CHARM: Control-point-based 3D Anime Hairstyle Auto-Regressive Modeling

YUZE HE*, Tsinghua University, China and Tencent AIPD, China
YANNING ZHOU†, Tencent AIPD, China
WANG ZHAO, Tsinghua University, China
JINGWEN YE, Tencent AIPD, China
YUSHI BAI, Tsinghua University, China
KAIWEN XIAO, Tencent AIPD, China
YONG-JIN LIU†, Tsinghua University, China
ZHONGQIAN SUN, Tencent AIPD, China
WEI YANG, Tencent AIPD, China

Fig. 1. Based on a novel parametric representation, CHARM provides a generative framework for 3D anime hairstyle modeling.

We present CHARM, a novel parametric representation and generative framework for anime hairstyle modeling. While traditional hair modeling methods focus on realistic hair using strand-based or volumetric representations, anime hairstyle exhibits highly stylized, piecewise-structured geometry that challenges existing techniques. Existing works often rely on dense mesh modeling or hand-crafted spline curves, making them inefficient for editing and unsuitable for scalable learning. CHARM introduces a compact, invertible control-point-based parameterization, where a sequence of control points represents each hair card, and each point is encoded with only five geometric parameters. This efficient and accurate representation supports both artist-friendly design and learning-based generation. Built upon this representation, CHARM introduces an autoregressive generative framework that effectively generates anime hairstyles from input images or point clouds. By interpreting anime hairstyles as a sequential "hair language", our autoregressive transformer captures both local geometry and global hairstyle topology, resulting in high-fidelity anime hairstyle creation. To facilitate both training and evaluation of anime hairstyle generation, we construct AnimeHair, a large-scale dataset of 37K high-quality anime hairstyles with separated hair cards and processed mesh data. Extensive experiments demonstrate state-of-the-art performance of CHARM in both reconstruction

*Work done during an internship at Tencent AIPD.
†Corresponding authors.

Authors' addresses: Yuze He, hyz22@mails.tsinghua.edu.cn, Tsinghua University, Beijing, China; Yanning Zhou, ynzhou0907@gmail.com, Tencent AIPD, Shenzhen, China; Wang Zhao, thuzhaowang@163.com, Tsinghua University, Beijing, China; Jingwen Ye, jingwenye@tencent.com, Tencent AIPD, Shenzhen, China; Yushi Bai, bys22@mails.tsinghua.edu.cn, Tsinghua University, Beijing, China; Kaiwen Xiao, scp173.cool@gmail.com, Tencent AIPD, Shenzhen, China; Yong-Jin Liu, liuyongjin@tsinghua.edu.cn, Tsinghua University, Beijing, China.

SA Conference Papers '25, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-2137-3/25/12
https://doi.org/10.1145/3757377.3763912

accuracy and generation quality, offering an expressive and scalable solution for anime hairstyle modeling. *Project page: https://hyzcluster.github.io/charm*

CCS Concepts: • **Computing methodologies** → **Mesh geometry models**; **Mesh models**; **Artificial intelligence**.

Additional Key Words and Phrases: Hair Modeling, Generative Models, Autoregressive Transformers

ACM Reference Format:
Yuze He, Yanning Zhou, Wang Zhao, Jingwen Ye, Yushi Bai, Kaiwen Xiao, Yong-Jin Liu, Zhongqian Sun, and Wei Yang. 2025. CHARM: Control-point-based 3D Anime Hairstyle Auto-Regressive Modeling. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25), December 15–18, 2025, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3757377.3763912

1 Introduction

Hair modeling is a long-standing research challenge in computer graphics. It plays a vital role in creating expressive and immersive digital avatars for animation, filming, and games. Among various types of digital hairstyles, anime hairstyle has emerged as a distinct and popular category, characterized by its stylized, exaggerated shapes and vibrant aesthetics.

So far, modeling anime hairstyles remains underexplored. Traditional methods primarily focus on realistic hair, where a rich body of work has explored explicit (e.g. strands [Chen et al. 1999; Koh and Huang 2001; Liang and Huang 2003; Yang et al. 2000], hair mesh [Yuksel et al. 2009]), implicit (e.g. implicit functions [Rosu et al. 2022; Sklyarova et al. 2023a]) and hybrid (e.g. strand-guided 3D Gaussians [Luo et al. 2024; Zakharov et al. 2024]) representations. In contrast, anime hairstyle is deliberately stylized with non-uniform strand thickness, variable hair density, non-fixed root positions and intra-hair structures, making it difficult to directly apply techniques developed for realistic hair.

Some existing character modeling methods [He et al. 2024b; Yan et al. 2024] create anime hairstyles directly as a monolithic polygonal mesh. Despite generality and simplicity, the lack of hair structural separation significantly complicates further editing and animation. In practice, artists often rely on parametric primitives and curves, such as Bézier splines, to define anime hair shapes. While intuitive and flexible for manual design or interaction, they present challenges for automatic reconstruction and generation, since fitting Bézier curves or learning to create those is highly non-trivial. Thus, a compact, structured parametric representation for anime hairstyle is essential, yet still absent, for enabling scalable and learnable modeling pipelines.

To address this gap, we propose a novel parametric representation for anime hairstyles (Fig. 1). Our design is inspired by the modeling format adopted in the popular industry platform VRoid Hub [VRoid 2022], which represents hair using repetitive geometric hair "units" (see Fig. 2). Rather than directly modeling these units with full-resolution meshes—as done in VRoid, which is highly redundant and suboptimal for learning—we introduce a compact control-point-based parameterization. Each unit is defined by a control point with only five parameters indicating 3D position, width, and thickness. A sequence of such control points then constitutes a single hair card. By further developing an optimization system that supports conversion between our proposed parametric model and conventional 3D mesh formats, we achieve a fully invertible, editable, and lightweight parameterization for anime-style hair geometry. This parametric representation is efficient, accurate, and also intuitive to manipulate and well-suited for integration with data-driven generative models.

Building on this representation, we further introduce an autoregressive generative model for creating anime hairstyles from casual inputs, such as point clouds or images. Leveraging the sequential nature of our representation, we conceptualize the anime hairstyle as a *hair language*, where each hair unit functions as a word, and each hair card as a sentence (see Fig. 4). This language-like innovative representation for anime hairstyle naturally captures both intra-hair and inter-hair structural relationships, and facilitates the generation of varying hair card counts and lengths. An auto-regressive transformer network is designed to effectively learn to generate this geometric hair language, under the condition of point clouds or images. Additionally, to train this model at scale, we construct AnimeHair, a large-scale dataset of 37K high-quality anime hairstyles derived from publicly available VRoid models. Each hairstyle is processed to isolate individual hair cards, and low-quality samples are filtered out. By training on this curated dataset, our model achieves high-quality and high-fidelity anime hairstyle generation, demonstrating the strong capability in both faithful reconstruction and creative hairstyle synthesis. Fig. 1 shows some examples of our generated anime hairstyles.

Our contributions are summarized as follows: 1) We introduce a novel, efficient, and accurate parametric representation tailored for anime hairstyle modeling, which is not only easy to edit or interact by human users, but also structured and scalable for learning deep models to reconstruct or generate with, 2) We formulate the anime hairstyle generation as a sequential language-like task via the proposed parametric representation, and design an autoregressive transformer network that effectively learns to synthesize hairstyles from diverse inputs, 3) A carefully curated dataset of 37K anime hairstyles with separated hair cards and high-resolution meshes, providing a solid foundation for both training and evaluations of anime hairstyle generation, 4) Extensive experiments demonstrate that our system can generate high-fidelity anime hairs, establishing new state-of-the-art for anime hairstyle generation.

2 Related Works

2.1 Hair Representation

2.1.1 Realistic Hair Representation. Modeling realistic virtual hair has been widely studied in computer graphics. Due to the high geometric complexity of hair and the variety of hairstyles, most parametric methods are based on controlling collections of hair strands, where, for simplification, a strand is parameterized by a sequence of connected 3D points. Various representations have been explored for hair strands, including parametric surfaces [Koh and Huang 2001; Liang and Huang 2003; Noble and Tang 2004], generalized cylinders [Chen et al. 1999; Kim and Neumann 2002; Xu and Yang 2001; Yang et al. 2000], and hair meshes [Yuksel et al. 2009]. For characterizing individual hair strand geometry, researchers have proposed generalized helicoids [Piuze et al. 2011], which represent both a single hair strand and its vicinity through three intuitive

curvature parameters and an elevation angle. Volumetric representation is also used for realistic-style hair modeling. [Saito et al. 2018] proposes to use volumetric orientation fields as an intermediate representation. However, the low resolution of voxels limits the fine detail reconstruction. CT2Hair [Shen et al. 2023] established a coarse-to-fine approach by creating density volumes of hair regions and estimating orientation fields to generate dense strands. Recently, hair modeling methods incorporating neural rendering methods [Wu et al. 2024]. GaussianHair [Luo et al. 2024] and Zakharov et al. [2024] design 3D Gaussians to represent the control points of the strands. The works [He et al. 2024a; Rosu et al. 2022; Sklyarova et al. 2023a,b] utilize neural scalp textures in UV-space as the base hair representation for explicit 3D geometry and appearance of strands, employing various generative models for hairstyle synthesis.

2.1.2 Anime Hair Representation. Anime hairstyle modeling aims to support virtual avatar creation and computer-aided cartoon production by non-photorealistic rendering. Unlike realistic style modeling, where hair strand thickness is usually uniform, anime hairstyle is **deliberately stylized with varying strand thicknesses** to create the iconic clustered appearance essential to the anime aesthetic. While most research has focused on photorealistic solutions, parametric modeling techniques specifically for anime hairstyles remain underexplored. Some attempt to use implicit surfaces [Sakai and Savchenko 2013], particle-based approaches [Shin et al. 2006], and cluster polygons [Mao et al. 2004]. However, these approaches have shown limited effectiveness as they result in incomplete hairstyles and still require manual modeling by artists. Another stream of works [Liu et al. 2013; McCann and Pollard 2009; Rivers et al. 2010; Yeh et al. 2014; Zhang et al. 2012] focused on 2.5D layered cartoon processing but lacked hair-specific design considerations or relied on 2D segmentation techniques that extract layers from reference views rather than generating complete 3D hairstyles. Our representation for anime hairstyles captures diverse 3D shapes while preserving anime aesthetics, offering a compact and invertible structure that is well-suited for generative learning frameworks.

2.2 Hair Modeling and Synthesis

Hair representation presents unique challenges in computer graphics, requiring specialized techniques for both reconstruction from existing data and generation of new hairstyles. While general head reconstruction methods [Gafni et al. 2021; Giebenhain et al. 2023; Hong et al. 2022; Zhang et al. 2020] incorporate hair, they typically produce only coarse outer shells with smooth geometry unsuitable for natural hair dynamics.

Specialized hair reconstruction approaches employ diverse techniques, from 3D orientation fields [Nam et al. 2019; Takimoto et al. 2024] to patch-based optimization [Wu et al. 2024] for high-fidelity exterior linemaps. Recent works have advanced along either designing more efficient representations [Luo et al. 2024] or leveraging data-driven approaches [Sklyarova et al. 2023a; Zakharov et al. 2024] that incorporate priors to enhance reconstruction quality in challenging scenarios. However, these methods are primarily designed for realistic hair reconstruction and still face limitations when reconstructing complex occluded regions.

3D generation has witnessed remarkable progress, ranging from general object synthesis [Petrov et al. 2024; Zhang et al. 2024], avatar creation [He et al. 2024b; Wang et al. 2024] to specialized realistic hair generation [He et al. 2024a; Long et al. 2025; Sklyarova et al. 2023b; Zhou et al. 2023]. For realistic hair generation, NeuralStrands [Rosu et al. 2022] and GroomGen [Zhou et al. 2023] employ modulator-synthesizer architectures and GAN-based generators respectively. Recently, HAAR [Sklyarova et al. 2023b] encodes hairstyle in UV feature space, and then leverage the latent diffusion model [Podell et al. 2023] for strand generation. TANGLED [Long et al. 2025] developed a multi-view lineart conditional diffusion approach followed by parametric braid inpainting. While it generates high-fidelity hair strands across diverse input styles, the resulting hair strands remain limited to realistic representations, which are unsuitable for anime character model requirements.

A key distinction exists between realistic and anime-style hair requirements. Anime hairstyle features deliberately stylized strands with non-uniform thickness, variable hair density, and irregular root positioning. These properties make fixed-resolution UV-space mapping unsuitable. StdGEN [He et al. 2024b] attempts to generate decomposed anime hair meshes along with cloth and body, but it suffers from over-smoothing. Auto-regressive transformers, which have excelled in language modeling [Brown et al. 2020; Radford et al. 2019] and 3D mesh generation [Chen et al. 2024a; Siddiqui et al. 2024], offer promising solutions for generating variable-length sequences. However, efficiently modeling anime hairstyle within an AR framework remains non-trivial due to the stylistic complexity and geometric variability inherent in such representations. Our anime hairstyle representation is both compact and invertible, enabling efficient learning in auto-regressive generation.

3 Method

Our proposed CHARM is the first comprehensive framework dedicated to 3D anime hairstyle generation. In Sec. 3.1, we formally define 3D anime hairstyle and describe our dataset construction process. Sec. 3.2 explores our novel hair parameterization approach, while Sec. 3.3 elaborates on the hairstyle transformer architecture and sequence formulation for auto-regressive generation. Finally, Sec. 3.4 outlines the training and inference mechanisms.

3.1 Definition and Dataset for 3D Anime Hairstyles

Anime hairstyles are widely utilized on non-realistic characters in video games, virtual reality, and other digital environments. They differ fundamentally from realistic hair representations. Realistic hair is typically represented as strands, where each individual hair is treated as a strand with positional information but lacking 3D structural details. Additionally, realistic hair models generally maintain a fixed number of strands with predetermined root positions.

In contrast, anime hairstyle consists of multiple independent 3D meshes, each representing a single hair card. These individual hair cards are composed of several small, repeating 3D units connected end-to-end, with variable root positions across the scalp. As illustrated in Fig. 2, the left side displays a complete anime hairstyle, with a highlighted hair card enlarged to reveal its structure. This individual hair consists of multiple quadrilateral pyramids

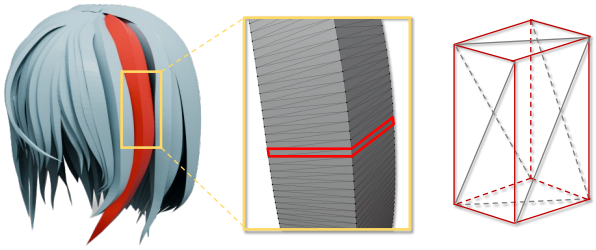

Fig. 2. Illustration of the Anime hairstyle structure. Left: A complete hairstyle sample. Middle: Repeating units in the highlighted hair card. Right: Mesh connection pattern of a single unit.

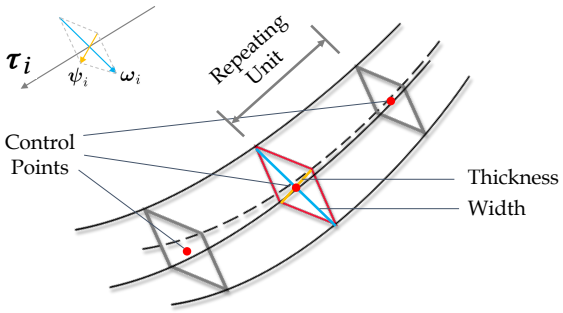

Fig. 3. Illustration of our hair parameterization. Within a single hair card, each bottom face of a repeating unit is defined by the 3D position of the centered control point, width, and thickness.

connected sequentially, featuring diamond-shaped top and bottom faces. Within each repeating unit, the mesh edge connections follow consistent and regular patterns, while maintaining connectivity between adjacent units despite variations in size. In the current industry production pipelines and available data, these repeating units are predominantly of two types: diamond-shaped quadrilateral pyramids and triangular pyramids. Each hair card typically comprises 20-60 repeating units, with a complete hairstyle consisting of 25-130 individual hair cards.

Currently there are no existing anime hairstyle datasets available for data-driven learning. To facilitate the procedural generation and manipulation of anime hairstyles, we have assembled AnimeHair, a large-scale dataset containing 37K distinct anime hairstyles. These samples were derived from publicly available 3D character models on VRoid-Hub. The extraction process involved isolating the hair components and implementing filters to eliminate corner cases or low-quality samples that deviated from standard hair construction principles (such as non-watertight meshes, discontinuous meshes, or arbitrary shapes). Further details of dataset construction and processing are provided in the supplementary material (Sec. B).

3.2 Invertible Hair Parameterization

To enable autoregressive generation of 3D anime-style hair meshes, it is necessary to compress the original geometry before generation. Directly generating high-resolution meshes is infeasible due to computational constraints: typical hairstyles contain 7K-50K faces, far exceeding the capacity of current autoregressive 3D mesh generation methods, making a compact representation essential.

Given the structured nature of anime hairstyles, a natural approach is to parameterize the repeating units. Upon inspecting template data from existing game assets, we observed that most of the hair cards are constructed using a Bézier surface and parameterized through two attributes: width and thickness. Each hair card is constrained to lie on a Bézier surface defined by a set of control points. The segment between any two consecutive control points can be treated as a repeating unit mentioned above. At each control point, the width and thickness value correspond to either the diagonals of a diamond or the base and height of an isosceles triangle, representing the hair's cross-sectional dimensions. The thickness direction aligns with the normal of the Bézier surface at that point, while the width direction is computed as the cross product of the surface normal and the tangent vector along the control curve.

A key challenge is determining a minimal yet expressive parameterization of each unit. Directly storing width and thickness directions per cross-section is intuitive but redundant and less learnable. Reconstructing the original Bézier surface from each hair card mesh is computationally expensive, unstable, and ill-posed due to the unknown and variable number of repeating units. This motivates the need for a compact and invertible parameterization scheme.

To address these issues, we design a fully invertible control-point-based parameterization tailored for anime hairstyles. Each hair card is represented by N control points, and each control point is characterized by only 5 float values: 3D position (x, y, z), width w, and thickness t (shown in Fig. 3). The tangent vector $\boldsymbol{\tau}_i$ at each control point i is estimated using a fourth-order accurate five-point central difference method. Given a control point i, its tangent is approximated using the coefficient vector $[-1/12, 2/3, 0, -2/3, 1/12]$ applied to its five nearest neighbors.

Estimating reliable width vectors $\boldsymbol{\omega}_i$ for each control point i by only the control point sequence poses an additional challenge. A common approach is to apply Principal Component Analysis (PCA) to the neighboring points and use the resulting normal vector as the width direction. However, this method performs poorly in regions where the hair are nearly straight, often producing unstable and noisy results. To address this, we reformulate the problem as an optimization task with two key objectives:

(1) The projection distances of neighboring points onto the normal plane should be minimized.
(2) Normals at adjacent points should vary smoothly.

Simply calculating initial normal vectors and propagating with Bishop frames [Bergou et al. 2008] suffers from precision issues due to the fewer and varied-sized repeating units, as well as the higher discretization characteristics of anime hairstyles. While gradient descent can be used to solve this problem and yields reasonable results, it is computationally expensive and lacks determinism, making it unsuitable for a template-driven generation framework. To improve efficiency and robustness, we reformulate the problem as a least-squares optimization, removing the unit-length constraint on normals and initializing the first normal (e.g., via PCA) to prevent

degenerate solutions. This formulation allows us to solve for all normals in a single linear system. For the coordinates p_i of each control point i, we compute the difference vectors with its neighbors:

$$d_{i,\text{prev}} = p_i - p_{i-1}, \quad d_{i,\text{next}} = p_i - p_{i+1} \tag{1}$$

We then form the data matrix:

$$D_i = d_{i,\text{prev}} d_{i,\text{prev}}^T + d_{i,\text{next}} d_{i,\text{next}}^T \tag{2}$$

The optimization objective consists of two terms. The first encourages the estimated normal $\mathbf{n}_i$ to be orthogonal to the local geometry encoded by D_i, minimizing the projection of the surrounding structure onto the normal direction. The second term imposes smoothness by penalizing differences between adjacent normals. The complete objective is given by:

$$\sum_i \left(\mathbf{n}_i^T D_i \mathbf{n}_i \right) + \lambda \sum_i \|\mathbf{n}_i - \mathbf{n}_{i+1}\|^2 \tag{3}$$

This formulation leads to a sparse linear system of the form $A\mathbf{n} = 0$, where the matrix A is block tridiagonal. Where each diagonal block is defined as $A[i][i] = D_i + 2\lambda I$, and off-diagonal blocks that connect neighboring normals are $A[i][i+1] = A[i+1][i] = -\lambda I$. To avoid the trivial zero solution, we fix the normal $\mathbf{n}_1$ at the first control point (e.g., via PCA initialization), effectively anchoring the solution. The system is then solved using standard least-squares methods to recover the normal vector $\mathbf{n}_i$ at each control point.

With the system fully defined and the constraint applied, we solve for the entire set of normal vectors using standard least-squares solvers, obtaining a globally consistent and smooth field of normals along the single hair card. The direction vectors of width $\boldsymbol{\omega}_i$ and thickness $\boldsymbol{\psi}_i$ at each control point i are then defined as:

$$\boldsymbol{\omega}_i = \mathbf{n}_i, \quad \boldsymbol{\psi}_i = \mathbf{n}_i \times \boldsymbol{\tau}_i \tag{4}$$

With the width and thickness direction vectors defined, we can now reconstruct the vertex positions of the bottom face of each repeated hair unit. Taking the diamond-shaped unit as an example, the four corner vertices corresponding to control point p_i are given by $[p_i + \frac{1}{2}\boldsymbol{\omega}_i, p_i - \frac{1}{2}\boldsymbol{\omega}_i, p_i + \frac{1}{2}\boldsymbol{\psi}_i, p_i - \frac{1}{2}\boldsymbol{\psi}_i]$. Using these vertices, we can assemble the geometry of each repeating unit and sequentially reconstruct the complete 3D mesh of a single hair card.

At this point, we have established a forward pipeline that reconstructs the full 3D mesh of a single hair card given only N control points, where each point stores five parameters: its 3D position $p = (x, y, z)$, width w, and thickness t. The inverse process is comparatively straightforward. For each repeating unit, we extract the centroid of its bottom face as the corresponding control point position. The diagonals (in the case of a diamond) or the base and height (in the case of an isosceles triangle) are directly mapped to the width and thickness values. Through this design, we achieve a fully invertible and compact parameterization of anime-style hair geometry, preserving anime styling at <2% of the original 3D mesh token cost (Fig. 9).

3.3 Auto-Regressive Hairstyle Transformer

We now design an efficient and effective network structure with a flexible sequence formulation to implement auto-regressive anime hairstyle generation.

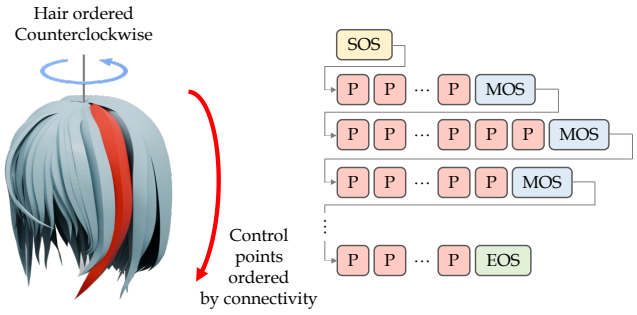

Fig. 4. Hair sequence construction. Left: Ordering strategy showing counterclockwise arrangement of hair cards around the head and root-to-tip connectivity within each hair. Right: Sequence formulation where each row corresponds to a single hair card terminated by a MOS token.

3.3.1 Network Structure. Our hairstyle transformer takes a condition C (in our case, a point cloud) as input and outputs a sequence of control points that constitute a hairstyle. The architecture comprises three primary components: a control point encoder $\mathcal{E}$, a decoder-only transformer $\mathcal{T}$, and a cascaded decoder $\mathcal{D}$. We adopt a piecewise discretization scheme to quantize each control point i's attributes, including position $p_i = (x_i, y_i, z_i)$, width w_i, and thickness t_i (detail in supplementary Sec. C), and treat them as discrete input tokens. These tokens are processed through learnable embeddings $\mathbf{e}_p$, $\mathbf{e}_w$, and $\mathbf{e}_t$, then fed into the linear control point encoder $\mathcal{E}$ to generate control point tokens h_i:

$$h_i = \mathcal{E}([\mathbf{e}_p(p_i); \mathbf{e}_w(w_i); \mathbf{e}_t(t_i)]) \tag{5}$$

The decoder-only transformer $\mathcal{T}$ performs next-token prediction at the control point level. Given input condition C and all previously predicted tokens $h_1, \ldots, h_{i-1}$, it generates the feature representation $\mathbf{f}_i$ for the next control point:

$$\mathbf{f}_i = \mathcal{T}(C; h_1, \ldots, h_{i-1}) \tag{6}$$

Subsequently, we decode specific attributes from this feature representation. Following previous generation works [Paschalidou et al. 2021; Ritchie et al. 2019; Wang et al. 2021; Ye et al. 2025], we employ a cascaded decoding structure that explicitly models the dependencies among control point attributes:

$$\hat{p}_i = \mathcal{D}_p(\mathbf{f}_i), \tag{7}$$

$$\hat{w}_i = \mathcal{D}_w(\mathbf{f}_i, \mathbf{e}_p(\hat{p}_i)), \tag{8}$$

$$\hat{t}_i = \mathcal{D}_t(\mathbf{f}_i, \mathbf{e}_p(\hat{p}_i), \mathbf{e}_w(\hat{w}_i)) \tag{9}$$

Where $\mathcal{D}_p$, $\mathcal{D}_w$, and $\mathcal{D}_t$ represent the position decoder, width decoder, and thickness decoder, respectively. Each decoder operates on the concatenated input of $\mathbf{f}_i$ and the embedded representations of past decoded attributes, outputting probability logits.

3.3.2 Sequence Formulation. Given that anime hairstyles comprise a variable number of individual hair cards, each with different control point counts and root positions, we have designed a flexible sequence formulation to interpret anime hairstyles as a sequential "hair language", enabling autoregressive generation of arbitrary hairstyles (shown in Fig. 4).

At the hairstyle level (the "sentence" level of our "hair language"), we first define a generation order starting from the back of the head. Each individual hair card is ordered based on its average x,z coordinates (with the y-axis pointing upward), arranged counterclockwise when viewed from above. This ordering approach effectively captures dependencies between adjacent hair cards and facilitates continuous, stable generation, as it ensures spatial continuity between consecutive hair cards in the generation sequence.

For the internal structure of each hair card (the "word" level within a "sentence"), rather than ordering control points strictly by position, we arrange them sequentially from root to tip according to their connectivity relationships. This better preserves continuity and respects the physical characteristics of hair. Considering the natural properties of hair, we define the root of a single hair card as the endpoint that lies closer to the top part of the y-axis (vertical axis) passing through the hairstyle's center.

After defining the ordering system, we concatenate all control points into a final sequence. Similar to other autoregressive generation methods, we prepend input condition tokens followed by a start token <SOS> at the beginning of the sequence and append an end token <EOS> at the end. Notably, we introduce a special middle token <MOS> between individual hair cards, which provides the model with explicit signals to transition between different structural components of the hairstyle. We implement two additional decoders, $\mathcal{D}_{mos}$ and $\mathcal{D}_{eos}$, which operate on the feature representation $\mathbf{f}_i$ to determine when to terminate a hair card and the entire hairstyle, respectively. This sequence formulation enables our model to generate hairstyles with arbitrary numbers of hair cards, each with variable lengths and control point configurations.

3.4 Training and Inference

3.4.1 Training Objective. We use the surface point cloud as the input condition C and employ Michelangelo [Zhao et al. 2023] encoder to transform it into a fixed-length token sequence. This sequence is prepended with the <SOS> token. The hairstyle transformer is trained using a next-step prediction objective with the following loss function: $\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{eos} + \mathcal{L}_{mos}$, Where $\mathcal{L}_{ce}$ denotes the cross-entropy loss used to supervise the discrete control point attributes p_i, w_i, t_i. The terms $\mathcal{L}_{eos}$, $\mathcal{L}_{mos}$ refer to binary cross-entropy losses applied to $\mathcal{D}_{eos}(\mathbf{f}_i)$ and $\mathcal{D}_{mos}(\mathbf{f}_i)$, which guide the predictions for hairstyle-level and hair-card-level termination, respectively.

3.4.2 Inference. Our inference begins with the input condition and <SOS> token, from which we auto-regressively generate control point features $\{\mathbf{f}_i\}_{i=1}^{n}$ and subsequently decode them into their corresponding attributes. When the MOS decoder determines that the current hair card should terminate, the current token is replaced with an MOS token to signal the end of that hair. Additionally, we implement a structural coherence constraint: if a generated control point is the sixth or subsequent point in a hair card and its position deviates significantly from the extrapolated position predicted by a cubic spline fitted to preceding points (we use 0.03 as the threshold), we replace it with an MOS token. This constraint prevents unrealistic geometric discontinuities within individual hair cards.

The generation process continues iteratively until the EOS judgment criterion is satisfied. Specifically, we allow the entire hairstyle to terminate when both the number of generated individual hair cards exceeds 10 and the feature representation $\mathbf{f}_i$ is classified as an EOS token by the decoder $\mathcal{D}_{eos}$, which prevents the early stop of the hairstyle generation.

4 Experiments

4.1 Experimental Setup

4.1.1 Baselines. We randomly selected 100 hairstyles from the AnimeHair dataset as our test set, ensuring no overlap with the training set. Given the absence of existing methods specifically designed for the shape-conditioned generation of anime-style hair meshes, we benchmark our approach against state-of-the-art shape-conditioned 3D mesh generation methods, including MeshAnything [Chen et al. 2024a], MeshAnythingV2 [Chen et al. 2024b], BPT [Weng et al. 2024] and DeepMesh [Zhao et al. 2025].

4.1.2 Metrics. We uniformly sampled point clouds on the surfaces of predicted anime-style hairstyle mesh predictions and compared them against ground-truth point clouds derived from original reference hairstyle meshes. We employed four distinct quantitative metrics: Chamfer Distance (CD), Earth Mover's Distance (EMD), Hausdorff Distance, and Voxel Intersection over Union (Voxel-IoU). The Voxel-IoU metric involved a preprocessing step where both predicted and ground-truth point clouds were voxelized at a uniform resolution of 16^3, and then computing their intersection over union. To ensure fair comparison given the different normalization approaches used by various methods, we standardize the evaluation process by normalizing the point clouds according to each method's default normalization procedure, and then unnormalizing the generated results using the same method-specific parameters to calculate the metrics. However, we recognize that geometric metrics alone cannot fully capture cases where a generated hairstyle might cover the surface adequately but exhibit poor morphological characteristics. To further reflect perceptual alignment, we apply fixed lighting conditions to all hairstyles and render each case from eight horizontally equidistant viewpoints. These renderings are then encoded using CLIP [Radford et al. 2021], and the average cosine similarity is calculated to quantify perceptual quality.

4.2 Results and Comparisons

Our quantitative comparison results are presented in Tab. 1, demonstrating that our method outperforms all other approaches across the evaluated metrics. Most baseline methods struggle to achieve accurate geometric reconstruction of anime-style hairstyles; even MeshAnythingV2, which exhibits more stable performance than other baselines, remains inferior to our approach.

Table 1. Geometric comparison on the AnimeHair test set.

Method	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
MeshAnything	0.0179	0.0190	0.0584	0.7014
MeshAnythingV2	0.0118	0.0132	0.0510	0.7503
BPT	0.0270	0.0317	0.0767	0.6328
DeepMesh	0.0172	0.0198	0.0613	0.7313
Ours	**0.0117**	**0.0128**	**0.0497**	**0.7566**

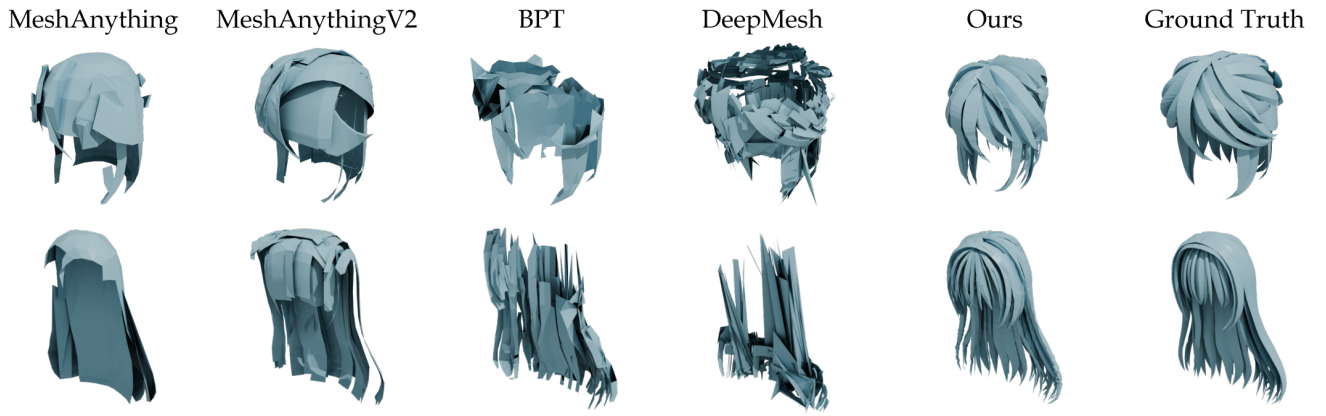

Fig. 5. Qualitative comparisons with other shape-conditioned 3D mesh generation methods.

The perceptual alignment comparison in Tab. 2 reveals a more pronounced advantage for our method, reflecting that other shape-conditioned mesh generation approaches fail to capture the distinctive volumetric structure and stylistic features characteristic of anime hairstyles. These methods typically model hairstyle as a continuous surface rather than as discrete, independently articulated hairs with specific geometric properties.

Table 2. Perceptual alignment comparison on the AnimeHair test set.

Method	CLIP Similarity ↑	Ranking ↓
MeshAnything [Chen et al. 2024a]	0.8435	4
MeshAnythingV2 [Chen et al. 2024b]	0.8618	2
BPT [Weng et al. 2024]	0.8404	5
DeepMesh [Zhao et al. 2025]	0.8576	3
Ours	**0.9258**	**1**

This distinction becomes evident in the qualitative comparisons illustrated in Fig. 5 as well. Other mesh generation methods produce results characterized by incompleteness, lower resolution, and a tendency to clump individual hair cards together. In contrast, our method maintains complete decoupling between different hair cards, resulting in superior human perceptual quality and physical plausibility. Each hair card maintains its distinct shape, flow, and volume while collectively forming a coherent hairstyle that preserves the aesthetic qualities demanded by anime-style representations.

4.3 Ablation Study

To validate the effectiveness of our parameterization method and sequence formulation, we conducted a comprehensive ablation study. For parameterization, we examined whether our current parameter set for each control point (xyz coordinates, width, thickness) is optimal. We introduced an alternative parameterization scheme called *Extended Vector Parameterization*, which additionally incorporates directional vectors for width and thickness as prediction parameters, resulting in 11 float values per control point. We also tested a more direct approach called *Explicit Vertex Parameterization*, where we generate the hairstyle mesh directly by predicting all the vertices.

Table 3. Ablation study on sequence ordering and parameterization.

Method	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
Sequence Ordering Strategy				
X-Axis Sorting	0.0127	0.0140	0.0531	0.7274
Y-Axis Sorting	0.0149	0.0154	0.0521	0.6832
Z-Axis Sorting	0.0122	0.0133	0.0527	0.7402
Counterclockwise (Ours)	**0.0117**	**0.0128**	**0.0497**	**0.7566**
Parameterization Method				
Extended Vector Param.	0.0129	0.0140	0.0526	0.7411
Explicit Vertex Param.	0.0225	0.0233	0.0857	0.6235
Ours	**0.0117**	**0.0128**	**0.0497**	**0.7566**

Since adopting original mesh tokenization would exceed current token limitations, we simplified the task by pre-specifying vertex ordering and face connectivity patterns, then reconstructing the original mesh and calculating metrics.

As shown in Tab. 3, the *Extended Vector Parameterization* exhibited some degradation in overall geometric accuracy despite providing more detailed control, highlighting the elegant balance between expressiveness and learnability in our proposed parameterization. The *Explicit Vertex Parameterization* demonstrated an even more significant performance decline, underscoring the necessity of transforming mesh prediction into a control-point-based hair language parameterization (as shown in Fig. 7).

For sequence formulation, we compared our counterclockwise hair ordering strategy against common alternatives that sort hair cards based on their average positions along coordinate axes (x-axis, y-axis, z-axis). Tab. 3 reveals that y-axis sorting (vertical axis) performed worst, while x-axis and z-axis sorting showed comparable results, though still inferior to our approach. This performance difference arises because counterclockwise hair ordering maximally captures inter-hair patterns and minimizes unpredictable transitions during generation. While x-axis and z-axis sorting maintain some pattern recognition, they frequently introduce spatial discontinuities (alternating between opposite sides of the head), whereas y-axis sorting reveals the least consistent patterns among hair cards.

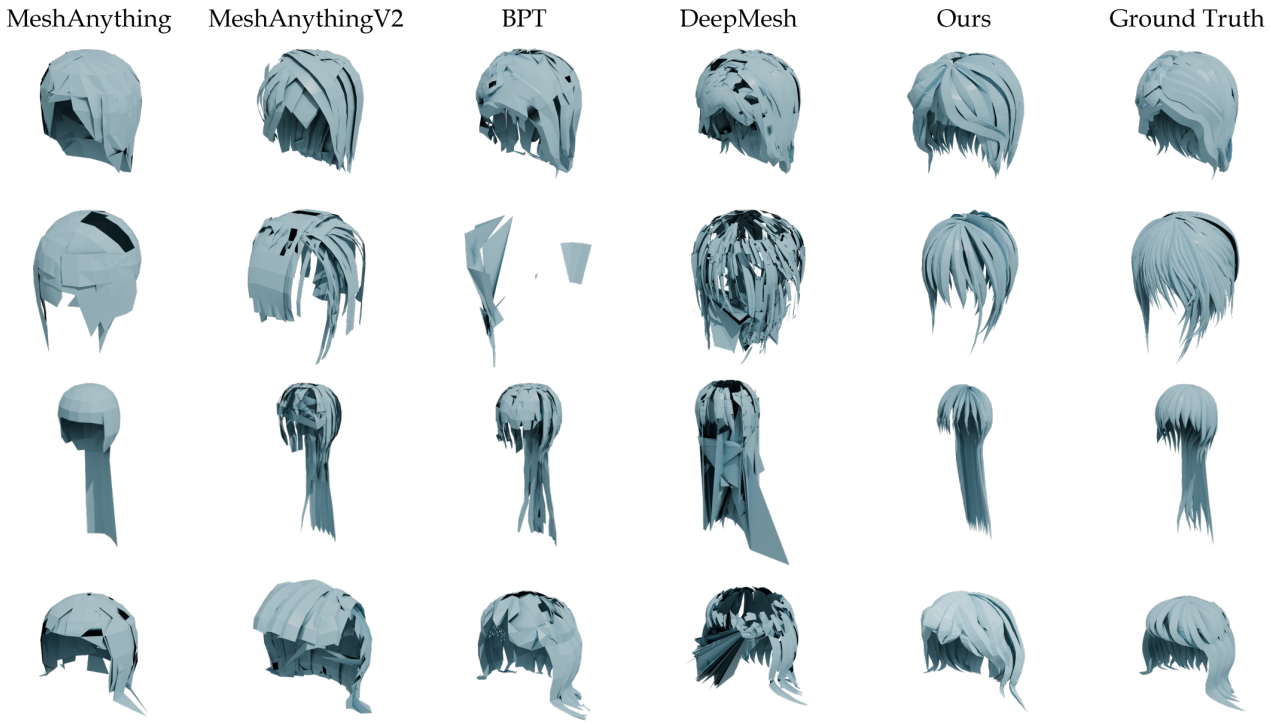

Fig. 6. More qualitative comparisons with other shape-conditioned 3D mesh generation methods.

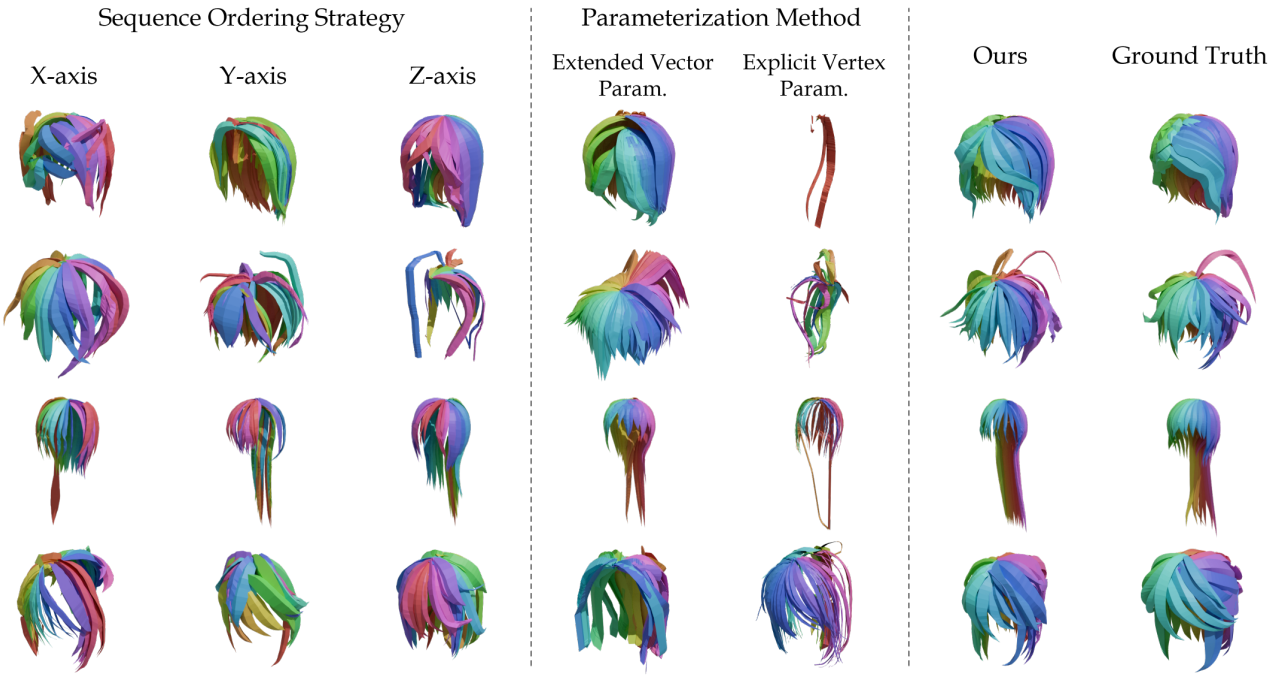

Fig. 7. Qualitative comparisons of ablation study. Colors indicate different hair strands.

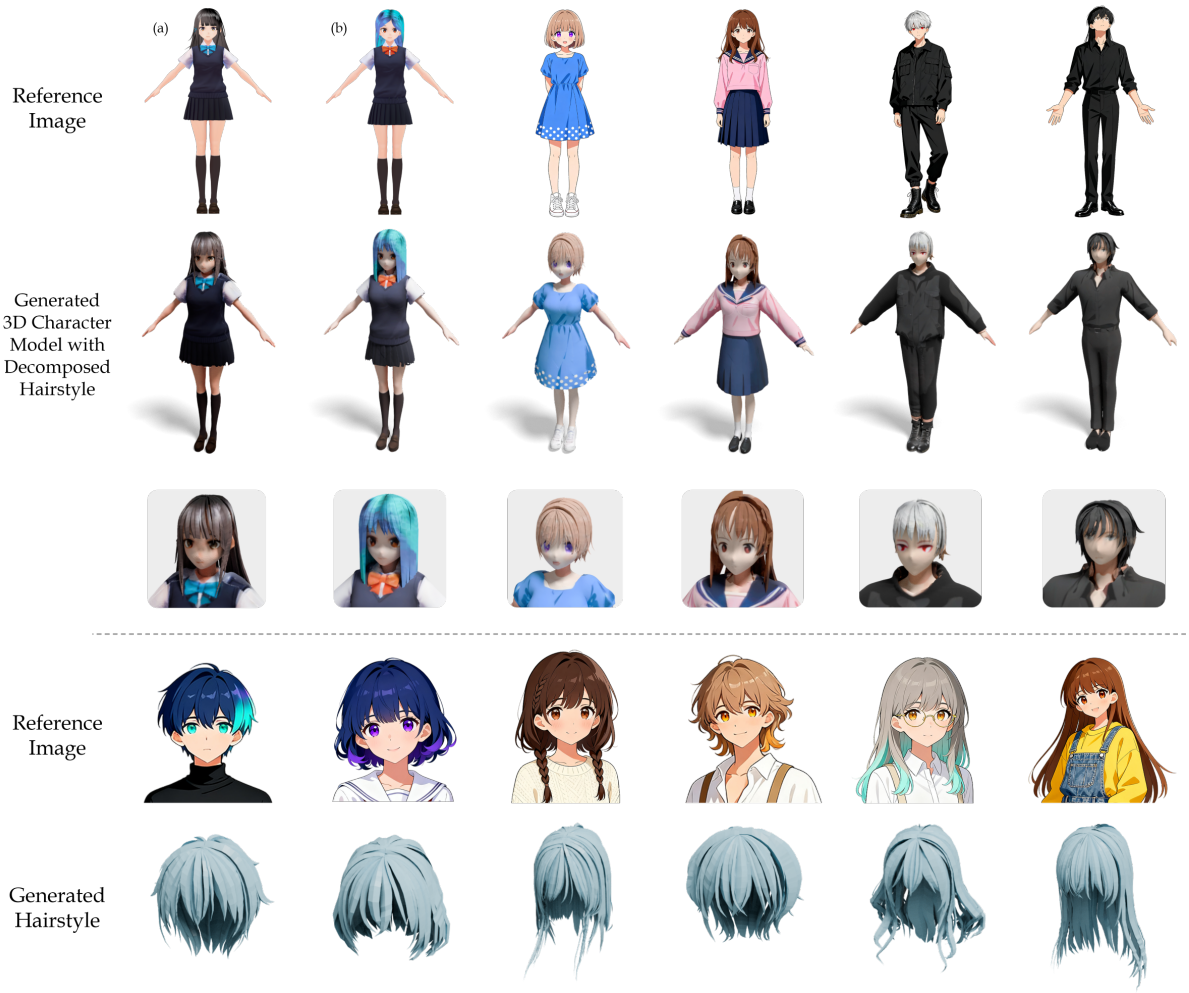

Fig. 8. Our framework supports flexible 3D anime hairstyle generation through various input conditions. The images in (a) and (b) are adapted from publicly available models on VRoid-Hub [VRoid 2022], while the remaining images are AI-generated.

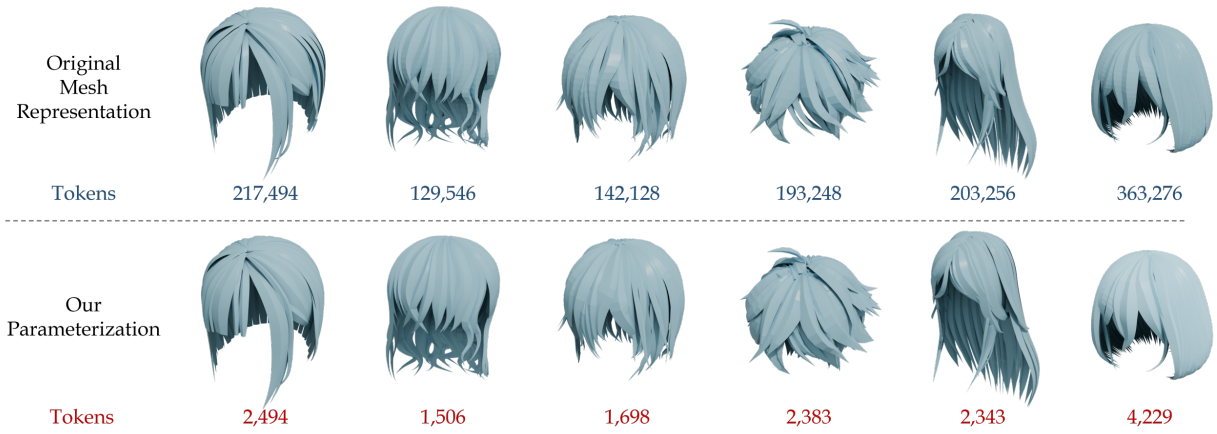

Fig. 9. Our compact and invertible parameterization achieves over 98% token compression compared to the original mesh.

Table 4. Ablation study on sequence ordering and parameterization approaches (hair card level). The best metric is highlighted in bold, and the second-best metric is underlined.

Method	CD ↓	EMD ↓	Hausdorff ↓	Voxel-IoU ↑
Sequence Ordering Strategy				
X-Axis Sorting	<u>0.0111</u>	<u>0.0092</u>	0.0289	0.6338
Y-Axis Sorting	0.0136	0.0110	0.0320	0.6021
Z-Axis Sorting	0.0114	0.0092	0.0306	0.6308
Counterclockwise (Ours)	**0.0105**	**0.0085**	<u>0.0289</u>	<u>0.6408</u>
Parameterization Method				
Extended Vector Param.	0.0117	0.0094	**0.0283**	**0.6576**
Explicit Vertex Param.	0.0149	0.0115	0.0317	0.6141
Ours	**0.0105**	**0.0085**	<u>0.0289</u>	<u>0.6408</u>

Given that our method enables decoupled hair representation, we can compute geometric metrics at the hair card level to better reflect the effectiveness of our approach. For each hair card in the prediction, we sample point clouds and identify the hair in the ground truth with the minimum Chamfer Distance; similarly, for each hair card in the ground truth, we find its closest counterpart in the prediction. We then calculate the average of all metrics across these matched pairs and aggregate them over the entire hairstyle, as shown in Tab. 4. The results indicate that our scheme also exhibits relatively strong coherence and accuracy at hair-card level, validating our approach's ability to generate not only globally plausible hairstyles, but also individually well-formed hairs.

4.4 Applications

Our framework supports flexible 3D anime hairstyle generation through various input conditions. As illustrated in Fig. 8, our approach can be integrated with StdGEN [He et al. 2024b], transforming its non-decomposable hair outputs into ready-to-use hairstyles that can be assembled with the original results to create 3D character models more suitable for practical applications. Additionally, our framework supports image-conditioned hairstyle generation by re-training with DINO image features via cross-attention, allowing users to specify desired hairstyle characteristics through reference images. Details are in the supplementary material (Sec. C).

4.5 Comparison with Realistic Hair Methods

To demonstrate the fundamental differences between realistic [Rosu et al. 2025; Zheng et al. 2023] and anime hairstyle generation, we conducted a qualitative comparison between image-conditioned methods for realistic hair generation (specifically DiffLocks [Rosu et al. 2025]) and our proposed anime-focused approach. The images used for this comparison are generated by AI and sourced from the AI-generated content website Doubao [Doubao 2025], to avoid the copyright and privacy complexities of using real photos. The prompts are available upon request. As illustrated in Fig. 10, the comparison reveals domain-specific limitations when applying methods across different hairstyle domains. Anime hairstyles typically lack realistic elements such as permed hairs, upward-styled curls, and long side-swept hair arrangements. This data distribution gap results in noticeable discrepancies between our predicted 3D hair geometry and the input realistic hair images, as demonstrated in the left and middle cases of Fig. 10, while DiffLocks shows better performance. Conversely, while realistic hair generation methods struggle to accurately predict bangs that fall across the forehead area (even for realistic style input), our anime-specialized approach successfully handles such hairstyles, as shown in the rightmost case. This contrast further underscores the fundamental data distribution differences between realistic and anime hair domains.

Fig. 10. Qualitative comparison between DiffLocks (realistic hair method) and our approach on image-conditioned 3D hairstyle generation. The images used for this comparison are AI-generated.

5 Conclusion

We present CHARM, the first framework for 3D anime hairstyle generation, featuring a compact and invertible control point-based parameterization that efficiently represents complex strand geometries. Our specialized hairstyle transformer enables autoregressive generation of detailed hairstyles with individually articulated hair cards. Trained on our curated AnimeHair dataset of 37K hairstyles, experiments show significant improvements in geometric accuracy and perceptual quality over existing methods.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (62461160309) and NSFC-RGC Joint Research Scheme (N_HKU705/24).

References

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM SIGGRAPH 2008 papers*. 1–12.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

Lieu-Hen Chen, Santi Saeyor, Hiroshi Dohi, and Mitsuru Ishizuka. 1999. A system of 3D hair style synthesis based on the wisp model. *The Visual Computer* 15, 4 (Jul 1999), 159–170. doi:10.1007/s003710050169

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. 2024a. MeshAnything: Artist-Created Mesh Generation with Autoregressive Transformers. *arXiv preprint arXiv:2406.10163* (2024).

Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. 2024b. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555* (2024).

Bytedance Doubao. 2025. Doubao. https://doubao.com/

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8649–8658.

Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. 2023. Learning neural parametric head models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21003–21012.

Chengan He, Xin Sun, Zhixin Shu, Fujun Luan, Sören Pirk, Jorge Alejandro Amador Herrera, Dominik L Michels, Tuanfeng Y Wang, Meng Zhang, Holly Rushmeier, et al. 2024a. Perm: A parametric representation for multi-style 3d hair modeling. *arXiv preprint arXiv:2407.19451* (2024).

Yuze He, Yanning Zhou, Wang Zhao, Zhongkai Wu, Kaiwen Xiao, Wei Yang, Yong-Jin Liu, and Xiao Han. 2024b. StdGEN: Semantic-Decomposed 3D Character Generation from Single Images. *arXiv preprint arXiv:2411.05738* (2024).

Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. 2022. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20374–20384.

Tae-Yong Kim and Ulrich Neumann. 2002. Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics* (Jul 2002), 620–629. doi:10.1145/566654.566627

Chuan Koon Koh and Zhiyong Huang. 2001. A simple physics model to animate human hair modeled in 2D strips in real time. In *Computer Animation and Simulation 2001: Proceedings of the Eurographics Workshop in Manchester, UK, September 2–3, 2001*. Springer, 127–138.

Wenqi Liang and Zhiyong Huang. 2003. An Enhanced Framework for Real-Time Hair Animation. In *PG*. 467–471. https://doi.org/10.1109/PCCGA.2003.1238296

Xueting Liu, Xiangyu Mao, Xuan Yang, Linling Zhang, and Tien-Tsin Wong. 2013. Stereoscopizing cel animations. (Nov 2013).

Pengyu Long, Zijun Zhao, Min Ouyang, Qingcheng Zhao, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. 2025. TANGLED: Generating 3D Hair Strands from Images with Arbitrary Styles and Viewpoints. *arXiv preprint arXiv:2502.06392* (2025).

Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. Gaussianhair: Hair modeling and rendering with light-aware gaussians. *arXiv preprint arXiv:2402.10483* (2024).

Xiaoyang Mao, Hiroki Kato, Atsumi Imamiya, and Ken Anjyo. 2004. Sketch interface based expressive hairstyle modelling and rendering. (Jun 2004).

James McCann and Nancy Pollard. 2009. Local layering. *ACM Transactions on Graphics* 28, 3 (Jul 2009), 1–7. doi:10.1145/1531326.1531390

Giljoo Nam, Chenglei Wu, Min H Kim, and Yaser Sheikh. 2019. Strand-accurate multi-view hair capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 155–164.

Paul Noble and Wen Tang. 2004. Modelling and Animating Cartoon Hair with NURBS Surfaces. In *Proceedings of the Computer Graphics International (CGI '04)*. IEEE Computer Society, USA, 60–67.

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. 2021. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 12013–12026.

Dmitry Petrov, Pradyumn Goyal, Vikas Thamizharasan, Vladimir Kim, Matheus Gadelha, Melinos Averkiou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2024. Gem3d: Generative medial abstractions for 3d shape synthesis. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Emmanuel Piuze, Paul G Kry, and Kaleem Siddiqi. 2011. Generalized helicoids for modeling hair geometry. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 247–256.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

Daniel Ritchie, Kai Wang, and Yu-an Lin. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6182–6190.

Alec Rivers, Takeo Igarashi, and Frédo Durand. 2010. 2.5D cartoon models. *ACM Transactions on Graphics* 29, 4 (Jul 2010), 1–7. doi:10.1145/1778765.1778796

Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. 2022. Neural strands: Learning hair geometry and appearance from multi-view images. In *European Conference on Computer Vision*. Springer, 73–89.

Radu Alexandru Rosu, Keyu Wu, Yao Feng, Youyi Zheng, and Michael J Black. 2025. DiffLocks: Generating 3D Hair from a Single Image using Diffusion Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 10847–10857.

Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 2018. 3D hair synthesis using volumetric variational autoencoders. *ACM Transactions on Graphics* (Dec 2018), 1–12. doi:10.1145/3272127.3275019

Takeyuki Sakai and Vladimir Savchenko. 2013. Skeleton-based anime hair modeling and visualization. In *2013 International Conference on Cyberworlds*. IEEE, 318–321.

Yuefan Shen, Shunsuke Saito, Ziyan Wang, Olivier Maury, Chenglei Wu, Jessica Hodgins, Youyi Zheng, and Giljoo Nam. 2023. CT2Hair: High-Fidelity 3D Hair Modeling using Computed Tomography. *ACM Transactions on Graphics* 42, 4, Article 75 (2023), 13 pages.

Jung Shin, Michael Haller, and R. Mukundan. 2006. A stylized cartoon hair renderer. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. 64. doi:10.1145/1178823.1178899

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2024. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19615–19625.

Vanessa Sklyarova, Jenya Chelishev, Andreea Dogaru, Igor Medvedev, Victor Lempitsky, and Egor Zakharov. 2023a. Neural haircut: Prior-guided strand-based hair reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19762–19773.

Vanessa Sklyarova, Egor Zakharov, Otmar Hilliges, Michael J Black, and Justus Thies. 2023b. Haar: Text-conditioned generative model of 3d strand-based human hairstyles. *arXiv preprint arXiv:2312.11666* (2023).

Yusuke Takimoto, Hikari Takehara, Hiroyuki Sato, Zihao Zhu, and Bo Zheng. 2024. Dr. Hair: Reconstructing Scalp-Connected Hair Strands without Pre-Training via Differentiable Rendering of Line Segments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20601–20611.

VRoid. 2022. VRoid Hub. https://vroid.com/

Hongsheng Wang, Xinrui Zhou, and Feng Lin. 2024. NOVA-3D: Non-overlapped Views for 3D Anime Character Reconstruction. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia Workshops*. 1–7.

Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. 2021. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 106–115.

Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, Tong Zhang, Shenghua Gao, and C. L. Philip Chen. 2024. Scaling Mesh Generation via Compressive Tokenization. *arXiv preprint arXiv:2411.07025* (2024).

Keyu Wu, Lingchen Yang, Zhiyi Kuang, Yao Feng, Xutao Han, Yuefan Shen, Hongbo Fu, Kun Zhou, and Youyi Zheng. 2024. Monohair: High-fidelity hair modeling from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24164–24173.

Zhan Xu and Xue Dong Yang. 2001. V-HairStudio: an interactive tool for hair design. *IEEE Computer Graphics and Applications* 21, 1 (Jan 2001), 36–43. doi:10.1109/38.920625

Han Yan, Yang Li, Zhennan Wu, Shenzhou Chen, Weixuan Sun, Taizhang Shang, Weizhe Liu, Tian Chen, Xiaqiang Dai, Chao Ma, et al. 2024. Frankenstein: Generating semantic-compositional 3d scenes in one tri-plane. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Xue Dong Yang, Zhan Xu, Jun Yang, and Tao Wang. 2000. The Cluster Hair Model. *Graphical Models* (Mar 2000), 85–103. doi:10.1006/gmod.1999.0518

Jingwen Ye, Yuze He, Yanning Zhou, Yiqin Zhu, Kaiwen Xiao, Yong-Jin Liu, Wei Yang, and Xiao Han. 2025. PrimitiveAnything: Human-Crafted 3D Primitive Assembly Generation with Auto-Regressive Transformer. arXiv:2505.04622 [cs.GR]

Chih-Kuo Yeh, Pradeep Kumar Jayaraman, Xiaopei Liu, Chi-Wing Fu, and Tong-Yee Lee. 2014. 2.5 D cartoon hair modeling and manipulation. *IEEE Transactions on Visualization and computer graphics* 21, 3 (2014), 304–314.

Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair meshes. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 1–7.

Egor Zakharov, Vanessa Sklyarova, Michael Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. 2024. Human hair reconstruction with strand-aligned 3d gaussians. In *European Conference on Computer Vision*. Springer, 409–425.

Lei Zhang, Hua Huang, and Hongbo Fu. 2012. EXCOL: An EXtract-and-COmplete Layering Approach to Cartoon Animation Reusing. *IEEE Transactions on Visualization and Computer Graphics,IEEE Transactions on Visualization and Computer Graphics* (Jul 2012).

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. CLAY: A Controllable Large-scale Generative

Model for Creating High-quality 3D Assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. 2020. H3dnet: 3d object detection using hybrid geometric primitives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 311–329.

Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. 2025. DeepMesh: Auto-Regressive Artist-mesh Creation with Reinforcement Learning. *arXiv preprint arXiv:2503.15265* (2025).

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. 2023. Michelangelo: Conditional 3D Shape Generation based on Shape-Image-Text Aligned Latent Representation. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=xmxgMij3LY

Yujian Zheng, Zirong Jin, Moran Li, Haibin Huang, Chongyang Ma, Shuguang Cui, and Xiaoguang Han. 2023. Hairstep: Transfer synthetic to real using strand and depth maps for single-view 3d hair modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12726–12735.

Yuxiao Zhou, Menglei Chai, Alessandro Pepe, Markus Gross, and Thabo Beeler. 2023. Groomgen: A high-quality generative hair model using hierarchical latent representations. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–16.