AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars

Paper Review

AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars

FANGZHOU HONG*, S-Lab, Nanyang Technological University, Singapore
MINGYUAN ZHANG*, S-Lab, Nanyang Technological University, Singapore
LIANG PAN, S-Lab, Nanyang Technological University, Singapore
ZHONGANG CAI, S-Lab, Nanyang Technological University, Singapore and SenseTime Research, China
LEI YANG, SenseTime Research, China
ZIWEI LIU†, S-Lab, Nanyang Technological University, Singapore

Introduction

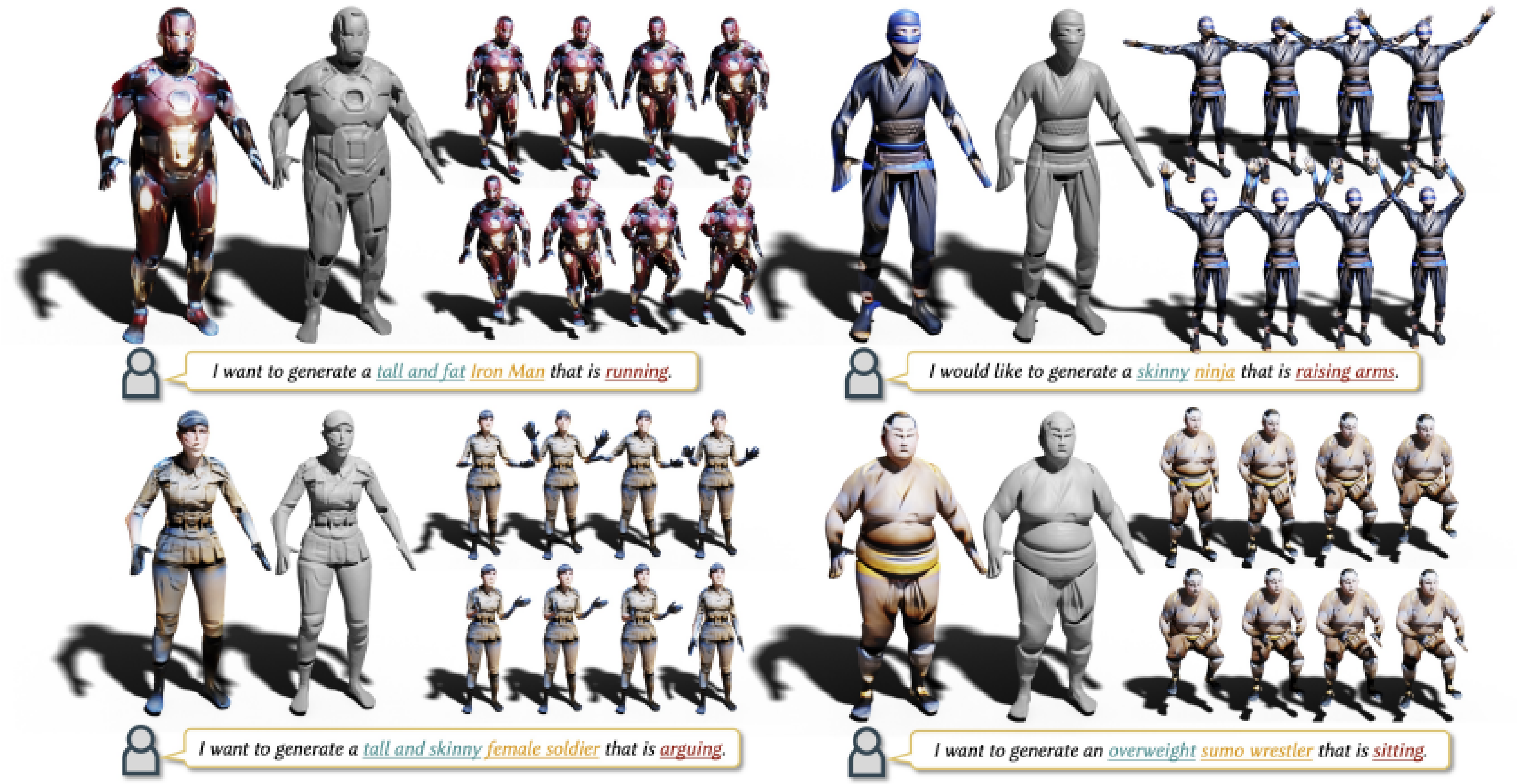

I want to generate a tall and fat Iron Man that is running.
I would like to generate a skinny ninja that is raising arms.
I want to generate a tall and skinny female soldier that is arguing.
I want to generate an overweight sumo wrestler that is sitting.

Fig. 1. In this work, we present AvatarCLIP, a novel zero-shot text-driven 3D avatar generation and animation pipeline. Driven by natural language descriptions of the desired shape, appearance and motion of the avatar, AvatarCLIP is capable of robustly generating 3D avatar models with vivid texture, high-quality geometry and reasonable motions.

3D 아바타의 중요성과 문제점

- 전통적인 3D 아바타 생성 및 애니메이션의 복잡성
- 전문 지식, 많은 인력, 고가의 장비가 필요
- 기존 방법의 유연성 부족, 생성 품질 불안정, 대중화의 어려움

AvatarCLIP

- 전문 지식 없이, 자연어 기반 3D 아바타 생성
- 형태, 텍스처 및 애니메이션의 맞춤형 생성 지원
- text = {t_shape, t_app, t_motion}

Preliminaries

CLIP

- 이미지 인코더 E$_I$, 텍스트 인코더 E$_T$
- 이미지와 텍스트 쌍은 가까워지고, 비쌍은 멀어지는 방식으로 학습
- 손실 함수 $L_{clip}(I,T) = 1 - \mathrm{norm}(E_I(I)) \cdot \mathrm{norm}(E_T(T))$

SMPL

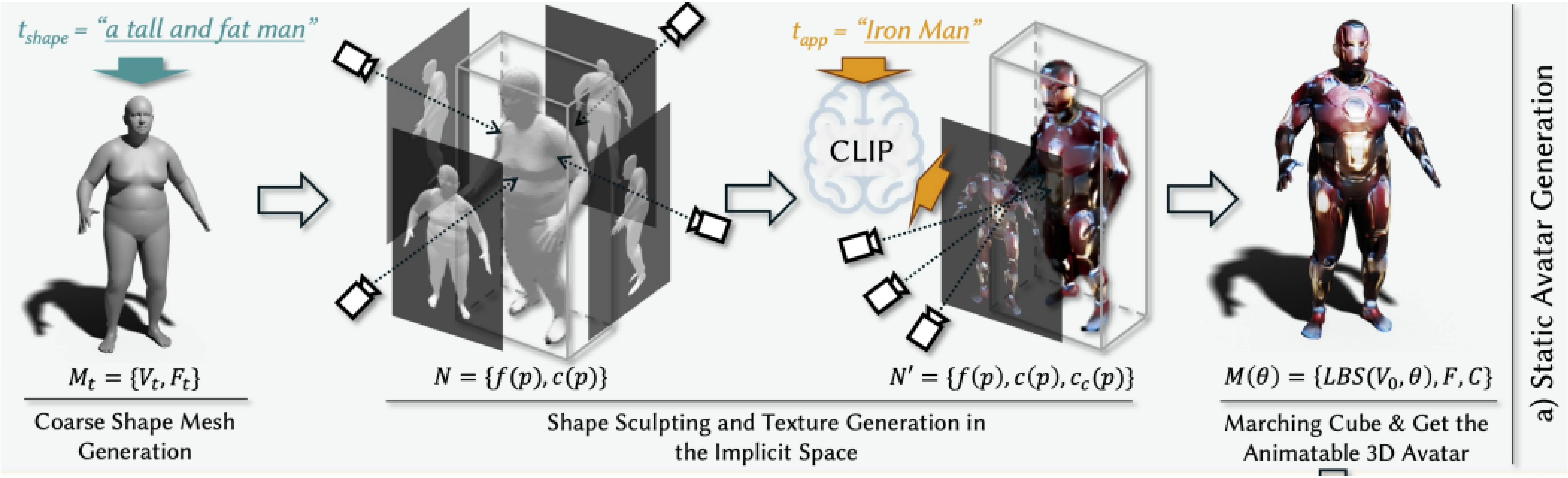
- 애니메이션 템플릿로 사용
- M $_{\text{SMPL}}$ (β,θ;Φ)

NeuS

- 서명 거리 함수(SDF)와 신경 방사장(NeRF)을 결합한 모델
- 렌더링을 지원하여 현실적인 3D 표현을 가능

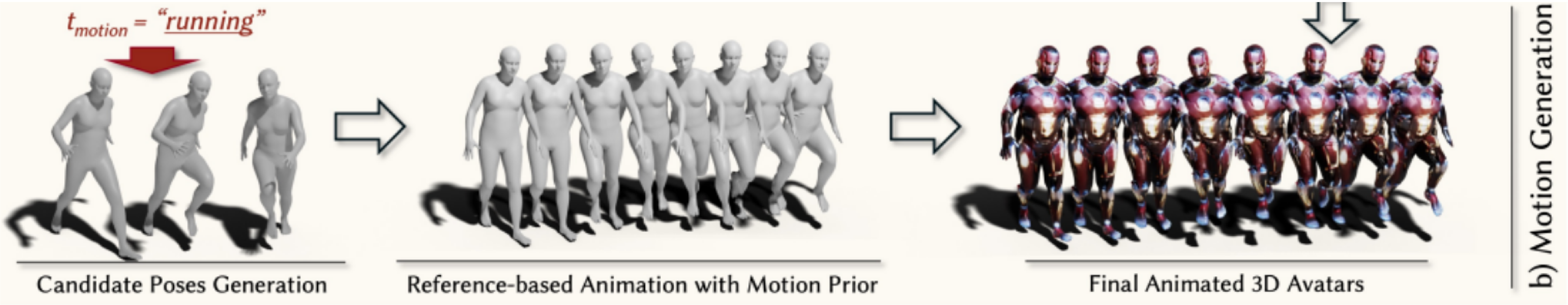
$$C(o,v) = \int_0^\infty w(t)c(p(t),v)dt \qquad w(t) = \frac{\phi_s(f(p(t)))}{\int_0^\infty \phi_s(f(p(u)))du}$$

Static Avatar Generation


t_{shape} = "a tall and fat man"
$M_t = \{V_t, F_t\}$
Coarse Shape Mesh Generation
$N = \{f(p), c(p)\}$
t_{app} = "Iron Man"
CLIP
$N' = \{f(p), c(p), c_c(p)\}$
Shape Sculpting and Texture Generation in the Implicit Space
$M(\theta) = \{LBS(V_0, \theta), F, C\}$
Marching Cube & Get the Animatable 3D Avatar
a) Static Avatar Generation

- 대략적인 형태 메쉬 생성 (Coarse Shape Mesh Generation)
- 형태 조각 및 텍스처 생성 (Shape Sculpting and Texture Generation)
- 애니메이션 가능한 3D 아바타 생성 (Get Animatable 3D Avatar)

Motion Generation


t_{motion} = "running"
Candidate Poses Generation
Reference-based Animation with Motion Prior
Final Animated 3D Avatars
b) Motion Generation

- 후보 포즈 생성 (Candidate Poses Generation)
- 참조 기반 애니메이션 생성 (Reference-based Animation)
- 애니메이션 가능한 3D 아바타 생성 (Get Animatable 3D Avatar)

Coarse Shape Generation

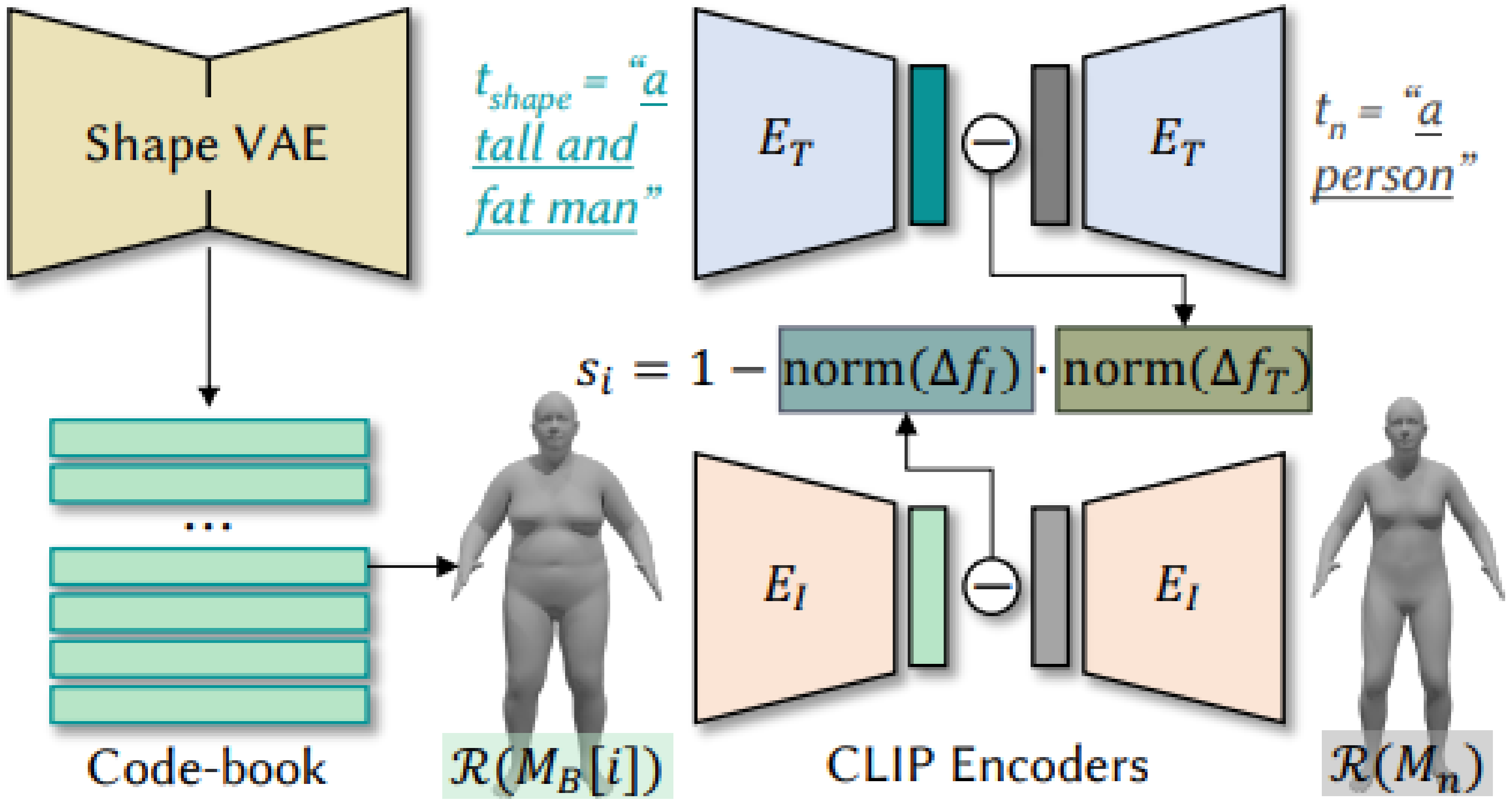


Fig. 3. **Illustration of the Coarse Shape Generation.** A shape VAE is trained to construct a code-book which is used for CLIP-guided query to get a best match for the input text t_{shape}. To introduce the awareness of body attributes like height, a neutral shape M_n and text t_n is defined as the anchor. The relative direction in latent space is used for the CLIP-guided query.

Shape VAE를 활용한 Code-book 생성

- 다양한 잠재 형태 생성
- 입력된 텍스트 설명과 관련된 형태 라이브러리 구축

CLIP을 통한 형태 생성 가이드

- 의미 기반 가이드
- 텍스트 설명과 형태 잠재 표현 간 유사성 계산

앵커 형태와 상대적 방향 최적화

- 중립 형태를 앵커로 활용
- 상대적 방향을 통해 형태 최적화

Initialization of Implicit 3D Avatar

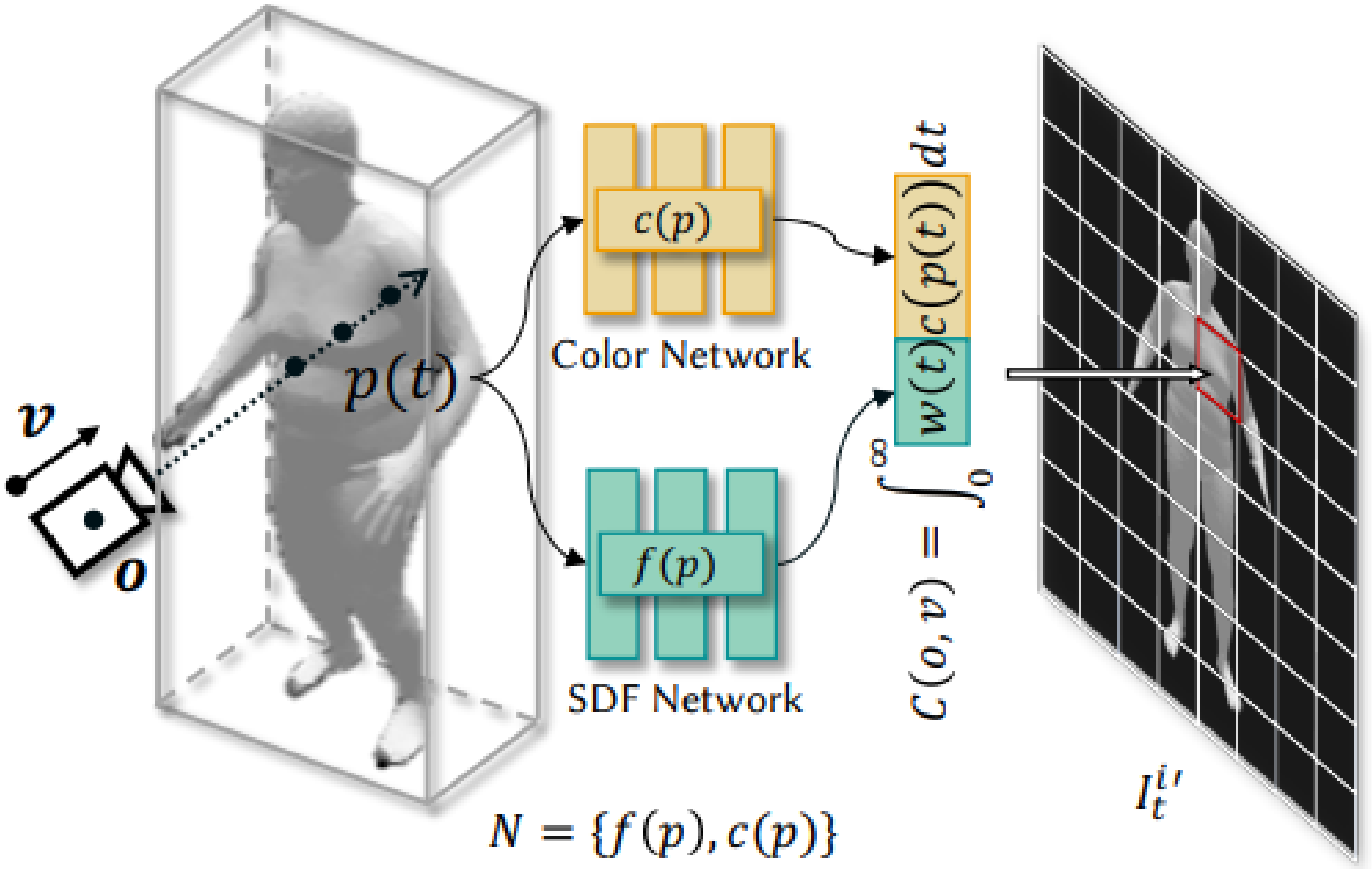


Fig. 4. **Initialization of the Implicit 3D Avatar.** Multi-view renderings of the template mesh M_t is used to optimize a randomly initialized NeuS network N, which is later used as an initialization of the implicit 3D avatar.

NeuS 네트워크의 구성

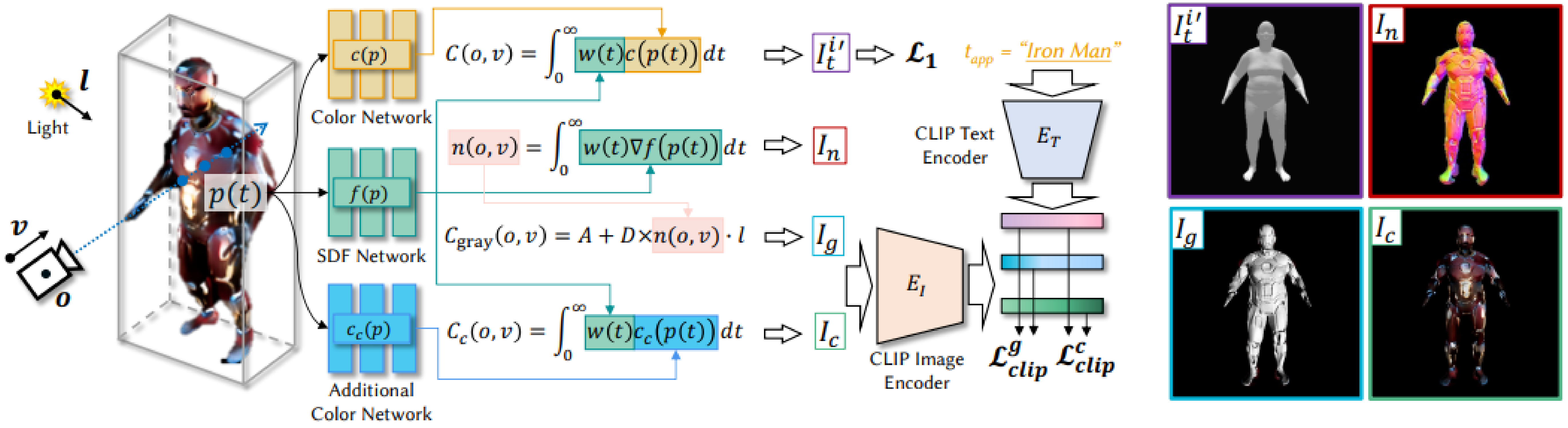
- SDF 네트워크
- 색상 네트워크

다중 시점 렌더링 및 가중치 계산

- 렌더링 적분 공식

$$C(o,v) = \int_0^\infty w(t)c(p(t),v)dt$$

Shape Sculpting and Texture Generation



a) Rendering the Implicit 3D Avatar $N' = \{f(p), c(p), c_c(p)\}$ b) Optimization c) Examples of Intermediate Results

Fig. 5. **Detailed Method of Shape Sculpting and Texture Generation.** An additional color network $c_c(p)$ is appended to the initialized implicit 3D avatar for texture generation, which is illustrated in a). Three types of constraints introduced for the optimization are shown in b), including reconstruction loss $\mathcal{L}_1$, CLIP-guided losses $\mathcal{L}^{g_{clip}}$ and $\mathcal{L}_{clip}^c$ for the geometry sculpting and texture generation, respectively. The sub-figure c) shows examples of intermediate results.

암시적 3D 아바타

- 여러 네트워크를 통해 형상과 색상을 세밀하게 조정

$$N' = \{f(p), c(p), c_c(p)\}$$

색상 값 렌더링

$$C(o, v) = \int_0^\infty w(t)c(p(t))dt$$

법선 값 계산

$$n(o, v) = \int_0^\infty w(t)\nabla f(p(t))dt$$

조명 계산

$$C_{gray}(o, v) = A + D \times n(o, v) \cdot l$$

추가 색상 네트워크

$$C_c(o, v) = \int_0^\infty w(t)c_c(p(t))dt$$

three-part loss function

$$L_2 = L_1 + \lambda_3 L_{clip}^c + \lambda_4 L_{clip}^g$$

Candidate Poses Generation

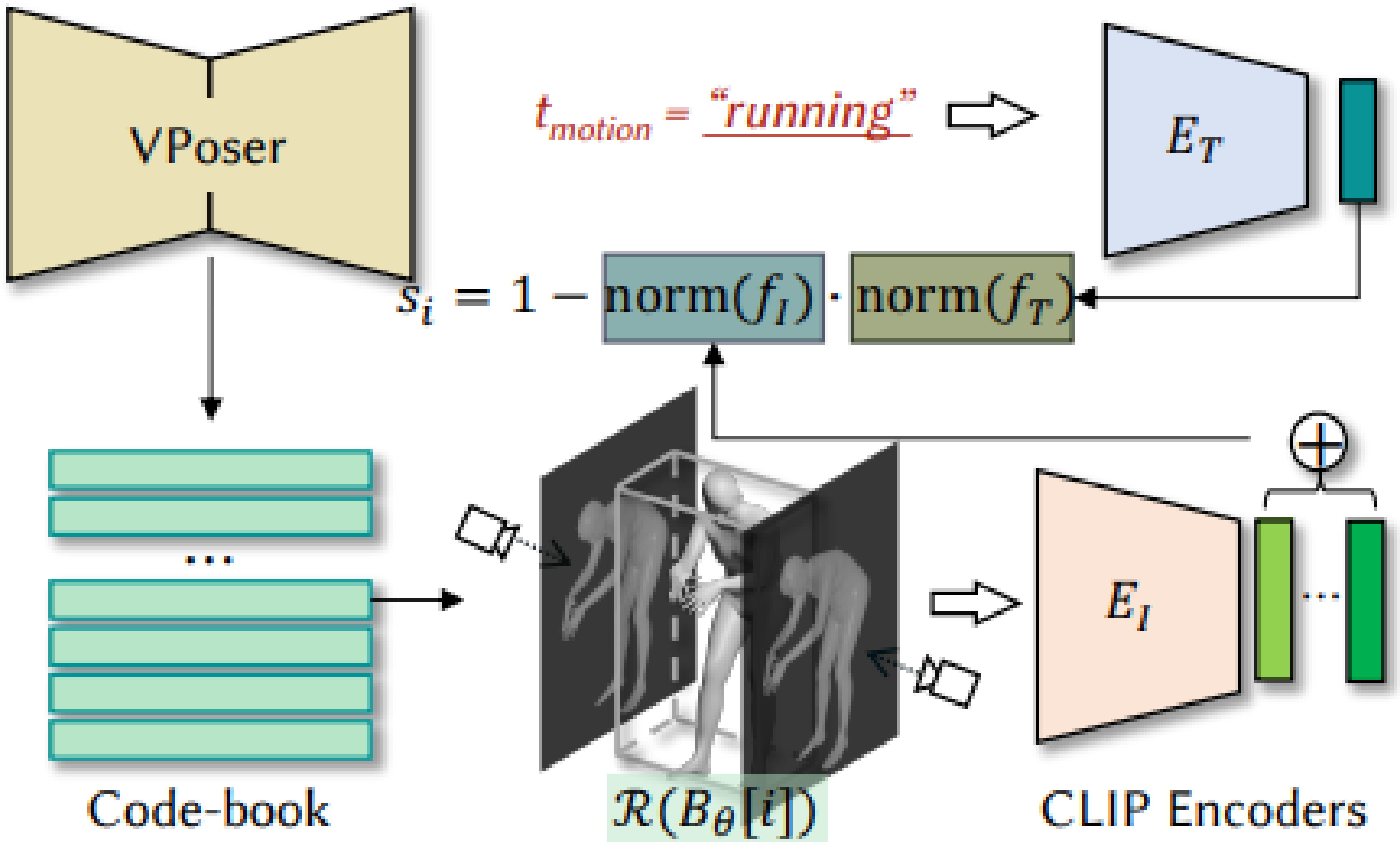

VPoser
t_{motion} = "running"
E_T
$s_i = 1 - \text{norm}(f_I) \cdot \text{norm}(f_T)$
...
E_I
...
Code-book
$\mathcal{R}(B_\theta[i])$
CLIP Encoders

Fig. 8. **Detailed Pipeline of Candidate Poses Generation.** The pre-trained VPoser is first used to build a code-book. Given text description t_{motion}, each pose feature f_I from the code-book is used to calculate the similarity with the text feature f_T, which is used to select Top-K entries as candidate poses.

Motion VAE

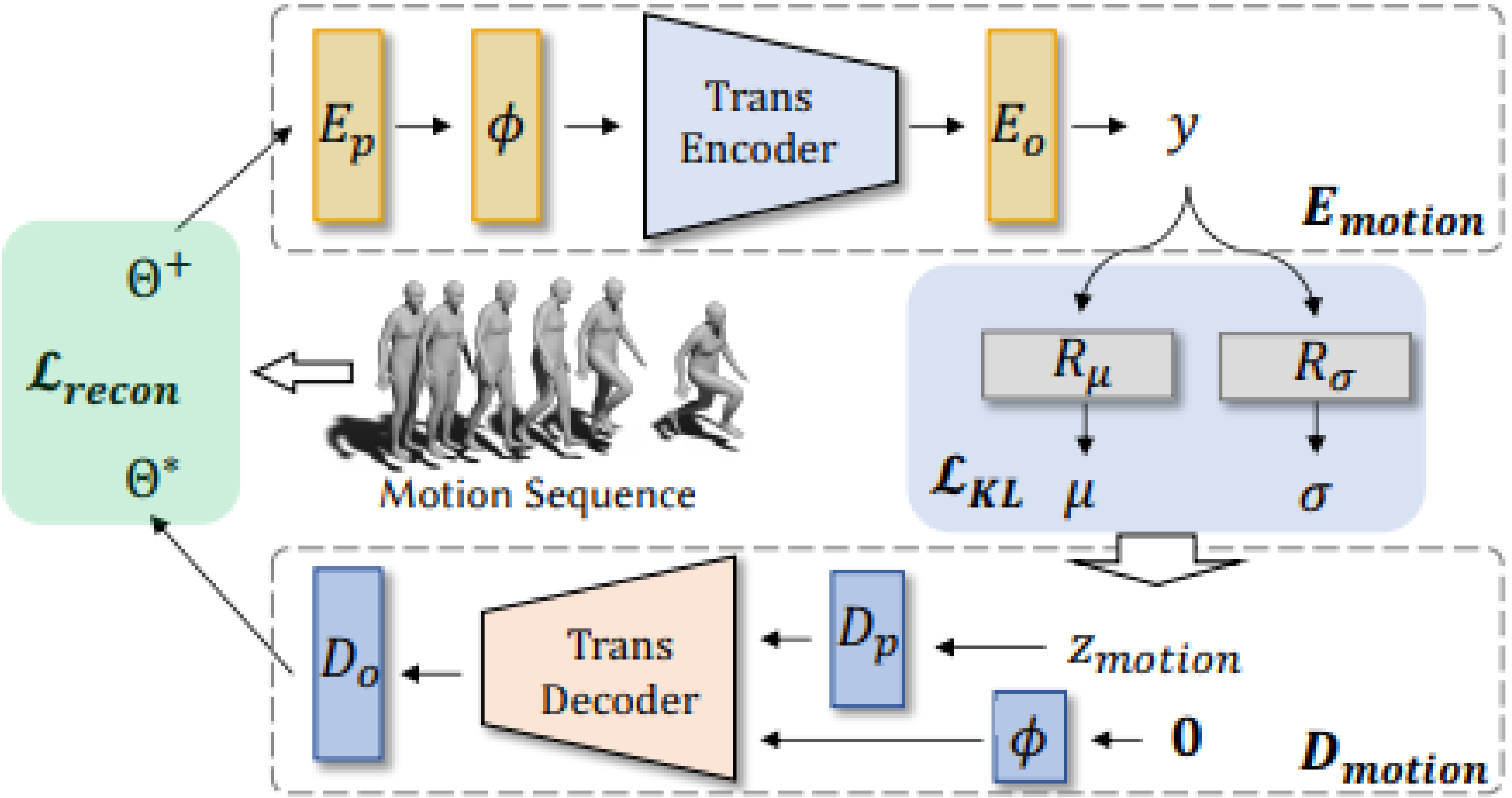


Fig. 9. **Structure of the Motion VAE.** The motion VAE contains three parts: the encoder E_{motion}, the decoder D_{motion}, and a reparameterization module. The reconstruction loss $\mathcal{L}_{\text{recon}}$ and the KL-divergence term $\mathcal{L}_{\text{KL}}$ are used for the motion VAE training.

Kullback-Leibler Divergence

- 표준 정규 분포와 가까워지도록 제약
- 잠재 표현의 연속성과 안정성을 보장

순환과 피드백

- 폐쇄 루프를 형성
- 지속적인 최적화

Overall Results of AvatarCLIP

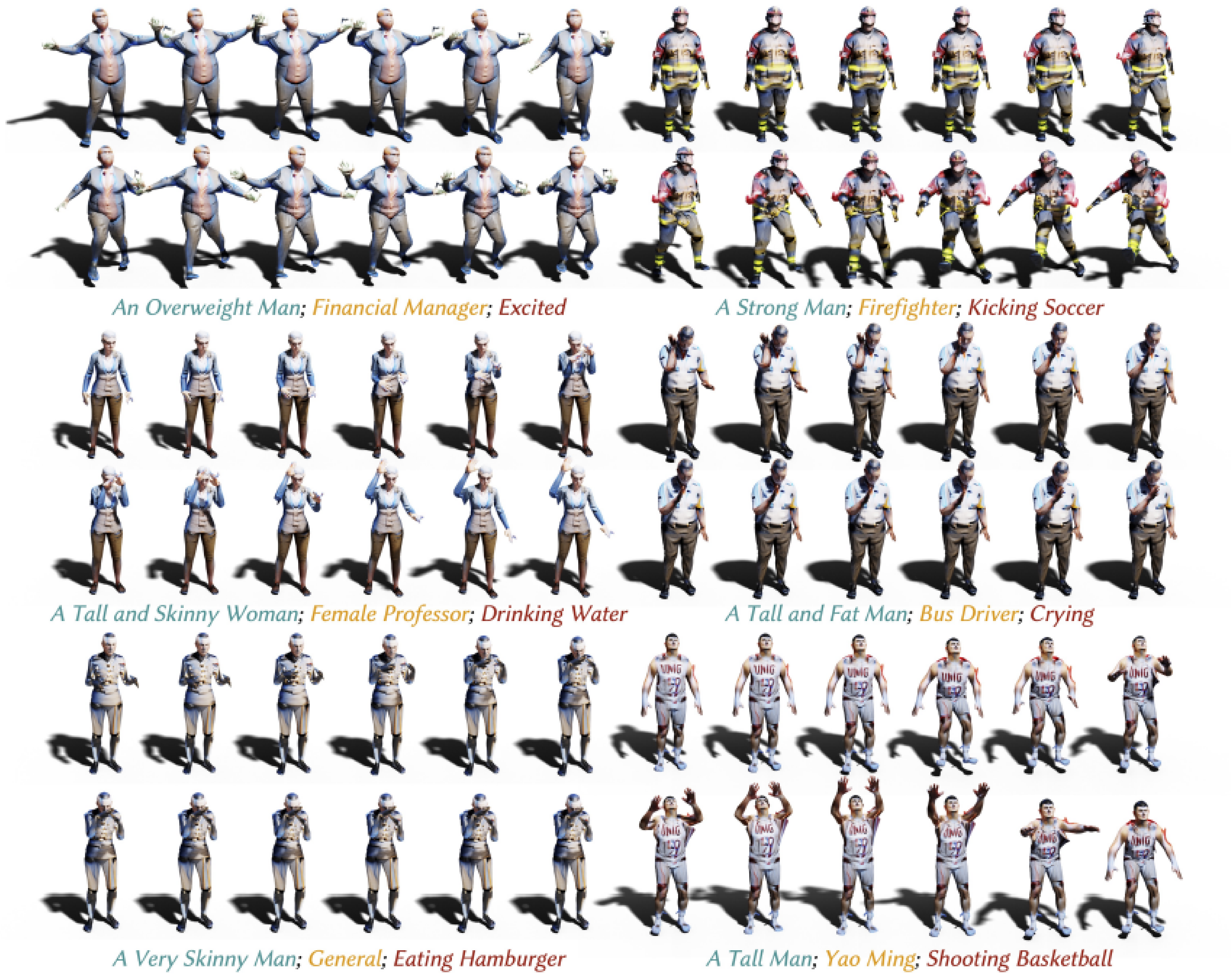

An Overweight Man; Financial Manager; Excited
A Strong Man; Firefighter; Kicking Soccer
A Tall and Skinny Woman; Female Professor; Drinking Water
A Tall and Fat Man; Bus Driver; Crying
A Very Skinny Man; General; Eating Hamburger
A Tall Man; Yao Ming; Shooting Basketball

Fig. 11. **Overall Results of AvatarCLIP.** Renderings of several animated 3D avatars are shown in sequence. The corresponding driving texts for shape, appearance and motion are put below the sequences.