7 Appendix

7.1 Blender Simulation Parameters

We provide additional details on the Blender scene setup and parameters used to generate our combined shirt dataset (including both suspended in-air and on-table configurations). The ratios of shirt features are selected to loosely reflect the distribution of shirts we test on the real system. Rendering 50 scenes with these parameters takes 10 hours on an NVIDIA RTX 4090 GPU.

Blender 4.2 Simulated Shirt Scene Dataset Parameters	
Scene Parameters	
Shirt Suspended in Air Scenes	1000 scenes
Shirt on Table Scenes	500 scenes
Cameras Rendered per Scene	3 cameras
Fabric Quality Steps	10
Render Quality	64
Shirt Parameters	
Mesh Vertex Density	2922
Shirt Thickness	0.4 mm
Sleeve Length Ratio in Dataset	65% short sleeve, 35% long sleeve
Neck Type Ratio in Dataset	80% U-Neck, 20%V-Neck
Collar Hem Ratio in Dataset	80% collar hems, 20% without collar hems
Bottom Hem Ratio in Dataset	70% without bottom bodice hems, 30% bottom bodice hems
Shirt Stiffness Range	Uniformly sampled between [7, 15]
Shirt Damping Range	Uniformly sampled between [5, 7]

Table 2: Scene parameters used for dataset generation in Blender 4.2.

7.2 Folding with Confidence-Based State Machine

We allow the robot to choose the most appropriate folding pick points based on which points it can confidently identify and grasp. Figure 7 shows the four different folding strategies (shoulder to shoulder, bottom to bottom, sleeve to sleeve, sleeve to bottom). Bottom refers to the bottom corner of the shirt, and sleeve refers to the bottom edge of the sleeve. The system starts by picking the shirt up from the table (looking for high-confidence correspondence regions), and all subsequent grasps are performed in air.

At each grasp attempt, the robot can query from three canonical regions (shoulder, sleeve, bottom) using our distributional dense correspondence network to generate confidence-weighted heatmaps. A grasp is executed only if both the correspondence confidence and grasp affordance (for hanging grasps) exceed predefined thresholds. Grasp success is validated by our tactile classifier (confirming fabric contact). If no grasp is attempted or the grasp attempt fails, the robot rotates the garment by 30° and re-evaluates. In cases where symmetry matters (e.g. grabbing the sleeve and end on same side of the shirt), we use the heuristic that the opposite corner features would be the lowest point, and therefore we mask out the bottom. If no pixel meets the threshold requirements, the robot grasps the lowest available high affordance point to change configurations and encourage the cloth to unfurl.

The very first grasp attempt is done on the table. If no high correspondence point is found within the robot's workspace, the robot's fallback strategy is to grasp the highest point. All subsequent grasps are performed in air. The robot continues switching arms until it has two successful grasps.

Once the shirt is grasped by two keypoints, the robot pulls the shirt until it is tensioned. We use shear as measured by marker tracking on the tactile sensor as an indication for when the shirt is in tension. Then, the robot brings the lifted shirt to one end of the workspace, lowers it to the table, lowers the grippers to the other end of the table while resting half the shirt, then folds the shirt over as the grippers return to the first side of the workspace. The robot uses vision to align the corners in the final folding motion.

Even with the confidence-based state machine, however, irrecoverable failure modes still occur. Figure 8 shows examples of these cases. Correspondence failures that result in grasps of internal points on the shirt (such as the body), grasping the correct feature but on the opposite side of the shirt, and grasping too many layers of fabric are some examples of failures that occur while folding.

We break down the failure cases by task:

Failure Breakdowns of 10 Trials	**Folding**	**Hanging**
Incorrect Correspondence	1	3
Diagonal Feature Grasp	2	0
Grasped Excess Layers	1	0
Total Failures	**4/10**	**3/10**
Total Successes	**6/10**	**7/10**

Recoverable failures include affordance failures leading to insufficient cloth in the grip and the cloth slipping out of the grip. Our tactile classifier informs the system if each grasp is successful. We use vision to ensure that the cloth is still in grip after moving the grippers.

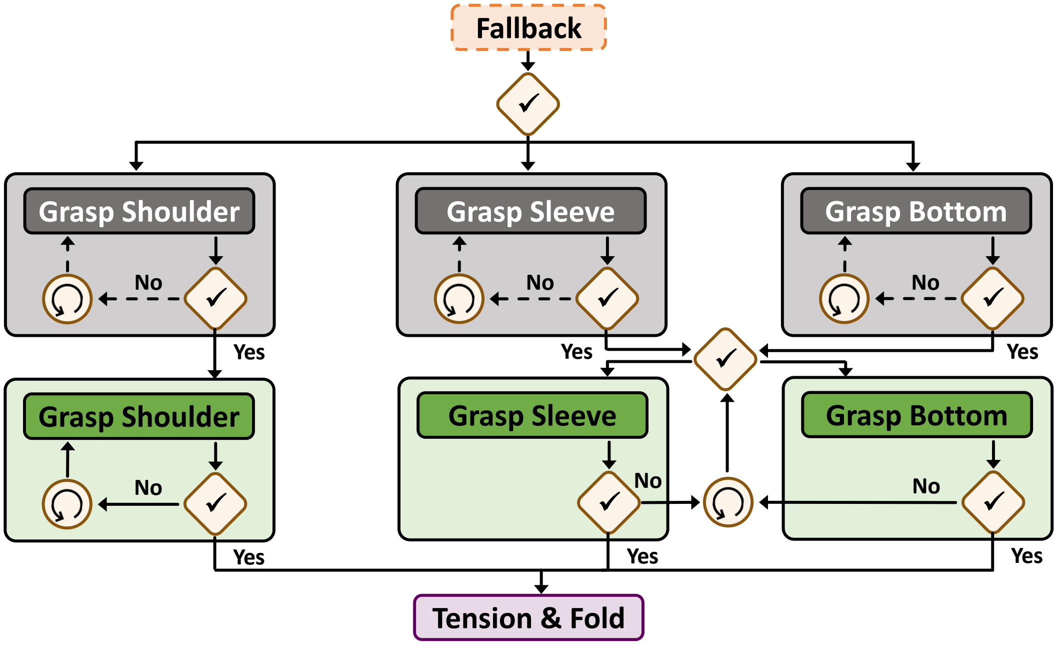

Figure 7: Confidence-based state machine for folding strategy. The robot dynamically chooses between folding strategies based on which points are visible and graspable. The initial grasp occurs on table, where the fallback strategy for low confidence is grasping the highest point. All subsequent grasps are attempted in air. The robot only attempts a grasp if correspondence confidence and grasp affordance exceed predefined thresholds. If no point is graspable, the robot rotates. If the robot completes a full rotation, the new fallback option is grabbing the lowest graspable point to help unfurl the cloth. Once two successful grasps are made, the robot tensions the cloth and folds.

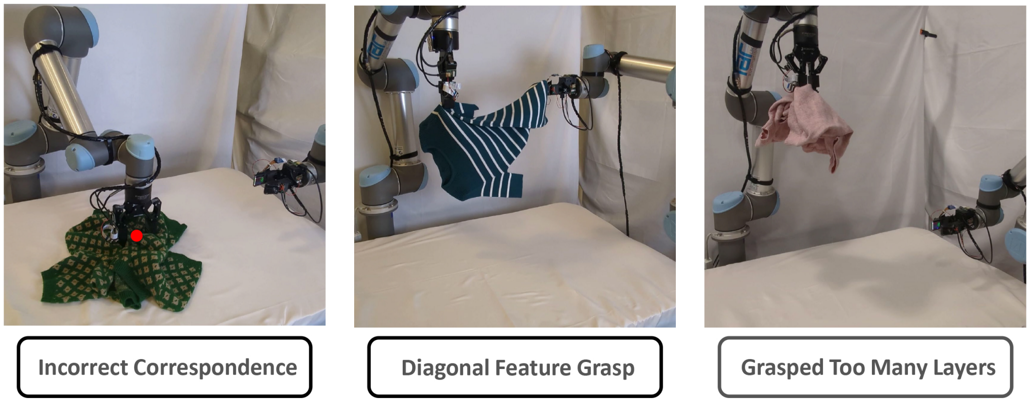

Figure 8: Irrecoverable Failure Modes of Folding. Though the confidence-based state machine is able to recover from mistakes in folding, some cases are unaccounted for and irrecoverable in the system. Incorrect correspondence grasps, picking the correct feature but on the wrong side, and grasping too much cloth are some of the failure cases.

7.3 Dense Correspondence Network Parameters

The mapping function f that generates the dense descriptor space is implemented as a 34-layer ResNet (pretrained on ImageNet) with a stride of 8 for computational efficiency (as in [26]). Bilinear upsampling is applied to the network's feature maps to align the output descriptor maps with the input image size (540×960 pixels). We train each of our final networks for approximately 10,000 iterations, which takes under 2 hours on an NVIDIA RTX 4090 GPU.

Hyperparameter Tuning We conducted a series of hyperparameter experiments to optimize the performance of our dense correspondence network. A key parameter was the descriptor dimension d, which controls the capacity of the embedding space. As shown in Figure 9, we tested dimensions of 3, 9, 16, and 25. A descriptor size of $d = 16$ consistently outperformed smaller and larger alternatives, striking a balance between sufficient representational capacity and generalization. Lower dimensions (e.g., $d = 3$) lacked expressivity, while higher dimensions (e.g., $d = 25$) did not offer noticeable improvements and introduced potential overfitting. Additionally, larger networks require more computation time.

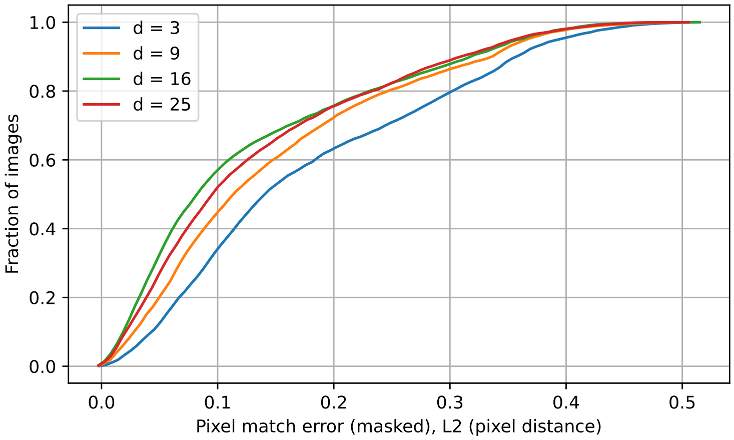

Figure 9: Cumulative pixel match error across different descriptor dimensions (d) evaluated on the simulated test set. The network was trained on a combined dataset of hanging and table shirts. A descriptor size of $d = 16$ provides the best trade-off between representational capacity and generalization, outperforming both smaller ($d = 3$, $d = 9$) and larger ($d = 25$) dimensions.

We also evaluated the effect of σ, the standard deviation of the Gaussian used for the distributional loss target. Figure 10 shows performance across σ values of 1, 2, 10, and 20. While $\sigma = 1$ yielded

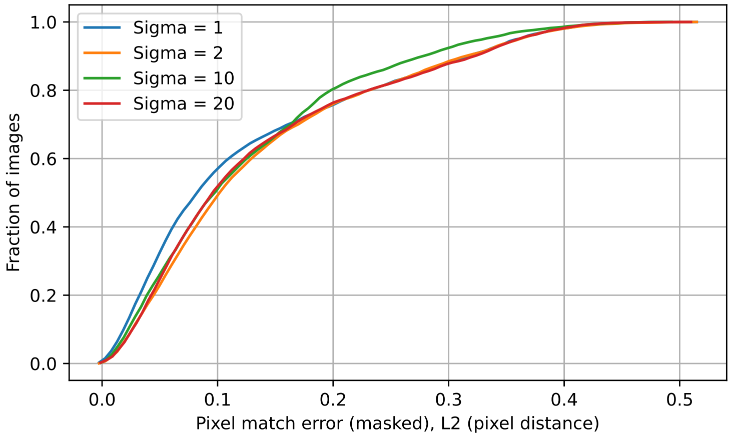

Figure 10: Cumulative pixel match error for different Gaussian σ values used in the distributional loss target. The network was trained on a combined dataset of hanging and table shirts. Smaller σ values (e.g., $\sigma = 1$) produce sharper distributions and yield slightly better accuracy in simulation, but larger σ values improve generalization to real-world data by promoting smoother gradients in the descriptor space.

sharper distributions and slightly better accuracy in simulation, we found that larger σ networks generalized better to real-world data. We hypothesize that broader Gaussians produce smoother gradients across the descriptor space, which in turn leads to more stable and consistent correspondence predictions. This smoothing effect could help mitigate sensitivity to local noise, masking artifacts, or out-of-distribution lighting. Sharper distributions (from smaller σ) can lead the network to overfit to high-frequency details in the simulated data, which don't transfer well to real-world images.

Model and Dataset Design Choices During early testing, we also experimented with several architectural variations. We evaluated higher-resolution ResNets and a DINOv2 backbone for the mapping function f, but found that DINOv2 performed significantly worse given our limited dataset size, and the higher-resolution ResNets did not yield noticeable improvements in correspondence accuracy. To improve confidence estimation, we attempted to train a separate confidence head; however, this approach did not reliably predict correspondence accuracy.

Additionally, our initial training dataset lacked hem and seam details, which led to poor differentiation between sleeve and torso ends when applied to real garments. Including these structural details in later dataset versions improved real-world performance. For hanging datasets with 1000 scenes, our best descriptor network with seams had a 73.3% classification success rate in the forward direction. Without seams, the best network had a 42.2% success rate. This network often misclassified sleeve regions as bottom regions.

We also experimented with incorporating depth information alongside RGB inputs but observed no significant gains. This suggests that in our cloth manipulation tasks, texture and color cues dominate the correspondence signal, and depth alone does not meaningfully contribute to distinguishing garment regions.

We found that adding artificial occlusions to training images did not seem to impact performance with simulated images (Figure 11), suggesting that the network was robust to minor occlusions. However, training with occlusions significantly improved performance on real systems, likely due to masking artifacts.

We compare performance of networks trained on exclusively hanging or table scenes to networks trained on a combined dataset (Figure 11, 12). The combined network performs marginally worse in both test sets compared to the specialized networks, but does not have significant performance loss. We found that simplifying table configurations during training to be more representative of those used in related works was necessary for improving the combined network's performance. The harder table training set had few distinguishing features, making correspondences more difficult to learn.

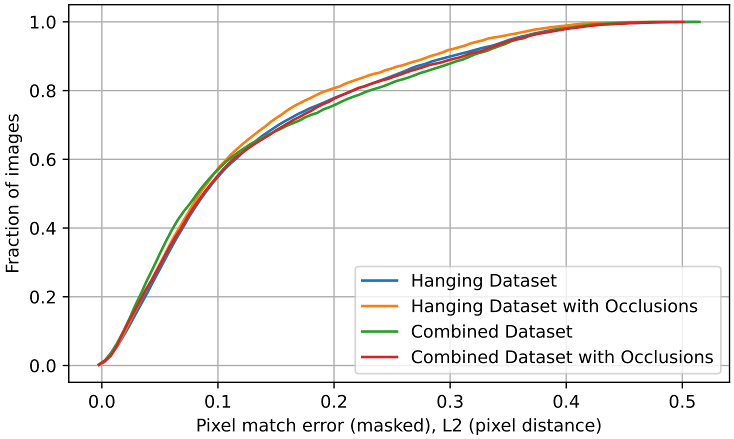

Figure 11: Cumulative pixel match error on hanging shirts for networks trained on hanging and combined (hanging and table) datasets with and without occlusions. The networks all perform similarly in simulation, but we found that on real data, occlusions and the specialized hanging network both performed better.

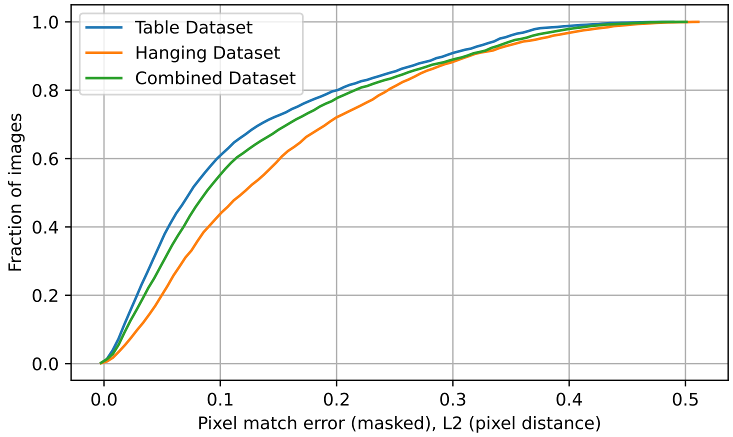

Figure 12: Cumulative pixel match error on shirts on a table for networks trained on table, hanging, and combined (hanging and table) datasets. As hypothesized, the specialty table network performs the best, followed by the network trained with the combined dataset. The hanging network is able to generalize its understanding to shirts on tables, but to a lesser degree of accuracy.

7.4 Dense Correspondence Evaluation

Our dataset and every single experiment include long-sleeve shirts, varied necklines, and a broad range of materials (including silk blends, stretchy spandex, fuzzy sweaters, etc.). We evaluate the real-world performance of our dense correspondence network using the color-coded regional classifications defined in Figure 13. In both folding and hanging scenarios, multiple grasp points can lead to the successful execution of a given strategy. Instead of requiring exact pixel-level matches, we divide the shirt into five regions and consider a trial successful if the network's high-confidence grasp prediction falls within the correct region on the physical shirt.

We conducted ROC studies to help determine optimal correspondence thresholds in simulation (where pixel-level ground truth is available). However, we noted that real-world transfer introduced high variability. In practice, we found that a confidence threshold of 6×10^{-6} reflects a clear inflection point where networks begin to assign high confidence to meaningful regions in real images. Individual pixel confidences peak at approximately 9×10^{-6}. Low confidence classifications are considered incorrect, but safe. To test in the forward direction (querying on the deformed shirt), we label query points while collecting images. In the inverse direction (querying from the canonical), we query collar, shoulder, sleeve, and bottom points and visualize high confidence matches across

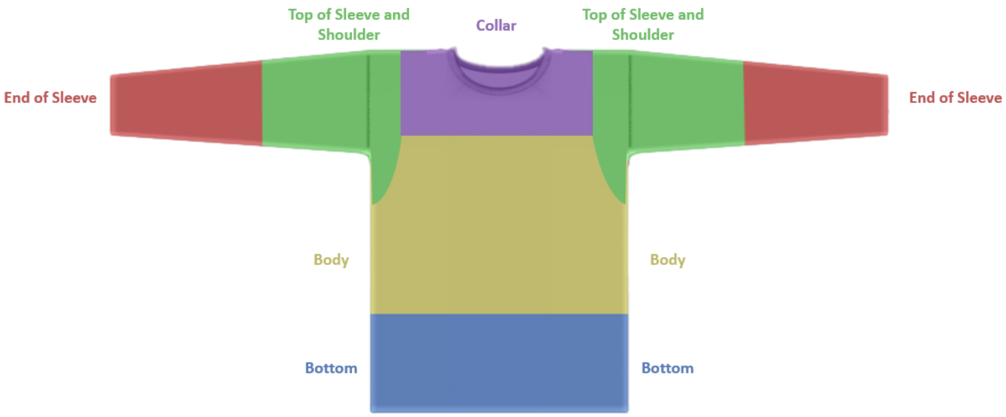

Figure 13: Shirt region classification used for real-world evaluation. During real-world evaluation of the dense correspondence network, a predicted grasp is considered correct if it falls within the same region as the predefined, ground-truth label.

all images in the dataset. Points that can be verified or rejected by a human are included in evaluation. Note that not every point is visible in the inverse queries, making low-confidence the ideal option.

We evaluate the accuracy of our dense correspondence network—trained on the combined hanging in-air and table configurations—when picking from the table by determining whether the high-confidence first grasp point the system chooses is within the appropriate region, as defined in Figure 13. We conduct 20 trials to evaluate the network's correspondence prediction success. The configurations of the shirt when picked from the table demonstrate a similar, if not more difficult, deformation as in [1] and [28]. Our method shows a comparable success rate to prior works, with the added capability of choosing grasp points from a highly deformed shirt hanging in air.

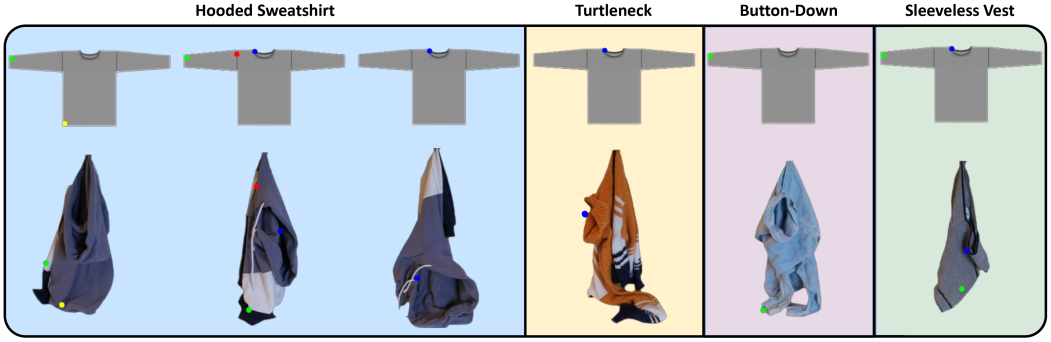

Figure 14: Examples of out-of-distribution shirts tested. We assess the zero-shot out-of-distribution generalization capabilities of our network by testing its predictions in the inverse direction on unseen shirt styles. In general, features such as hoods, turtleneck collars, and buttons not present in the simulated training dataset do not degrade the network's performance, as it is still able to classify shirt features accurately. Some misclassifications do occur with sleeveless shirts, as the network predicts the bottom of the shirt as the end of the sleeve. Overall, the network successfully generalizes to previously unseen shirt styles, demonstrating a visual understanding of the shirt structure.

The dataset simulated in Blender offers much flexibility in rendering a wide range of shirt geometries and details, including variations in body and sleeve length and shirt details. However, features such as hoods, turtleneck collars, buttons, and sleeveless shirts are not simulated. We assess our dense object network's zero-shot generalization capabilities to out-of-distribution shirts in the inverse direction. Notably, previously unseen visual features such as hoods, turtlenecks, and button-up collars do not seem to degrade the network's ability to distinguish the collar regions from the sleeves or bottoms of the shirts. Similarly, color-blocked patterns and buttons do not confuse the network, likely due to the wide range of textures and colors present in the simulated training data. Occasional misclassifications occur with sleeveless shirts and vests, where the network incorrectly predicts the shirt bottom as a sleeve when queried from the canonical shirt. We note, however, this error is also observed in some in-distribution examples. Overall, despite the unseen shirt types, our network demonstrates a general visual understanding of the shirt structure and effectively generalizes to styles beyond those seen in training. See Figure 14 for examples.

7.5 Visuotactile Grasp Affordance

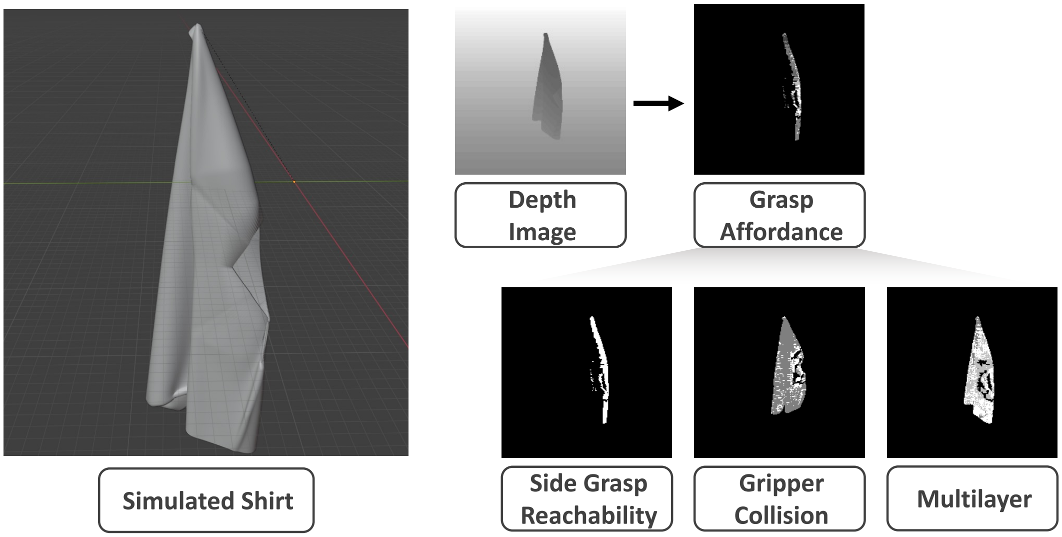

Figure 15: Visuotactile grasp affordance training in simulation. We generate affordance labels for entire images in simulation by evaluating grasp feasibility based on reachability with a side grasp, collision avoidance, and fabric layer count (restricted to two or fewer). We adapt the affordance data generation pipeline introduced in [19] to our simulation environment to obtain the affordance labels.

We compute per-pixel grasp affordance labels in simulation using an adapted version of the method from [19]. In our case, the goal is to identify viable side grasps for grasping shirts rather than edge grasps for towels, so we modify the criteria accordingly. Specifically, we remove the edge constraint used in the original formulation and allow up to two fabric layers instead of one. Affordance labels are computed by evaluating whether a candidate grasp point (1) is reachable by the right arm, (2) avoids collision with the cloth during the approach, and (3) results in no more than two layers of fabric between the gripper fingers. Figure 15 shows examples of the resulting simulation affordance labels. The network took under 2 hours to train on the simulated network on a Titan X Pascal GPU.

Collecting 8000 grasps on the robot supervised with our tactile classifier took approximately 14 hours. The tactile classifier cannot reliably determine whether the grasped region corresponds to the intended visual target. As a result, non-reachable pixels can yield positive tactile signals due to inadvertently grasping cloth in front of the target. To help address these challenges, we incorporate several training strategies:

- **Neighboring Pixel Loss with Gaussian Weighting:** To enhance robustness to small positional shifts, we include a neighborhood of pixels around the ground-truth grasp point. Each neighboring pixel's loss is weighted by a Gaussian function of its distance from the center:

$$\mathcal{L}_{\text{neighbor}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right) (A(i,j) - y_{\text{gt}})^2 \tag{2}$$

  where:
  - $\mathcal{N}$ is the set of neighboring pixels within a $d_{\text{px}} \times d_{\text{px}}$ window centered on the ground-truth grasp point, excluding any pixels outside image bounds.

- $d_{ij}^2 = (i - i_0)^2 + (j - j_0)^2$ is the squared Euclidean distance from the center pixel (i_0, j_0).
- σ controls the spread of the Gaussian weighting.
- $|\mathcal{N}|$ is the number of valid neighboring pixels for normalization.

- **Spatial Regularization:** Encourages smoothness in the affordance map by penalizing large gradients between adjacent pixels:

$$\mathcal{L}_{\text{spatial}} = \sum_{i,j} |A(i+1,j) - A(i,j)| + |A(i,j+1) - A(i,j)| \tag{3}$$

- **Simulation Regularization:** Ensures consistency between the fine-tuned real-world affordance network and the pretrained simulation network to maintain global structure even with unexplored points:

$$\mathcal{L}_{\text{sim}} = \|A_{\text{real}} - A_{\text{sim}}\|^2 \tag{4}$$

- **Weight Decay:** Applies L2 regularization to the network weights directly in the optimizer:

$$\mathcal{L}_{\text{weight}} = \lambda_{\text{weight}} \|\theta\|^2 \tag{5}$$

The total loss is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{neighbor}} + \lambda_{\text{spatial}}\mathcal{L}_{\text{spatial}} + \lambda_{\text{sim}}\mathcal{L}_{\text{sim}} \tag{6}$$

Figure 16 compares affordance predictions from networks trained in simulation and on real robot grasps.

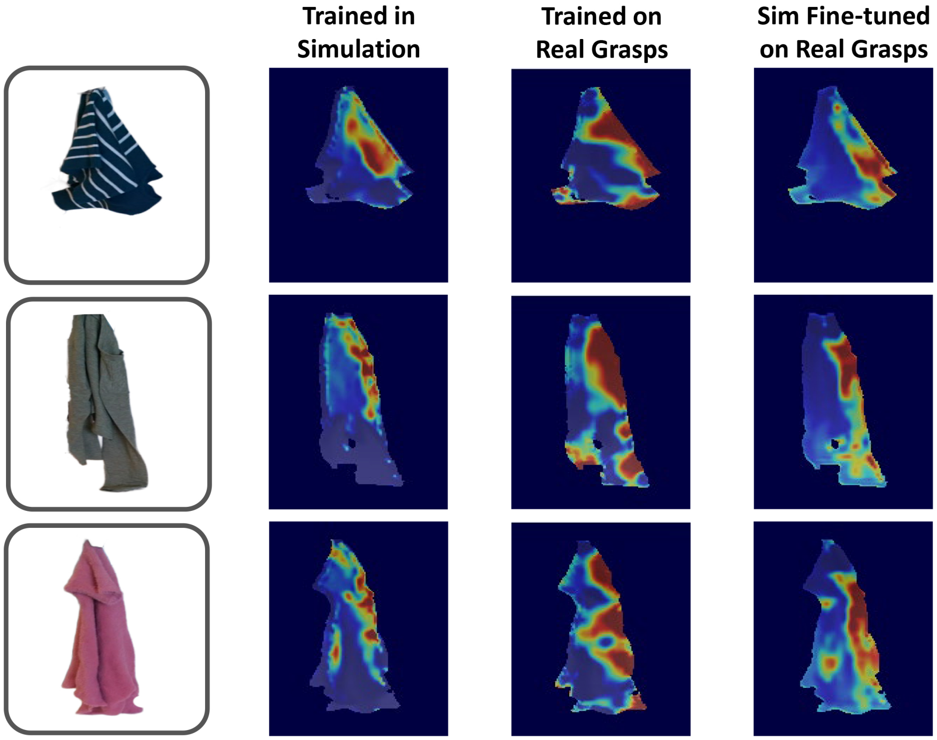

Figure 16: Fine-tuned visuotactile grasp affordance compared to baselines. The model trained in simulation (left, Sim2Real) is overly conservative, often failing to identify viable grasp points—particularly near the bottom of the shirt. In contrast, the model trained only on real robot grasps (middle, Real2Real) is overconfident in unexplored regions and is sensitive to misclassified grasps where the robot contacts fabric inside the shirt, rather than the intended target region, without regularization from the network trained in simulation.

7.6 Human Video Demonstrations

In order to extract grasp points from human video demonstrations, we trained a custom gesture recognizer based on MediaPipe's GestureRecognizer framework. This network allows us to track transitions between open and grasping hands and tracks the hand skeleton. We identify grasp events as frames in which both hands are in a grasping pose, and extract the first frame of these segments as key frames. The index fingertip of the lower hand is then used as a query point for our dense correspondence model to localize the intended grasp location on a canonical garment image (Figure 6). We apply a Segment Anything-based mask [38] to isolate the garment in the demonstration image.

While the full pipeline enables generalization across different users and environments, its success rate is currently limited. The gesture recognizer can misclassify ambiguous hand poses and the off-the-shelf skeleton tracker occasionally fails to accurately localize the hands. Additionally, the dense correspondence model struggles in frames where the hand occludes the target grasp point. To mitigate occlusion, we select a frame a few steps prior to the grasp, but in many cases, the cloth shifts between these frames, leading to inaccurate grasp localization. This pipeline is outside of the primary focus of our work, but rather a demonstration of the potential for using dense descriptors to interface with unconstrained human video data. With more focused development, these limitations could likely be addressed—for example, by training a more robust, domain-specific gesture recognizer or incorporating occlusion-aware correspondence networks. Despite its current limitations, this approach illustrates how our descriptor representation enables pick point extraction directly from raw demonstrations—a key step toward scaling data collection for garment manipulation.