MV-Fashion: Towards Enabling Virtual Try-On and Size Estimation with Multi-View Paired Data

Supplementary Material

In this supplementary document we describe the capture setup and dataset annotation processes in more detail and provide additional statistics. Later, we detail the benchmark setups and provide additional quantitative and qualitative results.

8. Dataset Acquisition

8.1. Setup

As described in the main paper, we use 60 Raspberry Pi (RPi) global shutter cameras (1.6 MP), and 8 Orbbec Femto Bolt cameras. We use global shutter cameras rather than rolling shutter ones, avoiding motion artifacts and making the data more consistent. The Bolt cameras are Time-of-Flight (ToF) devices comparable to the Azure Kinect which record 4K color footage and lower resolution (640x576) depth information. All the cameras are connected to the same external trigger signal that is generated by a Raspberry Pi Pico module. It is a PWM signal that triggers the exposure. To create a single depth frame, the Bolt cameras take several infrared (IR) captures. In order to avoid interference between these, there is a sub-millisecond delay between each Bolt camera. This results in an interleaved pattern for the IR captures, where only one Bolt camera is capturing at a time. This also means that there is a higher chance of motion blur in the captured depth images, as the exposure times are fixed. Most consumer depth cameras operate in this manner, and this cannot be avoided. The overall synchronization is verified using a fast flashing LED strip. The LED strip uses individually addressable LEDs and is connected to an Arduino microcontroller. This setup can achieve sub-millisecond flashes of unique patterns that can be later identified in each frame and cross-referenced. Our setup achieves sub-2-millisecond synchronization across cameras. The lighting is controlled by 40 LED panels. These provide uniform and diffused light, with a high color rendering index (CRI) rate. Using panels help with distributing the power load, thus minimizing the electromagnetic interference. They are also flicker-free, not affecting the exposure.

All RPis are connected through ethernet cables and switches to a central computer, and all recordings are saved directly to this computer over the network. The Bolt cameras are connected to a mini computer each. These cameras are USB cameras, and connecting them to the same computer can cause instabilities, even when using dedicated USB expansion cards. That is why we opted to connect each to a mini computer (Asus NUC) first. They still record directly to the central unit over the ethernet.

We developed a custom application that allows us to control all cameras and record the sequences in a structured manner.

8.2. Collection Protocol

We have three main types of sequences: *Body* sequences, which record the subjects in minimal clothing to capture the body shapes accurately; *Template* sequences use fixed poses and aim to capture each piece of garment separately in these predefined poses; *Motion* sequences that record the dynamic clothing following random poses. We also record the individual catalogue/flat images of each garment with a top-down view camera. This top camera is mounted in the center of the capture setup, at the top. We use a temporary white background and lay the garments flat. Each flat garment image is recorded in two styles. The first variation captures the garment fully flattened to reveal its complete structure. The second presents a compact, less structured arrangement, designed to mimic real-world images taken by users.

As auxiliary data, each time before a subject re-enters the capturing area, we record an empty scene for the extrinsics calibration purposes to ensure consistency across sessions. We use this information to recalibrate the cameras to avoid any potential error due to the movement of the cameras. At the beginning of each day, we also performed manual checks and removed any dust that had accumulated during the previous day.

A typical recording session lasted 2 hours. Each subject could participate in up to three different sessions. We explained the protocol to the subjects, described their task and asked them to sign all necessary documents. During each recording, the subjects received live instructions shown on a monitor inside the capturing setup as well as a live preview of themselves. Specifically, we showed 5 random fashion images of different poses that we asked them to imitate to the best of their abilities (see Fig. 7). These images were collected from the internet, and they portrayed full-body images of fashion-oriented standing poses, similar to the ones found in many online clothing retail websites. Since we observed different trends in women's and men's fashion, we grouped the gathered images into two categories. One focused on women's style, while the other focused on men's style. Each participant was allowed to choose whichever category they preferred. The subjects

Figure 7. Representative samples from MV-Fashion illustrating the range of garment typologies (single vs. multi-layer) and poses captured in the dataset.

were instructed to bring a wide variety of clothing styles (sports, home, formal, etc), including those with multiple layers whenever possible. In these multilayer cases, we first recorded a few sequences with only one layer, followed by additional clothing layers in further sequences. For each of these, we recorded new template sequences, as well as whenever the style of the clothing item changed (eg. the coat is open or closed). At the end of the session, the subjects were requested to review the recordings and confirm with an additional form.

9. Dataset Curation and Annotation

9.1. Quality Assurance

To ensure the reliability of the captured data, we apply a data-cleaning and validation pipeline combining automated procedures with manual inspection. The process consists of two main components:

- Performing frame-count and timestamp alignment to enforce cross-camera and cross-modal consistency;
- Conducting a light quality check to filter out improperly exposed frames.

Camera Alignment. Each camera records at a nominal rate of 15 fps for about 20 seconds, producing approximately 300 frames per sequence. Minor variations in frame counts may occur due to startup latency or early termination. To obtain a clean and temporally consistent subset, we apply a light-weight alignment and cleaning procedure: we match RGB and depth frames using timestamp proximity, remove occasional extra or invalid frames, and standardize all camera streams to a unified timeline. This yields a synchronized multi-camera set with consistent views and modalities across the entire sequence.

Exposure Anomaly Detection. To avoid occasional exposure issues from affecting the final data quality, we perform a lightweight exposure check (see Fig. 8). When a recording contains improperly exposed segments or incomplete captures, we discard the affected take or segment.

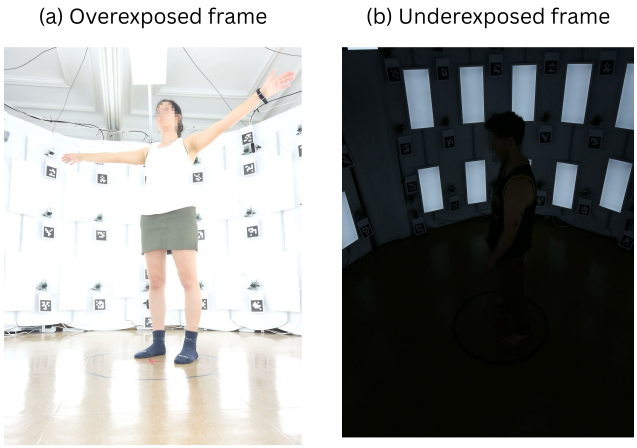

Figure 8. Examples of exposure anomalies: (a) An overexposed frame where strong illumination causes saturation and severe loss of texture. (b) An underexposed frame where insufficient lighting leads to a near-black appearance and loss of structural details.

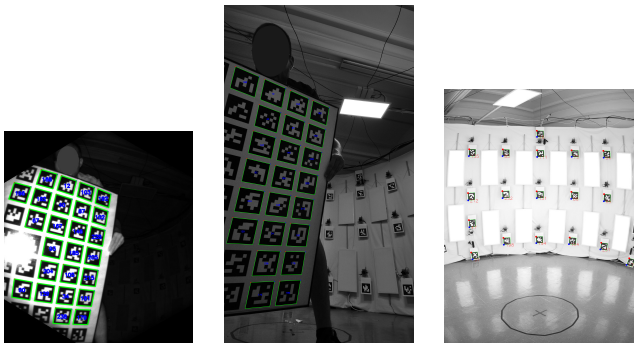

Figure 9. Examples of images and detected keypoints (in green rectangles) used in the intrinsics (a and b) and extrinsics (c) calibration.

9.2. Camera Calibration

We calibrate the intrinsics and eight distortion parameters with a matte printed AprilTag board using standard OpenCV functions. AprilTag has several advantages over checkerboard in precise multi-camera calibration, specifi-

cally due to the unique coding of feature points, the calibration is more robust when corners are partially visible. We repeat the intrinsics calibration three times: an initial pass to detect outlier keypoints, a refined second pass with outliers removed, consolidating the calibration. Finally, we correct misdetected keypoints in the undistorted images by intersecting the AprilTag edge lines at expected corner locations before performing the final calibration round. For the depth camera calibration, we adjust the IR image contrast to improve landmark detection. We then apply a stereo calibration between the depth and its corresponding RGB cameras. As a result, we achieve an average reprojection error of 0.4, 0.4 and 1.7 pixels for the RPi, Bolt IR and RGB cameras, respectively. For the extrinsics calibration among RGB cameras, including RPis and Bolts, we use AprilTag markers installed beside each camera using A5 flat boards. We apply open toolboxes [15, 60] for this task and achieve an average reprojection error of 0.3. See Fig. 9 for examples of AprilTag boards and detected keypoints used for calibration. Finally, for better alignment of the depth cameras and 3D fusion of 2.5D point clouds, we run ColorICP [61] between the overlapping depth cameras. See Fig. 10 for examples of the reconstructed point clouds.

Color Calibration. We perform standard polynomial color correction [24] between all cameras. We create a custom color calibration target as seen in Fig. 11. While we aim for a faithful reproduction of the colors, we cannot guarantee the exact color reproduction due to printing inaccuracies. Nonetheless, our main goal is to have consistent color profiles across each camera, which we can achieve with this method.

9.3. Segmentation

Segmentation Pipeline Details. As described in the main paper, we provide human foreground masks and layered garment masks for each frame. Representative examples of these segmentation outputs are shown in Fig. 12.

We adopt a unified prompt-driven two-stage segmentation pipeline: a task-appropriate initializer first produces a coarse guidance (e.g., via a text prompt or a bounding box), and then SAM2 [66] performs pixel-level segmentation. We instantiate this pipeline in three settings, with full implementation details as follows:

Flat Catalogue Segmentation. For static flat garment images with clean backgrounds and a single semantic category, we develop a batch-processing segmentation tool based on the open-source Lang-Segment-Anything framework [54]. Specifically, we apply GroundingDINO [48] to each RGB image using the fixed text prompt "clothes" (box threshold = 0.3, text threshold = 0.25) to detect garment regions. We retain the highest-confidence bounding box and use it as a prompt for SAM2.1-Hiera-Small to obtain a refined pixel-level garment mask. The SAM2 [66] probability map is then binarized using a threshold of 0.5 and, when necessary, resized to match the original image resolution. For each image, we generate two outputs: (i) an RGBA foreground image, where the alpha channel encodes the predicted garment mask, and (ii) a 0/255 binary mask PNG. All outputs follow the original directory structure of the dataset, enabling direct use in downstream modeling and analysis tasks.

Figure 10. Examples of the reconstructed point clouds visualized from different viewpoints. We apply **no post-processing** on the point clouds, including 3D and color smoothing. We preprocess the depth images to filter the boundary noise by masking out the pixels that have an absolute difference above 15 mm with their nearby pixels.

Foreground Segmentation. To obtain human foreground masks for all multi-view sequences, we adopt a video-level SAM2 [66] pipeline initialized from single-frame detection. For each camera sequence, we first apply

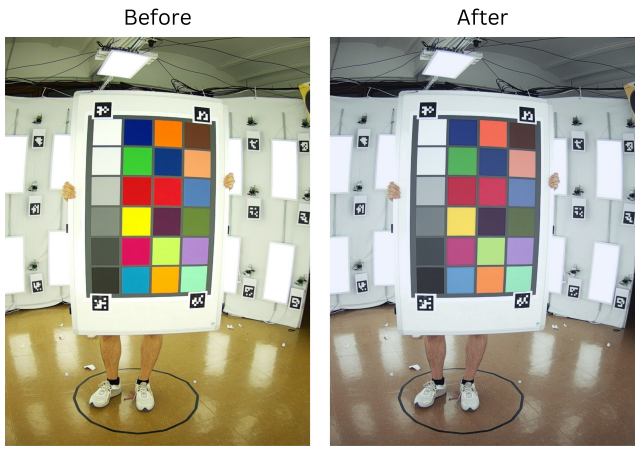

Figure 11. A sample image of the color calibration target before and after the color adjustments.

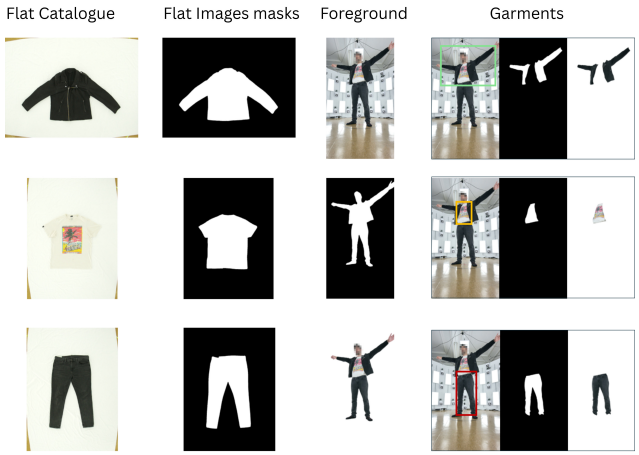

Figure 12. Representative segmentation outputs produced by our pipeline. From left to right: flat-catalogue garment images, corresponding flat garment masks, foreground masks, and layered on-body garment segmentation.

YOLOv8 [76] to the initial frame, retain person detections with confidence above 0.6, and keep the highest-confidence bounding box. This box is expanded by a factor of 1.05 to better cover fine structures such as hair, arms, and garment hems, and is used as the prompt for the SAM2.1-Hiera-Large video segmentation predictor. Since the first real frame may show boundary artifacts, we insert a duplicated version as a virtual frame 0 for SAM2 initialization. Segmentation runs from this virtual frame, but we keep results only from the original frame 1 onward to avoid these initialization artifacts. SAM2 then propagates the mask across the remaining frames. The output logits are binarized (threshold > 0) and resized to the original resolution when necessary. This pipeline is automatically applied to all sequences, and the resulting foreground masks follow the same directory organization as the raw captures to support downstream tasks.

Garment Segmentation. When obtaining layered garment annotations on dressed humans, we observe that existing methods (e.g., Sapiens [37] used in MVHumanNet++ [41]) often confuse inner and outer layers in multi-layer clothing scenarios. To address this issue, we leverage the semantic understanding capabilities of large vision–language models and design a two-stage fine-grained segmentation pipeline: (1) we prompt Qwen3-VL (Qwen3-VL-30B-A3B-Instruct-FP8) [65] with the image and the available metadata to produce initial garment bounding boxes, which serve as prompts; and (2) we apply a single SAM2 [66] model to propagate these masks across the sequence and obtain per-frame layered garment segmentation.

Unlike foreground segmentation, which only requires detecting a single human instance, garment segmentation must correctly identify multiple clothing layers (e.g., inner/middle/outer upper garments and the lower garment). For each camera sequence, we apply Qwen3-VL to the first frame to obtain a structured JSON output containing 2D bounding boxes for initialization. These normalized coordinates are then converted to pixel space to initialize the SAM2.1-Hiera-Large video predictor. For every detected garment, we assign a unique object ID and perform joint tracking within the same video predictor. Similar to our foreground pipeline, we duplicate the first frame as a virtual frame 0 to stabilize SAM2 initialization; effective segmentation begins at the original frame 1, avoiding occasional boundary fluctuations in the true first frame. During initialization, all garment bounding boxes are added at frame 0, after which SAM2 propagates the mask of each garment across the sequence. For each garment, we binarize the per-frame logits and resize them to the original resolution when necessary. Finally, we store masks individually for each garment to support downstream tasks such as size estimation, layered garment modeling, and virtual try-on.

Segmentation Verification. To ensure the reliability of large-scale automatic segmentation, we conduct lightweight human quality checks across all sequences. Flat Catalogue Segmentation and Foreground Segmentation operate in clean, single-category environments and therefore exhibit almost no noticeable errors in our random inspections. In contrast, Garment Segmentation involves multi-layer clothing and multiple object labels, making it more susceptible to layer-related ambiguities. For each sequence, we therefore generate a collage of the first frames and manually inspect potential failure cases. The primary segmentation-related issue we observe is that multiple garment layers may occasionally be merged into a single mask in complex layered-outfit scenarios. In such cases, we refine the results by adjusting prompts, replacing initializers, or manually correcting ambiguous masks, ensuring accurate and consistent final annotations.

9.4. Body Pose and Shape

Our body pose and shape annotations are based on SMPL-X [63], a widely used human parametric model. Given a sequence of n frames, we register body shape $\beta \in \mathbb{R}^{10}$, pose $\theta \in \mathbb{R}^{n\times 156}$ (body and hand pose, as well as global orientation), and translation $t \in \mathbb{R}^{n\times 3}$.

We infer these parameters with an automatic pipeline inspired by DNA-Rendering [11] and HuMMan [7]. (1) We detect 2D keypoints in the COCO-Wholebody format [33] for each camera view with a pretrained RTMW-x model [32]. Keypoints detected outside the segmentation mask are discarded. (2) We leverage the calibrated intrinsic and extrinsic parameters of the cameras to triangulate 2D keypoints, obtaining several 3D keypoints. These estimated 3D keypoints are averaged per joint and further improved through bundle adjustment, in which outlier 2D keypoints are filtered by a threshold. Additionally, a temporal median filter smooths their movement among frames. (3) Finally, we optimize the SMPL-X parameters via a modified version of SMPLify-X [63] in which we add the chamfer distance between SMPL-X vertices and the reconstructed point cloud. This extra loss provides enough guidance for the shape parameters fitting and is only performed for minimally clothed scans. Then, for all the clothed sequences of a subject, the shape remains fixed.

Quantitatively, our SMPLify-X pipeline achieves a mean per joint position error (MPJPE) of 23.49 mm and a chamfer distance of 55.13 mm. See Fig. 13 for qualitative examples of the fitted SMPL-X.

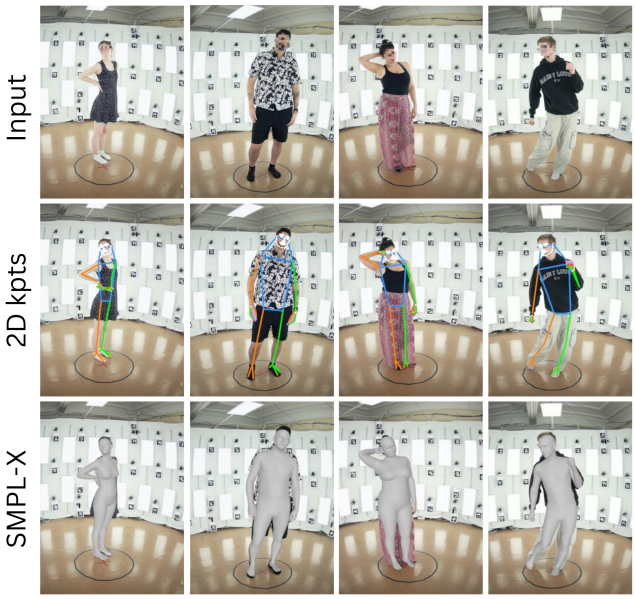

Figure 13. Qualitative results of our SMPL-X fitting pipeline. We display the input images, the 2D keypoints detected by RTMW-x in COCO-Wholebody format, and the final fitted SMPL-X model.

9.5. Garment Attributes

We define sizing chart annotations from a comprehensive set of 19 distinct measurement parts that span across the six garment groups (G1-G6) introduced in Sec. 3.3 of the main paper. In Fig. 14, we provide a visual definition of these measurements, illustrating the specific start and end points for each metric. Each garment group utilizes a specific subset of these measurement parts, which refer to the upper body (*Neck*, *Chest*, *Waist*, *Bottom*, *Sleeve*, *Bicep*, *Armhole*, *Shoulder*, *Body Height*, and *Sleeve Cuff*) or the lower body (*Bottom Waist*, *Bottom Hip*, *Bottom Bottom*, *Thigh*, *Leg Cuff*, *Front Crotch*, *Back Crotch*, *Leg Length*, and *Full Length*). Importantly, not all measurements within a group are applicable to every garment instance (e.g., a sleeveless T-shirt lacks *Sleeve*, *Bicep*, and *Sleeve Cuff*). In such cases, we assign a special invalidity mark to these entries in the annotations. This mechanism is critical for the size estimation baseline described in Sec. 4.2, as it allows the model to predict only the measurements present on the garment.

Regarding annotations beyond numerical measurements, we generate the textual description for each garment item in MV-Fashion using the multi-modal large language model, Qwen3-VL (Qwen3-VL-30B-A3B-Instruct-FP8) [65]. Fig. 15 illustrates the complete input for Qwen3-VL, including the textual prompt and both views of the garment, and a representative output.

We further enrich the dataset with four draping styles, defined across two layers: layer 1 (L1) for base garments, layer 2 (L2) for outerwear. The specific categories and their numerical encodings are detailed in Tab. 6. Finally, we annotate the draping style for each sequence as a composite style code that concatenates the numerical encodings of each attribute from L1 and L2, using the underscore (_) as a separator: (L1 Torso) _ (L1 Sleeve) _ (L1 Tucking) _ (L2 Torso) _ (L2 Sleeve) _ (L2 Hood) _ (L2 Tucking).

For more representative examples, we provide clear demonstrations of the garment's draping style and fitting appearance in Fig. 16. Additionally, full annotations set examples, including cloth category, material, elasticity, and textual description are given in Fig. 17.

9.6. Dataset Statistics

We provide additional statistics about the dataset in Fig. 18 and Fig. 19. The subjects present a balanced distribution across height, gender and weight, while it leans towards the younger age range below 40 for the majority of the dataset. The BMI distribution also confirms that while the extreme ranges have only a few representative samples, the overall distribution is close to a Gaussian.

We also expand on the distribution of the 14 garment categories in Tab. 7. It repeats the total, multi-layered, and styled distributions presented in the main paper. In the *total* distribution, shirt-blouse represents the majority, followed

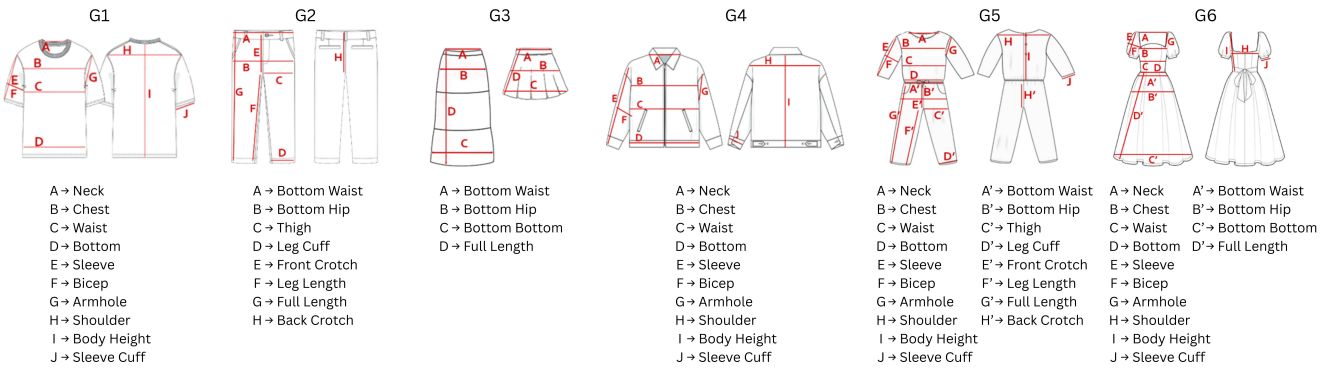

Figure 14. Visual definition of the sizing charts for each garment group (G1-G6). We annotate the measuring tape locations for all 19 measurement parts used in our dataset. Red lines indicate the distances measured.

Attribute	Layer	Categories	Encoding
Torso Closure Style	L1 & L2	n/a / fully closed / partially closed / fully open	0 / 1 / 2 / 3
Long Sleeve Style	L1 & L2	n/a / rolled up / rolled down	0 / 1 / 2
Tucking Style	L1 & L2	n/a / tucked / outside	0 / 1 / 2
Hood Style	L2 only	n/a / hood up / hood down	0 / 1 / 2

Table 6. Draping Style Attribute Encodings. n/a stands for not applicable.

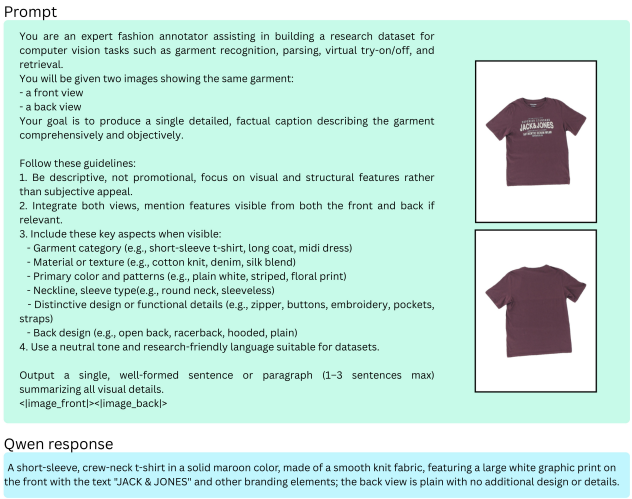

Figure 15. Qwen3-VL [65] garment description protocol. The input includes a detailed prompt and both views of the garment item to guide Qwen to generate descriptive and clear caption.

by pants and shorts, which is expected for these common outfit types. Dresses and skirts also represent more than 8% of the data. The rest are mostly outerwear, like jacket, sweater, sweatshirt or coat, which can appear in multi-layered clothing. The next *layer* row depicts the percentage of each category that was part of a multi-layered outfit. The outerwear categories are close to or exactly 100% since these are usually worn on top of another garment. Similarly, shirt-blouse has a high percentage as it is usually worn under an outer layer. Any lower body garment, like pants or skirts, is automatically 0% since no other garment is worn on top of those in this dataset. Lastly, the *style* row shows the percentage of each category that had multiple style variations. Again, outerwear-type garments have a high percentage, as these can often be worn open or closed. Shirts and sweaters are also present, since another common style variation is the rolling up of the sleeves or the tucking in of the shirt. Similarly to the layers, we do not define any styles for the lower body garments, as such these have 0%.

We also present some new statistics, namely, the garment fit (slim, regular and loose) and the recorded elasticity values' distribution. As expected, the regular fit is the most common in the majority of the categories, followed by the loose clothing in the range of 15-30% in most cases. The slim fit is limited to some specific categories, including shirt-blouse, dress, sweatshirt, and cardigan, showing below 15% representation. The exception is the tights-stockings with a 100% slim fit. Similarly, the recorded elasticity values show that the dataset has a balanced coverage of different materials, as the values from 1 to 3 are well represented in most categories. Values 4 and 5 appear more often in specific categories, including dress, sweater, cardigan, and tights-stockings, which is expected in real life.

10. Baselines Implementation Details

10.1. Virtual Try-On

As detailed in Sec. 4.1 of the main paper, we leverage the state-of-the-art architectures IDM-VTON [13] and InsertAnything [72] for Single-View baseline with their default training configurations and adapt IDM-VTON for the Semantic Controllability and Multi-View Geometric Analysis baselines. We provide the corresponding modifications in

	shirt-blouse	pants	shorts	dress	jacket	sweater	sweatshirt	skirt	coat	jumpsuit	vest	cardigan	blazer	tights-stockings	**Total**
Total	38.3	21.2	11.7	5.2	5.1	3.8	3.6	3.4	3.0	1.3	1.1	0.9	0.7	0.1	100.0
Layer	51.6	0.0	0.0	33.3	97.4	82.1	81.5	0.0	100.0	70.0	100.0	100.0	100.0	0.0	39.02
Style	7.7	0.0	0.0	0.0	86.8	39.2	55.5	0.0	59.1	10.0	62.5	28.6	60.0	0.0	14.13
Slim	12.2	5.0	0.0	15.4	7.9	3.6	14.3	7.7	0.0	10.0	0.0	14.3	0.0	100.0	8.1
Regular	68.3	55.3	79.1	59.0	71.1	64.3	71.4	65.4	68.2	70.0	100.0	42.9	40.0	0.0	64.1
Loose	19.5	39.6	20.9	25.6	21.1	32.1	14.3	26.9	31.8	20.0	0.0	42.9	60.0	0.0	24.7
Elasticity 1	18.9	60.7	59.3	25.0	69.2	0.0	13.8	48.1	76.2	11.1	62.5	28.6	83.3	0.0	37.9
Elasticity 2	30.2	27.6	27.5	30.0	23.1	32.1	51.7	44.4	23.8	44.4	0.0	0.0	0.0	0.0	29.2
Elasticity 3	40.9	11.7	9.9	25.0	7.7	46.4	27.6	7.4	0.0	33.3	37.5	42.9	16.7	0.0	25.1
Elasticity 4	9.6	0.0	2.2	15.0	0.0	21.4	6.9	0.0	0.0	11.1	0.0	28.6	0.0	100.0	6.2
Elasticity 5	0.3	0.0	1.1	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5

Table 7. This table shows the distribution of different clothing categories across the dataset (Total). It also shows how often each clothing type shows up in multi-layered outfits (Layer) or in outfits with style variations (Style). Additionally, we show the distribution of our fitting style labels (slim, regular or loose) and the distribution of our elasticity annotation values for each garment category. The last *Total* column shows the percentage of each annotation in the full dataset. All values are percentages. The percentages for layers are calculated outfit-wise, while the rest of the values are calculated garment-wise.

Fig. 20 and the following sections.

Semantic Controllability. This experiment extends IDM-VTON to utilize the novel styling (draping and fitting) annotation for fine-grained control over the synthesized garment appearance. We define the augmented text prompt T' as the union of the original text prompt T and a structured styling template derived from the categorical styling annotation $\mathbf{A}$ (defined in Sec. 3.3 of the main paper and Sec. 9.5):

$$T' = T \cup \text{Template}(\mathbf{A}) \tag{1}$$

where T describes the base garment attributes and $\text{Template}(\mathbf{A})$ converts the categorical styling annotation $\mathbf{A}$ into a precise textual suffix (e.g., "outerwear is fully closed"). This union operation represents the combination of the input strings before encoding, which then conditions the IDM-VTON synthesis.

Multi-View Geometric Analysis. For the *View-Adaptive Try-On*, we adapt the IDM-VTON architecture to leverage the paired frontal ($I_{C,F}$) and rear ($I_{C,R}$) garment views. We introduce a minimal modification involving feature fusion before conditioning the core model. We utilize one shared IP-Adapter [84] and one shared GarmentNet for processing both views. The resulting high-level feature tokens from the IP-Adapter ($\mathbf{Z}_{IP,F}, \mathbf{Z}_{IP,R}$) are concatenated to form a single combined feature vector $\mathbf{Z}_{IP} = \text{Concat}(\mathbf{Z}_{IP,F}, \mathbf{Z}_{IP,R})$. Similarly, low-level features $\mathbf{Z}_{GN}$ are fused by concatenating GarmentNet attention layers' features. Additionally, we extend places for IP-Adapter's tokens in the cross-attention operation of TryonNet to handle doubled token length. Finally, IDM-VTON is conditioned by the combined IP-Adapter feature vector $\mathbf{Z}_{IP}$, the combined GarmentNet features $\mathbf{Z}_{GN}$ and the prompt T. This stands in contrast to the *Cross-View Geometric Test*, where we quantify the challenge of cross-perspective mismatch by conditioning the model only with the frontal garment view $I_{C,F}$.

10.2. Size Estimation

As explained in Sec. 4.2, available approaches for garment sizing [40, 62] predict the measurements in controlled scenarios where the garments appear flat and unposed in the images. They rely on the 2D keypoints detection and perform the measurements between keypoints directly in the pixel space, which are not generalizable on our posed data. A natural extension of these works is to reconstruct the garment 3D mesh [94], lift the detected 2D keypoints into 3D and compute the sizes on the 3D surface. However, this approach requires complex intermediate processing of 3D data, and we consider 3D surface reconstruction out of the scope of this paper. Instead, SPnet [45] reconstructs the 3D surface by predicting the sewing pattern parameters from the normal image of the clothing, which is estimated beforehand. This normal image is given in a canonical T-pose to handle challenges like occlusions and wrinkles. Finally, the 3D mesh is generated from the sewing pattern by computer graphics techniques. We find the two-stage pipeline of SPnet is easily adaptable to our data without needing complex 3D processing, i.e., by replacing the sewing patterns data with our sizing charts. Specifically, our baseline consists of a garment normal predictor (Ψ) and a garment size regressor (Φ). Since an outfit in a given image can have upper and lower body garments with multiple layers, we only consider the outermost visible layer, assuming the garment category (and hence its group) is known beforehand. Therefore, we exclusively focus on the garment size estimation. Also, we deliberately avoid geometric data augmentation; by relying solely on our data, we assess if it provides sufficient information to estimate garment size on its own, demonstrating that the natural variation provided by our multi-subject, multi-view capture system is robust enough to generalize

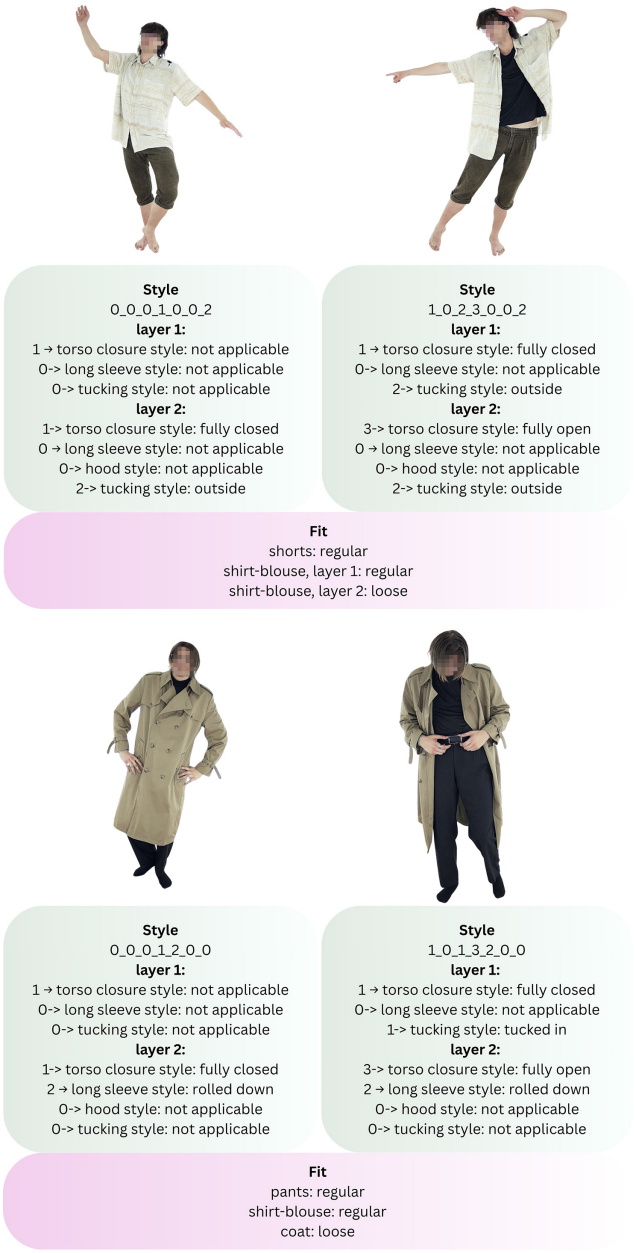

Figure 16. Some examples of the styling labels MV-Fashion provides. It is a unified string of numbers that encodes several styles we defined. We also provide the garment fit for each piece of clothing.

without synthetic modifications. Furthermore, augmentations like random scaling or cropping would corrupt the relation between G^t and our measurement annotations unless accompanied by highly specific, non-trivial adjustments.

Normal Predictor. For $\Psi(G^s, P^s, P^t)$, the goal is to transform the garment (G^s) from a posed state (P^s) into a canonical, unposed state (P^t) to facilitate the garment measurement. This process is illustrated in Fig. 21. All pose maps involved in the baseline (P^s and P^t) are semantic

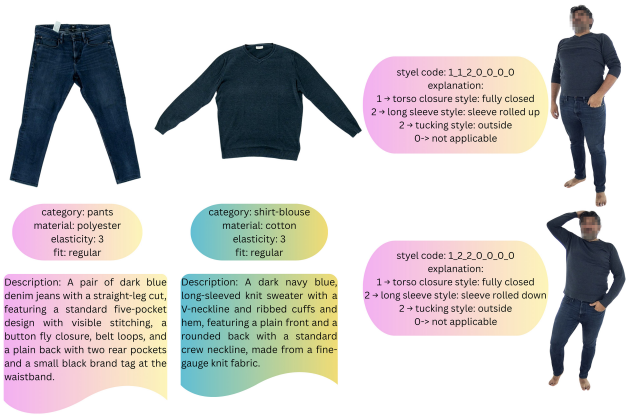

Figure 17. Additional example of text annotations MV-Fashion provides for each outfit. This includes a cloth category, material and elasticity details, our custom styling encoding and the detailed description.

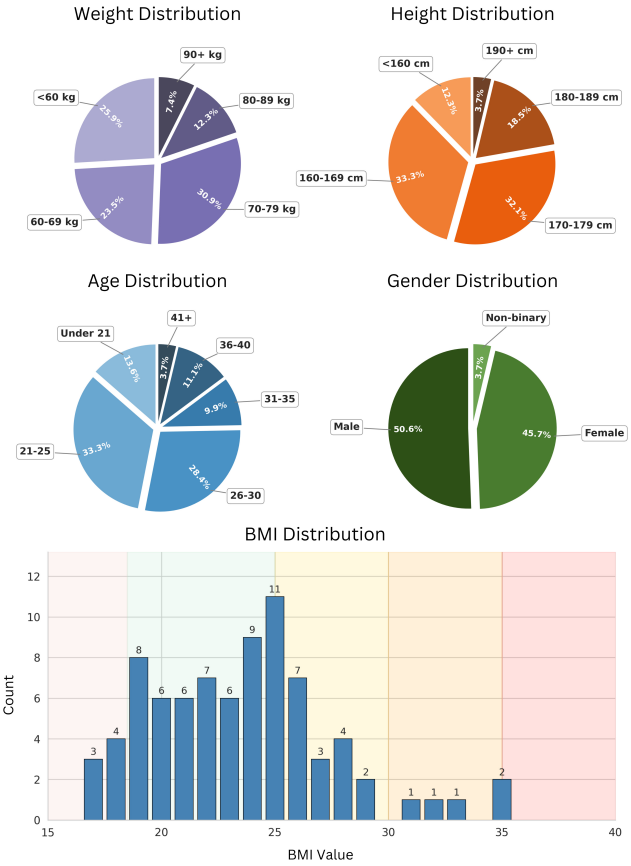

Figure 18. We collected additional information from each subject, like height, weight, age and gender, since these could be important for downstream tasks. They also help us evaluate and ensure a balanced distribution of our dataset.

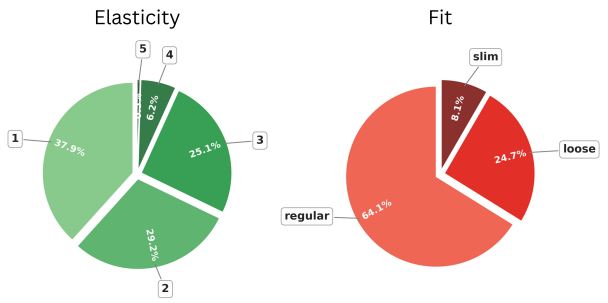

Figure 19. Distribution of different elasticity values and fit styles across the dataset. Highly elastic clothing items are rare, but the more common ones are well represented. The fit styles are also well distributed and MV-Fashion contains a high percentage of challenging, loose clothing.

body segmentation maps rather than simple skeleton renderings. Specifically, we apply an `argmax` operation to the SMPL-X blend weights to assign each pixel to a specific body part (e.g., the waist, the hip). By doing so, we provide the model with explicit spatial attention cues, directly linking visual body regions to the different parts of the garment. The data preparation process for G^s, P^s and P^t is detailed in Sec. 4.2 of the main paper.

Size Regressor. The second network estimates garment measurements from the canonical normals (G^t). We evaluate three architectures for Φ. (1) *Per-Group*: This is the direct implementation of the SPnet sewing pattern regressor repurposed for the garment measurement. Following their protocol, we train a separate network for each garment group (G1–G6, see Sec. 9.5), limiting the training data for each model to only the samples available for that specific group. (2) *Multi-Task*: To leverage the overlapping data among all the groups, we use our annotation protocol to perform multi-task learning. We train a single unified network by grouping common measurements (e.g., *Chest*) that appear across groups (see Fig. 14 for overlaps). This significantly increases the effective number of training samples per measurement, improving the model's ability to learn robust features for shared body regions. (3) *Multi-Task + SwinV2*: We replace the original SegNet [3] encoder with a pretrained SwinV2 [50], and additionally condition Φ on the target pose P^t to guide the network's spatial information towards the corresponding entries in the sizing chart.

10.3. Novel View Synthesis

As detailed in Sec. 4.3 of the main paper, to validate the applicability of MV-Fashion for NVS we run several benchmarks and ablations. We choose Nerfstudio as the framework to run our novel view synthesis benchmarks, as it aggregates multiple popular and state-of-the-art methods, making them easy to test. It also supports a common data preprocessing pipeline, which has added features that the original implementations might lack. With the three selected methods (`instant-ngp`, `nerfacto`, `splatfacto`), we can cover both major categories, those being Nerf and 3D Gaussian Splatting. Most recent downstream tasks rely on one of these methods; thus, validating them on our dataset is important.

Additionally, we briefly explore CEM-4DGS [35] a 4D Gaussian Splatting model to validate that our dataset can support dynamic novel view synthesis as well.

11. Experiments and Results

11.1. Virtual Try-On

Training Setup. We fine-tuned IDM-VTON and InsertAnything using their original training protocols. For all experiments, we uniformly resize input images to 1024×768. For the IDM-VTON protocol (applied to all IDM-VTON baselines), we utilize a batch size of 12 for 130 epochs, employing the AdamW optimizer [51] with a learning rate of 2×10^{-5}. For the InsertAnything protocol, we utilize the Prodigy optimizer [56] as the default with safeguard warmup and bias correction enabled. The initial step size (γ_k) is set to 1, and we use a weight decay of 0.01. We train the model for $15,000$ steps with a batch size of 6, leveraging Low-Rank Adaptation (LoRA) with $r = 256$ and $\alpha = 256$. We perform all training on a single NVIDIA H100 GPU (80GB), requiring approximately 90 hours for IDM-VTON and 100 hours for InsertAnything.

Qualitative Results for Single-View Baselines. Fig. 23 provides visual examples of the Single-View VTON Baselines, complementing the quantitative metrics reported in the main paper (Tab. 3). Both models achieve visually high-fidelity and geometrically plausible try-on results, successfully transferring the garment texture and structure onto the target person's pose. Notably, InsertAnything, which yielded the best quantitative scores, demonstrates exceptional realism, confirming that the frontal-paired subset of MV-Fashion is directly compatible with existing state-of-the-art VTON pipelines.

Semantic Controllability Qualitative Results. Fig. 24 provides additional examples of the Semantic Controllability experiment with draping style. As discussed in the main paper, when finetuning with MV-Fashion draping style-augmented data, we observe that the model responds to the styling prompt (column (b)) compared to the no-style prompt (column (a)). In both rows shown, the garments visually react to the prompt by opening the jacket buttons/zip. Additionally, we also expand the experiment that tests the model's capability when finetuning with fitting style-augmented data. Interestingly, the model shows an emerging ability to follow these styling instructions when comparing the styling prompt (column (d)) to the no-style prompt (column (c)). Specifically, the upper garment in the

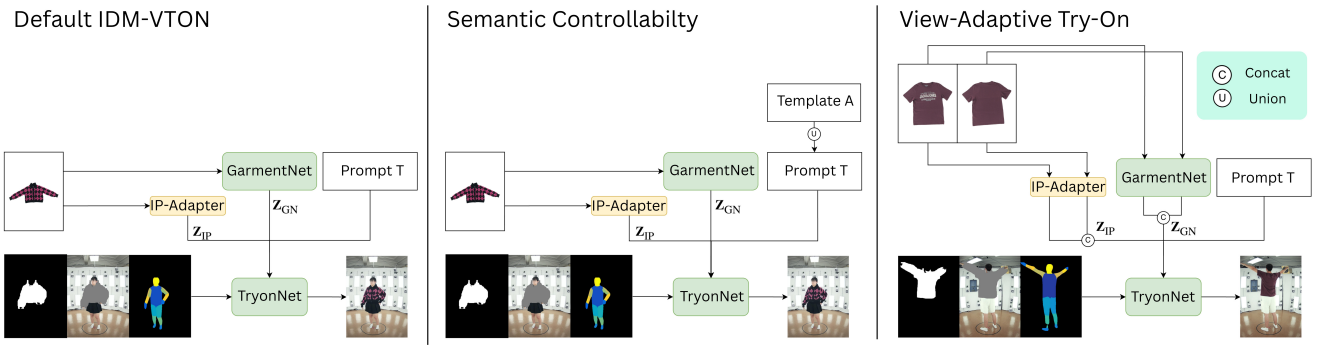

Figure 20. IDM-VTON architecture adaptation to perform Semantic Controllability and View-Adaptive Try-On tasks. For Semantic Controllability, we augment the prompt T by styling annotation ($\mathbf{A}$). For View-Adaptive Try-On, we introduce feature fusion before conditioning, which concatenates the IP-Adapter's features ($\mathbf{Z}_{IP}$) and GarmentNet's features ($\mathbf{Z}_{GN}$) of both the frontal and rear garment views.

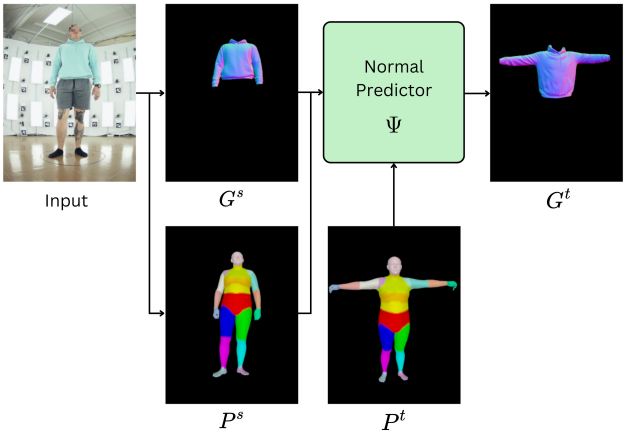

Figure 21. **The normal predictor (Ψ).** To disentangle intrinsic garment geometry from pose deformations, the network transforms the source garment normals G^s into a canonical, unposed state G^t. Both G^s and P^s are extracted from the input frame, and the transformation is explicitly conditioned on a target pose P^t. Crucially, both pose maps (P^s, P^t) utilize semantic body segmentation to provide spatial correspondence cues.

first row fits closer in the sleeve region, aligning with the "regular fit" prompt, while the garment in the second row becomes tighter in the chest area, aligning with the "slim fit" prompt.

Multi-View Qualitative Comparison. Fig. 25 provides the qualitative comparison between the two Multi-View Geometric Analysis experiments. Column (a), representing the Cross-View Geometric Test, shows the model's limitations when synthesizing rear target poses using only the frontal catalogue image. We observe issues stemming from perspective misinferring and poor cross-perspective consistency, leading to structural distortion and inaccurate pattern mapping. In contrast, column (b), the View-Adaptive Try-On, demonstrates a slight visual improvement. By utilizing both frontal and rear garment images, the model successfully mitigates several geometric and texture errors, achieving better pattern alignment and structural fidelity in the rear poses across examples.

In The Wild Images We further test our VTON baseline trained on MV-Fashion on in-the-wild images from Unsplash to evaluate its robustness. Our model shows promising generalization to these out-of-distribution samples, suggesting that our controlled setup translates effectively to less constrained environments. For a qualitative evaluation, please refer to Fig. 22.

Figure 22. Qualitative examples of our VTON pipeline when using in-the-wild images with complex backgrounds. Our model demonstrates effective generalization to these out-of-distribution samples. (images obtained from unsplash.com)

Conclusion. Overall, the results show that MV-Fashion dataset is directly compatible with standard Single-View VTON research. For Semantic Controllability, despite the observed successes, the overall task of fine-grained semantic control remains highly difficult and will require significant architectural advancements to achieve reliable and consistent control, for which our dataset serves as an essential testbed. For View-Adaptive Try-On, a partial re-

duction in failure cases validates the potential of MV-Fashion's multi-view pairs to enable effective viewpoint-aware VTON research, though achieving perfectly seamless cross-perspective fusion remains a valuable future investigation.

11.2. Size Estimation

Training Setup. We adhere to a consistent protocol for all baseline variants. Input images are resized to 256×256, while we utilize the standard Adam optimizer [39] with a learning rate of 1×10^{-4} and a batch size of 24. All models are trained on a single NVIDIA GeForce RTX 4090 GPU (24GB). Training duration varies by component: the canonical normal predictor (Ψ) requires approximately 34 hours to train for 5 epochs. In contrast, the size regressor (Φ) is lightweight; it takes less than one hour to train for 100 epochs (per model variant). We also train an *End-to-End* baseline in which Ψ and Φ are initialized from their independent pretrained models, Ψ is frozen and Φ is fine-tuned for 5 epochs in approximately 18 hours.

Qualitative Results. In Fig. 26, we provide qualitative results of the normal predictor across all six garment groups (G1-G6). These examples illustrate the inherent difficulty of the task: the source normals (G^s) are often heavily occluded (see G1) or distorted by perspective and body leaning (see G4 and G5). Despite these geometric challenges, the model successfully recovers the flattened, canonical shape of the garment. Furthermore, in the case of G3, the input normal map exhibits severe self-occlusion due to the leg movement; however, the predictor successfully recovers the missing geometry to reconstruct a complete canonical skirt.

Quantitative Results. Tab. 4 reveals a clear correlation between data frequency (see Tab. 2) and model performance for the *Per-Group* variant. Groups G1 and G2, which constitute the majority of the dataset (38.3% and 33.0%, respectively), achieve significantly lower errors than the statistically rarer groups. However, the *Multi-Task* strategy effectively mitigates this data imbalance; notably, it reduces the error for G5 from 12.109 cm to 4.295 cm.

To further analyse this improvement, we provide detailed breakdowns of Tab. 4 in Tabs. 8-13, reporting the error per measurement part (as defined in Sec. 9.5) for each garment group. These breakdowns demonstrate how the *Multi-Task* variant leverages shared measurement parts across different garment groups. For instance, the *Leg Length* error in G5 is reduced by over 26 cm compared to the *Per-Group* baseline.

Finally, to assess the feasibility of estimating size without relying on ground truth canonical garment normals (G^t), we introduce the *End-to-End* baseline results in Tab. 14. This model fine-tunes the best-performing variant of Φ (*Multi-Task + SwinV2*) to regress sizes directly from the input image. While this approach naturally yields a higher MAE than the others, it achieves a respectable average error of 6.134 cm. This demonstrates that the signals within MV-Fashion enable direct, image-based garment size learning.

Conclusion. The results indicate our annotations are sufficient to train viable garment size estimation models. By enabling the learning of mappings from in-the-wild deformations to garment measurements, MV-Fashion establishes a strong foundation for future research in cloth sizing.

Measurement	Per-Group	Multi-Task	Multi-Task + SwinV2
Neck	2.400 (1.976)	2.581 (2.163)	2.246 (1.852)
Chest	5.170 (4.053)	6.370 (4.423)	5.339 (3.946)
Waist	4.265 (4.114)	4.516 (4.344)	4.581 (4.490)
Bottom	3.553 (2.921)	4.347 (2.652)	3.659 (3.118)
Sleeve	7.903 (14.074)	8.022 (14.391)	7.058 (14.679)
Bicep	1.672 (1.177)	1.838 (1.418)	1.620 (1.096)
Armhole	2.461 (2.068)	2.538 (2.044)	2.459 (2.057)
Shoulder	4.605 (3.748)	4.592 (4.024)	4.986 (3.768)
Body Height	6.612 (6.712)	6.965 (6.925)	5.568 (6.954)
Sleeve Cuff	2.026 (1.646)	1.782 (1.492)	1.758 (1.513)
Average	4.069 (5.861)	4.361 (6.061)	3.933 (5.994)

Table 8. Detailed breakdown of Tab. 4 for group G1.

Measurement	Per-Group	Multi-Task	Multi-Task + SwinV2
Bottom Waist	4.177 (2.573)	4.266 (2.607)	3.572 (2.696)
Bottom Hip	3.223 (2.249)	3.609 (2.263)	2.993 (2.291)
Thigh	1.697 (1.302)	2.027 (1.515)	1.767 (1.316)
Leg Cuff	2.459 (2.415)	2.566 (2.669)	1.991 (2.171)
Front Crotch	2.561 (1.854)	2.803 (1.977)	1.946 (1.709)
Back Crotch	2.552 (2.404)	2.603 (2.402)	2.784 (2.243)
Leg Length	4.569 (3.315)	4.493 (3.451)	5.069 (3.264)
Full Length	2.812 (2.480)	3.117 (2.625)	4.412 (3.664)
Average	3.006 (2.541)	3.185 (2.619)	3.067 (2.762)

Table 9. Detailed breakdown of Tab. 4 for group G2.

Measurement	Per-Group	Multi-Task	Multi-Task + SwinV2
Bottom Waist	6.325 (4.495)	3.938 (3.331)	4.367 (2.685)
Bottom Hip	8.126 (4.236)	3.894 (2.101)	3.760 (2.766)
Bottom Bottom	17.300 (12.239)	15.506 (14.732)	14.483 (17.524)
Full Length	8.053 (5.055)	4.049 (2.694)	3.962 (3.377)
Average	9.951 (8.437)	6.847 (9.174)	6.643 (10.135)

Table 10. Detailed breakdown of Tab. 4 for group G3.

Measurement	Per-Group	Multi-Task	Multi-Task + SwinV2
Neck	1.768 (1.277)	2.444 (2.284)	1.815 (1.955)
Chest	6.279 (4.443)	7.493 (5.830)	5.996 (4.814)
Waist	4.656 (3.858)	4.946 (5.611)	3.721 (3.169)
Bottom	6.131 (5.767)	6.922 (6.804)	5.171 (5.246)
Sleeve	7.784 (6.336)	5.717 (4.442)	5.842 (4.986)
Bicep	2.319 (1.907)	2.116 (1.439)	1.619 (1.300)
Armhole	3.378 (2.535)	2.655 (1.942)	2.789 (2.070)
Shoulder	4.795 (3.342)	6.513 (5.562)	4.960 (3.969)
Body Height	11.390 (18.222)	15.403 (20.404)	11.106 (15.482)
Sleeve Cuff	2.419 (2.500)	2.230 (1.713)	1.344 (1.314)
Average	5.092 (7.377)	5.644 (8.563)	4.436 (6.537)

Table 11. Detailed breakdown of Tab. 4 for group G4.

Figure 23. Qualitative results of IDM-VTON and InsertAnything trained on the Single-View MV-Fashion benchmark. The high fidelity achieved by both models confirms the direct compatibility of MV-Fashion with existing state-of-the-art VTON datasets. The columns represent: (a) Input (Garment and Person Mask), (b) IDM-VTON Try-On Result, (c) InsertAnything Try-On Result, (d) Ground Truth.

Measurement	Per-Group	Multi-Task	Multi-Task + SwinV2
Neck	8.058 (2.290)	3.641 (2.401)	1.132 (0.590)
Chest	10.098 (1.426)	4.768 (2.114)	5.174 (1.231)
Waist	10.615 (1.759)	2.701 (1.207)	2.721 (1.042)
Bottom	11.753 (2.834)	3.985 (2.504)	3.945 (2.169)
Armhole	2.083 (1.326)	6.951 (1.245)	10.211 (2.226)
Shoulder	9.791 (1.785)	6.528 (3.385)	6.931 (3.096)
Body Height	2.657 (1.690)	6.099 (3.673)	18.046 (3.976)
Bottom Waist	13.330 (2.995)	2.774 (1.139)	3.435 (2.390)
Bottom Hip	17.475 (2.204)	2.865 (1.650)	4.415 (2.980)
Thigh	6.026 (2.525)	1.842 (1.769)	1.733 (0.960)
Leg Cuff	7.653 (1.858)	7.202 (1.799)	2.856 (2.309)
Front Crotch	3.712 (2.599)	6.807 (1.926)	4.859 (1.187)
Back Crotch	10.100 (2.872)	2.740 (1.297)	2.812 (1.832)
Leg Length	28.879 (6.945)	2.204 (2.118)	3.070 (1.579)
Full Length	39.399 (9.268)	3.314 (1.711)	1.991 (1.654)
Average	12.109 (10.313)	4.295 (2.746)	4.889 (4.633)

Table 12. Detailed breakdown of Tab. 4 for group G5. **Note:** *Sleeve*, *Bicep*, and *Sleeve Cuff* measurement parts are missing because all instances in the test set are sleeveless.

Measurement	Per-Group	Multi-Task	Multi-Task + SwinV2
Neck	4.285 (2.699)	2.944 (2.135)	2.995 (2.366)
Chest	8.361 (10.686)	6.902 (7.048)	6.050 (5.117)
Waist	7.555 (7.777)	4.553 (3.660)	3.700 (3.120)
Bottom	8.080 (7.073)	4.268 (3.630)	3.787 (2.954)
Sleeve	6.575 (4.483)	4.183 (2.600)	6.146 (4.889)
Bicep	3.446 (2.441)	2.957 (2.528)	3.781 (2.119)
Armhole	3.419 (2.241)	3.965 (2.777)	3.307 (2.515)
Shoulder	10.579 (12.815)	8.317 (8.678)	7.850 (6.972)
Body Height	12.486 (7.420)	15.193 (11.383)	13.174 (7.387)
Sleeve Cuff	3.619 (2.373)	2.041 (1.122)	2.410 (1.326)
Bottom Waist	8.159 (8.206)	5.857 (4.638)	5.724 (5.929)
Bottom Hip	9.029 (6.733)	7.764 (7.555)	8.634 (8.486)
Bottom Bottom	14.577 (9.603)	13.631 (10.468)	10.106 (9.316)
Full Length	5.804 (4.066)	6.116 (3.966)	9.826 (8.118)
Average	7.767 (7.954)	6.549 (7.234)	6.389 (6.562)

Table 13. Detailed breakdown of Tab. 4 for group G6.

11.3. Novel View Synthesis

Training Setup. To train each of the baseline methods, we run them for 50,000 iteration. Unless specified, we use the default parameters. For **instant-ngp**, we had to reduce the *–pipeline.target-num-samples*: ”The target number of samples to use for an entire batch of rays” value to 16,384 due to the implementation, which uses `nerfacc` library. This initializes the target volume in the beginning of the optimization, and with a large number of rays it requires significant amount of video memory. For **nerfacto and splatfacto** we use their ”big” variants, namely `nerfacto-big` and `splatfacto-big` which increases the number of parameters used and achieves slightly better results.

Ablation Study. We perform an ablation study over the effect of the number of views used for training. We evaluate

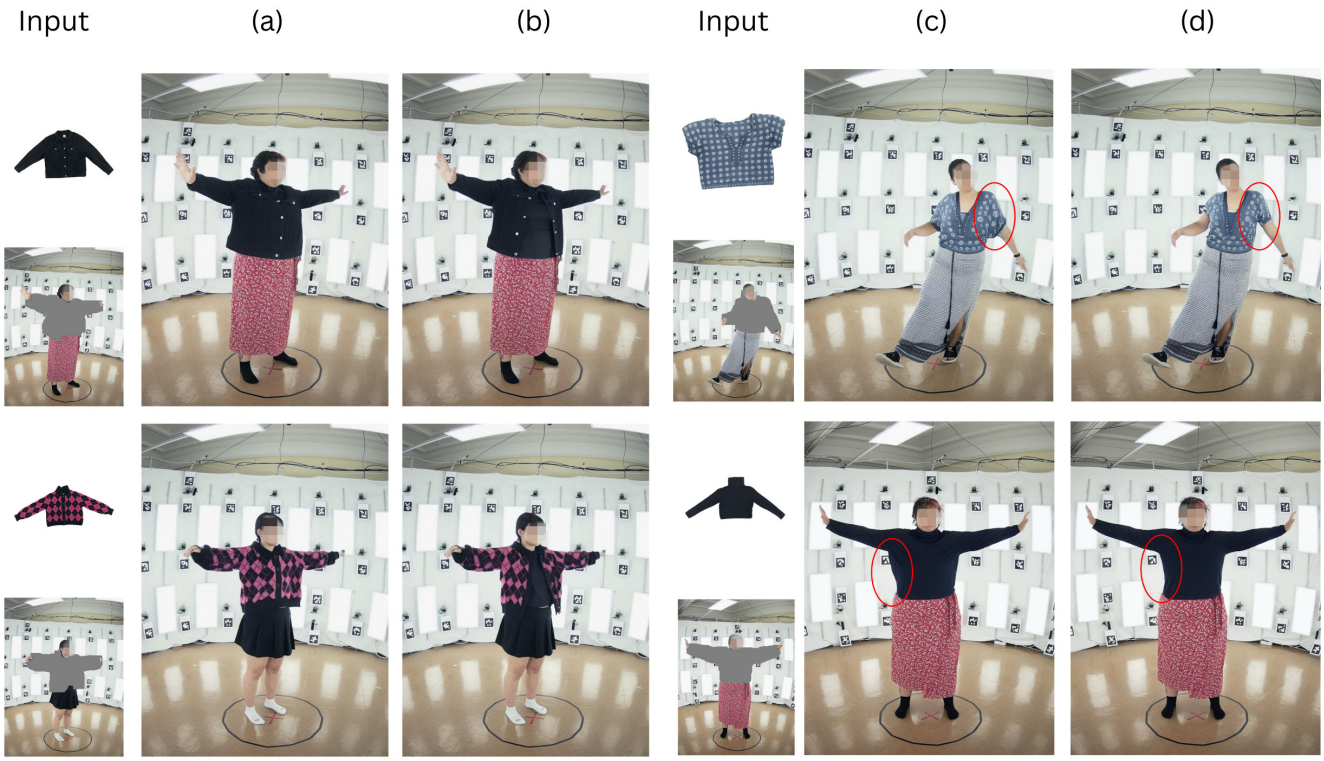

Figure 24. Extended qualitative results for Semantic Controllability. The original experiment augments draping style in prompt when training and compare (a) no-style to (b) draping styling prompt (*e.g. outerwear is fully open* for both rows) at test time. Additionally, we test the second experiment to train with fitting style and compare (c) no-style to (b) fitting styling prompt (*e.g. regular fit* for row 1 and *slim fit* for row 2) at test time.

Group	Per-Group	Multi-Task	Multi-Task + SwinV2	End-to-End
G1	4.069 (5.861)	4.361 (6.061)	3.933 (5.994)	5.316 (6.672)
G2	3.006 (2.541)	3.185 (2.619)	3.067 (2.762)	3.965 (3.789)
G3	9.951 (8.437)	6.847 (9.174)	6.643 (10.135)	10.565 (13.070)
G4	5.092 (7.377)	5.644 (8.563)	4.436 (6.537)	5.924 (6.705)
G5	12.109 (10.313)	4.295 (2.746)	4.889 (4.633)	6.347 (5.192)
G6	7.767 (7.954)	6.549 (7.234)	6.389 (6.562)	11.076 (11.976)
Average	4.904 (6.533)	4.710 (6.392)	4.279 (5.870)	6.134 (7.832)

Table 14. Quantitative results for the size regressor (Φ). We report MAE in cm as mean (std). This table expands Tab. 4 with the *End-to-End* baseline, which regresses size directly from the input image. The comparison assesses the performance with and without relying on ground truth canonical garment normals (G^t). Groups and model variants are defined in Sec. 3.3 and Sec. 4.2 of the main paper, respectively.

how the quality of the final results compares when using a very low number of input views (4), increase the number of views until we add all training views. All tests are performed on the RPi cameras only for consistency, but we also evaluate the higher resolution Bolt cameras. Since we only have 8 of Bolts available, the ablation is limited, but we test training on 4 random or 6 total Bolt cameras and evaluate on the remaining two. The metrics we use are affected by the resolution change. Therefore, they are not comparable between RPi and Bolt experiments. We restrict evaluation to same-resolution settings to ensure metric consistency.

Comparison of Baseline Methods. As discussed in the main paper, instant-ngp struggles with the fine details and even some geometrical features are missing or present artifacts. `nerfacto` reproduces the details better, though it is still prone to showing some geometrically incorrect artifacts. On the other hand, `splatfacto` can reproduce fine detail and is more geometrically correct. It can still present some artifacts specific to 3d gaussian splatting when the novel viewpoints are highly different compared to the train-

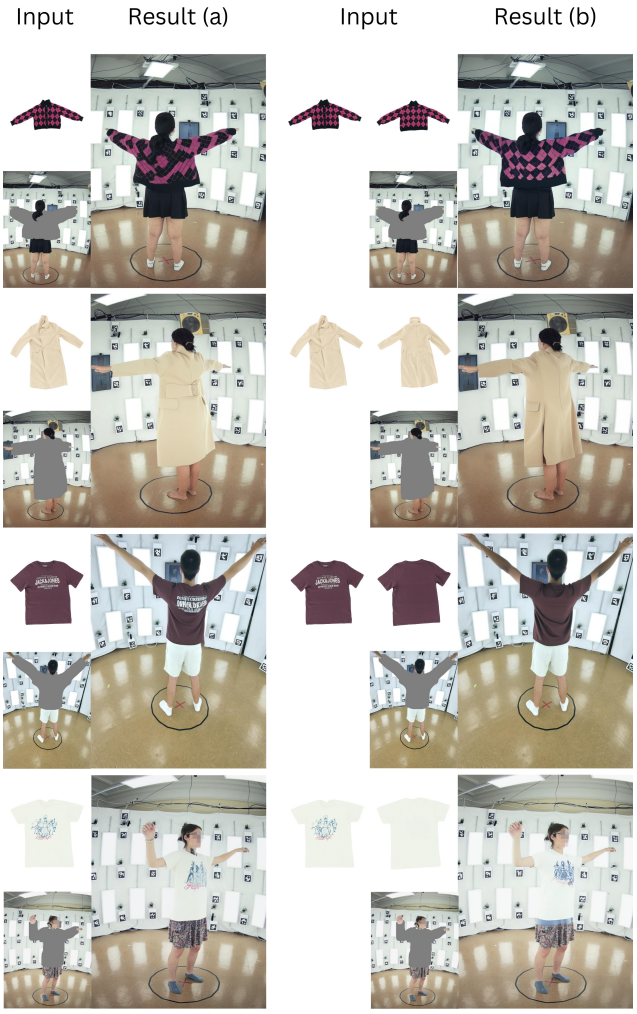

Figure 25. More qualitative results of IDM-VTON on (a) *Cross-View Geometric Test* vs (b) *View-Adaptive Try-On*. The updated IDM-VTON architecture can map between the catalogue view and the person's pose when both frontal and rear images of the garment are provided to the model.

ing views. We present additional qualitative results from more subjects and from multiple views in Fig. 27.

Ablation of Different Number of Training Views. As shown in Tab. 5, adding more training views improves the results as expected, though the results start saturating with more views. Performance saturation at higher view counts indicates diminishing returns. This suggests that the 60-camera setup provides an optimal trade-off between reconstruction fidelity and system complexity. We show some qualitative examples in Fig. 29.

Additionally, we evaluate the results with only the Bolt cameras, as can be seen qualitatively in Fig. 28 and quantitatively in Tab. 15. One can observe that `splatfacto` produces similar quality images, using the higher resolution Bolt cameras, to RPis. This is despite the fact that all Bolt cameras are at a low or high viewing angle, and none have a straight, perpendicular view to the subject. Since there are only 8 Bolt cameras designed to be spatially distributed, artifacts are visible in the validation views, as these are highly different from the training ones.

Views	PSNR ↑	SSIM ↑	LPIPS ↓
4	24.089(1.806)	0.952(0.012)	0.073(0.016)
6	25.831(1.777)	0.956(0.012)	0.061(0.013)

Table 15. Image quality metrics when using only Bolt cameras with `splatfacto`. Note that these values are not directly comparable with Tab. 5 due to the resolution difference. Also, because of the limited number of cameras, these values are relatively low.

4D Gaussian Splatting We run CEM-4DGS [35] with the default settings from the authors to validate our dataset for the application of dynamic novel view synthesis. We ran the method on a sequence of 20 frames from which the first seven can be seen in Fig. 30. This shows that while parts of the person with more movement like the hands are slightly blurry, the overall reconstruction is of acceptable quality. CEM-4DGS and similar methods treat static and dynamic elements of the scene separately, as such, they produce a sharp background similar to our 3DGS tests. Quantitative results can be seen in Tab. 16.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
CEM-4DGS	24.395	0.875	0.151

Table 16. Image quality metrics of the 4DGS reconstruction. As expected, slightly lower than static reconstruction, but still good quality. Values shown are means across all frames of the sequence.

Conclusion. Even with our relatively low-resolution cameras, we are able to achieve highly realistic novel view synthesis, which leads us to believe that MV-Fashion is suited for downstream tasks involving the discussed methods.

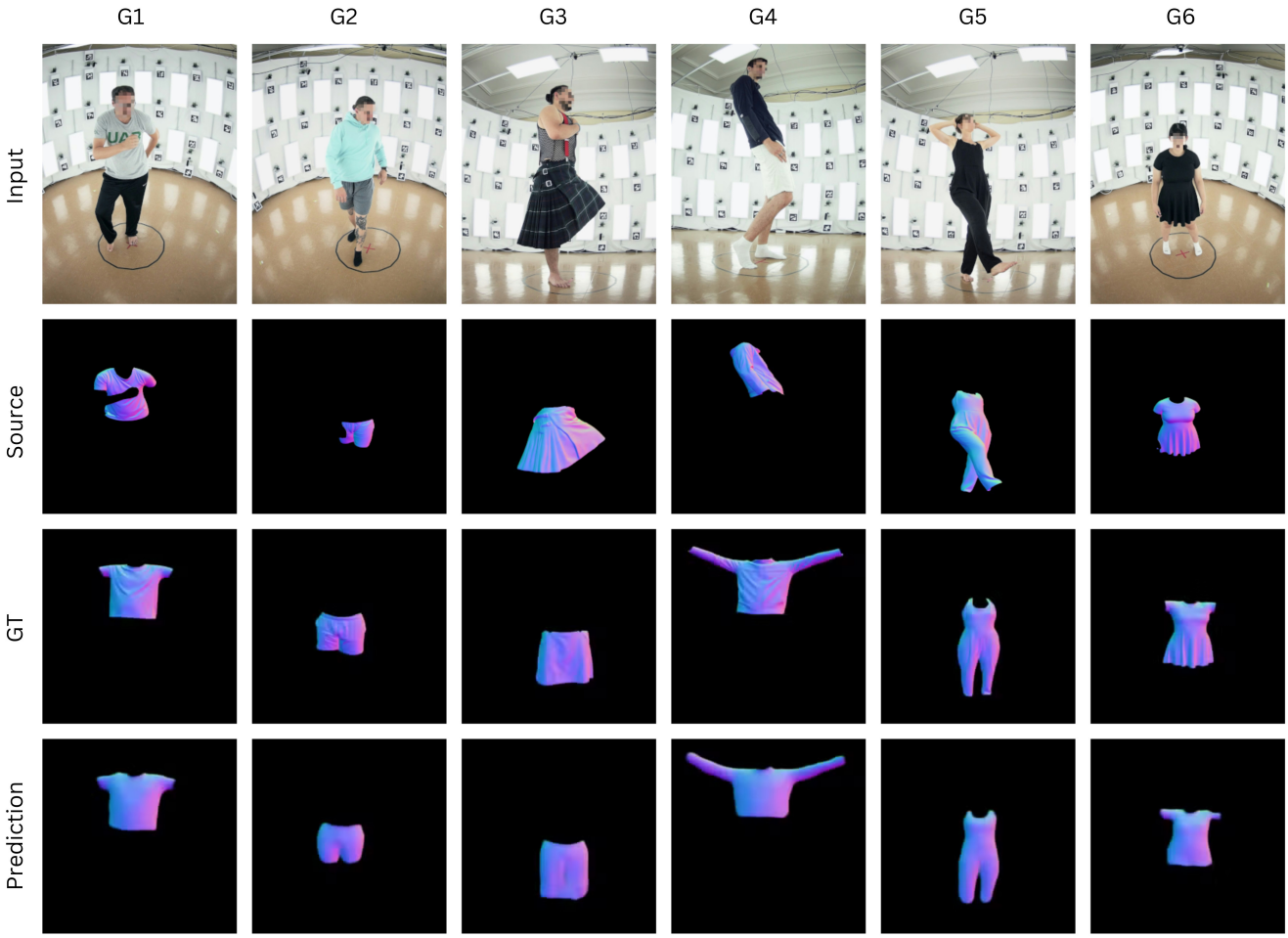

Figure 26. Qualitative results of the garment normal predictor (Ψ) across all garment groups (G1-G6). From top to bottom: the input RGB frame, the source garment normal map (G^s), the target ground truth canonical normal map (G^t), and the predicted canonical normals. The results demonstrate the model's ability to disentangle pose from garment geometry, effectively recovering the intrinsic shape even in cases of occlusion (e.g., G1) or complex deformations (e.g., G4).

Figure 27. Additional examples of the three NVS methods tested. Instant-NGP produces blurry results, while `nerfacto` is sharper, but it has some distinct artifacts. `splatfacto` on the other hand provides superior quality.

Figure 28. Qualitative results of the ablation over the number of views used for `splatfacto` when using only Bolt cameras. Because of the limited number of available cameras there are many artifacts.

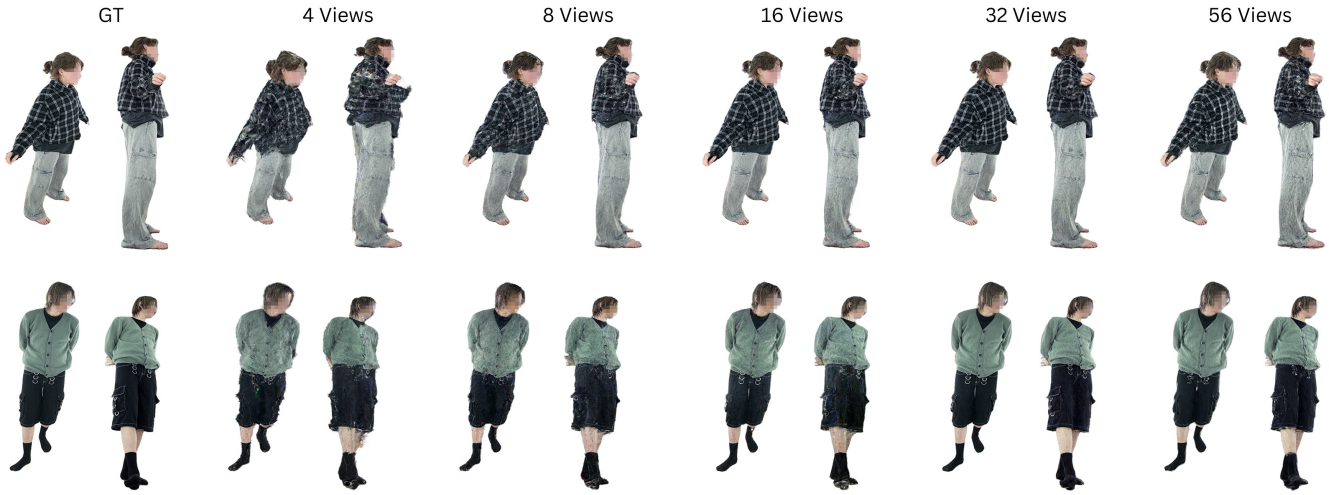

Figure 29. Qualitative results of the ablation over the number of views used for `splatfacto`. It shows that using very few views produces artifacts, but from 16 views it starts to improve and when using all views the results are almost indistinguishable from the ground truth.

Figure 30. Qualitative results of the dynamic novel view synthesis experiments. The results show slight blurring around the hands, but otherwise prvode good quality outputs. The frames shown are 6 consecutive frames from a 20 frame sequence.