Garment Particles:
A 2D–3D Symmetric Garment Representation for Generation and Editing

KIYOHIRO NAKAYAMA, Stanford University, USA
I-CHAO SHEN, The University of Tokyo, Japan
RUOFAN LIU, Institute of Science Tokyo, Japan
YIMING WANG, ETH Zurich, Switzerland
GORDON WETZSTEIN, Stanford University, USA
TAKEO IGARASHI, The University of Tokyo, Japan

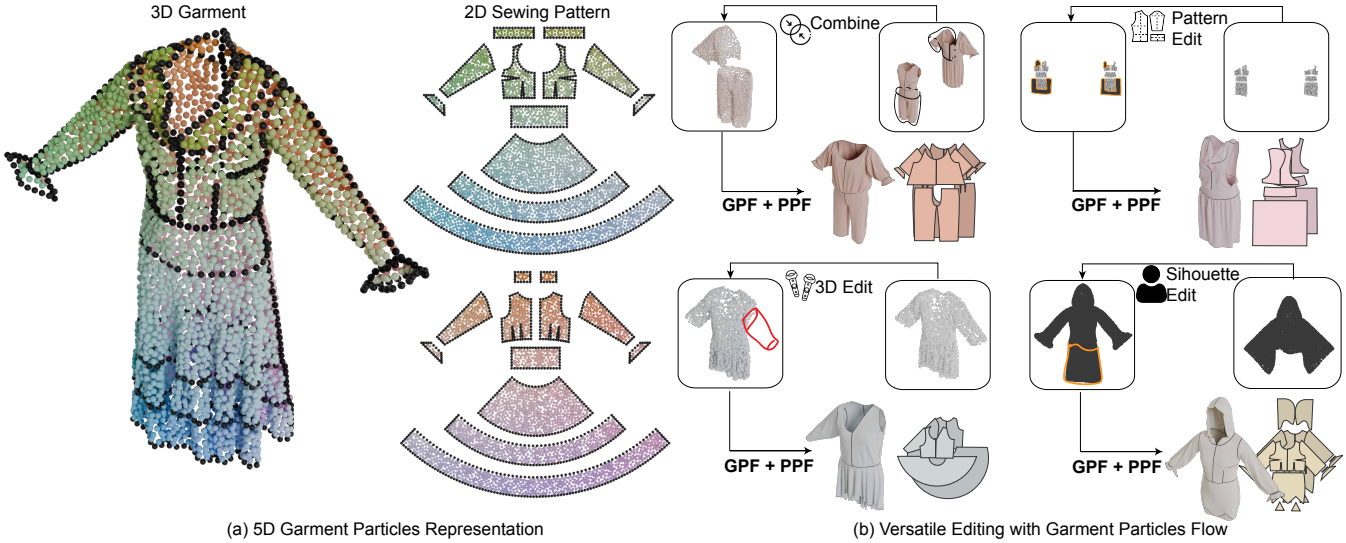

Fig. 1. **Garment Particles** is a garment representation that models both the sewing pattern and its draped garment geometry in a symmetric, 2D-3D point cloud. (a) shows the garment particles representation. The color on the 3D garment (left) and the 2D sewing pattern (right) indicate the same points. *Garment Particles Flow (GPF)*, a generative framework, generates garment particles from multimodal inputs. More importantly, the prior space of GPF enables versatile editing in both 3D garment geometry and 2D sewing pattern domains. Finally, Particles-to-Pattern Flow (PPF) converts the generated particles to simulation-ready sewing patterns. (b) shows the various editing applications enabled by GPF.

Practical garment design spans two modes: intuitive creation from high-level intent, such as a reference image or text description, and complex low-level editing across 2D sewing patterns and 3D draped geometry, which requires professional training to navigate their complex interdependencies. Yet existing frameworks address only part of this challenge, offering either garment generation from casual inputs or direct editing on sewing patterns. To support both ends of the spectrum, we propose Garment Particles, a 5D point-cloud representation that jointly encodes 2D sewing patterns and 3D geometry. This representation enables Garment Particles Flow (GPF), a rectified flow framework that supports intuitive generation from high-level inputs (text, images, sketches) and various editing operations on 2D sewing patterns and 3D geometries via diffusion posterior sampling. Finally, we introduce Particles-to-Pattern Flow that converts generated garment particles into curved-based patterns for simulation. We validate our model's generation ability on multiple datasets, achieving state-of-the-art garment generation results against competitive baselines. Our model also enables many garment editing scenarios, including garment interpolation, sewing pattern editing, point-cloud- and silhouette-conditioned garment generation. Our project website is at https://garment-particles.github.io.

CCS Concepts: • **Computing methodologies** → **Parametric curve and surface models**; **Point-based models**.

Additional Key Words and Phrases: Garment Generation, Garment Editing, Garment Representation, Diffusion Posterior Sampling

ACM Reference Format:
Kiyohiro Nakayama, I-Chao Shen, Ruofan Liu, Yiming Wang, Gordon Wetzstein, and Takeo Igarashi. 2026. Garment Particles: A 2D–3D Symmetric Garment Representation for Generation and Editing. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*

Authors' Contact Information: Kiyohiro Nakayama, Stanford University, USA; I-Chao Shen, The University of Tokyo, Japan; Ruofan Liu, Institute of Science Tokyo, Japan; Yiming Wang, ETH Zurich, Switzerland; Gordon Wetzstein, Stanford University, USA; Takeo Igarashi, The University of Tokyo, Japan.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811102

(SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA. ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3799902.3811102

1 Introduction

Garments are a fundamental aspect of everyday life, yet garment design remains technically demanding and largely left to professionals. The primary challenge lies in the pattern-making process, where 2D sewing panels must be carefully shaped to achieve the desired 3D appearance on the body. Professionals often rely on both intuitive creation from reference images or text descriptions and counter-intuitive 2D pattern edits, such as adding darts or adjusting seams, to achieve the desired 3D volume or curvature.

In the digital domain, two primary paradigms have emerged to facilitate the garment design process. Industry-standard tools, such as CLO3D [Fashion 2009], Style3D [Style3D 2015], or Marvelous Designer, allow users to directly edit structured, low-level garment representations, including topologically consistent panels and Bézier curves. While precise, these tools require strong pattern-making expertise, as users must understand the complex causal relationship between the geometry of a 2D sewing pattern and its 3D appearance.

Separately, generative models for sewing patterns [Bian et al. 2025; Liu et al. 2025a; Nakayama et al. 2025; Tatsukawa et al. 2025] create and edit sewing patterns given multimodal, high-level inputs such as texts, images, and 3D scans. While these methods enable rapid pattern development, they rely on modality-specific training, making them ill-suited for garment design, since it often requires different editing tools applied to both the 2D sewing pattern and 3D garment geometry. Further, training a single model to cover all operations is difficult because it would require operation-specific data collection, allocation of training budgets, and careful balancing of the combination of control signals [He et al. 2024a; Wang et al. 2025].

To address these limitations, we draw inspiration from the image generation community and cast different garment editing tasks as a training-free inverse problem using diffusion posterior sampling (DPS) [Chung et al. 2023; Kim et al. 2025; Patel et al. 2024]. With DPS, different garment editing applications can be solved by guiding the diffusion sampling process using a specific objective, without retraining the model. However, directly applying DPS to existing garment generative models is challenging, as these models are agnostic to the garment's 3D geometry post-draping. As a result, they learn a 3D-agnostic generative prior that cannot leverage the 3D drape to guide the generated sewing pattern toward a desired draped configuration. The geometry image representation [Gu et al. 2002; Li et al. 2025b; Yan et al. 2024] addresses this disconnect by rasterizing 2D sewing patterns into images whose opacity encodes the pattern shape and colors the 3D appearance. While this representation captures the 2D–3D duality, it exhibits an *asymmetry* in which the 3D appearance *depends on* the panel shape (i.e., pixel opacity). Consequently, to recover the 3D geometry, a non-differentiable discretization must be performed first to determine the occupied pixels. This makes it difficult to optimize DPS with any objective defined in 3D space directly with respect to the representation itself.

In this paper, we propose a novel point-based garment representation, *garment particles*. Garment particles encode both the 2D sewing pattern and its 3D geometry as a 5-dimensional point cloud (Figure 1a), which can be mapped to the 2D or 3D space via a simple, differentiable projection function. Using garment particles as its underlying representation, we train a flow-based generative model, *garment particles flow* (GPF), and a Particles-to-Pattern Flow (PPF) to generate simulation-ready sewing patterns from multimodal inputs such as sketches, images, and text prompts.

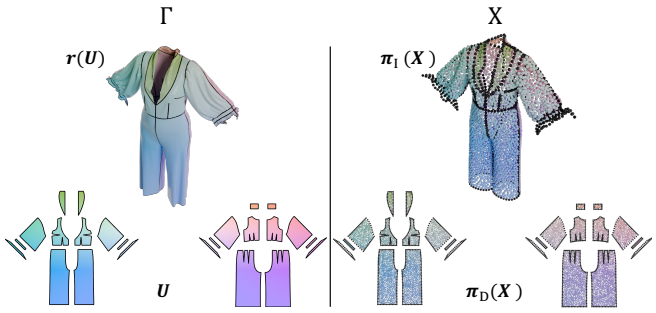

Fig. 2. **Garment Particles Illustration.** *(Left)* We model garments as the graph Γ of the parametric function mapping sewing pattern U in $\mathbb{R}^2$ to its draped geometry $\boldsymbol{r}(U)$ in $\mathbb{R}^3$. *(Right)* We discretize Γ by point samples, denoted as $\boldsymbol{X}_\Gamma$. Points with the same color in 2D and 3D represent the corresponding points in our representation. Black points mark the boundary of U.

More importantly, GPF enables applications of objective-guided sampling techniques, such as DPS, to garments by capturing symmetric relationships between the 2D sewing pattern and its 3D garment geometry. With GPF, we achieve symmetric, iterative garment editing using a diverse set of tools defined in both 2D pattern and 3D drape spaces. Finally, the generated garment particles can be converted back to simulation-ready sewing patterns using PPF.

Experimentally, we achieve state-of-the-art generation performance across input modalities and demonstrate the versatility of the GPF prior space across multiple garment-editing tasks. We implemented custom 2D and 3D interactive user interfaces that allow users to casually manipulate 2D sewing patterns, silhouettes, and 3D garment geometries, and use them to guide model generation while maintaining realism and validity.

Our contributions are:

- *Garment Particles.* A novel point-based representation that jointly encodes the 2D sewing pattern and 3D draped geometry of a garment, enabling a fully differentiable pipeline for objective-guided sampling.
- *Garment Particles Flow.* A rectified flow model that generates garment particles from multimodal inputs and a particles-to-pattern flow model that recovers simulation-ready sewing patterns from generated garment particles. Our framework achieves state-of-the-art performance in garment generation from texts and images.
- *Garment Editing via Diffusion Posterior Sampling.* The learned prior space enables various garment editing applications using DPS, including sewing pattern editing, point-cloud-conditioned garment generation, and silhouette-conditioned garment generation.

2 Related Work

2.1 Digital Garment Design

Many automation tools have been developed to accelerate the garment modeling process. Industrial software [Fashion 2009; Style3D 2015] digitizes the traditional pattern-making process by integrating pattern making and draping simulation. Meanwhile, academic research has focused on reducing manual effort and improving physical fidelity [Igarashi et al. 2008; Korosteleva and Sorkine-Hornung 2023; Rodríguez and Cirio 2022]. However, these tools require the users to understand the relationship between the sewing pattern and its 3D garment geometry.

To enable garment design for casual users, research has also focused on automating the pattern-making process. Early work [Umetani et al. 2011] relies on predefined templates to update 2D patterns from partial 3D edits, whereas later work [Bartle et al. 2016; Brouet et al. 2012; Liu et al. 2018; Meng et al. 2012b; Qi and Igarashi 2024; Skouras et al. 2014; Wolff et al. 2021; Zhang et al. 2019] eliminates the need for templates and enables garment editing via inverse cloth simulation and heuristics. In parallel, prior work also addresses specific components in pattern making, such as dart [de Malefette et al. 2023], pleat placement [Li et al. 2018], fabric texture alignment [Wolff and Sorkine-Hornung 2019], sewing pattern refitting [Chen et al. 2025; Eggler et al. 2024; Meng et al. 2012a] and repurposing [QI et al. 2025]. However, these methods are limited in their generalizability to complex garments, leading to unrealistic outputs and requiring long optimization times. These limitations make them unsuitable for practical garment design.

Another line of work directly recovers sewing patterns from casual inputs. [Daanen and Hong 2008; Pietroni et al. 2022; Sharp and Crane 2018; Yang et al. 2018] recovers sewing patterns from images or 3D draped garments using traditional geometric modeling and physics-based simulation. Recently, deep learning-based methods [Korosteleva and Lee 2022; Li et al. 2025a, 2024a, 2023, 2025d; Liu et al. 2023; Tatsukawa et al. 2025; Tian et al. 2025; Wang et al. 2018] enable sewing-pattern recovery from a broader range of input types but are limited to a single modality. To support multi-modal inputs, autoregressive-based [Bian et al. 2025; Can et al. 2026; He et al. 2024b; Nakayama et al. 2025; Zhou et al. 2024] and diffusion-based [Li et al. 2025b,c; Liu et al. 2025a] generative models have been adopted. However, these methods rely on a sewing-pattern-centric representation that is decoupled from the final draped garment geometry, making them unsuitable for iterative garment editing that requires symmetric interactions between 2D sewing patterns and 3D garments.

2.2 Diffusion Models for 3D Generation and Editing

Diffusion-based 3D generative models have advanced rapidly and employ different representations tailored to downstream applications. Specifically, closer to our representation, geometry-image models generate geometry images [Elizarov et al. 2024; Yan et al. 2024; Zhang et al. 2025] but ignore surface connectivity. BRep-based models [Lee et al. 2025; Liu et al. 2025c; Xu et al. 2024] rely on low-resolution surface patches and post-processing, losing geometric details and struggling with complex structures on sewing patterns, e.g., darts. In contrast, Garment Particles models 2D sewing patterns using a two-stage pipeline. The first stage learns a rich prior space over garment particles that encodes both the 2D sewing patterns and 3D draped geometries as a 5D point cloud. This enables casual generation and symmetric garment editing with DPS. The second stage recovers the sewing pattern from the generated garment particles for downstream applications such as cloth simulation.

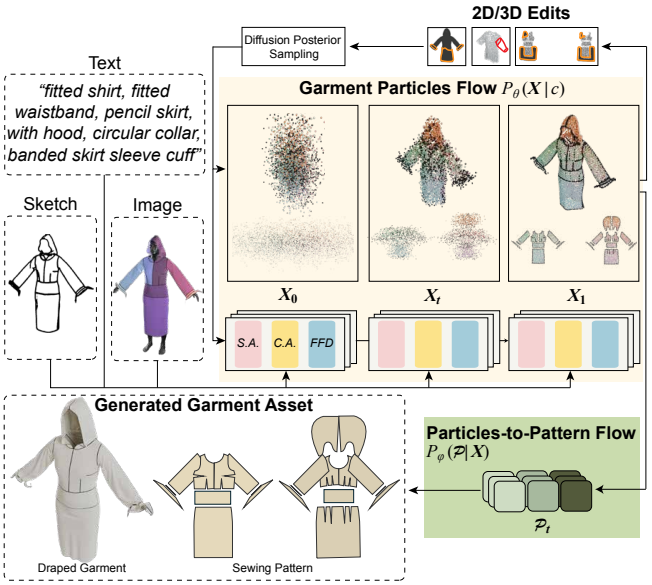

Fig. 3. **Garment Particles Flow (GPF)** is a generative model that generates simulation-ready garments via a two-stage pipeline. In the first stage, multimodal inputs, such as text, sketches, and images, are fed to GPF via cross-attention to generate garment particles $\boldsymbol{X}_1$. Diffusion posterior sampling guides the generation based on users' edits. The generated garment particles are then fed into Particles-to-Pattern Flow to generate a vectorized sewing pattern usable for downstream simulation.

Originally introduced for inverse image restoration tasks [Chung et al. 2023], diffusion posterior sampling (DPS) was used to guide the generation process toward specific objectives. Later works extended this technique to flow models [Kim et al. 2025; Patel et al. 2024] and to a broader range of generative domains, such as robotics and motion planning [Rempe et al. 2023; Yin et al. 2025], medical imaging [Li et al. 2024b], and audio signal processing [Taufik and Alkhalifah 2025]. We extend FlowDPS [Kim et al. 2025] to support diverse garment editing tasks using different guidance objectives.

3 Garment Particles

We build a garment representation that encodes both the 2D sewing pattern and the draped 3D garment geometry. Furthermore, the representation should enable both garment-generation and garment-editing applications powered by a learned model. To meet these two criteria, we propose Garment Particles, a garment representation consisting of 5-dimensional particles. Using garment particles, we propose Garment Particles Flow (GPF), a generative framework that can generate and edit garment particles, and convert them to simulation-ready garment assets (Figure 3).

3.1 Representing Garment as Particles

Mathematically, a cut-and-sew garment can be represented as a 2D parametric surface given by a parametric equation $\boldsymbol{r}: U \to \mathbb{R}^3$ where $U \subset \mathbb{R}^2$ is a compact domain. In this setting, the domain U is the sewing pattern, and its image $\boldsymbol{r}(U)$ is the 3D garment geometry after draping[1]. Therefore, modeling both the sewing pattern and the garment geometry reduces to modeling the graph of $\boldsymbol{r}$, which consists of points $\Gamma(\boldsymbol{r}) = \{(\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x}))\}$ for all $\boldsymbol{x} \in U$ in the sewing pattern (See left of Figure 2). We can readily recover both the sewing pattern U and its draped 3D garment geometry $\boldsymbol{r}(U)$ using the projection operators π_D and π_I onto the domain and the image. Such operators are differentiable and computationally simple, making them ideal for garment representation.

To train a generative model on $\Gamma(\boldsymbol{r})$, we discretize it as a 5D point cloud, which we dub the *garment particles* of $\boldsymbol{r}$ (See right of Figure 2). Notationally, we use $\boldsymbol{X}^{\boldsymbol{r}}$ to denote the garment particles of $\boldsymbol{r}$ and $\boldsymbol{x}^{\boldsymbol{r}}$ to denote a point sample from $\boldsymbol{X}^{\boldsymbol{r}}$. We drop the superscript $\boldsymbol{r}$ when there is no ambiguity.

In practice, we include an additional boundary flag, $f_{\boldsymbol{x}}$, indicating whether a point lies on the boundary of a panel to facilitate downstream sewing pattern reconstruction. Concretely, each of our garments $\boldsymbol{r}$ is converted to $\boldsymbol{X}^{\boldsymbol{r}}$ defined as

$$\boldsymbol{X}^{\boldsymbol{r}} = \left\{ (\boldsymbol{x}^{\boldsymbol{r}}, f_{\boldsymbol{x}^{\boldsymbol{r}}}) : \boldsymbol{x}^{\boldsymbol{r}} \in \Gamma(\boldsymbol{r}) \right\}, \quad f_{\boldsymbol{x}^{\boldsymbol{r}}} = \begin{cases} 1 & \pi_D(\boldsymbol{x}^{\boldsymbol{r}}) \in \partial U; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Garment Particles Construction. Given a garment as a mesh, we re-triangulate it with area constraints and use its vertices as point samples. This ensures the point count is roughly proportional to the panel area. To construct Γ for each garment, we place its sewing pattern in $\mathbb{R}^2$ without intersection. To ensure semantic consistency across garments, we initialize each panel's location from the 2D projection of its draping initialization transformation, and iteratively resolve panel-wise overlap based on each panel's label (e.g., sleeve, torso, or waistband). Figure 2 shows a packed sewing pattern (*left*) and its associated garment particles (*right*).

3.2 Garment Particles Flow (GPF)

We train a rectified flow model to learn a generative prior over garment particles, $P_\theta(\boldsymbol{X})$, for garment generation and editing. Specifically, we model a probability flow that maps noise $\boldsymbol{X}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \boldsymbol{X}_0 \in \mathbb{R}^{N \times 6}$ to garment particles $\boldsymbol{X}_1$ from the training data,

$$\frac{d\boldsymbol{X}_t}{dt} = \boldsymbol{v}_\theta(\boldsymbol{X}_t, t; \boldsymbol{c}) \tag{2}$$

where $\boldsymbol{v}_\theta$ models the drift field using weights θ. We can optionally supply $\boldsymbol{v}_\theta$ with a conditioning signal $\boldsymbol{c}$, which may take the form of text or images. In rectified flow, we assume linear interpolation between any pairs of $(\boldsymbol{X}_0, \boldsymbol{X}_1)$.

$$\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0 \quad \frac{d\boldsymbol{X}_t}{dt} = \boldsymbol{X}_1 - \boldsymbol{X}_0. \tag{3}$$

Thus, the learned drift field $\boldsymbol{v}_\theta$ should approximate $\boldsymbol{X}_1 - \boldsymbol{X}_0$ as much as possible. This motivates us with the standard flow matching loss

$$\mathcal{L}_{\text{flow}} = \| \boldsymbol{X}_1 - \boldsymbol{X}_0 - \boldsymbol{v}_\theta(\boldsymbol{X}_t, t; c) \|_2^2 \tag{4}$$

for training. We use Diffusion Transformers (DiT) [Peebles and Xie 2023; Yao et al. 2025] as our model architecture. Because the garment particles are unordered, we eliminate positional encodings in the DiT. We set a maximum number of points to 8192 and use masking during training. During inference, the number of points is provided as input, controlling the complexity of the generated sewing pattern.

Injecting Text Condition. We train GPF to optionally take an input text prompt to guide generation. Specifically, given a text prompt $\mathcal{T}$, we first encode it using CLIP [Radford et al. 2021] to obtain text embeddings $\boldsymbol{c} \in \mathbb{R}^{77 \times 768}$. Then $\boldsymbol{c}$ is linearly mapped to the same latent space as the GPF model and injected into $\boldsymbol{v}_\theta$ via cross attention following [Xiang et al. 2024].

We use the LightningDiT-XL variant from [Yao et al. 2024, 2025], which consists of 28 layers of transformer blocks. See the supplementary for additional training details.

Extending GPF to Image Conditions. After text-conditioned training, GPF learns a generalizable prior space that can be easily extended to other modalities without training from scratch. We follow [Ye et al. 2023; Zhang et al. 2024b] and extend GPF to accept images as an extra condition by adding an extra cross-attention in each transformer block of GPF. The images are tokenized with a frozen DINOv2 [Oquab et al. 2023] encoder and then attended to via cross-attention. We initialize the image-conditioned training from the pre-trained, text-conditioned GPF and fine-tune all layers for 160,000 iterations.

3.3 Recovering Sewing Patterns from Garment Particles

It is nontrivial to recover simulation-ready sewing patterns from generated garment particles, as the generation process may be corrupted by noise. To address this issue, we design Particles-to-Pattern Flow to convert garment particles to a curve-based sewing pattern. Specifically, we represent a sewing pattern $\mathcal{P}$ as a tensor of shape $M_{max} \times (E_{max}+1) \times D$ following [He et al. 2024b; Liu et al. 2025a; Nakayama et al. 2025]. A pattern consists of a maximum of M_{max} panels, each containing up to E_{max} ordered parametric edges (cubic Bézier curves or arcs). We set $M_{max} = E_{max} = 37$ to cover all patterns in our dataset and represent each panel P_i as:

$$P_i = \text{Stack}\big((T_i, R_i), e_1, \ldots, e_{E_{\max}}\big) \tag{5}$$

where $(T_i, R_i) \in \mathbb{R}^6$ encodes the panel's draping pose (translation and Euler angles), and each edge $e_j \in \mathbb{R}^{15}$ contains information such as control points ($\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{R}^4$), displacement from previous endpoint ($\delta\mathbf{x} \in \mathbb{R}^2$), arc flag ($f_{\text{arc}} \in \mathbb{R}$), stitching flag ($f_{\text{stitch}} \in \mathbb{R}$), stitch tag ($\boldsymbol{\tau} \in \mathbb{R}^3$), boundary condition type ($\mathbf{t}_{\text{attach}} \in \mathbb{R}^3$), and validity mask ($f_{\text{valid}} \in \mathbb{R}$).

We formulate sewing pattern recovery as a generation task conditioned on garment particles $\boldsymbol{X}$. We model this conditional distribution $P_\varphi(\mathcal{P}|\boldsymbol{X})$ using another flow model $\boldsymbol{v}_\varphi$. Unlike GPF, which generates an unordered set of points, sewing patterns consist of ordered edges and panels. Therefore, we use panel and edge embeddings to order the input tokens and employ cross-attention to condition the network.

PPF learns the garment-particles-to-sewing-pattern mapping purely from data. Although it does not enforce hard constraints—such as boundary points lying exactly on panel edges or interior points staying inside panels—PPF empirically offers the best trade-off between robustness to input noise and reconstruction accuracy[2].

[1] More precisely, $\boldsymbol{r}$ also depends on body shape and pose that the garment drapes on. Here we drop these dependencies by assuming the draping is performed on the same body for all garments.

[2] See Supplementary for a detailed analysis

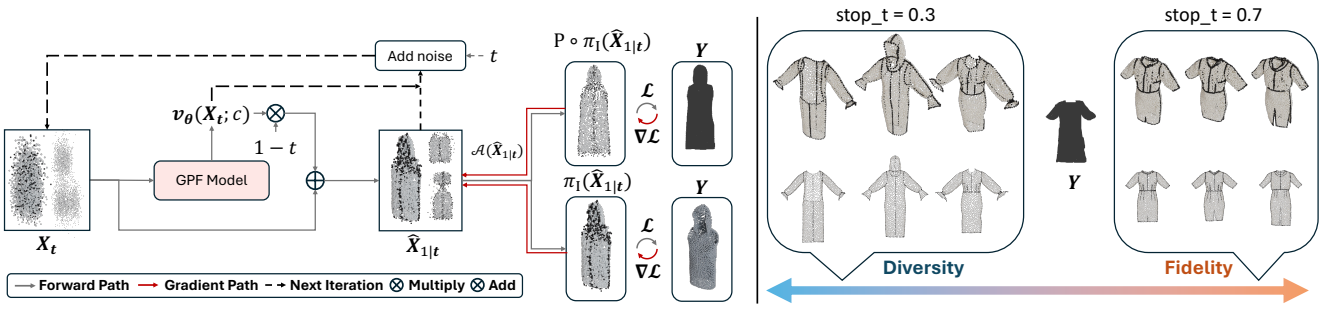

Fig. 4. **Objective Guided Interactions**. *(Left)* By leveraging a trained GPF model, we can optimize the posterior mean $\hat{X}_{1|t}$ at each step against an observation Y and guide the generation process towards a garment sample that minimizes our specified objective $\mathcal{L}$. *(Right)* By adjusting the hyperparameter `stop_t`, our objective-guided sampling can produce more faithful (higher `stop_t`) or more diverse (lower `stop_t`) results with different random noise input.

4 GPF-driven Garment Editing

After training GPF, we can leverage its prior distribution $P_\theta(X)$ for various *training-free* garment editing tasks.

4.1 Garment Interpolation

A quick way to generate a diverse set of garments is via interpolation in the prior space $P(X_0) = \mathcal{N}(\mathbf{0}, I)$ and passing the interpolated noise through GPF. We use spherical linear interpolation (SLERP) for noise interpolation. Because the particles are unordered, we follow [Lee et al. 2025] and first compute a linear assignment between the particles by summing pairwise distances across multiple denoising timesteps. Because GPF requires specifying the particle count per garment, we linearly interpolate the particle count for each intermediate generation when the endpoints' point counts differ.

4.2 Objective Guided Editing

To enable different garment editing tasks without dedicated training, we use diffusion posterior sampling (DPS) [Chung et al. 2023; Kim et al. 2025; Patel et al. 2024]. DPS solves inverse problems of the form

$$X^\star = \underset{X \sim P_\theta(X)}{\operatorname{argmin}} \mathcal{L}(\mathcal{A}(X), Y). \tag{6}$$

Here $\mathcal{L}$ is the objective function, $\mathcal{A}$ is the forward transformation, and Y is the observation. Given an observation Y which can be derived from some sample X via transformation $\mathcal{A}$, DPS seeks the closest sample from our trained prior distribution $P_\theta(X)$ that minimizes the objective function $\mathcal{L}$ after applying $\mathcal{A}$. Adapted from FlowDPS [Kim et al. 2025] for garment particle editing tasks, our sampling algorithm is summarized in Algorithm 1. By adjusting hyperparameters (`stop_t`, `opt_n`, and T), we can balance sample diversity and fidelity (Figure 4). By varying the conditioning vector c across different prompts, we can impose additional control over the generation while minimizing the objective function. We demonstrate four garment editing operations in the following. See Supplementary Material for specific hyperparameter settings.

Point-cloud-conditioned Garment Generation. Given the 3D garment geometry as a point cloud, we can generate a suitable sewing pattern that matches the input post-draping. We formulate this as a DPS task:

$$\begin{aligned} \mathcal{A} &= \pi_I : \mathbb{R}^{N\times 6} \to \mathbb{R}^{N\times 3} \\ \mathcal{L}(Y_1, Y_2) &= \mathrm{EMD}(Y_1, Y_2), Y_i \in \mathbb{R}^{N\times 3}. \end{aligned} \tag{7}$$

Here, $\mathcal{A}$ is a projection function that maps the 5D garment particles X_r to the image $r(U)$ and EMD is the Earth Mover Distance [Rubner et al. 2000]. The generated garment particles will approximate the given 3D geometry when projected to the image domain with π_I.

Garment Completion. Given an incomplete 3D garment geometry as a point cloud, we can complete it using DPS. For this task, we set

$$\begin{aligned} \mathcal{A} &= \pi_I : \mathbb{R}^{N\times 6} \to \mathbb{R}^{N\times 3} \\ \mathcal{L}(Y_1, Y_2) &= \sum_{\tilde{y}\in Y_2} \min_{y \in Y_1} \|\tilde{y} - y\|_2^2. \end{aligned} \tag{8}$$

Here, we set the objective function to the one-sided Chamfer Distance, which encourages Y_2 to lie within Y_1. In this way, the generated garment particles incorporates the observation Y into its geometry,

Algorithm 1 Diffusion Posterior Sampling of GPF

Require: Trained GPF v_θ, Objective function $\mathcal{L}$, Transformation $\mathcal{A}$, Sampling steps T, Observation Y, Conditioning c, Guidance Stop Time `stop_t`, Learning rate η, Optimization Steps `opt_n`.

$X \sim \mathcal{N}(\mathbf{0}, I)$ $\Delta t \leftarrow \frac{1}{T}$
for $\delta = 0,1,2,\ldots,T-1$ **do**
  $t \leftarrow \frac{\delta}{T}$, $v \leftarrow v_\theta(X, t; c)$
  $\hat{X}_{0|t} \leftarrow X - tv$, $\hat{X}_{1|t} \leftarrow X + (1-t)v$
  if $t \le$ `stop_t` **then**
    for $n = 1,\ldots,$`opt_n` **do**
      $\hat{X}_{1|t} \leftarrow \hat{X}_{1|t} - \eta \nabla_{\hat{X}_{1|t}} \mathcal{L}\left(\mathcal{A}\left(\hat{X}_{1|t}\right), Y\right)$
    end for
    $\varepsilon \sim \mathcal{N}(\mathbf{0}, I)$
    $X_{0|t} \leftarrow \sqrt{t+\Delta t} X_{0|t} + \sqrt{1-t-\Delta t}\varepsilon$
  end if
  $X \leftarrow (t+\Delta t)\hat{X}_{1|t} + (1-t-\Delta t)\hat{X}_{0|t}$
end for

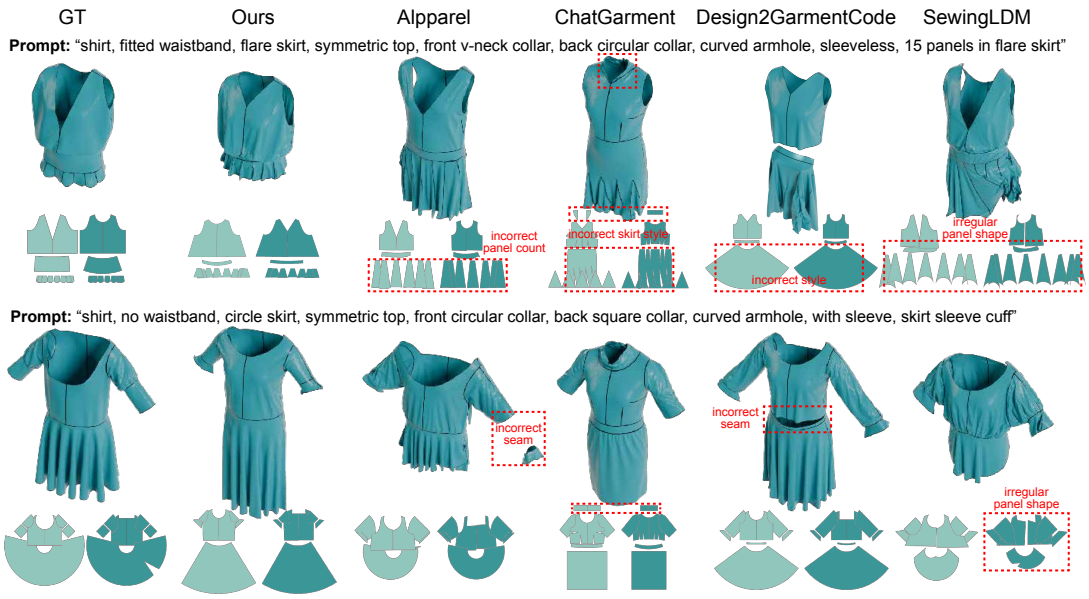

Fig. 5. **Text-conditioned Garment Generation.** The baselines exhibit artifacts, as indicated by the red boxes (e.g., incorrect panel shapes or styles). In contrast, our method outputs realistic garments that align with the input prompt.

thereby completing the 3D garment geometry in the image while generating its sewing pattern in the domain.

Sewing Pattern Editing. Thanks to our symmetric representation, we can also edit the 2D sewing pattern and obtain the resulting garment with our model. By replacing $\mathcal{A}$ in Equation 7 and Equation 8 with π_D, we can leverage DPS to reconstruct 3D garment geometry from a coarse or incomplete sewing pattern as the observation. Compared with directly editing vectorized sewing patterns, our method automatically generates stitching and draping initialization cues using our particle-to-pattern module.

Silhouette-conditioned Garment Generation from Arbitrary Views. Given the silhouette of a garment taken from view $P \in \mathbb{R}^{3\times2}$, we can guide our generation with DPS to sample $\boldsymbol{X}$ that has a similar silhouette when projected to P. This is done by setting

$$\begin{aligned} \mathcal{A} &= P \circ \pi_I : \mathbb{R}^{N\times6} \to \mathbb{R}^{N\times2} \\ \mathcal{L}(Y_1, Y_2) &= \mathrm{EMD}(Y_1, Y_2), Y_i \in \mathbb{R}^{N\times2}. \end{aligned} \tag{9}$$

$\mathcal{A}$ here is the composition of the coordinate projection π_I and the view projection P, mapping the garment particles into the camera view space. For the objective function, we use EMD in $\mathbb{R}^2$ to encourage the alignment of the generated and observed silhouettes.

5 Garment Generation Results

Dataset. We evaluate our model's generation performance on GarmentCodeDatav2 (GCDv2) [Korosteleva et al. 2024]. We obtain the garment particles of each garment in GCDv2 as described in Section 3.1. In total, we obtained 124,339 samples. We randomly split them into training and validation in a 9:1 ratio. To compute all metrics, we randomly sample 1,024 garments from the validation set.

Table 1. **Text-conditioned Garment Generation.**

Method	COV↑	MMD ($\times10^3$)↓	1-NNA↓	P-FID↓	CLIP↑	SSR↑
Omage	40.6	7.70	68.1	68.8	24.26	–
D2GC	27.6	6.29	81.2	18.9	**27.15**	85.4
AIpparel	24.6	8.21	87.6	89.4	24.11	70.7
ChatGarment	13.5	8.90	88.4	16.2	25.71	**99.5**
SewingLDM	40.1	5.29	62.7	3.69	26.17	81.3
Ours	**48.4**	**4.64**	**54.6**	**3.15**	26.42	91.7

Baselines. We compare our method against various multi-modal sewing pattern generation baselines, including AIpparel [Nakayama et al. 2025], ChatGarment [Bian et al. 2025], SewingLDM [Liu et al. 2025a], and Design2GarmentCode (D2GC) [Zhou et al. 2024]. We also compare with geometry-image-based methods Omages [Yan et al. 2024] by adapting for garment generation.

5.1 Text-based Generation

We evaluate the generation diversity and fidelity of our method when using text as control. We constructed the captions procedurally, leveraging GCDv2's design parameters, which specify the garments' design details.[3] To ensure a fair comparison, we retrain AIpparel, SewingLDM, and Omages on our captioned dataset. For ChatGarment and Design2GarmentCode, we use their released checkpoints as they use LLMs for text-prompt conversion.

Evaluation Metrics. We measure the 3D garment generation quality with a diverse set of metrics: *3D distribution metrics* (Coverage

[3] See Supplementary Material for construction details.

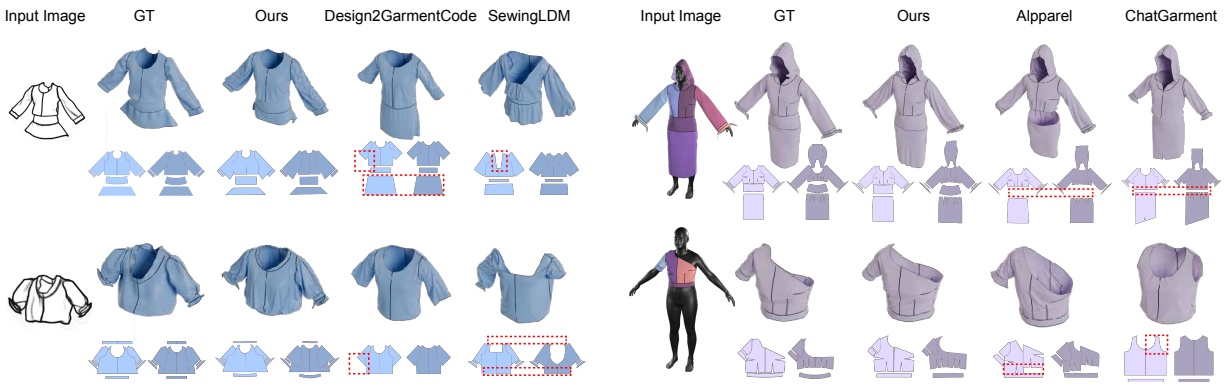

Fig. 6. **Image-conditioned Garment Generation.** Compared to the baselines, which exhibit incorrect pattern style and stitching, our method correctly generates a sewing pattern that yields a draped garment matching the input image for both sketch and GCDv2 image inputs.

Table 2. **Image-conditioned Garment Generation.**

	Panel Acc ↑	Panel IOU ↑	Stitch Acc ↑	SSR ↑	CD ↓
GCDv2					
AIpparel	75.49	75.47	64.67	80.57	16.3
ChatGarment	5.86	62.67	37.72	**95.12**	55.6
Ours	**83.01**	**78.20**	**69.47**	89.84	**7.0**
Garment Sketches					
SewingLDM	20.61	54.47	43.49	80.18	54.2
D2GC	21.48	58.92	40.41	**97.95**	32.9
Ours	**81.25**	**77.05**	**66.86**	92.97	**8.7**

(COV) score, Minimum Matching Distance (MMD), 1-NN classification Accuracy (1-NNA), and pointcloud-FID (p-FID)), simulation success rate (SSR), and CLIP score [4].

Results. As shown in Table 1, our method achieves the best scores on distribution metrics, demonstrating that the generated garment assets are the most diverse and most similar to the test set distribution. This performance boost can be attributed to our 2D–3D representation that captures the symmetric relationship between a sewing pattern and its draped geometry in 3D. By contrast, none of the baselines store 3D information in their representations, making their generation agnostic to the 3D shape after draping. While ChatGarment and Design2GarmentCode slightly outperform our method in simulation success rate and text alignment due to direct GarmentCode generation using LLMs, their outputs are less diverse, reflected by lower coverage score and higher 1-NN accuracy. Figure 5 shows qualitative comparisons of the generated garments on different text prompts. The baselines exhibit artifacts, including text misalignment and incorrect seams, panels, and garment styles. In comparison, our model generates high-quality sewing patterns while matching the text descriptions.

5.2 Image-based Generation

We extend GPF to image-based conditioning using the method described in Section 3.2. To evaluate across different image styles, we fine-tune GPF on three image datasets. (1) *GCDv2*: we use the front and back garment renderings from GCDv2 following [Nakayama et al. 2025]. (2) *Garment Sketches*: we follow [Liu et al. 2025a] to convert GCDv2 garment rendering to sketches using [Su et al. 2023]. (3) *Realistic Garments*: We use the dataset from [Bian et al. 2025], which contains realistically textured garments and humans in different poses. For evaluation, we select in-the-wild garment images from 4DDress [Wang et al. 2024] and Fashionpedia [Jia et al. 2020]. For GCDv2 and Garment Sketches, we train a multiview GPF that conditions on averaged DINOv2 features from front and back views.

Evaluation Metrics. We evaluate the quality of the generated sewing patterns against the ground truth using sewing pattern metrics used in prior work [Li et al. 2025b; Liu et al. 2023; Nakayama et al. 2025]. See the Supplementary for details.

Results. Table 2 presents quantitative comparisons between our method and baselines on the GCDv2 and garment sketches datasets. For fairness, we evaluate each baseline only on the dataset it was trained on. Our method consistently outperforms all baselines across metrics except for SSR. This indicates that our method recovers more accurate sewing patterns and produces draped garments that closely match the ground truth. While program-generation-based methods (ChatGarment and D2GC) achieve slightly higher SSR, our method still achieves around 90% SSR and outperforms SVG-generation-based methods (AIpparel and SewingLDM) by around 10%. We show qualitative comparisons on GCDv2 and garment sketches (Figure 6) and in-the-wild images (Figure 10). As shown by the red boxes, Baseline methods sometimes miss fine details such as sleeve cuffs or collars, or produce different garment styles in their generated patterns. In contrast, our method accurately captures garment style and pattern details.

6 Garment Editing Results

6.1 Garment Interpolation

We showcase garment interpolation in Figure 11 using unconditionally generated garments from SewingLDM, Omage, and GPF. SewingLDM exhibits abrupt transitions due to the latent space of its vectorized sewing patterns. Omage's geometry-image-based latent

[4] See the Supplementary Material for metric calculation details.

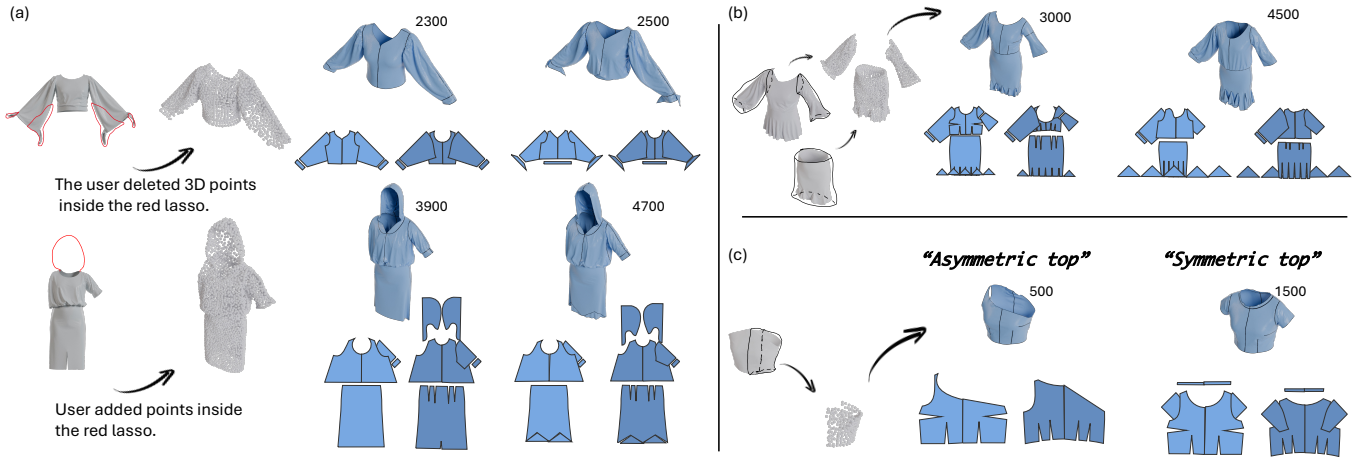

Fig. 7. **Point-Cloud-Conditioned Garment Generation.** We demonstrate various point-cloud-based garment editing applications enabled by GPF. (a) illustrates how users can directly edit an existing 3D garment to guide its generation. Addition and deletion of points are achieved using our 3D interface. (b) shows garment mixing, where components of two existing 3D garments are combined to generate a new garment. (c) shows text-conditioned generation given an incomplete 3D garment. The numbers indicate the number of garment particles used to generate each sample.

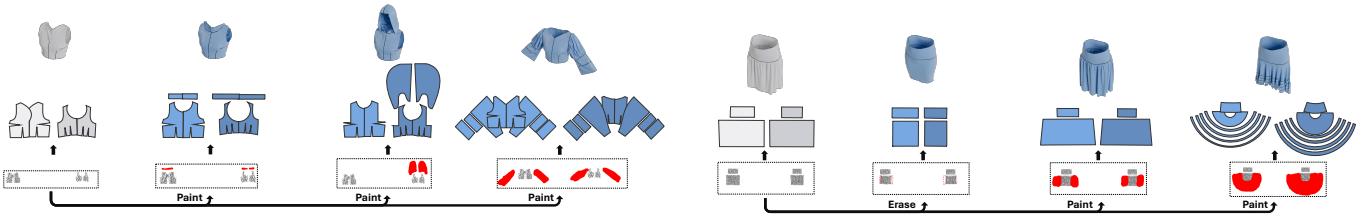

Fig. 8. **Sewing Pattern Editing.** Given a generated garment shown in grey, we edit the sewing pattern and use it to guide the garment generation process. The red part illustrates the user's addition with our 2D user interface.

space enables smoother overall shape interpolation but lacks panel-level correspondences, e.g., the waistband varies arbitrarily in size during the transition. In contrast, our method learn the prior over the joint 2D–3D space, enabling smooth and meaningful transitions even across topological changes, e.g., from asymmetric sleeves to symmetric ones.

6.2 Objective Guided Garment Editing

6.2.1 Point-cloud-conditioned Sewing Pattern Generation. DPS enables garment generation from point clouds without additional training. In Figure 7a, we present the generated garment assets from GPF edited directly in 3D. We generate two sewing pattern variations for each edited garment with varying particle count at input. In Figure 7b, we show an application for mixing the styles of two garments by combining components (the dress sleeve and the godet skirt) of existing garments to create an incomplete observation Y and obtain the resulting garment using Section 4.2. Similarly, we observe different geometries and sewing patterns across runs and input point counts, all of which closely match the observation. We further demonstrate the ability to control generated sewing patterns with text prompts in Figure 7c. We take a garment part (shown as a point cloud) and complete it using additional text prompts "asymmetric" and "symmetric top". Our model generates a garment that matches the input observation while obeying the specified garment style.

6.2.2 Sewing Pattern Editing. Figure 8 shows two examples where patterns generated by our method (grey) are coarsely modified (red part). The modified patterns are then used as observations for DPS. The blue garments show the generation results. We observe that all the patterns align with the inputs' designs while remaining realistic. The 2-panel skirt example shows how to convert it into narrower, wider, and circular variations, which is difficult to specify in 3D.

6.2.3 Silhouette-conditioned Garment Generation. Figure 9 shows silhouette-conditioned garment generation with DPS from different viewpoints. For each example, the left shows the input silhouette drawings. We generate garment variations utilizing different particle counts and text prompts. The generations achieve a balance between alignment with the silhouette, text prompt, and garment realism.

We showcase the fidelity and diversity trade-off by adjusting the parameter `opt_n` in Figure 4, right. Lower `opt_n` yields more diverse

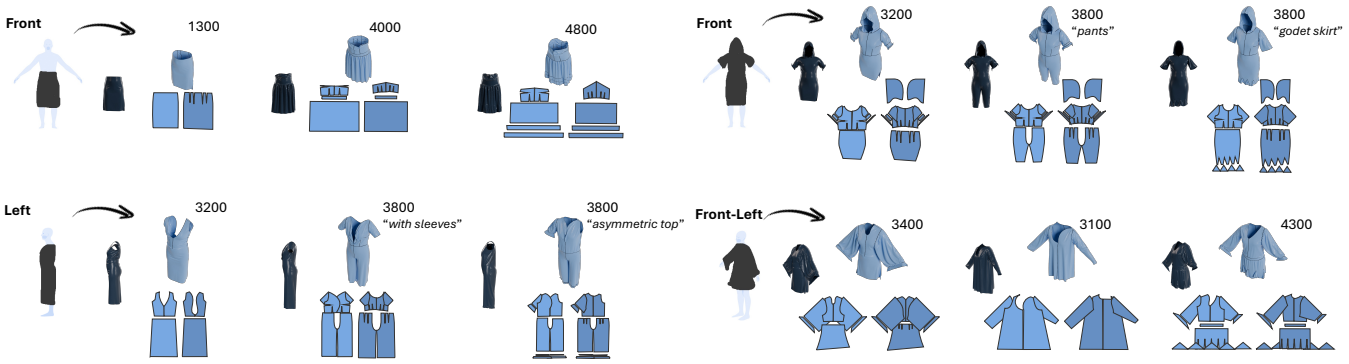

Fig. 9. **Silhouette Conditioned Garment Generation.** The user paints 2D projection to guide the garment generation process using our 2D user interface. The user can control the complexity of the generated garments as the number of points changes. The numbers indicate the number of garment particles used.

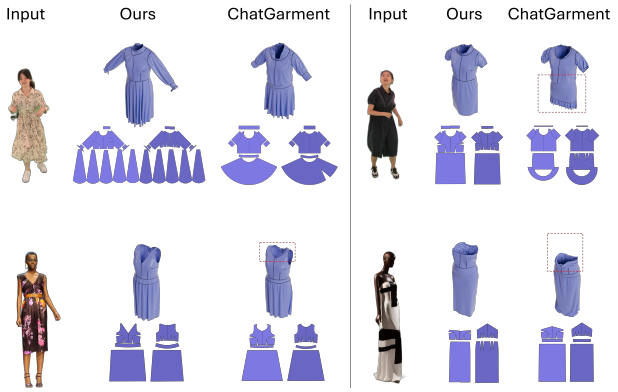

Fig. 10. **In-the-wild Image-conditioned Garment Generation.** Our method can generate more plausible sewing patterns than ChatGarment that match the garment style displayed in the in-the-wild images.

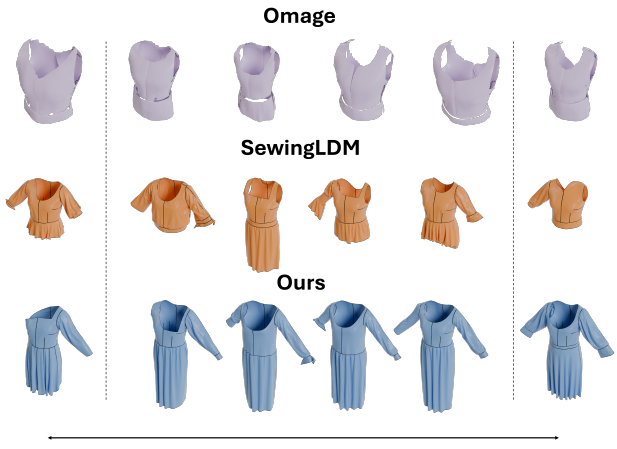

Fig. 11. **Garment Interpolation Comparison.** The interpolation results of baselines exhibit non-intuitive transitions and abrupt style changes. Our results show smooth variations in the 3D geometry that transition between different garment styles.

output, at the expense of fidelity to the input. On the other hand, a higher value of `opt_n` generates faithful samples, but the optimization constrains the sampling to more unimodal samples.

6.2.4 Multi-step Garment Editing Session with GPF. Figure 12 illustrates a multi-step garment editing session using different tools from our model. Given an in-the-wild image, we obtain an initial garment using GPF. We edit its 2D sewing pattern to enlarge the skirt, creating wrinkles in the garment geometry that are difficult to specify in 3D. Next, we use only the bottom part of the garment for 3D conditioned garment generation. Finally, we edited the front silhouette to enlarge the skirt and generate the final garment. We further constrain the generated garment style with a text prompt.

6.2.5 Fabrication Results. We hire a tailor and fabricate two garments, an asymmetric pencil skirt and a turtle-necked shirt, generated unconditionally from our pipeline in Figure 13.

7 Limitations and Future Work

Garment particles is a 5D point cloud representation that is a discrete sampling of a 3D surface and a 2D sewing pattern; therefore, it is difficult to represent continuous 3D surfaces and precise 2D pattern boundaries within a single-stage pipeline. The model is not suitable for fine-grained adjustments, such as the size of a dart, due to the limited particle resolution. In addition, our GPF requires the number of points as input, which may necessitate an additional mechanism, such as predicting an appropriate number of points for a given query. During iterative editing, the garment particles are resampled from the model, so fine details are not precisely preserved.

Editing via DPS is time-consuming, as users must wait before seeing results. For interactive applications, it is desirable to instantly obtain results to support direct manipulation, such as updates during mouse dragging. In addition, a garment defined by a 2D pattern can exhibit different 3D geometries depending on body size, posture, and fabric properties. Extending our 5D model to account for such variations is a promising direction for future work. Lastly, our model is trained only on garments produced using GarmentCode [Korosteleva and

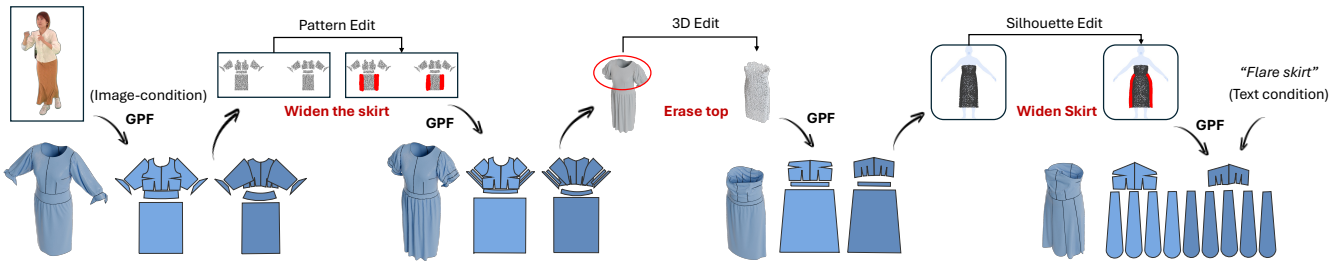

Fig. 12. **Multi-step Garment Editing Session.** We show a garment editing sequence combining various editing methods enabled by GPF.

Fig. 13. **Fabrication Examples.** We fabricated generated sewing patterns.

Sorkine-Hornung 2023]. We aim to expand our training dataset to include a wider variety of garments, such as those from [Li et al. 2025b].

Finally, we cannot guarantee that the generated sewing pattern exactly matches the input garment particles since PPF is purely data-driven. Future work includes enforcing hard constraints, such as preventing interior particles from leaving the regions of 2D panels.

8 Conclusion

We present Garment Particles, a 2D–3D symmetric garment representation that jointly encodes the sewing pattern and its draped garment geometry as a 5D point cloud. Using garment particles, we train Garment Particles Flow (GPF), a flow-based generative framework that learns a semantically rich prior space that enables state-of-the-art garment generation. More importantly, GPF naturally supports garment editing applications through diffusion posterior sampling with various objectives, including sewing pattern editing, as well as point-cloud- and silhouette-conditioned garment generation. To recover simulation-ready sewing patterns, we propose Particles-to-Pattern flow to convert the garment particles to vectorized sewing patterns.

Acknowledgments

This research is supported by JST ASPIRE, JPMJAP2401, Initiative on Recommendation Program for Young Researchers and Woman Researchers, Information Technology Center, The University of Tokyo, LVMH, Google, and the National Science Foundation Graduate Research Fellowship Program.

References

Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-driven pattern adjustment for direct 3D garment editing. *ACM Trans. Graph.* 35, 4, Article 50 (July 2016), 11 pages. doi:10.1145/2897824.2925896

Siyuan Bian, Chenghao Xu, Yuliang Xiu, Artur Grigorev, Zhen Liu, Cewu Lu, Michael J Black, and Yao Feng. 2025. ChatGarment: Garment Estimation, Generation and Editing via Large Language Models. (2025).

Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design preserving garment transfer. *ACM Trans. Graph.* 31, 4, Article 36 (July 2012), 11 pages. doi:10.1145/2185520.2185532

Selim Emir Can, Jan Ackermann, Kiyohiro Nakayama, Ruofan Liu, Tong Wu, Yang Zheng, Hugo Bertiche, Menglei Chai, Thabo Beeler, and Gordon Wetzstein. 2026. Image2Garment: Simulation-ready Garment Generation from a Single Image. *arXiv preprint arXiv:2601.09658* (2026).

Hsiao-Yu Chen, Egor Larionov, Ladislav Kavan, Gene Lin, Doug Roble, Olga Sorkine-Hornung, and Tuur Stuyck. 2025. Dress Anyone: Automatic Physically-Based Garment Pattern Refitting. *Proc. ACM Comput. Graph. Interact. Tech.* 8, 4, Article 56 (Aug. 2025), 17 pages. doi:10.1145/3747858

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. 2023. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *The Eleventh International Conference on Learning Representations.* https://openreview.net/forum?id=OnD9zGAGT0k

Hein Daanen and Sung-Ae Hong. 2008. Made-to-measure pattern development based on 3D whole body scans. *International Journal of Clothing Science and Technology* 20 (01 2008), 15–25. doi:10.1108/09556220810843502

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. arXiv:2205.14135 [cs.LG] https://arxiv.org/abs/2205.14135

Charles de Malefette, Anran Qi, Amal Dev Parakkat, Marie-Paule Cani, and Takeo Igarashi. 2023. PerfectDart: Automatic Dart Design for Garment Fitting. In *SIGGRAPH Asia 2023 Technical Communications* (Sydney, NSW, Australia) *(SA '23).* Association for Computing Machinery, New York, NY, USA, Article 7, 4 pages. doi:10.1145/3610543.3626154

A. M. Eggler, R. Falque, M. Liu, T. Vidal-Calleja, O. Sorkine-Hornung, and N. Pietroni. 2024. Digital Garment Alteration. *Computer Graphics Forum* 43, 7 (2024), e15248. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.15248 doi:10.1111/cgf.15248

Slava Elizarov, Ciara Rowles, and Simon Donné. 2024. Geometry image diffusion: Fast and data-efficient text-to-3d with image-based surface representation. *arXiv preprint arXiv:2409.03718* (2024).

CLO Virtual Fashion. 2009. *CLO 3D – 3D fashion design software.* https://www.clo3d.com/ Accessed: 2025-10-07.

Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. 2002. Geometry images. *ACM Trans. Graph.* 21, 3 (July 2002), 355–361. doi:10.1145/566654.566589

Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. 2024b. DressCode: Autoregressively Sewing and Generating Garments from Text Guidance. arXiv:2401.16465 [cs.CV] https://arxiv.org/abs/2401.16465

Qingdong He, Jinlong Peng, Pengcheng Xu, Boyuan Jiang, Xiaobin Hu, Donghao Luo, Yong Liu, Yabiao Wang, Chengjie Wang, Xiangtai Li, et al. 2024a. Dynamiccontrol: Adaptive condition selection for improved text-to-image generation. *arXiv preprint arXiv:2412.03255* (2024).

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. *ArXiv* abs/2104.08718 (2021). https://api.semanticscholar.org/CorpusID:233296711

Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2008. Automatically adding seam allowance to cloth pattern. In *ACM SIGGRAPH 2008 Posters* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 15, 1 pages. doi:10.1145/1400885.1400902

Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and Serge Belongie. 2020. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. In *European conference on computer vision*. Springer, 316–332.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. 2025. FlowDPS: Flow-Driven Posterior Sampling for Inverse Problems. arXiv:2503.08136 [cs.CV] https://arxiv.org/abs/2503.08136

Maria Korosteleva, Timur Levent Kesdogan, Fabian Kemper, Stephan Wenninger, Jasmin Koller, Yuhan Zhang, Mario Botsch, and Olga Sorkine-Hornung. 2024. GarmentCodeData: A Dataset of 3D Made-to-Measure Garments With Sewing Patterns. arXiv:2405.17609 [cs.CV] https://arxiv.org/abs/2405.17609

Maria Korosteleva and Sung-Hee Lee. 2022. NeuralTailor: Reconstructing Sewing Pattern Structures from 3D Point Clouds of Garments. *ACM Trans. Graph.* 41, 4 (2022), 16 pages. doi:10.1145/3528223.3530179

Maria Korosteleva and Olga Sorkine-Hornung. 2023. GarmentCode: Programming Parametric Sewing Patterns. *ACM Transaction on Graphics* 42, 6 (2023), 16 pages. doi:10.1145/3618351 SIGGRAPH ASIA 2023 issue.

Mingi Lee, Dongsu Zhang, Clément Jambon, and Young Min Kim. 2025. BrepDiff: Single-Stage B-rep Diffusion Model. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers* (Vancouver, BC, Canada) *(SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 103, 11 pages. doi:10.1145/3721238.3730698

Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018. Foldsketch: enriching garments with physically reproducible folds. *ACM Trans. Graph.* 37, 4, Article 133 (July 2018), 13 pages. doi:10.1145/3197517.3201310

Ren Li, Cong Cao, Corentin Dumery, Yingxuan You, Hao Li, and Pascal Fua. 2025a. Single View Garment Reconstruction Using Diffusion Mapping Via Pattern Coordinates. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 47, 11 pages. doi:10.1145/3721238.3730651

Ren Li, Corentin Dumery, Benoit Guillard, and Pascal Fua. 2024a. Garment Recovery with Shape and Deformation Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ren Li, Benoit Guillard, and Pascal Fua. 2023. ISP: Multi-Layered Garment Draping with Implicit Sewing Patterns. In *Advances in Neural Information Processing Systems*.

Shudong Li, Xiao Jiang, Matthew Tivnan, Grace J Gang, Yuan Shen, and J Webster Stayman. 2024b. CT reconstruction using diffusion posterior sampling conditioned on a nonlinear measurement model. *Journal of Medical Imaging* 11, 4 (2024), 043504–043504.

Siran Li, Ruiyang Liu, Chen Liu, Zhendong Wang, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. 2025b. GarmageNet: A Multimodal Generative Framework for Sewing Pattern Design and Generic Garment Modeling. *ACM Trans. Graph.* 44, 6, Article 216 (Dec. 2025), 23 pages. doi:10.1145/3763271

Xinyu Li, Qi Yao, and Yuanda Wang. 2025c. GarmentDiffusion: 3D garment sewing pattern generation with multimodal diffusion transformers. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence* (Montreal, Canada) *(IJCAI '25)*. Article 163, 9 pages. doi:10.24963/ijcai.2025/163

Xuan Li, Chang Yu, Wenxin Du, Ying Jiang, Tianyi Xie, Yunuo Chen, Yin Yang, and Chenfanfu Jiang. 2025d. Dress-1-to-3: Single Image to Simulation-Ready 3D Outfit with Diffusion Prior and Differentiable Physics. *ACM Trans. Graph.* 44, 4, Article 71 (July 2025), 16 pages. doi:10.1145/3731177

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. 2025b. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470* (2025).

Kaixuan Liu, Xianyi Zeng, Pascal Bruniaux, Xuyuan Tao, Xiaofeng Yao, Victoria Li, and Jianping Wang. 2018. 3D interactive garment pattern-making technology. *Computer-Aided Design* 104 (2018), 113–124. doi:10.1016/j.cad.2018.07.003

Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. 2023. Towards Garment Sewing Pattern Reconstruction from a Single Image. *ACM Transactions on Graphics (SIGGRAPH Asia)* (2023).

Shengqi Liu, Yuhao Cheng, Zhuo Chen, Xingyu Ren, Wenhan Zhu, Lincheng Li, Mengxiao Bi, Xiaokang Yang, and Yichao Yan. 2025a. Multimodal Latent Diffusion Model for Complex Sewing Pattern Generation. *International Conference on Computer Vision (ICCV)* (2025).

Yilin Liu, Duoteng Xu, Xingyao Yu, Xiang Xu, Daniel Cohen-Or, Hao Zhang, and Hui Huang. 2025c. Hola: B-rep generation using a holistic latent representation. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–25.

Yuwei Meng, Charlie C.L. Wang, and Xiaogang Jin. 2012a. Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design* 44, 1 (2012), 68–76. doi:10.1016/j.cad.2010.11.008 Digital Human Modeling in Product Design.

Yuwei Meng, Charlie C. L. Wang, and Xiaogang Jin. 2012b. Flexible shape control for automatic resizing of apparel products. *Comput. Aided Des.* 44, 1 (Jan. 2012), 68–76. doi:10.1016/j.cad.2010.11.008

Kiyohiro Nakayama, Jan Ackermann, Timur Levent Kesdogan, Yang Zheng, Maria Korosteleva, Olga Sorkine-Hornung, Leonidas Guibas, Guandao Yang, and Gordon Wetzstein. 2025. AIpparel: A Multimodal Foundation Model for Digital Garments. *Computer Vision and Pattern Recognition (CVPR)* (2025).

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).

Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. 2024. Steering Rectified Flow Models in the Vector Field for Controlled Image Generation. *arXiv preprint arXiv:2412.00100* (2024).

William Peebles and Saining Xie. 2023. Scalable Diffusion Models with Transformers. arXiv:2212.09748 [cs.CV] https://arxiv.org/abs/2212.09748

Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa A Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational pattern making from 3D garment models. *ACM Trans. Graph.* 41, 4 (2022), 157–1.

Anran Qi and Takeo Igarashi. 2024. PerfectTailor: Scale-Preserving 2-D Pattern Adjustment Driven by 3-D Garment Editing. *IEEE Comput. Graph. Appl.* 44, 4 (July 2024), 126–132. doi:10.1109/MCG.2024.3378171

ANRAN QI, NICO PIETRONI, MARIA KOROSTELEVA, OLGA SORKINE-HORNUNG, and ADRIEN BOUSSEAU. 2025. Rags2Riches: Computational Garment Reuse. In *SIGGRAPH 2025 Conference Paper*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV] https://arxiv.org/abs/2103.00020

Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. 2023. Trace and Pace: Controllable Pedestrian Animation via Guided Trajectory Diffusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alejandro Rodríguez and Gabriel Cirio. 2022. True seams: modeling seams in digital garments. *ACM Trans. Graph.* 41, 4, Article 62 (July 2022), 16 pages. doi:10.1145/3528223.3530128

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *Int. J. Comput. Vision* 40, 2 (Nov. 2000), 99–121. doi:10.1023/A:1026543900054

Nicholas Sharp and Keenan Crane. 2018. Variational Surface Cutting. *ACM Trans. Graph.* 37, 4 (2018).

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 (CVPR)*. 19615–19625.

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing inflatable structures. *ACM Trans. Graph.* 33, 4, Article 63 (July 2014), 10 pages. doi:10.1145/2601097.2601166

Style3D. 2015. *Style3D*. https://www.style3d.com/ Accessed: 2025-10-07.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063.

Zhuo Su, Jiehua Zhang, Longguang Wang, Hua Zhang, Zhen Liu, Matti Pietikäinen, and Li Liu. 2023. Lightweight Pixel Difference Networks for Efficient Visual Representation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 12 (2023), 14956–14974. doi:10.1109/TPAMI.2023.3300513

Yuki Tatsukawa, Anran Qi, I-Chao Shen, and Takeo Igarashi. 2025. GarmentImage: Raster Encoding of Garment Sewing Patterns with Diverse Topologies. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 46, 11 pages. doi:10.1145/3721238.3730632

Mohammad H Taufik and Tariq Alkhalifah. 2025. Diffusion Model-Based Posterior Sampling in Full Waveform Inversion. *arXiv preprint arXiv:2512.12797* (2025).

Hao Tian, Yu Cao, and P. Y. Mok. 2025. SewPCT: Sewing Pattern Reconstruction from Point Cloud with Transformer. In *Advances in Computer Graphics: 41st Computer Graphics International Conference, CGI 2024, Geneva, Switzerland, July 1–5, 2024,*

Proceedings, Part II (Geneva, Switzerland). Springer-Verlag, Berlin, Heidelberg, 211–223. doi:10.1007/978-3-031-82021-2_15

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 90, 12 pages. doi:10.1145/1964921.1964985

Haoxuan Wang, Jinlong Peng, Qingdong He, Hao Yang, Ying Jin, Jiafu Wu, Xiaobin Hu, Yanjie Pan, Zhenye Gan, Mingmin Chi, et al. 2025. Unicombine: Unified multi-conditional combination with diffusion transformer. *arXiv preprint arXiv:2503.09277* (2025).

Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popović, and Niloy J. Mitra. 2018. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.* 37, 6, Article 203 (Dec. 2018), 13 pages. doi:10.1145/3272127.3275074

Wenbo Wang, Hsuan-I Ho, Chen Guo, Boxiang Rong, Artur Grigorev, Jie Song, Juan Jose Zarate, and Otmar Hilliges. 2024. 4D-DRESS: A 4D Dataset of Real-world Human Clothing with Semantic Annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Katja Wolff, Philipp Herholz, Verena Ziegler, Frauke Link, Nico Brügel, and Olga Sorkine-Hornung. 2021. Designing Personalized Garments with Body Movement. *Computer Graphics Forum* 42 (2021). https://api.semanticscholar.org/CorpusID:252383414

Katja Wolff and Olga Sorkine-Hornung. 2019. Wallpaper pattern alignment along garment seams. *ACM Trans. Graph.* 38, 4, Article 62 (July 2019), 12 pages. doi:10.1145/3306346.3322991

Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. 2024. GPT-4V(ision) is a Human-Aligned Evaluator for Text-to-3D Generation. In *CVPR*.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506* (2024).

Xiang Xu, Joseph G. Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl D. D. Willis, and Yasutaka Furukawa. 2024. BrepGen: A B-rep Generative Diffusion Model with Structured Latent Geometry. arXiv:2401.15563 [cs.CV] https://arxiv.org/abs/2401.15563

Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X. Chang. 2024. An Object is Worth 64x64 Pixels: Generating 3D Object via Image Diffusion. arXiv:2408.03178 [cs.CV] https://arxiv.org/abs/2408.03178

Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C. Lin. 2018. Physics-Inspired Garment Recovery from a Single-View Image. *ACM Trans. Graph.* 37, 5, Article 170 (nov 2018), 14 pages. doi:10.1145/3026479

Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. 2024. Fasterdit: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems* 37 (2024), 56166–56189.

Jingfeng Yao, Bin Yang, and Xinggang Wang. 2025. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. (2023).

Zeya Yin, Tin Lai, Lucas Barcelos, Jayadeep Jacob, Yonghui Li, and Fabio Ramos. 2025. Diverse Motion Planning with Stein Diffusion Trajectory Inference. *2025 IEEE International Conference on Robotics and Automation (ICRA)* (2025), 15610–15616. https://api.semanticscholar.org/CorpusID:281093556

Jingdong Zhang, Weikai Chen, Yuan Liu, Jionghao Wang, Zhengming Yu, Zhuowen Shen, Bo Yang, Wenping Wang, and Xin Li. 2025. SPGen: Spherical Projection as Consistent and Flexible Representation for Single Image 3D Shape Generation. *arXiv preprint arXiv:2509.12721* (2025).

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024b. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Ruisi Zhang, Tianyu Liu, Will Feng, Andrew Gu, Sanket Purandare, Wanchao Liang, and Francisco Massa. 2024a. SimpleFSDP: Simpler Fully Sharded Data Parallel with torch.compile. arXiv:2411.00284 [cs.DC] https://arxiv.org/abs/2411.00284

Xiaoting Zhang, Guoxin Fang, Melina Skouras, Gwenda Gieseler, Charlie C. L. Wang, and Emily Whiting. 2019. Computational design of fabric formwork. *ACM Trans. Graph.* 38, 4, Article 109 (July 2019), 13 pages. doi:10.1145/3306346.3322988

Feng Zhou, Ruiyang Liu, Chen Liu, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. 2024. Design2GarmentCode: Turning Design Concepts to Tangible Garments Through Program Synthesis. *arXiv preprint arXiv:2412.08603* (2024).

Supplementary Material for Garment Particles: A 2D–3D Symmetric Garment Representation for Generation and Editing

Table of Contents

A Implementation, Dataset, and Metrics Details

A.1 Garment Particles Construction Details

We construct our garment particles datasets on GarmentCodeDatav2 (GCDv2) [Korosteleva et al. 2024]. Specifically, we first separate the front and back panels using the provided panel names. The front and panel panels are packed using the algorithm outlined in Section 3.1. To resolve panel overlaps, we use an iterative algorithm that pushes overlapping panels apart based on the distance and direction between their centers. Additionally, to ensure greater consistency across garments, we define a tree structure over the panels based on their semantic relationships. The repulsion algorithm is then applied first between siblings and then between children and parents. To ensure robustness, we pad each panel by 5 centimeters around its boundary when checking for overlaps. The repulsion algorithm will run until there are no overlaps between any panels or it reaches 500 steps. After constructing the garment particles, we filter out data samples whose sewing pattern U exceeds the bounding box $[-150,150]\times[-80,220]$. In total, we obtain 120k valid examples out of 130k garments in GCDv2.

A.2 GPF Training Details

To train the text-conditioned GPF model, we use 32 × NVIDIA H100 GPUs for a total of 210,000 iterations. To speed up training, we use Pytorch FSDP2 [Zhang et al. 2024a] and Flash Attention [Dao et al. 2022]. We use a dynamic batch sampler that distributes roughly equal numbers of tokens across GPUs, resulting in an average batch size of 250. We use a learning rate of 0.0001 with gradient clipping at 1. The training takes around 1.5 days.

A.3 Text Caption Dataset Construction

We construct a new text caption dataset to train our text-conditioned GPF model. We procedurally generate the text captions from the design parameters given in the GCDv2 dataset. Each text prompt consists of a set of short, keyword phrases, describing the make of different components of the garments (e.g., "with" v.s. "without sleeves", "fitted" v.s. "loose shirt", "pants" v.s. "skirts", etc.). Compared with existing text-prompt datasets, such as GCD-MM [Nakayama et al. 2025], our curated caption focuses more on garment style and is therefore better suited for integration with garment editing applications. During training, we sample a subset of these keywords and combine them into a single input for the GPF model.

A.4 Particles-to-Pattern Flow Training Details

We use the LightningDiT-L variant as our Flow architecture, which consists of around $500M$ trainable parameters. We apply both panel and edge-level positional embedding during training. In notations, we have

$$\boldsymbol{v}_{\varphi}(\mathcal{P}_t,t;\boldsymbol{X}) = \mathrm{DiT}\big(\mathrm{Flatten}(\boldsymbol{W}_{pos}+\mathcal{P}_t),t;\boldsymbol{X}\big) \tag{10}$$

where $\boldsymbol{W}_{pos}=\boldsymbol{W}_{panel}\oplus\boldsymbol{W}_{edge}$ is the outer sum of panel embeddings and edge embeddings, and the Flatten operator flattens the panel and edge dimension. We additionally apply RoPE [Su et al. 2024] to enhance the positional information. We use cross-attention to inform the network about $\boldsymbol{X}$. We use the same training hyperparameters as GPF model training on 16× NVIDIA H100 GPUs for a total of 160k iterations.

A.5 Objective Guided Editing Hyperparameters

We found that different sets of hyperparameters, such as the sampling steps T, `stop_t`, learning rate η, and optimization steps `opt_n`, work well for different objectives and applications. However, in general, for EMD loss, we use $\eta\in[0.02,0.1]$, `opt_n` $\in\{1,2,3,4\}$, $T\in[250,1000]$, and `stop_t` $\in[0,3,0.8]$. For Chamfer Distance as the objective, we keep the same T and `stop_t`, but increase `opt_n` to 10 and decrease η to between $[0.01,0.02]$.

A.6 Generation Metrics

We use a different set of metrics to measure the diversity and realism of our generated garments relative to the ground-truth garments in the test set. Specifically, they measure the following.

(1) *Generation diversity and distribution* using the Coverage (COV) score, Minimum Matching Distance (MMD), and 1-Nearest Neighbor classification Accuracy (1-NNA). Following [Li et al. 2025b; Zhang et al. 2024b], we also report pointcloud-FID (p-FID) to assess feature-embedding similarity against the reference set.
(2) *Draping quality* using simulation-success rate (SSR) of the generated garments. For all our baselines, we use GCDv2's provided draping simulator.
(3) *The alignment between generated garment and input text prompts* using CLIPScore [Hessel et al. 2021].

We define each of the metrics below.

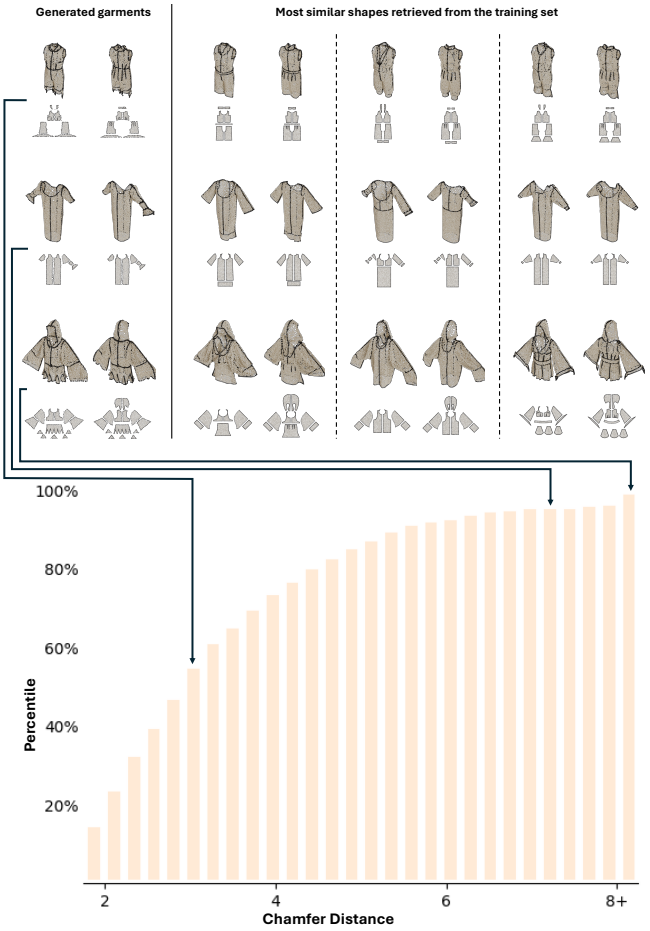

Fig. 14. **Closest Query Visualization**. *(Top)* We visualize the top three nearest neighbors in the training set to our generated garment particles (leftmost column). Our GPF model can generate novel garments with a distinct style compared to the training set. *(Bottom)* We plot the distance of our generated sets to the training set as a cumulative plot. The arrows indicate the bins to which each of the visualized garments belongs.

Coverage Score (COV). Coverage Score measures the percentage of reference garments that are matched by at least one generated sample. Given the set of generated garments S_g and reference garments S_r, we compute the score as

$$\mathrm{COV}(S_g,S_r)=\frac{100}{|S_r|}\left|\left\{\arg\min_{y\in S_r} D(x,y)\,|\,x\in S_g\right\}\right|, \tag{11}$$

where $D(\cdot,\cdot)$ denotes the Chamfer distance between two point clouds. A higher score indicates a more diverse set of generated samples that better covers the reference distribution. We uniformly sample 8,192 points on each garment mesh surface for computation.

Minimum Matching Distance (MMD). Minimum Matching Distance measures the fidelity of the generated set by computing the average distance from each reference sample to its closest generated sample. Given the set of generated garments S_g and reference garments S_r, we compute the score as

$$\mathrm{MMD}(S_g,S_r)=\frac{1}{|S_r|}\sum_{y\in S_r}\min_{x\in S_g} D(x,y), \tag{12}$$

where $D(\cdot,\cdot)$ denotes the Chamfer distance between two point clouds. A lower MMD indicates that the generated samples are closer to the reference distribution. We uniformly sample 8,192 points on each garment mesh surface for computation.

1-Nearest Neighbor Accuracy (1-NNA). The 1-Nearest Neighbor Accuracy evaluates whether the generated and reference samples are distinguishable by a nearest neighbor classifier. For each sample in $S_g\cup S_r$, we find its nearest neighbor (excluding itself) and check whether they belong to the same set:

$$\text{1-NNA}(S_g,S_r)=100\times\frac{\sum_{x\in S_g}\mathbf{1}[\mathrm{NN}(x)\in S_g]+\sum_{y\in S_r}\mathbf{1}[\mathrm{NN}(y)\in S_r]}{|S_g|+|S_r|}, \tag{13}$$

where $\mathrm{NN}(\cdot)$ returns the nearest neighbor in $S_g\cup S_r$ using Chamfer distance. An ideal generative model produces samples indistinguishable from the reference set, yielding a 1-NNA close to 50%.

Pointcloud FID (p-FID). Pointcloud FID measures the similarity between the feature distributions of generated and reference point clouds. We first extract features using a pretrained point cloud encoder, then compute the Fréchet distance between the two Gaussian-fitted distributions:

$$\text{p-FID}=\|\boldsymbol{\mu}_r-\boldsymbol{\mu}_g\|_2^2+\mathrm{Tr}\left(\Sigma_r+\Sigma_g-2\left(\Sigma_r\Sigma_g\right)^{1/2}\right), \tag{14}$$

where $(\boldsymbol{\mu}_g,\Sigma_g)$ and $(\boldsymbol{\mu}_r,\Sigma_r)$ are the mean and covariance of the generated and reference feature embeddings, respectively. A lower p-FID indicates greater similarity to the reference distribution.

Simulation Success Rate (SSR). Simulation Success Rate measures the physical plausibility of generated garments by evaluating whether they can be successfully draped on a human body without simulation failures (e.g.,, mesh interpenetration or divergence):

$$\mathrm{SSR}=\frac{N_{\text{success}}}{N_{\text{total}}}\times 100, \tag{15}$$

where N_{success} is the number of garments that complete the draping simulation without errors, and N_{total} is the total number of generated garments. A higher SSR indicates that the model generates more physically valid garment geometries.

CLIPScore. CLIPScore measures the semantic alignment between generated garments and input text prompts. For each successfully draped garment, we render $|V|=20$ views uniformly distributed along the equator and compute the average cosine similarity between image and text embeddings:

$$\mathrm{CLIPScore}=\frac{100}{|V|}\sum_{v\in V}\cos(\mathbf{E}_I(I_v),\mathbf{E}_T(t)), \tag{16}$$

where $\mathbf{E}_I$ and $\mathbf{E}_T$ are the CLIP image and text encoders, I_v is the rendered image from view v, and t is the input text prompt. A higher CLIPScore indicates better alignment between text and garment.

A.7 Sewing Pattern Metrics

To evaluate the quality of the generated sewing pattern relative to the ground truth, we use the sewing pattern metrics defined in prior

work [Li et al. 2025b; Nakayama et al. 2025]. Specifically, we use (1) panel accuracy (Panel Acc): the percentage of garments with the correct number of panels, (2) panel-wise intersection-over-union (Panel IOU): the average IOU between the generated and ground-truth panels, and (4) stitch accuracy (Stitch Acc): the percentage of correctly predicted stitching pairs. We also measure the simulation success rate (SSR) and the 3D Chamfer Distance (CD) of the draped garments.

A.8 Extended Discussion on Limitations and Future Works

Garment Particle Flow. While our GPF module learned the implicit consistency between the 2D sewing pattern 3D garment geometry from data, we do not gurantee hard constraints on this front. For example, manufaturing constraints such as developability of the 3D surface and near-isometry constraint between the pattern and the draped garment geometry. Enforcing these constraints either directly during training, or via inference-time scaling are interesting future work directions.

Additionally, our GPF model is trained on GarmentCodeData. While being the largest sewing pattern dataset online, they are synthetic and lack important components such as pockets, frills. Another future work direction is to integrate more realistic sewing pattern datasets (e.g., GarmageSet [Li et al. 2025b]) or potentially in-the-wild sewing patterns to the training pipeline.

Lastly, our GPF model is only trained on a single body type with the same pose. This limits our applications to be performed on the same human. One could extend our GPF to handle multi human body + human pose input, and extending our model for garment refitting applications.

Particle-to-Pattern Flow. Similar to GPF, our PPF model also learns the sewing pattern reconstruction purely from data, with a conditional generative model. While this approach make PPF more robust to noisy garment particle inputs, we cannot guarantee that the reconstructed garment is strictly consistency with respect to the input. One promising future direction is the improve the consistency via post-training strategies such as Flow-GRPO [Liu et al. 2025b], with non-differentiable rewards such as IOUs and accuracies as rewards.

Additionally, our sewing pattern representation still use the format popularized by GarmentCodeData [Korosteleva et al. 2024], which uses one-to-one stiches and rigid transformation based panel initializations. Recent works such as GarmageNet extended the representation to use point-to-point stitching and point-wise panel initialization, enabling more complext garment modeling capabilities. It would be interesting to extend our PPF module to allow these more flexible sewing pattern representations, therefore enabling more complext garment generation and editing.

B User Interfaces

To facilitate the use of GPF for various editing tasks in both 2D and 3D design spaces, we developed two interfaces as follows.

B.1 3D Interface

We developed a 3D user interface for interactive garment editing in augmented reality (AR), in which garment geometry is represented as a point cloud. Within the immersive environment, we designed and implemented a set of editing tools that enable users to directly manipulate the point cloud with controllers, including point addition, deletion, and translation. The system was implemented in *Unity*, a cross-platform 3D game engine, using the built-in render pipeline. *Meta Quest 3* was leveraged as the head-mounted display (HMD) to present the AR environment and support user interaction during garment editing. An overview of the 3D interactions is illustrated in Figure 15.

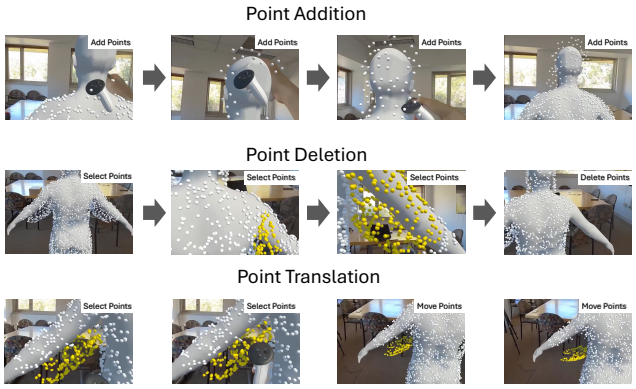

Fig. 15. **3D Interface Illustration**. Our 3D interface allows users to directly manipulate 3D geometry with operations such as point addition, deletion, and translation.

Addition. Users can add points by pressing the *A* button on the right controller, which inserts new points in the neighborhood of the controller's current position.

Deletion. To delete points, users first press the *Index Trigger* on the right controller to select points near its position, which are visually highlighted in yellow. Pressing the *B* button subsequently removes the selected points.

Translation. Point translation follows a procedure similar to deletion. Users first select the target points and then manipulate the *Joystick* on the right controller to move the selected points in a non-linear manner.

B.2 2D Interface

We implemented a 2D user interface that supports direct editing of sewing patterns and silhouettes from arbitrary camera viewpoints, as illustrated in Figure 16. Editing operations are implemented through a canvas-based tool that allows users to paint or erase on both the silhouette and pattern canvases. The edited canvas is subsequently converted into a 2D point cloud via area sampling and forwarded to GPF for various guided generation tasks.

C Additional Results

C.1 Ablation Study on Particles-to-Pattern Flow

We conduct an ablation study to evaluate the effectiveness of our particles-to-pattern flow module relative to alternative architectural choices. Specifically, we compare our *flow*-based network with two variants. *Regression*-based variant uses a feedforward network to directly predict the vectorized sewing pattern given garment particles.

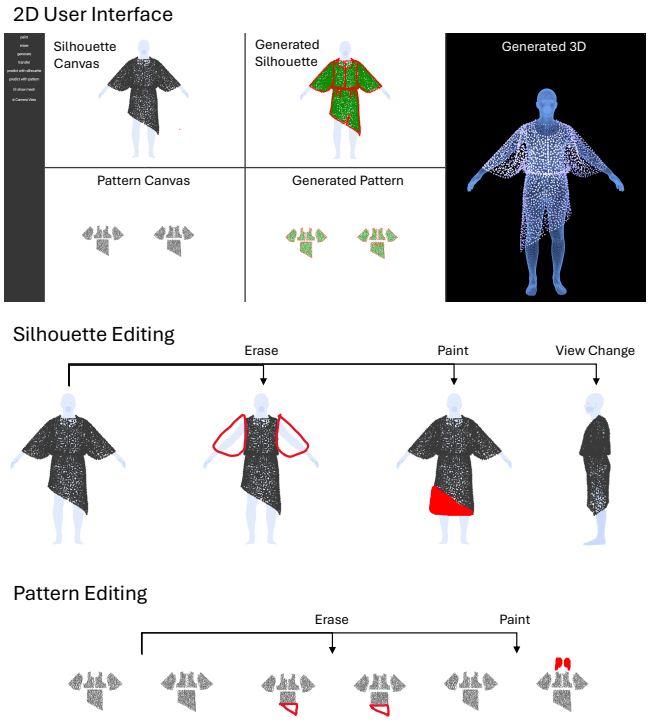

Fig. 16. **2D Interface Illustration**. Our 2D interface supports silhouette and pattern editing with paintbrushes. The top row shows our interface's layout, the next two rows show different operations we allow for silhouette and pattern editing.

Method	Panel Acc	Edge Acc	Stitch Acc	Panel IOU
No Noise				
Delaunay	**0.9912**	–	–	0.9034
Regression	0.9902	0.9177	0.6898	0.6727
Flow	0.9775	**0.9430**	**0.9333**	**0.9181**
+ 1% Noise				
Delaunay	0.0947	–	–	0.6688
Regression	**0.9902**	0.9074	0.6829	0.6717
Flow	0.9365	**0.8932**	**0.8770**	**0.8815**
+ 2% Noise				
Delaunay	0.0156	–	–	0.4327
Regression	**0.9902**	0.8951	0.6727	0.6687
Flow	0.9355	**0.9051**	**0.8793**	**0.8822**
+ 5% Noise				
Delaunay	0.0449	–	–	0.1666
Regression	**0.9658**	0.8596	0.6379	0.6524
Flow	0.9355	**0.8848**	**0.8540**	**0.8631**
+ 10% Noise				
Delaunay	0.0088	–	–	0.1306
Regression	0.7324	0.8187	0.5583	0.5953
Flow	**0.8955**	**0.8834**	**0.8348**	**0.8210**

Table 3. **Ablation Study on Particle-to-Pattern Module**. We compare variants of the Particle-to-Pattern Module across different levels of noise added to the first two coordinates.

We use the same architecture as our particles-to-pattern flow model, but simply change the loss to mean squared error. *Delaunay* variant reconstructs the sewing pattern as a flat triangle mesh by performing a sequence of training-free operations. Specifically, the garment particles are first clustered into panels using DBSCAN based on their layout when projected into the domain via π_D. Then, each cluster is triangulated with a Delaunay triangulation to obtain a convex hull of each panel. Finally, the boundary for each panel is recovered by removing triangles whose three vertices have all three boundary flags set to positive (on the boundary).

We compare the performance on the particles-to-pattern reconstruction task, using the sewing pattern metrics as described in Section A.7. Table 3 shows the comparison with different levels of noise added to the pattern space of the particles (i.e.,first two coordinates), mimicking the noisy generation result from GPF's output. Because the *Delaunay* variant does not predict edges and stitching information, we omit their scores for these two columns. The table shows that when no noise is added, the *Delaunay* variant performs the best in terms of panel IOU and accuracy. This suggests that the classical algorithm can almost completely recover the sewing pattern shapes without precision loss *if* there is no corruption in the garment particles. However, even with 1% noise added to the pattern coordinates, its performance drops drastically, because DBSCAN (clustering) and the triangulation process are very sensitive to outliers. Comparatively, the *regression*-based variant achieves better robustness against noisy input, but still is prone to error when the amount of noise added exceeds 5%. This is because a regression-based model learns a deterministic output for each garment particles, making it susceptible to out-of-distribution data. Lastly, the *Flow*-based variant retains its overall performance across all levels of added noise. This is because its formulation of the reconstruction task as a conditional generation problem, enabling the model to still output sensible patterns when the input is corrupted.

Effect of Boundary Flag. Table 5 shows the effect of having the boundary flag as input to the PPF module, for the sewing pattern reconstruction task. The resulting improvements in panel, edge, and stitch accuracies demonstrate boundary flag's effectiveness in recovering discrete structures from garment particles.

C.2 Unconditional Generation

C.2.1 Generation Gallery. In Figure 17, we showcase a gallery of our generated garments from GPF without conditioning. The generated garment particles are converted into a draping pattern using our particles-to-pattern flow module. As shown in the figure, we can generate a diverse set of garments, from simple to complex panel layouts. This demonstrates the pipeline's generation capability.

C.2.2 Generation Novelty Analysis. We validate our method's ability to generate novel garments not included in the training dataset. Following [Siddiqui et al. 2024], we generate 1,024 garments using our

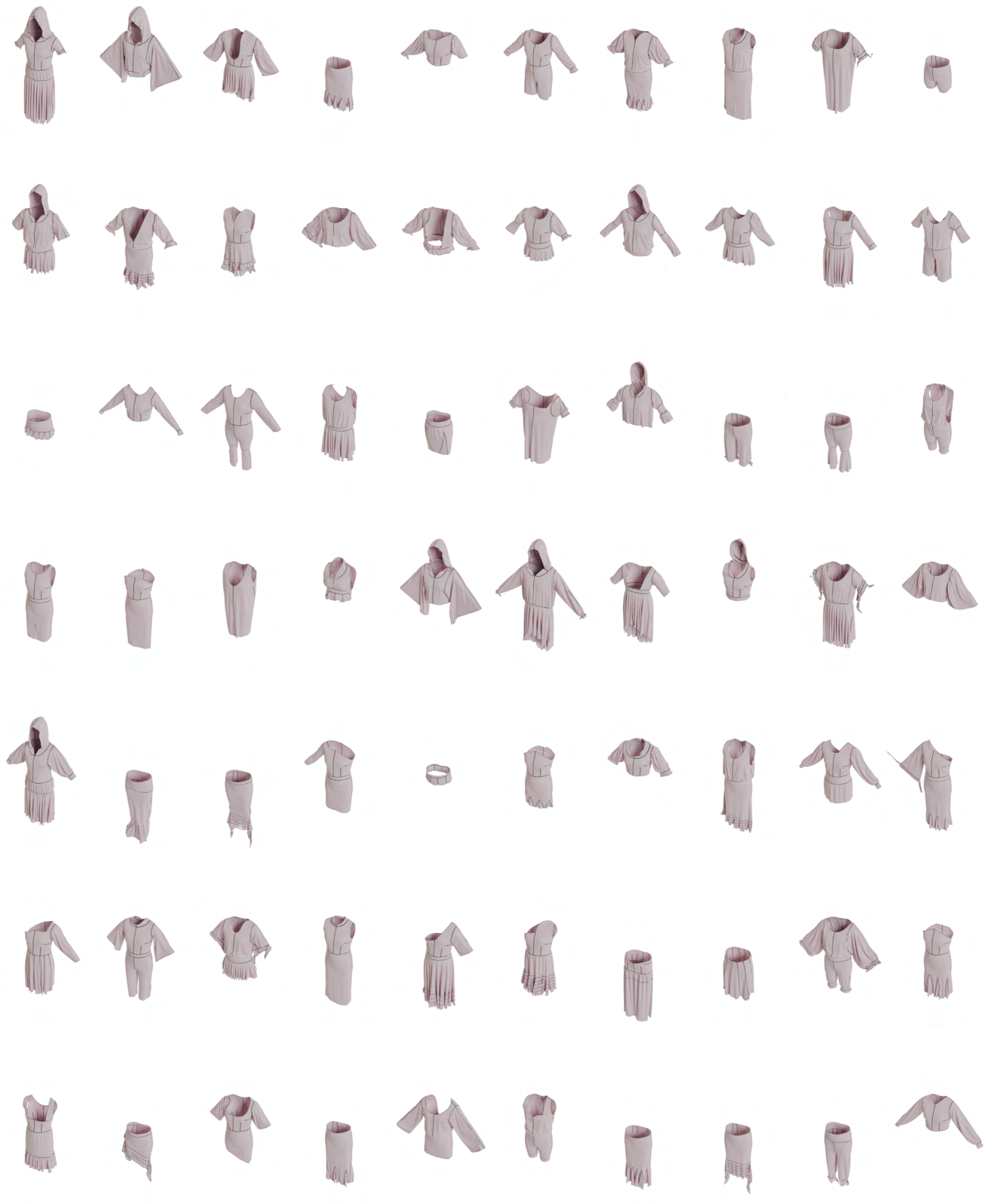

Fig. 17. **Unconditional Generation Gallery.**

Method	**Panel Acc** (%) ↑	**Panel IOU** (%) ↑	**Edge Acc** (%) ↑	**Stitch Acc** (%) ↑	**SSR** (%) ↑	**CD** ($\times 10^3$) ↓
GCDv2						
Ours-Singleview	83.01	77.54	78.20	69.47	89.84	7.0
Ours-Multiview	**85.35**	**79.63**	**80.09**	**71.49**	**90.23**	**5.5**
Garment Sketches						
Ours-Singleview	81.25	76.07	77.05	66.86	**92.97**	8.7
Ours-Multiview	**82.71**	**78.14**	**79.01**	**67.67**	88.18	**8.4**

Table 4. **Multiview-image-conditioned Garment Generation.**

Method	Panel Acc	Edge Acc	Stitch Acc	Panel IOU
PPF w/o BFlag	0.9121	0.8508	0.8485	0.9041
PPF	**0.9775**	**0.9430**	**0.9333**	**0.9181**

Table 5. **Ablation Study on Boundary Flag.**

GPF model without conditioning. In Figure 14, we show three generated garments and their top three nearest neighbors in the training set, measured using the 3D Chamfer distance. We also report the distance of this garment set to its nearest neighbor in the training set. These results show that our model can generate novel garment styles even at the 30th-percentile level (top row), demonstrating that it learns to generate unseen combinations of sewing patterns and garment geometry.

C.2.3 Text-Conditioned Generation Additional Visualization. We showcase additional comparison for text-conditioned garment generation in Figure 18. The baselines exhibit artifacts and even fail to produce the garment geometry, as shown by the red boxes. Compared with the text's description, our outputs closely match it while producing simulation-ready sewing patterns.

C.3 Image-Conditioned Generation Additional Visualization

We showcase additional comparisons of our image-conditioned GPF model against baselines on the GCDv2 and Garment Sketches datasets. The results are shown in Figure 20 and Figure 19. On both datasets, our model consistently generates garments that are better aligned with the input sketch and the ground-truth garment. On the other hand, baseline generation can include patterns that do not drape correctly (Sketch: first and second rows of SewingLDM, GCDv2: first and last rows of AIpparel), result in incorrect garment styles (Sketch: third and seventh rows of Design2GarmentCode, GCDv2: last and second to last rows of ChatGarment), or have incorrect panel shapes (Sketch: fifth row of SewingLDM, GCDv2: last row of AIpparel).

C.3.1 Extending GPF to more Modalities. We experiment with extending GPF to multiview images. We use the front and back renderings from the *GCDv2* and *Garment Sketches* datasets as conditioning for the GPF. To pass into the model, we average the DINOV2 features of the front and back images for each token. Table 4 shows the sewing pattern metric comparison against GPF with single-view input. Multiview conditioning consistently improves geometry reconstruction metrics, indicating that additional images provide useful constraints for generation.

C.3.2 Additional Garment Interpolation Results. We showcase additional results when interpolating between two generated garments from GPF in Figure 21. Our interpolation enables distinct garment style transitions (e.g.,pants-to-skirt: first row, single-sleeve-to-multi-sleeve (third, fifth rows)) and gradual size variations in different components of the garment (e.g.,larger sleeve (last row), shorter (fourth row), and longer skirts (second row)). These results demonstrate the representation effectiveness of our bidirectional garment particles representation.

C.3.3 Additional Sewing Pattern Editing Results. We showcase additional sewing pattern editing results in Figure 22. The left shows the original garment generated using GPF. The left shows garments generated after editing the input sewing pattern, with red paint indicating users' input. The results show that our generated garments closely follow users' edits to the 2D sewing pattern, while filling in missing details when the input is coarse. We optionally use text as an additional control signal to guide the generation. However, because the guiding objective is agnostic of panel boundaries, we cannot control the number of panels generated.

C.3.4 Additional Silhouette-conditioned Garment Generation Results. Figure 23 shows more results on silhouette-conditioned garment generation. The left column shows the original garment assets generated from GPF. We show three different edits on the silhouette, either erasing or adding, using our 2D user interface. The last row shows silhouette edits applied to different camera angles, including a top view where pants are converted to a skirt, and a side view where a skirt is widened and asymmetrized. We also optionally provide text as guidance to control the generated garment style. Our generated garments match the given silhouette while producing plausible draping results.

C.3.5 Additional Point-cloud-conditioned Garment Generation Results. Figure 24 shows additional point-cloud-conditioned garment generation results. The models complete the partial 3D garment point cloud shown on the left. For each row we show three variations of possible completions. The number associated with each example shows the number of points we use for GPF generation. We see that as we increase the number of points, the complexity of the generated garment generally increases.

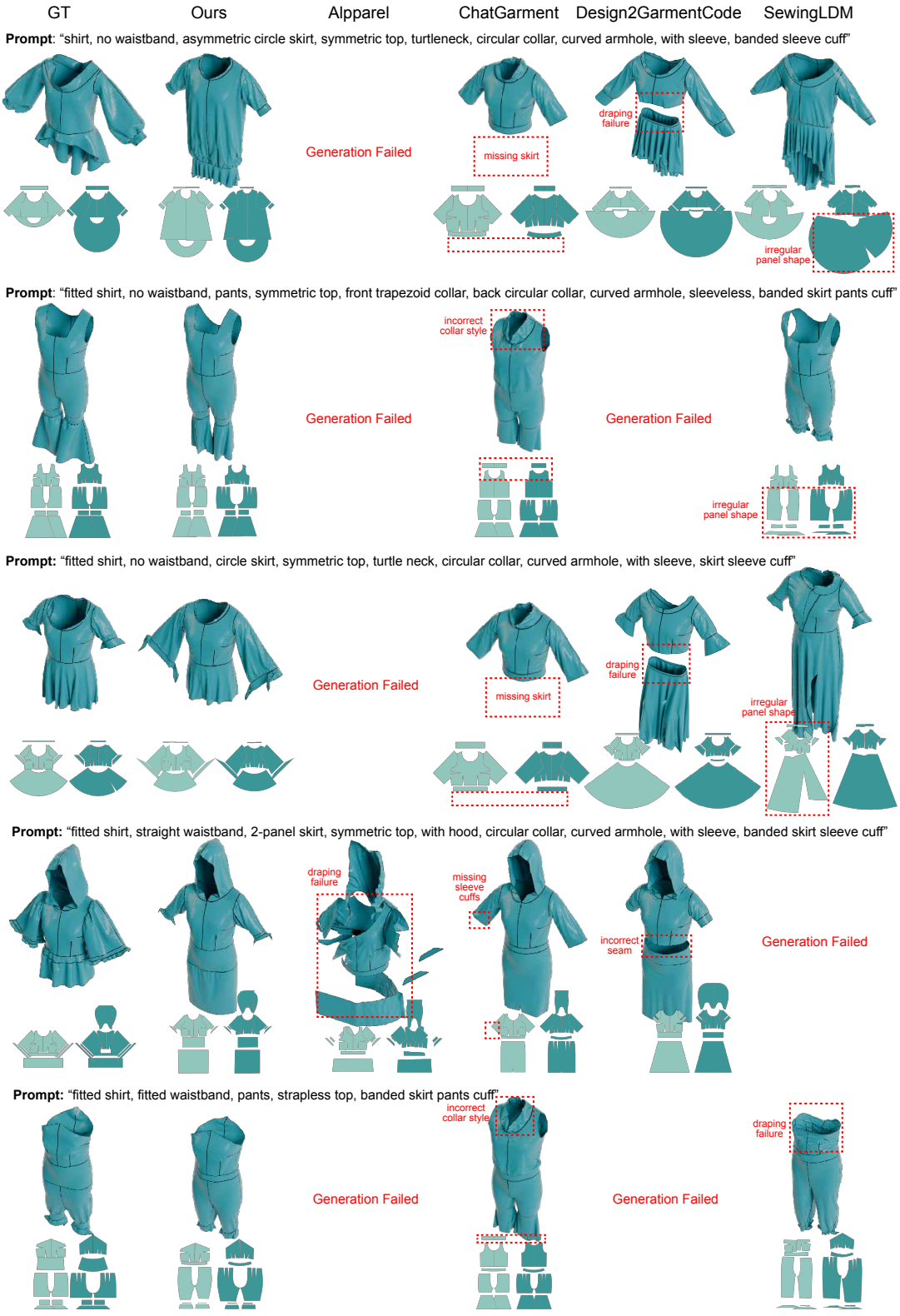

Fig. 18. **Text-Conditioned Generation: Additional Visualization.**

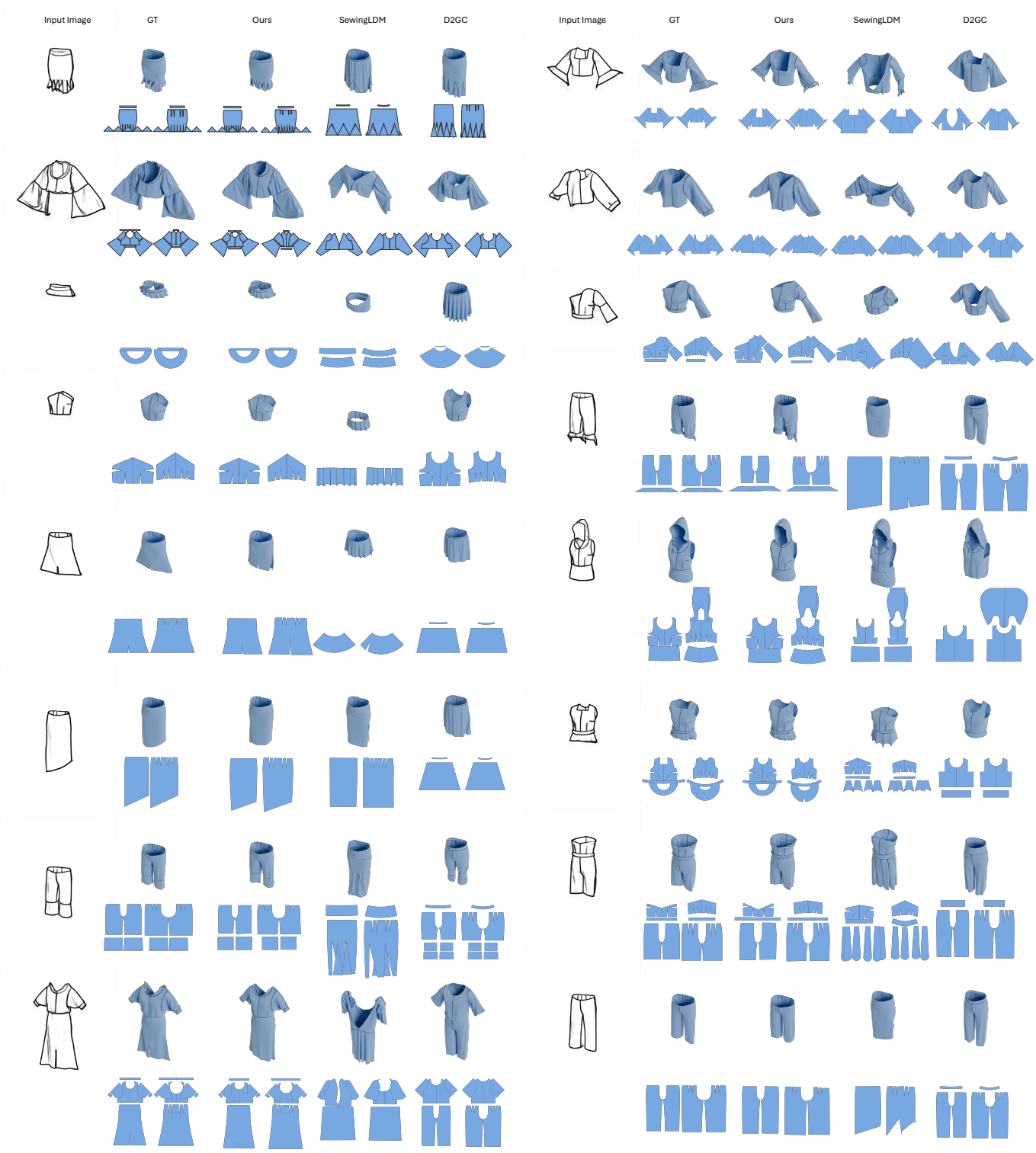

Fig. 19. **Image-Conditioned Generation: Additional Visualization on Garment Sketches Dataset.**

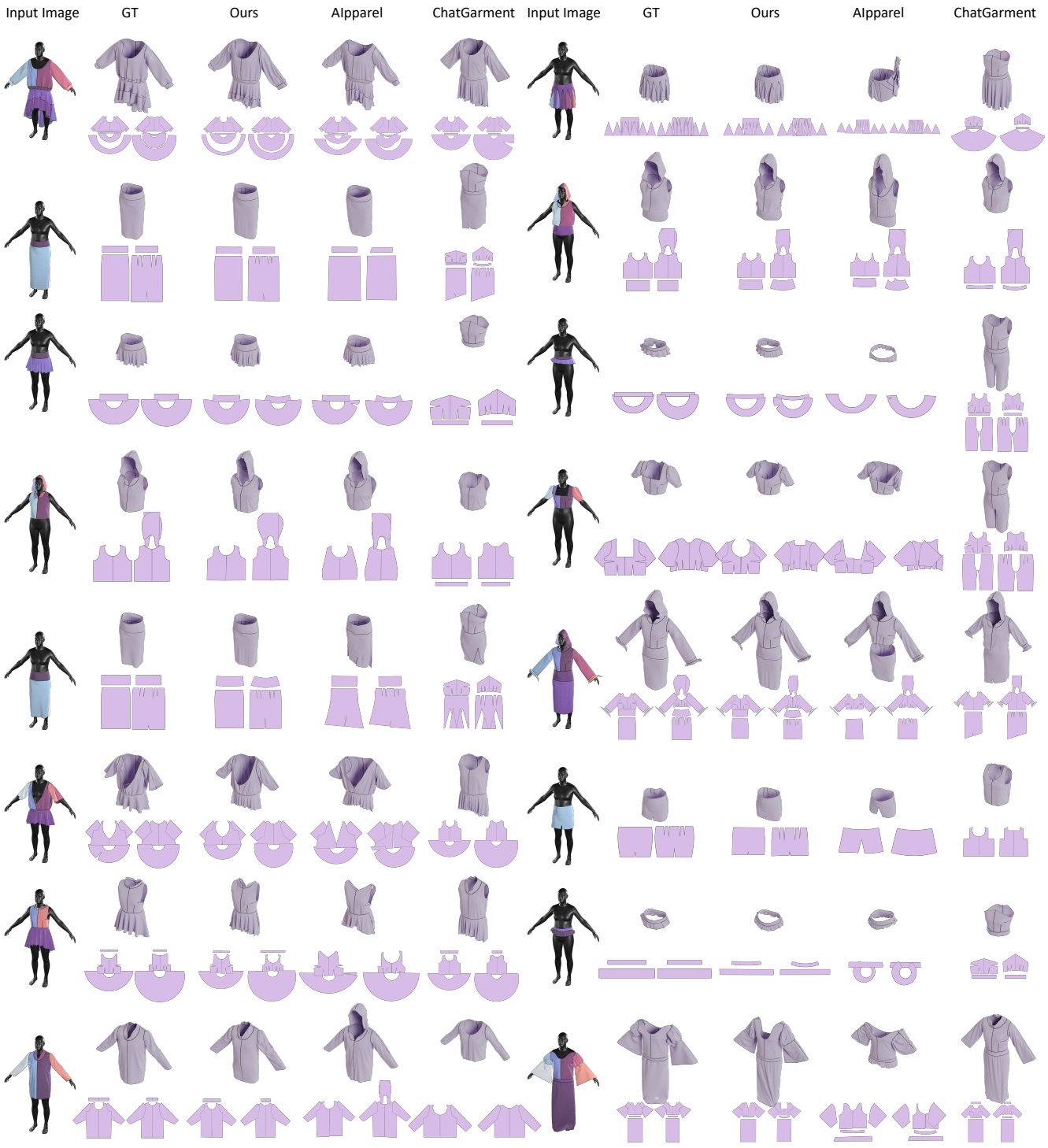

Fig. 20. **Image-Conditioned Generation: Additional Visualization on GCDV2 Dataset.**

Fig. 21. **Garment Interpolation Results.**

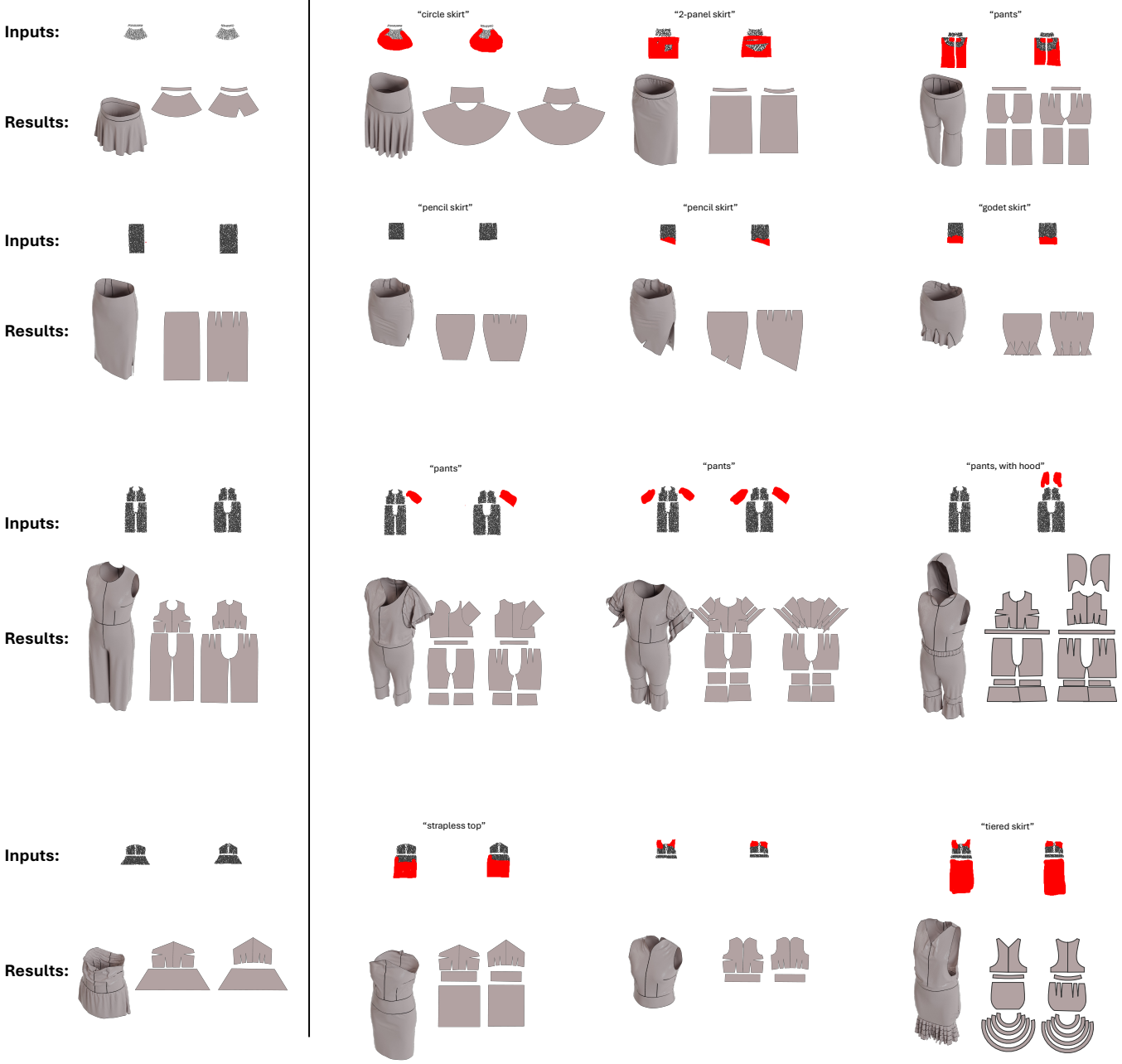

Fig. 22. **Additional Sewing Pattern Editing Results.** Each row showcases a modified sewing pattern of the garment asset on the left. The red paint indicates users' input. The modified sewing pattern, combined with an optional text prompt, guides the generation of GPF garments. The generated garment asset after draping and its sewing pattern are shown below the inputs.

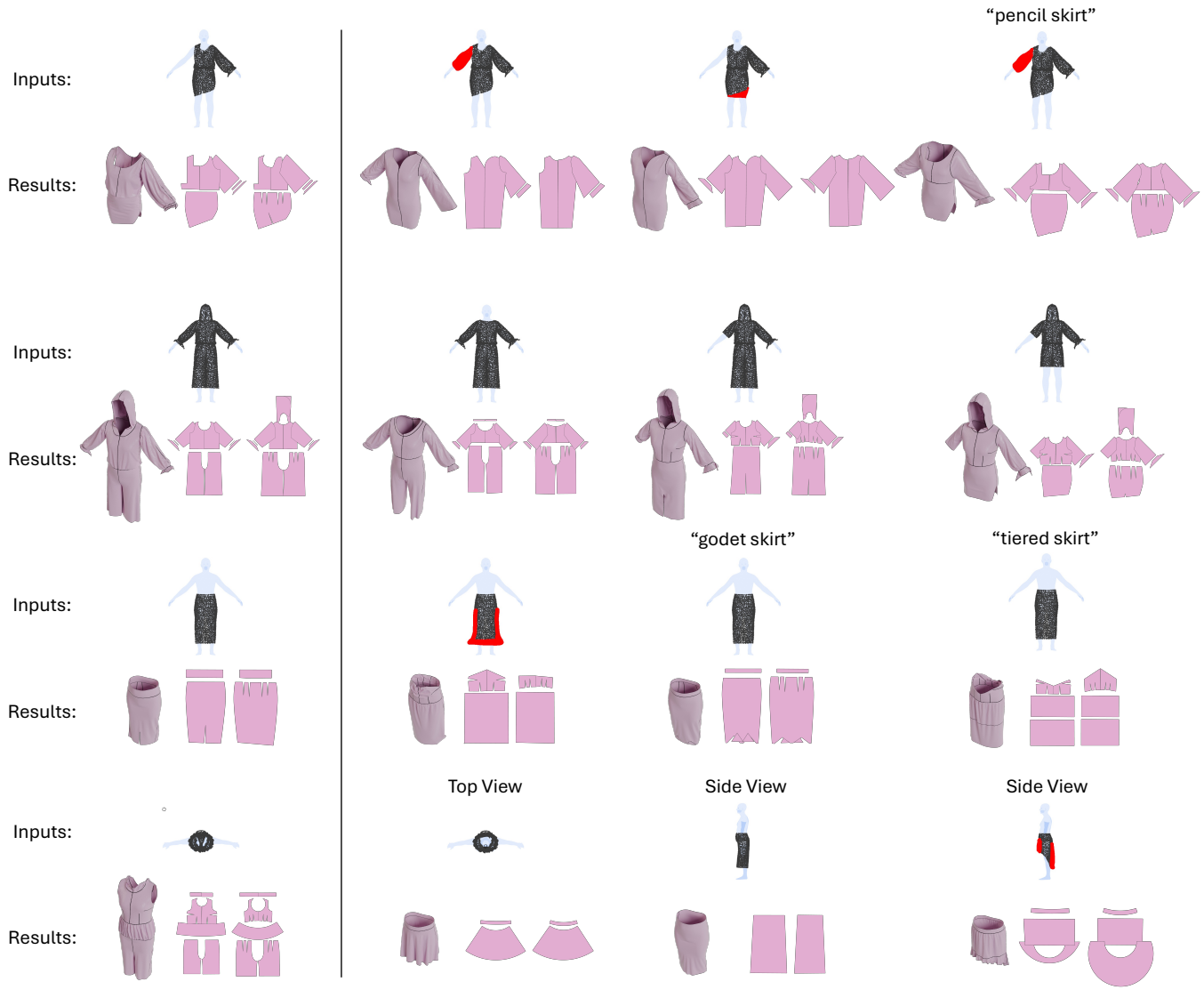

Fig. 23. **Additional Silhouette Editing Results.** Each row showcases a garment asset generated by GPF conditioned on the silhouette shown in the input. An additional text prompt is also fed into GPF for an extra constraint. The leftmost column shows the initial garment generated from GPF, from which the silhouette edits are performed. The red paint indicates a newly added silhouette using our 2D interface. The last row shows silhouettes from different views that are used as conditioning.

AIpparel	SewingLDM	D2G	ChatGarment	Ours
923.8	985.1	1048.5	1060.4	**1065.4**

Table 6. **Human Study ELO Ranking.** (Higher is better)

C.4 Human & VLM Study

We conduct a human study on Amazon Mechanical Turk (440 responses) comparing the text alignment, aesthetic quality, and physical plausibility of text-based garment generation across baselines. Specifically, we present two garment renderings from separate models and ask users to select the garment with higher aesthetic appeal, physical plausibility, and text alignment. All the garments are generated with a selected set of 18 text prompts. After collecting all the responses, we compute ELO rankings following [Wu et al. 2024], and the results are shown in Table 6. Our method achieves the highest score, indicating overall better alignment with text prompt, physical plausibility, and aesthetics compared with the baselines.

For a more detailed analysis, we also conduct a VLM study, adapting the setup from GPTEval3D [Wu et al. 2024] to evaluate the same three

Fig. 24. **Additional Point-cloud-conditioned Generation Results.**

Method	Garment Aesthetics	Text-Prompt Alignment	Physical Plausibility
AIpparel	821.6	888.2	857.1
SewingLDM	993.3	1042.2	963.1
D2GC	1060.9	1084.0	1030.5
ChatGarment	1040.3	858.6	1048.3
Ours	**1168.6**	**1143.8**	**1126.5**

Table 7. **VLM Evaluation Results**. (ELO rankings, higher is better)

criteria separately. We use Gemini-2.5-flash as our VLM model. The ELO ranking after running the study is shown in Table 7. We see that the general ranking trend aligns with that from the human study, but reveals variations in the baselines when evaluating the three criteria separately. For example, while Chatgarment is second-best in physical plausibility, its text-prompt alignment is poor. In the meantime, Design2GarmentCode achieves a good balance among the three criteria.

AIpparel	D2GC	ChatGarment	Ours (GPF/PPF)
4.59s	29.52s	14.24s	**4.01s**(2.48s/1.53s)

Table 8. **Garment Generation Runtime Comparison.** (Seconds, lower the better.)

C.5 Runtime Analysis

C.5.1 Generation Runtime Analysis. In Table 8, we compare with baselines the garment generation time(seconds) averaged over our test-set (1024 samples) using 100 denoising steps each. Our method achieves a similar runtime compared to AIpparel, and is much faster than ChatGarment and Design2GarmentCode since they require external LLM queries.

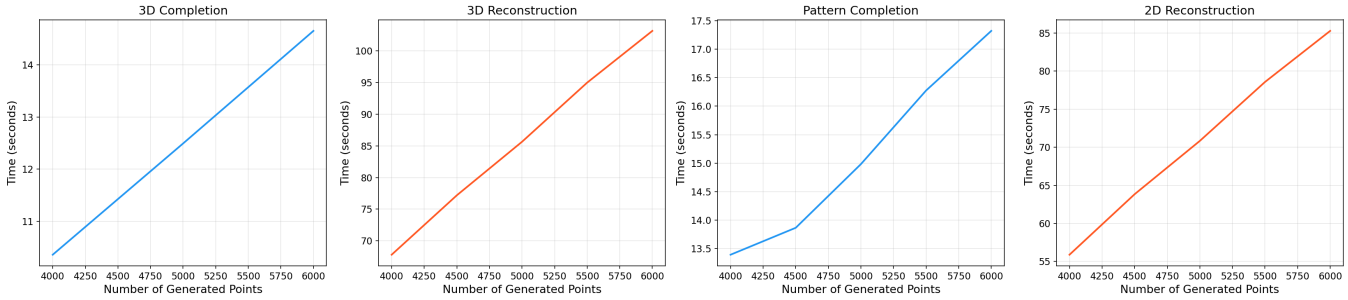

Fig. 25. **Runtime Analysis for Garment Editing Tasks.**

C.5.2 Editing Runtime Analysis. We also report garment editing runtime for the different tasks we showcased in the paper. Because the DPS algorithm's runtime depends on hyperparameters such as the input number of points, we plot the total runtime of DPS with different number of input points, given the same loss and observations. Figure 25 shows the runtime plot for the four garment editing tasks we shows in the paper. Specifically, 3D & pattern reconstructions use EMD as their losses and completions use Chamfer Distance. In general, the runtime grows roughly linearly as the number of points grows. This suggests that the main time-bottleneck comes from CD or EMD computation instead of our network forward pass.

C.6 Out-of-domain Evaluation

DMap	GarmentRecovery	D2GC	ChatGarment	Ours
18.729	10.830	14.005	**7.104**	7.813

Table 9. **Evalution on 4DDress Dataset** (Chamfer Distance, lower is better)

To evaluate our method's ability to generalize to out-of-domain input data, we quantitatively evaluated our method against baselines on a subset of the 4DDress [Wang et al. 2024] dataset for the task of image-to-sewing-pattern reconstruction. Table 9 shows the comparison with both 3D garment reconstruction baselines, such as DMap [Li et al. 2025a] and GarmentRecovery [Li et al. 2024a], as well as sewing pattern generation baselines like ChatGarment [Bian et al. 2025] and Design2GarmentCode [Zhou et al. 2024]. Compared to the baselines, our method achieves comparable reconstruction accuracy compared to the state-of-the-art method while outperforming the optimization-based approaches, which heavily rely on human pose and input image segmentations.