# CoDePose: Multi-View 3D Human Pose Estimation via Coupled 2D-3D Denoising Diffusion

Yanlu Cai[1], Yuxuan Liu[1], Weizhong Zhang[2,3], Yuan Wu[1], and Cheng Jin[1,4,⋆]

[1] College of Computer Science and Artificial Intelligence, Fudan University
[2] School of Data Science, Fudan University
[3] Shanghai Key Laboratory of Intelligent Information Processing
[4] MOE Laboratory for National Development and Intelligent Governance, Fudan University
{ylcai20, yuxuanliu24, weizhongzhang, wuyuan, jc}@fudan.edu.cn

**Abstract.** Recent multi-view 3D human pose estimation methods decompose the problem into per-image 2D pose detection and 3D lifting stages, which can naturally scale to multi-frame input and achieve strong results by leveraging temporal context. However, this 2D-3D lifting paradigm introduces two levels of geometric inconsistency: (i) projection inconsistency, i.e., the detected 2D keypoints may disagree with the 2D projections of the inferred 3D pose, and (ii) cross-view inconsistency, i.e., 2D poses across views are mutually inconsistent because off-the-shelf detectors operate independently per camera, so there may exist no single 3D pose whose projections match all detections. We observe that 2D detection errors closely follow a Gaussian distribution, meaning that detector outputs can be naturally viewed as noisy samples from a diffusion forward process. Based on this insight, we propose CoDePose, a coupled 2D-3D denoising diffusion framework that jointly performs multi-view denoising of the detected 2D poses and estimates the underlying 3D pose. As training proceeds, the generated joint distribution progressively approaches the real data distribution, leading to increased projection and cross-view consistency and more reliable 3D estimation. Unlike existing diffusion-based methods that condition on fixed 2D detections, CoDePose refines the 2D observations during the reverse process, preventing 2D detection errors and cross-view inconsistencies from being reinforced throughout the whole 3D pose generation. Experimental results on Human3.6M show that CoDePose achieves 12.1 mm MPJPE, a 19.9% improvement over the previous state-of-the-art method. Cross-dataset evaluation on HumanEva and MPI-INF-3DHP further demonstrates strong generalization.

**Keywords:** 3D Human Pose Estimation · Diffusion Models · Multi-View Fusion

⋆ Corresponding author.

## 1 Introduction

Recent multi-view 3D human pose estimation methods decompose the problem into per-image 2D pose detection followed by 3D lifting [24,33,34]. By operating on compact 2D pose sequences rather than raw images, these methods naturally scale to multi-frame input and achieve strong results by leveraging temporal context [2,16,17]. However, this 2D-3D lifting paradigm introduces two levels of geometric inconsistency. First, *projection inconsistency*: occlusion, motion blur, and background clutter cause per-view 2D detections to deviate from the true 2D projections of the underlying 3D pose. Second, *cross-view inconsistency*: off-the-shelf detectors such as CPN [4] and ViTPose [28] process each camera view independently, producing cross-view 2D predictions that are mutually inconsistent, so there may exist no single 3D pose whose projections match all detections. Existing multi-view fusion methods [5, 16, 17] work around these issues through confidence weighting and outlier dropping, but do not explicitly correct the 2D inputs themselves.

To address the above limitations, a natural idea is to build a coupled model that simultaneously denoises multi-view 2D poses and estimates the 3D pose: joint multi-view 2D denoising can enforce cross-view consistency, while the coupled 2D-3D modeling can enforce projection consistency. We observe that 2D detection errors closely follow a Gaussian distribution (Fig. 2), meaning that detector outputs can be naturally viewed as noisy samples from a diffusion forward process applied to the ground-truth 2D projections. This motivates **CoDePose** (Fig. 1), a coupled 2D-3D denoising diffusion framework that jointly performs multi-view denoising of the detected 2D poses and estimates the underlying 3D pose. As training proceeds, the generated joint distribution progressively approaches the real data distribution, leading to increased projection and cross-view consistency and more reliable 3D estimation. Unlike existing diffusion-based methods that condition on fixed 2D inputs, CoDePose refines the 2D observations during the reverse process, preventing detection errors and cross-view inconsistencies from being reinforced throughout generation. A key distinction from traditional diffusion models, which start all dimensions from pure noise at inference, is that CoDePose adopts an asymmetric initialization for inference: the 2D dimensions are initialized directly from detector outputs while only the 3D dimensions start from pure Gaussian noise. The reason is that as 2D detection errors follow a Gaussian distribution with small variance, the 2D detections can be naturally viewed as noisy samples from the diffusion trajectory close to 2D ground truth. This asymmetry allows the reverse process to converge rapidly, achieving accurate reconstruction in a single DDIM denoising step. To handle hard cases where complex noise patterns make 2D denoising difficult, we further introduce a per-joint uncertainty head and a body-coordinate consistency loss to enhance projection and cross-view consistency, respectively.

Experiments on Human3.6M show that CoDePose achieves 12.1 mm MPJPE, a 19.9% improvement over the previous state-of-the-art method [2] (15.1 mm) and a 10.4% improvement over 3D-only diffusion baseline (13.5 mm). Thanks to the asymmetric initialization, the reverse process converges in a single DDIM step

(12.15 mm). CoDePose also generalizes well across datasets, reducing MPJPE by 6.5% on HumanEva and 22.2% on MPI-INF-3DHP compared to FusionFormer.

Our contributions can be summarized as:

- A coupled 2D-3D denoising diffusion framework that jointly denoises multi-view 2D detections and predicts 3D pose, enforcing projection consistency through coupled 2D-3D modeling and cross-view consistency through joint multi-view denoising.
- An asymmetric noise strategy that initializes the 2D component from detector outputs at inference, enabling fast convergence in a single DDIM denoising step.
- State-of-the-art results on Human3.6M with strong cross-dataset generalization on HumanEva and MPI-INF-3DHP.

**Discussion.** Existing diffusion-based lifting methods [1, 7, 23] also employ iterative denoising for 3D pose estimation, but keep 2D poses as a fixed external condition that is never included in the diffusion state, so detection errors persist throughout the reverse process. In contrast, CoDePose incorporates 2D poses into the diffusion state, enabling joint refinement of 2D observations and 3D pose generation so that both types of inconsistency are progressively resolved.

## 2 Related Work

**Multi-view 3D pose estimation.** Multi-view setups resolve depth ambiguity through geometric constraints across cameras. Image-based methods operate on raw pixels and typically require known camera extrinsics: early works use differentiable triangulation [13, 14], epipolar constraints [9, 21, 32], volumetric fusion [27], or cross-view feature aggregation [11, 18, 22, 29], but scale poorly to multi-frame settings due to the cost of processing image features over time. Lifting-based methods instead operate on compact 2D pose sequences that naturally accommodate temporal context. Among camera-free approaches, FLEX [8] uses viewpoint-independent bone-angle representations but accumulates errors at distal joints; TransFusion [17] fuses cross-view features via epipolar-guided attention; MTF-Transformer [24] adaptively fuses multi-view and temporal features; SVTFormer [31] models intra-view joint and temporal relations before cross-view fusion; MVGFormer [16] injects multi-view geometric priors into attention; FusionFormer [2] proposes a unified feature fusion transformer; PoseIRM [3] applies invariant risk minimization for cross-camera generalization; and MV-SSM [5] replaces transformers with state-space models. All lifting-based methods produce deterministic outputs and consume 2D detections without correcting them.

**Diffusion models for 3D pose estimation.** DDPMs [10] learn to reverse a noise-corruption process, producing samples that match a target distribution. D3DP [23] first applied diffusion to 3D pose estimation, generating multiple 3D hypotheses from Gaussian noise conditioned on 2D keypoints and aggregating them via joint-wise reprojection. DiffPose [7] and DDHPose [1] disentangle bone

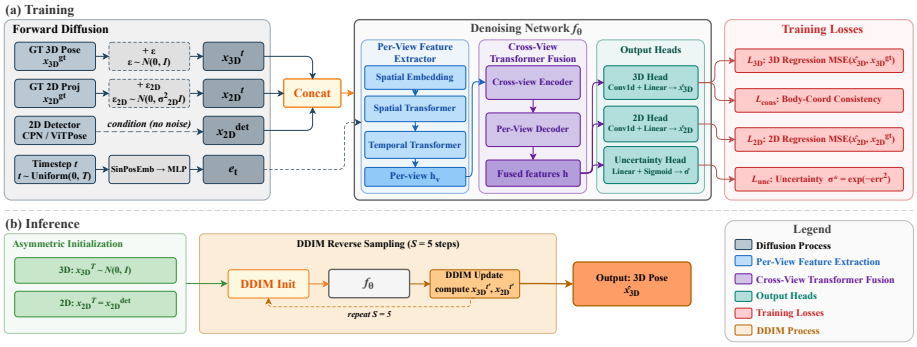


**Fig. 1:** Overview of CoDePose. During training, the forward process corrupts ground-truth 3D poses and 2D projections independently. The denoiser $f_\theta$ takes the concatenated noisy state and produces three outputs: denoised 3D pose, refined 2D projections, and per-joint uncertainty. Cross-view fusion exchanges information across $V$=4 cameras through an encoder-decoder transformer. At inference, the 3D component starts from Gaussian noise while the 2D component is initialized with detector outputs; DDIM reverse sampling recovers the final 3D pose in $S$=5 steps.

length and direction for more structured denoising. In these methods, 2D poses serve as fixed conditions; none include 2D poses in the diffusion state. CoDePose samples in coupled 2D-3D space with an uncertainty head embedded in the denoiser and supervised by 2D reconstruction error; see Supplementary Material for multi-hypothesis and uncertainty-aware baselines.

## 3 Method

### 3.1 Preliminaries

We review the standard diffusion formulation for 3D pose estimation [1, 7, 23]. Given a clean 3D pose $\mathbf{x}_0 \in \mathbb{R}^{J\times 3}$ with $J$ joints, the forward process adds noise over $T$ timesteps:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\,\mathbf{I}\big), \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ and $\beta_t$ follows a cosine schedule [20] with $t = 1, \ldots, T$. Following x-predict style diffusion [15], a network $f_\theta$ is trained to directly predict $\mathbf{x}_0$ from $(\mathbf{x}_t, \mathbf{c}, t)$, where $\mathbf{c}$ denotes the 2D conditioning signal. At inference, DDIM [26] recovers 3D poses from noise.

Prior diffusion-based pose methods [1, 7, 23] keep 2D input fixed throughout reverse sampling. When the 2D detector is inaccurate, detection errors are reinforced at every denoising step rather than corrected.

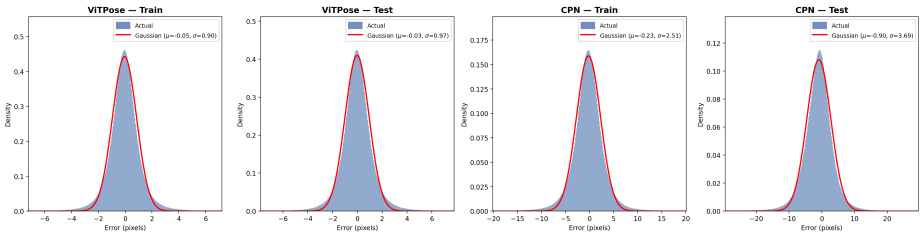


**Fig. 2:** Distribution of 2D detection errors (detector − GT, in pixels) on Human3.6M for ViTPose and CPN. Most (∼97%) errors follow a zero-mean Gaussian, supporting the coupled diffusion formulation: detector outputs can be treated as noisy samples from the forward process.

### 3.2 Coupled 2D-3D Diffusion

**Observation and motivation.** Most 2D detection errors closely follow a Gaussian distribution (Fig. 2). Since $\mathbf{x}_{\text{2D}}^{\text{det}} \approx \mathbf{x}_{\text{2D}}^{\text{gt}} + \boldsymbol{\delta}$ with $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{det}}^2 \mathbf{I})$ for most joints, detector output can be treated as a noisy sample from a diffusion forward process applied to the ground-truth 2D projection. Taken together, this observation motivates modeling 2D and 3D coordinates jointly in a coupled diffusion process. Although a small fraction of outliers (e.g., severe occlusion) deviate from this assumption, we handle them via the uncertainty head (Sec. 3.4).

**Coupled state.** Rather than diffusing 3D coordinates alone, we extend the diffusion state to include both 3D and 2D coordinates. For $V$ camera views, $T_f$ frames, and $J$ joints, the coupled diffusion state is:

$$\mathbf{x}_{\text{joint}}^{\text{gt}} = [\mathbf{x}_{\text{3D}}^{\text{gt}};\ \mathbf{x}_{\text{2D}}^{\text{gt}}] \in \mathbb{R}^{T_f \times V \times J \times 5}, \tag{2}$$

where $\mathbf{x}_{\text{3D}}^{\text{gt}} \in \mathbb{R}^{T_f \times V \times J \times 3}$ and $\mathbf{x}_{\text{2D}}^{\text{gt}} \in \mathbb{R}^{T_f \times V \times J \times 2}$. Since both components describe the same skeletal structure, they are diffused jointly and share the same joint topology.

**Forward process.** During training, both components are diffused from their respective ground truths with a shared noise schedule $\bar{\alpha}_t$:

$$\mathbf{x}_{\text{joint}}^{t} = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_{\text{joint}}^{\text{gt}} + \sqrt{1-\bar{\alpha}_t}\ \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} = \text{diag}(\mathbf{I}_3,\ \sigma_{\text{2D}}^2\, \mathbf{I}_2), \tag{3}$$

where $\sigma_{\text{2D}}$ controls 2D noise magnitude relative to the 3D component; to balance comparable noise levels between the two coordinate systems, we set $\sigma_{\text{2D}} = 1$ in main experiments. The denoiser $f_\theta$ predicts $\mathbf{x}_{\text{joint}}^{\text{gt}}$ from $(\mathbf{x}_{\text{joint}}^{t},\ \mathbf{c},\ t)$, where $\mathbf{c} = \mathbf{x}_{\text{2D}}^{\text{det}}$ is the detector output serving as conditioning.

**Asymmetric inference.** For efficient inference with fewer reverse steps, we adopt DDIM [26] as our sampling strategy. Given a sampling step from $t$ to $t'$ $(t > t')$:

$$\mathbf{x}_{\text{joint}}^{t'} = \sqrt{\bar{\alpha}_{t'}}\, \hat{\mathbf{x}}_{\text{joint}}^{0} + \sqrt{1-\bar{\alpha}_{t'}-\sigma_t^2} \cdot \frac{\mathbf{x}_{\text{joint}}^{t} - \sqrt{\bar{\alpha}_t}\, \hat{\mathbf{x}}_{\text{joint}}^{0}}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\, \boldsymbol{\epsilon}, \tag{4}$$

where $\hat{\mathbf{x}}^0_{\text{joint}} = f_\theta(\mathbf{x}^t_{\text{joint}},\ \mathbf{c},\ t)$ is the denoiser prediction at the current step, $\sigma_t = \eta\sqrt{(1-\bar{\alpha}_{t'}/\bar{\alpha}_t)(1-\bar{\alpha}_{t'})/(1-\bar{\alpha}_t)}$, and $\eta$ controls stochasticity ($\eta$=0 for deterministic DDIM). The denoiser $f_\theta$ is detailed in Sec. 3.3. Unlike standard diffusion models that initialize all dimensions from pure noise, we initialize the reverse process *asymmetrically*:

$$\mathbf{x}^T_{\text{3D}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \text{and} \quad \mathbf{x}^T_{\text{2D}} = \mathbf{x}^{\text{det}}_{\text{2D}}, \tag{5}$$

where the 3D component starts from pure noise and the 2D component from detector outputs. Because detector errors are approximately Gaussian for most joints, $\mathbf{x}^{\text{det}}_{\text{2D}}$ can be viewed as an intermediate sample on the forward trajectory near ground truth, avoiding train-test mismatch. As a consequence of this asymmetry, the reverse process can converge in as few as a single DDIM step.

### 3.3 Cross-View Transformer Denoiser

To denoise the coupled 2D-3D state across multiple views, our cross-view transformer denoiser $f_\theta$ (Fig. 1) follows a 3-stage architecture: per-view spatio-temporal feature extraction, cross-view transformer fusion, and multi-head prediction.

**Per-view feature extraction.** Given a multi-view sequence, we process each view independently with a PoseFormer-style [34] spatio-temporal Transformer. For view $v$, joint inputs are linearly projected to $d$-dimensional embeddings and augmented with learnable spatial positional embeddings. We apply $L$ spatial Transformer blocks within each frame to capture intra-frame kinematic dependencies among $J$ joints, aggregate joint tokens into a pose-level token per frame, and apply $L$ temporal Transformer blocks to model motion dynamics, yielding per-view features $\mathbf{h}_v \in \mathbb{R}^{T_f \times d_p}$.

**Cross-view fusion.** To fuse information across $V$ cameras, we use a Transformer encoder–decoder on the per-view feature sequences. The encoder concatenates $\{\mathbf{h}_v\}_{v=1}^V$ into a sequence of length $V \times T_f$ and applies self-attention for global cross-view and cross-frame interactions. The decoder uses each view sequence as queries and attends to the encoder memory via cross-attention, allowing each view to retrieve complementary evidence from other cameras. We reshape the fused representation $\mathbf{h}$ back to the per-view layout for prediction.

**Triple output heads.** The fused features $\mathbf{h}$ are projected through three independent heads:

$$\hat{\mathbf{x}}_{\text{3D}} = \phi_{\text{3D}}(\mathbf{h}), \quad \hat{\mathbf{x}}_{\text{2D}} = \phi_{\text{2D}}(\mathbf{h}), \quad \hat{\sigma} = \text{sigmoid}(\phi_{\text{unc}}(\mathbf{h})), \tag{6}$$

where each head $\phi$ is a temporal convolution followed by a linear projection. The 3D head outputs $\hat{\mathbf{x}}_{\text{3D}} \in \mathbb{R}^{V\times J\times 3}$ in camera coordinates of the center frames; the 2D head outputs refined projections $\hat{\mathbf{x}}_{\text{2D}} \in \mathbb{R}^{V\times J\times 2}$; and the uncertainty head outputs per-joint confidence $\hat{\sigma} \in [0,1]^{V\times J}$. Sigmoid activation maps outputs to $[0,1]$, with values near 1 indicating reliable 2D estimates and values near 0 flagging unreliable joints. The uncertainty head suppresses a key side effect of coupled 2D–3D modeling: by down-weighting unreliable joints, it prevents outlier detections from corrupting 3D reconstruction, which is essential for the coupled formulation to outperform the 3D-only baseline (Sec. 4.3).

### 3.4 Training Objective

The model is trained with a composite loss:

$$\mathcal{L} = w_t \cdot \big(\mathcal{L}_{\text{3D}} + \lambda_{\text{2D}}\,\mathcal{L}_{\text{2D}} + \lambda_{\text{unc}}\mathcal{L}_{\text{unc}} + \lambda_{\text{cons}}\,\mathcal{L}_{\text{cons}}\big), \tag{7}$$

where the components are defined below.

**3D and 2D regression losses.** The 3D and 2D heads are supervised with per-joint mean squared error:

$$\mathcal{L}_{\text{3D}} = \frac{1}{J}\sum_{j=1}^{J}\big\|\hat{\mathbf{x}}_{\text{3D}}^{(j)} - \mathbf{x}_{\text{3D}}^{\text{gt}(j)}\big\|^2, \quad \mathcal{L}_{\text{2D}} = \frac{1}{J}\sum_{j=1}^{J}\big\|\hat{\mathbf{x}}_{\text{2D}}^{(j)} - \mathbf{x}_{\text{2D}}^{\text{gt}(j)}\big\|^2. \tag{8}$$

We set $\lambda_{\text{2D}}$=0.1 to prevent the 2D gradient from dominating (Sec. 4.3).

**Uncertainty loss.** The uncertainty head uses a self-calibrated target:

$$\mathcal{L}_{\text{unc}} = \frac{1}{J}\sum_{j=1}^{J}\big\|\hat{\sigma}^{(j)} - \exp\big(-\|\hat{\mathbf{x}}_{\text{2D}}^{(j)} - \mathbf{x}_{\text{2D}}^{\text{gt}(j)}\|^2\big)\big\|^2. \tag{9}$$

The target $\sigma^* = \exp(-\|\hat{\mathbf{x}}_{\text{2D}} - \mathbf{x}_{\text{2D}}^{\text{gt}}\|^2)$ approaches 1 for accurate 2D predictions and decays toward 0 as error grows, providing a stable bootstrap: early in training $\sigma^*$ stays near 0, then increases as 2D predictions improve.

**Cross-view consistency via body coordinate.** CoDePose operates without camera extrinsics, so the 3D head predicts a separate pose per view in its own camera coordinate system. Predictions of the same person from different views should represent the same physical pose but differ by an unknown rigid transformation determined by relative camera configuration; direct comparison in camera coordinates is therefore meaningless. We instead construct a *body coordinate system* purely from the predicted skeleton, replacing the camera-dependent rigid transform with a canonical, view-invariant representation.

Let $\mathbf{p}_j = \hat{\mathbf{x}}_{\text{3D}}^{(j)}$ denote the predicted 3D position of joint $j$. We take the Hip (root) joint $\mathbf{p}_{\text{Hip}}$ as the origin and build an orthonormal basis in three steps. First, the $x$-axis is the unit vector from the left hip to the right hip:

$$\mathbf{e}_x = \frac{\mathbf{p}_{\text{RHip}} - \mathbf{p}_{\text{LHip}}}{\|\mathbf{p}_{\text{RHip}} - \mathbf{p}_{\text{LHip}}\|}. \tag{10}$$

Second, we compute a preliminary $y$-direction from the Spine joint toward the Hip and orthogonalize it against $\mathbf{e}_x$ via Gram–Schmidt:

$$\mathbf{e}'_y = \mathbf{p}_{\text{Spine}} - \mathbf{p}_{\text{Hip}}, \quad \mathbf{e}_y = \frac{\mathbf{e}'_y - (\mathbf{e}'_y \cdot \mathbf{e}_x)\,\mathbf{e}_x}{\|\mathbf{e}'_y - (\mathbf{e}'_y \cdot \mathbf{e}_x)\,\mathbf{e}_x\|}. \tag{11}$$

Third, the $z$-axis completes the right-handed basis via the cross product:

$$\mathbf{e}_z = \mathbf{e}_x \times \mathbf{e}_y. \tag{12}$$

The rotation matrix $\mathbf{R} = [\mathbf{e}_x;\ \mathbf{e}_y;\ \mathbf{e}_z]^\top$ together with $\mathbf{p}_{\text{Hip}}$ define a canonical body coordinate system invariant to camera viewpoint. Transforming each per-view prediction into this common frame enables direct cross-view comparison. The consistency loss is:

$$\mathcal{L}_{\text{cons}} = \frac{1}{\binom{V}{2}} \sum_{u \neq v} \left\| \mathbf{R}_u(\hat{\mathbf{x}}_{3\text{D},u} - \mathbf{p}_{\text{Hip},u}) - \mathbf{R}_v(\hat{\mathbf{x}}_{3\text{D},v} - \mathbf{p}_{\text{Hip},v}) \right\|^2, \tag{13}$$

where $u$ and $v$ index distinct views. This term acts as a consistency regularizer rather than a hard geometric constraint: under heavy occlusion or extreme poses, errors concentrate on distal joints while root and spine joints remain reliable, so the body coordinate system can still be established. Because occlusion rarely affects all views simultaneously, remaining views together with the $T_f$=27-frame temporal input can recover many occluded joints. For extreme poses where the hip–spine direction becomes unreliable, we clamp $\mathcal{L}_{\text{cons}}$ to prevent unstable gradients from dominating training.

**Time-dependent loss weighting.** Predictions at low-noise timesteps are more informative, so we upweight them:

$$w_t = \min\Big(1 + \frac{\bar{\alpha}_t}{\sqrt{1-\bar{\alpha}_t}},\ 3\Big), \tag{14}$$

where clipping at 3 prevents gradient explosion at small $t$ where $\bar{\alpha}_t \to 1$.

## 4 Experiments

### 4.1 Setup

**Dataset.** We evaluate primarily on Human3.6M [12] following the standard protocol: subjects S1, S5, S6, S7, S8 for training, S9 and S11 for testing, with 4-camera multi-view setups and the 17-joint skeleton. Temporal windows span $T_f$=27 frames. To assess transfer beyond the training domain, we further evaluate cross-dataset generalization on HumanEva [25] and MPI-INF-3DHP [19].

**2D input.** Following FusionFormer [2], we employ CPN [4] for fair comparison with 2D-to-3D lifting methods and ViTPose [28] for fair comparison with end-to-end image-to-3D methods that benefit from stronger built-in detectors. In both settings, $\mathbf{x}_{2\text{D}}^{\text{det}}$ serves as the conditioning signal throughout training and inference. The 2D diffusion component starts from ground-truth projections $\mathbf{x}_{2\text{D}}^{\text{gt}}$ during training and from $\mathbf{x}_{2\text{D}}^{\text{det}}$ during inference, as described in Sec. 3.2.

**Metrics.** We report Mean Per Joint Position Error (MPJPE, mm) and Procrustes-aligned MPJPE (P-MPJPE).

**Implementation.** We train 200 epochs with AdamW (learning rate $4\times10^{-4}$, weight decay 0.1) and exponential decay at 0.99 per epoch. Batch size is 600 across 12 Nvidia GeForce RTX 4090 GPUs. The per-view backbone uses $L$=2 spatial and temporal layers with joint embedding dimension $d$=32. We set $\lambda_{\text{unc}}$=1 and $\lambda_{\text{cons}} = 0.1$. Diffusion uses $T$=1000 training steps with a cosine schedule and $S$=5 DDIM steps at inference ($\eta$=0). The 2D noise scale is $\sigma_{2\text{D}}$=1.0.

**Table 1:** Per-action MPJPE (mm) on Human3.6M (S9 & S11, 4 cameras). "*" denotes end-to-end image-to-3D methods that do not use a separate 2D estimator. $T$ is the number of input frames. † denotes our 3D-only diffusion baseline.

| Method | Input | Dir. | Disc. | Eat | Greet | Phone | Photo | Pose | Purch. | Sit | SitD. | Smoke | Wait | WalkD. | Walk | WalkT. | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Monocular methods* | | | | | | | | | | | | | | | | | |
| PoseFormer [34] | (CPN, $T$=81) | 41.5 | 44.8 | 39.8 | 42.5 | 46.5 | 51.6 | 42.1 | 42.0 | 53.3 | 60.7 | 45.5 | 43.3 | 46.1 | 31.8 | 32.2 | 44.3 |
| D3DP [23] | (CPN, $T$=243) | 37.3 | 39.5 | 35.6 | 37.8 | 41.3 | 48.2 | 39.1 | 37.6 | 49.9 | 52.8 | 41.2 | 39.2 | 39.4 | 27.2 | 27.1 | 39.5 |
| DDHPose [1] | (CPN, $T$=243) | 36.4 | 39.5 | 34.9 | 37.6 | 40.1 | 45.9 | 37.8 | 37.8 | 51.5 | 52.2 | 40.8 | 38.3 | 38.3 | 27.0 | 27.0 | 39.0 |
| DiffPose [7] | (CPN, $T$=243) | 33.2 | 36.6 | 33.0 | 35.6 | 37.6 | 45.1 | 35.7 | 35.5 | 46.4 | 49.9 | 37.3 | 35.6 | 36.5 | 24.4 | 24.1 | 36.9 |
| MotionBERT [35] | (CPN, $T$=243) | 36.1 | 37.5 | 35.8 | 32.1 | 40.3 | 46.3 | 36.1 | 35.3 | 46.9 | 53.9 | 39.5 | 36.3 | 35.8 | 25.1 | 25.3 | 37.5 |
| *Image-based multi-view methods* | | | | | | | | | | | | | | | | | |
| DeepFuse [11] | (*, $T$=1) | 26.8 | 32.0 | 25.6 | 52.1 | 33.3 | 42.3 | 25.8 | 25.9 | 40.5 | 76.6 | 39.1 | 54.5 | 35.9 | 25.1 | 24.2 | 37.5 |
| Canonical Fusion [22] | (*, $T$=1) | 27.3 | 32.1 | 25.0 | 26.5 | 29.3 | 35.4 | 28.8 | 31.6 | 36.4 | 31.7 | 31.2 | 29.9 | 26.9 | 33.7 | 30.4 | 30.2 |
| Prob. Tri. [14] | (*, $T$=1) | 24.0 | 25.4 | 26.6 | 30.4 | 32.1 | 20.1 | 20.5 | 36.5 | 40.1 | 29.5 | 27.4 | 27.6 | 20.8 | 24.1 | 22.0 | 27.8 |
| Epipolar Trans. [9] | (*, $T$=1) | 25.7 | 27.7 | 23.7 | 24.8 | 26.9 | 31.4 | 24.9 | 26.5 | 28.8 | 31.7 | 28.2 | 26.4 | 23.6 | 28.3 | 23.5 | 26.9 |
| Crossview Fusion [21] | (*, $T$=1) | 24.0 | 26.7 | 23.2 | 24.3 | 24.8 | 22.8 | 24.1 | 28.6 | 32.1 | 26.9 | 31.0 | 25.6 | 25.0 | 28.0 | 24.4 | 26.2 |
| TransFusion [17] | (*, $T$=1) | 24.4 | 26.4 | 23.4 | 21.1 | 25.2 | 23.2 | 24.7 | 33.8 | 29.8 | 26.4 | 26.8 | 24.2 | 23.2 | 26.1 | 23.3 | 25.8 |
| Multi-PPT [18] | (*, $T$=1) | 21.8 | 26.5 | 21.0 | 22.4 | 23.7 | 23.1 | 23.2 | 27.9 | 30.7 | 24.6 | 26.7 | 23.3 | 21.2 | 25.3 | 22.6 | 24.4 |
| Learnable Tri. [13] | (*, $T$=1) | 19.9 | 20.0 | 18.9 | 18.5 | 20.5 | 19.4 | 18.4 | 22.1 | 22.5 | 28.7 | 21.2 | 20.8 | 19.7 | 22.1 | 20.2 | 20.8 |
| AdaFuse [32] | (*, $T$=1) | 17.8 | 19.5 | 17.6 | 20.7 | 19.3 | 16.8 | 18.9 | 20.2 | 25.7 | 20.1 | 19.2 | 20.5 | 17.2 | 20.5 | 17.3 | 19.5 |
| MvP [29] | (*, $T$=1) | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 18.6 |
| *Lifting-based multi-view methods* | | | | | | | | | | | | | | | | | |
| FLEX [8] | (CPN, $T$=27) | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 31.7 |
| FLEX [8] | (ResNet152, $T$=27) | 23.1 | 28.8 | 26.8 | 28.1 | 31.6 | 37.1 | 25.7 | 31.4 | 36.5 | 39.6 | 35.0 | 29.5 | 35.6 | 26.8 | 26.4 | 30.9 |
| SGraFormer [30] | (CPN, $T$=27) | 26.5 | 28.3 | 23.0 | 25.9 | 27.2 | 31.0 | 25.4 | 27.2 | 28.6 | 33.8 | 28.6 | 25.6 | 30.1 | 27.1 | 26.5 | 27.6 |
| MTF-Transformer [24] | (CPN, $T$=27) | 23.1 | 25.4 | 24.7 | 24.5 | 27.9 | 28.3 | 23.9 | 24.6 | 30.7 | 35.7 | 25.8 | 24.2 | 28.4 | 22.8 | 23.1 | 26.2 |
| SVTFormer [31] | (CPN, $T$=27) | 24.5 | 27.5 | 23.2 | 24.4 | 25.8 | 28.7 | 23.8 | 26.4 | 30.0 | 32.7 | 26.0 | 23.9 | 27.5 | 22.8 | 23.2 | 26.0 |
| UPose3D [6] | (CPN, $T$=27) | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 25.6 |
| FusionFormer [2] | (CPN, $T$=27) | 22.2 | 25.3 | 22.9 | 23.6 | 26.0 | 27.0 | 22.4 | 23.9 | 30.4 | 33.9 | 25.6 | 22.8 | 26.3 | 22.3 | 22.8 | 25.4 |
| PoseIRM [3] | (CPN, $T$=27) | 22.2 | 24.6 | 22.9 | 23.2 | 26.0 | 27.0 | 22.2 | 23.7 | 29.3 | 33.6 | 25.6 | 22.8 | 25.8 | 22.5 | 22.5 | 25.1 |
| FusionFormer [2] | (ViTPose, $T$=27) | 15.7 | 15.6 | 13.0 | 15.9 | 13.9 | 15.6 | 14.9 | 15.5 | 15.5 | 16.3 | 14.3 | 15.2 | 16.9 | 14.6 | 14.6 | 15.1 |
| 3D-only diff.† | (ViTPose, $T$=27) | 12.8 | 13.7 | 12.7 | 15.0 | 13.0 | 16.4 | 12.9 | 14.7 | 15.1 | 17.1 | 13.2 | 14.2 | 16.1 | 12.6 | 12.6 | 13.5 |
| **CoDePose (ours)** | (CPN, $T$=27) | 21.2 | 23.8 | 23.0 | 22.5 | 26.6 | 25.7 | 21.4 | 24.6 | 30.5 | 31.6 | 25.2 | 22.6 | 26.1 | 22.1 | 22.1 | 24.8 |
| **CoDePose (ours)** | (ViTPose, $T$=27) | **11.7** | **12.1** | **10.8** | **13.2** | **11.3** | **14.0** | **11.2** | **12.5** | **12.3** | **14.9** | **11.3** | **12.3** | **13.1** | **11.2** | **11.0** | **12.1** |

### 4.2 Comparison with State of the Art

Table 1 compares CoDePose with monocular, image-based, and lifting-based multi-view methods on Human3.6M using both CPN and ViTPose as 2D inputs.

**Comparison with lifting-based methods.** When CPN detections are used as input, CoDePose achieves 24.8 mm, outperforming all lifting-based methods, including PoseIRM [3] (25.1 mm) and FusionFormer [2] (25.4 mm). With ViTPose, CoDePose reaches **12.1 mm**—a 19.9% improvement over FusionFormer (15.1 mm) with the same 2D input. Relative to our 3D-only diffusion baseline (13.5 mm), coupled 2D-3D denoising reduces MPJPE by 1.4 mm (10.4%), confirming that joint 2D-3D denoising benefits 3D estimation beyond diffusion alone.

**Comparison with image-based methods.** Despite operating on detected 2D keypoints, CoDePose (12.1 mm) surpasses all image-based methods in Tab. 1, including MvP [29] (18.6 mm) and AdaFuse [32] (19.5 mm). Supplementary Material provides an extended comparison of image-based and lifting-based temporal settings.

**Generalization across datasets.** Following FusionFormer [2], we finetune CoDePose pretrained on Human3.6M for 10 epochs on HumanEva [25] and MPI-INF-3DHP [19]. Table 2 compares generalization with PoseFormer [34], MTF-Transformer [24], and FusionFormer under the same protocol. CoDePose achieves the best results on both datasets: 14.4 mm on HumanEva and 4.2 mm on MPI-INF-3DHP, outperforming FusionFormer by 6.5% and 22.2%, respectively. The

**Table 2:** Cross-dataset generalization. All models are pretrained on Human3.6M and finetuned for 10 epochs. $T$ denotes the number of input frames.

| Method | HumanEva | | MPI-INF-3DHP | |
|---|---|---|---|---|
| | MPJPE ↓ | P-MPJPE ↓ | MPJPE ↓ | P-MPJPE ↓ |
| PoseFormer [34] ($T$=27) | 35.9 | 30.1 | 38.5 | 31.8 |
| MTF-Trans. [24] ($T$=3) | 22.8 | 18.2 | 14.6 | 10.9 |
| FusionFormer [2] ($T$=3) | 15.4 | 12.7 | 5.4 | 3.2 |
| **CoDePose (ours)** ($T$=3) | 14.4 | 11.6 | 4.2 | 2.6 |

**Table 3:** Component analysis. MPJPE (mm) when removing each component from the full model. ViTPose, $\sigma_{2D}$=1.0.

| Configuration | 2D GT diff. | $\lambda_{2D}$=0.1 | Body coord. | Unc. head | MPJPE ↓ |
|---|---|---|---|---|---|
| **Full model** | ✓ | ✓ | ✓ | ✓ | **12.1** |
| w/o diffusion | - | - | - | - | 16.7 |
| 3D-only baseline | | - | - | - | 13.5 |
| − $\lambda_{2D}$=0.1 | ✓ | | ✓ | ✓ | 12.4 |
| − Body coord. | ✓ | ✓ | | ✓ | 12.7 |
| − Unc. head | ✓ | ✓ | ✓ | | 14.1 |

gain on MPI-INF-3DHP is particularly notable given its indoor and outdoor scenes that differ substantially from Human3.6M, indicating that coupled 2D-3D diffusion learns transferable representations.

### 4.3 Ablation Studies

To ensure consistent comparison across component removals, all ablations use Human3.6M with ViTPose detections.

**Component analysis.** Table 3 evaluates each component by removing it from the full model (12.1 mm). Replacing iterative denoising with single-pass deterministic regression (w/o diffusion) yields 16.7 mm (+4.6 mm), worse than FusionFormer (15.1 mm). Diffusion acts as a training-time curriculum over diverse noise levels, gradually aligning 2D and 3D components via a shared noise schedule rather than forcing reconciliation in one forward pass.

Removing the uncertainty head causes the largest degradation (+2.0 mm, to 14.1 mm), even worse than the 3D-only baseline. Removing body-coordinate consistency (+0.6 mm) weakens cross-view agreement. Setting $\lambda_{2D}$=1.0 (+0.3 mm) lets the 2D gradient dominate; $\lambda_{2D}$=0.1 keeps 2D refinement auxiliary and focuses learning on 3D estimation. The roles of diffusion, coupling, and stabilizer modules are summarized in Sec. 4.4.

**Asymmetric inference.** As established in Sec. 3.2, detector errors approximate Gaussian noise for most joints, so detector outputs lie on the forward

**Table 4:** Symmetric vs. asymmetric inference initialization. MPJPE (mm) at different DDIM steps $S$. ViTPose, $\sigma_{2D}$=1.0.

| Initialization | $S$=1 | $S$=5 | $S$=20 |
|---|---|---|---|
| Symmetric (both from noise) | 22.15 | 15.31 | 12.10 |
| **Asymmetric (ours)** | 12.15 | 12.07 | **12.06** |

**Table 5:** Effect of 2D noise scale $\sigma_{2D}$ on MPJPE (mm). ViTPose, clean-GT start, $S$=5, $N$=1.

| $\sigma_{2D}$ | MPJPE ↓ | P-MPJPE ↓ |
|---|---|---|
| 0.0 (no noise) | 13.5 | 9.78 |
| 0.25 | 12.4 | 8.94 |
| 0.5 | 12.3 | 8.91 |
| 1.0 | **12.1** | **8.86** |
| 2.0 | 12.6 | 8.96 |

trajectory from 2D ground truth. Table 4 compares asymmetric initialization (Eq. (5)) with symmetric initialization from pure noise. With sufficient steps ($S$=20), symmetric inference converges to comparable performance (12.10 mm vs. 12.06 mm), confirming that the denoiser can recover 2D from scratch given enough iterations. At low step counts, the gap widens: at $S$=1, symmetric yields 22.15 mm versus 12.15 mm for asymmetric. Because detector outputs already lie near ground truth on the forward trajectory, the 2D component requires minimal correction, enabling accurate reconstruction in as few as a single DDIM step.

**2D noise scale.** $\sigma_{2D}$ controls corruption magnitude applied to ground-truth 2D projections during training. Table 5 shows that $\sigma_{2D}$=0 (no noise) degrades to the 3D-only baseline (13.5 mm) because the denoiser never learns 2D refinement. Performance improves with increasing $\sigma_{2D}$, peaking at 1.0 (12.1 mm). Because both components share $\bar{\alpha}_t$, $\sigma_{2D}$=1.0 aligns 2D and 3D noise magnitudes during training. $\sigma_{2D}$=2.0 degrades performance (+0.5 mm).

**DDIM sampling steps.** Table 6 shows rapid convergence: even $S$=1 achieves 12.15 mm, only 0.09 mm worse than $S$=20, thanks to asymmetric initialization placing the input near the data manifold from the first step. Because $S$=5 matches the accuracy of $S$=10 at half the computational cost, we adopt it as the default inference setting.

**Number of hypotheses.** Diffusion models support multi-hypothesis estimation by generating multiple independent noise samples $\mathbf{x}_{3D}^T$ and averaging predictions. Table 6 shows that increasing $N$ from 1 to 5 yields a marginal 0.05 mm gain, with no further improvement at $N$=10. This contrasts with monocular diffusion methods such as D3DP [23], where multi-hypothesis aggregation is critical: monocular 2D-to-3D lifting is ill-posed and requires diverse hypotheses to cover ambiguity. In our multi-view setting, $V$=4 camera observations already provide strong geometric constraints that largely resolve depth ambiguity. Given

**Table 6:** Ablation on DDIM sampling steps $S$ and number of hypotheses $N$ for ViTPose ($\sigma_{2D}$=1.0).

| Setting | MPJPE ↓ | P-MPJPE ↓ |
|---|---|---|
| **Vary** $S$ **(fix** $N$=1) | | |
| $S = 1$ | 12.15 | 8.93 |
| $S = 2$ | 12.10 | 8.89 |
| $S = 5$ | 12.07 | 8.86 |
| $S = 10$ | 12.07 | 8.85 |
| $S = 20$ | **12.06** | **8.85** |
| **Vary** $N$ **(fix** $S$=5) | | |
| $N = 1$ | 12.07 | 8.86 |
| $N = 2$ | 12.05 | 8.85 |
| $N = 5$ | 12.02 | 8.83 |
| $N = 10$ | **12.02** | **8.83** |

**Table 7:** Effect of camera views on MPJPE (mm). ViTPose detections with temporal length $T_f$=27.

| Views | MPJPE ↓ |
|---|---|
| 4 | 12.1 |
| 3 | 17.2 |
| 2 | 21.4 |

the negligible benefit of multi-hypothesis aggregation under these constraints, we use $N$=1 by default for efficiency.

**View scalability.** Since standard benchmarks and practical deployments alike typically involve at most four cameras, Table 7 reports 2-, 3-, and 4-view settings; performance drops to 17.2 mm with 3 views and 21.4 mm with 2 views.

### 4.4 Robustness and Efficiency Analysis

**Attribution of performance gains.** We decompose improvement as follows. **(1) Diffusion.** It provides a training-time curriculum over diverse noise levels; removing diffusion (Tab. 3) yields 16.7 mm (+4.6 mm). **(2) Coupled 2D–3D**

**Table 8:** Heavy-tailed 2D noise effect on MPJPE (mm). Train/test detector pairs under CPN-based setup.

| Train | Test | MPJPE ↓ |
|---|---|---|
| CPN | CPN | 24.8 |
| CPN | Stud.-T | 32.7 |
| Stud.-T | Stud.-T | 25.7 |

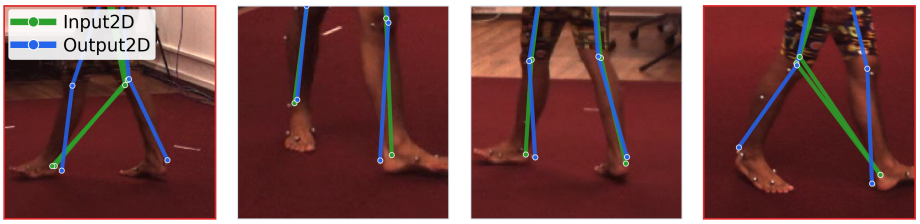


**Fig. 3:** Error propagation under noisy 2D detections. One camera view contains substantially noisier 2D inputs (left). CoDePose corrects corrupted 2D observations during coupled denoising rather than propagating them into the 3D prediction, maintaining projection consistency across views.

**Table 9:** Efficiency comparison on Human3.6M (ViTPose, $T_f$=27).

| Method | FLOPs | Params | MPJPE ↓ | FPS ↑ | Latency ↓ |
|---|---|---|---|---|---|
| MTF-Transformer | 1.83G | 7.50M | 15.1 | 82.41 | 12.13 ms |
| Ours ($S$=1) | 1.84G | 7.23M | 12.2 | 77.10 | 12.97 ms |
| Ours ($S$=5) | 9.23G | 7.23M | **12.1** | 15.19 | 65.82 ms |

**design.** (a) It refines 2D poses, improving projection and cross-view consistency, as detailed in the supplementary triangulation analysis. (b) Table 3 shows about 10% gain over the 3D-only baseline using 2D only as a fixed condition (13.5 mm → 12.1 mm). (c) It suppresses erroneous 2D detections rather than propagating them, as shown in Fig. 3.**(3) Uncertainty head and body-coordinate consistency.** Naive coupled design may propagate erroneous 2D predictions into 3D estimation. The uncertainty head suppresses this effect, and the body-coordinate term stabilizes cross-view consistency; as analyzed in Sec. 4.3, these modules stabilize the coupled framework rather than acting as independent add-ons.

**Heavy-tailed and non-Gaussian noise.** Because of data scarcity, we simulate failure-prone scenarios by injecting heavy-tailed Student-T noise ($\nu$=3). Table 8 shows that CPN/Student-T train-test mismatch degrades performance to 32.7 mm, while Student-T/Student-T recovers 25.7 mm, close to 24.8 mm under CPN/CPN, indicating adaptation to non-Gaussian, heavy-tailed detection errors when train and test distributions match.

**Computational efficiency.** Table 9 compares inference cost on Human3.6M. With $S$=1 DDIM step, CoDePose has similar FLOPs and FPS to MTF [24] but substantially lower MPJPE shown in Tab. 6. With $S$=5, we reach 12.1 mm at higher compute. With batch size 200, pose-lifting throughput exceeds 1000 FPS; in practice, image-to-2D pose estimation is the bottleneck.

### 4.5 Qualitative Comparison

Figure 4 shows qualitative comparisons on ten Human3.6M examples. Each panel displays the input image, ground-truth 3D pose, CoDePose prediction, and MTF-

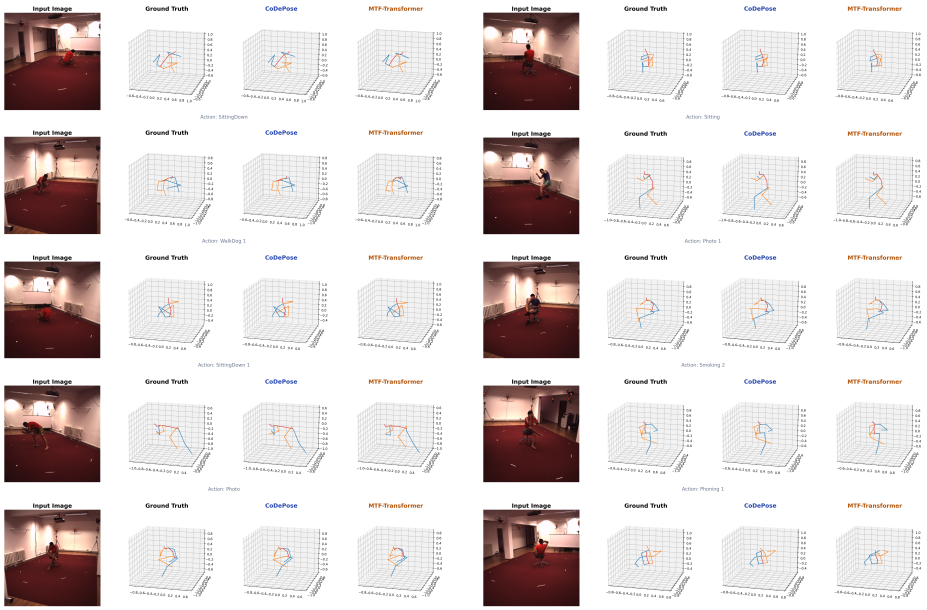


**Fig. 4:** Qualitative comparison on Human3.6M (S9 & S11). Each panel shows (left to right): input image, ground-truth 3D pose, CoDePose prediction, and MTF-Transformer prediction. CoDePose produces more accurate 3D poses across diverse actions and viewpoints.

Transformer prediction from left to right. These examples illustrate how the two types of consistency enforced by CoDePose—projection consistency and cross-view consistency—lead to improved 3D estimation in challenging scenarios.

**Seated, crouching, and bent-over poses.** The examples involve *Sitting*, *SittingDown*, *WalkDog*, and *Photo* actions, where the subject exhibits severely bent lower limbs, deep forward bends, or extensive self-occlusion. In such poses, 2D detectors frequently produce inaccurate joint locations due to limb overlap and foreshortening. MTF-Transformer, which treats these noisy 2D inputs as fixed conditions, propagates the detection errors directly into the 3D prediction, resulting in distorted leg positions and incorrect torso orientations. In contrast, CoDePose jointly denoises the 2D and 3D components through the coupled diffusion process, progressively refining the 2D inputs toward poses that are geometrically consistent with the predicted 3D structure (*projection consistency*). This allows CoDePose to recover accurate body positions even under heavy self-occlusion and unusual body orientations.

**Fine-grained upper-body articulation.** In actions such as *Phoning*, *Smoking*, and *Photo*, the subject raises one or both arms close to the head or torso, creating subtle joint positions that are difficult for per-view 2D detectors to localize consistently across cameras. MTF-Transformer exhibits notable discrepancies in arm and wrist positions because independently detected 2D poses across views

may not correspond to a single coherent 3D pose. CoDePose alleviates this issue through the body-coordinate consistency loss (*cross-view consistency*), which explicitly encourages all per-view 3D predictions to agree in a canonical, view-invariant body coordinate system. Combined with the uncertainty head that downweights unreliable joints, CoDePose produces joint positions that are both accurate and consistent across views.

**Limitations and future work.** CoDePose relies on off-the-shelf 2D pose detectors and therefore remains sensitive to severely corrupted 2D inputs (Supp. Material), although coupled denoising still recovers better than standard lifting baselines in such difficult cases. While most detection errors are approximately Gaussian in practice, a small but non-negligible heavy-tailed tail persists under occlusion and left–right ambiguity; our Student-T experiment (Tab. 8) shows that partial adaptation is possible when the train and test noise distributions match. An end-to-end formulation that jointly learns video representations together with the coupled diffusion process could, in principle, allow gradient flow from 3D supervision back to the 2D branch. Learning a separate, data-driven 2D noise schedule may further help capture joint- and view-specific detector errors that are not well modeled by a fixed Gaussian assumption. Extending the framework to dynamic camera configurations and in-the-wild settings remains an important direction for future work.

## 5 Conclusion

In this work, we presented CoDePose, which is a coupled 2D–3D diffusion framework for multi-view 3D human pose estimation. Grounded in the empirical finding that most 2D detection errors follow a Gaussian distribution, CoDePose jointly denoises 3D coordinates and per-view 2D projections within a unified reverse process. At inference time, an asymmetric noise strategy initializes the 2D component from detector outputs, enabling accurate reconstruction in a single DDIM step with inference cost that is comparable to MTF-Transformer (Tab. 9). The primary gain comes from coupled 2D–3D denoising; in addition, the uncertainty head and the body-coordinate loss stabilize this design by suppressing erroneous 2D propagation and by enforcing cross-view agreement. On Human3.6M, CoDePose achieves 12.1 mm MPJPE, surpassing all existing camera-free methods while operating entirely without camera parameters.

## Acknowledgements

This work was supported by the National Nature Science Foundation of China (Grant No. 62576104), Shanghai Municipal Science and Technology Commission (Grant No. 24Y32800203), Fudan Kunpeng&Ascend Center of Cultivation. The computations in this research were performed on the CFFF platform of Fudan University.

## References


1. Cai, Q., Hu, X., Hou, S., Yao, L., Huang, Y.: Disentangled diffusion-based 3d human pose estimation with hierarchical spatial and temporal denoiser. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 882–890 (2024)
2. Cai, Y., Zhang, W., Wu, Y., Jin, C.: Fusionformer: A concise unified feature fusion transformer for 3d pose estimation. Proceedings of the AAAI Conference on Artificial Intelligence **38**(2), 900–908 (Mar 2024)
3. Cai, Y., Zhang, W., Wu, Y., Jin, C.: PoseIRM: Enhance 3d human pose estimation on unseen camera settings via invariant risk minimization. In: CVPR (2024)
4. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: CVPR (2018)
5. Chharia, A., Gou, W., Dong, H.: Mv-ssm: Multi-view state space modeling for 3d human pose estimation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11590–11599 (2025)
6. Davoodnia, V., Ghorbani, S., Carbonneau, M.A., Messier, A., Etemad, A.: Upose3d: Uncertainty-aware 3d human pose estimation with cross-view and temporal cues (2024). `https://doi.org/https://doi.org/10.48550/arXiv.2404.14634`, `https://arxiv.org/abs/2404.14634`
7. Gong, J., Foo, L.G., Fan, Z., Ke, Q., Rahmani, H., Liu, J.: DiffPose: Toward more reliable 3d pose estimation. In: CVPR (2023)
8. Gordon, B., Raab, S., Azov, G., Giryes, R., Cohen-Or, D.: FLEX: Extrinsic parameters-free multi-view 3d human motion reconstruction. In: ECCV (2022)
9. He, Y., Yan, R., Fragkiadaki, K., Yu, S.I.: Epipolar transformers. In: CVPR (2020)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
11. Huang, F., Zeng, A., Liu, M., Lai, Q., Xu, Q.: Deepfuse: An imu-aware network for real-time 3d human pose estimation from multi-view image. In: WACV (2020)
12. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: 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)
13. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: ICCV (2019)
14. Jiang, B., Hu, L., Xia, S.: Probabilistic triangulation for uncalibrated multi-view 3d human pose estimation. In: ICCV (2023)
15. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025)
16. Liao, Z., Zhu, J., Wang, C., Hu, H., Waslander, S.L.: Multiple view geometry transformers for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 708–717 (2024)
17. Ma, H., Chen, L., Kong, D., Wang, Z., Liu, X., Tang, H., Yan, X., Xie, Y., Lin, S.Y., Xie, X.: TransFusion: Cross-view fusion with transformer for 3d human pose estimation. In: BMVC (2021)
18. Ma, H., Wang, Z., Chen, Y., Kong, D., Chen, L., Liu, X., Yan, X., Tang, H., Xie, X.: Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V. pp. 424–442. Springer (2022)

19. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In: 3D Vision (3DV), 2017 Fifth International Conference on. IEEE (2017). https://doi.org/10.1109/3dv.2017.00064, http://gvv.mpi-inf.mpg.de/3dhp_dataset
20. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International conference on machine learning. pp. 8162–8171. PMLR (2021)
21. Qiu, H., Wang, C., Wang, J., Wang, N., Zeng, W.: Cross view fusion for 3d human pose estimation. In: ICCV (2019)
22. Remelli, E., Han, S., Honari, S., Fua, P., Wang, R.: Lightweight multi-view 3d pose estimation through camera-disentangled representation. In: CVPR (2020)
23. Shan, W., Liu, Z., Zhang, X., Wang, Z., Han, K., Wang, S., Ma, S., Gao, W.: Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14761–14771 (October 2023)
24. Shuai, H., Wu, L., Liu, Q.: Adaptive multi-view and temporal fusing transformer for 3d human pose estimation. IEEE TPAMI (2022)
25. Sigal, L., Balan, A.O., Black, M.J.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International journal of computer vision **87**(1), 4–27 (2010)
26. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
27. Tu, H., Wang, C., Zeng, W.: Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In: European conference on computer vision. pp. 197–212. Springer (2020)
28. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: ViTPose: Simple vision transformer baselines for human pose estimation. In: NeurIPS (2022)
29. Zhang, J., Cai, Y., Yan, S., Feng, J., et al.: Direct multi-view multi-person 3d pose estimation. NeurIPS **34**, 13153–13164 (2021)
30. Zhang, L., Zhou, K., Lu, F., Zhou, X.D., Shi, Y.: Deep semantic graph transformer for multi-view 3d human pose estimation. Proceedings of the AAAI Conference on Artificial Intelligence **38**(7), 7205–7214 (Mar 2024)
31. Zhang, W., Liu, M., Liu, H., Li, W.: Svtformer: Spatial-view-temporal transformer for multi-view 3d human pose estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 10148–10156 (2025)
32. Zhang, Z., Wang, C., Qiu, W., Qin, W., Zeng, W.: AdaFuse: Adaptive multiview fusion for accurate human pose estimation in the wild. IJCV **129**, 703–718 (2021)
33. Zhao, Q., Zheng, C., Liu, M., Wang, P., Chen, C.: Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8877–8886 (June 2023)
34. Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., Ding, Z.: 3d human pose estimation with spatial and temporal transformers. In: ICCV (2021)
35. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: MotionBERT: A unified perspective on learning human motion representations. In: ICCV (2023)