Generative Proxemics: A Prior for 3D Social Interaction from Images
** Supplementary Material **

Lea Müller[1,4] Vickie Ye[1] Georgios Pavlakos[2,5] Michael Black[3] Angjoo Kanazawa[1]
[1]UC Berkeley [2]UT Austin [3]MPI for Intelligent Systems, Tübingen
{mueller,vye,kanazawa}@berkeley.edu, pavlakos@cs.utexas.edu, black@tuebingen.mpg.de

1. Creating Flicker Fits

To train BUDDI, we need 3D poses for two people interacting in close proximity. We create this training data by fitting SMPL-X to FlickrCI3D Signatures [3] with an optimization method that takes ground-truth contact annotations into account. FlickrCI3D Signatures is a publicly available datasets consisting of images collected from Flickr with 3D contact annotations. This is a complex task that requires data preprocessing steps. First, images in Flickr may contain children and infants who are not supported by SMPL-X [9]. Consequently, we follow previous work [8] and merge SMPL-X with SMIL-X [4] to represent a range of body shapes from children to adults. Second, we observe that keypoints detected by ViTPose [11] are more accurate than those detected by OpenPose [2], especially when people are occluding each other, except for the feet, which are often not detected. Therefore we merge OpenPose and ViTPose for our optimization method. Third, our method takes multiple modalities as input (keypoints and SMPL bodies estimated by BEV [10], and ground truth contact maps for Flickr images) and uses these in estimating high-quality SMPL bodies. Fourth, the detections of BEV are in SMPL format, while the ground-truth contact maps for Flickr are provided for the SMPL-X template mesh. Since these two body models are not compatible, *i.e.* their pose and especially their shape space is different, we create "approximate" BEV estimates in SMPL-X format by using the SMPL pose as if it was SMPL-X while properly converting the body shape from SMPL to SMPL-X.

1.1. Preprocessing

In the next section, we describe the preprocessing steps performed to create Flickr fits. We use the same preprocessing for our optimization with BUDDI.

Including children. Since SMPL [6] only models adult body shapes, most human pose and shape regressors do not consider child body shapes explicitly. However, we found that FlickrCI3D Signatures includes images of children (roughly 10% of the images). Following the SMPLA [8] convention, BEV also estimates a scale parameter s, which is used to interpolate between SMPL [6] (adult model) and SMIL [4] (infant model) for the template meshes and shape blend shapes. A scale value of $s = 0.0$ is equivalent to SMPL only, a scale value of $s = 1.0$ is equivalent to SMIL only, and all the values in between model intermediate stages. To extend this from SMPL to SMPL-X, we use the scale parameter estimated by BEV to interpolate between the SMPL-X and the SMIL-X template and shape blend shapes in SMPL-X topology. We visually found that this interpolation works well for $s \leq 0.8$, so we exclude pairs where the detected scale is $s > 0.8$ for one of the interaction partners. In practice, we concatenate the interpolation and body shape parameters such that $\boldsymbol{\beta} \in \mathbb{R}^{11}$. We refer to this model as SMPL-XA.

Matching input detections. As input, we have the estimated 3D bodies from BEV [10] and we have a dataset of ground-truth human-human contacts. The bodies in these two data sources are not in correspondence. To generate the Flickr Fits, we must first automatically put them in correspondence so that we can optimize the BEV bodies by exploiting the ground truth contact information.

In particular, we have (1) detected meshes from BEV, (2) 2D keypoint detections from ViTPose [11], and (3) ground-truth bounding boxes indicating the interacting pair of humans. We observed that the ground-truth bounding boxes typically match with the bounding boxes surrounding OpenPose [2] keypoint detections. As a result, we only need to correspond the OpenPose detections with ViTPose detections and the BEV bodies. Since we can reproject the 3D joints from BEV bodies to 2D keypoints, both correspondence problems require us to solve the assignment between sets of 2D keypoints. To do this, we compute a keypoint-cost matrix taking the detection confidence scores into account. We only consider keypoints with confidence score greater than 0.6 (for BEV all keypoints have by de-

[4] Portions of this work were done while LM was at MPI-IS, Tübingen.
[5] This work was done while GP was at UC Berkeley.

fault a score of 1.0 due to the amodal prediction of the human body). We make assignments in a greedy way, while also set a threshold (0.008) to discard matches with large matching distance.

Merging keypoints. Qualitatively, we found that ViTPose performs better than OpenPose, particularly for people that are heavily occluded. Since ViTPose (unlike OpenPose) does not detect keypoints on the feet, we can merge the ViTPose pose detections with feet keypoints detected by OpenPose. We perform this extension only if the L2 distance between ViTPose and OpenPose ankles is less than 5 pixels. Additionally, since many images in FlickrCI3D Signatures include people who are truncated below the waist, we often have missing or wrong keypoint detections for the lower body. Because of this, we use the projected BEV ankle joints, when the ankle keypoint detection confidence score is less than 0.2. Finally, the original keypoint values k_{orig} are normalized by the keypoint bounding box size via $k = k_{\text{orig}}/(\max(\text{bb}_{\text{height}}, \text{bb}_{\text{length}}) * 512)$. These steps give us a set of 2D keypoints that we use to generate the Flickr fits via an optimization method described below.

SMPL to SMPL-X body shape conversion. Our method takes BEV estimates as input and optimizes them to fit the image evidence. Since BEV estimates meshes in SMPL topology and the ground-truth contact maps are provided in SMPL-X format, we transfer the BEV estimate to SMPL-X. Ideally, one would fit SMPL-X to SMPL via optimization. This process is time consuming and we found that it is sufficient to initialize the optimization routine by using the SMPL pose parameters with the SMPL-X body. For body shape, we solve for the SMPL-X body shape using a simple least-squares optimization. The shaped vertices, V_{SMPL} and $V_{\text{SMPL-X}}$, are obtained via

$$\begin{aligned} V_{\text{SMPL}} &= T_{\text{SMPL}} + D_{\text{SMPL}}\beta_{\text{SMPL}}, \text{ and} \\ V_{\text{SMPL-X}} &= T_{\text{SMPL-X}} + D_{\text{SMPL-X}}\beta_{\text{SMPL-X}}, \end{aligned} \tag{1}$$

where T_{SMPL} and $T_{\text{SMPL-X}}$ are the SMPL and SMPL-X template meshes, D_{SMPL} and $D_{\text{SMPL-X}}$ the shape blend shapes, and β_{SMPL} and $\beta_{\text{SMPL-X}}$ the shape parameters. Only $\beta_{\text{SMPL-X}}$ is unknown. Since the topology between SMPL and SMPL-X is different, we use a SMPL-to-SMPL-X vertex mapping $M \in \mathbb{R}^{10475\times6890}$, such that $D_{\text{SMPL-X}} = MD_{\text{SMPL}}$. Then we can directly solve for body shape, $\beta_{\text{SMPL-X}}$, in a least-squares manner:

$$\beta_{\text{SMPL-X}} = (D^T_{\text{SMPL-X}}D_{\text{SMPL-X}})^{-1}D^T_{\text{SMPL-X}}MD_{\text{SMPL}}\beta_{\text{SMPL}}.$$

Additional details. We use the first 10 shape components and keep the facial expression and finger pose fixed. Note that, although we use SMPL-X, we do not optimize hand pose due to the lack of 3D data of close human interaction with hands, as well as the lack of robust finger keypoint detectors for people in close proximity. Extending this work to include detailed hand contact would be interesting future work.

1.2. Optimization with ground-truth contact annotations

We create SMPL-XA fits for Flickr images using ground-truth contact annotations. We use these fits to train our generative models, along with a small set of MoCap 3D poses. We also use them to evaluate the pose estimation error (JOINT PA-MPJPE) in Table 1 in the main manuscript.

Optimization-based methods for fitting 3D meshes to RGB images usually rely on sparse signals, like 2D keypoints (ground-truth or detected), and priors for human pose and shape [1, 9, 13]. Only a few methods explicitly use self- [7] or human-human [3] contact in their optimization.

Our optimization method takes as input the discrete human-human contact annotations and, for each person, detected 2D keypoints [2, 11], and initial estimates for their pose, $\tilde{\boldsymbol{\theta}}$, orientation, $\tilde{\boldsymbol{\phi}}$, shape, $\tilde{\boldsymbol{\beta}}$, and translation, $\tilde{\boldsymbol{\gamma}}$, which are provided by the output of BEV [10].

Given these inputs, we take a two-stage approach: In the first stage, we optimize pose, $\boldsymbol{\theta}$, shape, $\boldsymbol{\beta}$, and translation, $\boldsymbol{\gamma}$, encouraging contact between discretely annotated body regions, while allowing the bodies to intersect. In the second stage, we activate a new loss term to resolve human-human intersection. The output of the first stage is usually close to the final pose with only slight intersections, because of which we optimize only pose and translation and hold the body shape constant in stage two. The objective function is:

$$\begin{aligned} L_{\text{Cmap-fitting}} = &\lambda_J L_J + \lambda_{\bar{\theta}} L_{\bar{\theta}} + \lambda_\theta L_\theta + \\ &\lambda_{\boldsymbol{\beta}} L_{\boldsymbol{\beta}} + \lambda_P L_P + \lambda_{\mathcal{C}^D} L_{\mathcal{C}^D}, \end{aligned} \tag{2}$$

where L_J denotes the 2D re-projection error, $L_{\bar{\theta}}$ is a prior on the initial pose, L_θ is a Gaussian Mixture Model pose prior [1], and $L_{\boldsymbol{\beta}}$ an L2-prior that penalizes deviation from the SMPL-X mean shape. The discrete human-human contact loss, $L_{\mathcal{C}^D}$, minimizes the distance between vertices, v/u, assigned to regions, r, with annotated discrete human-human contact via:

$$L_{\mathcal{C}^D} = \sum_{i,j} \mathcal{C}^D_{ij} \min_{v\in r_i, u\in r_j} \|v-u\|^2. \tag{3}$$

L_P denotes an interpenetration loss, active in the second stage only, that pushes inside vertices to the surface. We use winding numbers to find intersecting vertices between two meshes, M^a and M^b, and vice versa. This operation is usually slow and memory intensive, which is why we use low-resolution meshes of SMPL-X with only 1K vertices.

With V_I^a we denote vertices of M^a intersecting the low-resolution mesh of M^b; V_I^b follows the same notation. The intersection loss term is defined as:

$$L_P = \sum_{v \in V_I^a} \min_{u \in V^b} \|v - u\|^2 + \sum_{v \in V_I^b} \min_{u \in V^a} \|v - u\|^2 . \quad (4)$$

We find functional weights, λ, for each term in the objective function (see Table 1). The results of this fitting approach are illustrated in Figure 1. We use this optimization routine to reconstruct interacting people depicted in the FlickrCI3D Signatures [3].

2. Diffusion model

Transformer architecture. To embed each body model parameter x_{ij} of person $j \in \{1, 2\}$ and parameters $i \in \{\boldsymbol{\phi}, \theta, \beta, \boldsymbol{\gamma}\}$ of size d_i in the latent space dimension $d_l = 152$, we use linear-SiLU-linear sequences:

$$f_{ij}(x_{ij}) = \text{SiLU}(x_{ij} A_{ij}^T + b_{ij}) B_{ij}^T + c_{ij},$$

where $A_{ij} \in \mathbb{R}^{d_l \times d_i}$, $b_{ij} \in \mathbb{R}^{d_l}$, $B_{ij} \in \mathbb{R}^{d_l \times d_l}$, and $c_{ij} \in \mathbb{R}^{d_l}$. After passing these parameters through the transformer, we again use a linear-SiLU-linear sequence to project them back into their original dimension d_i.

When BUDDI is trained with BEV [10] conditioning, we embed the conditioning in a similar fashion as the ground truth parameters, concatenate them along the token dimension, and add per-person and per-parameter embedding layers. In Fig. 2, we show the design of our conditional model.

3. Optimization

Here, we provide additional information for the optimization routines, *i.e.* optimization with contact map, VAE, heuristic, and BUDDI prior. In Table 1 we define the weights of each loss term. Every optimization runs for a maximum of 1000 iterations per stage, except optimization with BUDDI which we stop after 100 iterations. For termination, we use early stopping and we keep track of the loss value at the latest 10 iterations. The maximum runtime of our optimization method with BUDDI used as prior is about 160 seconds. We use these values to fit a line with linear regression $f(x) = ax + b$ and terminate if $a < -1e - 4$. We run each optimization for two stages. The second stage's reference poses, θ_0, which are used in $L_{\tilde{\theta}}$, are taken to be the output / last pose of the first stage. We provide pseudo code in Listing 1 showing the optimization routine with BUDDI used as prior.

4. Training and Testing Datasets

4.1. Flickr Fits

We split the Flickr [3] training images into training and validation sets and use the provided test split for testing. Fits

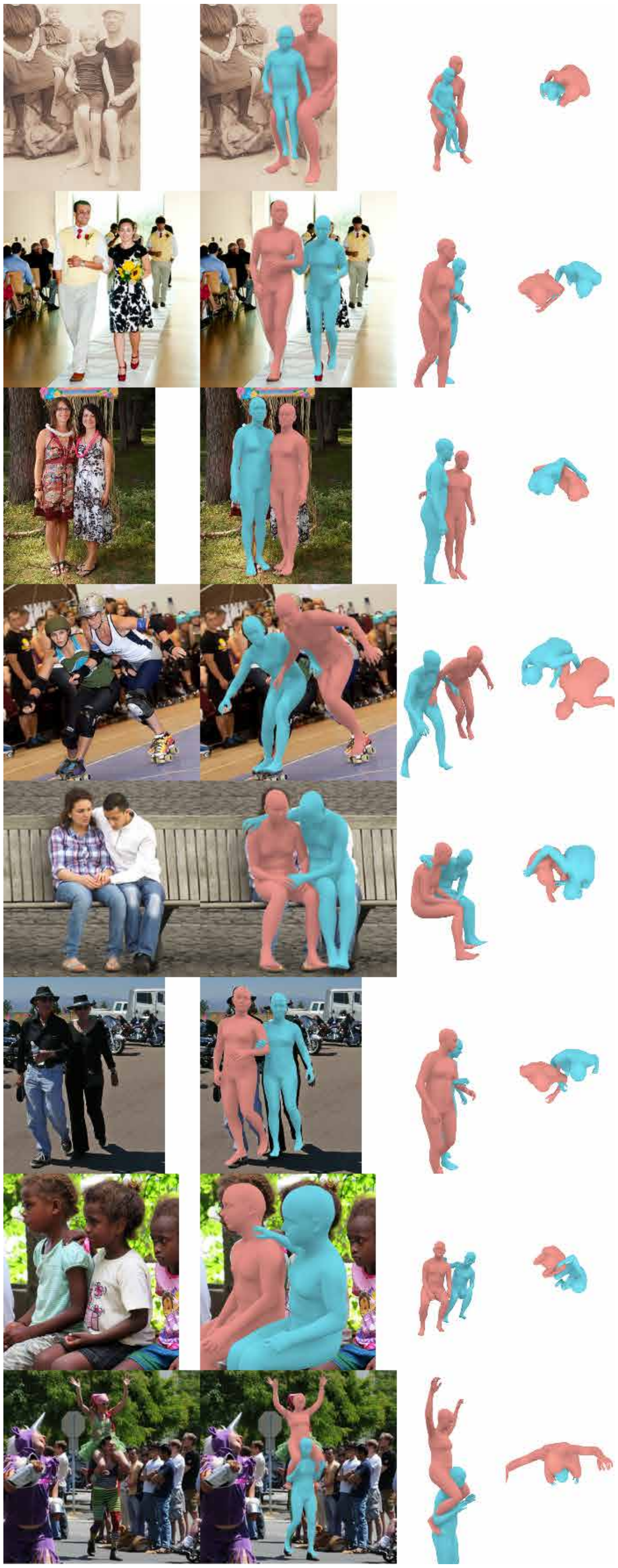


Figure 1. **Flickr Fits.** We visualize the output of the optimization process that reconstructs two people in close proximity using ground-truth contact maps, shown from three different views.

	λ_{J2D}	$\lambda_{\tilde{\theta}}$	λ_{θ}	λ_{β}	λ_{CB}	$\lambda_{d_{\text{min}}}$	λ_{P}	$\lambda_{\theta_{\text{BUDDI}}}$	$\lambda_{\gamma_{\text{BUDDI}}}$	$\lambda_{\beta_{\text{BUDDI}}}$	λ_{VAE}
Flickr Fits	0.04/0.1	200/200	4/4	40/0	10/10	0/0	0/1000	0/0	0/0	0/0	0/0
BUDDI	0.02/0.02	200/200	0/0	0/0	0/0	0/0	0/10	100/100	10/10	1e5/1e5	0/0
VAE	0.02/0.1	200/200	2/2	40/0	0/0	0/0	0/0.1	0/0	0/0	0/0	1/1
Heuristics	0.02/0.1	200/200	2/2	40/0	0/0	1e5/1e5	0/0.1	0/0	0/0	0/0	0/0
Heuristics (a)	0.04/0.1	200/200	4/4	40/0	0/0	1e5/1e5	0/1000	0/0	0/0	0/0	0/0
Heuristics (b)	0.02/0.02	200/200	4/4	40/0	0/0	1e5/1e5	0/10	0/0	0/0	0/0	0/0

Table 1. **Weights of the different loss term during the optimization.** We consider the case of using pseudo-ground truth contact maps, the heuristics, and BUDDI. Optimizations with BUDDI and pseudo-ground truth are run for two stages. The optimization with heuristics converges quickly so a single stage is enough.

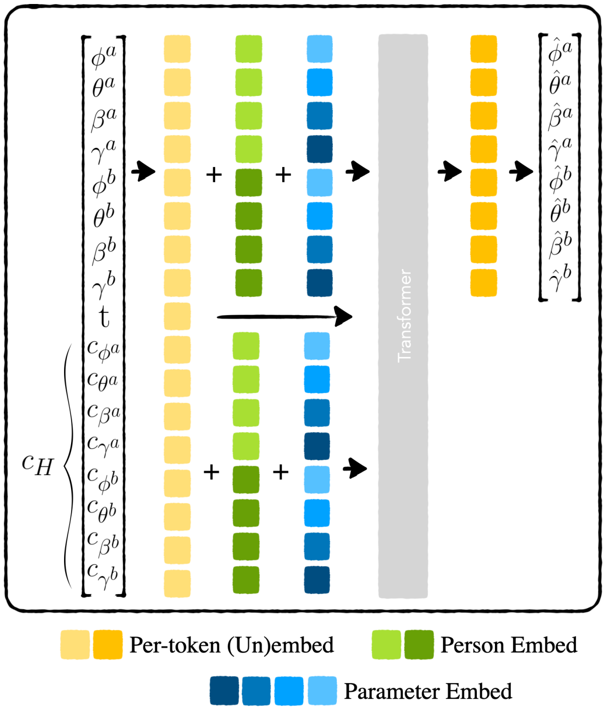


Figure 2. **Detailed architecture of BUDDI with conditioning.** When BUDDI is conditioned on model parameters, c_{H}, detected from BEV [10], we concatenate the detected parameters (body global orientation, pose, shape, and translation for person a/b), with the input parameters along the token dimension and add per-person and per-parameter embedding vectors.

can be noisy for example, when the assignment between contact annotations and keypoints is wrong or when keypoint detectors fail badly. To provide a reliable test set for 3D pose for images taken in the wild, we manually curate the Flickr Fits test set and detect 24 out of 1427 noisy fits. The final curated Flickr Test dataset contains 1403 interactions. We do not curate the training dataset.

We further evaluate the optimization method with ground truth contact maps on CHI3D (53/50mm PER-PERSON PA-MPJPE and 80mm JOINT PA-MPJPE) and on FlickrCI3D Signatures (45/87/97/99/100 PCC for radius 5/10/15/20/25).

4.2. Hi4D

Hi4D [12] is a MoCap dataset containing interaction between 20 pairs of people. Each pair performs about five interactions such as dancing, fighting, hugging, doing yoga, talking, etc. We split this dataset by subject pair into 14/3/3 for train/val/test. We use subjects [00, 01, 02, 09, 10, 13, 14, 17, 18, 21, 23, 27, 28, 37] for training, [16, 19, 22] for validation, and [12, 15, 32] for testing. Since Hi4D was originally provided in SMPL format, we fit SMPL-X to the estimates via optimization using the code provided in the SMPL-X repository [9]. The dataset provides a start and end frame from/to which each sequence involves physical contact between two people. We use every 5th frame from the contact sequence for training and testing.

4.3. CHI3D

CHI3D [3] is a MoCap dataset containing interactions between 3 pairs of people. Each pair performs eight interactions (grab, handshake, hit, holding hands, hug, kick, posing, and push) in various ways summing up to a total of about 120 sequences per subject pair. We use subjects [02, 04] for training and leave [03] for evaluation. Each sequence has a single frame with contact labels. We use this frame from each sequence for training and evaluation.

5. Evaluation

5.1. Baseline Methods

5.1.1 Transformer

We use the network design of BUDDI, *i.e.* embedding, person, and parameter layers, the transformer encoder block and layers to bring the latents back into parameter space. The network takes BEV [10] estimates as input and its task is to predict the correct SMPL-X parameters. We train this network on the same data as the conditional version of BUDDI. This baseline is equivalent to a single-shot (non-iterative) version of our diffusion model.

5.1.2 Contact Heuristic

We design an optimization method which is similar to the routine we use to create Flickr Fits, but replaces the $L_{\mathcal{C}^D}$, *i.e.* the loss that takes ground-truth contact maps into account, with a contact heuristic loss $L_{d_{\text{min}}}$. The contact heuristic loss encourages contact between the two people by minimizing their minimum distance. Given the vertices of each mesh, $v \in V_{X1}$ and $u \in V_{X2}$, we define the contact heuristic loss as

$$L_{d_{\text{min}}} = \min_{v,u} ||v - u||$$

and the overall objective function to be minimized becomes

$$\begin{aligned} L_{\text{Heuristic-fitting}} =& \lambda_J L_J + \lambda_{\bar{\theta}} L_{\bar{\theta}} + \lambda_\theta L_\theta + \\ & \lambda_{\boldsymbol{\beta}} L_{\boldsymbol{\beta}} + \lambda_P L_P + \lambda_{d_{\text{min}}} L_{d_{\text{min}}}. \end{aligned} \tag{5}$$

5.1.3 BUDDI (gen.)

The conditional version of BUDDI can generate human meshes in close social interaction from noise given a BEV estimate. We use these generations to initialize the optimization routine and evaluate them against the ground truth.

5.1.4 VAE

We also compare against VAE [5] using the same training data. This model projects the SMPL-X parameters of two people into latent vectors of size 64, modeling a distribution, and from the latent space back into parameter space. Similar to the design of BUDDI, we embed each parameter via an MLP. We use two encoder and two decoder layers. The VAE training loss is

$$L_{\text{VAE-training}} = L_\theta + L_\beta + L_\gamma + L_{v2v} + L_{\text{KL}}.$$

We use the same body model parameter losses as during BUDDI training. L_{KL} is a standard KL-divergence loss between two Gaussians:

$$L_{\text{KL}} = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

During optimization, instead of optimizing body model parameters, we optimize in the VAE's latent space. The optimization objective is:

$$\begin{aligned} L_{\text{VAE-fitting}} =& \lambda_J L_J + \lambda_{\bar{\theta}} L_{\bar{\theta}} + \lambda_\theta L_\theta + \\ & \lambda_{\boldsymbol{\beta}} L_{\boldsymbol{\beta}} + \lambda_P L_P + \lambda_{\text{VAE}} L_{\text{VAE}}, \end{aligned} \tag{6}$$

where L_{VAE} denotes a squared L2-loss on the VAE latent vector.

5.1.5 Ablation of baseline methods

We run our baseline methods under different conditions, *i.e.* we use different weights for the Heuristic for a better comparison against the weights used in Flickr Fits and when optimizing with BUDDI used as a prior. The loss weights of Heuristic (a) are similar to those of Flickr Fits and the weights of Heuristic (b) to those of BUDDI. We report these numbers in Table 2, Table 3, and Table 4.

5.2. Further analysis of optimization with BUDDI

Ablation of loss terms. To provide insights into the contribution of each loss term in the optimization routine with BUDDI used as prior, we consecutively add a term the the optimization routing starting from L_{J2D}, *i.e.* the 2D keypoint re-projection loss only, and evaluate the JOINT PA-MPJPE (↓) on Hi4D. The error is $118/118/111/99/99/98$ for L_{J2D}/+L_P/+$L_{\tilde{\theta}}$/+$L_{\gamma_{\text{BUDDI}}}$/+$L_{\beta_{\text{BUDDI}}}$/+$L_{\theta_{\text{BUDDI}}}$. This result emphasizes the importance of the translation prior, $L_{\gamma_{\text{BUDDI}}}$, which has the biggest impact on the final result. BUDDI knows about the relative 3D translation between two people in close social interaction and guides the optimization routine towards more plausible relative depth.

Heuristic vs. BUDDI. Our evaluation indicates that the heuristic provides a strong baseline as it achieves a low PA-MPJPE on all three datasets. Our hypothesis is that the Heuristic is particularly strong for poses with only a few physical contact points. To quantify this assumption, we compute the percentage of vertices that are in contact (with distance ≤ 10cm to the other person) for each action in Hi4D. Ordering activities by amount of contact in Hi4D gives: basketball (7%), dance (9%), fight (10%), highfive (12%), talk (18%), backhug (23%), cheers (24%), pose (29%), kiss (46%), sidehug (47%), hug (53%). The contact percentage is indicated in brackets; red means the heuristic is better than BUDDI and green means BUDDI is better the heuristic. As the amount of contact increases (i.e. becomes more complex), BUDDI significantly outperforms the heuristic. On average, BUDDI outperforms the heuristic by 23 mm.

5.3. Perceptual study

We provide several quantitative evaluations of BUDDI in the main paper but there are aspects of human interaction that are subtle and best judged by people. In the main part of this paper we present the results of the perceptual study that evaluates how realistic the generated interactions sampled from BUDDI are compared to meshes sampled from a VAE, the training data, and a random configuration of meshes. Here, we show the layout and instructions of the perceptual study in Figure 3. We randomly sample 256 meshes from one training batch of size 512 created with a 60/20/20 ratio of meshes from Flickr/Hi4D/CHI3D. The meshes from the

	PER PERSON ↓	JOINT ↓	JOINT PA-MPJPE ↓										
	PA-MPJPE	PA-MPJPE	backhug	basketball	cheers	dance	fight	highfive	hug	kiss	pose	sidehug	talk
Heuristic	67 / 71	121	168	83	94	131	94	68	159	159	118	113	109
Heuristic (a)	68 / 72	122	166	82	93	126	92	68	161	158	122	122	114
Heuristic (b)	68 / 73	124	164	90	92	130	95	68	161	158	125	124	117

Table 2. **Evaluation of BUDDI on Hi4D.** We compare the output of BUDDI to the proposed baseline methods on the Hi4D challenge. The first block shows methods that do not use Hi4D data during training or are optimization based without access to priors trained on Hi4D. BUDDI (F,C) in particular, is our model BUDDI trained on Flickr and CHI3D data only. All errors are reported in mm for 3D Joints.

	PER PERSON ↓	JOINT ↓
	PA-MPJPE	PA-MPJPE
Heuristic	49 46	105
Heuristic (a)	49 47	103
Heuristic (b)	47 45	103

Table 3. **Quantitative Evaluation on CHI3D.** We compare different versions of the baseline optimization with contact heuristic on CHI3D (pair s03). All errors reported in mm for 3D Joints.

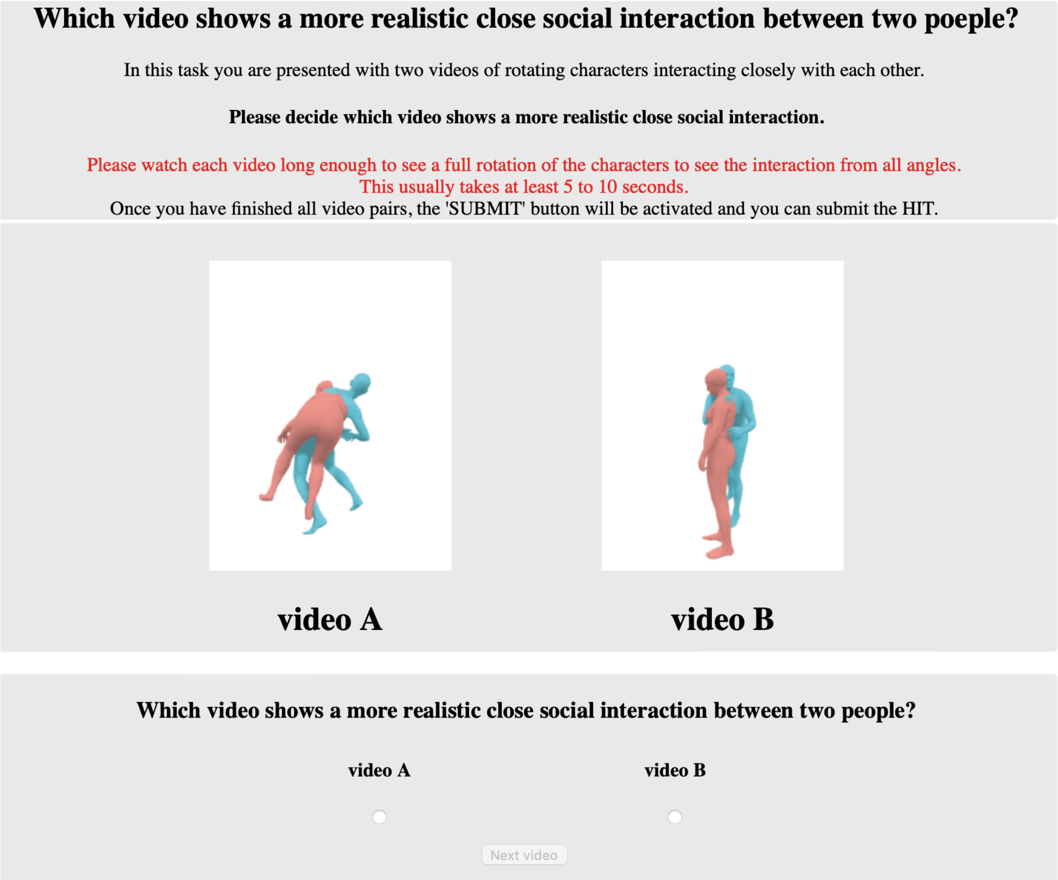
	JOINT ↓	PCC at radius ↑				
	PA-MPJPE	5	10	15	20	25
Heuristic	68	14	34	49	61	70
Heuristic (a)	69	11	30	45	57	66
Heuristic (b)	72	12	30	45	57	67

Table 4. **3D Pose Evaluation on FlickrCI3D Signatures.** We compare different versions of the baseline optimization with contact heuristic on the Flickr fits using their joint (two-person) PA-MPJPE expressed in mm. We also evaluate the percentage of correct contact points (PCC) for radius *r* mm.

training batch are real samples from MoCap or by fitting SMPL-X to images with ground-truth contact map annotations. We sample 256 from BUDDI (unconditional model) and the VAE. To create the random baseline, we center all meshes in the training batch, shuffle the people along batch and person dimensions, and sample 256 mesh pairs. This is equivalent to real samples, except that each person are sampled randomly and not as a pair. Using Amazon Mechanical Turk (AMT), each participant was asked to rate 68 video comparisons per human intelligence task (HIT) with each video showing one pair of meshes at 360-degree views. Each HIT starts with 10 training videos (not used in evaluation) and contains 10 catch trials. Catch trials show implausible interaction, *e.g*. two people with random poses placed on top of each other. The training videos are presented at the beginning of the task, and the method and catch trial videos appear in random order. The remaining 48 comparisons show one sample from BUDDI against either VAE / random baseline / or training data (12 comparisons per method). We randomly shuffle the video order per HIT and left / right. Each HIT is conducted by 6 participants. We exclude HITS where participants fail three or more catch trials. Our final results were computed with the responses from the 83/96 participants who passed.

5.4. Additional qualitative results and failure cases

We provide additional qualitative examples of optimization with BUDDI and compare them to optimization with heuristics and BEV in Figure 4 and Figure 6. Failure cases are provided in Figure 7.



Which video shows a more realistic close social interaction between two poeple?

In this task you are presented with two videos of rotating characters interacting closely with each other.

Please decide which video shows a more realistic close social interaction.

Please watch each video long enough to see a full rotation of the characters to see the interaction from all angles.
This usually takes at least 5 to 10 seconds.
Once you have finished all video pairs, the 'SUBMIT' button will be activated and you can submit the HIT.

video A **video B**

Which video shows a more realistic close social interaction between two people?

Figure 3. **Amazon Mechanical Turk perceptual study layout and instructions.** On the left, we show a 360-degree video of the two interacting people. On the right, the rating scale.

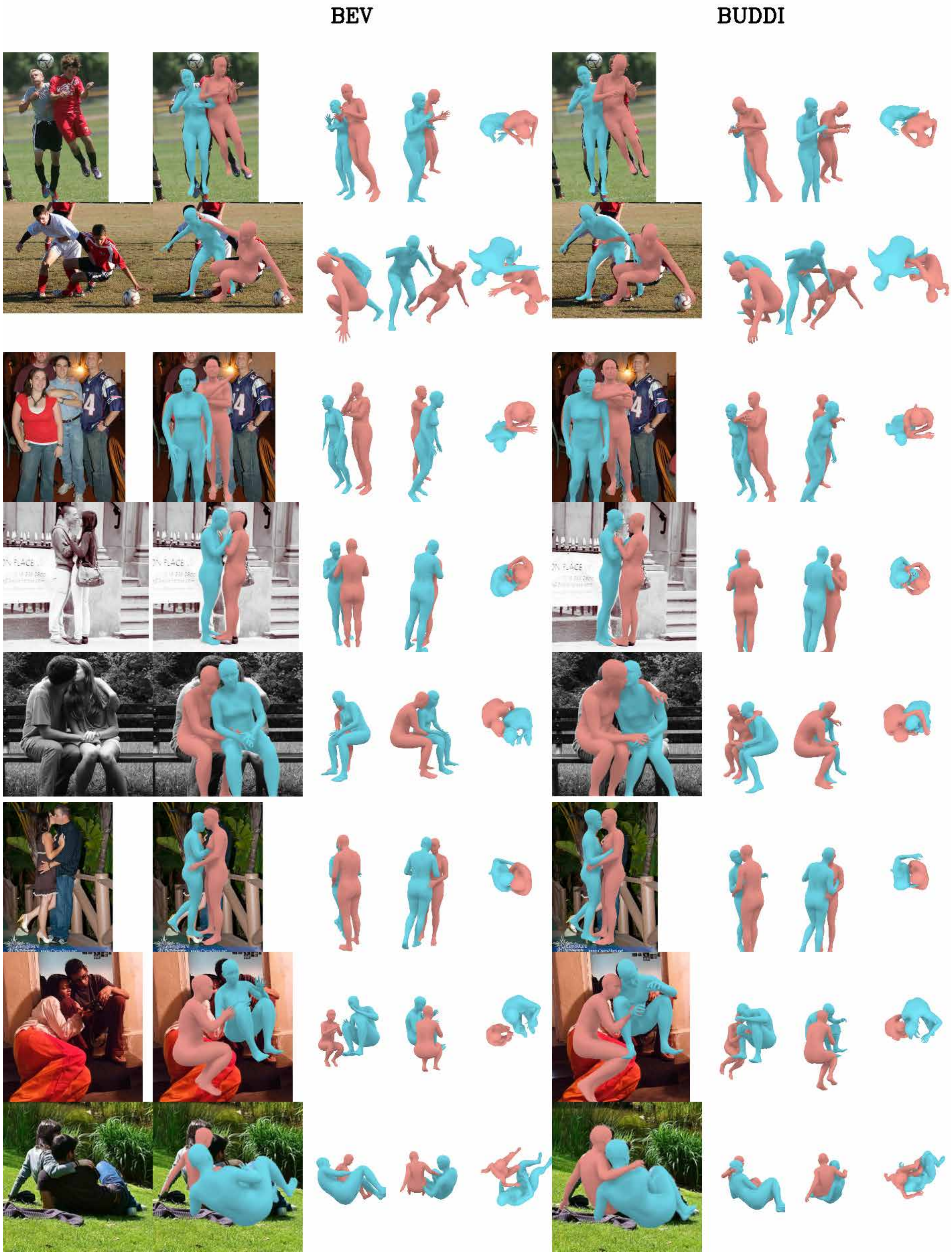


Figure 4. **Optimization with BUDDI.** Additional qualitative examples from optimization with BUDDI compared to BEV. We provide the overlay and three additional views per method. Optimization with BEV (first method / columns 2-5), optimization with BUDDI (second method / columns 6-9).

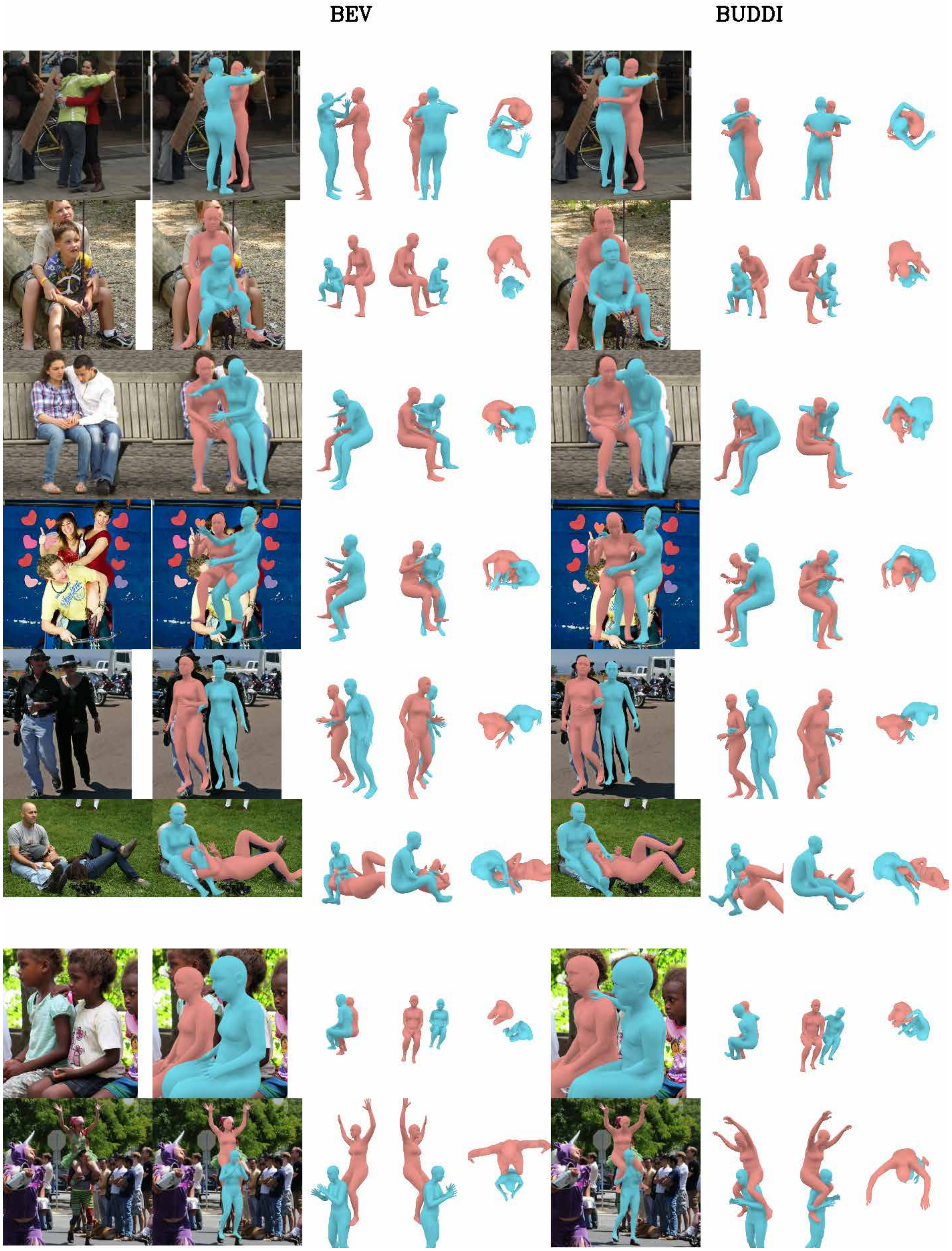


Figure 5. **Optimization with BUDDI (continuation).** Additional qualitative examples from optimization with BUDDI compared to BEV. We provide the overlay and three additional views per method. Optimization with BEV (first method / columns 2-5), optimization with BUDDI (second method / columns 6-9).

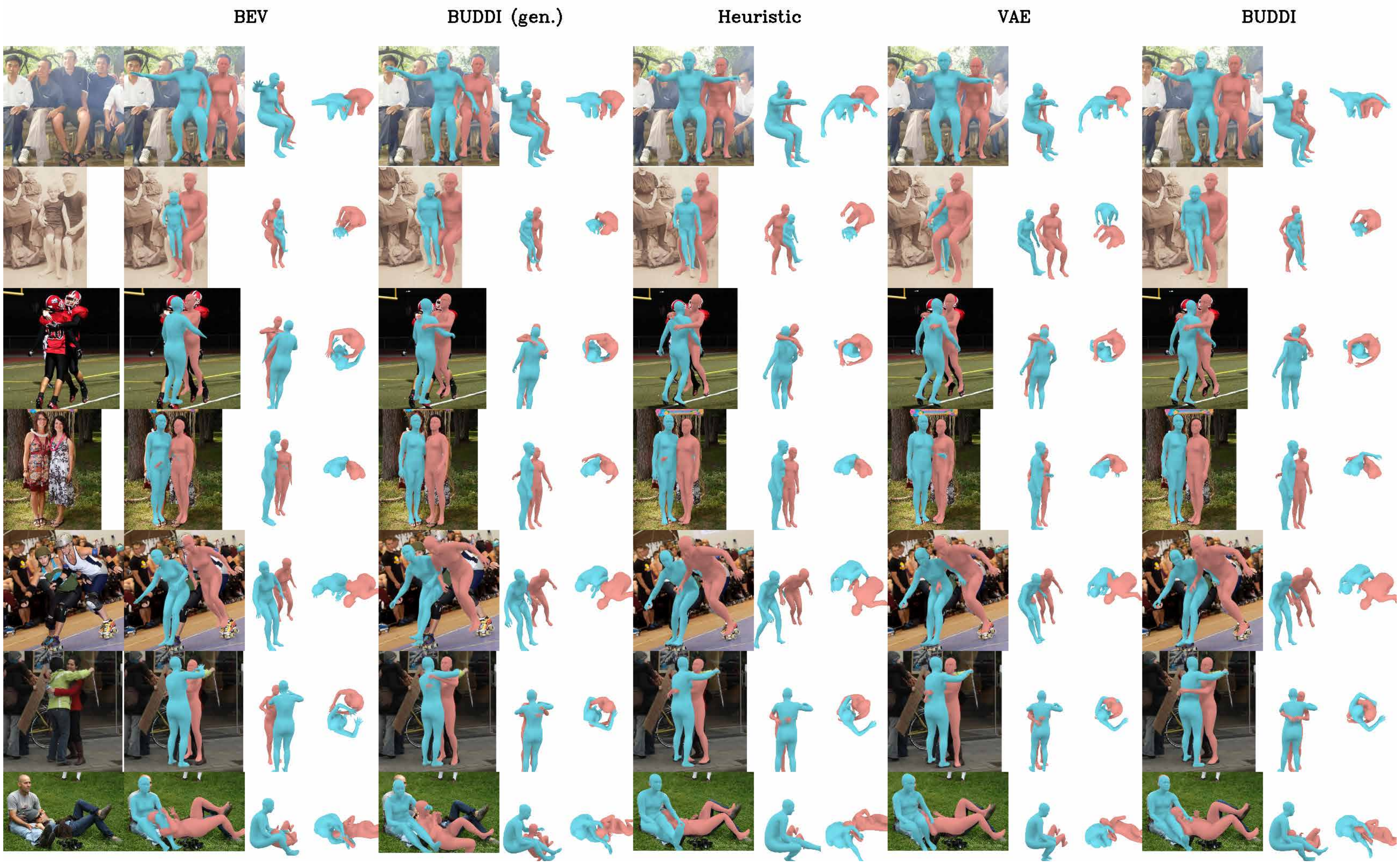


Figure 6. **Optimization with BUDDI.** Additional qualitative examples from optimization with BUDDI compared to BEV, BUDDI generations, optimization with heuristic, and optimization with VAE. We provide the overlay and three additional views per method. BEV (first method / columns 2-5), BUDDI (gen.) (second method / columns 6-9), optimization with heuristic (third method / columns 10-13), optimization with VAE (fourth method / columns 14-16), and optimization with BUDDI (fifth method / columns 17-20)
.

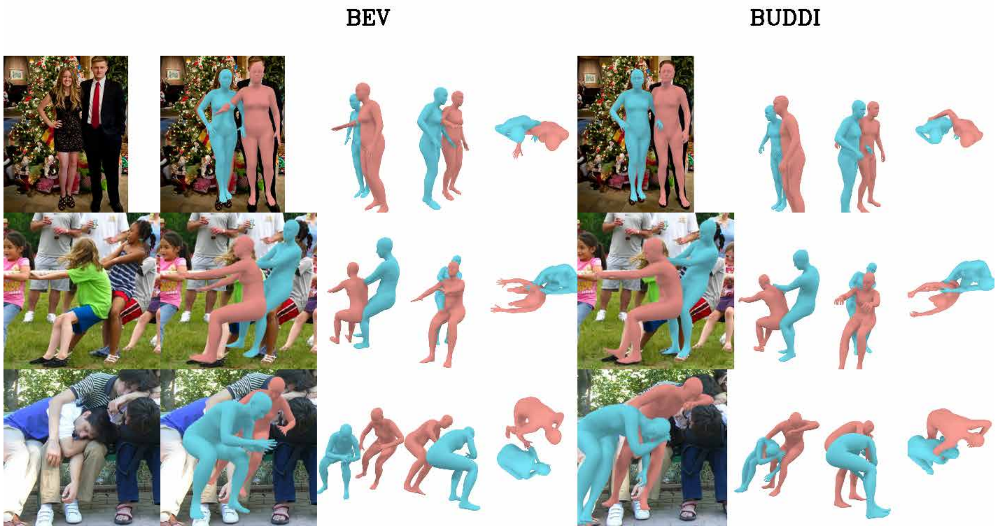


Figure 7. **Failure cases optimization with BUDDI.** Failure cases from optimization with BUDDI. In the first row the depth ordering of leg arm is wrong. The image in row 2 contains less common contact so that BUDDI suggests for blue to hold red's shoulders instead of the rope. The estimated predicted by our method suggests a plausible pose that is not consistent with the image due to single-view ambiguity. The last row shows a failure case due to intersection between arm and torso.

```
import smplx
import buddi

# optimization params
num_stages = 2
max_iterations = 100
t = 10 # noise level

# create smpl and buddi
smpl = smplx.create(model_folder)

# load buddi denoiser model (D)
buddi = buddi.create(checkpoint_path).eval()

# load detected keypoints and bev
kpts = load_keypoint_detections(img_path)
bev = load_bev_estimate(img_path)

# sample from buddi conditioned on BEV
buddi_sample = sample_from_buddi(cond=bev)

# initialize the optimization
smpl.params = buddi_sample

# run optimization
for ss in range(num_stages):
  optimizer = setup_optimizer(smpl, ss)

  for ii in range(max_iterations):
    # fitting losses
    fitting_loss = get_fitting_loss(
                        smpl, buddi_sample, kpts)

    # detach current smpl, then diffuse & denoise
    with torch.no_grad():
      diffused_smpl = smpl + sample_noise(t)
      denoised_smpl = buddi(diffused_smpl, t)

    # compute diffusion losses
    diffusion_loss = get_diffusion_loss(
                        smpl, denoised_smpl)

    # final loss of iteration ii of stage ss
    total_loss = fitting_loss + diffusion_loss

    # backprop
    optimizer.zero_grad()
    total_loss.backward()
    optimizer.step()

    # check stopping criterium
    if converted:
      break
```

Listing 1. Pseudo code for optimization with BUDDI.

References

[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, pages 561–578. Springer International Publishing, 2016.

[2] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43 (1):172–186, 2019.

[3] Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. Three-dimensional reconstruction of human interactions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 7214–7223, 2020.

[4] Nikolas Hesse, Sergi Pujades, Javier Romero, Michael J Black, Christoph Bodensteiner, Michael Arens, Ulrich G Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, et al. Learning an infant body model from rgb-d data for accurate full body motion analysis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 792–800. Springer, 2018.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *Transactions on Graphics (TOG)*, 34 (6):248:1–248:16, 2015.

[7] Lea Müller, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black. On self-contact and human pose. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9990–9999, 2021.

[8] Priyanka Patel, Chun-Hao Paul Huang, Joachim Tesch, David Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in geography optimized for regression analysis. In *Computer Vision and Pattern Recognition (CVPR)*, pages 13468–13478, 2021.

[9] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[10] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In *Computer Vision and Pattern Recognition (CVPR)*, pages 13243–13252, 2022.

[11] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022.

[12] Yifei Yin, Chen Guo, Manuel Kaufmann, Juan Zarate, Jie Song, and Otmar Hilliges. Hi4d: 4d instance segmentation of close human interaction. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[13] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2148–2157, 2018.