A Probabilistic Attention Model with Occlusion-aware Texture Regression for 3D Hand Reconstruction from a Single RGB Image (Supplementary)

Zheheng Jiang[1] Hossein Rahmani[1] Sue Black[2] Bryan M. Williams[1]
[1]Lancaster University [2]St John's College of the University of Oxford
{z.jiang11,h.rahmani,b.williams6}@lancaster.ac.uk, sue.black@sjc.ox.ac.uk

Derivation of equation (2)

$$
\begin{aligned}
\underset{\delta,\theta}{\operatorname{argmax}} \ln \mathcal{L}(\delta,\theta) &= \underset{\delta,\theta}{\operatorname{argmax}} \ln \prod_i P\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \delta\right) \\
&= \underset{\delta,\theta}{\operatorname{argmax}} \sum_i \ln P\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \delta\right) \\
&= \underset{\delta,\theta}{\operatorname{argmax}} \sum_i \ln \left(Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right) \frac{P\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \delta\right)}{Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right)} \right) \\
&= \underset{\delta,\theta}{\operatorname{argmax}} \sum_i \left(\ln Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right) + \ln \frac{P\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \delta\right)}{Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right)} \right) \\
&= \underset{\delta,\theta}{\operatorname{argmin}} \sum_i \left(-\ln Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right) - \ln \frac{P\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \delta\right)}{Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right)} \right) \\
&= \underset{\delta,\theta}{\operatorname{argmin}} \sum_i \left(-\ln Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right) + \ln \frac{Q\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \theta\right)}{P\left(J_{2D}^i, J_{3D}^i, V_{3D}^i, C^i | I^i; \delta\right)} \right)
\end{aligned} \tag{1}
$$

Here, Q relates to the estimation of the approximate probability distribution and P to prior net. The loss function for the **mesh vertices** in equation (3) is derived as follows, by maximising $P\left(V_{3D}^i | I^i; \delta\right)$. In line with equation (2), we have:

$$
\underset{\delta,\phi}{\operatorname{argmax}} \ln \prod_i P\left(V_{3D}^i | I^i; \delta\right) = \underset{\delta,\phi}{\operatorname{argmin}} \sum_i \left(-\ln Q\left(V_{3D}^i | I^i; \phi\right) + \ln \frac{Q\left(V_{3D}^i | I^i; \phi\right)}{P\left(V_{3D}^i | I^i; \delta\right)} \right). \tag{2}
$$

For supervised training, we introduce the ground truth into our probabilistic model. In the following equations, we use the caret to mark ground truth. By introducing ground truth, we have:

$$
\underset{\delta,\phi}{\operatorname{argmax}} \ln \prod_i P\left(V_{3D}^i | I^i; \delta\right) = \underset{\delta,\phi}{\operatorname{argmin}} \sum_i \left(-\ln Q\left(\bar{V}_{3D}^i | I^i; \phi\right) + \ln \frac{Q\left(V_{3D}^i | I^i; \phi\right)}{P\left(V_{3D}^i | I^i; \delta\right)} \right). \tag{3}
$$

Derivation of the loss function equation (3) for the mesh vertices

Considering the above components in turn, we assume that $\bar{V}_{3D}^{i,m} \sim \mathcal{N}(\mu_\phi^m, (\sigma_\phi^m)^2)$ and we have

$$
\begin{aligned}
\ln Q\left(\bar{V}_{3D}^{i}|I^{i};\phi\right) &= \ln\left(\prod_{m}\left(\frac{1}{\sigma_{\phi}^{m}\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{\bar{V}_{3D}^{i,m}-\mu_{\phi}^{m}}{\sigma_{\phi}^{m}}\right)^{2}\right)\right)\right)\\
&= \sum_{m}\left[\ln\frac{1}{\sigma_{\phi}^{m}\sqrt{2\pi}}+\ln\exp\left(-\frac{1}{2}\left(\frac{\bar{V}_{3D}^{i,m}-\mu_{\phi}^{m}}{\sigma_{\phi}^{m}}\right)^{2}\right)\right]\\
&= \sum_{m}\left[-\frac{1}{2}\left(\frac{\bar{V}_{3D}^{i,m}-\mu_{\phi}^{m}}{\sigma_{\phi}^{m}}\right)^{2}-\ln\left(\sqrt{2\pi}\sigma_{\phi}^{m}\right)\right]
\end{aligned}
\tag{4}
$$

and

$$
\begin{aligned}
\operatorname*{argmin}_{\delta,\phi}\ln\frac{Q\left(V_{3D}^{i}|I^{i};\phi\right)}{P\left(V_{3D}^{i}|I^{i};\delta\right)} &= \operatorname*{argmin}_{\delta,\phi}D_{KL}\left[Q\left(V_{3D}^{i}|I^{i};\phi\right)\|P\left(V_{3D}^{i}|I^{i};\delta\right)\right]\\
&= \operatorname*{argmin}_{\delta,\phi}\frac{1}{2}\left[\ln\prod_{m}\frac{\sigma_{v}^{m}}{\sigma_{\phi}^{m}}-d+\sum_{m}\frac{\sigma_{\phi}^{m}+\left(\mu_{v}^{m}-\mu_{\phi}^{m}\right)^{2}}{\sigma_{v}^{m}}\right].
\end{aligned}
\tag{5}
$$

Combining these, our loss function becomes:

$$
\mathcal{L}_{V_{3D}} = \sum_{m}\left[\frac{1}{2}\left(\frac{\bar{V}_{3D}^{i,m}-\mu_{\phi}^{m}}{\sigma_{\phi}^{m}}\right)^{2}+\ln\left(\sqrt{2\pi}\sigma_{\phi}^{m}\right)\right]+\frac{1}{2}\left[\ln\prod_{m}\frac{\sigma_{v}^{m}}{\sigma_{\phi}^{m}}-d+\sum_{m}\frac{\sigma_{\phi}^{m}+\left(\mu_{v}^{m}-\mu_{\phi}^{m}\right)^{2}}{\sigma_{v}^{m}}\right]
\tag{6}
$$

Derivation of the loss function equation (4) for the camera parameters

The loss function for the **camera parameters** in equation (4) is derived as follows:

$$
\begin{aligned}
\operatorname*{argmax}_{\delta,\gamma}\ln\prod_{i}P\left(C^{i}|I^{i};\delta\right) &= \operatorname*{argmin}_{\delta,\gamma}\sum_{i}-\ln Q\left(C^{i}|I^{i};\gamma\right)+\ln\frac{Q\left(C^{i}|I^{i};\gamma\right)}{P\left(C^{i}|I^{i};\delta\right)}\\
&= \operatorname*{argmin}_{\delta,\gamma}\sum_{i}-\ln Q\left(\bar{C}^{i}|I^{i};\gamma\right)+\ln\frac{Q\left(C^{i}|I^{i};\gamma\right)}{P\left(\bar{C}^{i}|I^{i};\delta\right)}
\end{aligned}
\tag{7}
$$

where $Q\left(C|J_{2D}^{i},I^{i};\gamma\right)$ and $P\left(\bar{C}|J_{2D}^{i},I^{i};\delta\right)$ are subject to Gaussian distributions $\mathcal{N}_{\gamma}\left(\mu_{\gamma},diag\left(\mathbf{1}\right)\right)$ and $\mathcal{N}\left(\bar{C}^{i},diag\left(\mathbf{1}\right)\right)$. $\bar{C}^{i}$ is the camera ground truth. Since we do not have a pre-trained priornet of the camera model, we create the camera priornet by using the ground truth as the mean and the identity matrix as the variance. We have

$$
\begin{aligned}
\ln Q\left(\bar{C}^{i}|I^{i};\delta\right) &= \ln\prod_{m}\left(\frac{1}{\mathbf{1}^{m}\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{\bar{C}^{i,m}-\mu_{\gamma}^{m}}{\mathbf{1}^{m}}\right)^{2}\right)\right)\\
&= \sum_{m}\ln\frac{1}{\sqrt{2\pi}}+\sum_{m}\ln\exp\left(-\frac{1}{2}\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}\right)\\
&= -\sum_{m}\left[\ln\sqrt{2\pi}+\frac{1}{2}\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}\right]
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
\ln\frac{Q\left(C^{i}|I^{i};\gamma\right)}{P\left(\bar{C}^{i}|I^{i};\delta\right)} &= \frac{1}{2}\left[\ln\prod_{m}\frac{\mathbf{1}^{m}}{\mathbf{1}^{m}}-d+\sum_{m}\frac{\mathbf{1}^{m}+\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}}{\mathbf{1}^{m}}\right]\\
&= \frac{1}{2}\left[0-d+d+\sum_{m}\frac{\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}}{\mathbf{1}^{m}}\right]\\
&= \frac{1}{2}\sum_{m}\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}
\end{aligned}
\tag{9}
$$

Combining these, we have:

$$
\begin{aligned}
\operatorname*{argmin}_{\delta,\gamma}\sum_{i}-\ln Q\left(\bar{C}^{i}|I^{i};\gamma\right)+\ln\frac{Q\left(C^{i}|I^{i};\gamma\right)}{P\left(\bar{C}^{i}|I^{i};\delta\right)} &= \operatorname{argmin}\sum_{m}\left[\ln\sqrt{2\pi}+\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}\right]\\
&= \operatorname{argmin}\sum_{m}\left(\bar{C}^{i,m}-\mu_{\gamma}^{m}\right)^{2}
\end{aligned}
\tag{10}
$$

and our loss function becomes:

$$\mathcal{L}_C = \sum_m \left(\bar{C}^{i,m} - \mu_\gamma^m\right)^2. \tag{11}$$

Derivation of the loss function equation (5) for the 3D joints

The loss function for the **3D joints** in equation (5) is derived as follows, by maximising $P\left(J_{3D}^i | V_{3D}^i; \delta\right)$. In line with equation (2), we have:

$$\underset{\delta,\phi}{\operatorname{argmax}} \ln \prod_i P\left(J_{3D}^i | V_{3D}^i; \delta\right) = \underset{\delta,\phi}{\operatorname{argmin}} \sum_i \left(- \ln Q\left(\bar{J}_{3D}^i | V_{3D}^i; \phi\right) + \ln \frac{Q\left(J_{3D}^i | V_{3D}^i; \phi\right)}{P\left(J_{3D}^i | V_{3D}^i; \delta\right)} \right). \tag{12}$$

Considering the components in turn, we assume that $\bar{J}_{3D}^{i,k} \sim \mathcal{N}((B\mu_\phi)_k, (B\sigma_\phi)_k^2)$ and we have

$$\begin{aligned}
\ln Q\left(\bar{J}_{3D}^i | V_{3D}^i; \delta\right) &= \ln\left(\mathcal{N}_\phi(\bar{J}_{3D}^i - B\mu_\phi, diag(B\sigma_\phi))\right) \\
&= \ln\left(\prod_k \left(\frac{1}{(B\sigma_\phi)_k \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\bar{J}_{3D}^{i,k} - (B\mu_\phi)_k}{(B\sigma_\phi)_k}\right)^2\right)\right)\right) \\
&= \sum_k \left[\ln \frac{1}{(B\sigma_\phi)_k \sqrt{2\pi}} + \ln\exp\left(-\frac{1}{2}\left(\frac{\bar{J}_{3D}^{i,k} - (B\mu_\phi)_k}{(B\sigma_\phi)_k}\right)^2\right)\right] \\
&= \sum_k \left[-\frac{1}{2}\left(\frac{\bar{J}_{3D}^{i,k} - (B\mu_\phi)_k}{(B\sigma_\phi)_k}\right)^2 - \ln\left(\left(\sqrt{2\pi}B\sigma_\phi\right)_k\right)\right]
\end{aligned} \tag{13}$$

and

$$\begin{aligned}
\underset{\delta,\phi}{\operatorname{argmin}} \ln \frac{Q\left(J_{3D}^i | V_{3D}^i; \phi\right)}{P\left(J_{3D}^i | V_{3D}^i; \delta\right)} &= \underset{\delta,\phi}{\operatorname{argmin}}\, D_{KL}\left[Q\left(J_{3D}^i | V_{3D}^i; \phi\right) \| P\left(J_{3D}^i | V_{3D}^i; \delta\right)\right] \\
&= \underset{\delta,\phi}{\operatorname{argmin}} \frac{1}{2}\left[\ln \prod_k \frac{(B\sigma_v)_k}{(B\sigma_\phi)_k} - d + \sum_k \frac{(B\sigma_\phi)_k + \left((B\mu_v)_k - (B\mu_\phi)_k\right)^2}{(B\sigma_v)_k}\right].
\end{aligned} \tag{14}$$

Combining these, our loss function becomes:

$$\begin{aligned}
\mathcal{L}_{J_{3D}} = &\sum_k \left[\frac{1}{2}\left(\frac{\bar{J}_{3D}^{i,k} - (B\mu_\phi)_k}{(B\sigma_\phi)_k}\right)^2 + \ln\left(\left(\sqrt{2\pi}B\sigma_\phi\right)_k\right)\right] \\
&+ \frac{1}{2}\left[\ln \prod_k \frac{(B\sigma_v)_k}{(B\sigma_\phi)_k} - d + \sum_k \frac{(B\sigma_\phi)_k + \left((B\mu_v)_k - (B\mu_\phi)_k\right)^2}{(B\sigma_v)_k}\right]
\end{aligned} \tag{15}$$

Derivation of the loss function equation (6) for the 2D joints

The loss function for the **2D joints** in equation (6) is derived as follows, by maximising $P\left(J_{2D}^i | J_{3D}^i, C^i; \delta\right)$. In line with equation (2), we have:

$$\underset{\delta,\phi}{\operatorname{argmax}} \ln \prod_i P\left(J_{2D}^i | J_{3D}^i, C^i; \delta\right) = \underset{\delta,\phi}{\operatorname{argmin}} \sum_i \left(- \ln Q\left(\bar{J}_{2D}^i | J_{3D}^i, C^i; \phi\right) + \ln \frac{Q\left(J_{2D}^i | J_{3D}^i, C^i; \phi\right)}{P\left(J_{2D}^i | J_{3D}^i, C^i; \delta\right)} \right). \tag{16}$$

Considering the components in turn, we assume that $\bar{J}_{2D}^{i,k} \sim \mathcal{N}(S_k(\mu_\phi), S_k^2(\sigma_\phi))$ and we have

$$\begin{aligned}
\ln Q\left(\bar{J}_{2D}^i | J_{3D}^i, C^i; \phi\right) &= \ln\left(\prod_k \left(\frac{1}{S_k(\sigma_\phi)\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\bar{J}_{2D}^{i,k} - S_k(\mu_\phi)}{S_k(\sigma_\phi)}\right)^2\right)\right)\right) \\
&= \sum_k \left[\ln \frac{1}{S_k(\sigma_\phi)\sqrt{2\pi}} + \ln\exp\left(-\frac{1}{2}\left(\frac{\bar{J}_{2D}^{i,k} - S_k(\mu_\phi)}{S_k(\sigma_\phi)}\right)^2\right)\right] \\
&= \sum_k \left[-\frac{1}{2}\left(\frac{\bar{J}_{2D}^{i,k} - S_k(\mu_\phi)}{S_k(\sigma_\phi)}\right)^2 - \ln\left(\sqrt{2\pi}S(\sigma_\phi)\right)\right]
\end{aligned} \tag{17}$$

where $S_k(x) = (sBxR + T)_k$ and

$$\begin{aligned}\underset{\delta,\phi}{\operatorname{argmin}} \ln \frac{Q\left(J_{2D}^i|J_{3D}^i, C^i; \phi\right)}{P\left(J_{2D}^i|J_{3D}^i, C^i; \delta\right)} &= \underset{\delta,\phi}{\operatorname{argmin}}\, D_{KL}\left[Q\left(J_{2D}^i|J_{3D}^i, C^i; \phi\right) \| P\left(J_{2D}^i|J_{3D}^i, C^i; \delta\right)\right] \\ &= \underset{\delta,\phi}{\operatorname{argmin}} \frac{1}{2}\left[\ln \prod_k \frac{S_k(\sigma_v)}{S_k(\sigma_\phi)} - d + \sum_k \frac{S_k(\sigma_\phi) + (S_k(\mu_v) - S_k(\mu_\phi))^2}{S_k(\sigma_v)}\right].\end{aligned} \tag{18}$$

Combining these, our loss function becomes:

$$\begin{aligned}\mathcal{L}_{J_{2D}} = \sum_k &\left[\frac{1}{2}\left(\frac{\bar{J}_{2D}^{i,k} - S_k(\mu_\phi)}{S_k(\sigma_\phi)}\right)^2 + \ln\left(\sqrt{2\pi} S_k(\sigma_\phi)\right)\right] \\ &+ \frac{1}{2}\left[\ln \prod_k \frac{S_k(\sigma_v)}{S_k(\sigma_\phi)} - d + \sum_k \frac{S_k(\sigma_\phi) + (S_k(\mu_v) - S_k(\mu_\phi))^2}{S_k(\sigma_v)}\right]\end{aligned} \tag{19}$$

Proof of equation (7)

$$\begin{aligned}&\ln \iiint P\left(J_{2D}^i, J_{3D}, V_{3D}, C^i|I^i; \delta\right) dJ_{3D} dV_{3D} dC \\ &= \ln \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \frac{P\left(J_{2D}^i, J_{3D}, V_{3D}, C^i|I^i; \delta\right)}{Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)} dJ_{3D} dV_{3D} dC \\ &\geq \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln \frac{P\left(J_{2D}^i, J_{3D}, V_{3D}, C^i|I^i; \delta\right)}{Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)} dJ_{3D} dV_{3D} dC \\ &= ELBO\end{aligned} \tag{20}$$

The second last step of this derivation used Jensen's inequality with a strictly concave function. ELBO denotes Evidence Lower Bound, which means we can maximize our log likelihood function by maximizing ELBO. $Q\left(J_{3D}, V_{3D}, C\right)$ is an approximate probability distribution over variables of J_{3D} andV_{3D}. Then we can derive the distance between our log likelihood and ELBO as following

$$\begin{aligned}&\ln P\left(J_{2D}^i|I^i; \delta\right) - \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln \frac{P\left(J_{2D}^i, J_{3D}, V_{3D}, C^i|I^i; \delta\right)}{Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)} dJ_{3D} dV_{3D} dC \\ &= \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln P\left(J_{2D}^i|I^i; \delta\right) dJ_{3D} dV_{3D} dC \\ &\quad - \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln \frac{P\left(J_{2D}^i, J_{3D}, V_{3D}, C^i|I^i; \delta\right)}{Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)} dJ_{3D} dV_{3D} dC \\ &= \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln \frac{P\left(J_{2D}^i|I^i; \delta\right) Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)}{P\left(J_{2D}^i, J_{3D}, V_{3D}, C^i|I^i; \delta\right)} dJ_{3D} dV_{3D} dC \\ &= \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln \frac{Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)}{P\left(J_{3D}, V_{3D}, C|J_{2D}^i, I^i; \delta\right)} dJ_{3D} dV_{3D} \\ &= \iiint Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \ln \frac{Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)}{P\left(J_{3D}|V_{3D}, J_{2D}^i; \delta\right) P\left(V_{3D}|J_{2D}^i, I^i; \delta\right) P\left(C|J_{2D}^i, I^i; \delta\right)} dJ_{3D} dV_{3D} dC \\ &= D_{KL}\left[Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) \| P\left(J_{3D}|V_{3D}, J_{2D}^i; \delta\right) P\left(V_{3D}|J_{2D}^i, I^i; \delta\right) P\left(C|J_{2D}^i, I^i; \delta\right)\right]\end{aligned} \tag{21}$$

where $D_{KL}\left[Q \| P\right]$ denotes the Kullback–Leibler divergence which measures how the approximate probability distribution of Q is different from the prior probability distribution of P.

Since we know $D_{KL}\left[\cdot \| \cdot\right] \geq 0$, to minimize D_{KL}, the probability distribution of $Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right)$ is encouraged to be close to the probability distribution of $P\left(J_{3D}|V_{3D}, J_{2D}^i; \delta\right) P\left(V_{3D}|J_{2D}^i, I^i; \delta\right) P\left(C|J_{2D}^i, I^i; \delta\right)$. This yields

$$Q\left(J_{3D}, V_{3D}, C|I^i; \theta\right) = Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(V_{3D}|J_{2D}^i, I^i; \gamma\right) \tag{22}$$

where $B \in \mathbb{R}^{K \times \mathcal{V}}$ is a pre-defined regression matrix described in section 2.1 and $\phi, \gamma \in \theta$ are the variational parameters. Then we have:

$$
\begin{aligned}
&D_{KL}\left[Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(V_{3D}|J_{2D}^i, I^i; \gamma\right) \parallel P\left(J_{3D}|V_{3D}, J_{2D}^i; \delta\right) P\left(V_{3D}|J_{2D}^i, I^i; \delta\right) P\left(C|J_{2D}^i, I^i; \delta\right)\right] \\
&\quad = D_{KL}\left[Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) \parallel P\left(J_{3D}|V_{3D}, J_{2D}^i; \delta\right)\right] \\
&\qquad + D_{KL}\left[Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) \parallel P\left(V_{3D}|J_{2D}^i, I^i; \delta\right)\right] \\
&\qquad + D_{KL}\left[Q_\gamma\left(C|J_{2D}^i, I^i; \gamma\right) \parallel P\left(C|J_{2D}^i, I^i; \delta\right)\right]
\end{aligned}
\tag{23}
$$

We assume the approximate probability distribution of $Q_\gamma\left(C|J_{2D}^i, I^i; \gamma\right)$ and prior probability distribution of $P\left(C|J_{2D}^i, I^i; \delta\right)$ take on Gaussian distributions $\mathcal{N}_\gamma\left(\mu_\gamma, diag\left(\sigma_\gamma\right)\right)$ and $\mathcal{N}\left(\mu_C, diag\left(\sigma_C\right)\right)$, respectively. Similarly, $Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right)$ and $P\left(V_{3D}|J_{2D}^i, I^i; \delta\right)$ are assumed to subject to Gaussian distributions $\mathcal{N}_\phi\left(\mu_\phi, diag\left(\sigma_\phi\right)\right)$ and $\mathcal{N}\left(\mu_v, diag\left(\sigma_v\right)\right)$, respectively. Then the approximate probability distribution of $Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right)$ and prior probability distribution of $P\left(J_{3D}|V_{3D}, J_{2D}^i; \delta\right)$ have the forms of $\mathcal{N}_B\left(B\mu_\phi, diag\left(B\sigma_\phi\right)\right)$ and $\mathcal{N}\left(B\mu_j, diag\left(B\sigma_v\right)\right)$ respectively. With above definition, we can obtain ELBO:

$$
\begin{aligned}
ELBO = &\iiint Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(C|J_{2D}^i, I^i; \phi\right) \\
&\quad Q_\gamma\left(V_{3D}|J_{2D}^i, I^i; \gamma\right) \ln \frac{P\left(J_{2D}^i, J_{3D}, V_{3D}, C|I^i; \delta\right)}{Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(C|J_{2D}^i, I^i; \gamma\right)} dJ_{3D} dV_{3D} dC \\
= &\iiint Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(C|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(V_{3D}|J_{2D}^i, I^i; \gamma\right) \\
&\quad \ln P\left(J_{2D}^i|J_{3D}, V_{3D}, C, I^i; \delta\right) dJ_{3D} dV_{3D} dC \\
&\quad + \iiint Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(C|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(V_{3D}|J_{2D}^i, I^i; \gamma\right) \\
&\quad \ln \frac{P\left(J_{3D}, V_{3D}, C|I^i; \delta\right)}{Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(C|J_{2D}^i, I^i; \gamma\right)} dJ_{3D} dV_{3D} dC \\
= &E_{V_{3d} \sim \mathcal{N}_\phi, C \sim \mathcal{N}_\gamma, J_{3D} \sim \mathcal{N}_B} \ln P\left(J_{2D}^i|J_{3D}, V_{3D}, C, I^i; \delta\right) - \\
&\quad D_{KL}\left[Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(C|J_{2D}^i, I^i; \gamma\right) \parallel P\left(J_{3D}, V_{3D}, C|I^i; \delta\right)\right] \\
= &E_{V_{3d} \sim \mathcal{N}_\phi, C \sim \mathcal{N}_\gamma, J_{3D} \sim \mathcal{N}_B} - \ln P\left(J_{2D}^i|J_{3D}, V_{3D}, C, I^i; \delta\right) + \\
&\quad D_{KL}\left[Q_B\left(J_{3D}|V_{3D}, J_{2D}^i; B\right) Q_\phi\left(V_{3D}|J_{2D}^i, I^i; \phi\right) Q_\gamma\left(C|J_{2D}^i, I^i; \gamma\right) \parallel P\left(J_{3D}, V_{3D}, C|I^i; \delta\right)\right]
\end{aligned}
\tag{24}
$$

where $E_{V_{3D} \sim \mathcal{N}_\phi, C \sim \mathcal{N}_\gamma, J_{3D} \sim \mathcal{N}_B} - \ln P\left(J_{2D}^i|J_{3D}, V_{3D}, C, I^i; \delta\right)$ can be computed using equation (6) after sampling V_{3D}, J_{3D} and C from probability distributions of $\mathcal{N}_\phi, \mathcal{N}_\gamma$ and $\mathcal{N}_B$. $P\left(J_{3D}, V_{3D}, C|I^i; \delta\right)$ is a prior probability, which is learned via MANO neural network and our pre-trained Camera Parameters model.

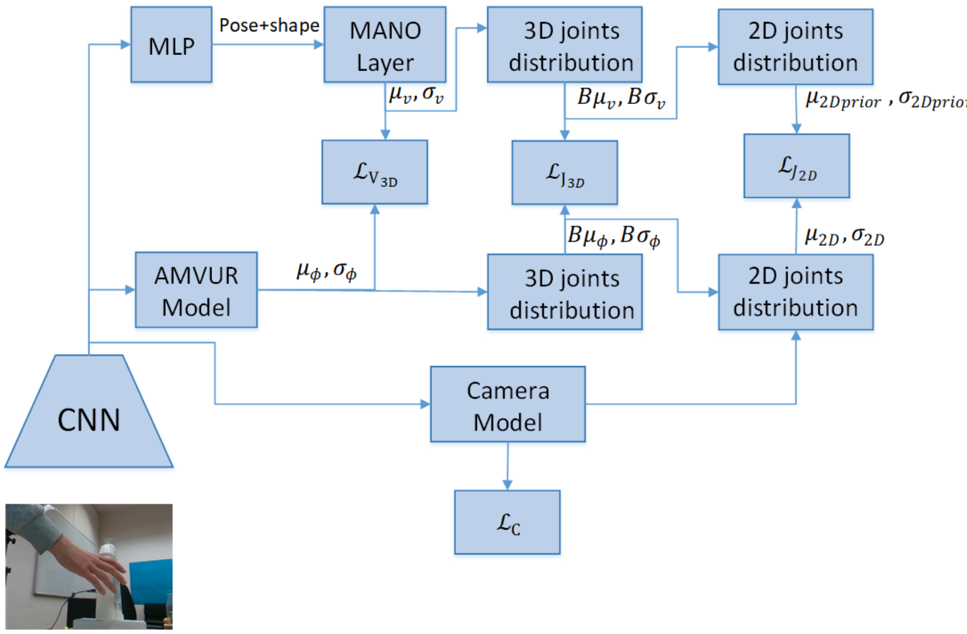

Figure 1. The detailed pipeline for supervised training. Here, $\mu_\phi, \sigma_\phi \in \mathbb{R}^{778\times 3}$ are outputs of AMVUR. $\mu_v \in \mathbb{R}^{778\times 3}$ is an output of the MANO Layer, and $\sigma_v \in \mathbb{R}^{778\times 3}$ is equal to $\mathbf{1}$. $B \in \mathbb{R}^{K\times \mathcal{V}}$ is a pre-defined regression matrix from the MANO model. $\mu_{2D} = sB\mu_\phi R + T$, $\sigma_{2D} = sB\sigma_\phi R + T$, $\mu_{2D_{prior}} = sB\mu_v R + T$ and $\sigma_{2D_{prior}} = sB\sigma_v R + T$. s, B and R are camera parameters obtained from the Camera Model.

The D_{KL} is estimated as below:

$$
\begin{aligned}
& D_{KL}\left[Q_\phi\left(V_{3D}|J_{2D}^i, I^i;\phi\right) \parallel P\left(V_{3D}|J_{2D}^i, I^i;\delta\right)\right] + D_{KL}\left[Q_B\left(J_{3D}|V_{3D}, J_{2D}^i;\phi\right) \parallel P\left(J_{3D}|V_{3D}, J_{2D}^i;\delta\right)\right] + \\
& D_{KL}\left[Q_\gamma\left(C|J_{2D}^i, I^i;\gamma\right) \parallel P\left(C|J_{2D}^i, I^i;\delta\right)\right] \\
& = \frac{1}{2}\left[\log\frac{|diag\left(\sigma_v\right)|}{|diag\left(\sigma_\phi\right)|} - d + tr\left\{diag\left(\sigma_v\right)^{-1} diag\left(\sigma_\phi\right)\right\} + \left(\mu_v - \mu_\phi\right)^T diag\left(\sigma_v\right)^{-1}\left(\mu_v - \mu_\phi\right)\right] + \\
& \frac{1}{2}\left[\log\frac{|diag\left(B\sigma_v\right)|}{|diag\left(B\sigma_\phi\right)|} - d + tr\left\{diag\left(B\sigma_v\right)^{-1} diag\left(B\sigma_\phi\right)\right\} + \left(diag\left(B\mu_v\right) - diag\left(B\mu_\phi\right)\right)^T diag\left(B\sigma_v\right)^{-1}\left(diag\left(B\mu_v\right) - diag\left(B\mu_\phi\right)\right)\right] + \\
& \frac{1}{2}\left[\log\frac{|diag\left(\sigma_C\right)|}{|diag\left(\sigma_\gamma\right)|} - d + tr\left\{diag\left(\sigma_C\right)^{-1} diag\left(\sigma_\gamma\right)\right\} + \left(\mu_C - \mu_\gamma\right)^T diag\left(\sigma_C\right)^{-1}\left(\mu_C - \mu_\gamma\right)\right] \\
& = \frac{1}{2}\left[\log\frac{\prod_m \sigma_v^m}{\prod_m \sigma_\phi^m} - d + \Sigma_m\frac{\sigma_\phi^m}{\sigma_v^m} + \Sigma_m\frac{\left(\mu_v^m - \mu_\phi^m\right)^2}{\sigma_v^m}\right] + \frac{1}{2}\left[\log\frac{\prod_k\left(sB\sigma_v\right)_k}{\prod_k\left(sB\sigma_\phi\right)\right)_k} - d + \Sigma_k\frac{\left(sB\sigma_\phi\right)_k}{\left(sB\sigma_v\right)_k} + \Sigma_k\frac{\left(\left(sB\mu_v\right)_k - \left(sB\mu_\phi\right)_k\right)^2}{\left(sB\sigma_v\right)_k}\right] + \\
& \frac{1}{2}\left[\log\frac{\prod_m \sigma_C^m}{\prod_m \sigma_\gamma^m} - d + \Sigma_m\frac{\sigma_\gamma^m}{\sigma_C^m} + \Sigma_m\frac{\left(\mu_C^m - \mu_\gamma^m\right)^2}{\sigma_C^m}\right]
\end{aligned}
\tag{25}
$$

where the mean μ_ϕ and variance σ_ϕ of the approximate probability distribution of $Q\left(J_{3D}|V_{3D}, J_{2D}^i;\phi\right)$ are outputs of two multilayer perceptron (MLP) nerual networks with $\phi \in \theta$. The mean μ_γ and variance σ_γ of the approximate probability distribution of $Q\left(C|J_{2D}^i, I^i;\gamma\right)$ are outputs of two multilayer perceptron (MLP) neural networks with $\gamma \in \theta$. μ_v and σ_v are the mean and variance of the prior probability distribution $\mathcal{N}\left(\mu_v, diag\left(\mathbf{1}\right)\right)$ estimated from MANO model. Our D_{KL} loss penalizes differences between the posterior distribution Q and the prior distribution P. During training, this KL loss pulls the posterior distribution and the prior distribution towards each other.

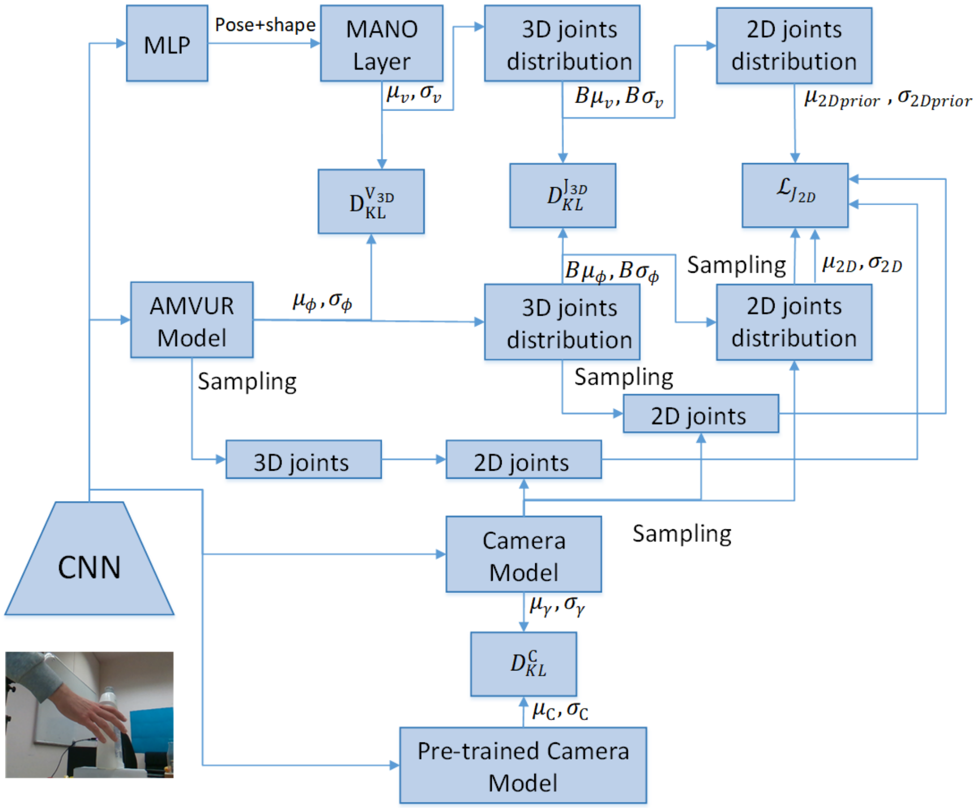

Figure 2. The detailed pipeline of weakly supervised training. Here, $\mu_\phi, \sigma_\phi, \mu_v$ and $\sigma_v \in \mathbb{R}^{778\times 3}$. μ_ϕ, σ_ϕ are output of AMVUR. μ_v is output of MANO Layer, σ_v is equal to $\mathbf{1}$. $\mu_C, \sigma_C, \mu_\gamma$ and $\sigma_\gamma \in \mathbb{R}^{10}$. μ_C is the output of pre-trained S^2HAND, σ_C is equal to $\mathbf{1}$. μ_γ and σ_γ are output of our Camera Model. $B \in \mathbb{R}^{K\times\mathcal{V}}$ is a pre-defined regression matrix from the MANO model. $\mu_{2D} = sB\mu_\phi R + T$, $\sigma_{2D} = sB\sigma_\phi R + T$, $\mu_{2D_{prior}} = sB\mu_v R + T$ and $\sigma_{2D_{prior}} = sB\sigma_v R + T$. s, B and R are camera parameters obtained from the Camera Model.

Table 1. Network architecture and configurations of the proposed model. MLP is multilayer perceptron

Stage	Configuration	Output
0	Input image	224×224×3
	Feature extraction	
1	Resnet50(shallow) [1]	112×112×64
1	Resnet50(global) [1]	2048×1
	Prior-Net	
2	MLP(Pose+shape)	48
2	MANO layer(mean)	778×3
	AMVUR(see Section 3.3 for more detail)	
3	Positional encoding	799×2051
3	cross-attention	778×512
3	self-attention(mean+variance)	778×3×2
	Camera Model	
4	MLP(mean of Rotation, Translation and Scale)	10
4	MLP(variance of Rotation, Translation and Scale)	10
	Texture Regression(see Section 3.4 for more detail)	
6	Occlusion-aware Rasterization	V_{2D},$\mathcal{M}$,triangle barycentric
6	Reverse Interpolation	778×2112
6	Positional encoding	778×2115
6	Self-attention	778×3
6	Interpolation	224×224×3

Training algorithm for supervised training

Algorithm 1 Training algorithm for supervised training.

Input: Image I; MANO layer and regression matrix B; Ground-truth: 3D Vertices $\bar{V}_{3D}$, Camera parameters $\bar{C}$, 2D Joints $\bar{J}_{2D}$

for epoch $e \leq E$ **do**
    for each image batch **do**
        Extract a global feature vector $\mathcal{F}$ from our ResNet50 [1] for each image
        Send $\mathcal{F}$ to Camera Model to generate Camera Parameters distribution $\mathcal{N}\left(\mu_\gamma, diag\left(\mathbf{1}\right)\right)$, $s, R, T \in \gamma$
        Send $\mathcal{F}$ to Prior-net to generate 3D vertices distribution $\mathcal{N}\left(\mu_v, diag\left(\mathbf{1}\right)\right)$), 3D joints distribution $\mathcal{N}\left(B\mu_v, Bdiag\left(\mathbf{1}\right)\right)$) and 2D joints distribution $\mathcal{N}\left(sB\mu_v R + T, sBdiag\left(\mathbf{1}\right)\right) + T)$
        Send $\mathcal{F}$ to AMVUR to generate 3D vertices distribution $\mathcal{N}\left(\mu_\phi, diag\left(\sigma_\phi\right)\right)$, 3D joints distribution $\mathcal{N}\left(B\mu_\phi, Bdiag\left(, diag\left(\sigma_\phi\right)\right)\right)$) and 2D joints distribution $\mathcal{N}\left(sB\mu_\phi R + T, sBdiag\left(diag\left(\sigma_\phi\right)\right)\right) R + T)$
        Compute loss equations 3-6
        Update model weights
    end for
end for

Training algorithm for weakly supervised training

Algorithm 2 Training algorithm for weakly supervised training.

Input: Image I; MANO layer and regression matrix B; Ground-truth: 2D Joints $\bar{J}_{2D}$

for epoch $e \leq E$ **do**
    for each image batch **do**
        Extract a global feature vector $\mathcal{F}$ from EfficientNet-B0 [2] for each image
        Send $\mathcal{F}$ to Camera Model to generate Camera Parameters distribution $\mathcal{N}\left(\mu_\gamma, diag\left(\mathbf{1}\right)\right)$
        Sample s,R,T from the distribution $\mathcal{N}\left(\mu_\gamma, diag\left(\mathbf{1}\right)\right)$
        Send $\mathcal{F}$ to Prior-net to generate 3D vertices distribution $\mathcal{N}\left(\mu_v, diag\left(\sigma_v\right)\right)$), 3D joints distribution $\mathcal{N}\left(B\mu_v, Bdiag\left(\sigma(v)\right)\right)$) and 2D joints distribution $\mathcal{N}\left(sB\mu_v R + T, sBdiag\left(\sigma(v)\right)\right) + T)$
        Sample 3D vertices,3D joints and 2D joints from the above distribution generated by the Prior-net
        Send $\mathcal{F}$ to AMVUR to generate 3D vertices distribution $\mathcal{N}\left(\mu_\phi, diag\left(\sigma_\phi\right)\right)$, 3D joints distribution $\mathcal{N}\left(B\mu_\phi, Bdiag\left(\sigma_\phi\right)\right)$) and 2D joints distribution $\mathcal{N}\left(sB\mu_\phi R + T, sBdiag\left(\sigma_\phi\right)\right) R + T)$
        Sample 3D vertices,3D joints and 2D joints from the above distribution generated by the AMVUR
        Compute loss equation 8
        Update model weights
    end for
end for

Testing algorithm for supervised and weakly supervised testing

Algorithm 3 Testing algorithm for weakly supervised training.

Input: Image I; MANO layer and regression matrix B; 2D Joints $\bar{J}_{2D}$

for each image batch **do**
    Extract a global feature vector $\mathcal{F}$ from our backbone Convolutional Neural Network (CNN) for each image
    Send $\mathcal{F}$ to Camera Model to generate Camera Parameters distribution $\mathcal{N}\left(\mu_\gamma, diag\left(1\right)\right)$
    Output μ_γ
    Send $\mathcal{F}$ to AMVUR to generate 3D vertices distribution $\mathcal{N}\left(\mu_\phi, diag\left(\sigma_\phi\right)\right)$, 3D joints distribution $\mathcal{N}\left(B\mu_\phi, Bdiag\left(\sigma_\phi\right)\right)$) and 2D joints distribution $\mathcal{N}\left(sB\mu_\phi R + T, sBdiag\left(\sigma_\phi\right)\right) R + T)$
    Output $\mu_\phi, B\mu_\phi$ and $sB\mu_\phi R + T$
end for

Training algorithm for Texture Regression

Algorithm 4 Training algorithm for Texture Regression.

Input: Image I; Our probabilistic model trained by above supervised or weakly supervised strategies

for epoch $e \leq E$ **do**
  for each image batch **do**
    Extract a global feature vector $\mathcal{F}$ and a shallow feature vector $\mathcal{F}_{map}$ from our backbone Convolutional Neural Network (CNN) for each image
    Send $\mathcal{F}$ to Camera Model to generate Camera Parameters distribution $\mathcal{N}\left(\mu_\gamma, diag\left(1\right)\right)$
    Output μ_γ
    Send $\mathcal{F}$ to AMVUR to generate 3D vertices distribution $\mathcal{N}\left(\mu_\phi, diag\left(\sigma_\phi\right)\right)$, 3D joints distribution $\mathcal{N}\left(B\mu_\phi, Bdiag\left(\sigma_\phi\right)\right)$ and 2D joints distribution $\mathcal{N}\left(sB\mu_\phi R + T, sBdiag\left(\sigma_\phi\right)\right) R + T)$
    Output μ_ϕ, $B\mu_\phi$ and $sB\mu_\phi R + T$
    Concatenate $\mathcal{F}$ and $\mathcal{F}_{map}$, followed by Reverse Interpolation for generating 3D vertex feature.
    Send Vertex feature to Positional encoding, follow by a self-attention for generating 3D vertex RGB
    Send μ_γ and μ_ϕ to Occlusion-aware Rasterization and obtain V_{2D},$\mathcal{M}$,triangle barycentric
    send 3D vertex RGB and the triangle barycentric to Interpolation to generate 2D rendered hand
    Compute loss in equation 12
    Update model weights
  end for
end for

Training for prior-net individually

Algorithm 5 Training for prior-net individually

Input: Image I; MANO layer and regression matrix B; Ground-truth: 3D Vertices $\bar{V}_{3D}$, Camera parameters $\bar{C}$, 2D Joints $\bar{J}_{2D}$ for supervised, and only 2D Joints $\bar{J}_{2D}$ for weakly supervised

for epoch $e \leq E$ **do**
  for each image batch **do**
    Extract a global feature vector $\mathcal{F}$ from backbone CNN for each image
    Send $\mathcal{F}$ to Camera Model to generate Camera Parameters
    Send $\mathcal{F}$ to Prior-net to generate 3D vertices, 3D joints and 2D joints
    For supervised, compute L2 loss on 3D Vertices, Camera parameters and 2D Joints. For weakly supervised, compute L2 loss on 2D joints only
    Update model weights
  end for
end for

Training for AMVUR individually

Algorithm 6 Training for AMVUR individually

Input: Image I; MANO layer and regression matrix B; Ground-truth: 3D Vertices $\bar{V}_{3D}$, Camera parameters $\bar{C}$, 2D Joints $\bar{J}_{2D}$ for supervised, and only 2D Joints $\bar{J}_{2D}$ for weakly supervised

for epoch $e \leq E$ **do**
  for each image batch **do**
    Extract a global feature vector $\mathcal{F}$ from backbone CNN for each image
    Send $\mathcal{F}$ to Camera Model to generate Camera Parameters
    Send $\mathcal{F}$ to AMVUR to generate 3D vertices, 3D joints and 2D joints
    For supervised, compute L2 loss on 3D Vertices, Camera parameters and 2D Joints. For weakly supervised, compute L2 loss on 2D joints only
    Update model weights
  end for
end for

Table 2. Hand reconstruction performance compared with state-of-the-art methods on FreiHAND testing dataset after Procrustes alignment. [4]* additionally uses 40,000+ synthetic images and 3D annotations of RHD dataset for training.

Training Scheme	Method	Category	$AUC_J \uparrow$	MPJPE↓	$AUC_V \uparrow$	MPVPE↓	$F_5 \uparrow$	$F_{15} \uparrow$
Supervised	Hasson et al. [5]	Model-based	0.74	13.3	0.74	13.3	0.43	0.91
	FreiHAND [3]	Model-based	0.35	35.0	0.74	13.2	0.43	0.90
	Boukhayma et al. [6]	Model-based	0.78	11.0	0.78	10.9	0.52	0.93
	Qian et al. [7]	Model-based	0.78	11.1	0.78	11.0	0.51	0.93
	I2L-MeshNet [8]	Model-free	-	7.4	-	7.6	0.681	0.973
	Pose2Mesh [9]	Model-free	-	7.7	-	7.8	0.674	0.969
	METRO [10]	Model-free	-	6.5	-	6.3	0.731	0.984
	Chen et al. [11]	Model-free	-	6.1	-	6.2	0.760	0.984
	Ours(final)	Probabilistic	**0.89**	**6.2**	**0.89**	**6.1**	**0.767**	**0.987**
Weakly-Supervised	S^2HAND [12]	Model-based	0.730	-	0.725	-	0.42	0.89
	MANO Fit [3]	Model-based	0.730	13.7	0.729	13.7	0.439	0.892
	BMC [4] *	Model-based	0.780	11.3	-	-	-	-
	Ours(final)	Probabilistic	**0.796**	**10.8**	**0.792**	**10.9**	**0.517**	**0.943**

Results

FreiHand

FreiHAND [3] is a large-scale 3D hand dataset that contains 130,240 training images and 3,960 testing images. Each training image has a green screen background or a synthetic background. Testing images are collected in controlled outdoor and office environments, which makes this dataset less challenging than the HO3Dv2 and v3 datasets. Experimental results and comparison with the state of the art approaches are shown in Table 2.

HO3Dv2

Experimental results of our model evaluated on the **HO3Dv2** dataset are shown for qualitative comparison of the mesh reconstruction with the state of the art approaches in figures 3 and 4, and of texture reconstruction in figures 5 and 6.

Bayesian vs. L1/L2

To better guide our proposed AMVUR model during training, our probabilistic model takes the MANO parametric model as a prior-net and AMVUR estimates the probability distribution of mesh vertices conditioned on the prior-net. L1/L2 loss is less able to capture the data distribution. KL-divergence allows our model to take into account the uncertainty and variability in the hand, which is important for modeling complex and varied 3D meshes. Further, sampling from the distribution during training of our probabilistic model allows the model to explore different variations of the mesh, leading to a more robust and generalizable model. As shown in Tab.3, replacing KL-divergence with L2 leads to decreased performance.

Impact of camera parameter loss in Table 3

The results reported in Table 3 of the paper are calculated *after* Procrustes alignment, eliminating differences in the underlying camera coordinate systems. As a result, the $\mathcal{L}_{V_{3D}}$ and $\mathcal{L}_{J_{3D}}$ loss terms are sufficient to reconstruct the hand in the wrist-relative coordinate system; the camera loss $\mathcal{L}_C$ has no impact on performance. To provide additional context, the ablation study in Tab.4 shows the impact of each loss term *before* Procrustes alignment, highlighting the impact of $\mathcal{L}_C$ in this setting.

Evaluation before Procrustes alignment

As reported in Tab.5, our probabilistic method achieves significant improvement over existing approaches in both supervised and weakly-supervised scenarios.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 7, 8

[2] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019. 8

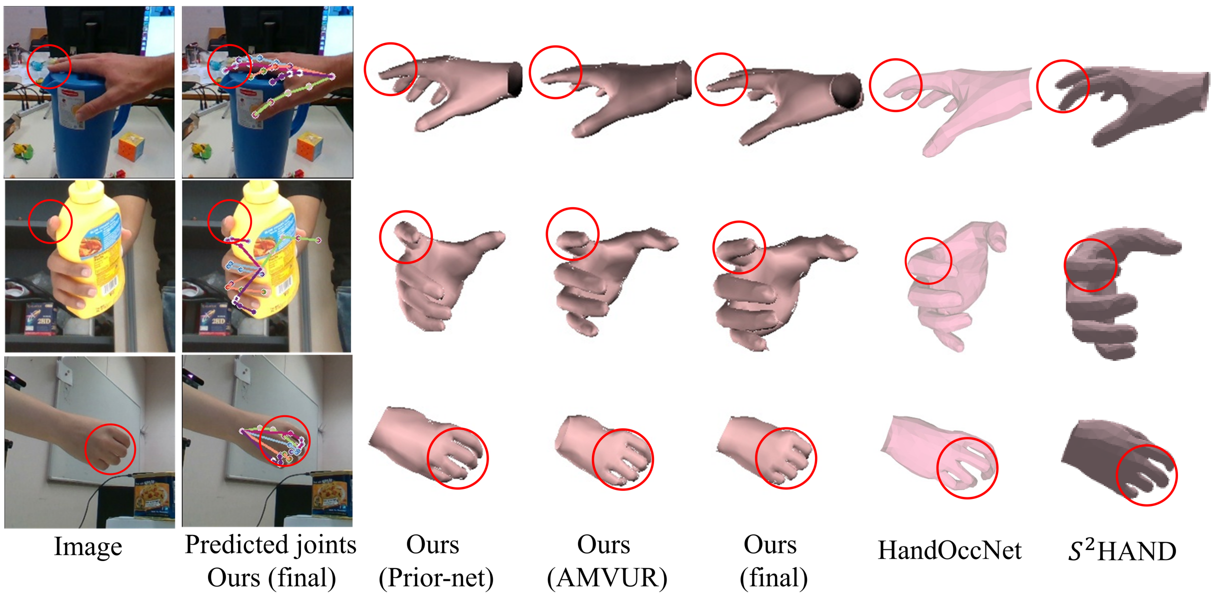

Figure 3. Qualitative comparison of the proposed models and state-of-the-art 3D hand mesh estimation methods HandOccNet [13] and S^2HAND [12] on HO3Dv2.

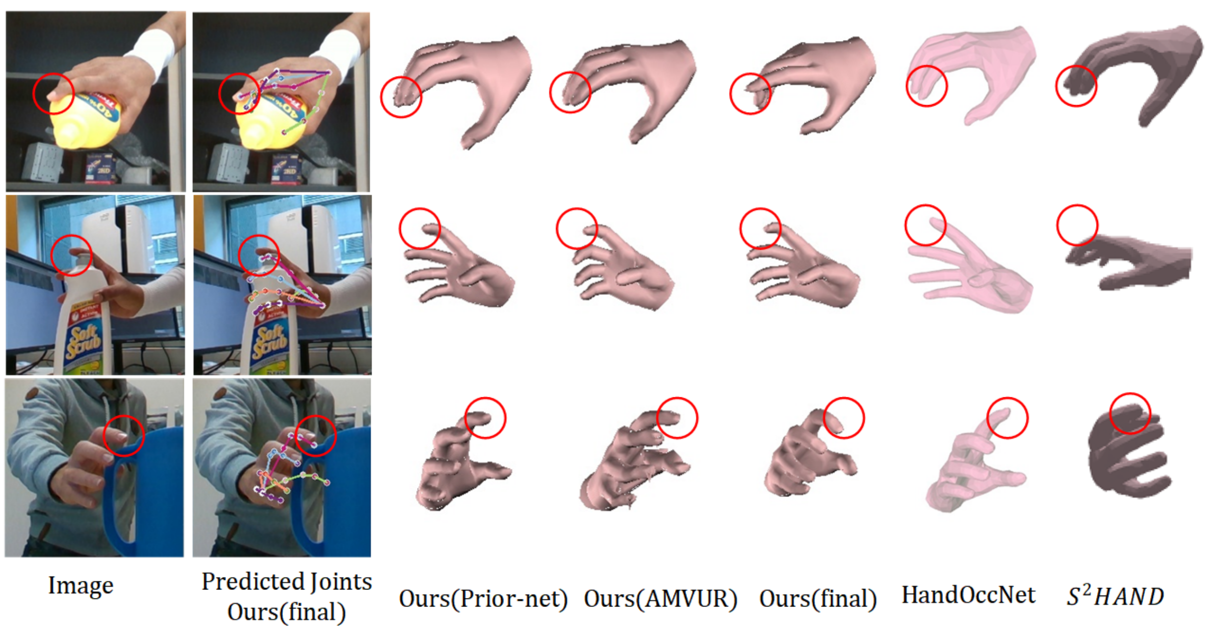

Figure 4. Qualitative comparison of the proposed models and state-of-the-art 3D hand mesh estimation methods HandOccNet [13] and S^2HAND [12] on HO3Dv2.

[3] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 813–822, 2019. 10

[4] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, and J. Kautz, "Weakly supervised 3d hand pose estimation via biomechanical constraints," in *European Conference on Computer Vision*, pp. 211–228, Springer, 2020. 10

[5] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11807–11816,

Figure 5. Qualitative comparison of our proposed model and state-of-the-art texture regression model S^2HAND [12] on HO3Dv2.

Figure 6. Qualitative comparison of our proposed model and state-of-the-art texture regression model S^2HAND [12] on HO3Dv2.

Table 3. Our Bayesian versus L2 on HO3Dv2 dataset.

Setting	Method	$AUC_J\uparrow$	MPJPE↓	$AUC_V\uparrow$	MPVPE↓	$F_5\uparrow$	$F_{15}\uparrow$
Supervised	L2	0.816	9.2	0.817	9.4	0.541	0.959
	Ours	**0.835**	**8.3**	**0.836**	**8.2**	**0.608**	**0.965**
Weakly supervised	L2	0.776	10.9	0.775	11.3	0.453	0.934
	Ours	**0.787**	**10.3**	**0.784**	**10.8**	**0.482**	**0.949**

2019. 10

[6] A. Boukhayma, R. d. Bem, and P. H. Torr, “3d hand shape and pose from images in the wild,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10843–10852, 2019. 10

Table 4. Loss impact in supervised training on HO3Dv2 before Procrustes alignment.

Loss terms				MPJPE↓	MPVPE↓	$F_5\uparrow$	$F_{15}\uparrow$
$\mathcal{L}_{J_{2D}}$	$\mathcal{L}_C$	$\mathcal{L}_{J_{3D}}$	$\mathcal{L}_{V_{3D}}$				
✓				29.4	28.8	0.207	0.659
✓	✓			24.3	27.6	0.211	0.673
✓	✓	✓		23.8	27.2	0.213	0.676
✓	✓	✓	✓	**19.2**	**18.6**	**0.305**	**0.799**

Table 5. Results on HO3Dv2 before Procrustes alignment.

	Supervised methods						Weakly Supervised methods	
	[11]	[35]	[33]	[26]	[14]	**Ours**	[13]	**Ours**
MPJPE↓	33.2	26.8	55.2	30.0	24.9	**19.2**	47.1	**29.4**

[7] N. Qian, J. Wang, F. Mueller, F. Bernard, V. Golyanik, and C. Theobalt, "Html: A parametric hand texture model for 3d hand reconstruction and personalization," in *European Conference on Computer Vision*, pp. 54–71, Springer, 2020. 10

[8] G. Moon and K. M. Lee, "I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image," in *European Conference on Computer Vision*, pp. 752–768, Springer, 2020. 10

[9] H. Choi, G. Moon, and K. M. Lee, "Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose," in *European Conference on Computer Vision*, pp. 769–787, Springer, 2020. 10

[10] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1954–1963, 2021. 10

[11] X. Chen, Y. Liu, Y. Dong, X. Zhang, C. Ma, Y. Xiong, Y. Zhang, and X. Guo, "Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20544–20554, 2022. 10

[12] Y. Chen, Z. Tu, D. Kang, L. Bao, Y. Zhang, X. Zhe, R. Chen, and J. Yuan, "Model-based 3d hand reconstruction via self-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10451–10460, 2021. 10, 11, 12

[13] J. Park, Y. Oh, G. Moon, H. Choi, and K. M. Lee, "Handoccnet: Occlusion-robust 3d hand mesh estimation network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1496–1505, 2022. 11