits in the test set with different random seeds.

Q: Describe the visual elements shown on the clothing.

A (GPT-4V): The individual is wearing a white sports jersey. Prominently displayed across the chest is the logo "bwin.com" in black lettering. The jersey is part of a Real Madrid kit, recognizable by the small Real Madrid crest on the left chest. There are also three dark blue Adidas stripes on the shoulders, a common design element for Adidas apparel.

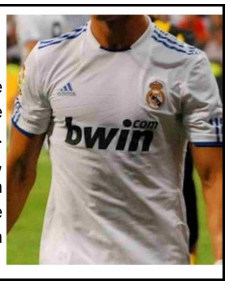

Fig. 17. The annotation process using GPT4-V. Image ©Jan S0L0.

Evaluation metrics. We propose three metrics to quantify the visual resemblance of the generated results with the two types of references, *i.e.*, the input portraits and the ground truths renderings. The first metric $\mathcal{F}_{\text{CLIP}}$ measures the similarity between the synthesized LEGO®-body images and the renderings of LEGO® bodies from ground truths in the CLIP space. The second metric $\mathcal{F}_{\text{CLIP-MAPPER}}$ measures the similarity between the synthesized images and the input portraits in the aligned CLIP space obtained in Stage 1. Specifically, we first map the two images into the CLIP space, then align them with the mapper networks pre-trained in Stage 1. The third metric $\mathcal{F}_{\text{GPT}}$ measures the similarity between the synthesized images and the captions of input portraits. In practice, we first generate the captions for each portrait and then calculate the text-image similarity in the CLIP space.

$\mathcal{F}_{\text{CLIP-MAPPER}}$ and $\mathcal{F}_{\text{GPT}}$ are proposed to bridge the domain gap between the synthesized LEGO®-body images and the portrait photos. $\mathcal{F}_{\text{CLIP-MAPPER}}$ borrows the prior knowledge obtained from Stage 1, facilitating the quantification of the similarity between the two domains, as the comparisons are conducted within the aligned CLIP space. Whereas $\mathcal{F}_{\text{GPT}}$ leverages the most advanced multi-modal large language model, GPT-4V, to generate a caption for each portrait, and applies the widely used CLIP text-image similarity in our specific task. These two metrics align the criteria with human perception by leveraging perceptual knowledge from the pre-trained models. In practice, we present a portrait to GPT-4V and ask it to "Describe the visual elements shown on the clothing"; see Figure 17 for an example. We observe that GPT-4V exhibits an exceptional ability to recognize, describe, and summarize intricate details of the clothing. Thus, $\mathcal{F}_{\text{GPT}}$ is a good indicator in quantifying the perceptual similarity between the reference images and the synthesized LEGO®-body images.

Compare with four alternative approaches. Our method is benchmarked against four baseline approaches for a comprehensive comparison. All four approaches generate decals on the LEGO®-body renderings conditioning on the reference images. However, they differ in the schemes of extracting and fusing the features from the reference images. The four approaches are: (1)*DINO*: we fine-tune a Stable Diffusion model with the text encoder replaced by the DINO-V2 [Oquab et al. 2023] image encoder. (2)*DINO+ControlNet*: we fine-tune a Stable Diffusion model with a standalone encoder network (ControlNet) to extract the multi-level features from the reference images. In addition, the instance-level features are extracted by the DINO-V2 encoder and fused into denoising U-Net and ControlNet. (3)*Text*: We fine-tune a Stable Diffusion model on the caption-LEGO®-body rendering pairs, where the captions are generated by GPT-4V and mapped into the textual space with CLIP text encoder. (4)*Text+ControlNet*: We combine the ControlNet with the CLIP text encoder, leveraging both the structural features of the reference images and their captions. Note that for fairness, in all four baseline approaches, we provide frontal LEGO®-body renderings x_{body} as constraints. We perform the experiment by applying the four approaches to synthesize decals following the same setting as our approach, *i.e.*, to generate 4,000 decals from 200 portraits in the test set. Then, we evaluate the results using the three metrics.

Table 2. Quantitative evaluations in Stage 2. The four baseline approaches all adopt the Stable Diffusion style generative model, yet extract features from reference images in different ways. The three metrics measure the similarity between our generated decals with the ground truth LEGO® -body images, the portraits, and the image captions of the portraits, respectively. Our method outperforms all four baseline approaches in all three metrics.

Approach	$\mathcal{F}_{\text{CLIP}} \uparrow$	$\mathcal{F}_{\text{CLIP-MAPPER}} \uparrow$	$\mathcal{F}_{\text{gpt}} \uparrow$
DINO	55.08	48.08	57.64
DINO+ControlNet	55.13	47.52	56.87
Text	56.24	52.94	64.09
Text+ControlNet	55.27	51.25	64.21
Ours	**60.62**	**61.25**	**66.15**

As reported in Table 2, our method outperforms all four baseline approaches in all three metrics. This indicates that the decals generated by our method are more similar to the ground truth models and the input portraits following human perception. When incorporating ControlNet, a notable drop in similarity with the portraits is observed, suggesting that ControlNet is not an ideal option for our specific task as discussed in Section 5. The underlying reason is that the ControlNet is designed to preserve the structural integrity of a reference image, yet there is no explicit positional correspondence between a portrait and a LEGO® figurine. Additionally, we find that methods using text encoders generally perform better than those with image encoders. This indicates the effectiveness of GPT-4V in capturing semantic information from the reference images. However, these methods fall short of preserving meaningful elements, which arises from the limited capability of text description to encapsulate low-level details.

In Figure 18, we visually compare our approach with the baseline methods. Although the baselines using text encoders can extract the semantic level information from the reference images, they may fail to correctly reproduce the sizes, shapes, and orientation of the garment elements; see the black and white cat in the third row. In addition, GPT-4V sometimes yields incorrect information. For example, the missile in the first row is mistaken for fish and the woman profile (marked by a red rectangle) in the second row is mistaken for zebra. On the other hand, the baselines using DINO encoders barely capture the distinctive features of the garment, only preserving the lower-level information in the reference image, like colors and layouts.

Fig. 18. Qualitative evaluation between baseline approaches and ours. The results are sorted by their visual similarities with both the input portrait photos and the ground truths. The red rectangles highlight the detailed woman profile that can be well captured by our approach, but not by the baseline approaches, nor appears in the ground-truth model.

Our approach creates compelling results that adhere to the multi-level garment details in the input photos faithfully. This is signified by the correct reproduction of various visual elements in the reference images, including (i) text, *e.g.*, the words "BLACK" and "WHITE" in the first row; (ii) graphic prints, *e.g.*, the cat in the third row and the woman profile printed on the clothes, highlighted in the second row; (iii) patterns, *e.g.*, the stripes in the fourth row; and (iv) tiny accessories, *e.g.*, the necklace in the third row.

7.5 Ablation Study

To demonstrate the necessity of Stage 1, we execute our method in Stage 2 without the input of x_{body}. This leads to the generation of decals with noticeable artifacts, as shown in Figure 19. To quantitatively assess the impact of this omission, we employ the $\mathcal{F}_{\text{SSIM}}$ metric. This metric evaluates the structural similarity between the synthesized LEGO®-body images and the ground truth LEGO®-body renderings. The SSIM scores are particularly effective in detecting artifacts in the generated results, as such artifacts typically result in structural inconsistencies in the LEGO® models. When using our full methodology, which includes the x_{body} condition, the average SSIM score was a robust **81.56**. In contrast, when removing the x_{body} condition, the score significantly decreased to **70.09**. This decrease highlights the degraded quality of the LEGO®-body images produced without the x_{body} condition. Furthermore, the absence of x_{body} adversely affected the alignment of the synthesized LEGO®-body images with the corresponding LEGO® figurines. This misalignment posed challenges in the subsequent process of reprojecting the decals onto the LEGO® bricks during Stage 3.

7.6 Interview with Human Designers

Experiment setting. To assess the practical applicability of our approach, we conducted individual interviews with five human designers and requested their evaluations of the results generated by our method. The participants comprised three full-time expert LEGO® designers (P1, P2, & P3), each with over a year of experience in LEGO® figurine design, and two LEGO® fans (P4 & P5), who have been trained to design figurines manually. The interview process was divided into two parts. In Part 1, we asked each participant to provide subjective feedback on our results. Subsequently, in Part 2, we requested the participant to quantitatively assess the results produced by different methods. Below, we describe the details of each part and summarize the interview results for each part separately.

Fig. 19. The qualitative comparison between our full method and the ablated one. The red boxes indicate artifacts. The first three models contain degenerated bricks and the last model exhibits a brick overlap.

Part 1: Subjective feedback. To acquire subjective feedback from the participants, we asked each participant to review 50 results produced by our approach; (see the first 50 results in Part E of the supplementary) and answer the following three questions.

- **Design Quality**: *"Please share your opinion on the overall quality of the result."*
- **Design Speed**: *"For each result, please estimate the time it would take you to manually create a LEGO® figurine model from the same input portrait."*
- **Potential Limitations**: *"Please identify any potential limitations of our approach in practical applications."*

We asked each participant to evaluate one result at a time, estimating the manual design time. Then, after all results were evaluated, the participant was requested to discuss the general design quality of the 50 results and to identify potential limitations. On average, individual interviews lasted approximately 30 minutes.

Feedback summary. Table 3 reports statistics of manual design time estimated by the five participants on the 50 results presented to them. It turns out that the design speed of expert designers (P1-P3) is often much faster than that of general LEGO® fans (P4, P5). Yet, on average, our method takes only 40 seconds to create a LEGO® figurine model. Further, we quote and summarize some of the comments from the participants after reviewing all results below:

- **Design quality**: P1: "The design restores the symmetry of clothing patterns, even when these patterns are distorted in the portrait." P2: "The designs capture the distinct characteristics of different clothing styles and exhibit high-quality decals." P4: "The results reproduce small details I initially overlooked from the portrait." Overall, all participants assessed our results as good.
- **Design speed**: P2: "It takes me about ten minutes to create a LEGO® figurine if the decals can be matted from the portrait or found in the previous designs." P3: "For simple figurines, it takes about fifteen minutes, but a figurine with highly complex decals that need to be drawn from sketch will take more." P5: "Creating a figurine with detailed decals takes me more than half an hour, and achieving the quality of the results presented is challenging." The cost time mainly depends on the proficiency of the designer and the complexity of the decal.
- **Potential limitations**: P1: "Sometimes, the template model without decals does not closely match the portrait, which compromises the final quality." P4: "Details such as emblems and text lines are blurry, which could affect user satisfaction." According to the feedback, potential limitations arise from mismatches in the LEGO® body part from Stage 1 and the limited resolution of the output from Stage 2.

Table 3. Statistics of design time to manually create a result estimated by each participant.

Participant	Estimated Time (minute) Min	Mean	Max
P1	8	14.5	18
P2	5	10.5	18
P3	5	10.1	20
P4	30	87	120
P5	30	44	80

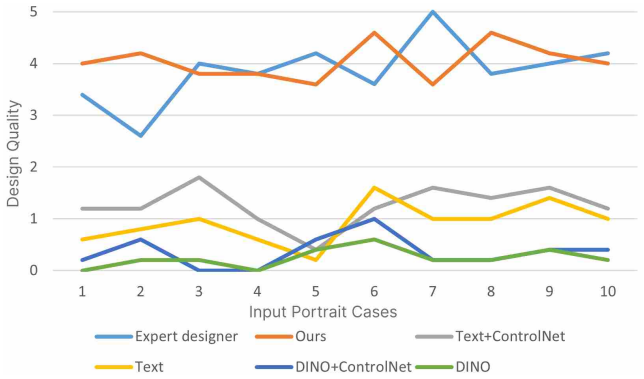

Fig. 20. Quantitative comparison on the design quality of our approach, human experts, and four baseline approaches, on ten input portrait cases (horizontal axis). The line charts present the average design quality score of each result assessed by the five participants, revealing that our approach produces high-quality results comparable with those crafted by human experts, while greatly outperforming the baselines.

Part 2: Quantitative comparison. In Part 2, we evaluate the design quality of our approach relative to that of human experts and four baseline approaches: *Text+ControlNet*, *Text*, *DINO+ControlNet*, and *DINO* (refer to Section 7.4 and Figure 18). We randomly selected ten input portraits from our test set and obtained 60 results (10 inputs × 6 approaches). We then asked each participant to rate each of the 60 results on a scale from zero (worst) to five (best). Note that the human experts who helped create the 10 results did not participate in this experiment.

Figure 20 plots the average ratings for the design quality of the results obtained from the six approaches. This plot indicates that the participants rated our approach as comparable with expert designers and significantly superior over the four baselines.

8 CONCLUSION, LIMITATIONS, AND FUTURE WORKS

This paper presents the first computational pipeline to create the physically-assemble-able LEGO® figurine models from portrait photos. The created figurines consist of coherently-connected LEGO® bricks adorned with cartoon-styled surface decals, vividly resembling the characters in photos. Altogether, this paper makes the following notable contributions. (i) The presented framework, as a very first attempt to enable personal fabrication for average users, demonstrates the technical feasibility of easily creating high-quality personalized LEGO® figurine models from casual portraits, while keeping cost-effective production. (ii) We have developed a three-stage pipeline to reproduce both the prominent features and intricate details of the figures. This pipeline includes a CLIP-based module leveraging the CLIP space with rich semantic knowledge for model retrieval, a symmetric U-Nets architecture with the appearance adaptation strategy for creating high-quality decals, and a physical production process to facilitate the fabrication. (iii) We conducted comprehensive evaluations to demonstrate the effectiveness of our approach. This includes producing eight hundred figurines from various types of input portraits and evaluating them against expert-designed models using a comprehensive set of metrics. The metrics creatively leverage the prior knowledge in the pre-trained model and the descriptive capability of the very new GPT-4V large language model. Further, we compared our pipeline with several alternative approaches and obtained feedback from human designers. Both quantitative and visual results demonstrate the superior performance of our approach, highlighting its great capability and robustness of producing user-desired personalized LEGO® figurines.

Limitations. There are three main limitations to our method. First, our method generates a figurine whose front view resembles the input, neglecting the side and back views. As a result, it sometimes fails to capture elements visible from these perspectives. Second, the retrieved figurine may not always precisely match the hairstyle and garments in the input photo. This limitation becomes apparent with unique and uncommon accessories. A potential solution to this issue is to continually expand our dataset's size, enabling better representation and matching of a wider variety of styles and accessories. Finally, our method has limitations in generating extremely complex decals on garments. While we strive to achieve the best possible results under any condition, the resolution and physical constraints of the LEGO® bricks limit our ability to create highly intricate decals. Examples of such complexity include colorful feathers on ethnic minority costumes or traditional East Asian clothing adorned with numerous decorative patterns. Please refer to Part B of the supplementary for example failure cases.

Discussion and future works. Addressing the limitations above already suggests a more comprehensive LEGO® Figurine computational system. This would involve enabling multi-view decal generation, supporting a broader range of accessories and decorations, and accommodating high-resolution decals. Furthermore, the scope of our computational pipeline could be expanded beyond figurine design, to encompass more types of personalized objects that hold wide appeal, such as pet models or miniature houses.

Discussion on ethics. There are two potential ethical concerns: (i) the handling of personally identifiable information (PII) and (ii) the mitigation of bias stemming from data imbalance. For the first concern, we limit the display of portrait images with PII within the main paper to those of celebrities and participants hired by us. When showing the portrait photos of the participants in the supplementary material, we avoid PII by cropping or masking out the facial identities. For the potential bias, we filter and balance the dataset across various demographics, aiming to reduce biases.

ACKNOWLEDGMENTS

We thank all the anonymous reviewers for their comments and feedback. We also thank the LEGO® designers for their assistance in crafting our dataset and participating in our interviews. This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK 14201921).

REFERENCES

Patrick Baudisch and Stefanie Mueller. 2017. Personal Fabrication. *Foundations and Trends® in Human–Computer Interaction* 10, 3–4 (2017), 165–293. https://doi.org/10.1561/1100000055

Quentin Becker, Seiichi Suzuki, Yingying Ren, Davide Pellis, Francis Julian Panetta, and Mark Pauly. 2023. C-shells: Deployable Gridshells with Curved Beams. *ACM Transactions on Graphics* 42, 6 (2023), 1–17.

BrickLink. 2024. *Bricklink Color Guide.* https://www.bricklink.com/catalogColors.asp

Kaidi Cao, Jing Liao, and Lu Yuan. 2018. Carigans: Unpaired photo-to-caricature translation. *arXiv preprint arXiv:1811.00222* (2018).

Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Rui, Xuhui Jia, Ming-Wei Chang, and William W Cohen. 2023a. Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186* (2023).

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. 2023b. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481* (2023).

Blender Online Community. 2024. *Blender - a 3D modelling and rendering package.* Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

Noah Duncan, Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. 2017. Approximate dissections. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

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

Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022b. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–13.

Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. 2019. A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. *CVPR* (2019).

Rebecca A. H. Gower, Agnes E. Heydtmann, and Henrik G. Petersen. 1998. LEGO: Automated Model Construction. In *European Study Group with Industry.* 81–94.

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. 2023. Animate Anyone: Consistent and Controllable Image-to-Video Synthesis for Character Animation. *arXiv preprint arXiv:2311.17117* (2023).

Xuhui Jia, Yang Zhao, Kelvin CK Chan, Yandong Li, Han Zhang, Boqing Gong, Tingbo Hou, Huisheng Wang, and Yu-Chuan Su. 2023. Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. *arXiv preprint arXiv:2304.02642* (2023).

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *YOLO by Ultralytics*. https://github.com/ultralytics/ultralytics

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020. Training generative adversarial networks with limited data. *Advances in neural information processing systems* 33 (2020), 12104–12114.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.

Maria Korosteleva and Olga Sorkine-Hornung. 2023. GarmentCode: Programming Parametric Sewing Patterns. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–15.

Gihyun Kwon and Jong Chul Ye. 2023. Diffusion-based Image Translation using disentangled style and content representation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/pdf?id=Nayau9fwXU

Seung-Mok Lee, Jae Woo Kim, and Hyun Myung. 2018. Split-and-Merge-Based Genetic Algorithm (SM-GA) for LEGO Brick Sculpture Optimization. *IEEE Access* 6 (2018), 40429–40438.

Kyle Lennon, Katharina Fransen, Alexander O'Brien, Yumeng Cao, Yamin Beveridge, Matthewand Arefeen, Nikhil Singh, and Iddo Drori. 2021. Image2lego: Customized lego set generation from images. *arXiv preprint arXiv:2108.08477* (2021).

Dongxu Li, Junnan Li, and Steven CH Hoi. 2023. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *arXiv preprint arXiv:2305.14720* (2023).

Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. 2015a. Deep human parsing with active template regression. *IEEE transactions on pattern analysis and machine intelligence* 37, 12 (2015), 2402–2414.

Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. 2015b. Human parsing with contextualized convolutional neural network. In *Proceedings of the IEEE international conference on computer vision*. 1386–1394.

Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. 2015. Legolization: Optimizing LEGO Designs. 34, 6 (2015). Article no. 222.

Jun Mitani and Hiromasa Suzuki. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM transactions on graphics (TOG)* 23, 3 (2004), 259–263.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).

Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Pavel Petrovič. 2001. Solving LEGO Brick Layout Problem using Evolutionary Algorithms. In *Proc. NIK (Norsk Informatikkonferanse)*. 87–97.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. 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*. 22500–22510.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. 2022. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–10.

Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. 2023. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411* (2023).

Guoxian Song, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. 2021. AgileGAN: stylizing portraits by inversion-consistent transfer learning. *ACM Trans. Graph.* 40, 4, Article 117 (jul 2021), 13 pages. https://doi.org/10.1145/3450626.3459771

Peng Song, Xiaofei Wang, Xiao Tang, Chi-Wing Fu, Hongfei Xu, Ligang Liu, and Niloy J Mitra. 2017. Computational design of wind-up toys. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Enrique Soriano, Ramon Sastre, and Dionis Boixader. 2019. G-shells: Flat collapsible geodesic mechanisms for gridshells. In *Proceedings of IASS annual symposia*, Vol. 2019. International Association for Shell and Spatial Structures (IASS), 1–8.

Ben Stephenson. 2016. A Multi-Phase Search Approach to the LEGO Construction Problem. In *Proc. Symposium on Combinatorial Search (SoCS)*. 89–97.

Romain Testuz, Yuliy Schwartzburg, and Mark Pauly. 2013. Automatic Generation of Constructible Brick Sculptures. In *Eurographics (short paper)*. 81–84.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. 2022. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10748–10757.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1921–1930.

Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. 2022. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952* (2022).

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

David V. Winkler. 2005. Automated Brick Layout. In *Proc. BrickFest*. 145–166.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems* 34 (2021), 12077–12090.

Hao Xu, Ka-Hei Hui, Chi-Wing Fu, and Hao Zhang. 2021. Computational LEGO Technic Design. *ACM Trans. Graph.* 38, 6, Article 196 (aug 2021), 14 pages. https://doi.org/10.1145/3355089.3356504

Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. 2023. MagicAnimate: Temporally Consistent Human Image Animation using Diffusion Model. *arXiv preprint arXiv:2311.16498* (2023).

Zhijin Yang, Pengfei Xu, Hongbo Fu, and Hui Huang. 2021. WireRoom: Model-Guided Explorative Design of Abstract Wire Art. *ACM Transactions on Graphics* 40, 4 (July 2021), 128:1–128:13. https://doi.org/10.1145/3450626.3459796

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023c. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

Ran Zhang, Thomas Auzinger, and Bernd Bickel. 2021. Computational Design of Planar Multistable Compliant Structures. *ACM Transactions on Graphics* 40, 5 (Oct. 2021), 186:1–186:16. https://doi.org/10.1145/3453477

Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. 2023a. ProSpect: Prompt Spectrum for Attribute-Aware Personalization of Diffusion Models. 42, 6, Article 244 (dec 2023), 14 pages. https://doi.org/10.1145/3618342

Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. 2023b. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10146–10156.

Jie Zhou, Xuejin Chen, and Y Xu. 2019. Automatic generation of vivid LEGO architectural sculptures. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 31–42.

Mingjun Zhou, Jiahao Ge, Hao Xu, and Chi-Wing Fu. 2023. Computational Design of LEGO® Sketch Art. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–15.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.