Inverse Garment Design with Visual Agents

Nanxiang Jiang







Inverse Garment Design



2D Image

Inverse Graphics

Computer Graphics



3D Model

Simulation

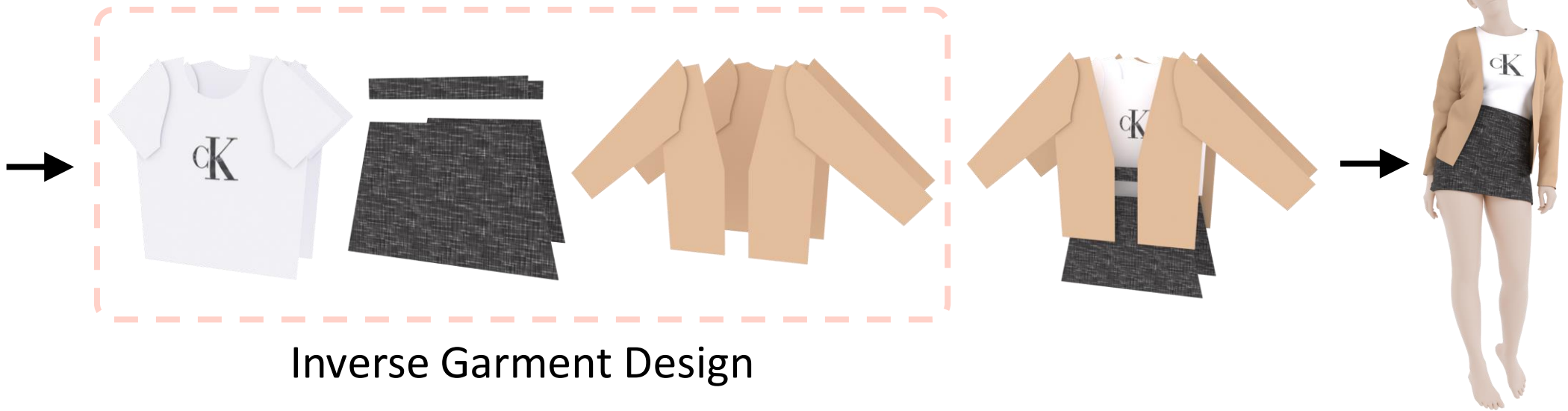


3D Garment Representation

- Simulation Ready
- High Fidelity Texture
- Ready to Manufacture

Sewing Pattern

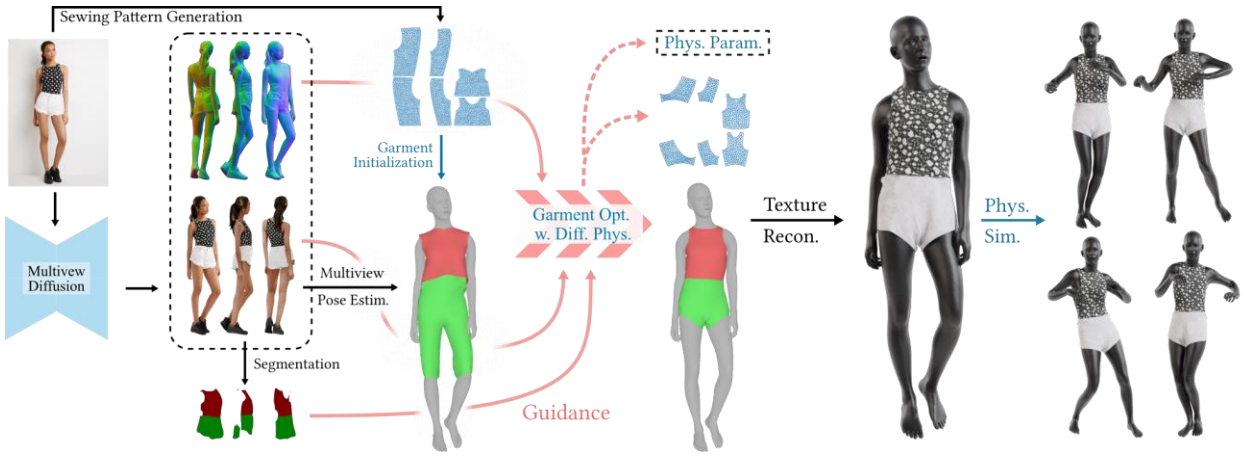




Inverse Garment Design

Related Work on Sewing Pattern Inverse Design

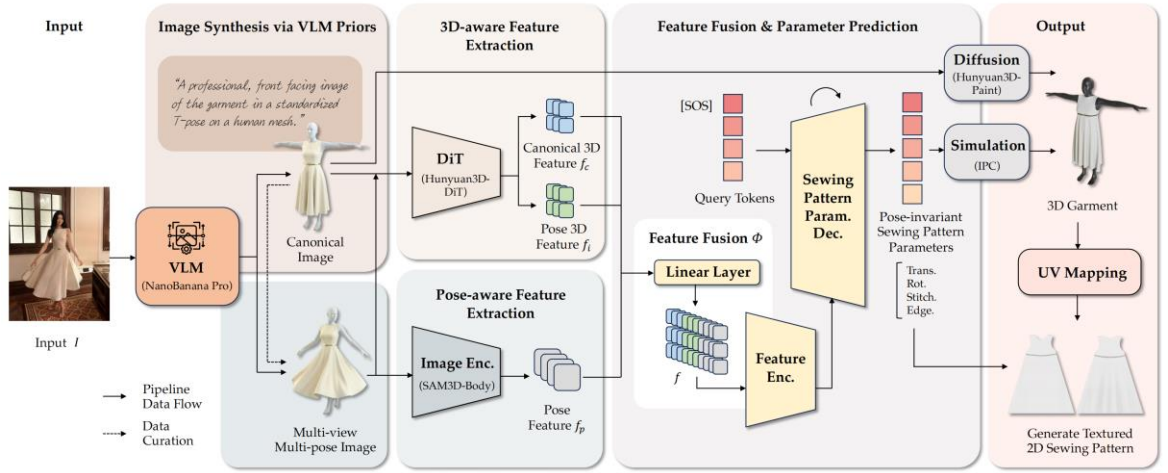
- **Differentiable Optimization**



Dress-1-to-3, Li et al., SIGGRAPH 2025

- Pros: Accurate, Generalizable
- Cons: Time / Resource Consuming, Labor Intensive, Local Minimum

- **Feed-Forward Method**



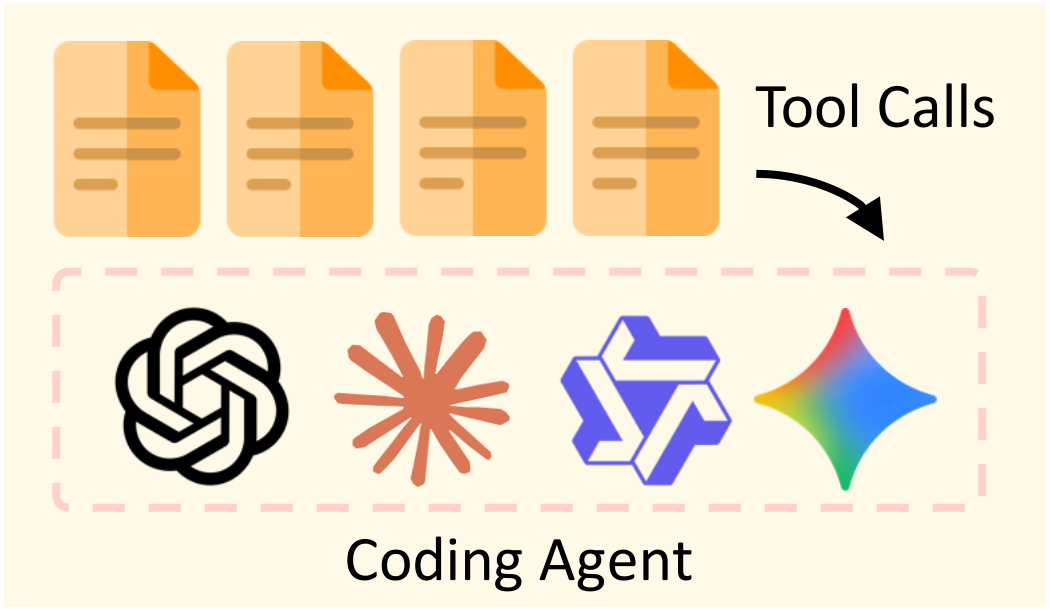
DressWild, Tao et al., NeurIPS 2026

- Pros: Fast, Unified
- Cons: Limited to Data Distribution, Lower Accuracy and Generalization

Agentic Inverse Garment Design

Q: Can we design a method that both approaches the **quality and generalization** of differentiable optimization, and the **speed** of feed-forward methods?









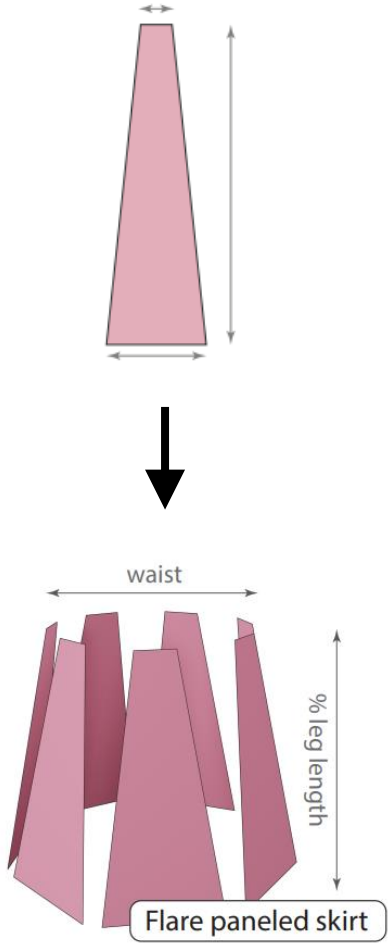
Closed Loop Refinement

DressAgent: Hierarchical Agentic System

- Garment Language
- Stage 1: Layer Planning and Coarse Geometry Estimation
- Stage 2: Fine-Grained Geometry Refinement
- Post Processing (Texture and Simulation)

GarmentCode: The Garment Language






```python
class ThinSkirtPanel(pyp.Panel):
    """One panel of a panel skirt"""

    def __init__(self, name, top_width=10, bottom_width=20, length=70) -> None:
        super().__init__(name)

        # define edge loop
        self.flare = (bottom_width - top_width) / 2
        self.edges = pyp.esf.from_verts(
            [0,0], [self.flare, length], [self.flare + top_width, length], [self.flare * 2 + top_width, 0],
            loop=True)

        # w.r.t. top left point
        self.set_pivot(self.edges[0].end)

        self.interfaces = {
            'right': pyp.Interface(self, self.edges[0]),
            'top': pyp.Interface(self, self.edges[1]),
            'left': pyp.Interface(self, self.edges[2])
        }
```

```python
class SkirtManyPanels(pyp.Component):
    """Round Skirt with many panels"""

    def __init__(self, body, design) -> None:
        super().__init__(f'{self.__class__.__name__}_{design["flare-skirt"]["n_panels"]["v"]}')

        waist = body['waist']    # Fit to waist

        design = design['flare-skirt']
        n_panels = design['n_panels']['v']

        # Length is dependent on length of legs
        length = body['hips_line'] + design['length']['v'] * body['leg_length']

        flare_coeff_pi = 1 + design['suns']['v'] * length * 2 * np.pi / waist

        self.front = ThinSkirtPanel('front', panel_w:=waist / n_panels,
                                    bottom_width=panel_w * flare_coeff_pi,
                                    length=length )
        self.front.translate_to([-waist / 4, body['waist_level'], 0])
        # Align with a body
        self.front.rotate_by(R.from_euler('XYZ', [0, -90, 0], degrees=True))
        self.front.rotate_align([-waist / 4, 0, panel_w / 2])

        # Create new panels
        self.subs = pyp.ops.distribute_Y(self.front, n_panels, odd_copy_shift=15)

        # Stitch new components
        for i in range(1, n_panels):
            self.stitching_rules.append((self.subs[i - 1].interfaces['left'], self.subs[i].interfaces['right']))

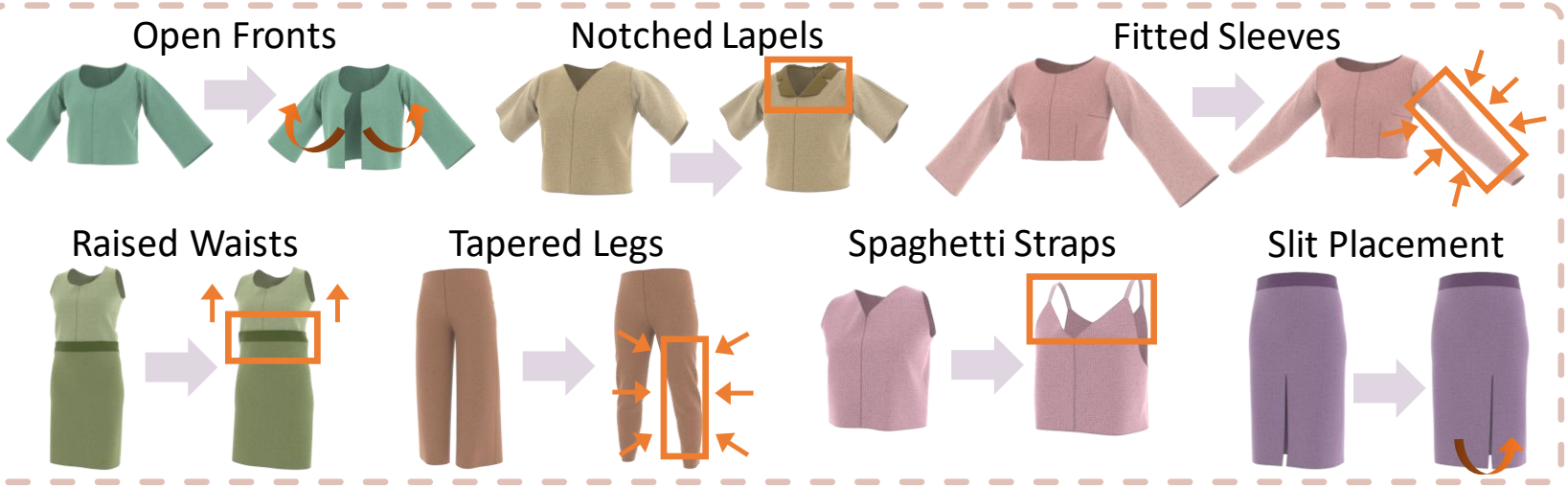
        self.stitching_rules.append((self.subs[-1].interfaces['left'], self.subs[0].interfaces['right']))

        # Define the interface
        self.interfaces = {
            'top': pyp.Interface.from_multiple(*[sub.interfaces['top'] for sub in self.subs])
        }
```

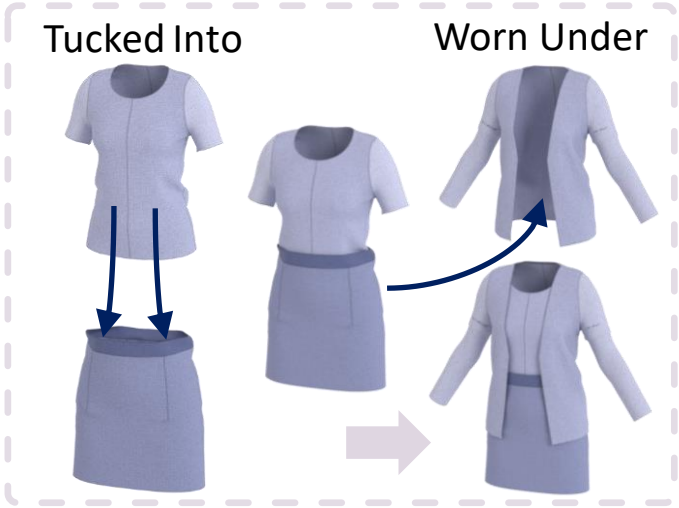
GarmentCode, Korosteleva et al., SIGGRAPH Asia 2023

GarmentCode-XL: Extended and Layered Language

Enhanced Expressivity



Layered Representation





1 Layer → 4 Layers

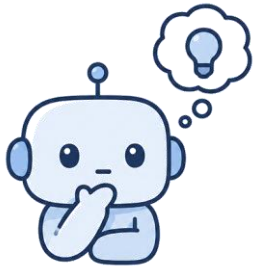
GarmentCodeData-XL 15K

Stage 1: Layer Planning and Coarse Geometry Estimation





Canonical A-pose

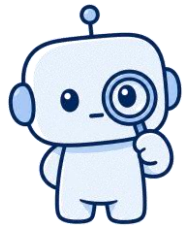


Layer Planner

*"The garment contains two layers: Cardigan over tee and skirt. Now I'll call the **image editing tools** to peel the beige cardigan ..."*



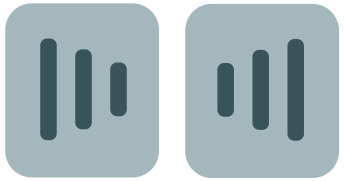
Layer 1 Layer 2



Visual Reader

GarmentCode-XL Syntax

Distill



Encoder-Decoder



Coarse Pattern

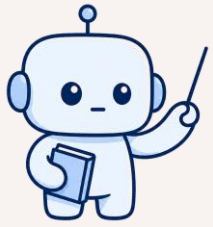
Stage 2: Fine-Grained Geometry Refinement





Before

Agentic Trio



Orchestrator



Critic



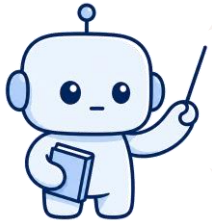
Tailor



Memory



After



Orchestrator

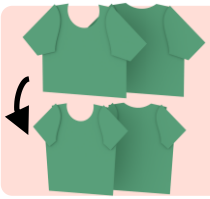
First, I will Call `Critic()` *to judge the current step.*

Last step (Iter 1) is worth keeping. Now I'll call `Tailor()` *to slim the tee, raise neckline and shorten sleeves.*



This step is better, but the tee is too wide, too deep at the neck, too long in the sleeves.

`Measure(step)` `Scoring(step)`

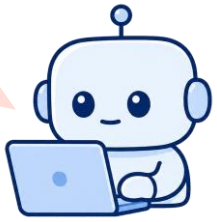


I have to modify several parameters.

`Fit(parameters)` `Simulate(step)`



Critic



Tailor

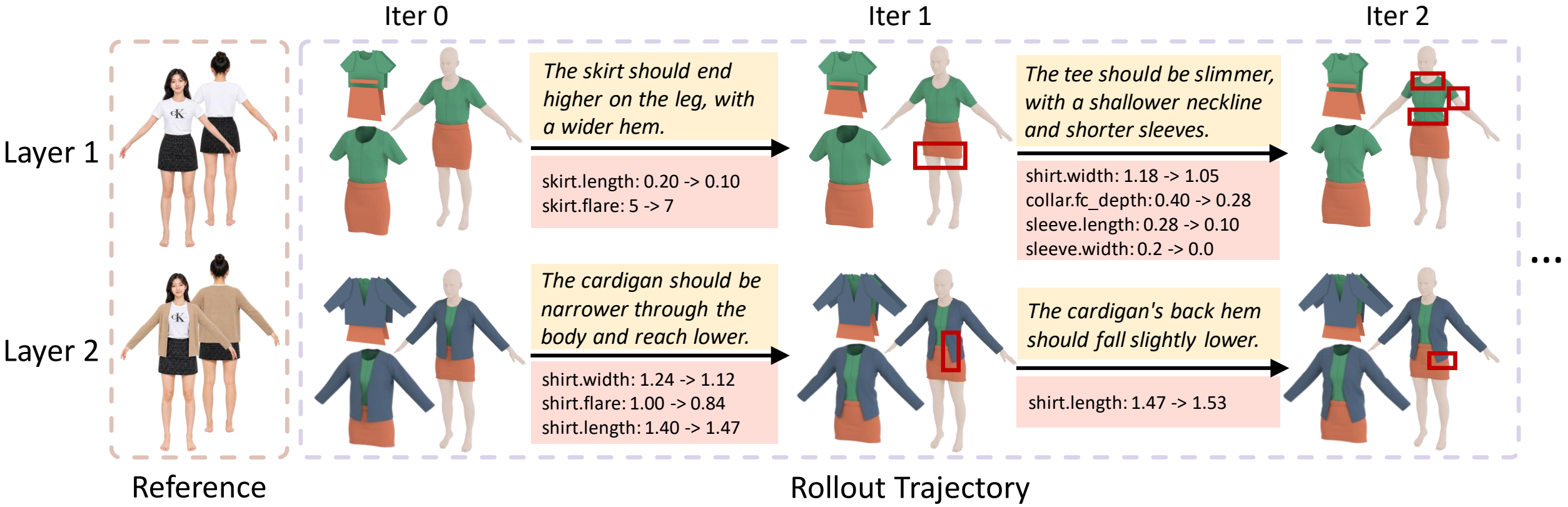


Before



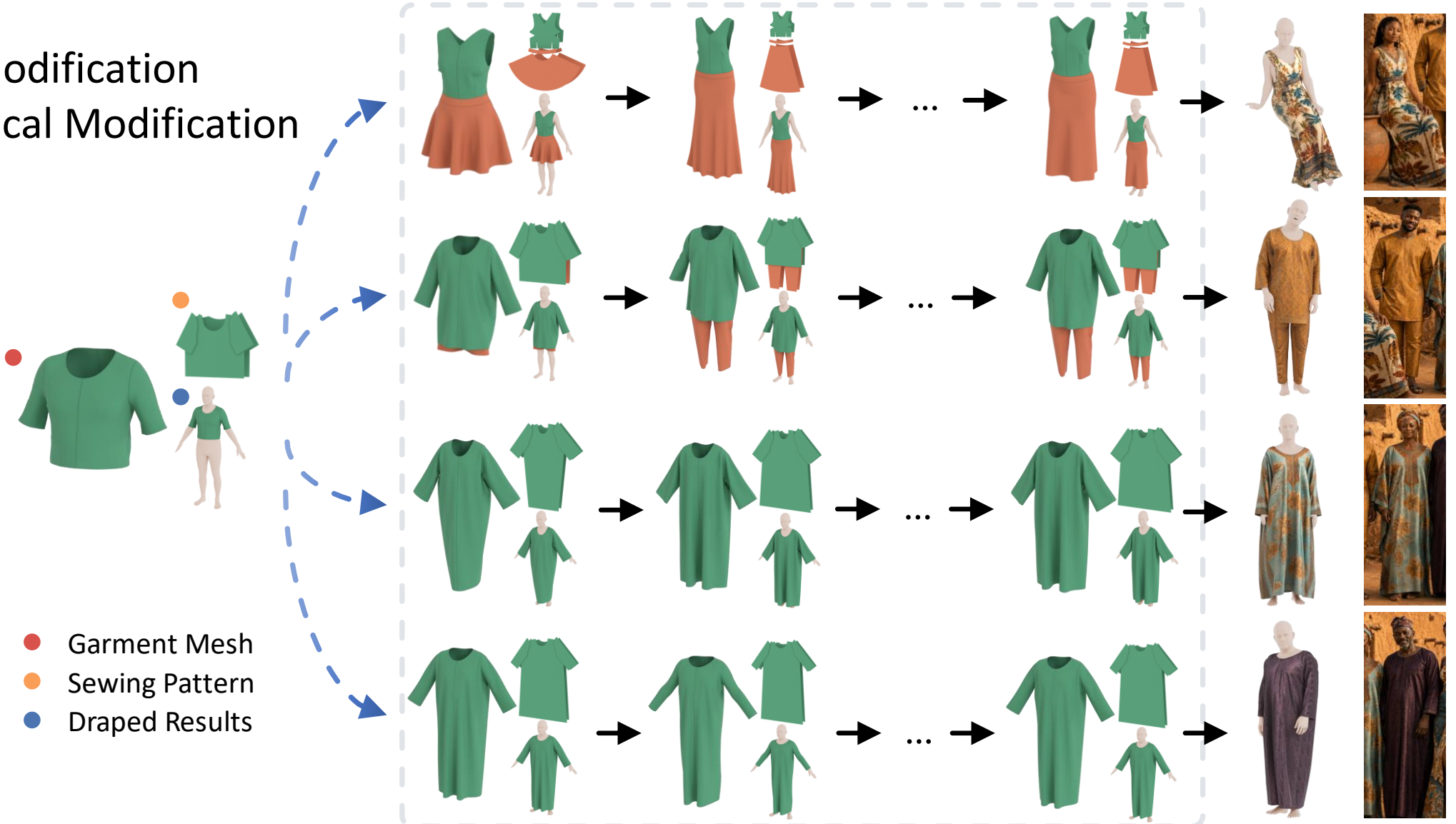
After

Stage 2: Fine-Grained Geometry Refinement



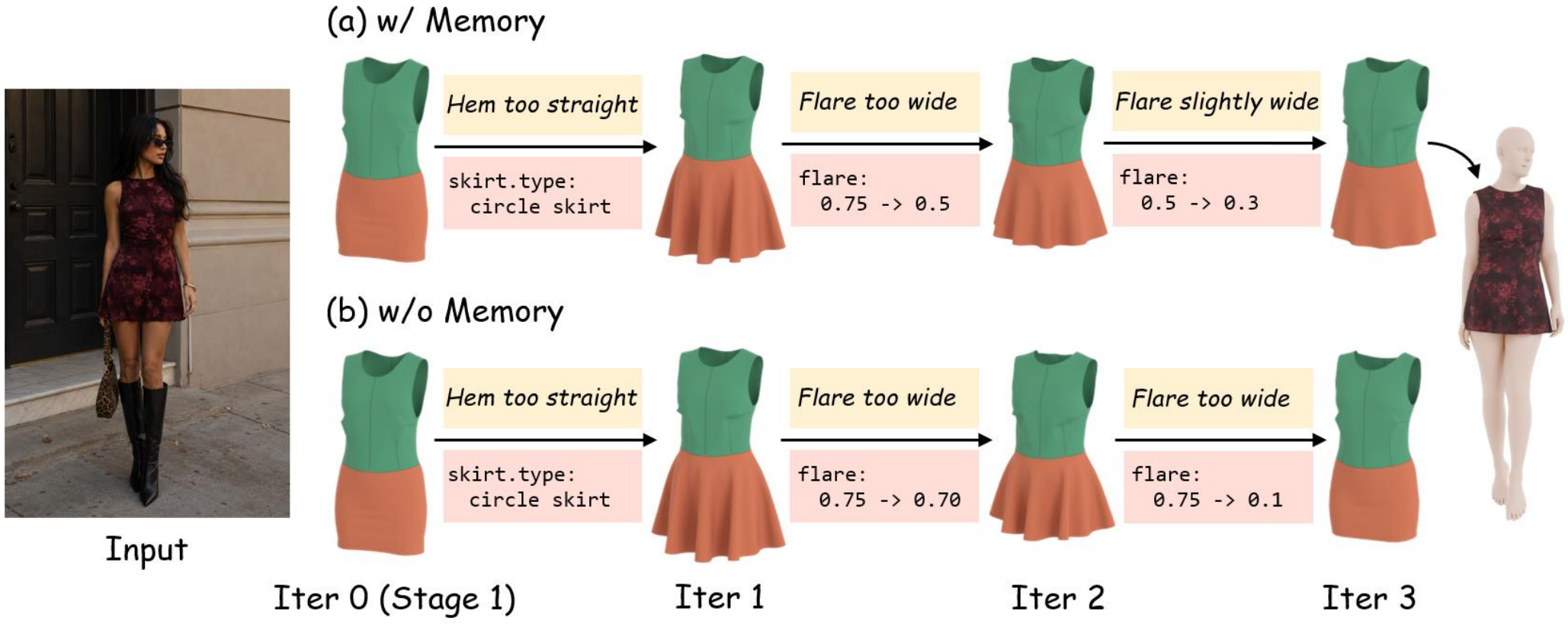
Stage 2: Fine-Grained Geometry Refinement

- Style Modification
- Numerical Modification



Stage 2: Fine-Grained Geometry Refinement

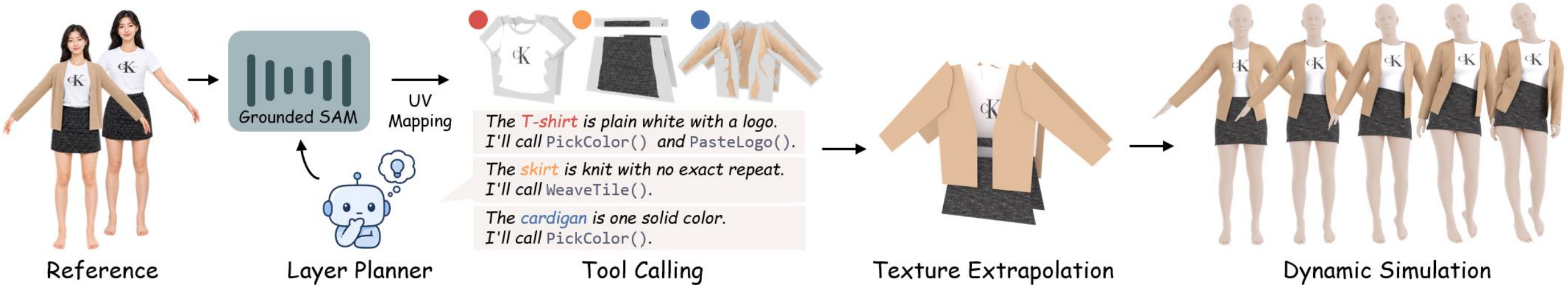
- Evolving Memory for Convergence



Post Processing: Texture Generation

- Five Tools for Texture Generation



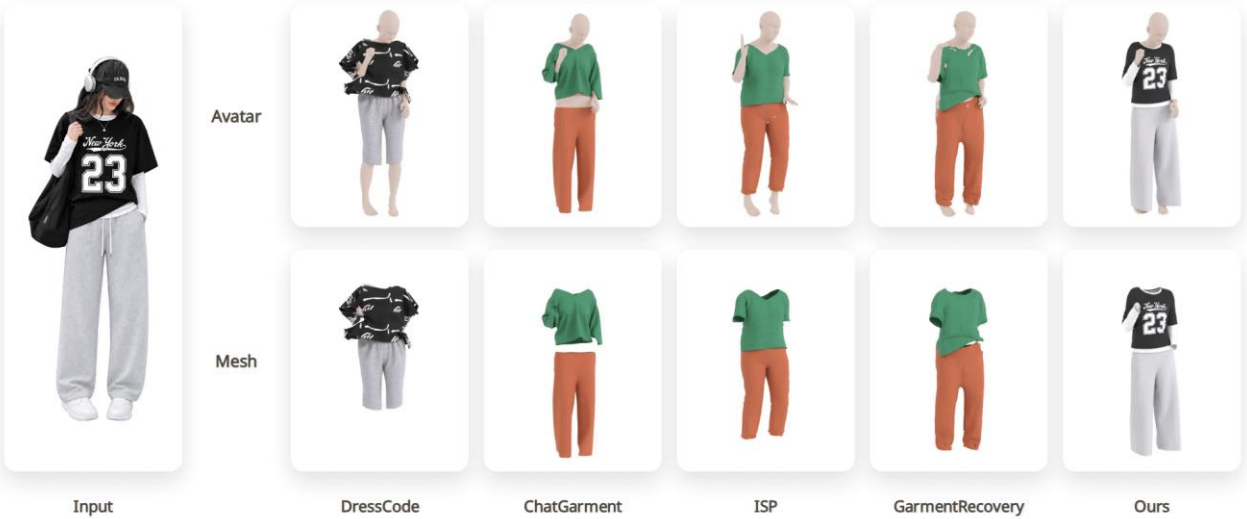


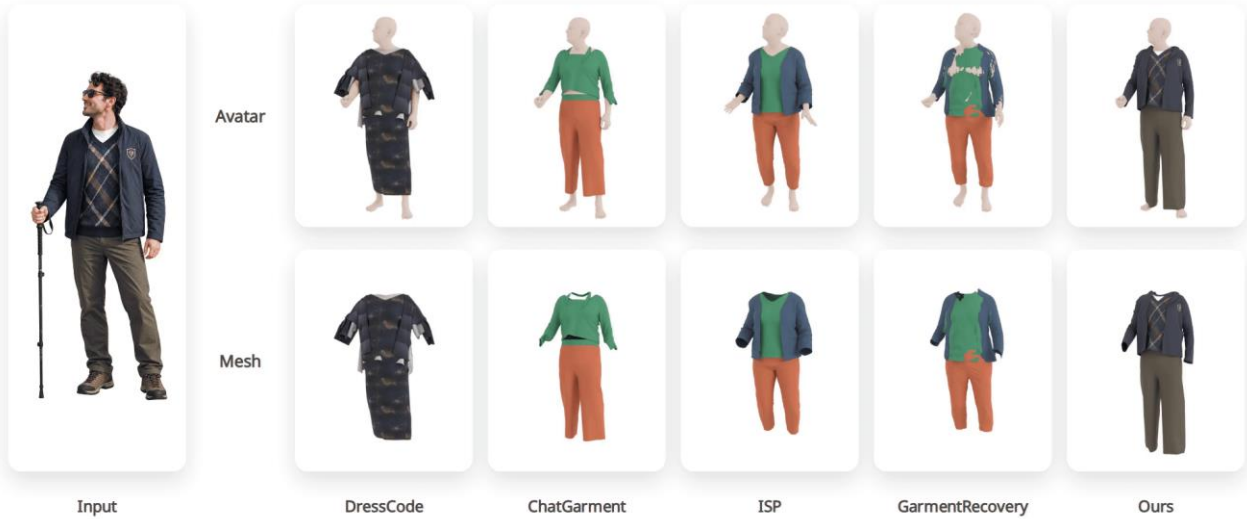
Experiments

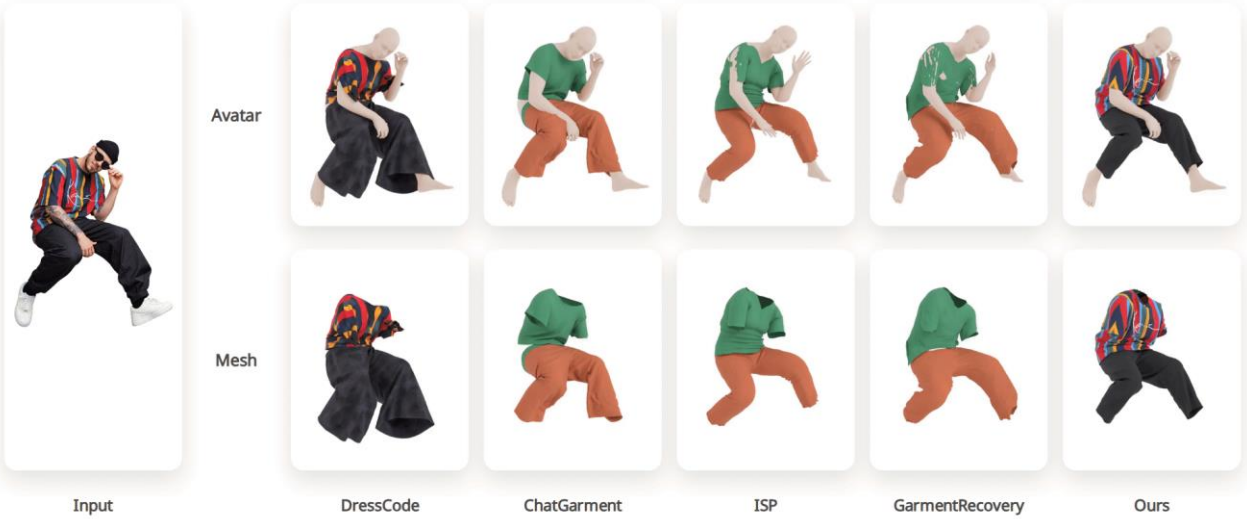


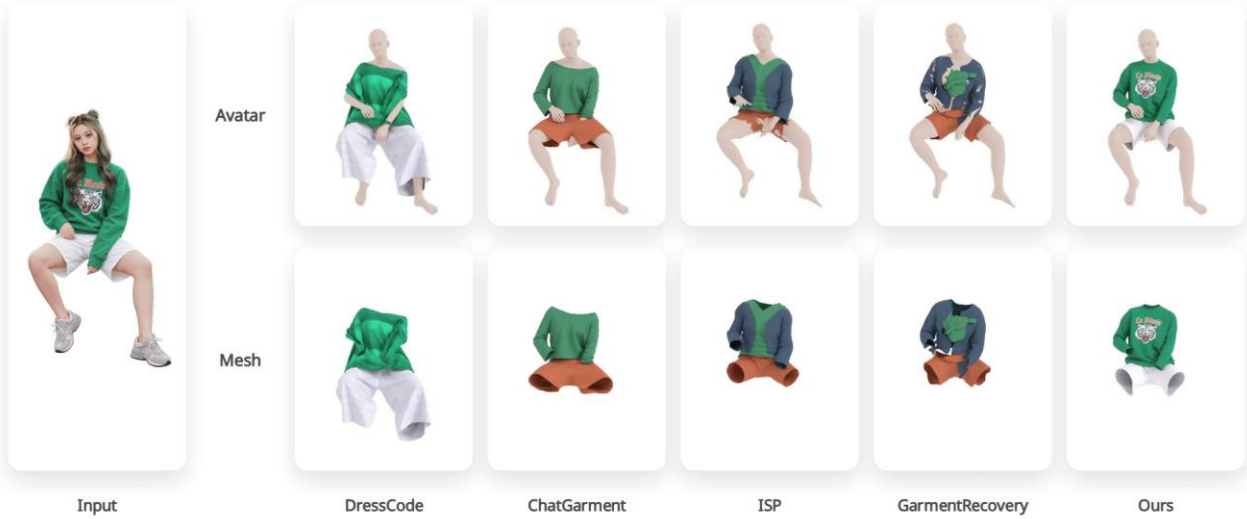
For Interactive Results:
https://nxjiang-jnx.github.io/dress-agent
Or Google Search: ***Inverse Garment Design with Visual Agents***

Experiments: Comparison with Baselines







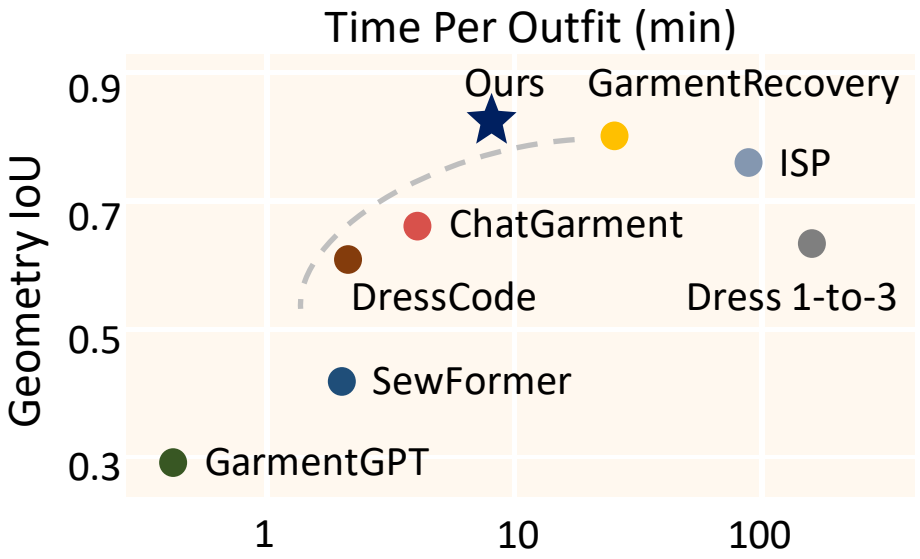


Experiments: Pareto Frontier

- New Pareto Between Accuracy and Speed

Table 1: **Quantitative comparison.** We report the structure accuracy of sewing patterns and the geometry accuracy of draped garments. ISP and GarmentRecovery are excluded from structure accuracy, as they do not produce explicit sewing patterns with panel geometry. Our method achieves the best quality on most metrics, and is significantly faster than differentiable optimization.

Method	Structure Acc. ↑			Geometry (CloSe)		Geometry (4D-Dress)		Avg. Time ↓ (min)
	Panel	Edge	Stitch	CD ↓	IoU ↑	CD ↓	IoU ↑	
SewFormer	18.67	28.80	7.24	7.40	0.348	7.58	0.496	2.0
ChatGarment	51.84	57.15	70.18	2.39	0.580	1.94	0.763	4.2
GarmentGPT	43.62	38.54	65.43	5.50	0.381	10.46	0.204	0.4
DressCode	19.67	26.47	30.97	2.77	0.542	2.56	0.685	2.2
ISP	-	-	-	1.79	0.687	**1.16**	0.859	90.0
GarmentRecovery	-	-	-	1.77	0.740	1.51	0.865	25.0
Dress 1-to-3	17.67	31.07	26.14	1.63	0.652	2.05	0.617	176.0
Ours	**87.36**	**76.56**	**88.90**	**1.20**	**0.764**	1.21	**0.880**	8.6



- User Study










Failure Case and Future Work

- Introduce Free-Form Representations to Reduce Inductive Bias
- Let Coding Agents Freely Extend Garment Language
- Introduce Fast-GS / Particle Based Methods to Accelerate Simulation



Input



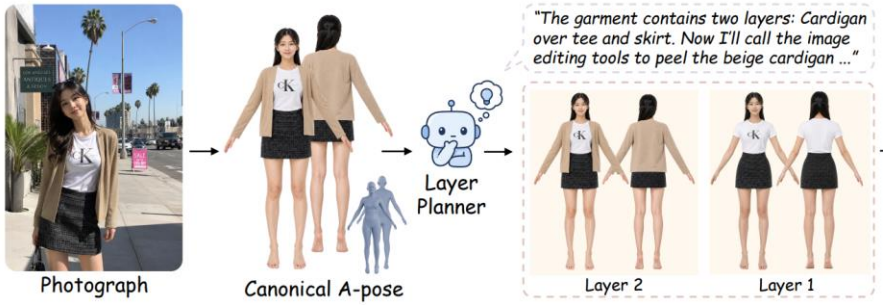
DressAgent

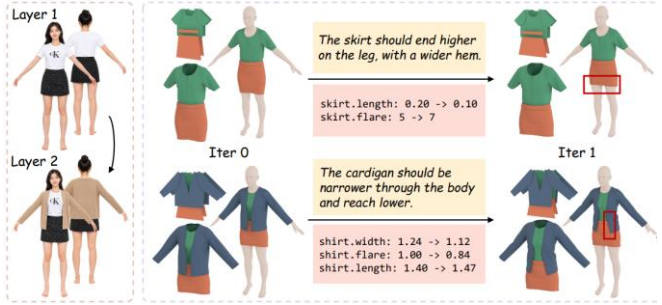
Conclusion: DressAgent System

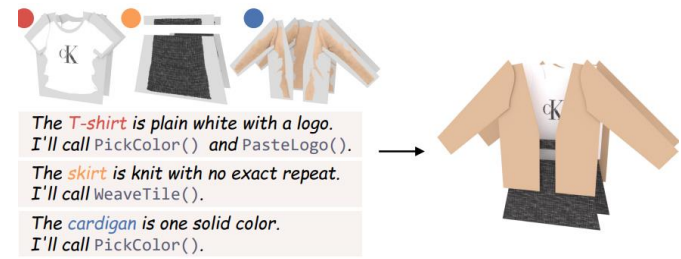
A couturier must be an architect for design, a sculptor for shape, a painter for color, a musician for harmony, and a philosopher for temperance.

— *Cristóbal Balenciaga*

- Hierarchical Agentic System (Planning + Refinement + Memory + Tools + Texture)





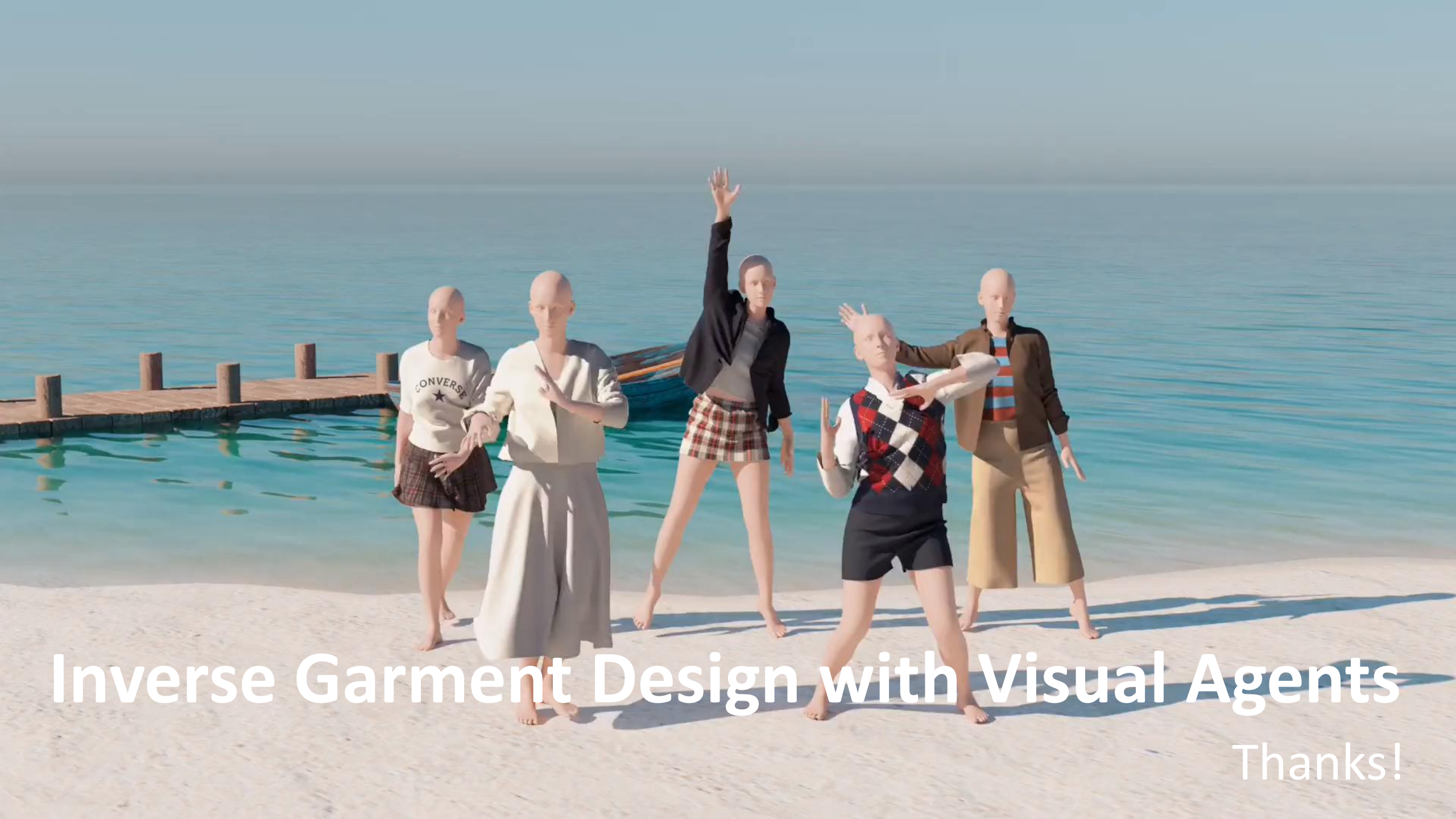


- Infrastructure: GarmentCode-XL, GarmentCodeData-XL (15K), Garment Simulation





- Experiments: Accuracy, Zero-Shot Generalization and Efficiency



Inverse Garment Design with Visual Agents

Thanks!