LAYEREDFASHION: RECOVERING COMPLETE GARMENT LAYERS THROUGH LAYERWISE IMAGE DECOMPOSITION

Anonymous authors
Paper under double-blind review

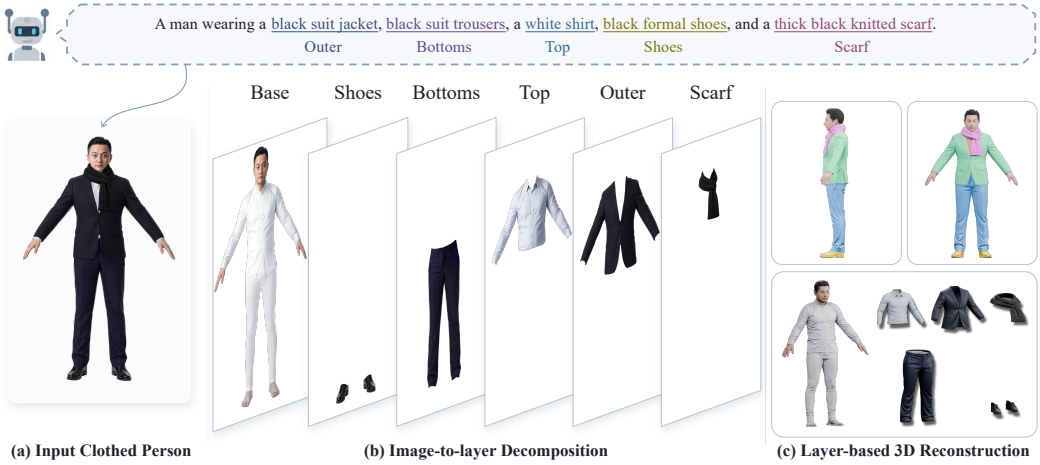

Figure 1: **Human clothing represented as aligned garment layers.** Recovered RGBA garments and a separate person underlay support garment-level editing and layered 3D reuse, illustrated by semantic-color views and textured meshes.

ABSTRACT

A clothed person is a stack of garments, yet human images are usually processed as flat pixels. We present LayeredFashion, which decomposes a clothed-person image and outfit description into an editable PSD-style stack of a variable number of named RGBA garments over a person underlay. We reverse dressing: *Extraction* recovers each garment, and *Restoration* reconstructs the clothing or body beneath it for the next step. To correct missing regions, false positives, and boundary errors, we introduce the *RGB-Semantic Alpha Refiner* (RSAR), combining current-image RGB, garment semantics, and whole-image context to refine alpha while preserving RGB. We train extraction and restoration LoRAs on reference and model-generated intermediate states to accommodate inherited errors. For stepwise supervision, we pair over 22K outfits with nearly 100K complete garment layers, together with person underlays and composition-consistent intermediate images. We achieve the best mean among compared general-purpose decomposition methods on all twelve garment, underlay, and recomposition metrics. The editable layers also support text-to-PSD composition and provide a basis for downstream layered 3D reconstruction.

1 INTRODUCTION

A shirt remains whole beneath a blazer, yet human parsing labels only visible regions (Li et al., 2019) and virtual try-on renders flat images (Choi et al., 2024; Chong et al., 2025). Neither provides individually retrievable, editable, reusable garments. We therefore study *clothed-human layer*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

decomposition: recovering a variable number of named RGBA garments and a person underlay from a dressed-person image and outfit description (Figure 1). Our Photoshop Document (PSD)-style stack preserves complete garment appearance, silhouette, and layer order.

Recovery requires plausible hidden content and coherent remaining clothing and body after each removal. Overlapping garments often share color and material, making separation ambiguous. Existing layer decomposition methods (Yin et al., 2026; Rowles et al., 2026; Wang et al., 2026a; Liu et al., 2026) target general images and graphic designs without explicitly addressing these clothing-specific challenges (Figure 4).

To address these challenges, we introduce LayeredFashion, a *reverse-dressing* framework (Figure 2). Our Garment Description Parser reverses the description's bottom-to-top order. At each step, *Extraction* produces a garment latent and initial alpha, which we refine and pair with current-image RGB. *Restoration* then reconstructs the clothing and body beneath it: removing a jacket exposes a reconstructed shirt, and repetition yields the garment stack and person underlay.

To make this process reliable, we address two coupled challenges. First, ambiguous pixel assignments and boundaries can make initial alpha mattes omit target regions or include neighboring garments, compromising saved layers and subsequent restoration. To correct these errors, we introduce an *RGB-Semantic Alpha Refiner* (RSAR), which combines current-image RGB, garment semantics, and whole-image context to refine opacity at native resolution without changing RGB.

Second, *inter-layer dependency* allows alpha errors and restoration artifacts to accumulate across removal steps. We therefore decouple saved-layer opacity from a cumulative image-repair region to repair earlier residuals without enlarging garment alpha or changing pixels outside that region. To handle inherited errors, we train extraction and restoration LoRA adapters on reference and model-produced states, exposing both stages to imperfect inputs absent from clean-reference training.

To supervise each extraction–restoration step, we need complete garments and underlying images unavailable in dressed photographs. Following generator-based task-specific self-supervision (Cai et al., 2025), we independently generate garments by category on a shared person underlay and extract body-aligned RGBA layers before outfit composition. We then compose accepted layers in sampled orders, saving pre-occlusion references and intermediate restoration targets. The resulting LayeredFashion dataset contains 22,431 outfits and 97,775 garment layers (Section 3.2).

On this dataset, we achieve an alpha IoU of 0.9135 versus 0.8446 for the strongest baseline, with the best reported mean among compared methods on all twelve garment, underlay, and composition metrics (Table 1). Beyond decomposition, the resulting layered representation supports text-to-PSD composition and provides a basis for exploring layered 3D character reconstruction.

Our contributions are as follows:

- To the best of our knowledge, we are the first to formulate clothed-human layer decomposition as a PSD decomposition problem that jointly recovers complete, body-aligned RGBA garments and a separately completed person underlay in an editable 2D stack.
- We introduce reverse dressing with garment alpha decoupled from the cumulative repair region, training extraction and restoration LoRAs on reference and generated intermediate states to address inter-layer dependencies.
- We introduce a novel RSAR module that uses current-image RGB, garment semantics, and whole-image context to correct pixel assignments and refine alpha boundaries without changing RGB.
- We build LayeredFashion with varied garment counts, combinations, and layer orders, pairing complete RGBA garments and person underlays with consistent intermediate composites.

2 Related Work

Multi-layer Image Generation. Layered image generation synthesizes RGBA content (Zhang & Agrawala, 2024; Huang et al., 2026) or text-conditioned foreground layers, masks, and backgrounds (Huang et al., 2024). Other methods decompose generated images (Kang et al., 2025) or existing composites, separating superimposed signals (Zou et al., 2020) and completing occluded content using order and boundary cues (Zhan et al., 2020; Ke et al., 2021). RGBA methods recover complete layers (Yin et al., 2026; Rowles et al., 2026; Chen et al., 2026a; Wang et al., 2026a; Chen et al., 2026b), with broader formulations modeling object-associated effects and partial transparency (Yang

et al., 2025; Wang et al., 2026b). Controlled granularity and layout (Suzuki et al., 2025; Liu et al., 2025) and individually addressable layers (Tang et al., 2026; Li et al., 2026) enable editing.

However, sequential decomposition can propagate errors through restored inputs. CSDNet jointly trains instance segmentation and scene completion on completed intermediate images to address completion artifacts (Zheng et al., 2021). Successive extraction and erasure share this dependency (Liu et al., 2026). We therefore separate saved-layer opacity from the cumulative region used for restoration and train on model-produced removal states. Complementing existing resources (Tudosiu et al., 2024; Chen et al., 2026d), we generate body-aligned garments before composition and pair them with stepwise restoration targets.

Multi-layer Clothing Decomposition. Human parsing labels visible body and garment regions (Gong et al., 2017; 2019; Yang et al., 2019), whereas human de-occlusion completes a person as one object (Zhou et al., 2021). For independent reuse, illustrated-character methods use semantic layers for interleaved drawing and animation (Lin et al., 2026; Chen et al., 2026c). Single-view 3D methods recover geometry and appearance (Xiu et al., 2023; Li et al., 2025; Long et al., 2024), including separable garment surfaces (Aggarwal et al., 2022) and clothing/body geometry (Nam et al., 2025); scan-based layered assets also complete hidden geometry and appearance (Kim et al., 2024b). Animation, deformation, and simulation motivate separate body meshes and clothing radiance fields (Feng et al., 2022), garment templates with pose-dependent wrinkles (Zhang et al., 2023), and editable sewing patterns (Bian et al., 2025). We instead target independent image editing, recovering each named garment as a complete, image-aligned RGBA layer over a separate person underlay with explicit compositing order.

Image Restoration. Matting estimates continuous opacity (Xu et al., 2017; Lu et al., 2019), sometimes jointly recovering foreground appearance (Hou & Liu, 2019). Pretrained features and detail modules improve opacity estimation (Yao et al., 2024), while visual or linguistic prompts guide matte selection and refinement (Li et al., 2024a). In contrast, inpainting synthesizes missing RGB (Yu et al., 2018; 2019; Suvorov et al., 2022; Li et al., 2022); task prompts distinguish completion, insertion, and removal (Zhuang et al., 2024), and harmonization aligns foreground and background appearance (Dalva et al., 2025). In our pipeline, we use RSAR to refine alpha from current-image RGB and semantic context without changing RGB, while Restoration reconstructs underlying clothing and body. We couple these stages recurrently because each restored image feeds the next extraction.

Virtual Try-On. Virtual try-on renders dressed people from person and garment inputs (Han et al., 2018), preserving appearance through flow-based transfer, content preservation, and alignment (Han et al., 2019; Ge et al., 2021; Yang et al., 2020; Choi et al., 2021) or diffusion-based synthesis (Zhu et al., 2023; Kim et al., 2024a; Choi et al., 2024; Chong et al., 2025). Multiple garments additionally require control over wearing order and drape. Recurrent dressing and rendering policies manage overlap (Cui et al., 2021; Li et al., 2024b), while text-conditioned layouts support multi-garment try-on and editing (Zhu et al., 2024). We study the reverse direction: recovering editable RGBA garments from dressed images. These assets also support T2P composition as an editable stack.

3 Problem Formulation and Data Curation

3.1 Clothing-layer representation

For image-to-PSD (I2P), we recover a stack of named, individually editable full-canvas RGBA garments and a person underlay from a dressed-person image and outfit description. Each garment is body-aligned and includes plausibly completed occluded regions. The Photoshop Document (PSD) output contains two to seven layers, including the underlay.

To compose these garments, we follow the prompt's bottom-to-top layer order. Our Garment Description Parser extracts this order, garment categories, and verbatim descriptions. We support six optional categories: shoes, bottoms, inner top, outerwear, scarf, and hat.

We combine the layers using source-over compositing (Porter & Duff, 1984). Different stacks can render the same image, so recomposition alone cannot establish garment, underlay, or order accuracy.

We also use this representation for text-to-PSD (T2P), building outfits one garment at a time on an independently supplied person image rather than I2P predictions.

3.2 THE LAYEREDFASHION DATASET

Existing human-parsing and virtual try-on datasets provide visible garment labels or person–garment pairs (Gong et al., 2017; Choi et al., 2021). Our task instead requires complete, body-aligned garment layers and an underlying image for each removal step. To fill this supervision gap, we construct LayeredFashion with one to six garment layers, varying category combinations, compositing orders, styles, materials, patterns, and fits.

We first remove backgrounds from studio photographs to obtain person underlays. For each category, we independently prompt an image generation model to dress the same underlay with the target garment. We localize and segment each garment with Grounding DINO and SAM 2 (Liu et al., 2024; Ravi et al., 2025), then screen its contour for body alignment. We also manually inspect every sample for garment-description consistency, body alignment, and segmentation completeness. For each dataset record, we sample one of the six permutations of shoes, bottoms, and inner top, followed by outerwear, scarf, and hat, omitting absent categories. Prompts follow the sampled order (Appendix A.1). We then compose accepted garments into casual, streetwear, athletic, and formal outfits, saving each intermediate image as an exact next-step target for reverse dressing. The individual RGBA garments provide pre-occlusion references. We count shoes, scarves, and hats as separate layers. Appendix A details construction, full/visible masks, annotations, and acceptance checks.

This process yields a core of 22,431 person-outfit samples and 97,775 garment layers, averaging 4.36 per sample. We allocate 16,431 samples to training and 3,000 each to validation and testing. To separate training and evaluation, we keep both body underlays and garment assets disjoint across splits, allowing reuse and Cartesian recombination only within a split.

We also include different age groups and body builds to complement the variation in clothing. Appendix A reports the demographic and garment distributions of the profiled subset.

4 METHOD

We alternate outermost-garment *Extraction* and image *Restoration* to reverse dressing (Figure 2). Each restored image feeds the next extraction, coupling alpha and image errors. To stabilize this recursion, we refine alpha with our RGB-Semantic Alpha Refiner (RSAR), constrain image repair separately, and train recurrent stages on model-produced intermediate images.

4.1 GARMENT DESCRIPTION PARSER

To target one garment at a time, we predict its category and select its verbatim description using frozen FLUX.1-Kontext T5 features (Labs et al., 2025). We reverse the prompt's bottom-to-top layer order for image-to-PSD extraction and follow it for text-to-PSD composition. Inference uses text without target images, category labels, masks, or lower composites (Appendix E).

4.2 EXTRACTION

Given the selected description, we identify the target garment in the current RGB image I_{t-1} at removal step t, with I_0 the dressed input. We adapt a frozen FLUX backbone with an extraction LoRA (Hu et al., 2021) to produce a garment latent conditioned on the selected text. To locate the garment without a target mask, we also use category-specific coordinate priors, garment-length cues, and local image content (Appendix C). We then decode its initial opacity matte $\widetilde{\alpha}_t \in [0, 1]$ with a dedicated alpha decoder. Specialized transparency decoding is also used for character and referring-layer decomposition (Lin et al., 2026; Chen et al., 2026a).

The initial matte can miss garment regions, include neighboring garments, or lose fine boundaries. To correct these errors, we introduce the RGB-Semantic Alpha Refiner (RSAR). We first correct pixel membership across the full image, then refine opacity at native resolution. We use current-image RGB for appearance and boundaries, and garment category and text to identify the target. For whole-image

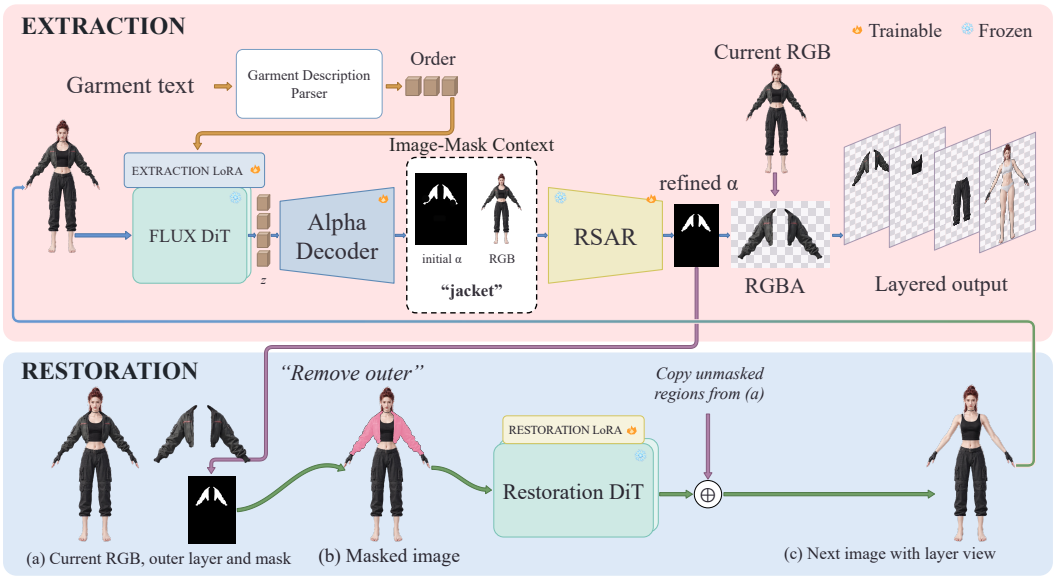

Figure 2: **Image-to-PSD pipeline.** We extract garments in reverse parser order, refining alpha with RSAR and preserving current-image RGB. *Restoration* exposes the next garment by updating only the pink region S_t; the right-hand stack accumulates layers. Flames/snowflakes mark trainable/frozen components during training; all inference weights are fixed.

context, a frozen coarse semantic branch reads RGB, initial alpha, and frozen SigLIP2 features (Tschannen et al., 2025).

We implement the decision stage as a U-Net D_ϕ, where ϕ denotes RSAR's trainable parameters. Let r_t encode the garment category, e_t the outfit and garment descriptions, and c_t coarse-matte and visual context. We predict per-pixel probabilities of COPY, ADD, and REMOVE, which retain the initial opacity, set it to one, and set it to zero, respectively:

$$(p_t^{\text{copy}}, p_t^{\text{add}}, p_t^{\text{remove}}) = \operatorname{softmax} D_\phi(I_{t-1}, \widetilde{\alpha}_t, r_t, e_t, c_t), \tag{1}$$

$$\overline{\alpha}_t = p_t^{\text{copy}} \widetilde{\alpha}_t + p_t^{\text{add}}, \qquad \alpha_t = \sigma(\operatorname{logit}(\overline{\alpha}_t) + \Delta_\phi)\,. \tag{2}$$

To refine $\overline{\alpha}_t$, we predict a bounded logit residual Δ_ϕ from RGB, the initial and decision-stage mattes, and semantic context. Here σ is the sigmoid function and logit maps opacity to log-odds. Both stages cover the full canvas and change only alpha (Appendix B.2). Finally, we combine the refined alpha with current-image RGB to form the garment layer $G_t = (I_{t-1}, \alpha_t)$ without postprocessing.

4.3 RESTORATION

Once we have extracted a garment, we reconstruct the clothing and body beneath it to expose the next target. We condition frozen FLUX and a step-shared restoration LoRA on the current image I_{t-1}, a target-removal instruction, and remaining-layer context to propose $\widetilde{I}_t$.

Repairing only the current garment can leave residual content from earlier removal steps. We therefore separate garment opacity from a binary repair region S_t. We threshold the refined alpha into a binary garment region M_t, take its union with all preceding garment regions, and apply fixed boundary treatment to obtain S_t (Appendix C.2). No separate category-prior mask enters this union. We can thus repair these residual regions without enlarging garment alpha.

To preserve the surroundings while updating this repair region, we use the same S_t to condition restoration and to combine its proposal with the current image:

$$I_t = S_t \odot \widetilde{I}_t + (1 - S_t) \odot I_{t-1}. \tag{3}$$

Here $\odot$ denotes pixelwise multiplication broadcast over RGB channels. We copy every pixel outside S_t exactly and pass I_t to the next extraction. After removing all T garments, we obtain person alpha

α_B from a segmentation module and form the underlay $B = (I_T, \alpha_B)$. We arrange the extracted garments above it in the parser's bottom-to-top order. Training and inference use the same region construction and compositing rules. Inference uses no reference alpha, lower composite, or target image.

4.4 TEXT-TO-PSD COMPOSITION

Our extractor also supports text-guided outfit composition (Figure 3). Starting from a fixed, independently supplied person underlay, we add garments in the parser's forward order. We use FLUX to propose each dressed image, then the frozen extractor, alpha decoder, and RSAR to isolate its new garment. We recompose the extracted RGBA garments over the underlay so that the next input matches the saved layer stack. This builds an editable stack without Restoration. We compare the resulting garment layers and final compositions in Section 6.3 and provide further examples in Appendix H.2.

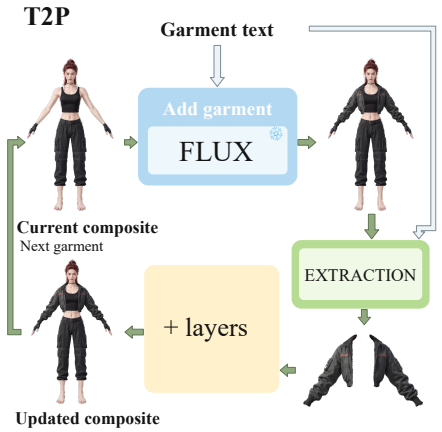

Figure 3: **Text-to-PSD composition.** Each extracted garment joins the saved stack; recomposition drives the next generation.

4.5 TRAINING

Because later steps inherit restoration errors, we mix reference images with intermediate images generated by the current cascade to train each LoRA. We refresh these on-policy inputs as the trainable component changes, retaining reference garments and lower composites as targets.

We train components sequentially. We first train the extraction LoRA using frozen early-stage alpha-decoder and restoration checkpoints in its cascade. We then train the alpha decoder on cached latents from the frozen extractor and reference garments encoded by the frozen VAE. For blank-target latents, we penalize nontransparent outputs. To learn matte correction, RSAR uses reference or perturbed mattes and cached extraction errors with dataset-alpha supervision. We train only its decision and detail branches, keeping its coarse and vision modules frozen. Neither alpha stage refreshes online trajectories. Finally, we freeze the extraction components, including RSAR, and train the restoration LoRA using the full decomposition cascade (Appendix B.3).

5 EXPERIMENTS

To evaluate both the recovered layers and their recomposition, we use a fixed, stratified core-test cohort of 200 people with 780 garment layers. We use the same records for quantitative comparisons and supporting text-to-PSD evaluation, with reference assets used only for post-generation matching and scoring. We evaluate decomposition at a resolution of 1024×1024.

Evaluation Metrics. We use twelve metrics to evaluate stack completeness, garment quality, and reconstruction. For completeness, we measure garment recall and Exact order, which requires all garments in prompt-defined order and includes incomplete outputs in the denominator. For individual garments, we evaluate shape and contours with Alpha IoU and Boundary F-score, and opacity outside reference garments with Leakage. Visible- and occluded-region RGB MAE then assess appearance in observed and constructed hidden regions. Finally, we assess recovered underlays and recomposed stacks using RGBA MAE. We also report LPIPS for perceptual similarity and CLIPScore and VQAScore for image–text consistency. To distinguish alpha errors from recurrent-state errors, we evaluate seven ablations against the full model using four error metrics. Full-canvas Alpha MAE and boundary-alpha MAE within a reference-defined band measure continuous opacity, unlike thresholded alpha IoU. Outside-support RGB MAE measures preservation beyond the reference scoring region, while trajectory drift tracks errors propagated between extractions. Appendices D.1.1 and D define matching, aggregation, and metrics.

Baselines. We compare Qwen-Image-Layered (Yin et al., 2026), Stable-Layers (Rowles et al., 2026), RevealLayer (Wang et al., 2026a), and OmniPSD (Liu et al., 2026) with identical dressed images and global prompts, providing automatic boxes to RevealLayer. To test clothing-domain adaptation, we train OmniPSD's Front/Back LoRAs using its official training code on 15,000 people

from our training split. We evaluate this model, denoted OmniPSD (train15k), with our task adapter. GPT6-Astra High appears only in qualitative comparisons.

6 RESULTS

6.1 LAYER DECOMPOSITION



Figure 4: **I2P comparison for two clothed people.** Inputs appear left, with parallel method columns showing garment layers above recompositions over recovered person underlays. Checkerboards indicate transparency; blank slots are unused. GPT6-Astra High is qualitative-only.

Table 1: **Image-to-PSD comparison.** Best means are bold. OmniPSD (train15k) uses our training split and task adapter. Underlay/recomposition use RGBA MAE; see Appendix D.1.1 for matching and scoring.

Method	Recall↑	α IoU↑	Bound. F↑	Visible MAE↓	Occl. MAE↓	Underlay MAE↓
Qwen-Image-Layered	0.9100	0.3923	0.3533	0.1985	0.2550	0.1218
Stable-Layers	0.9056	0.5160	0.4903	0.1222	0.1803	0.0587
RevealLayer (auto boxes)	0.9778	0.7709	0.6541	0.0870	0.2884	0.0625
OmniPSD (train15k)	0.9550	0.8446	0.7441	0.1190	0.2295	0.0240
LayeredFashion	**1.0000**	**0.9135**	**0.8095**	**0.0295**	**0.1565**	**0.0207**

Method	Recomp. MAE↓	LPIPS↓	CLIP Score↑	VQA Score↑	Leakage↓	Exact order↑
Qwen-Image-Layered	0.0159	0.0810	0.2790	0.7990	0.1690	0.1050
Stable-Layers	0.0132	0.0680	0.3010	0.6920	0.1020	0.2000
RevealLayer (auto boxes)	0.0352	0.1300	0.2560	0.7010	0.0580	0.6600
OmniPSD (train15k)	0.0236	0.1940	0.2590	0.7430	0.0390	0.6900
LayeredFashion	**0.0028**	**0.0210**	**0.3380**	**0.8420**	**0.0160**	**1.0000**

We compare recovered garments with their references (Table 1). We achieve 0.9135 alpha IoU versus 0.8446 for the strongest IoU baseline, OmniPSD (train15k). Visible-region MAE is 0.0295

versus RevealLayer's best baseline value of 0.0870. Occluded-region MAE is 0.1565 versus Stable-Layers' 0.1803, measuring agreement with pre-occlusion references. Including underlay recovery and recomposition, we achieve the best mean scores on all 12 metrics.

Despite clothing-domain training (Section 5), OmniPSD (train15k) still has higher leakage (0.039 versus 0.016) and occluded-region MAE (0.2295 versus 0.1565), with lower boundary F-score (0.7441 versus 0.8095). Figure 4 shows cross-layer contamination and residual person content in garment layers. To explain these gains, we next isolate the effects of our components.

6.2 Ablation Studies

To isolate component effects, we compare our full model (Full) with seven ablated variants, changing only the specified component in each case (Table 2). We use this controlled comparison to examine alpha estimation and recurrent-state preservation separately. For alpha estimation, we measure continuous opacity error over all image pixels with full-canvas Alpha MAE. Boundary-alpha MAE restricts this comparison to a reference-defined band around garment contours. Unlike thresholded alpha IoU, both metrics retain fractional opacity differences, allowing us to assess matte errors rather than only binary garment support. We first replace the dedicated alpha decoder with the base RGBA decoder while retaining RSAR. Boundary-alpha MAE increases from 0.1355 to 0.2486. We then bypass RSAR while retaining the alpha decoder: full-canvas Alpha MAE rises from 0.0026 to 0.0768, and trajectory drift rises from 0.0019 to 0.0109.

Table 2: Ablation studies (lower is better). Values use four decimals; ↑ marks selected error increases relative to Full (worse performance). Outside denotes outside-support RGB MAE. See Appendices D and D.3.

Method	α MAE	Bound. α MAE	Outside RGB MAE	Trajectory drift
w/o Alpha Decoder	0.0100↑	0.2486↑	0.0004	0.0020
w/o RSAR	0.0768↑	0.3610↑	0.0004	0.0109↑
w/o recurrent state update	0.0163↑	0.1910↑	0.0004	0.0110↑
w/o support-constrained update	0.0037↑	0.1602↑	0.0051↑	0.0070↑
RSAR w/o RGB evidence	0.0516↑	0.3521↑	0.0004	0.0058↑
RSAR w/o semantic/context evidence	0.0249↑	0.2171↑	0.0018↑	0.0060↑
w/o full-view decision	0.0112↑	0.1824↑	0.0004	0.0021
Full	**0.0026**	**0.1355**	**0.0004**	**0.0019**

Within RSAR, we next examine which inputs and decisions enable this refinement. We replace RGB evidence with a neutral constant in one variant and remove text, category, and visual context in another. Alpha MAE rises to 0.0516 without RGB evidence and to 0.0249 without semantic/context evidence. Thus, both appearance and semantic/context cues contribute to accurate alpha estimation. Having tested these inputs, we assess the full-view decision stage by replacing its output with coarse alpha. Alpha MAE rises to 0.0112 and boundary-alpha MAE to 0.1824. This comparison shows the benefit of learned full-image correction beyond the coarse alpha estimate.

Beyond alpha refinement, we assess whether recurrent updates preserve a stable decomposition trajectory. To test recurrence, we feed the original dressed image into every step instead of the updated composite. We still generate and save restorations in this variant; the change is whether later steps use the updated image. Trajectory drift increases from 0.0019 to 0.0110. This metric measures RGBA error against reference intermediate composites. We then assess the support constraint by removing the update restriction in Eq. 3, so restoration can change pixels outside the repair region. Outside-support RGB MAE rises from 0.0004 to 0.0051, and trajectory drift rises to 0.0070. We compute outside-support RGB MAE outside a fixed reference region, which need not coincide with the predicted repair region used to constrain an individual update. Together, alpha refinement and constrained recurrent updates support both garment quality and decomposition stability.

6.3 Text-to-PSD composition

Beyond decomposition, we build outfits from text. Using the same input person and original prompt, we compare our garment layers and final composition with baselines decomposing a shared Kontext-

generated image (Figure 5). This comparison assesses garment shape and appearance; Appendix H.2 provides further examples.

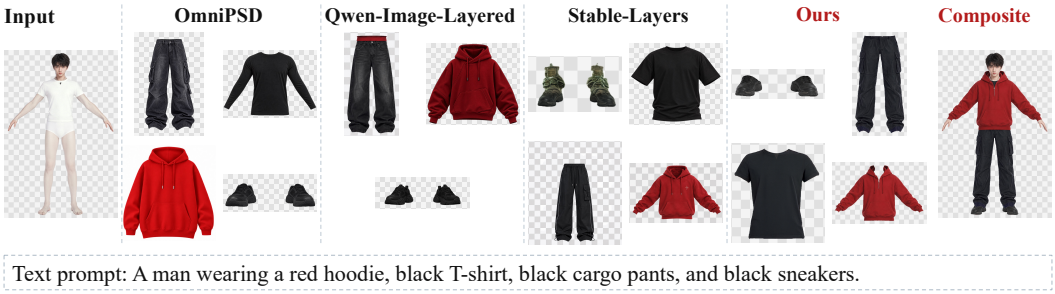

Figure 5: **Text-to-PSD comparison.** All methods use the same input person and original prompt. Layer outputs appear beside the input, with our final composition at the far right. Empty layers are omitted. Display crops are independently scaled for visibility.

6.4 Layer-based 3D reconstruction

Finally, we demonstrate 3D reuse through separate textured meshes of the person and garments for individual inspection and editing (Figure 6; Appendix H.4).



Figure 6: **Layer-based 3D reconstruction.** Inputs accompany category-colored front, side, and back views. Each box contains separate textured meshes of the person underlay and garments.

7 Conclusion

We presented LayeredFashion, coupling extraction and restoration through reverse dressing to recover editable garment layers. To address inter-layer errors, we use alpha refinement, constrained updates, and mixed-state training. Our dataset supplies complete garments and aligned intermediate states. We achieve the best mean on all 12 metrics among compared methods. Beyond decomposition, these layers support text-to-PSD composition and layered 3D reuse.

Limitations. Our current dataset covers a limited range of garment categories. Extending its coverage is an important direction for supporting a broader range of clothing. Our model recovers 2D garment layers aligned with the input view; 3D reconstruction is a downstream application of these layers. A single image does not uniquely determine fully occluded garment regions or the body beneath clothing. We infer these regions from visible context and learned priors, so the completed shapes and textures may differ from the actual hidden appearance. We currently use outfit descriptions to specify garment identities and order. Inferring this structure directly from images is another direction. Finally, sequential extraction and restoration increase inference cost with garment count.

AI USE STATEMENT

In this work, we used generative AI tools in a limited capacity to assist with drafting and revising parts of the manuscript, to provide limited coding assistance, and to generate images used in the dataset-construction workflow described in Section 3. All reported measurements were obtained from experiments, not from AI-generated text. The authors reviewed all AI-assisted content and take responsibility for the final content of this work, including the implementation, measurements, citations, data provenance, and scientific conclusions.

ETHICS STATEMENT

Person-image processing raises privacy, consent, licensing, representation, and misuse concerns. The intended use is research on editable garment layers in person images, not recognition, identity decisions, or claims of garment ownership. Responsible use requires respecting source-image licenses and consent, protecting identifiable information, and assessing risks associated with the intended application. Any public release must specify source permissions, permitted uses, and privacy safeguards, and provide a removal mechanism.

REPRODUCIBILITY STATEMENT

Appendix B specifies model components, training settings, and evaluation checkpoints; Appendix C describes prompt and spatial-control policies; Appendix D defines the evaluation procedures; Appendix G describes baseline protocols and PSD export checks.

REFERENCES

Alakh Aggarwal, Jikai Wang, Steven Hogue, Saifeng Ni, Madhukar Budagavi, and Xiaohu Guo. Layered-garment net: Generating multiple implicit garment layers from a single image. In *Asian Conference on Computer Vision*, pp. 378–395. Springer, 2022.

Siyuan Bian, Chenghao Xu, Yuliang Xiu, Artur Grigorev, Zhen Liu, Cewu Lu, Michael J Black, and Yao Feng. Chatgarment: Garment estimation, generation and editing via large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2924–2934. IEEE, 2025.

Shengqu Cai, Eric Ryan Chan, Yunzhi Zhang, Leonidas Guibas, Jiajun Wu, and Gordon Wetzstein. Diffusion self-distillation for zero-shot customized image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18434–18443. IEEE, 2025.

Fangyi Chen, Yaojie Shen, Lu Xu, Ye Yuan, Shu Zhang, Yulei Niu, and Longyin Wen. Referring layer decomposition. *arXiv preprint arXiv:2602.19358*, 2026a.

Jingxi Chen, Yixiao Zhang, Xiaoye Qian, Zongxia Li, Cornelia Fermuller, Caren Chen, and Yiannis Aloimonos. From inpainting to layer decomposition: Repurposing generative inpainting models for image layer decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16054–16063, 2026b.

Junhao Chen, Jingjia Mao, Dayong Li, Chenghai Li, Saining Zhang, Zhihao Li, Hao Zhao, Yufei Wang, and Ruqi Huang. Bunraku: Turning a single illustration into an editable live2d character. *arXiv preprint arXiv:2607.27348*, 2026c.

Junwen Chen, Heyang Jiang, Keming Wu, Ji Li, Chao Zhang, Keiji Yanai, Dong Chen, and Yuhui Yuan. Prismlayers: Open data for high-quality multi-layer transparent image generative models, 2026d.

Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. *arXiv preprint arXiv:2103.16874*, 2021.

Yisol Choi, Sangkyung Kwak, Kyungmin Lee, Hyungwon Choi, and Jinwoo Shin. Improving diffusion models for authentic virtual try-on in the wild. In *European Conference on Computer Vision*, pp. 206–235, 2024.

Zheng Chong, Xiao Dong, Haoxiang Li, shiyue Zhang, Wenqing Zhang, Hanqing Zhao, xujie zhang, Dongmei Jiang, and Xiaodan Liang. CatVTON: Concatenation is all you need for virtual try-on with diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14618–14627. IEEE, 2021.

Yusuf Dalva, Yijun Li, Qing Liu, Nanxuan Zhao, Jianming Zhang, Zhe Lin, and Pinar Yanardag. Layerfusion: Harmonized multi-layer text-to-image generation with generative priors, 2025.

Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9, 2022.

Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. Parser-free virtual try-on via distilling appearance flows. *arXiv preprint arXiv:2103.04559*, 2021.

Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6757–6765. IEEE, 2017.

Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7442–7451. IEEE, 2019.

Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7543–7552. IEEE, 2018.

Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10471–10480, 2019.

Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4129–4138. IEEE, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Dingbang Huang, Wenbo Li, Yifei Zhao, Xinyu Pan, Yanhong Zeng, and Bo Dai. Psdiffusion: Harmonized multi-layer image generation via layout and appearance alignment. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3233–3242. IEEE, 2026.

Runhui Huang, Kaixin Cai, Jianhua Han, Xiaodan Liang, Renjing Pei, Guansong Lu, Songcen Xu, Wei Zhang, and Hang Xu. Layerdiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model. In *European Conference on Computer Vision*, pp. 144–160. Springer, 2024.

Kyoungkook Kang, Gyujin Sim, Geonung Kim, Donguk Kim, Seungho Nam, and Sunghyun Cho. Layeringdiff: Layered image synthesis via generation, then disassembly with generative knowledge. *arXiv preprint arXiv:2501.01197*, 2025.

Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4018–4027. IEEE, 2021.

Jeongho Kim, Gyojung Gu, Minho Park, Sunghyun Park, and Jaegul Choo. Stable viton: Learning semantic correspondence with latent diffusion model for virtual try-on. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8176–8185. IEEE, 2024a.

Taeksoo Kim, Byungjun Kim, Shunsuke Saito, and Hanbyul Joo. Gala: Generating animatable layered assets from a single scan. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1535–1545. IEEE, 2024b.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1775–1785. IEEE, 2024a.

Kedan Li, Jeffrey Zhang, Shao-Yu Chang, and David Forsyth. Controlling virtual try-on pipeline through rendering policies. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5854–5863. IEEE, 2024b.

Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *CoRR*, abs/1910.09777, 2019.

Peng Li, Wangguandong Zheng, Yuan Liu, Tao Yu, Yangguang Li, Xingqun Qi, Xiaowei Chi, Siyu Xia, Yan-Pei Cao, Wei Xue, et al. Pshuman: Photorealistic single-image 3d human reconstruction using cross-scale multiview diffusion and explicit remeshing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16008–16018. IEEE, 2025.

Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10748–10758. IEEE, 2022.

Zongjian Li, Zhiyuan Yan, Chenxu Bai, Chen Chen, Haoxiang Sun, Shaodong Wang, Feize Wu, Shenghai Yuan, Bin Lin, Zheyuan Liu, et al. Uniworld-design: From pixel generation to layer-native design. *arXiv preprint arXiv:2608.03971*, 2026.

Jian Lin, Chengze Li, Haoyun Qin, Kwun Wang Chan, Yanghua Jin, Hanyuan Liu, Chun Wang Choy, and Xueting Liu. See-through: Single-image layer decomposition for anime characters. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

Cheng Liu, Yiren Song, Haofan Wang, and Mike Zheng Shou. Ominpsd: Layered psd generation with diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4190–4201, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Zihao Liu, Zunnan Xu, Shi Shu, Jun Zhou, Ruicheng Zhang, Zhenchao Tang, and Xiu Li. Controllable layer decomposition for reversible multi-layer image generation. *arXiv preprint arXiv:2511.16249*, 2025.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9970–9980. IEEE, 2024.

Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3265–3274. IEEE, 2019.

Hyeongjin Nam, Donghwan Kim, Jeongtaek Oh, and Kyoung Mu Lee. Decloth: Decomposable 3d cloth and human body reconstruction from a single image. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5636–5645. IEEE, 2025.

Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pp. 253–259, 1984.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Ciara Rowles, Reshinth Adithyan, Nikhil Pinnaparaju, Vikram Voleti, and Mark Boss. Stable-layers: Fine-tuning image layer decomposition models with vlm-scored reinforcement learning. *arXiv preprint arXiv:2605.30257*, 2026.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3172–3182. IEEE, 2022.

Tomoyuki Suzuki, Kang-Jun Liu, Naoto Inoue, and Kota Yamaguchi. Layerd: Decomposing raster graphic designs into layers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17783–17792. IEEE, 2025.

Zhicong Tang, Zhao Zhang, Jingye Chen, Mohan Zhou, Yifan Pu, Yuchi Liu, Yalong Bai, Ethan Smith, and Yuhui Yuan. Mrt: Masked region transformer for layered image generation and editing at scale. *arXiv preprint arXiv:2605.27235*, 2026.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. Mulan: A multi layer annotated dataset for controllable text-to-image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22413–22422. IEEE, 2024.

Binhao Wang, Shihao Zhao, Bo Cheng, Qiuyu Ji, Yuhang Ma, Liebucha Wu, Shanyuan Liu, Dawei Leng, and Yuhui Yin. Reveallayer: Disentangling hidden and visible layers via occlusion-aware image decomposition. *arXiv preprint arXiv:2605.11818*, 2026a.

Zitong Wang, Hang Zhao, Qianyu Zhou, Xuequan Lu, Xiangtai Li, Hao Yang, Bo Yang, and Yiren Song. Diffdecompose: Layer-wise decomposition of alpha-composited images via diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4624–4634, 2026b.

Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J Black. Econ: Explicit clothed humans optimized via normal integration. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 512–523. IEEE, 2023.

Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 311–320. IEEE, 2017.

Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating↔preserving image content. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7847–7856. IEEE, 2020.

Jinrui Yang, Qing Liu, Yijun Li, Soo Ye Kim, Daniil Pakhomov, Mengwei Ren, Jianming Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Generative image layer decomposition with visual effects. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7643–7653. IEEE, 2025.

Lu Yang, Qing Song, Zhihui Wang, and Ming Jiang. Parsing r-cnn for instance-level human analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 364–373. IEEE, 2019.

Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion*, 103:102091, 2024.

Shengming Yin, Zekai Zhang, Zecheng Tang, Kaiyuan Gao, Xiao Xu, Kun Yan, Jiahao Li, Yilei Chen, Yuxiang Chen, Heung-Yeung Shum, et al. Qwen-image-layered: Towards inherent editability via layer decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16196–16205, 2026.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 5505–5514. IEEE, 2018.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4471–4480, 2019.

Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3783–3791. IEEE, 2020.

Hongwen Zhang, Siyou Lin, Ruizhi Shao, Yuxiang Zhang, Zerong Zheng, Han Huang, Yandong Guo, and Yebin Liu. Closet: Modeling clothed humans on continuous surface with explicit template decomposition. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 501–511. IEEE, 2023.

Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Chuanxia Zheng, Duy-Son Dao, Guoxian Song, Tat-Jen Cham, and Jianfei Cai. Visiting the invisible: Layer-by-layer completed scene decomposition. *International Journal of Computer Vision*, 129 (12):3195–3215, 2021.

Qiang Zhou, Shiyin Wang, Yitong Wang, Zilong Huang, and Xinggang Wang. Human de-occlusion: Invisible perception and recovery for humans. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3690–3700. IEEE, 2021.

Luyang Zhu, Dawei Yang, Tyler Zhu, Fitsum Reda, William Chan, Chitwan Saharia, Mohammad Norouzi, and Ira Kemelmacher-Shlizerman. Tryondiffusion: A tale of two unets. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4606–4615. IEEE, 2023.

Luyang Zhu, Yingwei Li, Nan Liu, Hao Peng, Dawei Yang, and Ira Kemelmacher-Shlizerman. M&m vto: Multi-garment virtual try-on and editing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1346–1356. IEEE, 2024.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pp. 195–211. Springer, 2024.

Zhengxia Zou, Sen Lei, Tianyang Shi, Zhenwei Shi, and Jieping Ye. Deep adversarial decomposition: A unified framework for separating superimposed images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12803–12813. IEEE, 2020.

A Dataset Composition and Curation

A.1 Core construction

To obtain paired references for layer recovery, we generate and segment body-aligned garments before composing them into outfits. We retain their pre-composition content as supervision when other garments subsequently occlude them.

All core records follow this workflow. The complete core comprises 22,431 person-outfit records (16,431 / 3,000 / 3,000) and 97,775 garment-layer instances on a shared 1024×1024 canvas with a uniform protocol for annotating garment categories and order. Core totals and splits are given in Section 3.2.

We remove the background from studio model photographs to obtain RGBA person underlays. This step makes the background transparent; it does not remove clothing from the person. The target garments are generated separately in the next stage. We assign a gender category during underlay preparation and use the same person underlay for all garments in an outfit.

We use an image generation model to inpaint each of the six garment categories, shoes, bottoms, inner top, outerwear, scarf, and hat, separately on the shared body underlay. Descriptions are drawn from a fixed, closed prompt pool. The category-specific generations provide candidate assets for later combination; they are not successive edits that accumulate a complete outfit during generation.

Grounding DINO localizes each target garment, and SAM 2 produces candidate masks (Liu et al., 2024; Ravi et al., 2025). The text query changes with the target category, and `multimask_output=True` is enabled. For outerwear, we use a dedicated mask-repair procedure and manually retouch the garment images. We compare each garment contour with the body underlay and discard assets that fail the contour-alignment check. The extracted mask supplies the garment alpha; the generation model does not directly supply the final RGBA asset. We manually inspect every sample for consistency with its garment description, alignment with the person underlay, and segmentation completeness before accepting its garments for composition.

The accessory group is stored as three category-specific sublayers, shoes, scarf, and hat, together with a merged accessory layer. Outfit combinations are formed from the Cartesian product of bottoms, inner-top, outerwear, and accessory choices on a shared body. Layer depth counts the present garment categories individually, including each accessory category. The auxiliary merged accessory layer is neither an additional category nor an additional instance in the garment-layer totals.

For each record, we randomly permute shoes, bottoms, and inner top, and then place outerwear, scarf, and hat above them. The person underlay precedes all garments, and absent categories are omitted. The sampled order is used for both reference construction and serialization. The outfit prompt lists garment descriptions in this same bottom-to-top wearing order, making the target order part of the model input rather than an image-only latent variable. For each garment, the full mask is its segmented mask before occlusion by other outfit layers. The visible mask excludes regions covered by layers above it. Here, "full" refers to the pre-composition garment asset, not a measurement of every unseen surface of a physical garment.

The task supports all six bottom-to-top permutations of shoes, bottoms, and inner top. Outerwear, scarf, and hat follow in that order. The person underlay precedes the listed garments in every case, and absent categories are omitted.

- Template A: shoes, bottoms, inner top, outerwear, scarf, hat.
- Template B: shoes, inner top, bottoms, outerwear, scarf, hat.
- Template C: bottoms, shoes, inner top, outerwear, scarf, hat.
- Template D: bottoms, inner top, shoes, outerwear, scarf, hat.
- Template E: inner top, shoes, bottoms, outerwear, scarf, hat.
- Template F: inner top, bottoms, shoes, outerwear, scarf, hat.

For each core record, we sample one of these templates. The selected template specifies the prompt-defined, stored wearing order at the garment-category level, not the physical depth of every visible body part.

Garment attributes are taken from the prompts used for generation. Color labels are determined from image pixels within the garment matte rather than copied from the requested color alone. The paired records and their attributes, garment categories, and compositing order are serialized in JSONL.

A.2 PAIRED REFERENCES

Starting from the reference person underlay, we composite the accepted garments one at a time in bottom-to-top order and save the image after each addition. Each removal example pairs one such image with the last garment added and the image saved immediately before that addition. Thus every step has a current-image reference, a garment target, and a lower-image target. The final removal target is the original person underlay.

Source-over composition of the accepted assets produces the dressed image and all intermediate reference images. Regions hidden by higher garments retain the RGB and alpha of the individually generated and segmented assets. Hidden-content evaluation therefore measures agreement with these constructed references.

A.3 QUALITY CHECKS

All images and assets share a 1024×1024 canvas. Alpha is stored as a soft channel in $[0, 1]$; RGB is straight (non-premultiplied). Before a record enters the core, we check:

- the sample-level manual checks described above;
- garment-contour alignment with the body underlay, rejecting assets that fail this check;
- sample/asset ID and garment-category count consistency with the dataset annotations;
- non-degenerate alpha support (≥ 0.001 of canvas) for every present category;
- source-over reconstruction of the dressed image from the stored stack with mean absolute RGB error at or below 10^{-4};
- consistency of stored prefix lower composites with the peel targets used for supervision.

A.4 SUBSET COMPOSITION

The complete core contains 22,431 person-outfit samples and 97,775 garment layers, equivalent to 44,862 task records across I2P and T2P and 4.36 garment layers per sample. The demographic and garment summaries below describe the 18,231-sample profiled subset.

Table 3 summarizes approximate sample-level demographic marginals for the 18,231 profiled core person-outfit samples. We assign a gender category during body-underlay preparation, as described above. The gender, age, race/ethnicity, and body-build labels are dataset curation categories.

Table 3: Approximate demographic composition of the profiled core subset. Shares are sample-level marginals, not exact counts.

Dimension	Categories	Approximate shares
Age group	Children / young / older people	20% / 40% / 40%
Gender label	Male / female	50% / 50%
Race/ethnicity label	Asian / White / Black	One third each
Body build	Slim / medium / larger build	40% / 20% / 40%

We join each sample in the profiled subset to its source annotation by dataset and sample ID, requiring exactly one match. We count each sample once across its I2P/T2P tasks. This join accounts for all 77,744 garment-layer instances in that subset, excluding the body underlay and auxiliary accessory composites. Table 4 reports category presence, which is multi-label rather than a partition. Outfits in the profiled subset contain one to six garment layers. Of its 18,231 samples, 14,439 (79.2%) have at least four layers and 14,305 (78.5%) contain both an inner top and outerwear, providing substantial layered-outfit coverage.

Source attributes include casual, streetwear, athletic, and formal styling; denim, leather, knit, cotton, wool, lace, and satin; and graphic, animal, floral, striped, camouflage, and plaid patterns. Garment-fit

Table 4: Garment coverage in the profiled core subset. Percentages use 18,231 samples as the denominator and need not sum to 100%.

Category	Layer instances	Sample coverage	Examples in source annotations
Inner top	15,830	86.8%	T-shirts, tank tops, polo shirts
Bottoms	15,657	85.9%	Shorts, jeans, skirts, cargo pants
Outerwear	15,388	84.4%	Leather/bomber jackets, coats, blazers
Shoes	15,551	85.3%	Sneakers, boots, loafers, oxfords
Hats	7,906	43.4%	Baseball caps, beanies, bucket hats
Scarves	7,412	40.7%	Neck scarves, bandanas, infinity scarves

descriptors include fitted, relaxed, and oversized silhouettes; these are clothing attributes, not body-build labels. This variation complements category and layering coverage with different textures, silhouettes, and coordinated accessories. Within the profiled subset, five-layer outfits account for 63.7% of samples, and black is the primary color label for 48.6% of garment instances. Style tags are descriptive labels drawn from the fixed prompt pool.

A.5 Challenge curation

The challenge subset is separately catalogued but included in the complete core totals. Following the core workflow in Appendix A.1, we generate each target garment separately on a shared person underlay and extract it before outfit composition. We use its generation prompt as the text query for Grounding DINO and use SAM 2 to obtain its mask (Liu et al., 2024; Ravi et al., 2025). We then check garment geometry and pose before composing the outfit. Each garment's alpha defines its full mask; removing regions covered by higher layers gives its visible region.

During core curation, we also perform structural and prompt checks, semantic screening, and near-duplicate checks. We exclude 379 people before finalizing the data. We keep body underlays and garment assets disjoint across the training, validation, and test splits, while allowing person and garment reuse within each split. Demographic shares describe the profiled subset, not the challenge subset.

B Model Components and Training

We train extraction, alpha decoding, RSAR, and restoration separately (Figure 7). Table 5 reports the three challenge-extension continuations; training rows and optimizer steps are distinct from evaluation sample counts. RSAR training is specified in Appendix B.2.

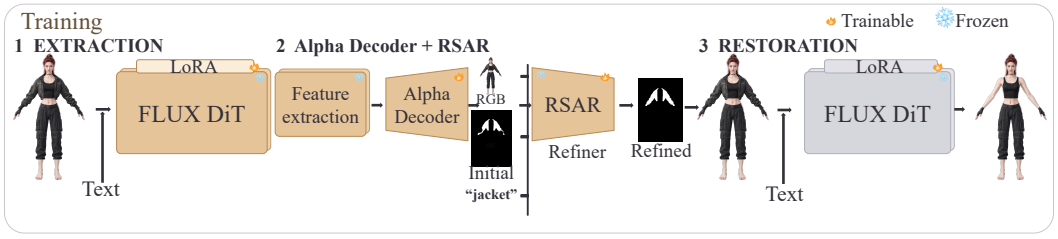

Figure 7: **Component training in LayeredFashion.** The schematic groups extraction, alpha decoding and refinement, and restoration. Components are optimized separately, including separate alpha-decoder and RSAR stages. Flames mark trainable components and snowflakes mark frozen components within the corresponding training stage. Initial and Refined denote the alpha mattes before and after RSAR. Training objectives and checkpoint dependencies are detailed in Appendix B.2.

Table 5: Continuation training on the challenge extension for the final frozen I2P model. These counts exclude earlier pretraining. The alpha decoder completed one effective epoch.

Component	Trainable part	Rank	Training rows	Training steps	Objective or scope
Extraction	DiT LoRA	256	20,724	1,296 opt. steps	full-latent flow matching
Alpha decoder	decoder only	–	20,724	5,181 opt. steps	one-epoch fixed-latent continuation
Restoration	DiT LoRA	256	17,260	8,630 opt. steps	full-canvas preservation continuation

B.1 Evaluation checkpoints

We fix the checkpoints in Table 6 before evaluation. Ablations retain these checkpoints and the shared inference policy except for the specified component or input changes.

Table 6: Checkpoints used for evaluation.

Experiment	Model checkpoints
Table 1 (200-person core-test)	Final extraction step 1,296 + alpha-decoder epoch-001 + frozen RSAR + restoration step 8,630
Table 2 (200-person core-test ablation)	Same final checkpoints and policy as Table 1, with the designated ablation changes
Final T2P execution test	Final extraction LoRA + alpha decoder + frozen RSAR (no *Restoration*)

B.2 Training settings and objectives

The three challenge-extension continuations in Table 5 use 1024×1024 images, four distributed workers, AdamW, and gradient-norm clipping at 1.0. We specify their objectives below and the stage-specific initialization, frozen components, and state sampling in Appendix B.3.

The extraction LoRA uses a nominal learning rate of 10^{-6}, weight decay 0.01, per-worker microbatch size one, and four accumulation steps, giving up to 16 records per optimizer update. One pass over 20,724 records yields 1,296 updates, including the final partial accumulation. The restoration LoRA uses a nominal learning rate of 10^{-5}, weight decay 0.01, and four records per update without accumulation; two passes over 17,260 records from 3,452 training outfits yield 8,630 updates. Both adapters have rank 256 and adapt the diffusion Transformer's attention, feed-forward, and modulation projections. The LoRA runtime uses `ConstantLR`: the post-update learning rate is one-third of the nominal rate at the first update and reaches the nominal rate later in the continuation. Both LoRAs use seed 20260812.

The alpha-decoder stage optimizes 49,546,628 decoder parameters in float32, while the baseline VAE and candidate encoder remain frozen. Its learning rate is fixed at 10^{-6} with zero weight decay, four records per update, and seed 20260812. One pass over 20,724 records yields 5,181 updates. Records are shuffled deterministically without replacement. At inference, we retain the specialized decoder's alpha channel as the initial matte for RSAR. The delivered garment pairs RSAR's refined alpha with RGB copied from the current person image, as in Eq. 6.

We supervise the extraction diffusion model with full-latent squared prediction error against the diffusion training target. To supervise both changing regions and those that should be preserved during restoration, we combine full-canvas error with separately weighted errors in changing and unchanged regions. Let e be the elementwise squared prediction error. A continuous change map $m \in [0, 1]$, broadcast over latent channels, assigns higher weights to changing regions. Its complement weights unchanged regions. The weighted mean error is $\langle e \rangle_m = \sum me / \max(1, \sum m)$, where the sums run over the latent elements. Before applying the diffusion scheduler's timestep weight, the restoration objective is

$$\mathcal{L}_{\mathrm{R}} = 0.25 \ \mathrm{mean}(e) + 0.375 \ \langle e \rangle_m + 0.375 \ \langle e \rangle_{1-m}. \tag{4}$$

We compute the map from the maximum absolute current-to-target RGB difference over channels, rescale it to $[0, 1]$, and apply box downsampling to 128×128. If necessary, the training current image is reconstructed from the target garment over its lower composite. No binary threshold is applied. This map weights the training loss only; it is not the binary inpainting mask used at inference and is excluded from inference conditions.

The alpha decoder is supervised using two latent sources: the frozen extraction diffusion model and the frozen VAE encoding of each target asset. The model-latent asset term supervises decoding from garment latents produced by the frozen extraction model. These latents are cached once and held fixed throughout the alpha-decoder stage. The reconstruction term applies the same asset loss to the frozen VAE's target-asset encodings. The blank term penalizes nontransparent output for blank-target latents. We combine these three terms as

$$\mathcal{L}_\alpha = \mathcal{L}_{\text{asset}}^{\text{model-latent}} + 0.5\,\mathcal{L}_{\text{asset}}^{\text{reconstruction}} + \mathcal{L}_{\text{blank}}. \tag{5}$$

To supervise opacity across the garment and its boundary, the asset loss combines alpha absolute error globally, on transparent pixels ($\alpha \le 1/255$), on foreground pixels ($\alpha > 0.05$), and in a radius-three morphological boundary band, with weights 2, 4, 2, and 8. Alpha-gradient error has weight 2.

We penalize alpha leakage at canvas borders separately. A 16-pixel target-transparent canvas border receives absolute-error and top-64 positive-alpha penalties with weights 8 and 12. For a one-pixel canvas border that the target does not touch at threshold $32/255$, the absolute-error and top-64 excess-above-$32/255$ penalties have weights 8 and 16. We also supervise the appearance obtained by compositing the predicted alpha: premultiplied-RGB error and RGB compositing error over black, gray, and white backgrounds each have weight 2. These training losses use frozen baseline RGB rather than candidate RGB; this is separate from the current-image RGB used to form the delivered layer at inference. An alpha-range penalty has weight 0.5.

For blank-target latents, the blank loss penalizes opacity throughout the canvas as well as at its borders and corners. It weights global absolute alpha, border absolute alpha, top-64 absolute alpha, top-64 positive alpha, positive corner alpha, and alpha-range error by 4, 12, 4, 16, 16, and 0.5, respectively.

We train RSAR to correct an initial matte rather than segment a garment from scratch. We use two trainable branches to separate full-image garment-ownership decisions from native-resolution opacity refinement. A frozen coarse predictor supplies their semantic context. It operates at 256^2 and supplies a coarse alpha and features. Its visual condition comes from the frozen `google/siglip2-base-patch16-224` encoder: the current RGB is resized to 224^2, and its final 196×768 patch tokens are rearranged and projected into the coarse predictor's bottleneck. A frozen SigLIP2 text encoder supplies a 768-dimensional pooled garment-text feature; the global description additionally contributes a 128-dimensional lexical feature, and the garment category a 24-dimensional embedding. The coarse predictor uses a 96-dimensional condition, which also conditions the trainable branches through FiLM scale and bias.

The full-view decision U-Net takes the complete observation at 256^2, with 23 input channels: RGB, initial and coarse alpha, their disagreement and certainty, and 16 projected feature channels. Four encoder levels have widths 16, 32, 64, and 128; three decoder levels have widths 64, 32, and 16. Residual blocks use two 3×3 convolutions, GroupNorm with eight groups, and SiLU. Stride-two 3×3 convolutions downsample; bilinear upsampling joins the corresponding skip features. A 1×1 head produces the soft COPY, ADD, and REMOVE logits. At 1024^2, the detail branch projects 25 input channels to width 16, applies three width-16 residual blocks with dilation rates 1, 2, and 4, and predicts one logit residual with a 1×1 head. The detail branch retains the native pixel grid. Its residual is bounded by $20\tanh(\cdot)$ before combination with the decision alpha in Eq. 2. We optimize only the soft decision U-Net and the native-resolution detail branch, for a total of 798,660 trainable parameters. The coarse semantic branch and SigLIP2 vision encoder remain in evaluation mode. We initialize the decision head with a soft copy bias and the detail head with zeros, so refinement begins close to the initial matte without imposing a hard identity rule. After training, we freeze the entire RSAR network; it returns only the refined alpha.

For this correction task, LayeredFashion's layered RGBA records provide paired supervision: an intermediate person composite, the stored target-garment alpha, and its category and text conditions. For each garment anchor, we create two initial-matte views $\widetilde{\alpha}_t$ that share one RGB image and the same reference target, category, and text. Training uses 16,431 people and 70,071 garment anchors, giving 140,142 views per epoch. The fixed validation split contains 3,000 people and 12,700 anchors.

The first view is equally likely to use the reference alpha or a lightly perturbed version with small shifts or boundary noise. We sample the second view uniformly from large missing regions, an empty matte, inclusion of a lower layer, replacement by another layer, and ordinary boundary errors. Missing regions can form large blocks or disconnected pieces, or remove garment parts such as sleeves, hems, collars, and scarf tails. False positives include over-expanded boundaries, background spill, and

alpha copied from a sibling garment; missing and false-positive errors can also occur together. Fine-boundary degradation includes erosion, blur, resampling, jagged edges, and low-resolution upsampling.

To expose RSAR to errors produced by *Extraction*, we also retain cached current-RGB and initial-alpha pairs. These examples include long coats and skirts, scarf loops, tails, folds, and fringe, as well as thin structures, openings, and mesh. They also cover low-contrast or same-color layers, patterned, reflective, and knitted materials, occlusion by hands, hair, or accessories, and confusion between adjacent garment categories. For each anchor with a matching cached pair, we select that pair with probability 0.7 in each epoch. When selected, the cached current RGB replaces the dataset composite for both views, while its paired cached alpha supplies only the second view's initial matte. The first view still uses reference or lightly perturbed alpha. Both views retain the dataset reference alpha as their supervision target; cached alpha never replaces this target. If no cache is matched or selected, both views use the dataset RGB and the initial-alpha sampling rules above. Across four epochs, we used cached extraction predictions for 7,605 of 280,284 second views (2.71%); the conditional probability above applies only to anchors with a matching cache, not to all second views. Reference mattes supervise training only; inference receives no target matte, quality score, restoration mask, or reference image.

To supervise preservation and correction within the same matte, we balance absolute error across preserved foreground, missing foreground, false-positive foreground, and background. These regions receive soft weights of 4, 4, 4, and 1 in the normalized loss, without hard thresholding. The final matte also receives binary cross-entropy, soft Dice, and spatial-gradient losses with coefficients 1, 1, and 2. We supervise the intermediate soft decision matte $\overline{\alpha}_t$ with the same loss, weighted by 0.5 and without the gradient term. Region normalization uses 10^{-8}, Dice uses 10^{-6}, and the logit input is clipped by 10^{-4}. We do not assign categorical targets to the COPY, ADD, and REMOVE probabilities, because different mixtures can represent the same continuous opacity. The action-class cross-entropy weight is therefore zero.

A single GPU processes four anchors and their two views per optimizer step (eight views), without gradient accumulation. AdamW uses weight decay 10^{-4}, $\beta = (0.9, 0.999)$, and $\epsilon = 10^{-8}$; gradients are clipped to norm 1. A 256-step pilot uses a cosine learning rate from 2×10^{-5} to 2×10^{-6}. Training then resumes the weights and optimizer state at 10^{-5}, without warmup or EMA. Forward computation uses BF16 autocast, while losses and optimizer states use FP32. Training and validation seeds are 20261010 and 20261022. The fixed validation set is evaluated at each epoch end; checkpoints are also saved every 500 steps. The candidate monitor combines overall validation loss, six-category macro loss, and outer/scarf loss, accepting a replacement only after a strict improvement. The reported model uses the completed epoch-4 checkpoint at step 70,072, selected before a plateau learning-rate reduction or automatic early stopping. Its fixed-validation mean alpha IoU is 0.9890. The checkpoint is `completed_epoch_04_step_0070072.pt`; its SHA-256 is `33a54cb5d246f4646a73856f75d354ada9b065c2be9918453569f8c5cc51863e`.

B.3 TRAINING PROTOCOL

We train extraction LoRA, alpha decoder, RSAR, and restoration LoRA in that order (Table 7), optimizing only the designated component group at each stage. For the extraction continuation, which predates RSAR, we generate rollout states with extraction diffusion → alpha decoder → restoration. For the final restoration continuation, we add frozen RSAR after alpha decoding, before constructing write supports and the next composite. We call the cascade snapshot used to generate these states the producer.

Both LoRAs use 1024×1024 images, AdamW, gradient clipping at 1.0, and rank 256, with flow matching for extraction and Eq. 4 for restoration. Training programs use the reference garment-category set and order from dataset annotations. Training and inference share the `target_first_role_pure_v1` prompt policy and `prompt_role_v5_extent` prior policy. The main I2P evaluations (Table 1) use the final checkpoints in Table 7. Ablations (Table 2) share these checkpoints and policies except for their specified interventions, including the independent *w/o RSAR* bypass.

For each peel transition in either LoRA stage, an independent Bernoulli draw chooses a teacher-forced reference state with probability 0.6 or an on-policy state with probability 0.4 (60/40). Reference

Table 7: Sequential training stages. Components outside the trainable group remain frozen.

Stage	Trainable group	Initialization, fixed inputs, and output
Extraction	Extraction LoRA	Initialize from the early-stage extraction LoRA. Frozen early-stage alpha-decoder and restoration checkpoints complete its rollout cascade. Output: extraction step 1,296.
Alpha decoder	Alpha decoder only	Initialize from the early-stage decoder. Frozen extraction step 1,296 supplies cached garment latents; frozen-VAE target-asset encodings and blank latents supply the other supervision. The cache stays fixed for one epoch (5,181 updates). Output: epoch-001.
RSAR	Full-view soft decision U-Net and native-resolution detail branch	Keep the coarse semantic branch, whole-image SigLIP2 vision encoder, and all pre-existing extraction and restoration components frozen. Train with paired initial mattes and dataset target alpha. Output: a frozen post-decoder refiner returning corrected alpha only.
Restoration	Restoration LoRA	Initialize from the early-stage restoration LoRA. The preceding extraction LoRA, alpha decoder, and frozen RSAR supply its rollout states. Output: restoration step 8,630.

transitions use the matched current image and, for restoration, its write support under the stage policy. For an on-policy transition t, we hold the current producer snapshot fixed and run exactly the preceding $t-1$ peels from I_0 to obtain I_{t-1}; at $t=1$, we use I_0 without a rollout.

Restoration rollouts retain $U_{t-1} = \bigcup_{j=1}^{t-1} M_j$, with $U_0 = \varnothing$. Frozen extraction diffusion, alpha decoding, and RSAR produce the current refined matte, which is thresholded to obtain M_t. The same fixed boundary treatment used at inference maps $U_{t-1} \cup M_t$ to S_t (Appendix C.2). Each record therefore contains the matched image, history, and support. Each preceding restoration uses its own support for both conditioning and hard composition (Eq. 3); the committed RGB feeds the next peel. The transition being trained never conditions on its own restoration output.

To keep on-policy states consistent with the current producer, we record the producing checkpoint hash with each online record and invalidate the preceding producer version after every LoRA optimizer update. After selecting a batch's sample IDs, peel indices, and reference/on-policy branches, we generate only the required missing or stale on-policy records before use, including for the first batch. We attach reference garment or lower-composite targets by sample ID and peel index regardless of the sampled branch.

The alpha decoder uses the fixed model-latent, reconstruction, and blank supervision in Table 7, without mixed state sampling. RSAR follows the two-view sampling rule and objective in Appendix B.2: both views share dataset or matched cached current RGB, category, text, and the dataset target alpha. When a cached extraction-error pair is selected, its RGB supplies both views and its alpha only the second; the first retains reference or lightly perturbed alpha. Neither stage refreshes online trajectories. The extraction latents and the offline cache of extraction-error pairs stay fixed throughout each stage, separate from the versioned rollout buffer. Target-side encodings may be reused while source assets and the VAE remain unchanged. Target or reference images are never inference inputs.

Algorithm 1: Component training with versioned conditioning states

Inputs: training data, reference targets, and the stage settings above.

1. For each stage in Table 7, load its initialization and reference targets; freeze all but its trainable group. For a LoRA stage, start an empty versioned rollout buffer.
2. For each scheduled update, select a batch. For a LoRA stage, select peel indices and independent reference/on-policy branches, refresh required missing or stale states with the fixed producer snapshot, and attach the reference targets. For RSAR, sample the paired views; for the alpha decoder, load fixed latent supervision.
3. Compute the stage objective and update only its trainable group. After each LoRA update, advance the producer version and invalidate older rollout records. Repeat steps 2–3 for the stage's scheduled updates.
4. Save the output checkpoint separately from its initialization, recording the initialization checkpoint, data, and configuration identifiers; continue to the next stage.

C Inference Controls

All image pipelines use a 1024^2 canvas. Each I2P extraction and restoration diffusion call uses 36 denoising steps. Sequential T2P uses 28 steps per base-FLUX proposal and 36 per extraction call, without *Restoration*. The alpha decoder and RSAR each add one non-diffusion forward pass per extraction.

C.1 Input conditioning

To guide *Extraction* toward the requested garment, we provide both an approximate spatial region and local appearance evidence from the current image. We encode these cues in two control images: a spatial-prior image J and a local-appearance image H. Neither uses a target mask or target layer.

We use the same prompt policy during training and inference, `target_first_role_pure_v1`. Its instructions describe the current garment appearance from the prompt, exclude other garment categories, clarify that the spatial prior is coarse, and request an RGBA output. This keeps the garment meaning and requested output format consistent across the two phases. *Restoration* instructions instead specify which garment to remove and describe the remaining layers.

We construct J using the final prior policy, `prompt_role_v5_extent`. It rasterizes category-specific normalized-coordinate primitives into a single-channel grayscale prior and adjusts garment length for a limited set of prompt phrases, such as explicitly long outerwear. The resulting opaque black control image contains these grayscale primitives, a red bounding box, and garment-category text. To construct H, we crop local content from the current recurrent image with fixed category-specific padding and blur. We soften the crop boundary, place the crop back at its original coordinates on a black full canvas, and draw a green box. The two images thus provide a coarse location cue and current-image appearance without reading garment references.

For *Restoration*, we normalize instruction text with a fixed semantic-v3 rule. Apart from trimming span-edge whitespace and a terminal period, it handles one semantic special case: when an `inner_top` contains *zhongyi* but lacks *cross-collar*, it appends a cross-collar inner-garment description; if the source span also contains narrow-sleeve cues, it adds the train-supported wrist-length, narrow-sleeve description. This process reads no target image, alpha, lower composite, or validation annotation.

C.2 Alpha and restoration support

The native-resolution RSAR detail branch is conditioned on current RGB, initial and decision mattes, the upsampled coarse semantic matte, disagreement and certainty maps, and frozen spatial features. Feature-wise linear modulation injects garment-category, text, and context conditions. Stride-one residual blocks with dilated convolutions (rates 1, 2, and 4) predict a logit residual, bounded by $20\tanh(\cdot)$ before addition to the decision matte's logit. RSAR has no thresholded acceptance gate or morphological edit at inference. Its output is only the refined matte, with no RGB or restoration-region prediction. The two stages operate on the full canvas, at 256^2 for the decision and 1024^2 for native-resolution detail. Detailed layer dimensions and losses are given in Appendix B.2.

Garment alpha represents continuous opacity and is passed unchanged from RSAR into the saved RGBA asset. As described in Section 4.2, we pair it with current-image RGB:

$$G_t = \big(I_{t-1}, \alpha_t\big). \tag{6}$$

We construct a separate binary write support to control changes to the recurrent RGB image. At each peel, we clip RSAR-refined alpha α_t to $[0, 1]$, round $255\alpha_t$ to 8-bit values, and select values at least 64 to obtain the current binary garment region M_t. The regions from all peels up to and including t form the historical union. Fixed boundary treatment of this union produces S_t; thresholding the current matte alone does not define the final support. Garment category and text influence RSAR, but no separate category-prior mask is added to the union. Both alpha and support are constructed at 1024^2, without intervening resizing or rethresholding. The binary support conditions *Restoration* as a mask, not as an extra channel concatenated to RGB. The identical mask governs hard composition in Eq. 3, copying every pixel outside it from the image entering that peel. This rule is shared by training and inference.

After the last peel, a segmentation module supplies person alpha for the final underlay. This underlay alpha is distinct from both garment opacity and restoration support.

D Evaluation Definitions

D.1 Asset evaluation

D.1.1 Twelve-metric comparison

To assess recovered garment assets under a common protocol, we compare all methods on a fixed core-test cohort (Table 1).

We draw 200 person-outfit records directly from the full core-test split of 3,000 records by stratified random sampling with seed 10086. The strata are garment-layer depths 1–6, with respective sample counts of 20, 20, 40, 40, 40, and 40 (ratio 1:1:2:2:2:2), comprising 780 garment layers in total. Every method is evaluated on this same fixed cohort, which is also used for the supporting T2P evaluation. Sampling does not depend on method outputs or their validity. The sampling seed is separate from the training and inference seeds. Metric-specific treatment of empty reference regions is specified below and does not alter cohort selection.

Predictions use the method-specific checkpoints and output adapters listed in Appendix G. We retain predictions in their original output order. For evaluation matching, we use Hungarian assignment with equal-weight soft-alpha MAE and premultiplied-RGB MAE as the cost, after applying the layer-activity rule (alpha support covering at least 0.001 of the canvas).

Unmatched targets count as misses for Recall. Unmatched predictions and extra layers are excluded from matched-layer metrics; for Leakage they contribute their full predicted alpha mass. After matching, we compute order-related metrics on the saved native category sequence. Recomposition uses the method's native saved order together with the method-predicted underlay.

Metric	Definition (this run)	Aggregation notes
Recall ↑	Target category retrieved if matched by Hungarian assignment and alpha support ≥ 0.001 of canvas.	Per-person fraction, equal average over the cohort.
α IoU ↑	Binary IoU after thresholding both predicted and reference alpha at 0.5.	Matched active layers → people → equal average.
Boundary F ↑	3×3 erosion boundary; two-pixel Chebyshev matching on original resolution.	Per-layer F then equal person average.
Visible MAE ↓	Straight-RGB MAE on reference garment pixels not covered by higher layers.	Pixel → layer → person; empty regions omitted.
Occluded MAE ↓	Straight-RGB MAE on reference garment pixels covered by higher layers; hidden RGB from the stored reference asset.	Same aggregation; empty-occlusion cases skipped.
Underlay MAE ↓	Full-canvas RGBA MAE between method-predicted underlay and reference underlay.	Equal average over the cohort.
Recomp. MAE ↓	Full-canvas RGBA MAE of native-order recomposition (predicted underlay + predicted layers) versus dressed reference.	Equal average over the cohort.
LPIPS ↓	AlexNet LPIPS (Zhang et al., 2018) with the official weights; as a preprocessing choice of this evaluation run, images are resized to 64×64 before encoding; transparent regions are composited over white.	One score per person, equal average.
CLIPScore ↑	CLIP ViT-B/32 (OpenAI weights); full outfit prompt truncated at 77 tokens; L2-normalized cosine, clipped at zero, no extra scaling.	One score per person, equal average.
VQAScore ↑	The public VQAScore implementation and its accompanying model checkpoint (see evaluation records); template "Does the image match the description?"; affirmative-token probability.	One score per person, equal average.
Leakage ↓	Soft-alpha mass of prediction outside matched reference support, normalized by total predicted alpha mass. Unmatched layers contribute their entire alpha mass.	Layer → person → equal average.
Exact order ↑	After Hungarian matching, a person scores 1 if and only if every target category is matched and the matched category sequence is identical to the bottom-to-top wearing order explicitly specified by the prompt; otherwise 0. A missing or unmatched target category (including zero matches) yields score 0 and remains in the denominator.	Accuracy $k/200$ over the same 200 people for every method. Each person contributes one 0/1 score; none is excluded from this denominator. Counts are given below.

Counts.

Method	Correct people k	Scored people N	Exact order $k/200$
LayeredFashion	200	200	1.0000
Qwen-Image-Layered	21	200	0.1050
Stable-Layers	40	200	0.2000
RevealLayer + auto boxes	132	200	0.6600
OmniPSD (train15k)	138	200	0.6900

Every method is scored on the same fixed 200-person cohort. A person scores 1 only when every target category is matched and the matched native category sequence agrees with the prompt-defined wearing order. Order errors, missing target categories, zero matches, and an absent or unreadable category sequence receive 0; they do not remove the person from the denominator. Thus Exact order is the mean of 200 per-person 0/1 indicators for every method.

D.1.2 GARMENT APPEARANCE REGIONS

We define each reference garment's support as pixels with alpha at least 0.5. Its visible region contains support pixels outside the union of the supports of higher layers; its occluded region contains the remaining support pixels, which lie within that union. Straight-RGB MAE is measured in these target-coordinate regions, averaging garment categories within each person and then valid people. Empty occluded regions are omitted, not assigned zero error.

D.2 ABLATION METRICS

For full-canvas Alpha MAE, predicted and corresponding reference garment alphas are mapped to $[0, 1]$ and compared at every pixel. We average the absolute error over extraction steps within each person, then give each of the 200 people equal weight. This step-paired score neither binarizes alpha nor uses Hungarian layer matching. It is distinct from the boundary-alpha MAE evaluated in the same ablation table and from the alpha IoU in the main I2P comparison.

For boundary-alpha MAE, the evaluation band is defined from the reference alpha alone at the native 1024×1024 resolution. Let $m(p) = \mathbf{1}[\alpha^\star(p) \geq 0.5]$ and let $\mathcal{N}_4(p)$ be the square neighborhood within four pixels of p along either image axis. The band contains pixels whose neighborhood includes both reference foreground and background:

$$\mathcal{B} = \{p : \min_{q \in \mathcal{N}_4(p)} m(q) = 0 \text{ and } \max_{q \in \mathcal{N}_4(p)} m(q) = 1\}. \tag{7}$$

For each corresponding predicted/reference garment pair, we compute $|\mathcal{B}|^{-1} \sum_{p \in \mathcal{B}} |\widehat{\alpha}(p) - \alpha^\star(p)|$ using continuous alpha values in $[0, 1]$. Layer scores are averaged within each person, followed by an equal-weight average over the 200 people. Predicted alpha does not determine the band. This evaluation band is distinct from the alpha-decoder training loss's radius-three boundary band and from Boundary F-score's matching tolerance.

We compute outside-support RGB MAE outside a fixed, reference-derived restoration scoring region. We use this region only for evaluation, independently of the step-local runtime write support S_t. During inference, we preserve pixels outside S_t exactly.

To measure error in the recurrent image, we compute trajectory drift as the mean RGBA MAE between the input to each extraction step and its corresponding reference intermediate composite.

D.3 CONTROLS AND ABLATIONS

The main I2P comparison and component ablations use the core test split. The supporting T2P evaluation uses the same 200 person-outfit test records as the main I2P comparison. Parser selection uses validation data. The full LayeredFashion I2P and T2P systems apply frozen RSAR after the alpha decoder. Component ablations retain the final checkpoints and runtime policy except for their specified bypasses or input substitutions.

Within the 200-person comparison, all methods use the same test conditions, shared evaluation settings, and hardware and software environment, with the same computing resources available to each method. RevealLayer receives no target boxes or masks. Our OmniPSD reproduction uses its official training code on 15,000 train-only people and a minimal Kontext I2P adapter. The adapter

receives the same Garment Description Parser's category program. These shared controls apply to evaluation; training and task adaptation remain method-specific. Appendix G details the output adapters and evaluation scope. We show GPT6-Astra High only in Figure 4, excluding it from Table 1.

We compare seven ablated variants against the full model on 200 core-test people from 25 asset groups at 1024^2 resolution. We score every variant on all four metrics, averaging extraction steps within each person and then weighting people equally. The *w/o Alpha Decoder* variant removes only the dedicated alpha decoder: the pipeline's base RGBA VAE decoder still supplies alpha, which passes unchanged through RGB postprocessing and becomes RSAR's initial matte. RSAR continues to receive the current person-image RGB. The *w/o RSAR* variant bypasses alpha refinement. The *w/o recurrent state update* variant generates and saves every restoration result but restarts each extraction from the original dressed image, not the preceding restoration or a reference lower composite. The *w/o support-constrained update* variant removes the outside-support preservation constraint.

To assess RSAR's input evidence and full-view decision, we test three variants. One replaces direct RGB with a neutral constant; another removes text, category, and visual-context evidence; the third substitutes coarse alpha for the full-view decision. Table 2 compares these variants; Appendix D defines their metrics.

We select among five parser candidates using the standard and challenge validation subsets (Appendix E). We then evaluate complete-program accuracy on the test set, retaining every test example in the denominator. Category, text-selection, order, and rejection errors all count as failures. On this test, the parser obtains $192/200$ program-exact predictions (96%).

For T2P, execution checks and qualitative examples assess whether the representation supports sequential garment composition. A controlled baseline comparison holds test inputs, original prompts, shared evaluation settings, environment resources, and random seeds fixed across methods within this task. Appendix G reports execution and PSD-export checks.

E Garment Description Parser

We compare five parser candidates on 1,884 validation examples from the standard and challenge subsets (Table 8). The selected Garment Description Parser uses a $d = 512$ pointer head with 2,106,886 trainable parameters: a shared $4096 \to 512$ projection, six start queries, six end queries, and a six-dimensional category-presence head. Within this head, we explicitly stop gradients through the frozen T5 features. Reported latency is the batch-1 end-to-end median for tokenization, frozen T5, the pointer head, exact source slicing, and constrained order decoding on an NVIDIA RTX PRO 6000 Blackwell Server Edition; it excludes the I2P/T2P image pipelines. The median and 95th-percentile latencies are 13.589 and 19.250 ms, respectively.

The shared linear projection maps frozen T5 token features to the pointer head's feature dimension. Tokens with a positive-width source character offset are valid candidates. Their mean projected feature is passed through category-specific linear classifiers and sigmoid functions to predict category presence. For each category, learned start and end queries score the projected tokens by dot product divided by the square root of the feature dimension. The objective averages binary cross-entropy over the six category-presence predictions and adds start/end cross-entropy only for present categories. At inference, the head selects the highest-scoring valid start–end pair, requiring the start to precede or coincide with the end. It then uses source character offsets for exact slicing; it never generates or rewrites a layer prompt.

Because the parser selects descriptions verbatim, we evaluate the complete program against the source text rather than accept paraphrases. Program exact requires the complete category set, every present category's source span after the specified normalization, the filtered bottom-to-top order, and no abstention. Normalization is limited to Unicode NFKC, whitespace collapse and trimming, and terminal ASCII-punctuation removal followed by trimming. Paraphrases are not exact matches.

We select the parser using validation program-exact accuracy. All three learned candidates reach their best primary-metric checkpoint at epoch 7. Test data remain separate from model selection.

Table 8: Garment Description Parser selection on validation prompts. Program exact requires correct garment categories, source-text spans, and bottom-to-top order, with no abstention. The domain macro-average gives equal weight to program-exact accuracy on the standard and challenge subsets. Scores measure text-program accuracy.

Candidate	Epoch	Standard (%)	Challenge (%)	Domain macro (%)	Pooled (%)	Median (ms)
T5 pointer $d = 512$	7	90.60	94.53	92.57	91.40	13.589
T5 pointer $d = 256$	7	90.20	94.01	92.11	90.98	13.574
T5 pointer $d = 128$	7	89.73	93.75	91.74	90.55	13.629
Rule spans + order router	–	56.47	92.97	74.72	63.91	1.094
First-mention rule	–	13.80	92.97	53.38	29.94	1.087

F Runtime Configuration

F.1 Shared configuration

The shared configuration is `configs/layeredfashion_d512_integrated_freeze_v2.json` (SHA-256 prefix: `9624075ca641...`). It fixes the $d = 512$ parser, image-generation and PSD-export settings, and inference policies. We use the checkpoints in Table 6, including frozen RSAR after alpha decoding. A single restoration LoRA handles both intermediate and terminal I2P transitions.

Both tasks use the same parser program: I2P follows reverse order and T2P forward order. We retain semantic-v3 restoration instructions and the fixed alias policy `v2_deterministic_prompt_aliases_v2_role_exclusive`. Its auxiliary semantic cues cannot change the parser's categories, source spans, or order. Appendix D.3 reports parser evaluation on all test examples, counting abstentions and incomplete outputs as failures.

F.2 Text-guided composition

T2P takes a supplied raw-person image as an independent input. We use bicubic resizing to form an immutable 1024^2 person underlay while retaining its source background. Starting from this underlay, each iteration adds one garment in the order supplied by the Garment Description Parser. We may add auxiliary semantic cues while keeping the parser's categories, source spans, and order fixed.

Base FLUX.1-Kontext (Labs et al., 2025) edits the current recomposed image to add the next garment. Its instruction uses that category's exact prompt span and asks to preserve the person and existing garments. This full-image output is a temporary proposal, not the next recurrent input.

The frozen *Extraction* module, including its alpha decoder and RSAR, isolates the requested garment from the proposal. It reuses the extraction weights and spatial controls, with the layer construction in Eq. 6: the proposal supplies appearance conditioning and the layer's RGB, the alpha decoder supplies an initial matte, and RSAR returns the refined opacity. T2P never calls *Restoration*.

We add only the newly extracted garment to the saved layer stack, then composite the stack over the fixed person underlay. This recomposed image becomes the input for proposing the next garment. Stored underlays and earlier assets remain unchanged.

G Baselines and Runtime Checks

We follow the evaluation controls in Appendix D.3, holding inputs, original prompts, preprocessing, and scoring fixed within each task. A shared seed schedule assigns the same seed to each evaluation record across methods; training and task adaptation remain method-specific. The I2P and supporting T2P comparisons share the stratified 200-person cohort in Appendix D.1.1. RevealLayer's automatic-box adapter remains fixed on this cohort, and OmniPSD uses Front/Back LoRAs trained on 15,000 train-only people.

G.1 EXECUTION AND PSD ROUND TRIPS

To verify generation and PSD serialization, we run the T2P pipeline on 200 core-test records with 780 requested garment layers. Each layer requires one FLUX dressing proposal and one frozen extraction diffusion call, totaling 1,560 calls. Generation reads no target files. All 780 returned assets have nondegenerate alpha, and all 200 PSD files pass the save/reload checks. We additionally test image generation, prediction-only packaging, PSD saving, unpacking, reloading, and a second save/reload for 200 I2P and 200 T2P people (Table 9). These checks assess execution and serialization, not garment-image quality.

Table 9: PSD round-trip evaluation. Both tasks use the frozen $d = 512$ parser. Target reads count reference-file access during generation; Composite MAE measures round-trip error over each task's full test cohort.

Task	Test people	Runtime inputs	Image program	Target reads	Composite MAE
I2P	200	dressed image + global prompt	*Extraction*; *Restoration*	0	0.00006908
T2P	200	raw image + global prompt	FLUX proposal; *Extraction*; no *Restoration*	0	0.00007104

G.2 GPT6-ASTRA HIGH INPUT PROMPTS FOR I2P COMPARISON

For the qualitative I2P comparison in Figure 4, we provide GPT6-Astra High with each dressed-person image and its corresponding prompt. We reproduce the two input prompts verbatim below.

Prompt 1.

> Task: image-to-layer decomposition for a clothed person.
>
> Input: the supplied dressed-person image and the global clothing prompt below.
>
> Return the editable RGBA garment layers and one base/underlay layer if the model supports them.
>
> Preserve identity, pose, silhouette, garments, and visible details. Do not add background, text, people, or invented clothing.
>
> Do not use target PSD, ground truth, alpha mattes, or any reference output.
>
> GLOBAL PROMPT:
>
> A tight base-body person wearing Chinese ancient dark brown leather flat boots, Chinese ancient brown narrow tapered cloth trousers cuffed at ankle, Chinese ancient plum purple narrow-sleeve cross-collar short ruqun top, Chinese ancient crimson dragon pattern long hanfu robe wide sleeves, Chinese ancient plain linen thin neck cloth, and Chinese ancient silk wrapped headband.

Prompt 2.

> Task: image-to-layer decomposition for a clothed person.
>
> Input: the supplied dressed-person image and the global clothing prompt below.
>
> Return the editable RGBA garment layers and one base/underlay layer if the model supports them.
>
> Preserve identity, pose, silhouette, garments, and visible details. Do not add background, text, people, or invented clothing.
>
> Do not use target PSD, ground truth, alpha mattes, or any reference output.
>
> GLOBAL PROMPT:
>
> A tight base-body person wearing black trousers (bottoms), a black zip-up athletic jacket (outer), and sneakers (shoes).

H Qualitative Examples and Applications

The complete I2P baseline comparison appears in Figure 4 in the main paper. This appendix presents the T2P comparison, additional I2P outputs, and applications of the recovered garment layers.

H.1 Garment-layer applications

Recovered layers support garment-level editing and 3D asset organization (Figure 1). An editor can select a garment by its layer address and recompose the stack after an operation, retaining the other garment objects. Semantic-color views identify clothing categories, while separate textured meshes support asset inspection.

H.2 T2P comparison

To compare sequential T2P composition with one-shot decomposition, we use matched input people and original prompts in Figure 8, extending Figure 5. The baselines decompose a shared Kontext edit, whereas our method extracts each proposed garment and recomposes the saved stack before the next step (Appendix F.2). Execution and PSD-export checks appear in Appendix G.

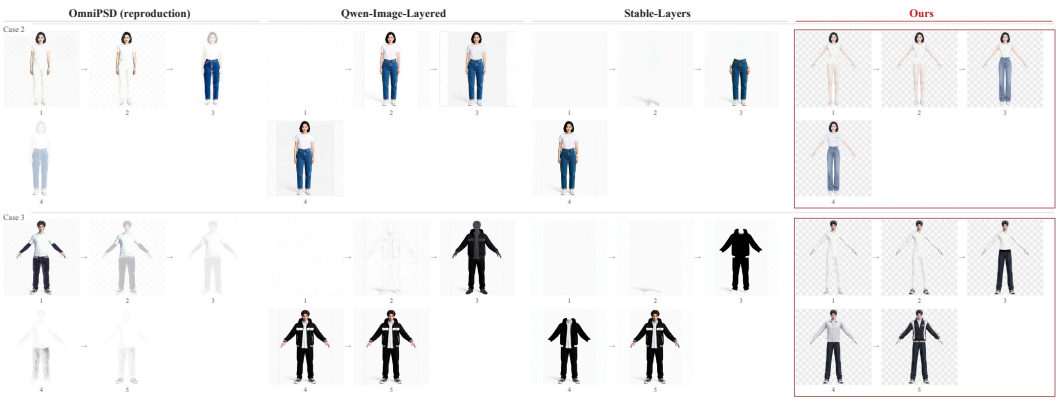

Figure 8: **Text-guided composition with matched inputs and original prompts.** Rows share the same input person and original prompt across methods; all columns contain actual model outputs. Numbered states add returned layers in each method's output order through a common alpha-over compositor. Each column retains the returned layer stack.

H.3 ADDITIONAL I2P OUTPUTS

Figures 9–13 show further outputs from the I2P runtime across different layer counts, garment combinations, body builds, and clothing styles. Each row presents the clothed input and returned garment layers on a common checkerboard canvas. Here, layer counts include all PSD layers, including the person underlay. Dataset garment counts and evaluation depths elsewhere exclude the underlay.



Figure 9: **Additional I2P outputs with five PSD layers, including the person underlay.** Rows cover varied body builds and contemporary outfits with patterned garments, long outerwear, and leather or puffer jackets.

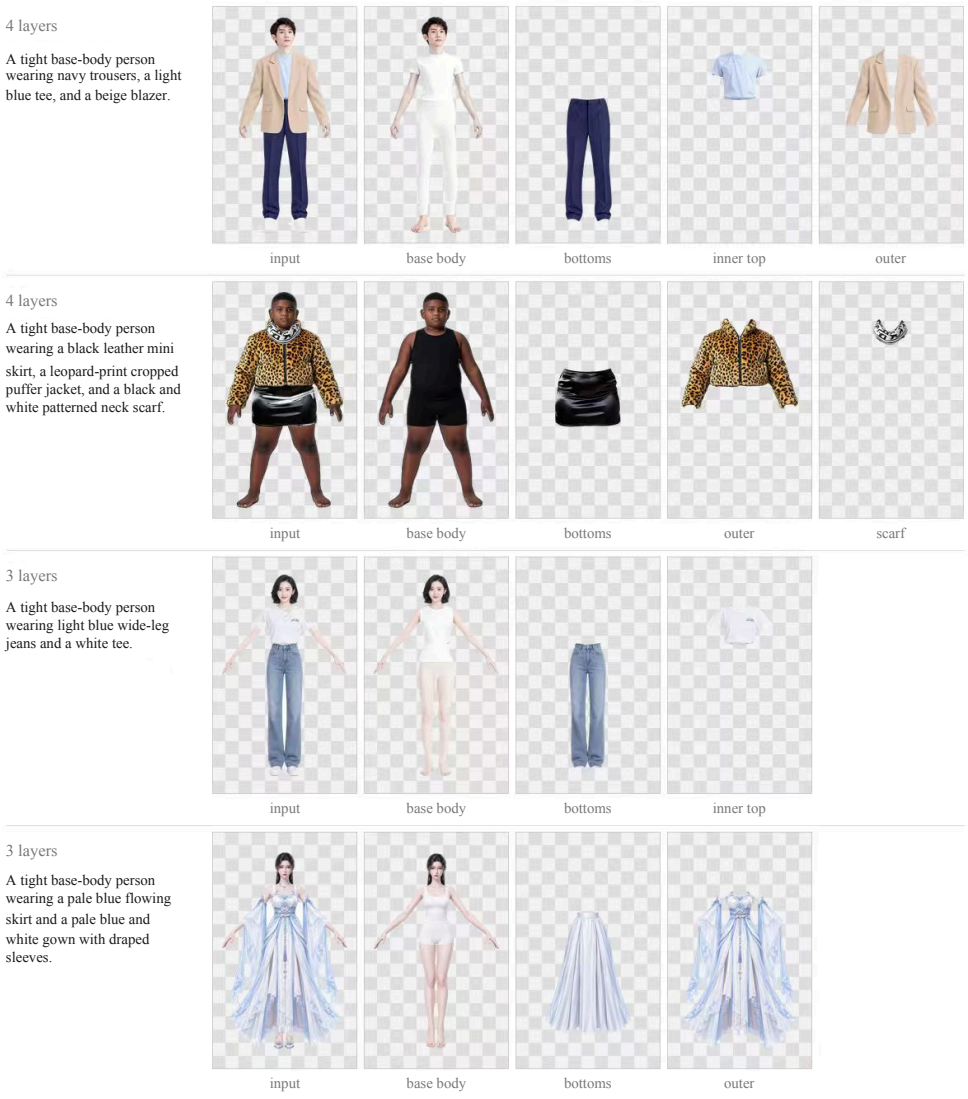

Figure 10: **Additional I2P outputs with three or four PSD layers, including the person underlay.** The examples include tailored clothing, patterned garments, casual outfits, and a flowing dress.

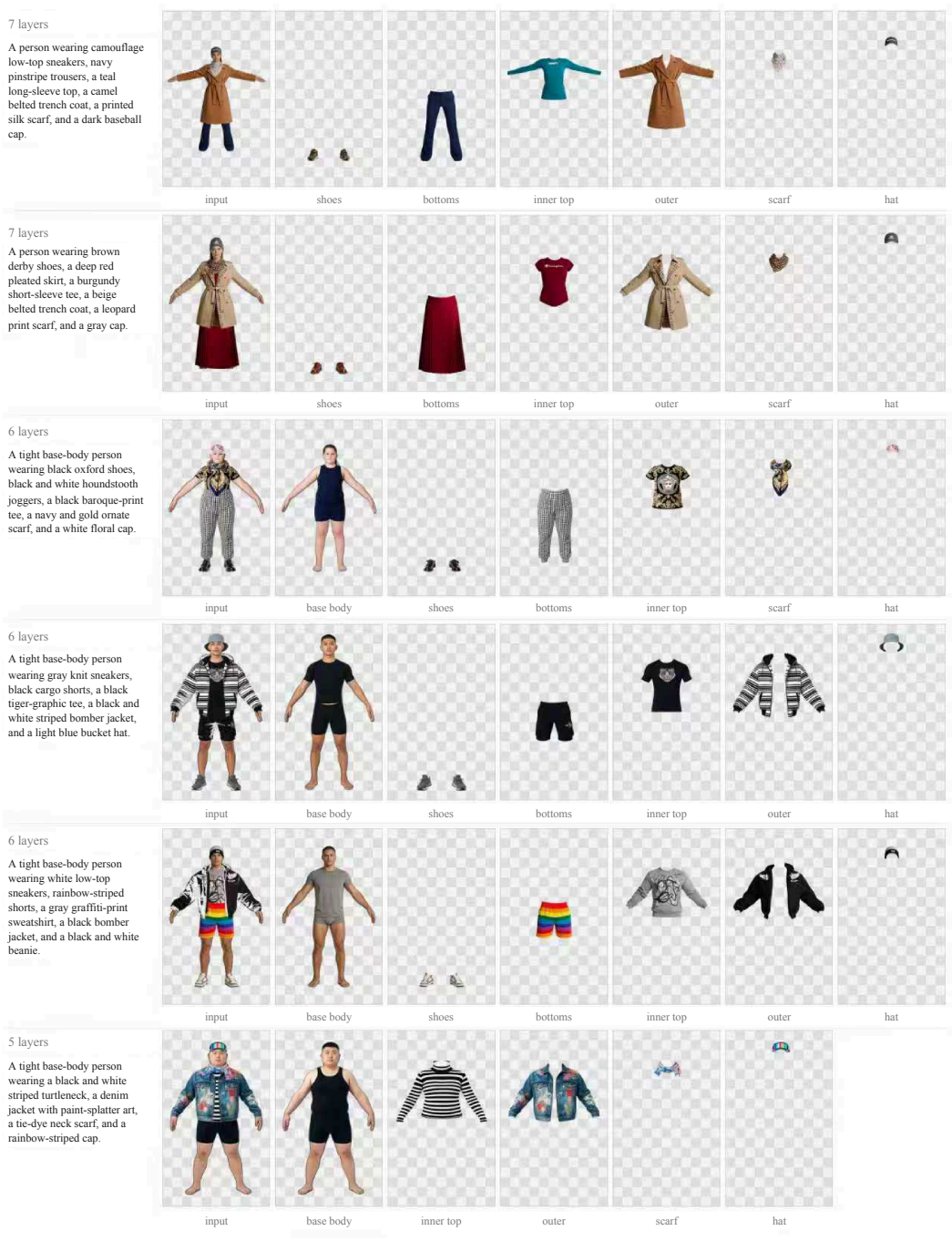

Figure 11: **Additional I2P outputs with five to seven PSD layers, including the person underlay.** Patterned, casual, and formal outfits combine tops, bottoms, outerwear, and small scarves and hats.

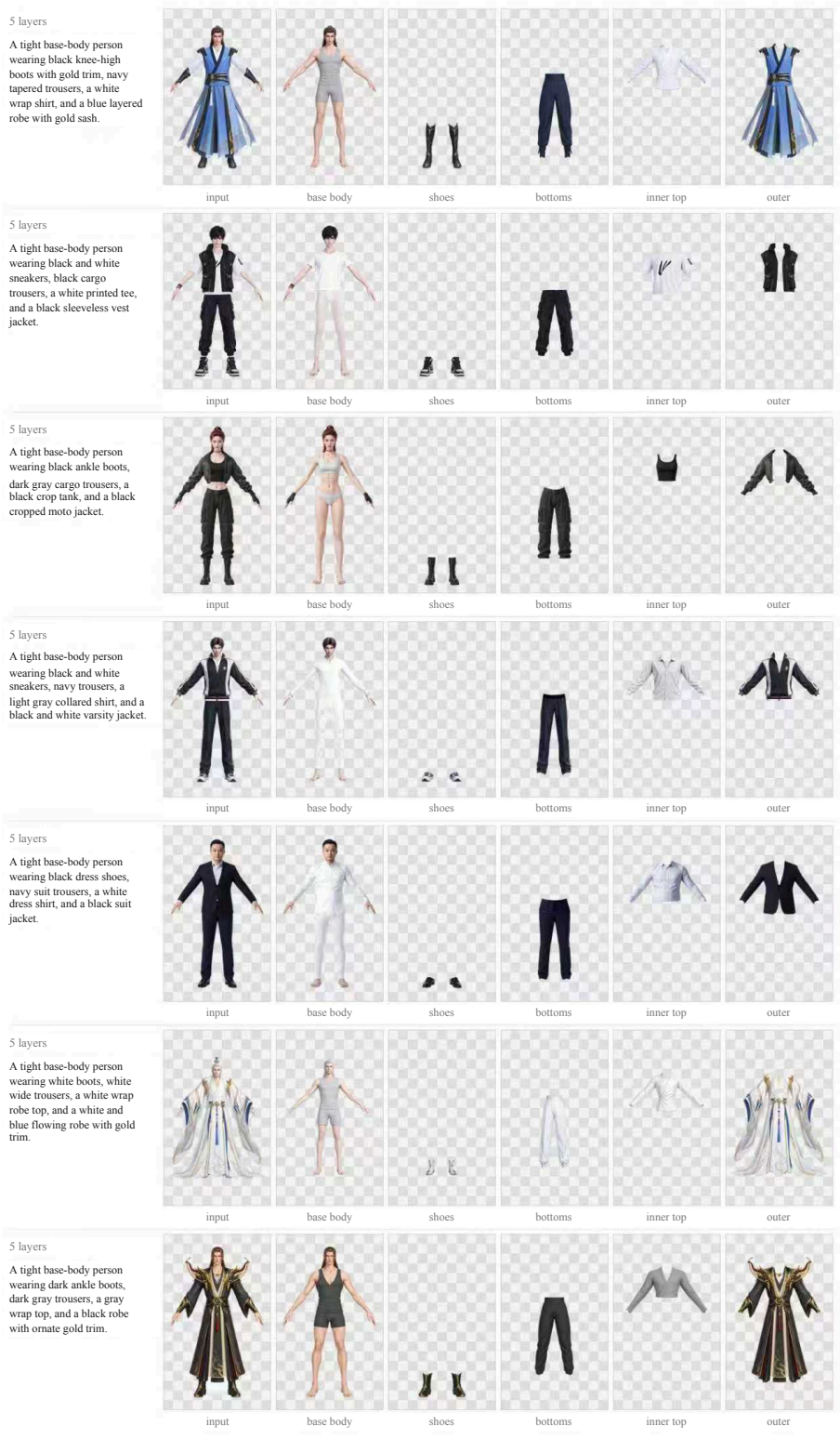

Figure 12: **Additional I2P outputs with five PSD layers, including the person underlay.** Rows span traditional robes, streetwear, varsity clothing, business attire, and layered formal garments.



Figure 13: **Additional I2P outputs with five PSD layers, including the person underlay.** The examples include tactical outfits and historical-fantasy garments with long, layered silhouettes.

H.4 3D ASSET EXAMPLES

Figures 14–16 pair each clothed input with its recovered person underlay, garment layers, and 3D visualization. Semantic colors are applied to the clothed-person mesh; separate garment renders retain their source textures.

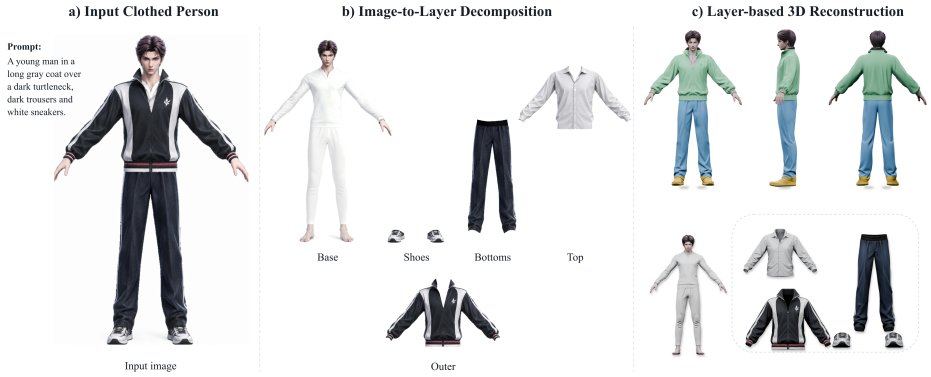

Figure 14: Shen Qingzhu: a black-and-white jacket, white shirt, dark trousers, and sneakers.

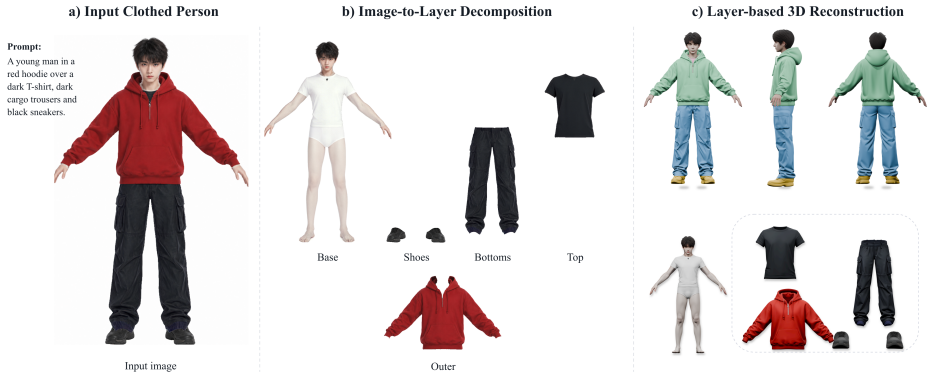

Figure 15: Wang Junkai: a red hoodie, dark T-shirt, cargo trousers, and black sneakers.

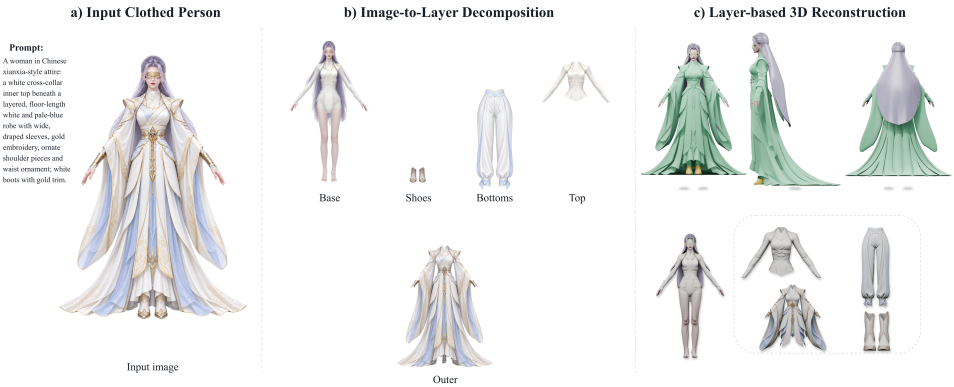

Figure 16: Yun Xiao: historical-fantasy attire with a cross-collar top, layered robe, and boots.