MODELING UNCERTAINTY PROPAGATION IN 3D MORPHABLE MODELS

Anonymous authors
Paper under double-blind review

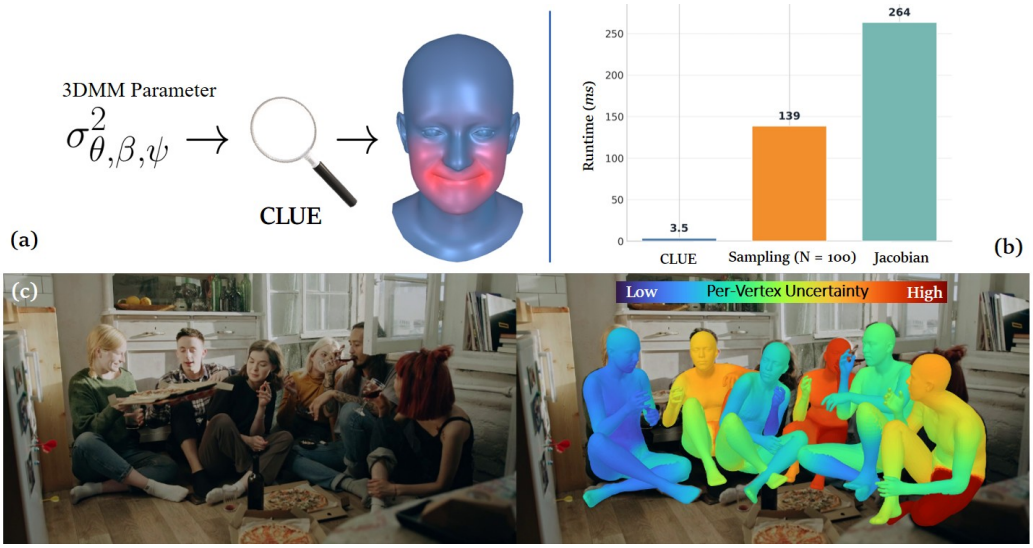


Figure 1: a) **CLUE**: **C**losed-form **L**ocal **U**ncertainty **E**stimation is a novel transformation that maps parameter-space distribution to local, per-vertex distribution for 3D Morphable Models. b) **Accurate and Real-time**: CLUE closely approximates sampling-based methods, while being $\sim$ **70**$\times$ faster. c) **In-the-wild**: We share ***SAM 3D Body (CLUE)***, augmenting a frozen SAM 3D Body model with high-resolution uncertainty maps.

ABSTRACT

We study the problem of mapping parameter-space variance of a 3D Morphable Model (3DMM) to per-vertex variance. Traditionally, per-vertex variance has been used to visually interpret the learnt parameter uncertainty. However, mapping parameter-space variance to per-vertex using sampling or error propagation is computationally expensive. We address this limitation with **C**losed-form **L**ocal **U**ncertainty **E**stimation (CLUE), which transforms the parameter-space distribution of 3DMMs to per-vertex through a closed-form approximation. CLUE emerges from applying the law of total variance to the 3DMM formulation and is modular, computationally inexpensive, and accurate. Subsequently, we study integrating CLUE for in-the-wild human mesh recovery training and inference. We note that CLUE enables probabilistic mesh alignment as a training objective, and uses the geometry of the mesh as an inductive bias when predicting the variance. Through extensive qualitative and quantitative experiments, we show that when used in training, CLUE (1) outpeforms probabilistic and deterministic baselines on metrics evaluating joints, mesh and uncertainty, (2) is at par with SAM 3D Body on 3DPW using 3$\times$ fewer training samples, and (3) has the highest uncertainty-to-error spearman correlation, with applications including data mining and interpretable uncertainty. Finally, we share SAM 3D Body (CLUE) which augments the regressor with high resolution uncertainty maps.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

3D Morphable Models (3DMMs) Loper et al. (2015); Pavlakos et al. (2019); Li et al. (2017) have been critical to the success of digital humans in real world applications. A popular application of 3DMMs is in-the-wild Human Mesh Recovery (HMR) (Kanazawa et al., 2018; Bogo et al., 2016; Yang et al., 2026b), which reconstructs humans by estimating their 3DMM parameters from in-the-wild images. However, in-the-wild images often include factors such as occlusions, poor image quality and challenging poses. To address these uncertainties, methods have been developed to predict a distribution over the 3DMM parameters using the the Gaussian Negative Log-Likelihood (GNLL) and its variants Sengupta et al. (2021c;b); Dwivedi et al. (2024). The parameter variance is then mapped to the mesh to obtain interpretable, spatially localized per-vertex uncertainty. However, mapping the parameter distribution onto the mesh is non-trivial.

Popular methods in human mesh regression Sengupta et al. (2021c;b) compute per-vertex variances through repeated sampling. A large ($\approx$ 100) number of samples are drawn from the parameter distribution which are then used to obtain a set of meshes. The per-vertex variance is computed empirically from this set of meshes. This approach is, however, computationally intensive and is sensitive to the number of samples. In statistics, error propagation for Guides in Metrology (2008); Shukla et al. (2024) is a powerful tool to map uncertainty across a smooth function. While error propagation provides a closed-form approximation of the per-vertex uncertainty, the technique is also computationally expensive. Moreover, operators such as the Jacobian may lack compatibility with popular on-device machine learning frameworks such as XNNPACK Google (2026).

To address this limitation, we present Closed-form Local Uncertainty Estimation (CLUE), a novel formulation to map the parameter-space variance to per-vertex. CLUE is an *accurate* closed-form transformation that is *computationally efficient* and renders 3DMM uncertainty *interpretable*. CLUE uses the law of total variance and error propagation to analyse the 3DMM formulation, disentangling variance from pose, shape and expression. Furthermore, we study integrating CLUE in training probabilistic in-the-wild human mesh regressors. Recent advances in optimizing under uncertainty (Shukla et al., 2024) highlight the use of inductive bias when modeling the predicted variance. In a similar vein, CLUE enables *probabilistic mesh alignment as a training objective* and guides the predicted parameter variance through the geometry of the 3DMM, serving as an inductive bias. As a result, we observe significantly improved optimization of the HMR model. CLUE is also agnostic to the architecture of the HMR model, allowing for a seamless integration in training and inference.

We study three popular 3DMMs: (1) FLAME (Li et al., 2017), (2) SMPL-X (Pavlakos et al., 2019), and (3) MHR (Ferguson et al., 2025), which are also representative of SMPL Loper et al. (2015) and STAR Osman et al. (2020). Next, we perform two sets of experiments: (a) Statistical experiments to evaluate CLUE as a transformation from parameter to per-vertex distribution, and (b) Real-world experiments to qualitatively and quantitatively evaluate the use of CLUE in training HMR models. Our experiments confirm that CLUE is accurate and computationally inexpensive, as well as outperforms probabilistic losses over joints, mesh, negative log-likelihood and correlation metrics. In comparison to state-of-the-art SAM 3D Body (Yang et al., 2026b), our model with an identical architecture but trained from scratch with CLUE achieves similar performance with $3\times$ fewer data points on 3DPW. Furthermore, we discuss applying CLUE for data mining and interpretable uncertainty for downstream decision making. Finally, we share SAM 3D Body *(CLUE)* which augments SAM 3D Body with probabilistic vertices for uncertainty based reasoning. We share example visualizations in Figs. 10, 11.

2 RELATED WORK

3D Morphable Model. 3DMMs are data-driven statistical models that represent different variations in human identity, expression and pose. They are typically constructed by applying Principal Component Analysis (PCA) to a registered collection of 3D meshes. These models represent geometry in rest-pose through a linear combination of shape, expression and pose-corrective bases. The rest-pose geometry can further be articulated through Linear Blend Skinning (LBS).

$$v_i' = \sum_{j \in \mathcal{J}} w_{i,j} \mathbf{G}_j(\theta, \mathbf{J}) \left(\overline{v}_i + B_{S,i}(\beta) + B_{E,i}(\psi) + \mathcal{K}_{P,i}(\theta) \right) \tag{1}$$

Following Eq. 1, a 3DMM modifies a mean template $\bar{v}_i$ through a series of additive offsets. In this formulation, B_S, B_E, and $\mathcal{K}_P$ are the linear bases representing shape, expression, and pose correctives, respectively. To articulate the mesh, $\mathbf{G}_j$ represents the rigid transformation of each joint J, while $w_{i,j}$ denote the skinning weights that determine how these transformations influence each vertex. We kindly refer to FLAME Li et al. (2017), SMPL-X Pavlakos et al. (2019) for a more detailed description of how 3DMMs are built.

Deterministic mesh regression. There have been significant works which leverage 3DMMs for HMR such as Kanazawa et al. (2018), SPIN (Kolotouros et al., 2019), PARE (Kocabas et al., 2021), Mesh Graphormer (Lin et al., 2021), CLIFF (Li et al., 2022), SAM 3D Body (Yang et al., 2026b) and Fast SAM 3D Body (Yang et al., 2026a). For faces, the classical 3DMM Blanz & Vetter (2023) underpins a large family of analysis-by-synthesis fitters, and deep regressors Deng et al. (2019) extend this to in-the-wild imagery. Yet, deterministic methods produce a point estimate and do not capture ambiguities in their prediction.

Probabilistic approaches. Under strong occlusion, ProHMR (Kolotouros et al., 2021) explicitly encodes the posterior as a multimodal distribution captured using normalizing flows. The use of normalizing flows is further refined in HuManiFlow (Sengupta et al., 2023) and Biggs et al. (2020). Diffusion and score-based generative models (Cho & Kim, 2023; Stathopoulos et al., 2024) explore its use in improved HMR. Similarly, for faces under occlusion, Diverse3DFace (Dey & Boddeti, 2022) and OFER (Selvaraju et al., 2025) use generative modeling over the FLAME parameters. While multimodal approaches effectively capture diverse modes through sampling, they do not explicitly model the confidence of individual modes. Unimodal methods focus on modeling ambiguity due to occlusion, lighting, noise and partial visibility. One line of work models pose parameters which live on $SO(3)$, requiring that pose distributions must respect the rotation manifold (Sengupta et al., 2021b; Dünkel et al., 2024; Shetty et al., 2024). Methods such as Egger et al. (2016; 2018); Li et al. (2023) focus on inference under occlusion and do not explicitly model the uncertainty. Along similar lines as this work, Sengupta et al. (2021a) uses error propagation, but the solution is applicable only in the rest pose with no rotations. When predicting Gaussians over both the shape *and* pose parameters, Sengupta et al. (2021c) defaults to using Monte Carlo (MC) sampling. To eliminate Monte Carlo sampling, POCO (Dwivedi et al., 2024) regresses uncertainty in a single forward pass. However, per-part uncertainty is derived heuristically and does not address the mapping of parameter distribution to the target distribution.

Uncertainty Estimation. Several approaches exist (Nix & Weigend, 1994; Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Blundell et al., 2015) to quantify uncertainty in the prediction. Among them, Kendall & Gal (2017); Nix & Weigend (1994) jointly optimize the mean and variance of the prediction using GNLL. In our context, this translates to learning the 3DMM parameters and its variance. Transforming the parameter distribution to the mesh involves error propagation (for Guides in Metrology, 2008; Schaefer et al., 2023), with the linear variant being $\Sigma_y = J\Sigma_x J^T$, where $y = f(x)$. While Sengupta et al. (2021a) applies this, it assumes that there is no rotation *w.r.t* the canonical pose. TIC-TAC (Shukla et al., 2024) extends error propagation to general nonlinear networks, but remains computationally expensive and abstracts away the interpretability in the formulation. This leaves a gap in recent works on a closed-form transformation which is not only agnostic to any 3DMM formulation but is also accurate, fast and interpretable.

3 Closed-form Local Uncertainty Estimation

Let X be the random variable denoting a vertex after applying additive offsets, and Y be the random variable denoting the rigid transformation of the vertex. We derive CLUE by asking what is Var(Y)? Our key insight is the law of total variance, which ties the uncertainty in Y as a function of an explanatory variable X. We provide brief insights here and detail in the appendix (Sec. A).

Assumptions. We inherit our assumptions from the standard 3DMM formulation and related literature (Loper et al., 2015; Sengupta et al., 2021c;b; Schaefer et al., 2023; Dwivedi et al., 2024): the pose θ, identity β, and expression ψ parameters are independent. Next, we drop the pose correctives $\mathcal{K}_{P,i}(\theta)$ to allow for a simpler formulation, since the magnitude of change induced by the pose is much larger than those from pose correctives (15 v.s. 0.5cm) (Osman et al., 2020). Dropping pose correctives allows for a simpler formulation; yet for completeness we discuss incorporating pose correctives in the appendix (Sec: A.6).

3.1 DERIVATION

Additive Offset Variance. Formally, let X_i be the random variable denoting the vertex i after additive offsets, $X_i = \bar{v}_i + \vec{\beta} B_{S,i} + \vec{\psi} B_{E,i}$. Since (i) the template mesh is constant, (ii) the identity $\mathrm{Var}\big[AZ\big] = A\Sigma_Z A^T$, and (iii) we are interested in per-vertex variance (and not covariance), we have

$$\mathrm{Var}[X_i] = \sum_m \sigma_\beta^2[m]\, B_{S,i}^2[m] + \sum_n \sigma_\psi^2[n]\, B_{E,i}^2[n]\ . \tag{2}$$

The kth component ($k \in 1, 2, 3$) of this variance is denoted by $\sigma_{i,k}^2$.

Law of Total Variance. The skinned vertex is denoted as $v_i' = \sum_{j\in\mathcal{J}} w_{i,j}\, \mathbf{G}_j(\theta, \mathbf{J})\, X_i$, where $\mathcal{J}$ is the set of joints and $w_{i,j}$ are the skinning weights. We have two sources of randomness, (1) the rest-pose vertex X_i, and (2) the pose θ. The law of total variance splits $\mathrm{Var}(v_i')$ into the variance due to the pose and the rotation:

$$\mathrm{Var}(v_i') = \underbrace{\mathbb{E}_X\big[\mathrm{Var}_\theta(v_i' \mid X_i)\big]}_{\text{rotational variance}} + \underbrace{\mathrm{Var}_X\big(\mathbb{E}_\theta[v_i' \mid X_i]\big)}_{\text{shape and expression variance}}\ . \tag{3}$$

Intuitively, the rotational variance quantifies how much a vertex moves when the pose varies for an expected body shape. The shape and expression variance quantifies the movement due to the shape and expression given the expected pose.

Rotational Variance. Let us fix the rest-pose vertex and let the pose vary. Because $\mathbf{G}_j$ is composed along the kinematic chain, a small rotation $\delta\omega_a$ of joint a swings all descendent joints $\mathcal{D}(a)$ rigidly about the mean posed location $\bar{\mathbf{t}}_a$ of that joint. A point at x thus moves by $\delta\omega_a \times (x - \bar{\mathbf{t}}_a)$, a small angle times a lever arm. For the skinned vertex, the lever arm is averaged over the joints that carry it and lie below a, and the displacements of all joints add:

$$v_i' \ \approx\ \sum_{j\in\mathcal{J}} w_{i,j}\, \bar{\mathbf{G}}_j X_i \ +\ \sum_{a\in\mathcal{J}} \delta\omega_a \times \boldsymbol{\ell}_{i,a}\,, \qquad \boldsymbol{\ell}_{i,a} = \sum_{j\in\mathcal{D}(a)} w_{i,j}\,\big(\bar{\mathbf{G}}_j X_i - \bar{\mathbf{t}}_a\big)\,.$$

A joint contributes only if a descendent of the joint skins vertex i. For instance uncertainty travels down the chain, from the shoulder to the hand, but not across it. The lever arm grows with the distance from the joint, so the same angular uncertainty at the shoulder produces a larger positional uncertainty at the fingertip than at the elbow. The perturbations are small, independent, and have per-axis variances $\sigma^2_{\theta_a,x}, \sigma^2_{\theta_a,y}, \sigma^2_{\theta_a,z}$. The variance of the displacement follows from the cross product. Intuitively, a rotation about y moves the point in x by its z lever arm and in z by its x lever arm, and likewise for the other axes. Evaluating the lever arm at the mean vertex, with components $(\ell_{i,a}^{(x)}, \ell_{i,a}^{(y)}, \ell_{i,a}^{(z)})$, the rotational variance is

$$\mathbb{E}_X\big[\mathrm{Var}_\theta(v_i' \,|\, X_i)\big] \approx \sum_{a\in\mathcal{J}} \begin{bmatrix} \sigma^2_{\theta_a,y}\big(\ell_{i,a}^{(z)}\big)^2 + \sigma^2_{\theta_a,z}\big(\ell_{i,a}^{(y)}\big)^2 \\ \sigma^2_{\theta_a,x}\big(\ell_{i,a}^{(z)}\big)^2 + \sigma^2_{\theta_a,z}\big(\ell_{i,a}^{(x)}\big)^2 \\ \sigma^2_{\theta_a,x}\big(\ell_{i,a}^{(y)}\big)^2 + \sigma^2_{\theta_a,y}\big(\ell_{i,a}^{(x)}\big)^2 \end{bmatrix}\ . \tag{4}$$

The full derivation is given in the appendix.

Shape and Expression Variance. Let us now fix the pose at its mean and let the rest-pose offsets vary. The expected posed vertex is $\mathbb{E}_\theta[v_i' \,|\, X_i] = \big(\sum_{j\in\mathcal{J}} w_{i,j}\, \bar{\mathbf{G}}_j\big) X_i$. The offset variance of Eq. 2 is simply rotated into the posed body by the vertex's blended rotation $\mathbf{R}_i^{\text{eff}} = \sum_{j\in\mathcal{J}} w_{i,j}\, \bar{\mathbf{R}}_j$. Using our identity $\mathrm{Var}\big[AZ\big] = A\Sigma_Z A^T$, we have

$$\mathrm{Var}_X\big(\mathbb{E}_\theta[v_i' \,|\, X_i]\big) = \begin{bmatrix} \sum_{k=1}^3 \big(\mathbf{R}^{\text{eff}}_{i,1,k}\big)^2 \sigma^2_{i,k} \\ \sum_{k=1}^3 \big(\mathbf{R}^{\text{eff}}_{i,2,k}\big)^2 \sigma^2_{i,k} \\ \sum_{k=1}^3 \big(\mathbf{R}^{\text{eff}}_{i,3,k}\big)^2 \sigma^2_{i,k} \end{bmatrix}\ . \tag{5}$$

Substituting Eq. 4 and Eq. 5 into Eq. 3 gives the per-vertex variance of the 3DMM formulation.

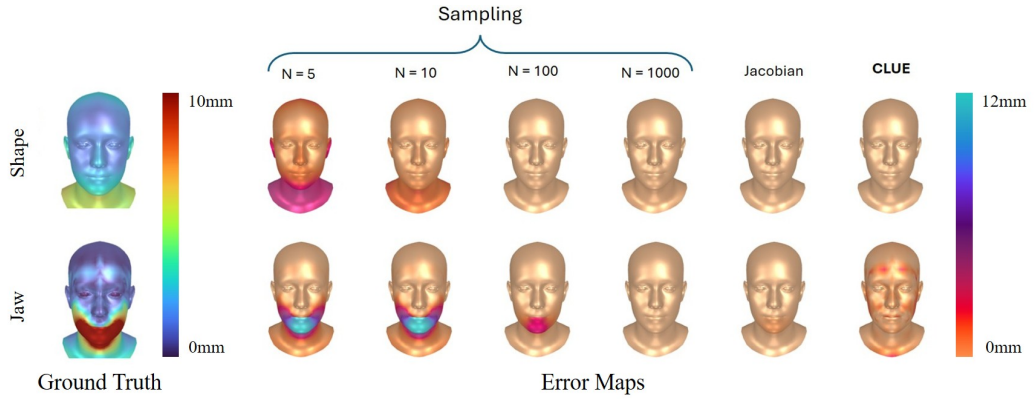


Figure 2: **FLAME Error Map**: We visualize the error maps between the ground truth variance (computed with $N = 10,000$) and the estimated variance from different baselines. We observe that sampling with N=1000, jacobian based error propagation and CLUE have low errors. For the jaw parameter, the error map for CLUE mirrors the forehead pattern in the ground-truth variance, which we attribute to its lack of pose correctives. Additional visualizations in Fig. 7.

The per-vertex variance cleanly disentangles the different sources of uncertainty and combines them through a simple summation. CLUE leverages error propagation without needing to compute the Jacobian. As a result, estimating this quantity is inexpensive in terms of compute time and memory while remaining accurate. In the supplementary, we provide videos to visualize how shape, expression and rotational parameter variance map to the mesh. We also provide pseudo-code in the appendix (Sec. C), and demo code in the supplementary.

3.2 Training: CLUE with SAM 3D Body

Deterministic mesh alignment as a training objective is redundant if deterministic parameter alignment is also used. This is because for both the objectives, minima is achieved when the predicted parameters match the ground truth. However, probabilistic training through the negative log-likelihood introduces new quirks due to the lack of direct supervision over the predicted variance. Without supervision, the predicted variance may take on arbitrary values that minimize the objective incorrectly, thereby affecting convergence (Seitzer et al.; Stirn et al., 2023; Immer et al., 2023; Shukla et al., 2024; 2025). A solution proposed by Shukla et al. (2025) was the use of inductive bias when modeling the predicted variance. CLUE presents one such inductive bias which guides the predicted parameter variance through the geometry of the 3DMM, thereby improving convergence of the parameter variance.

Variance Head. CLUE is architecture agnostic, hence we make no changes to the architecture of SAM 3D Body. We use minimal additional parameters to create a variance head. The variance head takes as input the pose token from SAM 3D Body which is also consumed by the MHR head. In addition, the variance head also takes in the mean parameters output by the MHR head. Both the pose token and the mean parameters are concatenated before being fed to a two layer feedforward network with one hidden layer of width 1024 and ReLU activation. The output of this network is the variance for the shape, expression and rotation parameters, constrained by sigmoid activation but scaled by learnable scalars. These predicted variances are consumed by CLUE.

Losses. By enabling probabilistic mesh alignment, CLUE enables two training objectives: vertex space as well as joint space negative log-likelihood. The latter is enabled since the predicted joints are a linear combination of the predicted vertices, enabling the propagation of variance from vertices to joints. We later perform an ablation study (Tab. 4) to determine the effectiveness of vertex-NLL and joints-NLL. In addition to vertex-NLL and joints-NLL, we also retain parameter-NLL as a way to supervise the convergence of the parameters. We note that all three objectives share the output of the variance head, which undergoes propagation depending on the training objective.

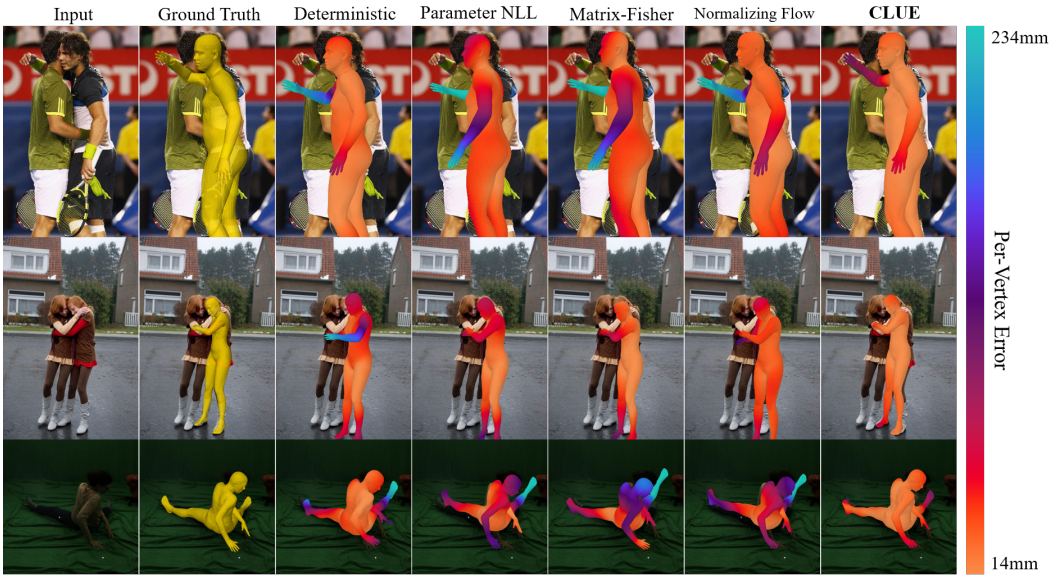


Figure 3: **Per-vertex error maps.** Each row shows the input, the reference mesh (yellow), and the predictions of different baselines, coloured by the per-vertex error. The largest baseline errors sit on limbs that are occluded or having complex poses. Additional visualization in Fig. 8.

4 Experiments

4.1 Statistical Evaluation

The first set of experiments evaluates CLUE as a transformation from parameter-space to the space of meshes, studying accuracy and compute cost. These experiments rely on synthesizing parameter variance, and comparing how CLUE propagates these variances in comparison to sampling with $N = \{5, 10, 100, 1000\}$ and error propagation. We perform these studies across FLAME, SMPL-X and Momentum Human Rig (MHR).

Accuracy (Fig. 2). We compute the *ground truth* variance by simulating parameter variance, drawing $N = 10,000$ samples, and computing per-vertex variance empirically from the resulting meshes. For each baseline, we compute the absolute difference between its variance and that of the ground truth. We visualize these as error maps for different parameter groups: jaw, neck, expression, and shape (or identity). For sampling based methods, as the value of N increases, there is a drop in the error across all parameter groups. We also observe that $N = 1000$ and jacobian error propagation closely approximates the ground truth. While CLUE similarly has low errors across different parameter groups, we highlight the error map *w.r.t* the jaw. We note that the ground truth variance contains an intricate pattern on the forehead, which also comes up in the error map for CLUE. We hypothesize that since CLUE does not model pose correctives, it does not capture intricate pose correctives in the mesh as the jaw joint moves.

Compute. We evaluate on runtime, peak incremental memory, and GFLOPs. CLUE has the fastest runtime, with error propagation via the Jacobian consuming the largest memory. CLUE requires $\approx$ 4MB of additional memory because the tensors needed to compute the variance are already on-device. The low computational costs as well as accuracy of the transformation makes CLUE suitable for on-device uncertainty estimation.

Table 1: **FLAME compute.** Median of 10 trials after 3 warm-ups on one CPU (Intel i7-8700). Additional results in Tab. 6a, 6b, 6c.

Method	Time (ms)	Mem. (MB)	GFLOPs
Sampling N=100	138.7	14.2	1.43
Sampling N=1000	1379.4	140.9	14.26
Jacobian	226.4	410.4	6.01
CLUE (ours)	**3.4**	**4.3**	**0.026**

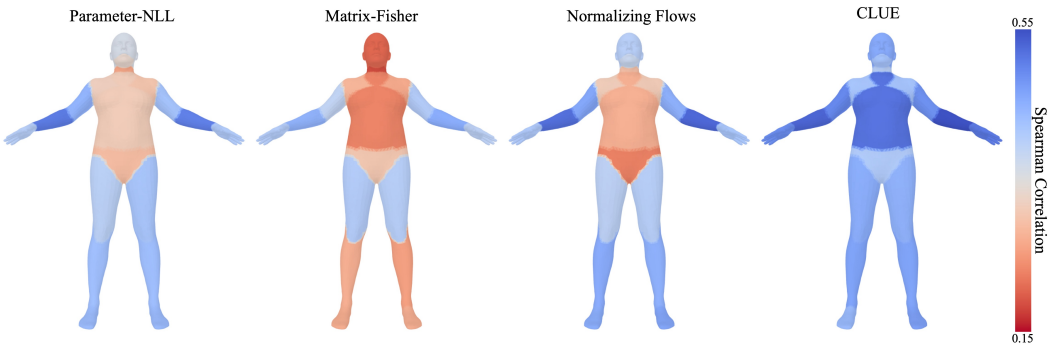


Figure 4: **Correlations across vertex groups.** Per-body-group Spearman correlation between the predicted per-vertex σ and the rigidly aligned vertex error, computed for each of the 19 MHR vertex groups and averaged over the seven pseudo-ground-truth datasets.

Table 2: **3DPW test split (MoCap SMPL).** All metrics in mm (↓). The first row measures SAM 3D Body on our evaluation pipeline. The other rows compare deterministic and probabilistic losses under identical training and evaluation protocol.

	Method	#Training Images	MPJPE	PA-MPJPE	PVE	PA-PVE
	SAM 3D Body (released)*	3.8M (w/ multi-view)	56.0	35.1	65.2	45.1
Same training recipe	Deterministic Training	1.2M (single-view)	63.9	39.6	75.3	51.8
	Parameter NLL	”	58.9	36.9	69.6	48.3
	Matrix-Fisher	”	59.2	37.0	69.8	48.2
	Normalizing Flow	”	59.6	37.4	70.3	48.6
	CLUE (ours)	”	**56.2**	**35.2**	**66.1**	**46.1**

4.2 Probabilistic In-the-wild Human Mesh Recovery

The second set of experiments studies CLUE when used in the training of unimodal in-the-wild human mesh recovery models such as SAM 3D Body. We compare our method with deterministic training, and architecture-agnostic probabilistic losses for uncertainty such as 3DMM Parameter-NLL, Matrix Fisher, and Normalizing Flows (Sengupta et al., 2021c;b; Dwivedi et al., 2024; Kolotouros et al., 2021). We randomly initialize the SAM 3D Body Yang et al. (2026b) architecture with frozen DINOv3 Siméoni et al. (2026) backbone as our HMR model. We describe the losses, datasets and training process in the appendix (Sec: B).

4.2.1 Quantitative Study

3DPW. (Table 2) We observe that all probabilistic losses outperform the deterministic loss. CLUE by introducing inductive bias when modeling the mesh results in improved optimization and is the strongest baseline over all metrics. We also report the evaluation of SAM 3D Body on our pipeline. Our numbers differ from those published due to possible differences with the SAM 3D Body evaluation pipeline which is not public. We also note that CLUE trained on 1.2M single-view images of SA-1B with a frozen DINOv3 and randomly initialized SAM 3D Body architecture is at par with the released checkpoint.

Pseudo Ground Truth datasets. Table 3 scores each baseline against the released SAM 3D Body as ground truth on seven image datasets without motion capture, in MHR space. CLUE is the only probabilistic objective that improves every group of metrics. Figure 4 visualizes the spearman correlation between the vertex error after rigid alignment of the ground truth and predicted mesh. We observe that CLUE has the highest correlation across all groups.

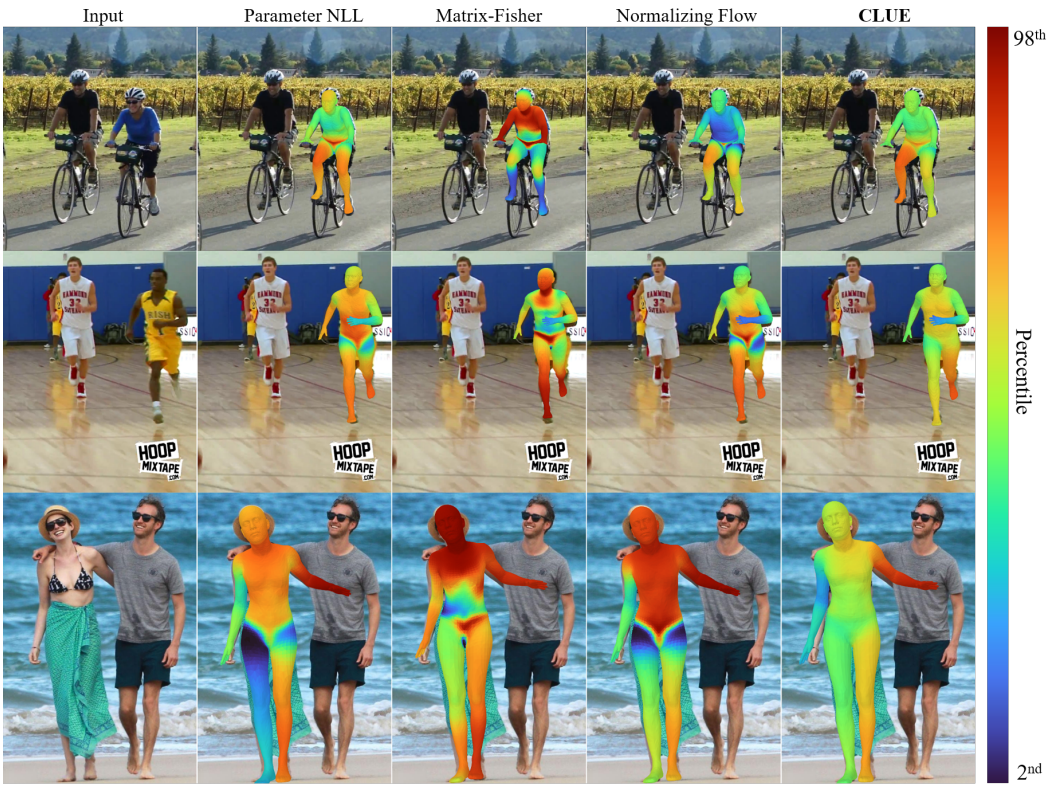


Figure 5: **Uncertainty based data-mining.** Similar to spearman correlation, for each method we normalize a vertex's uncertainty based on its per-vertex percentile for that method. These images were retrieved algorithmically by querieng for images with the highest uncertainty discrepancy between a limb and the body. In contrast to CLUE, the baselines' uncertainty on the same images is body-wide or diffused and lacks fine-grained resolution. Additional visualizations in Fig. 9.

Table 3: **Pseudo-ground-truth evaluation**, *averaged* over seven in-the-wild datasets (H36M, COCO, MPI-INF-3DHP, PoseTrack, MPII, CrowdPose, OCHuman): joint and surface accuracy (mm, ↓), vertex NLL (↓) and Pearson correlation (↑). Per-dataset results are in the appendix (Sec. 5).

	Joints		Surface		Uncertainty	
Method	MPJPE	PA-MPJPE	PVE	PA-PVE	vNLL	ρ
Deterministic	<u>30.9</u>	19.7	38.9	29.0	–	–
Param-NLL	31.9	19.3	<u>37.0</u>	<u>25.3</u>	<u>2.62</u>	<u>0.65</u>
Matrix-Fisher	34.9	20.6	40.3	27.1	2.92	0.61
Normalizing Flow	33.0	<u>19.1</u>	38.2	25.5	3.87	<u>0.65</u>
CLUE (ours)	**29.0**	**18.6**	**31.9**	**22.4**	**2.23**	**0.70**

4.2.2 Qualitative Study

Error Maps. Figure 3 colours each prediction by its per-vertex error against the reference mesh, on one shared scale. We observe two patterns. First, the baselines struggle with limbs that are occluded or have complex poses. CLUE is resistant to occlusion due to the inductive bias of the mesh in modeling the variance and allows for probabilistic reasoning under occlusion. Second, in complex body poses such as floor exercises and splits, the parameter-space baselines spread error over the torso and head as well, whereas CLUE confines the error to the extremities.

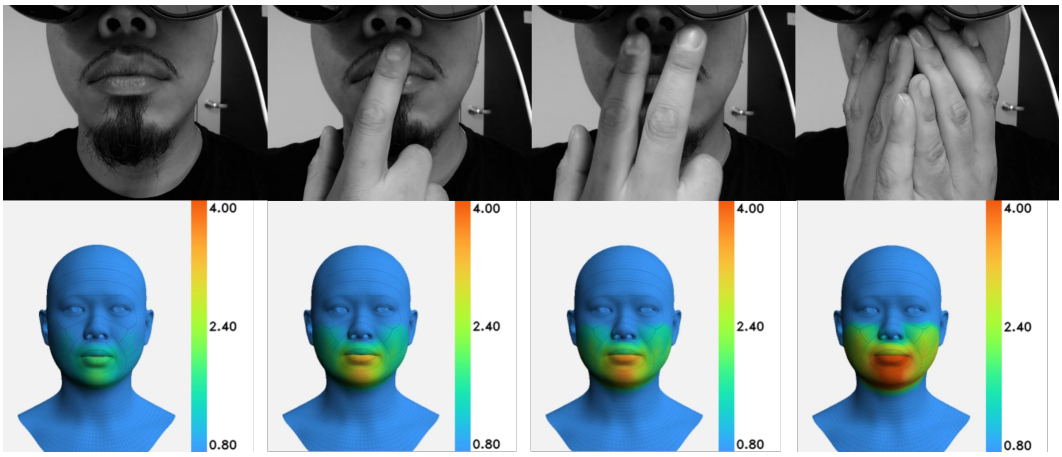


Figure 6: **Uncertainty for downstream decision making.** Top: frames from a headset's egocentric camera with the mouth progressively occluded. Bottom: the tracked 3DMM per-vertex standard deviation (mm). The uncertainty concentrates on the occluded region, rising monotonically with the degree of occlusion. With this signal, a system can suspend or down-weight vision-based face tracking during occlusion, or defer to another modality, instead of tracking blindly.

Uncertainty based data mining. Since CLUE predicts fine-grained uncertainty maps (evidenced by high spearman correlations), we can answer questions that parameter-space uncertainty cannot: *which* vertex groups of the body is the model unsure about? For this study, we look for a limb whose uncertainty stands out from the rest of the body. However, raw uncertainty values are not comparable as they differ in scale between methods and vertices. We therefore convert each vertex's uncertainty into a percentile, i.e. how unusual that value is for that vertex and that method, measured against the SA-1B training set. In Fig. 5 we observe that CLUE correctly distinguishes vertex groups which are occluded or undergo motion blur, while the rest of the body remains confident. The baselines saturate uncertainty across the whole body. While they can say that an image is hard, they cannot localize where.

Uncertainty based decision making. We present a use-case in egocentric face tracking. This study uses an internal dataset and GNM Ploumpis et al. (2026) as the 3DMM. The goal is to regress a 3D mesh that reproduces the lower-face expression of a person from egocentric viewpoints. This setup is typical of VR-headsets such as Apple Vision Pro, Meta Quest Pro, and Galaxy XR, which further impose strict runtime and memory restrictions to enable realtime on-device operation. We employ a similar training protocol as described earlier to train the probabilistic regressor. We observe in Fig. 6 that as the degree of occlusion increases, the uncertainty is localized and scaled correctly. Given this uncertainty signal, a system no longer has to track blindly through an occlusion: it can pause face tracking, lower its weight in the fusion, or hand over to another sensor until the face is visible again.

5 Conclusion

We proposed CLUE, a novel formulation which transforms the parameter-space distribution onto the mesh in closed-form. Through statistical experiments across multiple 3DMMs, we showed that CLUE is an accurate transformation while being computationally efficient. We studied integrating CLUE for in-the-wild human mesh recovery training and inference. Because CLUE carries the predicted parameter variance through the geometry of the 3DMM, it makes likelihoods on the mesh and on the joints available as training objectives. By incorporating this inductive bias, we show that CLUE outperforms deterministic training as well as unimodal 3DMM uncertainty losses. Moreover, our trained model is at par with SAM 3D Body when evaluated on 3DPW with $3\times$ fewer training images. Next, we analyze the variance learnt by CLUE, showing that it has the highest spearman correlation for all vertex groups, and discuss applications of the learnt uncertainty in data mining and downstream decision making. Finally, we share SAM 3D Body (CLUE) to make per-vertex uncertainty a standard output of in-the-wild mesh recovery.

AI USE STATEMENT

In this work, we used generative AI tools for minor edits in writing. This includes finding ways to convey our thoughts in a compact fashion. GenAI tools were also used for grammatical checks or minor corrections. Bulk of the paper was written by humans, for humans. No AI was used in writing this statement. We additionally used GenAI tools for code implementation and documentation. However, there was always a human-in-the-loop to verify code correctness. Finally, GenAI was used to double-check the math in this paper. However, the authors developed the idea and formulated the derivation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide significant implementation details: the pseudo-code for CLUE, and the variance head when using CLUE for in-the-wild human mesh recovery. We also use the open-weight SAM 3D Body model as well as publicly available datasets. Barring our study on downstream decision making, all other experiments can be reproduced with publicly available data, models, and the implementation details provided in the paper.

REFERENCES

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Benjamin Biggs, David Novotny, Sebastien Ehrhardt, Hanbyul Joo, Ben Graham, and Andrea Vedaldi. 3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data. *Advances in neural information processing systems*, 33:20496–20507, 2020.

Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 157–164. 2023.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

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*, pp. 561–578. Springer, 2016.

Hanbyel Cho and Junmo Kim. Generative approach for probabilistic human mesh recovery using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4183–4188, 2023.

Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

Rahul Dey and Vishnu Naresh Boddeti. Generating diverse 3d reconstructions from a single occluded face image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1547–1557, 2022.

Olaf Dünkel, Tim Salzmann, and Florian Pfaff. Normalizing flows on the product space of so (3) manifolds for probabilistic human pose modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2285–2294, 2024.

Sai Kumar Dwivedi, Cordelia Schmid, Hongwei Yi, Michael J Black, and Dimitrios Tzionas. Poco: 3d pose and shape estimation with confidence. In *2024 International Conference on 3D Vision (3DV)*, pp. 85–95. IEEE, 2024.

Bernhard Egger, Andreas Schneider, Clemens Blumer, Sandro Schönborn, and Thomas Vetter. Occlusion-aware 3d morphable face models. 2016.

Bernhard Egger, Sandro Schönborn, Andreas Schneider, Adam Kortylewski, Andreas Morel-Forster, Clemens Blumer, and Thomas Vetter. Occlusion-aware 3d morphable models and an illumination prior for face image analysis. *International Journal of Computer Vision*, 126(12): 1269–1287, 2018.

Aaron Ferguson, Ahmed AA Osman, Berta Bescos, Carsten Stoll, Chris Twigg, Christoph Lassner, David Otte, Eric Vignola, Fabian Prada, Federica Bogo, et al. Mhr: Momentum human rig. *arXiv preprint arXiv:2511.15586*, 2025.

Joint Committee for Guides in Metrology. Evaluation of measurement data—guide to the expression of uncertainty in measurement. *JCGM*, 100(2008):1–116, 2008.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Google. XNNPACK: High-efficiency floating-point neural network inference operators for mobile, server, and web, 2026. URL `https://github.com/google/XNNPACK`. Accessed: 2026-02-24.

Alexander Immer, Emanuele Palumbo, Alexander Marx, and Julia Vogt. Effective bayesian heteroscedastic regression with deep neural networks. *Advances in neural information processing systems*, 36:53996–54019, 2023.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7122–7131, 2018.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf`.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11127–11137, 2021.

Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2252–2261, 2019.

Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11605–11614, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Chunlu Li, Andreas Morel-Forster, Thomas Vetter, Bernhard Egger, and Adam Kortylewski. Robust model-based face reconstruction through weakly-supervised outlier segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 372–381, 2023.

Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Haoshu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 10863–10872. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01112. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Li_CrowdPose_Efficient_Crowded_Scenes_Pose_Estimation_and_a_New_Benchmark_CVPR_2019_paper.html.

Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):1–17, 2017.

Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *European Conference on Computer Vision*, pp. 590–606. Springer, 2022.

Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12939–12948, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. doi: 10.1109/3dv.2017.00064. URL http://gvv.mpi-inf.mpg.de/3dhp_dataset.

D.A. Nix and A.S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 1, pp. 55–60 vol.1, 1994. doi: 10.1109/ICNN.1994.374138.

Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *European Conference on Computer Vision*, pp. 598–613. Springer, 2020.

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 *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10975–10985, 2019.

S. Ploumpis, J. Bednarik, G. Zoss, R. Guseinov, L. Prasso, P. Chandran, O. Boyne, V. Choutas, T. Bolkart, D. Wang, M. Chai, D. Qiu, S. Winberg, G. Rainer, L. Bridgeman, D. Vicini, J. Riviere, Y. Boetzel, A. Koumis, J. Busch, C. Herrera, J. Still, S. Ysebert, P. Lincoln, S. O. Escolano, C. Rhemann, E. Wood, T. Beeler, and S. Zafeiriou. Gnm head: A generative anthropometric model of the human head. 2026. URL https://arxiv.org/abs/2607.23687.

Simon Schaefer, Dorian F Henning, and Stefan Leutenegger. Glopro: Globally-consistent uncertainty-aware 3d human pose estimation & tracking in the wild. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3803–3810. IEEE, 2023.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations (ICLR) 2022*.

Pratheba Selvaraju, Victoria Fernández Abrevaya, Timo Bolkart, Rick Akkerman, Tianyu Ding, Faezeh Amjadi, and Ilya Zharkov. Ofer: Occluded face expression reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26985–26995, 2025.

Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Probabilistic estimation of 3d human shape and pose with a semantic local parametric model. *arXiv preprint arXiv:2111.15404*, 2021a.

Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Hierarchical kinematic probability distributions for 3d human shape and pose estimation from images in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11219–11229, 2021b.

Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Probabilistic 3d human shape and pose estimation from multiple unconstrained images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16094–16104, 2021c.

Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Humaniflow: Ancestor-conditioned normalising flows on so (3) manifolds for human pose and shape distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4779–4789, 2023.

Karthik Shetty, Annette Birkhold, Bernhard Egger, Srikrishna Jaganathan, Norbert Strobel, Markus Kowarschik, and Andreas Maier. Propliks: Probablistic 3d human body pose estimation. *arXiv preprint arXiv:2412.04665*, 2024.

Megh Shukla, Mathieu Salzmann, and Alexandre Alahi. TIC-TAC: A framework for improved covariance estimation in deep heteroscedastic regression. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45244–45257. PMLR, 2024. URL https://proceedings.mlr.press/v235/shukla24a.html.

Megh Shukla, Aziz Shameem, Mathieu Salzmann, and Alexandre Alahi. Towards self-supervised covariance estimation in deep heteroscedastic regression. In *International Conference on Learning Representations*, volume 2025, pp. 93359–93381, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2NlGyqNjns. Featured Certification.

Anastasis Stathopoulos, Ligong Han, and Dimitris Metaxas. Score-guided diffusion for 3d human recovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 906–915, 2024.

Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. Faithful heteroscedastic regression with neural networks. In *International conference on artificial intelligence and statistics*, pp. 5593–5613. PMLR, 2023.

Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

Timing Yang, Sicheng He, Hongyi Jing, Jiawei Yang, Zhijian Liu, Chuhang Zou, and Yue Wang. Fast sam 3d body: Accelerating sam 3d body for real-time full-body human mesh recovery. *arXiv preprint arXiv:2603.15603*, 2026a.

Xitong Yang, Devansh Kukreja, Don Pinkus, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jia-Wei Liu, Nicolás Ugrinovic, Anushka Sagar, et al. Sam 3d body: Robust full-body human mesh recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7209–7219, 2026b.

Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 889–898. IEEE, 2019.

A Derivation Details

We provide a detailed, annotated version of the derivation for convenience.

Assumptions and approximations. CLUE is a first-order error propagation specialised to the skinning equation. We make several assumptions that allow this approximation to be real-time yet accurate in comparison to large-scale sampling and jacobian based error propagation.

1. *Independence and diagonal covariance.* β, ψ and θ are mutually independent with diagonal covariances, and the perturbations of different joints are independent with zero mean. We keep only per-vertex, per-axis marginal variances and not covariances for simplicity.
2. *First order in the pose.* Terms of second or higher order in the pose perturbations are dropped. In particular $\mathbb{E}_\theta[\mathbf{G}_j] = \bar{\mathbf{G}}_j + O(\sigma_\theta^2)$, and we drop the $O(\sigma_\theta^2 \sigma_X^2)$ cross term. This simplicity allows us to have a lower computational cost in relative to full error propagation.
3. *Tangent-space pose variance.* $\Sigma_{\theta_a} = \mathrm{diag}(\sigma^2_{\theta_a,x}, \sigma^2_{\theta_a,y}, \sigma^2_{\theta_a,z})$ is the covariance of the small rotation $\delta\omega_a$ applied on the left of $\exp(\bar{\omega}_a)$ and expressed in world coordinates. Relating it to the covariance of the axis-angle coordinates of joint a would require (i) conjugation by $\bar{\mathbf{R}}_{p(a)}$ and (ii) the left Jacobian $J_l(\bar{\omega}_a)$ of $SO(3)$; we omit both for simplicity. When the variances are learned (Sec. 3.2) the variance head predicts the tangent-space quantity directly, so neither map is needed at training time.
4. *Joint locations at the mean shape.* In the shape and expression term, the joint locations $\mathbf{J}$ (equivalently the bone lengths) and hence $\bar{\mathbf{t}}_j$ are held at the mean shape. This neglects the shape-induced displacement of the joints propagated along the kinematic chain. The neglected term vanishes exactly when the joints above vertex i share its rotation (rest pose, or a rigid transformation of the whole chain), and is otherwise bounded by twice the shape-induced variation of the bone lengths on the chain above vertex i.
5. *No pose correctives.* As stated in Sec. 3; Sec. A.6 shows how they can be added.

A.1 3DMM Formulation.

Rest-pose vertex. Vertex i in the rest pose is the template vertex with additive linear offsets of shape and expression,

$$X_i = \bar{v}_i + \vec{\beta}\, B_{S,i} + \vec{\psi}\, B_{E,i}\,.$$

Kinematic chain. Each joint k has a parent $p(k)$, a rest-pose location $\mathbf{J}_k$, an offset $\mathbf{j}_k = \mathbf{J}_k - \mathbf{J}_{p(k)}$ from its parent, and a local rotation $\exp(\omega_k)$ given by the pose $\theta = \{\omega_k\}_{k\in\mathcal{J}}$, where $\mathcal{J}$ is the set of joints. Let $A(j)$ be the ordered set of joints on the path from the root to j, including j. The global transformation of joint j is the chain product

$$G_j(\theta, \mathbf{J}) = \prod_{k\in A(j)} \begin{bmatrix} \exp(\omega_k) & \mathbf{j}_k \\ 0 & 1 \end{bmatrix} =: \begin{bmatrix} \mathbf{R}_j & \mathbf{t}_j \\ 0 & 1 \end{bmatrix},$$

where $\mathbf{R}_j$ is the global rotation of joint j and $\mathbf{t}_j$ its location in the posed body. The chain can be represented as a recursive relation between a joint and its parent, $\mathbf{R}_j = \mathbf{R}_{p(j)} \exp(\omega_j)$ and $\mathbf{t}_j = \mathbf{R}_{p(j)}\, \mathbf{j}_j + \mathbf{t}_{p(j)}$.

We use two properties. First, ω_a appears in G_j if and only if $a \in A(j)$; we write $\mathcal{D}(a) = \{j : a \in A(j)\}$ for the joints at or below a. Second, each factor is a translation followed by a rotation,

$$\begin{bmatrix} \exp(\omega_k) & \mathbf{j}_k \\ 0 & 1 \end{bmatrix} = T(\mathbf{j}_k)\, \mathrm{Rot}\big(\exp(\omega_k)\big)\,, \qquad T(\mathbf{u}) = \begin{bmatrix} I & \mathbf{u} \\ 0 & 1 \end{bmatrix}, \quad \mathrm{Rot}(\mathbf{R}) = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & 1 \end{bmatrix}.$$

Pivot removal. SMPL applies G_j after removing the rest-pose transformation, $\mathbf{G}_j = G_j(\theta, \mathbf{J})\, G_j(\theta^*, \mathbf{J})^{-1}$. In the rest pose all rotations are the identity, so $G_j(\theta^*, \mathbf{J}) = T(\mathbf{J}_j)$ and

$$\mathbf{G}_j\, v = \mathbf{R}_j\, (v - \mathbf{J}_j) + \mathbf{t}_j\,, \qquad \text{in particular} \qquad \mathbf{G}_j\, \mathbf{J}_j = \mathbf{t}_j\,.$$

Skinning. The posed vertex is a weighted average of the vertex transformed by each joint,

$$v'_i = \sum_{j \in \mathcal{J}} w_{i,j}\, \mathbf{G}_j\, X_i\,, \tag{6}$$

with $w_{i,j}$ the skinning weights, which are non-zero only for a few joints near vertex i.

A.2 Additive Variance.

The shape and expression parameters are assumed independent, with diagonal variances σ_β^2 and σ_ψ^2. Since X_i is linear in these parameters and the template is constant, the variance of a linear map, $\mathrm{Var}[AZ] = A\Sigma_Z A^T$, gives, keeping only the per-vertex variance (the diagonal),

$$\mathrm{Var}[X_i] = \sum_m \sigma_\beta^2[m]\, B_{S,i}^2[m] + \sum_n \sigma_\psi^2[n]\, B_{E,i}^2[n]\,. \tag{7}$$

We denote its kth component by $\sigma_{i,k}^2$, $k \in \{1,2,3\}$. Throughout the derivation we keep only per-vertex variances, never covariances.

A.3 Law of Total Variance.

We have two independent sources of randomness in the posed vertex v'_i: namely the shape and expression parameters, through X_i, and the pose θ, through the transformations $\mathbf{G}_j$. The law of total variance separates them,

$$\mathrm{Var}(v'_i) = \underbrace{\mathbb{E}_X\big[\mathrm{Var}_\theta(v'_i \mid X_i)\big]}_{\text{rotation variance}} + \underbrace{\mathrm{Var}_X\big(\mathbb{E}_\theta[v'_i \mid X_i]\big)}_{\text{shape and expression variance}}\,. \tag{8}$$

The first term asks that for a fixed rest-pose vertex, how much does the vertex move when the pose varies? The second asks whether for the expected pose, how much does the vertex move when the shape and expression vary? We derive the two terms in turn.

A.4 Rotation Variance.

The intuition is to fix the rest-pose vertex X_i and let only the pose vary.

A.4.1 Step 1: Effect of one joint's perturbation *(joint a)* on one transformation (j)

Let $a \in A(j)$. Splitting the chain product of G_j at joint a and using the translate-then-rotate form of the ath factor,

$$G_j = \underbrace{G_{p(a)}\, T(\mathbf{j}_a)}_{H_a}\, \mathrm{Rot}\big(\exp(\omega_a)\big) \prod_{k \in A(j),\ k \text{ below } a} \begin{bmatrix} \exp(\omega_k) & \mathbf{j}_k \\ 0 & 1 \end{bmatrix}, \qquad H_a = \begin{bmatrix} \mathbf{R}_{p(a)} & \mathbf{t}_a \\ 0 & 1 \end{bmatrix},$$

where the block form of H_a follows from the recursion $\mathbf{t}_a = \mathbf{R}_{p(a)}\,\mathbf{j}_a + \mathbf{t}_{p(a)}$. H_a is the rigid transformation that places the frame of joint a in the world: rotation $\mathbf{R}_{p(a)}$, origin at the posed joint location $\mathbf{t}_a$.

Now let us perturb the rotation of joint a by a small rotation, $\exp(\omega_a) \to (I + [\delta_a]_\times)\exp(\omega_a)$ (Rodrigues formula), and substituting it in G_j we have,

$$G_j^{\text{pert}} = \underbrace{H_a\, \mathrm{Rot}\big(I + [\delta_a]_\times\big)\, H_a^{-1}}_{\text{extra factor}}\, G_j\ :$$

the perturbed transformation is the unperturbed one followed by an extra factor. To see what this factor does, follow a point x through its three matrices, from right to left.

(i) H_a^{-1} expresses x in the frame of joint a. The inverse of a rigid transformation subtracts the translation and undoes the rotation,

$$y = H_a^{-1} x = \mathbf{R}_{p(a)}^T\,(x - \mathbf{t}_a)\,.$$

(ii) $\mathrm{Rot}(I + [\delta_a]_\times)$ applies the small rotation in that frame,

$$y' = (I + [\delta_a]_\times)\, y = y + \delta_a \times y\,.$$

(iii) H_a brings the result back to world coordinates,

$$x' = H_a\, y' = \mathbf{R}_{p(a)}\, y' + \mathbf{t}_a = \underbrace{\mathbf{R}_{p(a)}\, y + \mathbf{t}_a}_{=\,x} + \mathbf{R}_{p(a)}\,(\delta_a \times y)\,.$$

Using the property, $\mathbf{R}\,(\mathbf{u}\times\mathbf{v}) = (\mathbf{R}\mathbf{u})\times(\mathbf{R}\mathbf{v})$ and (i), the last term equals $(\mathbf{R}_{p(a)}\delta_a)\times(\mathbf{R}_{p(a)}\, y) = (\mathbf{R}_{p(a)}\delta_a)\times(x - \mathbf{t}_a)$. Writing $\delta\omega_a = \mathbf{R}_{p(a)}\,\delta_a$ for the perturbation expressed in world coordinates, the extra factor acts as

$$x \;\longmapsto\; x + \delta\omega_a \times (x - \mathbf{t}_a)\,.$$

Intuitively, perturbing the rotation of joint a rotates every point carried by the chain below a by the small angle $\delta\omega_a$ about the posed location $\mathbf{t}_a$ of joint a.

From the joint transformation to the vertex. We write $F_a = H_a\,\mathrm{Rot}\big(I + [\delta_a]_\times\big)\,H_a^{-1}$ for the extra factor, so that $G_j^{\text{pert}} = F_a\, G_j$ and $F_a\, x = x + \delta\omega_a \times (x - \mathbf{t}_a)$ for any point x. To accurately locate the vertex, we need to adjust for the pivot. However, pivot removal for G_j^{pert} and G_j is the same since this step does not depend on the pose, perturbed or not. Therefore, $\mathbf{G}_j^{\text{pert}} = F_a\,\mathbf{G}_j$. Hence, F_a acts on the posed vertex $\mathbf{G}_j X_i$ like on any other point. For every joint j at or below a,

$$\mathbf{G}_j^{\text{pert}} X_i = F_a\,(\mathbf{G}_j X_i) = \mathbf{G}_j X_i + \delta\omega_a \times \big(\mathbf{G}_j X_i - \mathbf{t}_a\big)\,, \qquad j \in \mathcal{D}(a)\,.$$

For $j \notin \mathcal{D}(a)$, ω_a does not appear in the chain of G_j, so there is no factor F_a and $\mathbf{G}_j X_i$ is unchanged.

A.4.2 Step 2: Effect of perturbing all joints

Let every joint be perturbed. Each $a \in A(j)$ contributes an extra perturbation of the form $I + (\text{small})$ to G_j. Expanding the product and dropping products of two or more small terms, the displacements of Step 1 add up to,

$$\mathbf{G}_j(\theta + \delta\theta)\, X_i \;\approx\; \bar{\mathbf{G}}_j X_i + \sum_{a \in A(j)} \delta\omega_a \times \big(\bar{\mathbf{G}}_j X_i - \bar{\mathbf{t}}_a\big)\,,$$

where a bar denotes evaluation at the mean pose μ_θ. Substituting into the skinning equation (Eq. 6),

$$v_i' \;\approx\; \sum_{j \in \mathcal{J}} w_{i,j}\, \bar{\mathbf{G}}_j X_i \;+\; \sum_{j \in \mathcal{J}} \sum_{a \in A(j)} w_{i,j}\, \delta\omega_a \times \big(\bar{\mathbf{G}}_j X_i - \bar{\mathbf{t}}_a\big)\,.$$

The double sum runs over the pairs (j, a) with $a \in A(j)$, which is the same set of pairs as $j \in \mathcal{D}(a)$. Exchanging the order of summation therefore gives $\sum_{j\in\mathcal{J}} \sum_{a\in A(j)} = \sum_{a\in\mathcal{J}} \sum_{j\in\mathcal{D}(a)}$, and since $\delta\omega_a$ does not depend on j it factors out of the inner sum

$$v_i' \;\approx\; \sum_{j \in \mathcal{J}} w_{i,j}\, \bar{\mathbf{G}}_j X_i \;+\; \sum_{a \in \mathcal{J}} \delta\omega_a \times \boldsymbol{\ell}_{i,a}\,, \qquad \boldsymbol{\ell}_{i,a} = \sum_{j \in \mathcal{D}(a)} w_{i,j}\, \big(\bar{\mathbf{G}}_j X_i - \bar{\mathbf{t}}_a\big)\,.$$

For each joint a it collects exactly those skinning terms whose joint lies at or below a, the only terms that ω_a affects. Since $\bar{\mathbf{t}}_a$ does not depend on j,

$$\boldsymbol{\ell}_{i,a} = \sum_{j \in \mathcal{D}(a)} w_{i,j}\, \bar{\mathbf{G}}_j X_i \;-\; W_{i,a}\, \bar{\mathbf{t}}_a\,, \qquad W_{i,a} = \sum_{j \in \mathcal{D}(a)} w_{i,j}\,,$$

with $W_{i,a}$ the total skinning weight of vertex i on the joints at or below a. We call $\boldsymbol{\ell}_{i,a}$ the lever arm of vertex i about joint a. It vanishes when no joint in $\mathcal{D}(a)$ skins vertex i, so only the joints on the chains above vertex i contribute.

A.4.3 STEP 3: VARIANCE

The first sum in Step 2 is deterministic whereas the second is the random. Writing $\delta\omega_a \times \boldsymbol{\ell}_{i,a} = -\boldsymbol{\ell}_{i,a} \times \delta\omega_a = -[\boldsymbol{\ell}_{i,a}]_\times \, \delta\omega_a$ and assuming the perturbations of different joints to be independent with zero mean,

$$\mathrm{Var}_\theta(v_i' \mid X_i) = \sum_{a \in \mathcal{J}} [\boldsymbol{\ell}_{i,a}]_\times \, \Sigma_{\theta_a} \, [\boldsymbol{\ell}_{i,a}]_\times^T , \qquad \Sigma_{\theta_a} = \mathrm{diag}\big(\sigma^2_{\theta_a,x},\, \sigma^2_{\theta_a,y},\, \sigma^2_{\theta_a,z}\big) ,$$

where $\sigma^2_{\theta_a,\cdot}$ are the predicted per-axis variances of the pose parameters of joint a. The outer expectation over X_i in Eq. 8 is taken to first order by evaluating the lever arm at the mean rest-pose vertex μ_{X_i},

$$\bar{\boldsymbol{\ell}}_{i,a} = \sum_{j \in \mathcal{D}(a)} w_{i,j} \, \bar{\mathbf{G}}_j \, \mu_{X_i} - W_{i,a} \, \bar{\mathbf{t}}_a = \big(\ell^{(x)}_{i,a},\, \ell^{(y)}_{i,a},\, \ell^{(z)}_{i,a}\big)^T .$$

Writing $[\bar{\boldsymbol{\ell}}_{i,a}]_\times$ as the cross-product matrix of this vector, the rotation variance is the diagonal of $[\bar{\boldsymbol{\ell}}_{i,a}]_\times \Sigma_{\theta_a} [\bar{\boldsymbol{\ell}}_{i,a}]_\times^T$. This gives us

$$\mathbb{E}_X\big[\mathrm{Var}_\theta(v_i' \mid X_i)\big] \approx \sum_{a \in \mathcal{J}} \begin{bmatrix} \sigma^2_{\theta_a,y}\big(\ell^{(z)}_{i,a}\big)^2 + \sigma^2_{\theta_a,z}\big(\ell^{(y)}_{i,a}\big)^2 \\ \sigma^2_{\theta_a,x}\big(\ell^{(z)}_{i,a}\big)^2 + \sigma^2_{\theta_a,z}\big(\ell^{(x)}_{i,a}\big)^2 \\ \sigma^2_{\theta_a,x}\big(\ell^{(y)}_{i,a}\big)^2 + \sigma^2_{\theta_a,y}\big(\ell^{(x)}_{i,a}\big)^2 \end{bmatrix} . \tag{9}$$

A.5 SHAPE AND EXPRESSION VARIANCE.

Now let the pose take its expected value and let only the shape and expression vary. To first order in the pose perturbation (A2), the expected transformation of each joint is the transformation at the mean pose, $\bar{\mathbf{G}}_j$. The expected vertex is then

$$\mathbb{E}_\theta[v_i' \mid X_i] = \Big(\sum_{j \in \mathcal{J}} w_{i,j} \, \bar{\mathbf{G}}_j \Big) X_i \, .$$

We note that X_i is random only in its first three components (the offsets). The joint locations $\mathbf{J}$ and the translations $\bar{\mathbf{t}}_j$ are evaluated at the mean shape and interact only with the homogeneous coordinate of 1 and do not contribute to the variance. What remains is the skinning-blended rotation

$$\mathbf{R}_i^{\mathrm{eff}} = \sum_{j \in \mathcal{J}} w_{i,j} \, \bar{\mathbf{R}}_j$$

applied to the rest-pose offsets. Using $\mathrm{Var}[AZ] = A \Sigma_Z A^T$ with the variance $\sigma^2_{i,k}$ of Eq. 7 and keeping the diagonal, the shape and expression variance is

$$\mathrm{Var}_X\big(\mathbb{E}_\theta[v_i' \mid X_i]\big) = \begin{bmatrix} \sum_{k=1}^3 \big(\mathbf{R}^{\mathrm{eff}}_{i,1,k}\big)^2 \sigma^2_{i,k} \\ \sum_{k=1}^3 \big(\mathbf{R}^{\mathrm{eff}}_{i,2,k}\big)^2 \sigma^2_{i,k} \\ \sum_{k=1}^3 \big(\mathbf{R}^{\mathrm{eff}}_{i,3,k}\big)^2 \sigma^2_{i,k} \end{bmatrix} . \tag{10}$$

Intuitively, the rest-pose offset variance of the vertex is rotated into the posed body by the vertex's own blended rotation.

Finally, substituting Eq. 9 and Eq. 10 into Eq. 8 gives the per-vertex variance

$$\mathrm{Var}(v_i') \approx \sum_{a \in \mathcal{J}} \begin{bmatrix} \sigma^2_{\theta_a,y}\big(\ell^{(z)}_{i,a}\big)^2 + \sigma^2_{\theta_a,z}\big(\ell^{(y)}_{i,a}\big)^2 \\ \sigma^2_{\theta_a,x}\big(\ell^{(z)}_{i,a}\big)^2 + \sigma^2_{\theta_a,z}\big(\ell^{(x)}_{i,a}\big)^2 \\ \sigma^2_{\theta_a,x}\big(\ell^{(y)}_{i,a}\big)^2 + \sigma^2_{\theta_a,y}\big(\ell^{(x)}_{i,a}\big)^2 \end{bmatrix} + \begin{bmatrix} \sum_{k=1}^3 \big(\mathbf{R}^{\mathrm{eff}}_{i,1,k}\big)^2 \sigma^2_{i,k} \\ \sum_{k=1}^3 \big(\mathbf{R}^{\mathrm{eff}}_{i,2,k}\big)^2 \sigma^2_{i,k} \\ \sum_{k=1}^3 \big(\mathbf{R}^{\mathrm{eff}}_{i,3,k}\big)^2 \sigma^2_{i,k} \end{bmatrix} . \tag{11}$$

A.6 POSE CORRECTIVES

We use STAR as the 3DMM to analyse pose correctives. Following STAR's notation convention, P_j is the per-joint pose corrective function which regresses corrective offsets given the joint and its direct neighbors' quaternion values.

$$P_j(\boldsymbol{q}_{ne(j)}, \beta_2; \mathcal{K}_j) = \mathcal{K}_j \left((\boldsymbol{q}_{ne(j)} - \boldsymbol{q}^*_{ne(j)})^T \mid \beta_2 \right)^T$$

Here, the second principal component β_2 as well as $\boldsymbol{q}_{ne(j)}$ are stochastic, while $\boldsymbol{q}^*_{ne(j)}$ and $\mathcal{K}_j$ are deterministic. We note that the variance of this equation follows the familiar $\text{Var}\big[AZ\big] = A\Sigma_Z A^T$ format where A is a deterministic matrix and Z is random.

$$B^j_P(\boldsymbol{q}_{ne(j)}; \mathbf{A}_j, \mathcal{K}_j) = \phi(\mathbf{A}_j) \circ P_j(\boldsymbol{q}_{ne(j)}, \beta_2; \mathcal{K}_j) \tag{12}$$

Since the activation weights $\mathbf{A}_j$ are fixed, $\phi(\mathbf{A}_j)$ is a deterministic vector. The variance of a hadamard product is $\text{Var}\big[A \odot Z\big]_i = a_i^2 \sigma^2_{Z_i}$. This allows us to compute the variance from pose correctives.

B Training Details

Training. All methods share the same training recipe and differ only in the probabilistic modeling. The DINOv3 ViT-H/16+ image encoder is initialised and kept frozen. All other modules of the SAM 3D Body architecture are randomly initialized and trained from scratch. We optimise with AdamW (learning rate 10^{-3}, $\beta = (0.9, 0.999)$, weight decay 0.05), a linear warm-up over the first 1k steps followed by cosine decay to 5% of the peak learning rate, and gradient clipping at norm 1.0. Each step processes 32 images per GPU on 4 GPUs (128 images per step); we train for 30k steps.

Losses. All methods share the same deterministic terms: an L1 loss on the 2D keypoints using the predicted perspective camera, an L1 loss on the 3D joints, and a joint-limit penalty on body joint angles. The methods differ only on how the MHR parameters are supervised. Deterministic training uses L2 loss on all MHR parameters. Parameter-space NLL replaces these deterministic losses on the MHR parameters. POCO applies NLL and RealNVP flow NLL on the joint parameters, keeping shape loss deterministic. Hierarchical Kinematic uses a matrix-Fisher NLL on each supervised joint rotation, with concentrations predicted hierarchically along the kinematic chain, and the Gaussian NLL for lengths and shape. CLUE uses the same losses as parameter-space NLL, but also adds vertex-NLL and joint-NLL. All loss terms have unit weight except the joint-limit penalty (0.01). Following Shukla et al. (2024), the uncertainty head is detached from SAM 3D Body to stabilize joint optimization of the mean and variance heads.

Metrics. We follow standard practices and report (1) joint based metrics such as MPJPE and PA-MPJPE, and (2) vertex based metrics such as PVE and PA-PVE. When the prediction and (pseudo) ground truth are in the same space (MHR space), we additionally report variance based metrics such as vertex-NLL, spearman and pearson correlation. Since baselines measure uncertainty in the parameter space, we draw 100 samples per prediction to compute the mesh space variance.

Datasets. For training, we use the Segment Anything-1B (SA-1B) dataset which consists of 1.2 million+ training images Kirillov et al. (2023). For our evaluation, we use 3DPW von Marcard et al. (2018) as our primary dataset containing ground truth motion capture data in SMPL format. Since our predictions are in MHR space and the ground truth is in SMPL, we use on-the-fly optimzation to map MHR to SMPL as done in Yang et al. (2026b); Ferguson et al. (2025). Additionally, we also use MPII Andriluka et al. (2014), Human3.6M Ionescu et al. (2014), COCO Lin et al. (2014), MPI-INF-3D (Mehta et al., 2017), PoseTrack Andriluka et al. (2018), CrowdPose Li et al. (2019) and OCHuman Zhang et al. (2019). Unlike 3DPW, these datasets do not have 3DMM annotations, and hence we use SAM 3D Body as the pseudo ground truth.

C Pseudo-Code

We present the algorithm for an efficient implementation of Closed-form Local Uncertainty Estimation in Algorithm 1. CLUE requires as input the variance in the shape, expression and rotation parameters. In case of non-availability, the variance for the corresponding parameter group can be set to zero. Since CLUE expresses per-vertex variance through the 3DMM, we also require the shape and expression basis, the mean shape, expression and rotation parameters, the kinematic chain parent map, the joint regressor matrix and the skinning weights.

A couple of computations can be done at initialization. These consist of computing the ancestor matrix and optionally saving an element-wise squared version of the basis. We also save the tensor

Table 4: **CLUE Ablation** on the 3DPW test split. We study the use of joints and vertex-NLL losses in addition to parameter-level loss. The joints-NLL and the vertex-NLL are complementary: when combined, they improve upon or at par with either losses used standalone.

	CLUE LOSSES		JOINTS		SURFACE	
Base	joints-NLL	vertex-NLL	MPJPE	PA-MPJPE	PVE	PA-PVE
Param-NLL		✓	57.4	35.5	67.3	46.3
Param-NLL	✓		57.5	36.0	67.9	47.1
Param-NLL	✓	✓	**56.2**	**35.2**	**66.1**	**46.1**
Normalizing Flow		✓	57.2	35.5	67.1	46.2
Normalizing Flow	✓		56.8	36.1	67.2	47.2
Normalizing Flow	✓	✓	**56.2**	35.5	**66.1**	46.4

formed by fusing the skinning weights with the ancestor matrix. The ancestor matrix ensures that we correctly propagate the variance down the kinematic chains of the 3DMM: the uncertainty of a joint reaches every vertex skinned by a joint below it. From an implementation perspective, we can split the algorithm into six steps.

The first step computes the shape and expression variance which arises from the linear additive offsets to the template mesh, defined in Eq. 7.

The second step computes the shaped mesh and its rest-pose joint locations, converts the axis-angle representation of the mean pose to rotation matrices, and runs forward kinematics along the kinematic chain to obtain the global rotation and translation of every joint. Applying these to the rest-pose joint locations places the joints in the posed body. These posed joint centres are the pivots about which rotational perturbations act.

The third step transforms the vertices and computes their offset with respect to the pivots defined above. We need to compute offsets *w.r.t* the pivot since rotations are defined around it. To make this set of computations efficient, we couple the vertex and its per-joint transformation into a single tensor, eliminating for loops. Next, with the help of broadcasting, we efficiently compute the offsets of our vertex-joint coupling *w.r.t* each pivot. However, this introduces an additional dimension for the joint due to the use of pairwise computations. As a result, we reduce over this additional dimension for the joints using our previously computed fused weights-and-ancestor tensor. This brings us back to the space representing the coupling of vertex and joints, and yields the lever arm of every vertex about every joint, $\bar{\boldsymbol{\ell}}_{v,j}$ in the derivation above.

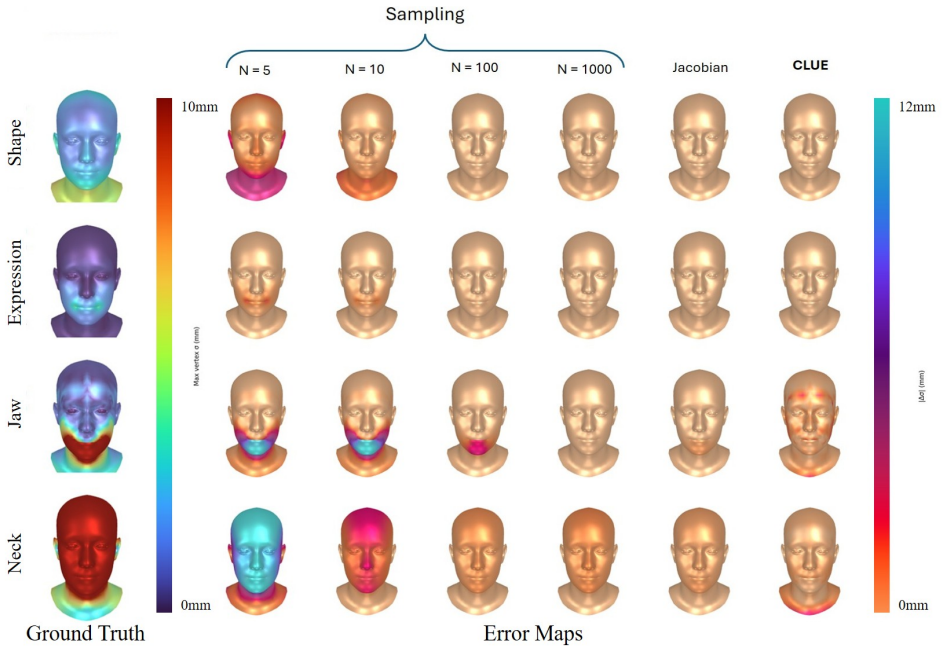
The fourth step computes the rotational variance from the lever arms and the per-axis pose variances, following Eq. 9.

The fifth step rotates the shape and expression variance of the first step into the posed body using the LBS-blended rotation of each vertex, following Eq. 10.

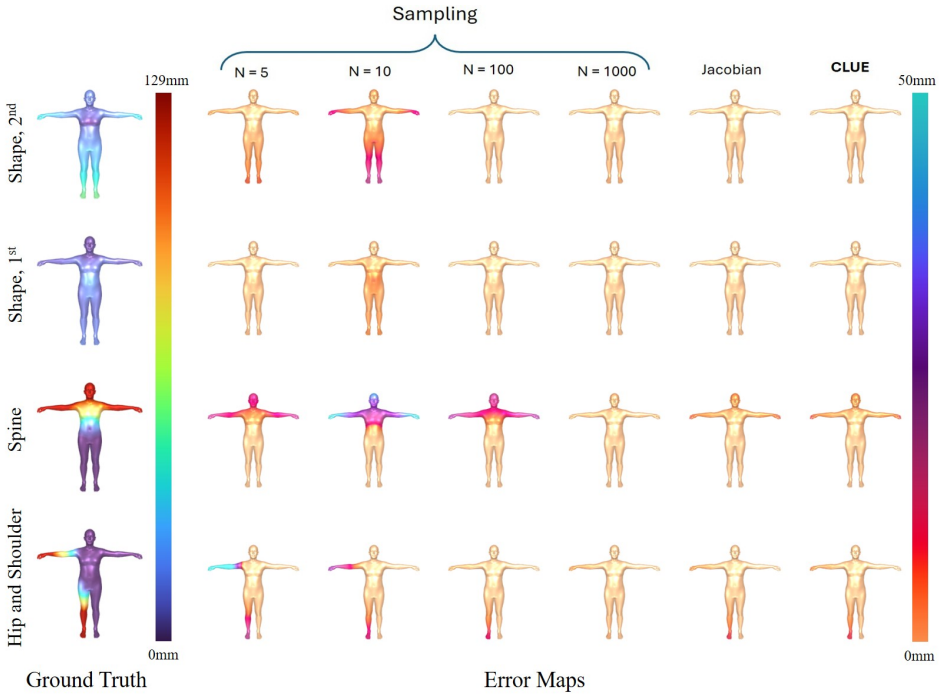
Finally, the sixth step combines the two variances into a single per-vertex quantity following Eq. 11.

D FULL RESULTS

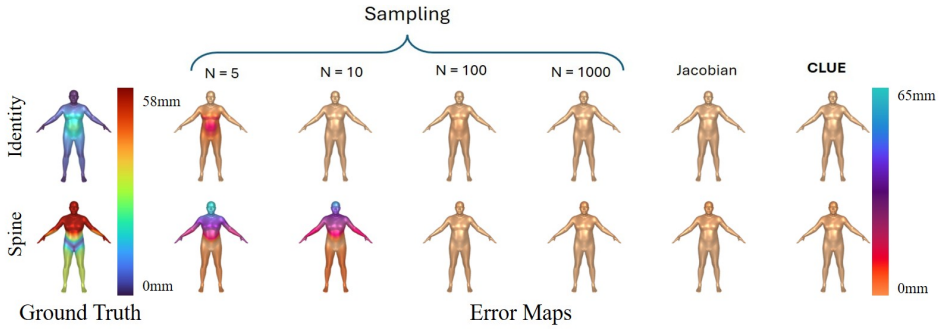
We gather our results and visualizations in this section.



(a) FLAME



(b) SMPLX



(c) MHR

Figure 7: Complete error maps for FLAME, SMPLX, MHR

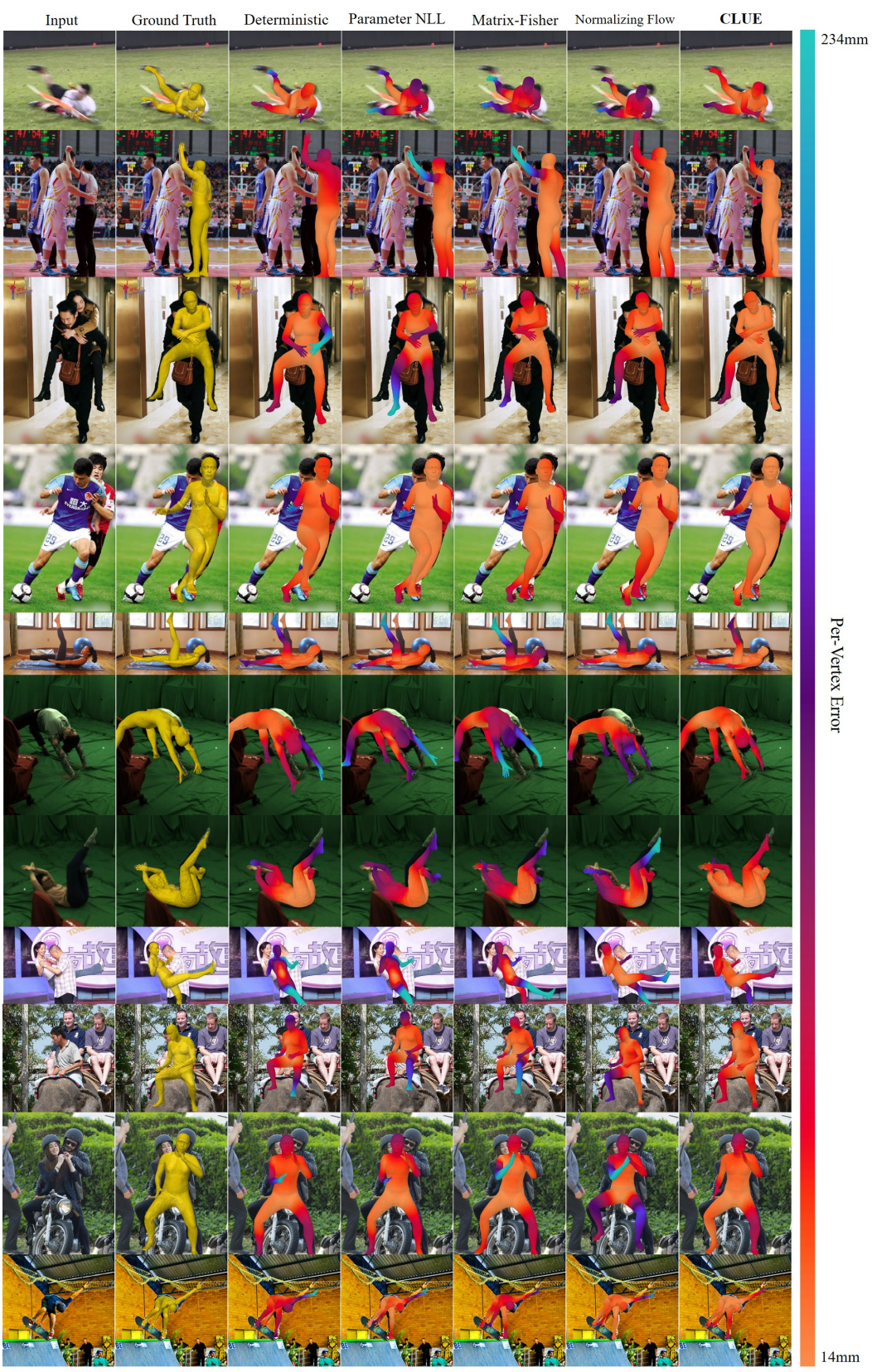


Figure 8: Error Maps for in-the-wild Human Mesh Recovery. We show that CLUE has higher accuracy when dealing with occlusions or complex poses.

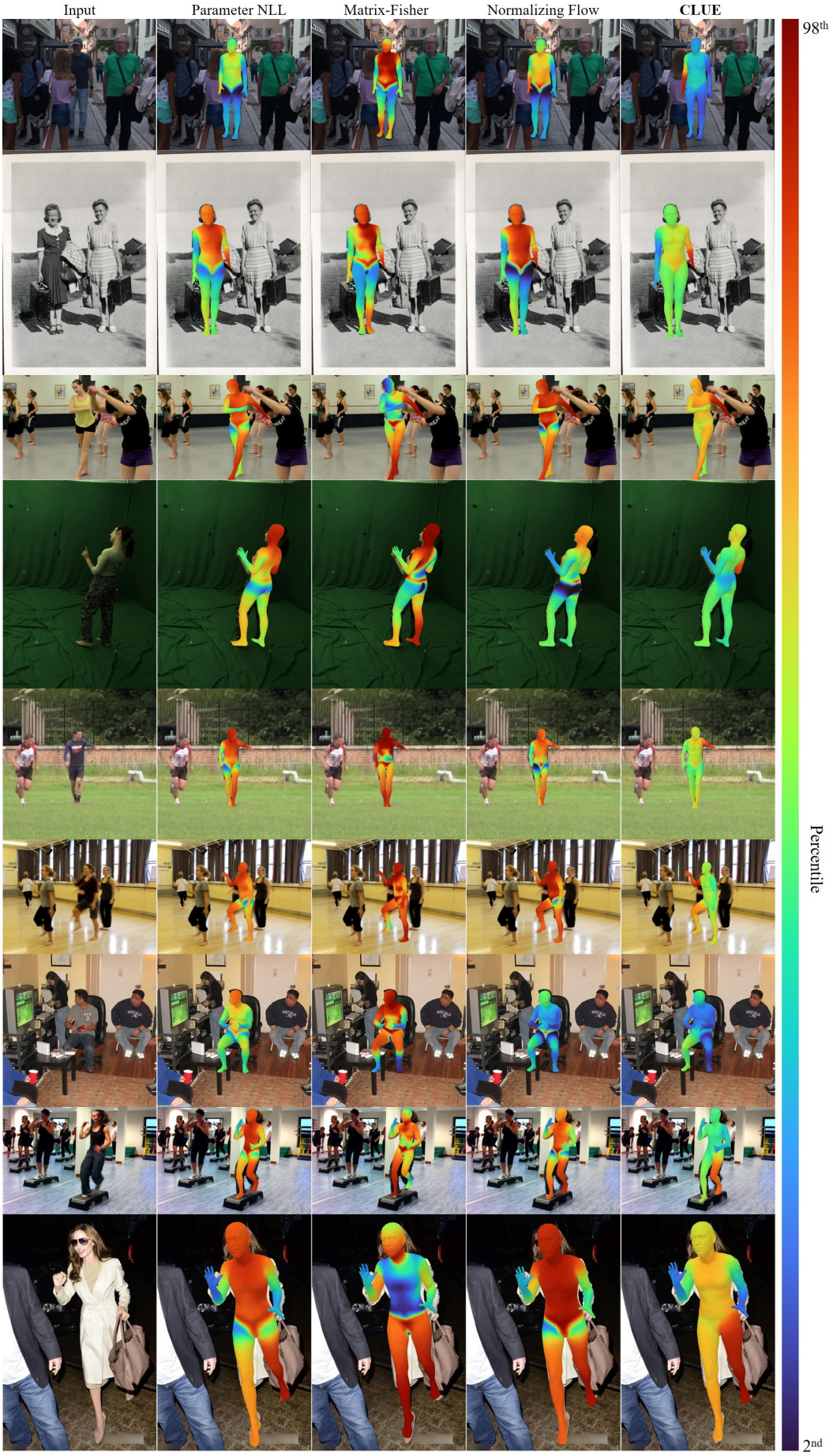


Figure 9: Uncertainty based data-mining.

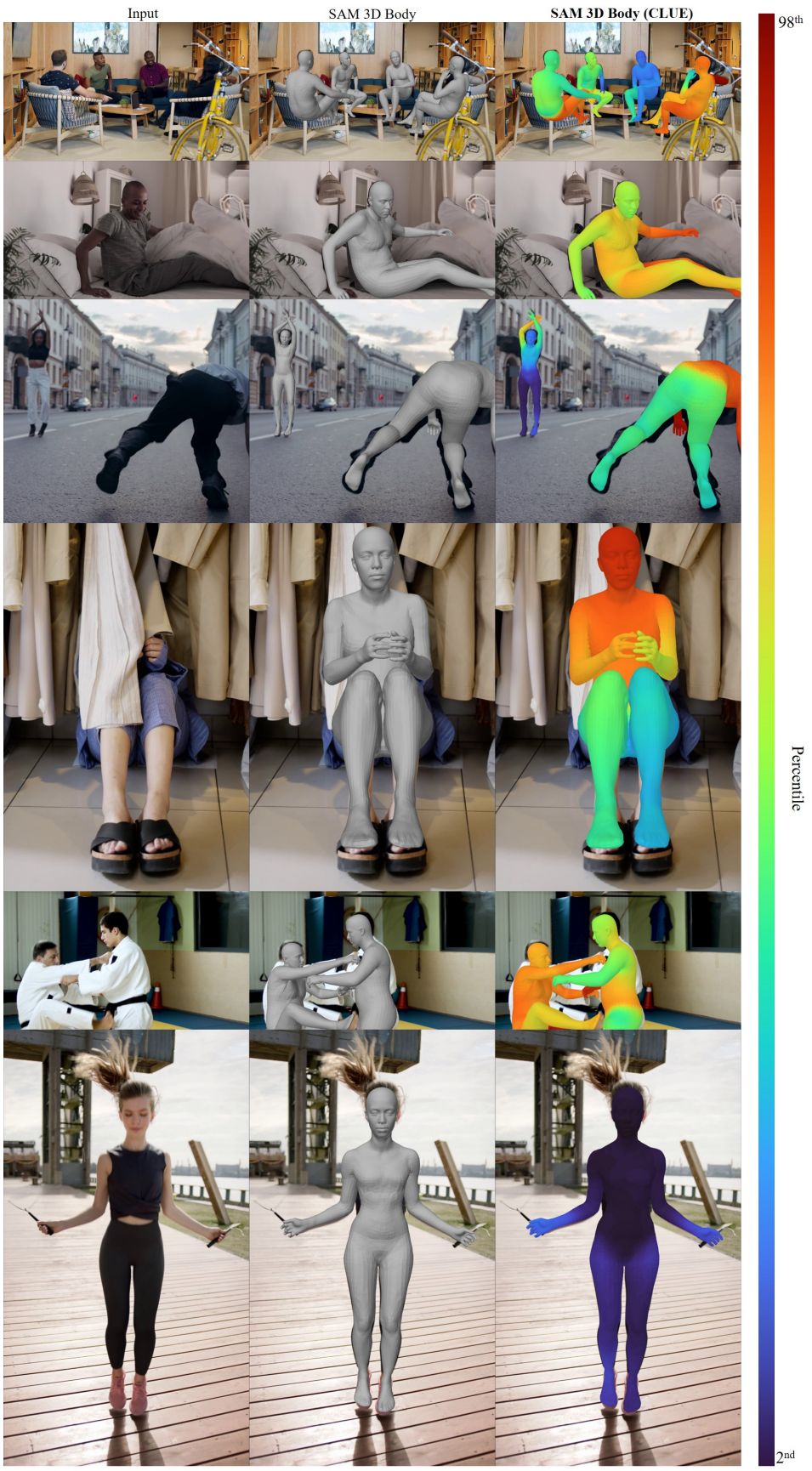


Figure 10: Augmenting SAM 3D Body with probabilistic reasoning using CLUE (Part 1).

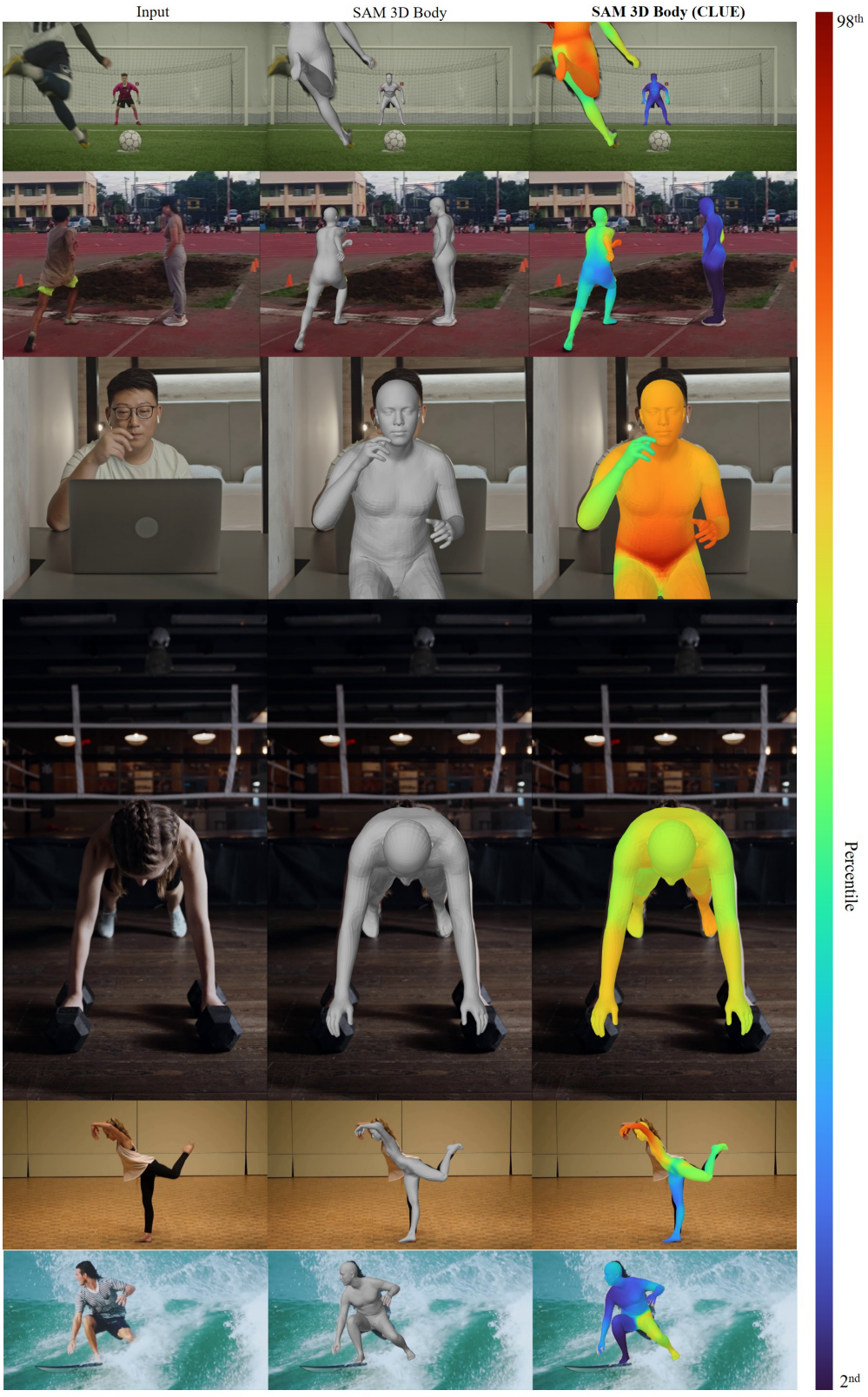


Figure 11: Augmenting SAM 3D Body with probabilistic reasoning using CLUE (Part 2).

Table 5: **Pseudo-ground-truth evaluation on seven in-the-wild datasets**: (a) joint accuracy, (b) surface accuracy (mm, ↓), and (c) uncertainty quality — per-vertex NLL (↓) and the Pearson correlation between predicted σ and vertex error (↑).

(a) Joints: MPJPE / PA-MPJPE

	H36M		COCO		MPI-3D		PoseTrack		MPII		CrowdPose		OCHuman		*Average*	
Deterministic	19.4	13.8	27.1	16.8	27.3	18.2	28.3	18.3	33.2	21.2	35.2	21.5	45.6	28.3	30.9	19.7
Param-NLL	20.6	13.6	28.5	16.9	29.9	17.9	28.9	18.1	33.3	20.6	36.3	21.3	46.0	27.0	31.9	19.3
Matrix-Fisher	24.8	15.0	30.9	18.1	34.0	19.9	32.6	19.4	36.8	21.7	37.1	21.6	47.7	28.3	34.9	20.6
Normalizing Flow	23.2	14.2	30.0	16.7	31.4	18.3	30.6	17.9	33.2	**19.1**	36.1	20.6	46.1	26.7	33.0	19.1
CLUE (ours)	**18.9**	**13.2**	**26.5**	**16.2**	**26.3**	**17.4**	**27.2**	**17.7**	**30.5**	19.4	**33.0**	**20.2**	**40.4**	**25.7**	**29.0**	**18.6**

(b) Surface: PVE / PA-PVE

	H36M		COCO		MPI-3D		PoseTrack		MPII		CrowdPose		OCHuman		*Average*	
Deterministic	28.4	23.7	34.6	25.7	35.9	27.3	35.9	27.0	39.3	29.2	41.7	29.8	56.5	40.0	38.9	29.0
Param-NLL	25.7	18.7	32.7	22.2	34.6	23.1	34.0	24.0	37.3	26.1	40.3	27.1	53.9	35.5	37.0	25.3
Matrix-Fisher	29.2	21.1	36.1	24.0	39.5	25.8	37.7	25.8	41.5	27.8	42.0	27.8	55.9	37.3	40.3	27.1
Normalizing Flow	28.3	20.2	34.6	22.7	36.9	24.4	35.5	24.2	37.8	25.5	40.5	26.4	53.8	35.1	38.2	25.5
CLUE (ours)	**21.6**	**17.1**	**28.8**	**19.3**	**29.5**	**21.1**	**30.1**	**21.2**	**32.1**	**22.4**	**35.2**	**24.0**	**45.7**	**31.9**	**31.9**	**22.4**

(c) Uncertainty: vertex-NLL / σ–error correlation

	H36M		COCO		MPI-3D		PoseTrack		MPII		CrowdPose		OCHuman		*Average*	
Deterministic	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Param-NLL	2.33	0.61	2.59	0.58	2.66	0.65	2.57	0.65	2.67	0.66	2.62	0.70	2.90	0.71	2.62	0.65
Matrix-Fisher	2.81	0.51	2.86	0.57	2.90	0.60	2.92	0.63	2.94	0.66	2.94	0.61	3.08	0.67	2.92	0.61
Normalizing Flow	3.81	0.64	4.17	0.59	4.05	0.68	3.86	0.63	3.83	0.69	3.70	0.67	3.69	0.69	3.87	0.65
CLUE (ours)	**1.94**	**0.67**	**2.10**	**0.65**	**2.16**	**0.69**	**2.23**	**0.68**	**2.23**	**0.74**	**2.29**	**0.72**	**2.63**	**0.74**	**2.23**	**0.70**

Table 6: **Compute cost across body models.** Median of 10 trials after 3 warm-ups on one CPU (Intel i7-8700).

Method	Time (ms)	Mem. (MB)	GFLOPs
Sampling N=100	141.6	14.2	1.43
Sampling N=1000	1404.7	140.9	14.33
Jacobian	228.6	410.4	6.08
CLUE (ours)	**3.5**	**4.3**	**0.027**

(a) FLAME

Method	Time (ms)	Mem. (MB)	GFLOPs
Sampling N=100	368.5	29.5	5.42
Sampling N=1000	3733.6	293.6	54.20
Jacobian	177.0	383.6	10.23
CLUE (ours)	**4.0**	**8.3**	**0.012**

(b) SMPL-X

Method	Time (ms)	Mem. (MB)	GFLOPs
Sampling N=100	681.7	44.5	1.71
Sampling N=1000	6802.4	442.9	17.09
Jacobian	3272.2	1984.4	6.40
CLUE (ours)	**10.9**	**16.9**	**0.041**

(c) MHR

Algorithm 1 Efficient CLUE

Require: $\boldsymbol{\sigma}_\beta^2 \in \mathbb{R}^{B\times C}$: shape and expression parameter variances *concatenated*; $\boldsymbol{\sigma}_\theta^2 \in \mathbb{R}^{B\times J\times 3}$: pose parameter variances; $\overline{v} \in \mathbb{R}^{V\times 3}$: template mesh; $\mathbf{B} \in \mathbb{R}^{V\times 3\times C}$: blendshape basis (shape $\cup$ expression); $\mathcal{R} \in \mathbb{R}^{J\times V}$: joint regressor; $W \in \mathbb{R}^{V\times J}$: LBS weights; $\vec{\beta} \in \mathbb{R}^{B\times C}$: mean shape and expression coefficients concatenated; $\vec{\theta} \in \mathbb{R}^{B\times J\times 3}$: mean pose parameters (axis-angle); π: parent map

Ensure: $\mathrm{Var}(v_i') \in \mathbb{R}^{B\times V\times 3}$: per-vertex variance

— Precompute once at initialisation to reduce runtime —

Build ancestor matrix $P \in \{0,1\}^{J\times J}$:
for $k = 0, \ldots, J{-}1$ **do**
  $j \leftarrow k$
  No parent is represented as -1
  while $j \geq 0$ **do**
    $P_{k,j} \leftarrow 1; \quad j \leftarrow \pi(j)$
  end while
end for

$\mathbf{B}^2 \leftarrow \mathbf{B} \odot \mathbf{B}$ ▷ $(V \times 3 \times C)$, element-wise square
$\widetilde{W} \leftarrow W[:, :, \mathrm{None}] \cdot P[\mathrm{None}, :, :]$ ▷ Fusing skinning weights and ancestor matrix

— Per-forward pass —

Step 1: Shape / expression variance ▷ Eq. 7
$(\Sigma_X)_{b,v,3} \leftarrow \sum_m \sigma_\beta^2[b,m]\, \mathbf{B}^2_{v,3,m}$ ▷ shape $(B \times V \times 3)$

Step 2: Forward kinematics
Compute joint centres which will act as rotational pivots
$\hat{v} \leftarrow \overline{v} + \mathbf{B}\,\vec{\beta}$ ▷ $(B \times V \times 3)$, shaped mesh
$\mathbf{J} \leftarrow \mathcal{R}\,\hat{v}$ ▷ $(B \times J \times 3)$, joint locations
$\{\mathbf{R}_j, \mathbf{t}_j\}_{j=1}^J \leftarrow \mathrm{FK}(\vec{\theta}, \mathbf{J}, \pi)$ ▷ rotation $\mathbf{R}_j \in \mathbb{R}^{3\times 3}$ and translation $\mathbf{t}_j \in \mathbb{R}^3$ of the pivot-removed transformation $\mathbf{G}_j$
$\mathbf{T}_j \leftarrow \mathbf{R}_j\,\mathbf{J}_j + \mathbf{t}_j \quad \forall j$ ▷ $(B \times J \times 3)$, posed joint centres $\mathbf{G}_j\mathbf{J}_j$

Step 3: Vertex transformation
Modelling effect of different joint transformations on the vertex
$\tilde{v} \leftarrow \mathbf{R}\,\hat{v} + \mathbf{t}$ ▷ `bjpq,bvq->bvjp`; $(B \times V \times J \times 3)$, $\mathbf{G}_j\hat{v}$
Broadcast helps in parallelizing offset calculation w.r.t pivot
$r \leftarrow \tilde{v}[:, :, :, \mathrm{None}, :] - \mathbf{T}[:, \mathrm{None}, \mathrm{None}, :, :]$ ▷ rotation around pivot; $(B \times V \times K \times J \times 3)$
Reducing this w.r.t fused skinning weights + ancestor matrix
$\mathbf{r}_{b,v,j,3} \leftarrow \sum_k \tilde{W} \cdot r$ ▷ `vkj,bvkjd->bvjd`; $(B \times V \times J \times 3)$, lever arm $\bar{\ell}_{v,j}$ about joint j

Step 4: Rotational variance ▷ Eq. 9
$a_j,\, b_j,\, c_j \leftarrow \mathbf{r}_{v,j}^{(x)},\, \mathbf{r}_{v,j}^{(y)},\, \mathbf{r}_{v,j}^{(z)}$ ▷ $(B \times V \times J)$ each
$\sigma_{j,1},\, \sigma_{j,2},\, \sigma_{j,3} \leftarrow \boldsymbol{\sigma}^2_{\theta,j,x},\, \boldsymbol{\sigma}^2_{\theta,j,y},\, \boldsymbol{\sigma}^2_{\theta,j,z}$ ▷ $(B \times J)$ each

$$\mathrm{Var}_\theta(v') \leftarrow \sum_{j=1}^{J} \begin{bmatrix} \sigma_{j,2}\, c_j^2 + \sigma_{j,3}\, b_j^2 \\ \sigma_{j,1}\, c_j^2 + \sigma_{j,3}\, a_j^2 \\ \sigma_{j,1}\, b_j^2 + \sigma_{j,2}\, a_j^2 \end{bmatrix} \quad \triangleright\ (B \times V \times 3)$$

Step 5: Rotated shape / expression variance ▷ Eq. 10
$\mathbf{R}_v^{\mathrm{eff}} \leftarrow \sum_{j=1}^{J} w_{v,j}\,\mathbf{R}_j$ ▷ $(B \times V \times 3 \times 3)$, LBS-blended rotation

$$\mathrm{Var}_\beta(v')_p \leftarrow \sum_{q=1}^{3} \left(\mathbf{R}^{\mathrm{eff}}_{v,p,q}\right)^2 (\Sigma_X)_{v,q} \quad \forall p \in \{1,2,3\} \quad \triangleright\ (B \times V \times 3)$$

Step 6: Total variance ▷ Eq. 11
return $\mathrm{Var}(v') \leftarrow \mathrm{Var}_\theta(v') + \mathrm{Var}_\beta(v')$