

PAD-Hand: Physics-Aware Diffusion for Hand Motion Recovery

Supplementary Material

The supplementary material is composed of an appendix and a video file.

Appendix

- Section A: Details of the Euler-Lagrange Terms
- Section B: Bayesian Inference: Derivations
- Section C: Additional Experiments
- Section D: Additional Visualizations
- Section E: Discussion: PAD-Hand vs. DIP

Video file

- **“Hand Motion Comparison With Variance.mp4”**: Hand motion comparison with image-based estimates from WiLoR [45] and includes our dynamic variance estimation visualizations at joint, mesh level.

A. Details of the Euler-Lagrange Terms

In this section, we further clarify the terms in Eq. 1. Generalized mass matrix M captures how the hand’s mass and inertia are distributed across joints. It determines how joint accelerations are related to the required generalized forces. The term C represents Coriolis and centrifugal effects caused by motion, accounting for velocity-dependent interactions between different joints. The term g corresponds to gravitational forces, i.e., the torques induced by gravity under the current hand configuration. Finally, \mathcal{F} denotes the net generalized forces that explain the observed hand motion, including the forces needed to produce the measured accelerations and to balance the dynamic effects captured by M , C , and g .

B. Bayesian Inference: Derivations

In this section, we derive equations 14, 15 and 16. We start with our sampling equation

$$x_{1:T}^{n-1} = A_n x_{1:T}^n + B_n \hat{x}_{1:T} + \Sigma_n \epsilon \quad (18)$$

Taking variance on both sides leads to

$$\begin{aligned} \text{Var}(x_{1:T}^{n-1}) &= A_n^2 \text{Var}(x_{1:T}^n) + B_n^2 \text{Var}(\hat{x}_{1:T}) \\ &\quad + \Sigma_n^2 + 2A_n B_n \text{Cov}(x_{1:T}^n, \hat{x}_{1:T}) \\ &\quad + 2A_n C_n \text{Cov}(x_{1:T}^n, \epsilon) + 2B_n C_n \text{Cov}(\hat{x}_{1:T}, \epsilon) \end{aligned} \quad (19)$$

Since $\epsilon \sim \mathcal{N}(0, I)$ is independent of $x_{1:T}^n$ and $\hat{x}_{1:T}$, their covariances $\text{Cov}(x_{1:T}^n, \epsilon)$, $\text{Cov}(\hat{x}_{1:T}, \epsilon)$ are zero. Hence we have

$$\begin{aligned} \text{Var}(x_{1:T}^{n-1}) &= A_n^2 \text{Var}(x_{1:T}^n) + B_n^2 \text{Var}(\hat{x}_{1:T}) \\ &\quad + \Sigma_n^2 + 2A_n B_n \text{Cov}(x_{1:T}^n, \hat{x}_{1:T}) \end{aligned} \quad (20)$$

Now we derive the covariance $\text{Cov}(x_{1:T}^n, \hat{x}_{1:T})$ as follows:

$$\text{Cov}(x_{1:T}^n, \hat{x}_{1:T}) = \mathbb{E}[x_{1:T}^n \cdot \hat{x}_{1:T}] - \mathbb{E}[x_{1:T}^n] \cdot \mathbb{E}[\hat{x}_{1:T}] \quad (21)$$

Using the law of total expectation $\mathbb{E}[\mathbb{E}[X|Y]] = \mathbb{E}[X]$, $\mathbb{E}[\hat{x}_{1:T}]$ can be written as

$$\mathbb{E}[\hat{x}_{1:T}] = \mathbb{E}_{x_{1:T}^n}[\mathbb{E}[\hat{x}_{1:T}|x_{1:T}^n]] = \mathbb{E}_{x_{1:T}^n}[\hat{X}_{1:T}] \quad (22)$$

where $\hat{X}_{1:T} = f_\phi(x_{1:T}^n, y_{1:T}, n)$ and for the first term

$$\begin{aligned} \mathbb{E}[x_{1:T}^n \cdot \hat{x}_{1:T}] &= \mathbb{E}_{x_{1:T}^n}[x_{1:T}^n \cdot \mathbb{E}[\hat{x}_{1:T}|x_{1:T}^n]] \\ &= \mathbb{E}_{x_{1:T}^n}[x_{1:T}^n \cdot \hat{X}_{1:T}] \end{aligned} \quad (23)$$

Integrating equations 22 and 23 to the equation 21 leads to

$$\begin{aligned} \text{Cov}(x_{1:T}^n, \hat{x}_{1:T}) &= \mathbb{E}_{x_{1:T}^n}[x_{1:T}^n \cdot \hat{X}_{1:T}] \\ &\quad - \mathbb{E}[x_{1:T}^n] \cdot \mathbb{E}_{x_{1:T}^n}[\hat{X}_{1:T}] \end{aligned} \quad (24)$$

which we approximate via Monte Carlo (MC) estimation:

$$\begin{aligned} \text{Cov}(x_{1:T}^n, \hat{x}_{1:T}) &\approx \frac{1}{S} \sum_{i=1}^S (x_{1:T}^{n,i} \cdot \hat{X}_{1:T}^i) \\ &\quad - \mathbb{E}[x_{1:T}^n] \cdot \frac{1}{S} \sum_{i=1}^S \hat{X}_{1:T}^i \end{aligned} \quad (25)$$

where S is the sample size.

For the equation 16, taking expectation on both sides of equation 18 leads to

$$\mathbb{E}[x_{1:T}^{n-1}] = A_n \mathbb{E}[x_{1:T}^n] + B_n \mathbb{E}[\hat{x}_{1:T}] + \Sigma_n \mathbb{E}[\epsilon] \quad (26)$$

Since $\epsilon \sim \mathcal{N}(0, I)$, $\mathbb{E}[\epsilon]$ is zero and by using law of total expectation on $\mathbb{E}[\hat{x}_{1:T}]$ we get

$$\mathbb{E}[x_{1:T}^{n-1}] = A_n \mathbb{E}[x_{1:T}^n] + B_n \mathbb{E}_{x_{1:T}^n}[\hat{X}_{1:T}] \quad (27)$$

C. Additional Experiments

C.1. Ablation Study on Data-driven Losses

In this section, we perform an ablation study on data-driven losses which are the backbone loss \mathcal{L}_b , geometric loss \mathcal{L}_g , and consistency regularization \mathcal{L}_r and report the results in Table 5. \mathcal{L}_g and \mathcal{L}_r enable data-driven improvements, while further adding our physics loss yields the best performance.

Table 5. Effectiveness of data-driven losses.

Loss				Evaluation Metrics		
\mathcal{L}_b	\mathcal{L}_g	\mathcal{L}_r	\mathcal{L}_{EL}	PA-MPJPE↓	MPJPE↓	ACCEL↓
WiLoR				4.88	12.75	6.70
✓				5.01	12.57	3.83
✓	✓			4.83	10.89	3.57
✓	✓	✓		4.65	10.62	3.36
✓	✓	✓	✓	4.63	10.56	3.34

C.2. Refining Additional Baselines and Robustness to Initial Pose Estimates

Since PAD-Hand takes the predictions of a baseline model as input and refines them, it can be applied on top of different hand pose estimators. It is also robust to poor initial poses. To demonstrate this generality, we further evaluate PAD-Hand using the outputs of two recent baseline models, as shown in Tab. 6. In both cases, PAD-Hand consistently improves PA-MPJPE, MPJPE, and ACCEL, showing that our method serves as an effective refinement module across different baselines. We further test robustness by replacing 80% of a sequence with Gaussian noise, which increases the initial PA-MPJPE to 24.27. Under this severe corruption, our model reduces it to 6.53.

Table 6. Comparison to more SOTAs on DexYCB.

Method	PA-MPJPE↓	MPJPE↓	ACCEL↓
HaMeR [43]	4.70	16.71	7.14
HaMeR+Ours	4.60	11.09	3.37
HandOccNet [42]	5.80	14.0	9.03
HandOccNet+Ours	5.31	10.95	3.38

C.3. Statistical Significance Analysis

In this section, we verify that the performance gains obtained with \mathcal{L}_{EL} are statistically significant. To this end, we train two model variants on DexYCB using 5 random seeds: one with \mathcal{L}_{EL} and one without it. We then report the mean and standard deviation over the evaluation metrics. PAD-Hand achieves 4.61 ± 0.02 , 10.57 ± 0.02 , and 3.33 ± 0.01 for PA-MPJPE, MPJPE, and ACCEL, respectively, compared with 4.66 ± 0.01 , 10.65 ± 0.02 , and 3.36 ± 0.01 for the variant without the physics residual. These improvements are statistically significant under a paired t -test, with p -values of 0.001, 0.001, and 0.002, respectively.

C.4. Ablation Study on a Challenging Dataset

In this section, we evaluate the generalization ability of PAD-Hand and significance of \mathcal{L}_{EL} by testing it on a

dataset that was not used during training. For this, we employ a more challenging hands–object interaction dataset TACO [37], which contains diverse and complex manipulation sequences. The quantitative results in Table 7 show that PAD-Hand reduces PA-MPJPE from 8.37 mm to 8.02 mm and MPJPE from 25.13 mm to 24.38 mm. In addition, we observe a substantial improvement in acceleration error, which decreases from 5.47 mm/frame² to 1.84 mm/frame², indicating that our method produces smoother and more physically plausible motions in this challenging setting. Furthermore, using \mathcal{L}_{EL} shows stronger generalization, improving PA-MPJPE by 0.35 compared to 0.13 with \mathcal{L}_{data} alone (Tab. 7), ensuring physically grounded motion beyond the training distribution. In addition, when trained with only 10% of the DexYCB training data, adding this loss during WiLoR refinement effectively reduces performance degradation as shown in Table 8. The PA-MPJPE is 6.67 with \mathcal{L}_{EL} and degrades to 7.90 without \mathcal{L}_{EL} , demonstrating \mathcal{L}_{EL} ’s value as a physical regularizer under limited data.

Table 7. Generalization to unseen TACO [37]. S1 testing split is used for evaluation.

Method	PA-MPJPE↓	MPJPE↓	ACCEL↓
WiLoR	8.37	25.13	5.47
\mathcal{L}_{data}	8.24	25.02	2.44
$\mathcal{L}_{data} + \mathcal{L}_{EL}$ (Ours)	8.02	24.38	1.87

Table 8. Data efficiency under \mathcal{L}_{EL} on DexYCB.

Method	PA-MPJPE↓	MPJPE↓
\mathcal{L}_{data}	7.90	27.52
$\mathcal{L}_{data} + \mathcal{L}_{EL}$ (Ours)	6.67	25.84

C.5. Additional Results on Physical Plausibility

In this section, we further assess the effectiveness of our physics integration by computing the Euler-Lagrange residual as an additional metric of physical plausibility, defined as

$$\mathcal{R}(\mathbf{q}) = \frac{1}{T} \sum_{t=1}^T \|\mathbf{M}_t \ddot{\mathbf{q}} + \mathbf{c}_t + \mathbf{g}_t - \bar{\mathcal{F}}\|_1 \quad (28)$$

where $\bar{\mathcal{F}}$ is the pseudoforce computed from ground-truth motion data. Unlike the acceleration error, which primarily captures temporal smoothness, $\mathcal{R}(\mathbf{q})$ directly measures consistency with our physics formulation and therefore offers a complementary perspective on motion quality. As reported in Table 9, our physics-aware integration achieves

Table 9. **Ablation on physics integration on DexYCB.** † denotes that values are scaled by 10^{-3} .

Loss	PA-MPJPE↓	MPJPE↓	ACCEL↓	† $\mathcal{R}(q)$ ↓
*WiLoR [45]	4.88	12.75	6.70	19.12
SmoothFilter [61]	4.77	12.82	3.10	14.33
\mathcal{L}_{data}	4.65	10.62	3.36	15.16
$\mathcal{L}_{data} + \mathcal{L}_{EL}^D$	4.66	10.61	3.35	13.45
Ours	4.63	10.56	3.34	9.04

a lower residual than the deterministic counterpart, indicating that it produces motions that are more consistent with the underlying dynamics. Also, deterministic integration results in worse performance in PA-MPJPE (4.66 mm vs. 4.65 mm) in comparison to the data-driven approach (i.e., \mathcal{L}_{data}). For reference, we also include results for the SmoothFilter [61] which applies Gaussian smoothing to the WiLoR’s predictions and can be viewed as a naive baseline that implicitly drives high-frequency temporal variations (e.g., acceleration) toward zero, without modeling uncertainty in the underlying image-based estimates. Notably, although SmoothFilter outperforms PAD-Hand in terms of acceleration error, reconstruction accuracy is far below than us indicating the need to have $\mathcal{R}(q)$ metric for a better comparison in terms of physical plausibility.

D. Additional Visualizations

In this section, we present additional visualizations on three representative HO3D sequences to further illustrate the effectiveness of PAD-Hand (Figure 6). Across all examples, our pipeline not only refines the jittery motions produced by the baseline but also reliably flags segments with rapid changes through elevated variance estimates. For instance, in sequence III (a), there is a sudden change in the thumb motion and the corresponding variance maps in sequence III (b) assign a higher variance to the thumb joint, indicating an increased dynamic uncertainty in this fast-motion regime. Similar patterns are observed in the other sequences, where motion discontinuities and high-frequency jitter are both smoothed in the refined trajectories and highlighted by increased variance, providing an interpretable indicator of challenging image-based estimates. We also provide video versions of these sequences in “Hand Motion Comparison With Variance.mp4” file.

E. Discussion: PAD-Hand vs. DIP

Closest to our work is the DIP model proposed by Zhang *et al.* [69], which also employs a conditional diffusion model for hand motion recovery. However, our approach differs from DIP in several important aspects:

- DIP is trained with synthetic noise by adding Gaussian noise to the ground-truth motion, whereas PAD-Hand takes the predictions of an image-based estimator as input during training.
- DIP additionally outputs hand shape parameters and is supervised with a shape loss, while in our pipeline the shape parameters are kept fixed and we focus on refining the pose parameters.

Because of these design differences, a strictly fair comparison is not possible. Nevertheless, we report DIP results and use their officially released models for evaluation. Note that evaluating $\mathcal{R}(q)$ on HO3D is not feasible, since pose parameters for the test set are not publicly available. As shown in Tables 10 and 11, PAD-Hand consistently outperforms DIP in terms of reconstruction accuracy. Although DIP achieves a lower acceleration error on DexYCB (Table 10), its Euler-Lagrange residual $\mathcal{R}(q)$ remains higher than ours, indicating that our trajectories are better aligned with the underlying dynamics.

Table 10. **Comparison to DIP [69] on DexYCB.** † denotes that values are scaled by 10^{-3} .

Loss	PA-MPJPE↓	MPJPE↓	ACCEL↓	† $\mathcal{R}(q)$ ↓
*WiLoR [45]	4.88	12.75	6.70	19.12
WiLoR + DIP [69]	4.81	12.88	3.23	13.22
WiLoR + Ours	4.63	10.56	3.34	9.04

Table 11. **Comparison to DIP [69] on HO3D.**

Loss	PA-MPJPE↓	ACCEL↓
*WiLoR [45]	7.50	4.98
WiLoR + DIP [69]	7.55	3.03
WiLoR + Ours	7.43	2.71

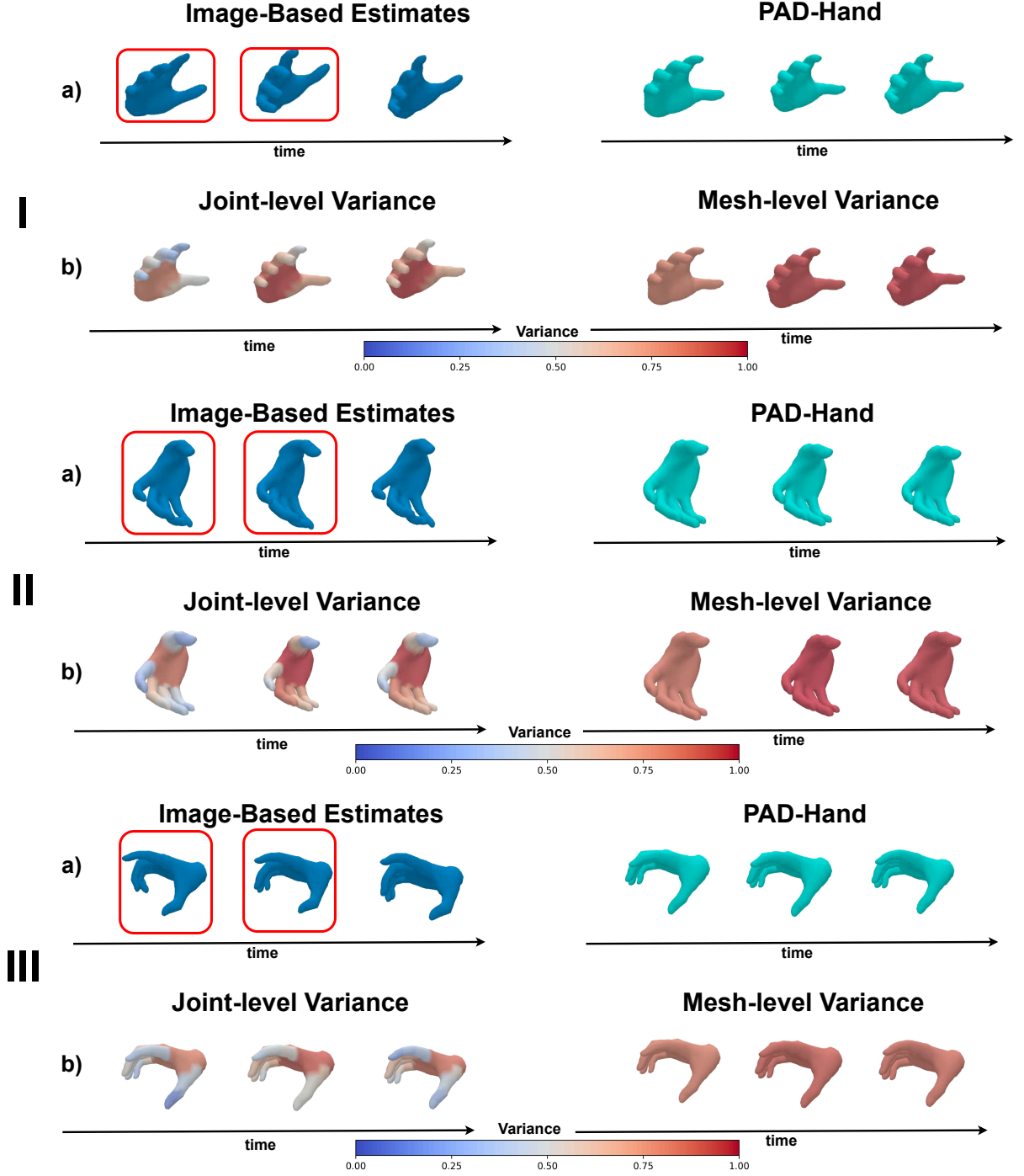


Figure 6. **Refined motion estimates by PAD-Hand with dynamic variance on HO3D.** We visualize three representative sequences (I–III). In each block, row (a) compares the original image-based motion estimates to the trajectories refined by PAD-Hand, while row (b) shows the corresponding variance estimations in terms of joint-level and mesh-level dynamic variance. The red boxes highlight frames where the image-based estimates exhibit strong jitter.