# PatternGSL: A Structured Specification Language for Template-Free and Simulation-Ready 3D Garments

ZHENYANG LI*, The University of Hong Kong, Hong Kong SAR, China
LUTAO JIANG*, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
YIZHOU ZHAO, Carnegie Mellon University, Pittsburgh, PA, USA
YING-CONG CHEN, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
XIN WANG†, LIGHTSPEED, Shenzhen, China
WEIKAI CHEN†‡, LIGHTSPEED, Los Angeles, CA, USA
YIFAN PENG†, The University of Hong Kong, Hong Kong SAR, China

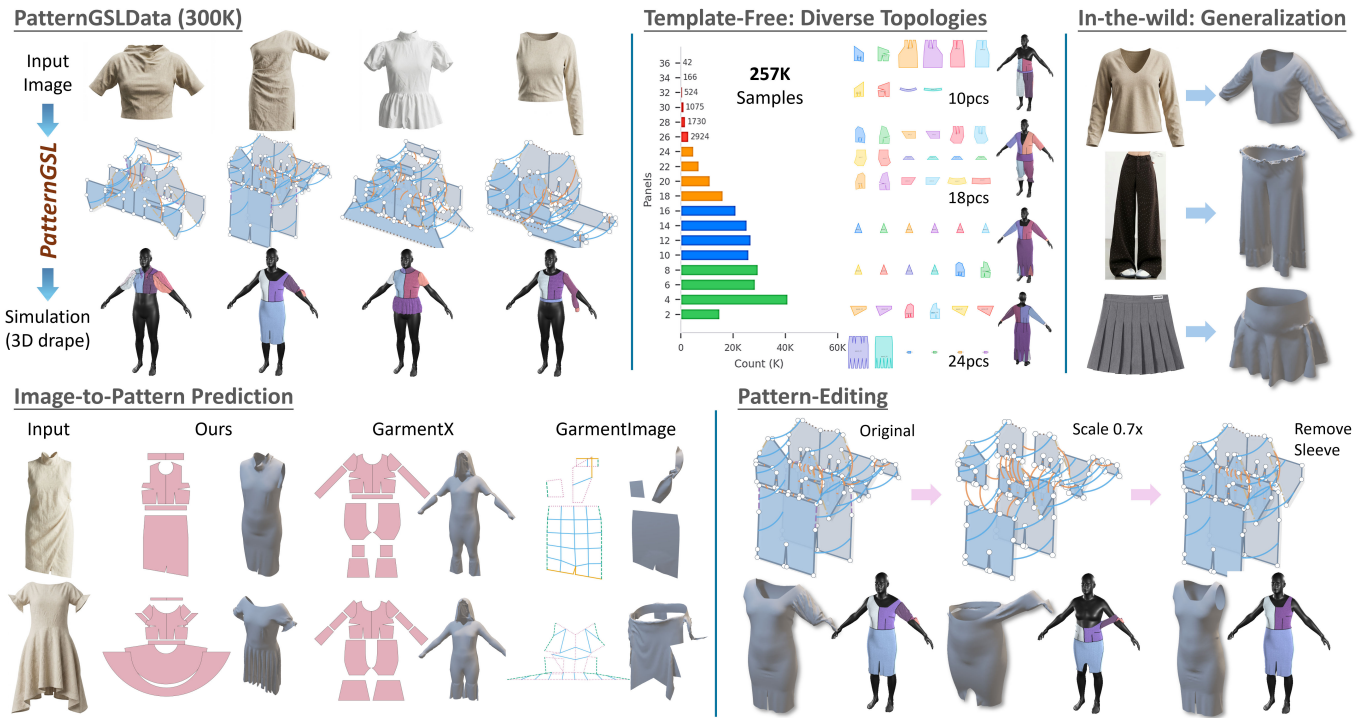


Fig. 1. We present PatternGSL, a template-free approach for reconstructing garment sewing patterns from images. Built upon PatternGSLData—a large-scale image-to-pattern dataset with 300K samples—our method learns to predict structured pattern representations that can be directly simulated as 3D garments. In our current dataset and evaluation, PatternGSL covers diverse topologies ranging from 2 to 37 panels and generalizes to in-the-wild photographs. Compared to prior methods, our approach produces higher-fidelity reconstructions while maintaining simulation compatibility. The predicted patterns further support intuitive editing operations such as scaling and component removal.

Reconstructing realistic, physically plausible garments from a single image remains a fundamental challenge. Template-free methods capture surface geometry but lack explicit sewing structure for simulation; while programmatic systems are simulation-ready but constrained by predefined templates. This reveals a fundamental representation gap between geometric reconstruction and structured garment construction. We present PatternGSL, a

*Both authors contributed equally to this research. This work was completed during their internship.
†Corresponding author.
‡Project Lead.

Authors' Contact Information: Zhenyang Li, The University of Hong Kong, Hong Kong, Hong Kong SAR, China; Lutao Jiang, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangzhou, China; Yizhou Zhao, Carnegie Mellon University, Pittsburgh, Pittsburgh, PA, USA; Ying-Cong Chen, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangzhou, China; Xin Wang, LIGHTSPEED, Shenzhen, Shenzhen, China; Weikai Chen, LIGHTSPEED, Los Angeles, Los Angeles, CA, USA; Yifan Peng, The University of Hong Kong, Hong Kong, Hong Kong SAR, China.


ACM 1557-7368/2026/6-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

structured garment representation in the form of a template-free and learnable specification language that encodes complete sewing patterns, including panel boundaries, parameterized seams, and explicit stitch topology, in a compact and standardized form. PatternGSL preserves the physical rigor of pattern-based models while removing template dependence, elevating sewing structure as a first-class target for generative modeling. We further propose a vision-language framework that predicts PatternGSL specifications directly from a single image and decodes them into garments using lightweight deterministic validity handling, without optimization-based refinement or manual cleanup. In addition, we introduce PatternGSLData, the first large-scale image-to-GSL paired dataset comprising 300K samples with complete sewing pattern annotations, enabling supervised VLM training for structured garment reconstruction. Experiments demonstrate improved pattern accuracy over prior baselines, explicit sewing-structure recovery, reliable cloth simulation, and pattern-level editing through the same deterministic decoding pipeline. Code and data-processing scripts will be released at https://github.com/PatternGSL/PatternGSL.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Shape modeling**; *Computer vision*; *Machine learning*.

Additional Key Words and Phrases: Garment modeling, Sewing patterns, Structured specification language, Simulation-ready garments, Image-based 3D generation

**ACM Reference Format:**
Zhenyang Li, Lutao Jiang, Yizhou Zhao, Ying-Cong Chen, Xin Wang, Weikai Chen, and Yifan Peng. 2026. PatternGSL: A Structured Specification Language for Template-Free and Simulation-Ready 3D Garments. *ACM Trans. Graph.* 1, 1 (June 2026), 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Reconstructing realistic and physically plausible garments from images remains a key challenge in digital fashion and 3D reconstruction. Beyond visual fidelity, practical solutions must seamlessly support physics-based simulation and downstream design workflows. Yet many prevailing learning-based approaches [Corona et al. 2021; Saito et al. 2019, 2020; Zhu et al. 2020] represent garments as neural implicit shapes, optimized primarily for visual plausibility. While such methods can capture diverse shapes without predefined templates, their geometry-first formulations obscure sewing structures and tend to smooth out sharp seams and boundaries, which in turn hinders their direct use in cloth simulation (Fig. 1).

At the other end of the spectrum, recent programmatic systems for garment modeling highlight the advantages of explicit structure. GarmentCode [Korosteleva and Lee 2023], for example, introduces a modular, simulation-ready framework in which garments are constructed through compositional programs that define components and stitching operations. This design enables precise control over garment construction and physical behavior. Nonetheless, GarmentCode's expressiveness is rooted in hand-crafted garment programs and predefined components, which constrain the space of admissible topologies. Consequently, the framework remains fundamentally template-driven and has limited ability to infer novel sewing structures or arbitrary garment topologies directly from images.

These observations reveal a persistent representation gap. Real garments are constructed from 2D fabric panels whose shapes, seam curves, and stitching relationships jointly determine their 3D form and physical behavior. Bridging this gap calls for a representation that combines the **physical rigor of pattern-based models** with the **flexibility of template-free approaches**, while remaining compatible with learning-based systems. Such a representation should explicitly encode sewing structure – panels, curves, and stitches – so that reconstructed garments are immediately usable in cloth simulation, yet avoid rigid, hand-crafted templates by supporting arbitrary topologies. At the same time, it must be sufficiently regular and standardized to act as a stable, learnable target for image-conditioned generative models.

To this end, we introduce **PatternGSL**, a new representation that rethinks how sewing patterns are encoded for learning and simulation. Rather than treating garments as fixed, hand-crafted programs as in prior work, PatternGSL reframes sewing patterns as a regularized, extensible, and learnable specification. It distills the essential elements of garment construction into a standardized language in which panels, seam curves, and stitch correspondences are first-class primitives, decoupled from rigid program templates. In doing so, we preserve the geometric precision and physical validity that make pattern-based approaches appealing, while replacing monolithic programs with a compositional representation that scales to arbitrary garment topologies and can be predicted directly by modern generative models.

Representation-wise, as shown in Fig. 2, we organize garments into a clear hierarchy of panels, parameterized edges, and stitches, explicitly separating continuous geometry from discrete topology while preserving their correspondence. Panel boundaries are encoded with curve parameterizations that regularize shape yet retain high fidelity along seams. Stitch relationships are structured as explicit pairwise correspondences, rather than being inferred heuristically. Collectively, this design elevates sewing patterns from ad hoc, program-specific artifacts to a consistent, learnable format, providing a stable interface between perception and simulation. Algorithm-wise, we formulate single-image garment reconstruction as the problem of generating a PatternGSL configuration. We train a vision-language model to predict this structured representation directly from an input image, leveraging its ability to generate long, hierarchical outputs that adhere to the constraints of the language. The resulting specification is parsed and decoded into a complete sewing pattern with deterministic validity handling, and can then be fed to a cloth simulator without optimization-based refinement or manual cleanup.

Beyond reconstruction, PatternGSL's *explicit parameterization* enables direct and intuitive pattern editing. Designers can resize panels by modifying vertex coordinates, reshape boundaries by adjusting curve parameters, or filter panels by removing parts. The edited specification is then processed by the same deterministic decoder used for reconstruction, preserving compatibility with the simulation pipeline. This *editability* naturally extends to in-the-wild scenarios, where patterns predicted from real-world images can be iteratively refined to meet design requirements. As shown in Fig. 1, experiments on synthetic and in-the-wild datasets show that PatternGSL improves pattern accuracy and simulation reliability over prior baselines, achieving 99.2% simulation success while supporting flexible pattern editing. In summary, our contributions are:

- A structured and template-free garment specification language, dubbed PatternGSL, that regularizes sewing patterns as a learnable representation, explicitly encoding panels, edges, and stitches while supporting arbitrary garment topologies;
- A vision-language-based generation framework that predicts PatternGSL directly from a single image, together with a parser and deterministic decoder that converts the prediction into complete sewing patterns for simulation;
- PatternGSLData, the first large-scale image-to-GSL paired dataset comprising 300K samples (250K synthetic + 50K photorealistic) with complete panel, edge, and stitch annotations, enabling supervised VLM training;
- Empirical validation on pattern accuracy, stitch correctness, draping reliability, editing validity, and in-the-wild generalization.

## 2 Related Work

*3D Garment Reconstruction.* Template-based approaches [Guan et al. 2012; Pons-Moll et al. 2017; Santesteban et al. 2019; Vidaurre et al. 2020] deform predefined meshes using parametric body models [Loper et al. 2015; Pavlakos et al. 2019; Xu et al. 2020]. Implicit representations [Corona et al. 2021; He et al. 2021; Saito et al. 2019, 2020; Xiu et al. 2022] enable high-resolution digitization but produce fused geometry. BCNet [Jiang et al. 2020] separates layers via depth prediction; DeepFashion3D [Zhu et al. 2020] and MGN [Bhatnagar et al. 2019] provide benchmarks. Text-driven methods [Huang et al. 2024; Liu et al. 2024b; Luo et al. 2024; Srivastava et al. 2024] generate garments from descriptions. These output meshes or implicit fields rather than simulation-ready sewing patterns.

*Sewing Pattern Representations.* Existing approaches show the value of explicit sewing structure but often tie prediction to predefined garment families. NeuralTailor [Korosteleva and Lee 2022] predicts from point clouds but requires template selection; SewFormer [Liu et al. 2023c] uses transformers for panel prediction; GarmentCode [Korosteleva and Lee 2023] provides a powerful programmatic construction system with reusable components and stitching operations. Our work builds on this sewing-based view, but targets a canonicalized sequence representation for learning: panels are deterministically ordered, vertices are preserved as indexed arrays, and stitches explicitly reference panel-edge pairs. Thus PatternGSL is not only a JSON serialization of sewing patterns, but a stable output space for image-conditioned generation.

Recent methods also differ in output space and evaluation target. GarmentX [Guo et al. 2025] uses autoregressive garment generation, while our representation keeps explicit panel, edge, and stitch topology as the supervised prediction target. Dress-1-to-3 [Li et al. 2025b] recovers 3D garments using diffusion priors and differentiable physics; in contrast, we directly predict 2D sewing patterns as the primary output, enabling topology-aware supervision, pattern editing, and direct cloth simulation. GarmentImage [Li et al. 2024] encodes patterns for CNN prediction but loses boundary information; GarmentDiffusion [Li et al. 2025a] generates patterns via diffusion transformers. Physics-based cloth simulation [Li et al. 2020; Narain et al. 2012; Tang et al. 2018] and differentiable simulators [Li et al. 2022; Liang et al. 2019] provide the simulation backends used by many of these systems.

*Generative Models and Structured Prediction.* Diffusion models [Ho et al. 2020; Song et al. 2021a,b] achieve remarkable success in image synthesis [Rombach et al. 2022; Saharia et al. 2022] with classifier-free guidance [Ho and Salimans 2022]. For 3D generation, methods span text-to-3D [Lin et al. 2023; Poole et al. 2023], multi-view synthesis [Liu et al. 2023b; Shi et al. 2024; Tang et al. 2024], and meshes [Gao et al. 2022; Siddiqui et al. 2024]. Structured outputs include layouts [Chai et al. 2023; Inoue et al. 2023] and physics-aware motion [Tevet et al. 2023; Yuan et al. 2023]. Large VLMs [Alayrac et al. 2022; Bai et al. 2023; Li et al. 2023; Liu et al. 2024a] combine visual encoders [Dosovitskiy et al. 2021; Radford et al. 2021] with language models [Chiang et al. 2023; Touvron et al. 2023], excelling at structured generation [Chen et al. 2021; Raffel et al. 2020] with efficient LoRA fine-tuning [Hu et al. 2021]. We leverage VLMs by designing GSL as a compact structured format.

## 3 PatternGSL

### 3.1 Design Goal

The goal of PatternGSL is to make sewing structure a direct prediction target. A garment pattern is naturally discrete-continuous: panels define 2D fabric pieces, boundary curves define their shape, 3D transforms initialize draping, and stitches define how edges are assembled. We therefore seek a representation that is compact enough for sequence generation, expressive enough for arbitrary panel layouts, and explicit enough to be decoded into a cloth simulation input.

PatternGSL follows this organization rather than hiding it in an implicit embedding. Geometry and topology are separated, but their correspondence is kept through indexed references: panels contain ordered vertices and edges, and stitches point to specific panel-edge pairs. This makes the output interpretable, editable, and learnable as a structured language.

### 3.2 Representation Design

We instantiate PatternGSL as a hierarchical JSON structure with three components (Fig. 2, top-left): (1) a `meta` block storing the scale factor $\sigma_{\text{scale}}$, panel count, and boundary sample count $N_s$; (2) a `panels` array storing geometry and placement; and (3) a `stitches` array storing topology. The scale factor is part of the representation rather than the output of a separate monocular scale estimator. During training, coordinates are normalized using a fixed scale prior and the corresponding value is stored in the `meta` fields; at inference, the VLM predicts this scale token jointly with panel geometry and stitch topology.

*Panel Geometry.* Each panel $\mathcal{P}_p$ is encoded as a JSON object containing its 2D boundary and 3D placement. The boundary is represented by an ordered vertex sequence $\{(x_j, y_j)\}_{j=1}^{n_p}$ defining the panel polygon. To make numeric tokens stable across garment sizes, local coordinates are normalized as

$$\mathbf{v}_{\text{gsl}} = \sigma_{\text{scale}} \cdot \mathbf{v}_{\text{local}}. \tag{1}$$

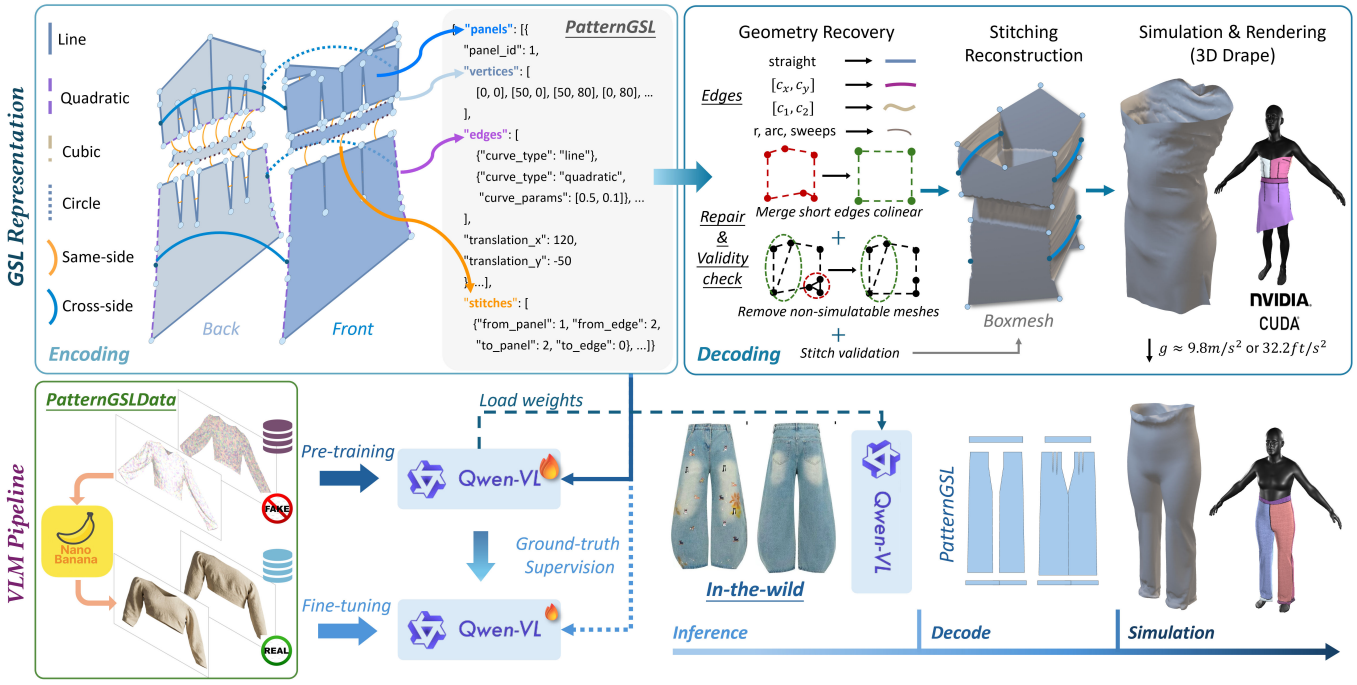


Fig. 2. The PatternGSL framework. **(Top-left)** GSL Representation: legend denotes edge types (line/quadratic/cubic/circle) and stitch types (same-side/cross-side); sewing patterns are encoded into PatternGSL while preserving vertices, curves, and topology. **(Top-right)** Decoding: geometry recovery, repair (merging short edges, removing invalid panels), stitching reconstruction, boxmesh generation, and physics simulation (NVIDIA Warp). **(Bottom-left)** VLM Pipeline: PatternGSLData is combined with NanoBanana for back-view synthesis and photorealistic rendering; two-stage training (pre-training on FAKE, fine-tuning on REAL) with ground-truth supervision; fire icons indicate trainable models. **(Bottom-right)** Inference: frozen Qwen-VL predicts PatternGSL from in-the-wild images, then decoded and simulated into 3D garments.

For 3D placement initialization during draping simulation, each panel stores six transform parameters: translations $(t_x, t_y, t_z)$ and rotations $(r_x, r_y, r_z)$. The horizontal translations are normalized relative to a reference template width as $(t_x^{\text{gsl}}, t_y^{\text{gsl}}) = (t_x, t_y) \cdot \sigma_{\text{scale}}/W_{\text{ref}}$, while depth and rotations are preserved directly. Each panel carries a binary `side` attribute ("front" or "back") determined by $\text{sign}(t_z)$, aiding VLM reasoning about garment structure.

*Edge and Curve Representation.* Edges connect consecutive vertices and may be straight or curved. GSL supports four curve types with explicit parameterization. To avoid absolute control-point coordinates that vary with garment scale, curve parameters are expressed relative to each edge endpoint. Line segments require no additional parameters. Quadratic Bézier curves store $(c_0, c_1)$ and reconstruct the control point as $\mathbf{p}_c = \mathbf{p}_0 + (c_0, c_1) \cdot L$, where $L$ is the edge length. Cubic Bézier curves store four relative parameters for two control points, and circular arcs store radius and orientation flags.

While parametric representations suffice for simulation, VLMs benefit from explicit geometric supervision on curve shapes. For curved edges, GSL additionally stores $N_s$ uniformly sampled points along the curve. Critically, straight edges omit sample points entirely, reducing token count substantially since most garment edges are straight, while key curved edges (e.g., necklines, armholes, hemlines) still receive rich geometric supervision.

*Stitch Topology.* Stitch relationships encode how panel edges are sewn together during assembly. Each stitch is represented as a JSON object with a unique `stitch_id`, along with source identifiers (`from_panel_id`, `from_edge_index`) and target identifiers (`to_panel_id`, `to_edge_index`) that specify exactly which edges are connected. This explicit encoding avoids ambiguity when geometrically close edges (e.g., front and back necklines) belong to different panels. Unlike implicit representations where stitches must be inferred from geometric proximity, GSL stores exact connectivity, enabling precise topology recovery.

## 3.3 Canonicalized Encoding

Before tokenization, we canonicalize each garment specification so that equivalent inputs produce a stable training target. Panel names are sorted to assign deterministic panel IDs from 1 to $P$; vertex ordering within each panel is preserved through indexed arrays; and stitches are rewritten as explicit (`panel_id`, `edge_index`) references. This stable ordering reduces avoidable output ambiguity for the VLM.

Given panels $\{(\mathcal{P}_p, \mathbf{T}_p)\}_{p=1}^{P}$ and stitches $\{\mathcal{S}_s\}_{s=1}^{S}$, encoding then applies the normalization above and rounds numeric fields to two decimal places. Edge types and relative curve parameters are inferred from the input curvature fields. For curved edges, $N_s$ boundary samples are evaluated along the parametric curve; for straight

edges, samples are omitted. Finally, stitches are converted through the canonical panel-ID mapping and invalid references are discarded.

### 3.4 Deterministic Decoding and Validity Handling

The decoder inverts the structured encoding and performs lightweight deterministic validity handling before simulation (Fig. 2, top-right). It first reads the predicted $\sigma_{\text{scale}}$ token from the `meta` block and applies inverse normalization,

$$\mathbf{v}_{\text{local}} = \mathbf{v}_{\text{gsl}}/\sigma_{\text{scale}}, \tag{2}$$

with translations recovered analogously. No standalone body-scale estimator is used. The depth $t_z$ is read directly when present or initialized from the predicted `side` attribute.

Edges are reconstructed from the vertex arrays and curve parameters. When predicted curve parameters are missing or corrupted, boundary samples provide a deterministic fallback; for example, a quadratic Bézier control point can be recovered from the midpoint sample in closed form. Stitches are reconstructed by mapping panel IDs back to panel names. Before simulation, the decoder applies deterministic checks such as merging short collinear edges, removing invalid panels, and validating stitch references. These steps are rule-based sanitation rather than optimization-based refinement or manual correction.

### 3.5 Representation Properties

The design gives PatternGSL four properties needed for image-to-pattern learning.

*Canonical topology*: sorted panel IDs, preserved vertex order, and explicit stitch references produce a consistent GSL target for a given garment specification. Connectivity is stored directly as index pairs rather than inferred from geometric proximity.

*Geometric precision*: coordinate quantization to two decimal places bounds error to $\epsilon < 0.01/\sigma_{\text{scale}} \approx 0.04$ mm, below typical simulation tolerances (∼0.1 mm). Relative curve parameterization keeps this precision independent of absolute garment scale.

*Token efficiency*: storing only boundary vertices yields $O(n)$ tokens per panel, relative curve parameters require 2–4 values per curved edge, and straight edges omit samples. Typical garments encode within ∼10K tokens.

*Editability*: geometric parameters have direct semantic meaning, so users can modify vertex coordinates, curve parameters, or placements and then run the same deterministic decoder. The complete GSL schema is provided in the supplementary material.

## 4 Image-to-Pattern Prediction via Vision-Language Model

We use PatternGSL as the output space for single-image sewing-pattern prediction.

### 4.1 Problem Formulation

As shown in Fig. 3, given a front-view RGB image, NanoBanana synthesizes an auxiliary back view, yielding a paired input $\mathcal{I} \in \mathbb{R}^{2\times3\times H_{\text{img}}\times W_{\text{img}}}$. We predict the complete GSL specification, including panel vertices, curve parameters, 3D transforms, stitches, and the scale token. The task is conditional sequence generation:

$$P(J|\mathcal{I}) = \prod_{t=1}^{T} P(j_t|j_{<t}, \mathcal{I}; \theta), \tag{3}$$

where $J = \{j_1, \ldots, j_T\}$ represents the GSL tokens output and $\theta$ denotes network's parameters. This formulation naturally handles variable garment complexity through variable-length sequences while maintaining structured output constraints.

### 4.2 Model Architecture

We employ Qwen3-VL [Yang et al. 2025], consisting of a vision encoder and language decoder. The front and synthesized back views are patchified and encoded independently with shared weights, producing $\mathcal{T}_{\text{front}}$ and $\mathcal{T}_{\text{back}}$. They are fused only in the language decoder after concatenation with the instruction tokens. Thus the synthesized back view is an auxiliary cue for hidden regions, rather than a hard geometric constraint imposed before generation.

*Vision Encoder.* The front view and synthesized back view are partitioned into flattened image patches $\mathcal{I}_p \in \mathbb{R}^{2\times n\times l}$, where $n$ is the number of patches per image and $l$ is the number of pixel values in each patch. A shared-weight ViT encoder maps these patches to visual tokens, $\mathcal{T}_{\text{front}} = \{\mathbf{f}_1^{\text{front}}, \ldots, \mathbf{f}_n^{\text{front}}\}$ and $\mathcal{T}_{\text{back}} = \{\mathbf{f}_1^{\text{back}}, \ldots, \mathbf{f}_n^{\text{back}}\}$, with each token summarizing local appearance and shape cues. Processing the two views independently avoids imposing an early geometric alignment; cross-view reasoning is left to the language decoder during structured generation.

*Language Decoder.* Given visual tokens and a prompt such as "Generate garment specification JSON", the decoder autoregressively generates GSL tokens until the terminal ⟨end⟩ token is produced. Variable-length generation naturally supports garments with different numbers of panels, edges, and stitches.

### 4.3 PatternGSLData Dataset Construction

To our best knowledge, no existing dataset offers large-scale paired image-pattern annotations for learning-based garment reconstruction. We introduce PatternGSLData, organized for two-stage training (Fig. 2, bottom-left): (1) 250K synthetic rendered images generated from GarmentCodeData specifications with diverse viewpoints and lighting; and (2) 50K photorealistic images produced by NanoBanana for domain adaptation. NanoBanana is used as an image-to-image translation and view-synthesis module: it transfers synthetic renderings toward more realistic textures and lighting and synthesizes a back view from the available front-view image. The REAL subset is not manually annotated from unconstrained photographs. Instead, garments with known sewing patterns are rendered and translated to more realistic appearance, so the ground-truth GSL annotations are inherited from the original structured specifications. Truly in-the-wild photographs are reserved for evaluation.

### 4.4 Training Strategy

Inspired by Visual Instruct Tuning [Liu et al. 2023a], we reformulate garment parsing as a visual instruction-following task. The fire icons in Fig. 2 indicate trainable model components during our two-stage training process. We aim to train a large vision-language

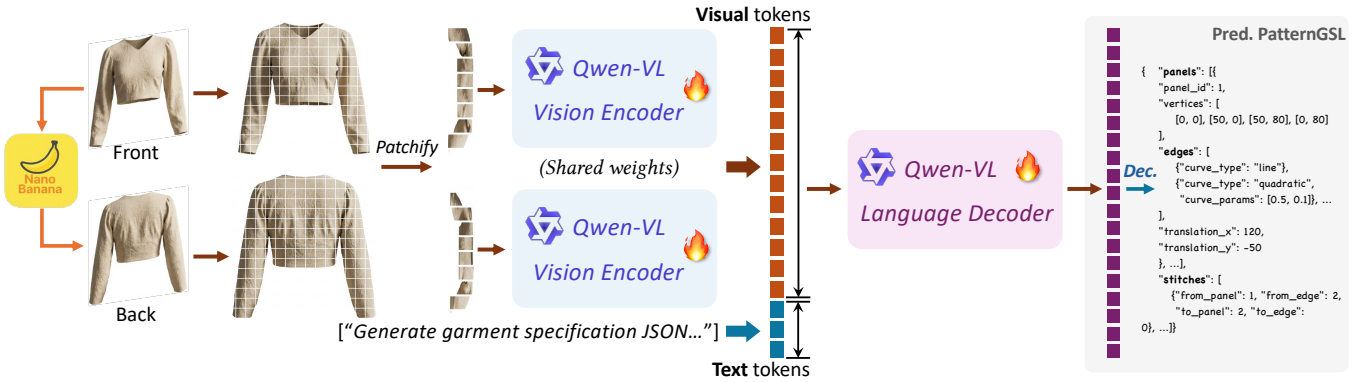


Fig. 3. VLM architecture for image-to-pattern prediction. Given a front-view image, NanoBanana synthesizes the back view. Both views are patchified and processed by shared-weight Qwen-VL Vision Encoders to produce visual tokens. These are concatenated with text tokens from the instruction prompt and fed into Qwen-VL Language Decoder, which autoregressively generates PatternGSL tokens. A decoder (Dec.) converts the predicted tokens into structured JSON format. Fire icons indicate trainable components.

model to generate structured garment sequences conditioned on user instructions. The dataset is constructed using a single-turn dialogue template, where the user input concatenates visual tokens with a textual command: "⟨image⟩ ··· ⟨/image⟩ Generate the garment specification JSON according to the input images." The system assistant's response is filled with the ground-truth GSL sequence.

We fine-tune Qwen3-VL using a standard cross-entropy loss: $\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T} \log P(j_t^{\text{gt}} | j_{<t}^{\text{gt}}, \mathcal{I}; \theta)$, where $j_t^{\text{gt}}$ denote ground-truth tokens. This provides direct supervision on structured outputs. At inference time (Fig. 2, bottom-right), the frozen model predicts PatternGSL from in-the-wild images, which are then decoded and simulated to produce 3D draped garments.

## 5 Experiments

We validate our approach through: (1) *quantitative comparison* on 2D pattern accuracy and 3D simulation quality; (2) *ablation studies* on boundary sampling and VLM training; (3) *pattern editing* via panel scaling, curve adjustment, and component removal; (4) *qualitative comparison*; and (5) *in-the-wild generalization.*

### 5.1 Experimental Configuration

*Dataset.* We use PatternGSLData described in Sec. 4.3, with a train/validation split. The validation set includes garments with novel topologies unseen during training. On the pre-tokenization PatternGSL JSON split, the dataset contains 257,956 garments, 2.84M panels, and 8.17M stitches; edges per panel average 6.91 and stitches per garment average 31.66, with long tails up to 40 edges and 106 stitches. Full topology statistics are provided in the supplementary material.

*Baselines.* We compare against: (1) **SewFormer** [Liu et al. 2023c], transformer-based for sewing pattern prediction; (2) **LGM** [Tang et al. 2024], large multi-view Gaussian model for 3D generation; (3) **GarmentX** [Guo et al. 2025], autoregressive method for garment generation; (4) **GarmentImage** [Li et al. 2024], which uses raster representation with CNN prediction. Dress-1-to-3 [Li et al. 2025b] targets 3D garment recovery rather than explicit 2D sewing-pattern prediction; we discuss this output-space difference in the supplementary material.

Table 1. Quantitative comparison on the validation set.

| Method | 2D Chamfer (mm)↓ | 2D IoU (%)↑ | Stitch Acc (%)↑ |
|---|---|---|---|
| SewFormer | 108.22 | 20.70 | 0.04 |
| GarmentX | 35.07 | 58.30 | 38.81 |
| GarmentImage | 29.07 | 58.50 | N/A |
| **Ours** | **5.78** | **86.34** | **98.48** |

*Metrics and Simulation.* We evaluate both 2D pattern quality (2D Chamfer distance in mm, 2D IoU, stitch connection accuracy) and 3D simulation quality (draping success rate, 3D Chamfer distance after simulation). All methods use one fixed cloth protocol with global settings, not tuned per sample, and no extra learned pose-estimation module. Key parameters are sim_fps=60, sim_substeps=10, and max_sim_steps=2400; the full protocol is in the supplementary material.

### 5.2 Quantitative Comparison

We evaluate on two complementary aspects: 2D pattern reconstruction accuracy and 3D simulation quality. Table 1 compares pattern-level metrics against baselines on the validation set. Our approach achieves substantially lower 2D Chamfer distance (5.78 mm vs. 29.07 mm for the next best) and higher 2D IoU (86.34% vs. 58.50%), indicating superior geometric fidelity. The 98.48% stitch accuracy confirms that our explicit topology encoding preserves connectivity information that implicit representations cannot capture.

Beyond 2D accuracy, Table 2 evaluates draping success rate and 3D Chamfer distance after physics-based simulation. Our method achieves 99.2% success rate compared to 4.7% for GarmentImage, validating that sub-millimeter coordinate precision ($\epsilon < 0.04$ mm) and explicit topology encoding satisfy the simulator constraints in most cases. Note that LGM generates 3D meshes rather than sewing patterns, so draping success rate is not applicable; we report

Table 2. Simulation quality assessment.

| Method | Success Rate (%)↑ | 3D Chamfer (mm)↓ |
|---|---|---|
| SewFormer | 81.2 | 666.77 |
| LGM | N/A | 26.79 |
| GarmentX | 95.3 | 68.62 |
| GarmentImage | 4.7 | 139.49 |
| **Ours** | **99.2** | **6.31** |

only its 3D Chamfer distance for geometric comparison. The low 3D Chamfer distance (6.31 mm) further confirms that predicted patterns produce realistic draping behavior without optimization-based refinement or manual cleanup. Figure 4 shows generalization to novel topologies unseen during training, including garments with over 12 panels and complex asymmetric designs.

We further measure generation reliability before and after deterministic validation. The raw JSON validity rate is 100.0%, the strict no-repair simulator-ready rate is 95.31%, and the final simulator-ready rate after deterministic validation is 99.2%. Strict no-repair failures are dominated by seam and topology inconsistencies and degenerate triangles rather than syntax errors, with the full breakdown in the supplementary material.

We also evaluate robustness to the auxiliary back-view cue. In a unified setting where the synthesized back view is unavailable or unreliable, the model still achieves 12.23 mm 2D Chamfer, 85.14% stitch accuracy, and 97.78% simulation success. In a stronger mismatch setting where the synthesized back view is inconsistent with the front view in sewing pattern, color, or style, performance degrades to 31.35 mm, 50.57%, and 66.18%, respectively, indicating graceful degradation under severe hidden-view ambiguity.

## 5.3 Ablation Studies

We conduct ablation studies to validate key design choices in both representation design and VLM training configuration.

*Representation Design: Boundary Sampling.* Table 3 ablates our boundary sampling strategy, providing explicit geometric supervision for curved edges. We vary both sample count $N_s \in \{4, 8, 16\}$ and sampling scope (curved edges only vs. all edges). Results show that $N_s = 8$ achieves optimal balance between geometric precision and token efficiency: fewer samples under-constrain curve shapes, while more samples mainly increase sequence length without proportional accuracy gains. Notably, sampling only curved edges outperforms sampling all edges, validating our design intuition that straight edges do not require explicit geometric supervision and extra samples on them contribute noise rather than useful signal.

Table 3. Ablation on boundary sampling strategy.

| Configuration | 2D Chamfer (mm)↓ | 3D Chamfer (mm)↓ |
|---|---|---|
| *Sample count (curved edges only):* | | |
| $N_s = 4$ | 8.42 | 9.15 |
| $N_s = 8$ **(Ours)** | **5.78** | **6.31** |
| $N_s = 16$ | 6.03 | 6.58 |
| *Sampling scope ($N_s = 8$):* | | |
| All edges | 7.21 | 7.89 |
| **Curved edges only (Ours)** | **5.78** | **6.31** |

Table 4. Ablation on VLM training configurations.

| Configuration | 2D IoU↑ | 2D Chamfer↓ | 3D Chamfer↓ |
|---|---|---|---|
| 4B w/ LoRA | 71.46 | 26.82 | 72.35 |
| 4B w/ Freeze | 66.79 | 32.39 | 124.79 |
| 4B w/o Freeze | 84.16 | 13.57 | 36.64 |
| **8B w/o Freeze** | **85.14** | **12.38** | **14.47** |

*VLM Training Configuration.* Table 4 ablates model capacity and fine-tuning strategies. Full fine-tuning of the 8B model with an unfrozen vision encoder achieves best performance. LoRA-based fine-tuning underperforms full fine-tuning, suggesting that accurate GSL generation requires adapting both visual understanding and language generation capabilities—the structured output format demands tight coordination between perceiving garment geometry and generating valid JSON sequences. Freezing the vision encoder also degrades performance, indicating that generic pre-trained visual features are suboptimal for extracting garment-specific geometric cues and benefit from task-specific adaptation.

## 5.4 Pattern Editing

A key advantage of explicit parameterization is the ability to edit predicted patterns directly. Since GSL stores geometric parameters with semantic meaning, designers can modify the JSON specification and run the same deterministic decoder.

We implement four common operations, illustrated in Fig. 5: panel scaling, curve adjustment, component removal, and sleeve spread. Each operation modifies explicit GSL fields such as vertices, curve parameters, panel labels, or translations; mathematical details are provided in the supplementary material.

Figure 5 demonstrates these operations across four garment samples with diverse topologies. Panel scaling (0.7–1.2×), curve adjustment, component removal, and sleeve spread (+10cm, +20cm) produce corresponding changes in both 2D layouts and 3D drapes. Across 40 edited patterns, all cases pass physics-based simulation, achieving 100% draping success. Additional editing formulations are provided in the supplementary material.

## 5.5 Qualitative Results

Figure 4 presents side-by-side comparisons across seven garment samples with diverse topologies. Our method consistently produces accurate sewing patterns that closely match ground truth geometry and drape realistically on the human body. GarmentX generates reasonable patterns but exhibits geometric distortions in complex regions such as sleeve-body connections. GarmentImage suffers from severe draping failures across most samples—the predicted patterns fail to properly cover the body due to imprecise geometry and invalid stitch topology, confirming the quantitative finding of only 4.7% simulation success rate. SewFormer completely fails on two samples (rows 4 and 6), producing garbled patterns that cannot be simulated. LGM generates plausible 3D meshes but lacks sewing pattern output entirely, limiting its utility for downstream manufacturing applications. These results demonstrate that our explicit GSL encoding enables both accurate pattern reconstruction and reliable physics-based simulation.

### 5.6 In-the-Wild Generalization

To validate practical applicability beyond synthetic data, we evaluate on in-the-wild garment images collected from e-commerce websites and fashion photography. These images exhibit significant domain shift from training data, including complex backgrounds, varied lighting conditions, occlusions, and diverse garment styles unseen during training. Figure 6 compares our method against baselines on seven in-the-wild samples. Our model achieves 100% simulation success, correctly inferring panel geometry and stitch topology despite challenging imaging conditions. In contrast, SewFormer fails on 6/7 samples due to garbled pattern outputs, while GarmentX fails on 3/7 samples, struggling with pants and complex dresses. LGM produces plausible 3D meshes but lacks sewing pattern output entirely. Extended evaluation on 524 in-the-wild samples with comprehensive baseline comparisons is provided in the supplementary material.

## 6 Conclusion

We have presented PatternGSL, a structured representation for image-to-pattern reconstruction, along with PatternGSLData, a large-scale dataset of 300K paired samples for supervised VLM training. PatternGSL encodes geometric and topological sewing information in a compact format, achieving a 2D Chamfer distance of 5.78 mm and 99.2% draping success. The explicit parameterization further enables pattern-level editing applications – including panel scaling, curve adjustment, and component editing – while preserving compatibility with the deterministic decoding and simulation pipeline.

*Limitations and Future Work.* Our current evaluation covers an empirical topology range of 2–37 panels; higher panel counts and longer stitch sequences may require stronger long-context generation or hierarchical decoding. The synthesized back view is useful for hidden regions, and our robustness analysis shows graceful degradation when this cue is missing or inconsistent, but severe front-back ambiguity remains challenging. The current simulator also focuses on closed garments and static draping. Future work includes reducing seam and topology failures directly during generation, extending to open-boundary garments, and modeling dynamic cloth behavior.

## References


Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. In *NeurIPS*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966* (2023).

Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. 2019. Multi-garment net: Learning to dress 3D people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5420–5430.

Shang Chai, Liansheng Zhuang, and Fengying Yan. 2023. LayoutDM: Transformer-based diffusion model for layout generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 18349–18358.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. 2021. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852* (2021).

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. *See https://vicuna.lmsys.org* (2023).

Enric Corona, Albert Pumarola, Guillem Alenya, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. SMPLicit: Topology-aware generative model for clothed people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11875–11885.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (2021).

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. 2022. GET3D: A generative model of high quality 3D textured shapes learned from images. In *Advances in Neural Information Processing Systems*, Vol. 35. 31841–31854.

Peng Guan, Lorraine Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. 2012. DRAPE: Dressing any person. *ACM Transactions on Graphics* 31, 4 (2012), 35:1–35:10.

Jiachen Guo, Jiaqi Chen, Wenqian Chen, Zhihao Sun, Linghao Li, Borong Zhao, Lingjie Zhu, Xinyi Wang, and Qiang Liu. 2025. GarmentX: Autoregressive parametric representations for high-fidelity 3D garment generation. *arXiv preprint arXiv:2504.20409* (2025).

Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. 2021. ARCH++: Animation-ready clothed human reconstruction revisited. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11046–11056.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Vol. 33. 6840–6851.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685* (2021).

Yangyi Huang, Hongwei Yi, Yuliang Xiu, Tingting Liao, Jiaxiang Tang, Deng Cai, and Justus Thies. 2024. TeCH: Text-guided reconstruction of lifelike clothed humans. *arXiv preprint arXiv:2308.08545* (2024).

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. 2023. LayoutDM: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10167–10176.

Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. 2020. BCNet: Learning body and cloth shape from a single image. In *Proceedings of the European Conference on Computer Vision*. 18–35.

Maria Korosteleva and Sung-Hee Lee. 2022. NeuralTailor: Reconstructing sewing pattern structures from 3D point clouds of garments. In *ACM Transactions on Graphics*, Vol. 41. 65:1–65:16.

Maria Korosteleva and Sung-Hee Lee. 2023. GarmentCode: Programming parametric sewing patterns. *ACM Transactions on Graphics* 42, 6 (2023), 215:1–215:14.

Cheng Li, Min Tang, Ruofeng Tong, Ming Cai, Jieyi Zhao, and Dinesh Manocha. 2020. P-Cloth: Interactive complex cloth simulation on multi-GPU systems using dynamic matrix assembly and pipelined implicit integrators. *ACM Transactions on Graphics* 39, 6 (2020), 180:1–180:15.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv preprint arXiv:2301.12597* (2023).

Xiang Li et al. 2024. GarmentImage: Raster encoding of garment sewing patterns with diverse topologies. *arXiv preprint* (2024).

Xinyu Li, Qi Yao, and Yuanda Wang. 2025a. GarmentDiffusion: 3D garment sewing pattern generation with multimodal diffusion transformers. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1458–1466.

Xuan Li, Chang Yu, Wenxin Du, Ying Jiang, Tianyi Xie, Yunuo Chen, Yin Yang, and Chenfanfu Jiang. 2025b. Dress-1-to-3: Single image to simulation-ready 3D outfit with diffusion prior and differentiable physics. *ACM Transactions on Graphics* 44, 4 (2025).

Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. 2022. DiffCloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics* 42, 1 (2022), 2:1–2:20.

Junbang Liang, Ming C Lin, and Vladlen Koltun. 2019. Differentiable cloth simulation for inverse problems. In *Advances in Neural Information Processing Systems*, Vol. 32.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-resolution text-to-3D content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 300–309.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. In *arXiv preprint arXiv:2401.05267*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *Advances in neural information processing systems* 36 (2023), 34892–34916.

Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. 2023c. Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on*

*Graphics* 42, 6 (2023), 218:1–218:15.

Ruoshi Liu, Nikhila Ravi, Yuanzhen Wu, Christoph Feichtenhofer, Trevor Darrell, and Alexander C Berg. 2023b. Zero-1-to-3: Zero-shot one image to 3D object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9298–9309.

Yufei Liu, Junshu Tang, Zheng Chu, et al. 2024b. ClotheDreamer: Text-guided garment generation with 3D Gaussians. *arXiv preprint arXiv:2406.16815* (2024).

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics* 34, 6 (2015), 248:1–248:16.

Zhongjin Luo, Haolin Liu, Chenghong Li, et al. 2024. GarVerseLOD: High-fidelity 3D garment reconstruction from a single in-the-wild image using a dataset with levels of details. *ACM Transactions on Graphics* 43, 6 (2024).

Rahul Narain, Armin Samii, and James F O'Brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics* 31, 6 (2012), 152:1–152:10.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. 2019. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10975–10985.

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. 2017. ClothCap: Seamless 4D clothing capture and retargeting. In *ACM Transactions on Graphics*, Vol. 36. 73:1–73:15.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D diffusion. In *International Conference on Learning Representations*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. In *JMLR*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, Vol. 35. 36479–36494.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2304–2314.

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 84–93.

Igor Santesteban, Miguel A Otaduy, and Dan Casas. 2019. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, Vol. 38. 355–366.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. 2024. MVDream: Multi-view diffusion for 3D generation. In *International Conference on Learning Representations*.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Siber, Thomas Brox, 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*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021a. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021b. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Astitva Srivastava, Pranav Manu, Amit Raj, Varun Jampani, and Avinash Sharma. 2024. WordRobe: Text-guided generation of textured 3D garments. *arXiv preprint arXiv:2403.17541* (2024).

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024. LGM: Large multi-view Gaussian model for high-resolution 3D content creation. In *Proceedings of the European Conference on Computer Vision*.

Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. A GPU-based streaming algorithm for high-resolution cloth simulation. *Computer Graphics Forum* 37, 7 (2018), 21–32.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. 2023. Human motion diffusion model. In *International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. 2020. Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum*, Vol. 39. 145–156.

Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. 2022. ICON: Implicit clothed humans obtained from normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13296–13306.

Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. 2020. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6184–6193.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. PhysDiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16010–16021.

Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. 2020. DeepFashion3D: A dataset and benchmark for 3D garment reconstruction from single images. In *Proceedings of the European Conference on Computer Vision*. 512–530.


Fig. 4. Qualitative comparison with baselines. For each sample, we show 3D draping results and 2D sewing patterns. Our method produces accurate patterns with realistic draping across diverse garment types. GarmentImage frequently fails to drape correctly due to imprecise geometry. SewFormer fails on complex topologies (tops 4 and 5). LGM generates plausible 3D meshes but only without sewing patterns.

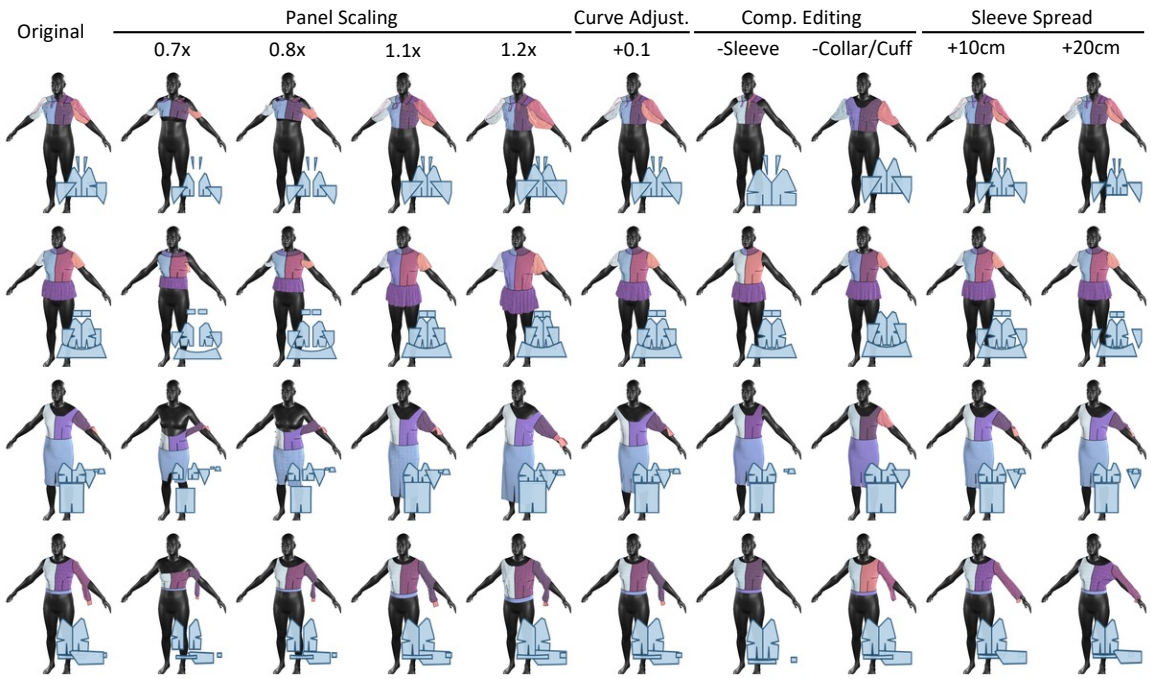


Fig. 5. Pattern editing via direct PatternGSLmanipulation. Each row shows one garment undergoing systematic edits: *Panel Scaling* (0.7×–1.2×) uniformly resizes all panels; *Curve Adjustment* reshapes boundaries via Bézier curvature; *Component Removal* eliminates semantic parts (sleeves or collars/cuffs); *Sleeve Spread* (+10cm, +20cm) adjusts panel layout by spreading sleeves outward. Inset patterns visualize the 2D sewing layout. All 40 edited configurations produce valid, simulation-ready outputs with 100% draping success, demonstrating that our explicit parameterization preserves geometric consistency under arbitrary modifications.

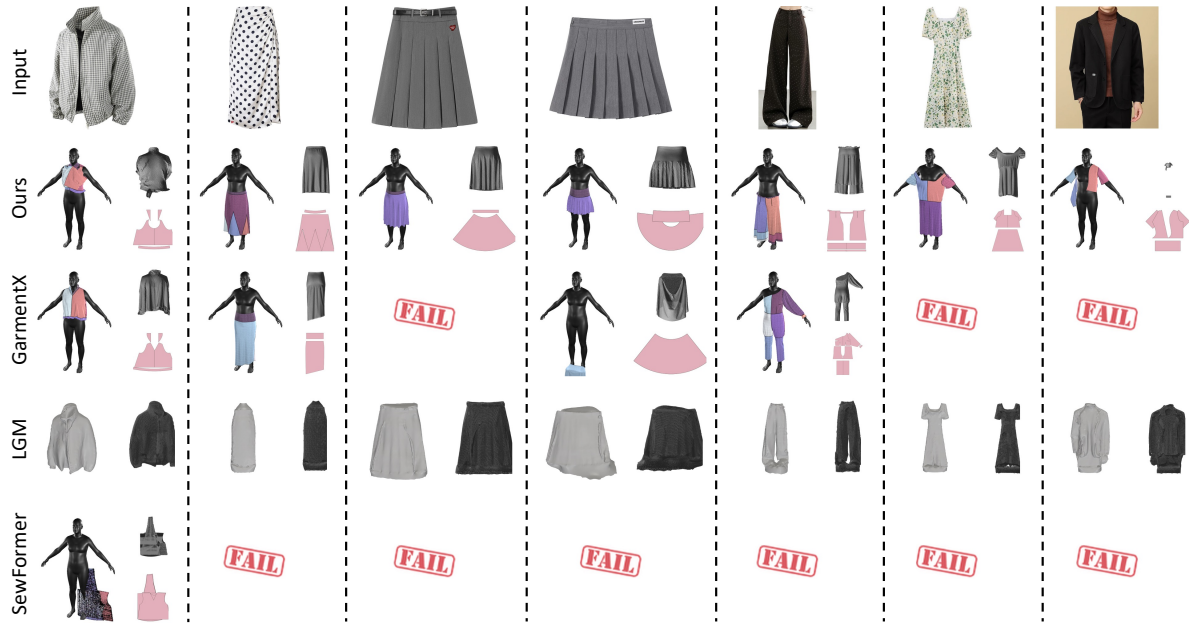


Fig. 6. In-the-wild generalization across seven garment samples. Each column shows an input image (row 1) and results from four methods: Ours (row 2), GarmentX (row 3), LGM (row 4), and SewFormer (row 5). For pattern-based methods, we display both 3D draping and 2D sewing pattern; LGM outputs only 3D meshes without patterns. "FAIL" indicates simulation failure. Our method achieves 100% success, while GarmentX fails on 3/7 and SewFormer fails on 6/7 samples. **More qualitative comparisons are in the supplementary materials**.