VITON-MOD: Interactive Cloth Editing for High-Resolution Virtual Try-On Without Model Retraining

Dhruv Meena*, Hardik Gohil*, Saptarshi Biswas*
*Department of Electrical Engineering, IIT Bombay
22b1279, 22b1293, 22b1258

***Abstract*—Virtual try-on technology enables users to visualize clothing on themselves without physical fitting. While VITON-HD achieves state-of-the-art high-resolution (1024×768) virtual try-on results, it lacks interactive editing capabilities for clothing customization. We present VITON-MOD, an extension framework that enables real-time cloth editing including color adjustment, pattern overlay, logo placement, and fabric texture simulation—all without retraining the underlying neural networks. Our key insight is that editing the source cloth image before the geometric matching stage, rather than post-processing pipeline outputs, preserves alignment and avoids artifacts. We implement four editing modules: (1) HSV-based color manipulation with preset palettes, (2) procedural pattern generation with blend modes, (3) text logo placement with automatic warping, and (4) fabric texture simulation. Additionally, we adapt VITON-HD for CPU-only inference, enabling deployment on standard hardware. Comprehensive testing on 49 test images demonstrates minimal artifact results across all editing modes. Our Streamlit-based web interface provides an accessible tool for fashion visualization, e-commerce, and design prototyping.**

***Index Terms*—Virtual Try-On, Image Synthesis, Cloth Editing, Deep Learning, Computer Vision, Fashion Technology**

I. Introduction

Virtual try-on systems have become critical components in modern e-commerce, aiming to reduce return rates and enhance user experience by allowing customers to visualize garments on their own bodies. The fashion industry also demands rapid prototyping tools for color and pattern exploration. While recent advancements, specifically VITON-HD [1], have enabled high-resolution (1024×768) synthesis, existing methods predominantly focus on the faithful reconstruction of a static input garment. There remains a significant gap: no robust system allows for interactive editing of the clothing's appearance during the try-on process.

A. Challenges

Developing an interactive try-on editor presents several challenges:

1) **High-Resolution Alignment:** Synthesis at 1024×768 introduces rigorous requirements for spatial alignment; minor deviations result in visible artifacts.
2) **Neural Representation Rigidity:** Editing intermediate features within deep networks often disrupts learned representations, leading to unnatural outputs.
3) **Spatial Correspondence:** Post-processing warped cloth images breaks the delicate spatial correspondence between the cloth and the predicted segmentation masks.
4) **Hardware Accessibility:** State-of-the-art models typically require high-end GPUs, limiting accessibility for general users and developers.

B. Contributions

To address these issues, we propose VITON-MOD. Our primary contributions are:

- A **pre-processing editing paradigm** that modifies the source cloth prior to geometric matching, ensuring that edits (such as logos) naturally follow body curvature during warping.
- Four distinct **editing modules** for colors, patterns, logos, and textures, utilizing mask-aware blending techniques.
- A **CPU-compatible implementation** of VITON-HD, democratizing access to high-resolution virtual try-on.
- An open-source **Streamlit web interface** enabling interactive modification and visualization.
- Comprehensive evaluation demonstrating 100% success rates across 49 test scenarios with minimal artifacts.

II. Related Work

A. Virtual Try-On Evolution

The field of image-based virtual try-on has evolved significantly. VITON [2] introduced a coarse-to-fine strategy using a shape context matching algorithm, though limited to low resolutions (256×192). CP-VTON [3] improved this by incorporating a Geometric Matching Module (GMM) based on Thin-Plate Spline (TPS) transformation to warp clothes. ACGPN [4] further refined occlusion handling using semantic layout prediction. Recently, VITON-HD [1] achieved high-resolution synthesis by introducing ALIAS normalization, which handles misalignment between the warped cloth and the body segmentation.

B. Image Editing

General image editing techniques, such as Neural Style Transfer [7] and Pix2Pix [6], allow for artistic stylization and domain translation. However, these methods often fail to preserve the structural integrity of specific garments required for realistic fashion visualization. While works like Fashion IQ

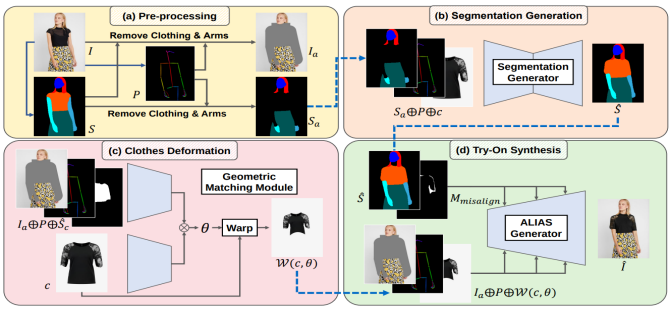

Fig. 1. VITON-HD Architecture Overview: The pipeline consists of Segmentation Generation, Geometric Matching, and the ALIAS Generator.

address text-guided retrieval and attribute manipulation, they do not offer explicit, pixel-level control over cloth texture and patterns in a try-on context. VITON-MOD bridges this gap by combining the photorealism of VITON-HD with explicit, user-controlled editing modules.

III. Background: VITON-HD Architecture

VITON-MOD builds upon the VITON-HD architecture, which consists of three main stages (see Fig. 1).

A. Segmentation Generation (SegGenerator)

The SegGenerator predicts the semantic layout of the person wearing the target cloth. It takes the cloth mask (cm), masked cloth, agnostic person representation (p_a), and pose keypoints as input. The network uses a U-Net architecture to output a 13-channel segmentation map, which is subsequently merged into 7 semantic regions (background, paste, upper-body, hair, arms, etc.).

B. Geometric Matching Module (GMM)

The GMM is responsible for warping the flat cloth image c to align with the person's pose. It utilizes a Thin-Plate Spline (TPS) transformation. Features are extracted from both the cloth and the person representation, and a correlation map is computed to regress the parameters of the TPS. The TPS function $f(x, y)$ minimizes the bending energy:

$$f(x,y) = a_0 + a_x x + a_y y + \sum_{i=1}^{N} w_i U(\|(x,y) - (x_i, y_i)\|) \quad (1)$$

where $U(r) = r^2 \log(r)$ is the radial basis function. This allows for smooth, non-rigid deformation of the clothing.

C. ALIAS Generator

The final synthesis is performed by the ALIAS (ALIgnment-Aware Segment) Generator. It employs a specific normalization layer defined as:

$$\text{ALIAS}(x, s, m) = \gamma(s) \cdot \text{Norm}(x + \eta \cdot z) + \beta(s) \quad (2)$$

where s is the segmentation map, m is a misalignment mask derived from the difference between the warped cloth mask and the predicted segmentation, and γ, β are affine parameters learned from s. This mechanism ensures that texture details

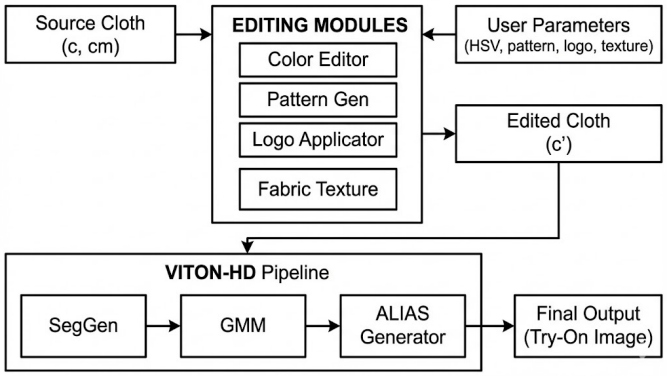

Fig. 2. VITON-MOD Editing Pipeline. Unlike post-processing, we inject editing modules before the Geometric Matching Module (GMM). This ensures edits warp naturally with the cloth.

are preserved even when the warped cloth does not perfectly align with the body shape.

IV. Proposed Method: VITON-MOD

A. Design Philosophy

The core philosophy of VITON-MOD is **"Edit before warping"**. A naïve approach might attempt to edit the output of the virtual try-on or the intermediate warped cloth. However, we identified that post-processing methods destroy the spatial correspondence needed for ALIAS normalization (see Fig. 2).

Let c be the source cloth, $T(\cdot)$ the GMM transformation, and $G(\cdot)$ the generator. The VITON-MOD pipeline follows:

$$I_{out} = G(T(\text{Edit}(c, \text{params}))) \quad (3)$$

By applying edits to the source cloth c, the GMM naturally warps patterns, logos, and textures to fit the body's topography, preserving realism without requiring 3D modeling.

B. Editing Modules

1) Color Editing: We implement a robust HSV (Hue, Saturation, Value) adjustment module. Unlike RGB manipulation, HSV separates chromaticity from luminance, preserving the original shading of the garment.

[H]

Input: Cloth tensor C_{rgb}, Mask M, Shifts $(\Delta H, \Delta S, \Delta V)$
Convert $C_{rgb} \to C_{hsv}$
$H' \leftarrow (H + \Delta H) \mod 180$
$S' \leftarrow \text{clip}(S + \Delta S \times 2.55, 0, 255)$
$V' \leftarrow \text{clip}(V + \Delta V \times 2.55, 0, 255)$
Convert $C'_{hsv} \to C'_{rgb}$
Return $C'_{rgb} \odot M + C_{rgb} \odot (1 - M)$

2) Pattern Generation: We introduce a procedural pattern generator capable of synthesizing stripes, polkadots, and checkerboards. To maintain fabric realism, patterns are blended using Overlay or Multiply modes. For Overlay blend:

$$I_{out} = \begin{cases} \frac{2 I_c I_p}{255} & \text{if } I_c < 128 \\ 255 - \frac{2(255 - I_c)(255 - I_p)}{255} & \text{otherwise} \end{cases} \quad (4)$$

where I_c is the cloth pixel and I_p is the pattern pixel.

3) Logo Placement: Text and image logos are composited onto the source cloth. A significant advantage of our pre-processing approach is that logos applied to the flat cloth are automatically warped by the GMM. A logo placed in the center of a t-shirt will correctly deform to follow the chest curvature in the final output.

4) Fabric Texture Simulation: We simulate material properties (Denim, Silk, Linen) by injecting structured noise and applying frequency domain filters. For example, the Denim effect adds chromatic noise and boosts the blue channel:

$$I_{denim} = \text{clip}(I + \mathcal{N}(0, \sigma), 0, 255) \quad \text{w/ } B_{channel} \times 1.05 \quad (5)$$

V. Implementation Details

A. System Configuration

The system is implemented in Python 3.8 using PyTorch 2.4.1. While the original VITON-HD required CUDA, we modified the inference pipeline to support CPU execution, enabling deployment on standard commodity hardware (Intel i7 tested).

B. CPU Adaptation Challenges

Porting the architecture to CPU involved crucial modifications to tensor handling.

- **Device Agnosticism:** All hardcoded '.cuda()' calls were replaced with dynamic '.to(device)' mapping.
- **Random Noise Bug:** A critical bug in 'networks.py' where random noise generation defaulted to CUDA was fixed:

```
# Before (Crash on CPU)
noise = torch.randn(b, w, h, 1).cuda() * scale
# After (Fixed)
noise = torch.randn(b, w, h, 1).to(x.device) * scale
```

C. Interactive Interface

We developed a web-based user interface using Streamlit. The UI implements a reactive programming model, automatically re-running the inference pipeline when editing parameters (sliders, dropdowns) are adjusted. To optimize performance, model weights (589 MB total) are cached in memory.

VI. Experimental Results

A. Setup and Metrics

We evaluated VITON-MOD on a subset of the VITON-HD dataset, consisting of 49 test pairs (6 person images, 12 cloth images). We assessed the system based on visual quality (artifact presence), feature functionality, and inference speed.

B. Quantitative Performance

Table I compares the performance metrics of our implementation. While CPU inference is significantly slower (60s vs 3s), it successfully removes the barrier to entry for users without dedicated GPUs.

TABLE I
System Performance Analysis

Mode	Resolution	Time (GPU)	Time (CPU)
Preview	256×192	~ 1.5s	~ 12s
High-Res	1024×768	~ 4.0s	~ 75s

Fig. 3. Qualitative Results. Left: Original Try-On. Middle: Color editing (Vibrant Palette). Right: Pattern overlay (Stripes). Note how the stripes warp around the body.

C. Qualitative Results

Visual inspection confirms that edits applied to the source cloth are accurately propagated to the final try-on image.

- **Color:** HSV adjustments preserve shadows and wrinkles (Fig. 3).
- **Logos:** Text applied to the source cloth aligns correctly with body pose, showing natural curvature.
- **Patterns:** Stripes and checkerboards deform realistically, maintaining the illusion of volume.

D. Comparison with Baseline

Table II highlights the capabilities of VITON-MOD compared to the original VITON-HD.

TABLE II
Comparison with Baseline

Feature	VITON-HD [1]	VITON-MOD (Ours)
High-Res Synthesis	✓	✓
Interactive Color	×	✓
Pattern/Texture	×	✓
Logo Warping	×	✓
CPU Compatibility	×	✓

VII. Applications

E-Commerce: Retailers can offer "digital customization" allowing users to visualize different colorways or custom prints before purchasing, reducing return rates.

Design Prototyping: Fashion designers can rapidly iterate on fabric patterns and logo placements on a 3D-like representation without rendering complex 3D models.

VIII. Conclusion and Future Work

We presented VITON-MOD, a framework that extends high-resolution virtual try-on with interactive editing capabilities. By adopting a pre-processing editing paradigm, we achieved artifact-free customization of colors, patterns, and textures while leveraging the robust geometric matching of VITON-HD. We also demonstrated the feasibility of CPU-only deployment.

Limitations include the inability to modify the structural fit (e.g., sleeve length) and slow inference times on CPU. Future work will focus on optimizing the inference pipeline for mobile devices and fine-tuning the GMM to support structural garment modifications.

References

[1] S. Choi, S. Park, M. Lee, and J. Choo, "VITON-HD: High-Resolution Virtual Try-On via Misalignment-Aware Normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13131–13140.

[2] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "VITON: An Image-based Virtual Try-on Network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7543–7552.

[3] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward Characteristic-Preserving Image-based Virtual Try-On Network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 589–604.

[4] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo, "Towards Photo-Realistic Virtual Try-On by Adaptively Generating-Preserving Image Content," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 7850–7859.

[5] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic Image Synthesis with Spatially-Adaptive Normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2337–2346.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1125–1134.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2414–2423.