# CTRLRECON: STEERING EXPRESSIVE HUMAN RECONSTRUCTION TOWARD PRESCRIBED STATES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Expressive human pose and shape estimation (EHPS) increasingly underpins digital-human systems by mapping visual inputs to the poses, hand articulations, body shape and facial expressions enacted by an avatar. If an adversary can control these outputs, an innocuous input could induce attacker-specified behavior absent from the source, elevating the threat from reconstruction degradation to deliberate behavioral manipulation. Yet existing attacks against EHPS remain predominantly untargeted, showing that reconstructions can be corrupted or driven toward implausible configurations, but not whether an adversary can steer them toward a prescribed state. To address this gap, we introduce CTRLRECON, a framework for targeted attacks on EHPS. Specifically, CTRLRECON constructs an image-specific target by integrating attacker-specified articulations with the corresponding clean predictions, holding it constant throughout the optimization process. Guided by this static target, a representation-aware objective jointly enforces the prescribed modifications and preserves the unmodified components. Subsequently, a two-stage procedure optimizes over the bounded continuous space and refines the result into the discrete 8-bit domain, ensuring adversarial effectiveness is fully maintained after standard image export. Experiments across three EHPS estimators validate the effectiveness of CTRLRECON for targeted control. In vision-language models (VLMs)-based evaluation, CTRLRECON receives 60.10% of the votes, exceeding the strongest baseline by 36.68 percentage points.

## 1 INTRODUCTION

Expressive human pose and shape estimation (EHPS) (Li et al., 2022; Lin et al., 2023a; Sárándi & Pons-Moll, 2024; Xiang et al., 2025) is a core perception component of digital-human systems, recovering body pose, hand articulation, facial expression, and shape from monocular images for motion capture (Xiang et al., 2019; Yang et al., 2026), live streaming (Lu et al., 2021; Wang et al., 2025), virtual reality (Shen et al., 2023; Combe et al., 2024), and human–computer interaction (Pavlakos et al., 2019; Moon et al., 2022; Lin et al., 2023b; Cai et al., 2023). In these applications, EHPS predictions are not merely intermediate geometric outputs: they directly drive the avatar's body movements, hand gestures, and facial expressions (Chen et al., 2025a;b). This direct mapping from EHPS predictions to avatar behavior (Zhao & Tulsiani, 2024) makes EHPS a natural attack surface for adversaries seeking to manipulate how the avatar acts. By crafting visually inconspicuous perturbations, an attacker could induce behaviors absent from the source image, such as sexually suggestive, offensive, or extremist-associated actions. In live or interactive scenarios, such manipulation could lead to content-safety incidents and undermine trust in digital-human systems. However, existing adversarial studies on EHPS remain predominantly untargeted (Li et al., 2026; 2025), demonstrating that reconstruction outputs can be disrupted without establishing whether they can be deliberately controlled. This distinction is illustrated in Figure 1. This gap motivates a more fundamental question: can an attacker use visually inconspicuous perturbations to steer selected pose or expression components toward a prescribed state while preserving the remaining reconstruction?

Existing adversarial attacks relevant to EHPS can be broadly grouped into two lines. General-purpose methods, such as FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017), and C&W (Carlini & Wagner, 2017), provide established gradient- or optimization-based mechanisms for constructing adversarial perturbations. EHPS-specific attacks further tailor adversarial objectives to avatar generation: TBA (Li et al., 2026) amplifies joint- and vertex-level reconstruction



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

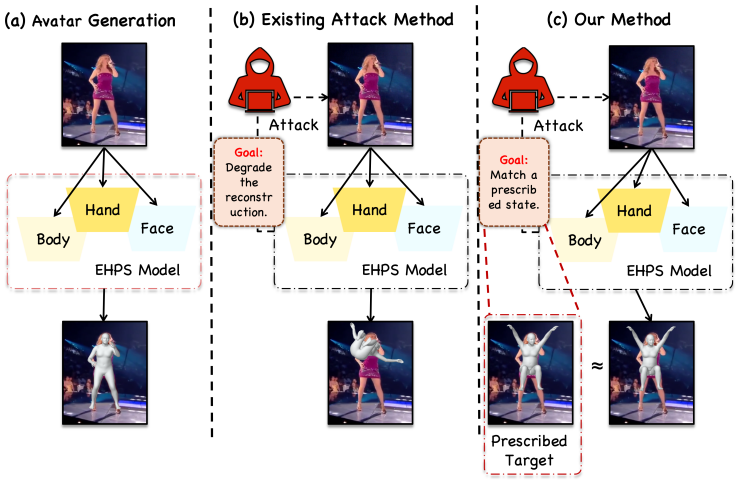



Figure 1: Overview of adversarial attacks in avatar generation. (a) Standard EHPS inference reconstructs avatar states from the input. (b) Existing attacks induce untargeted reconstruction deviations. (c) Our method enables targeted attacks that steer the avatar toward attacker-prescribed poses, gestures, or expressions.

errors, while LatentStealth (Li et al., 2025) maximizes reconstruction discrepancy under perceptual and perturbation constraints. Together, these methods establish the vulnerability of EHPS, but primarily pursue untargeted deviation from the clean reconstruction. Targeted manipulation instead requires steering selected pose or expression components toward prescribed states while preserving the remaining outputs. This becomes nontrivial in EHPS because its components are represented in heterogeneous parameter spaces, for which a single generic target distance is insufficient.

To address this challenge, we introduce CTRLRECON, a framework for targeted attacks on EHPS. Given a prescribed body pose, hand gesture, or facial expression, CTRLRECON constructs a fixed image-specific target by replacing selected components with prescribed values while retaining clean predictions elsewhere, thereby specifying both the desired change and what should remain unchanged. Optimizing toward this target is challenging because EHPS outputs span heterogeneous representations, including articulated rotations, 6D hand representations, expression parameters, and camera translation. We therefore develop a representation-aware objective that measures target agreement in the appropriate space for each component while jointly accounting for target attainment and preservation. Since continuous optimization does not directly produce a standard 8-bit image, quantization may introduce additional mismatch from the target. We address this with a two-stage procedure that performs bounded continuous search followed by refinement on the feasible 8-bit image space. Experiments across multiple benchmark datasets and EHPS models demonstrate effective targeted control over body pose, hand articulation, and facial expression. In vision-language models (VLMs)-based evaluation on UBody, CTRLRECON achieves an average vote share of 60.10% across all voting groups, exceeding the strongest baseline by 36.68 percentage points. Our findings demonstrate that targeted control of EHPS is feasible, extending the demonstrated threat from reconstruction degradation to deliberate manipulation of reconstructed behavior.

Our contributions are threefold: **1)** We introduce an image-specific target construction that assigns prescribed values to selected articulations while retaining clean predictions for the remaining components, yielding a fixed target for EHPS control. **2)** We propose CTRLRECON, a framework for targeted attacks on EHPS that steers reconstructions toward prescribed states. **3)** Experiments across multiple EHPS models and datasets demonstrate effective targeted control of body poses, hand gestures, and facial expressions.

## 2 RELATED WORK

**Expressive human pose and shape estimation.** Expressive human pose and shape estimation (EHPS) recovers body pose, hand articulation, facial expression, and shape from monocular im-

ages, with SMPL-X providing a unified parametric representation of the human body, hands, and face (Pavlakos et al., 2019). Early approaches improve local reconstruction quality through part-specific modeling and fusion (Choutas et al., 2020; Moon et al., 2022; Zhang et al., 2023), while more recent methods adopt unified whole-body architectures to jointly model different human components (Lin et al., 2023b; Sun et al., 2024). Further advances exploit larger models and training data to improve reconstruction accuracy and generalization (Cai et al., 2023; Yin et al., 2025). These developments have substantially improved expressive human reconstruction, making EHPS an increasingly important component of digital-human systems.

**Adversarial attacks on EHPS.** Classical adversarial methods such as FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017), and C&W (Carlini & Wagner, 2017) provide general mechanisms for optimizing visually small input perturbations. Recent work has extended adversarial attacks to avatar generation, with objectives tailored to the structured outputs of EHPS. TBA increases joint- and vertex-level reconstruction errors through adversarial perturbation and refinement (Li et al., 2026), while LatentStealth focuses on perceptually inconspicuous and efficient attacks while maximizing reconstruction discrepancy (Li et al., 2025). Despite their different designs, existing EHPS attacks primarily pursue untargeted degradation: they encourage the reconstruction to deviate from its clean prediction without specifying a desired output state. As a result, they do not address whether an attacker can steer selected pose or expression components toward prescribed states while preserving the remaining reconstruction. Our work instead considers a stronger attack setting in which the adversary seeks to make the reconstructed avatar exhibit attacker-specified poses, gestures, or expressions, rather than merely inducing arbitrary reconstruction errors.

# 3 Preliminary

## 3.1 Expressive Human Reconstruction with SMPL-X

**Structured human representation.** Expressive human pose and shape estimation (EHPS) reconstructs a parameterized digital human from a monocular RGB image. Given a preprocessed image $\mathbf{x} \in [0,1]^{H\times W\times 3}$, an estimator $f$ predicts

$$\mathbf{y} = f(\mathbf{x}) = (\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\psi}, \mathbf{c}), \tag{1}$$

where $\boldsymbol{\theta}$, $\boldsymbol{\beta}$, $\boldsymbol{\psi}$, and $\mathbf{c}$ denote pose, body shape, facial expression, and camera parameters, respectively. The pose vector is partitioned into

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_{\text{root}}, \boldsymbol{\theta}_{\text{body}}, \boldsymbol{\theta}_{\text{lhand}}, \boldsymbol{\theta}_{\text{rhand}}, \boldsymbol{\theta}_{\text{jaw}}]. \tag{2}$$

Each pose block parameterizes joint rotations, with the root defining the global orientation and the remaining joints defined locally relative to their kinematic parents. For joint $j$, the pose parameter $\boldsymbol{\theta}_j$ represents a rotation whose matrix form is denoted by $\mathbf{R}_j = \text{Rot}(\boldsymbol{\theta}_j) \in \text{SO}(3)$. Thus, $\boldsymbol{\theta}_j$ and $\mathbf{R}_j$ describe the same joint rotation in parameter and matrix form, respectively. The estimator may internally predict this rotation using a different representation before producing the pose parameter consumed by SMPL-X. In particular, when a continuous six-dimensional representation is used, the raw output can be written as $\mathbf{A}_j = [\mathbf{a}_{j,1}, \mathbf{a}_{j,2}] \in \mathbb{R}^{3\times 2}$ and orthogonalized by Gram–Schmidt to obtain $\mathbf{R}_j = G(\mathbf{A}_j)$. The subsequent conversion to the native pose parameterization follows estimator-specific coordinate and decoder-mean conventions (Appendix C.1).

**Geometric reconstruction.** The fixed SMPL-X decoder (Pavlakos et al., 2019) maps the predicted human parameters to three-dimensional geometry:

$$(\mathbf{V}, \mathbf{J}) = \mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\psi}), \tag{3}$$

where $\mathbf{V} \in \mathbb{R}^{N_V\times 3}$ and $\mathbf{J} \in \mathbb{R}^{N_J\times 3}$ denote the mesh vertices and joint coordinates, respectively. SMPL-X propagates the joint rotations through the kinematic hierarchy to articulate the body, while $\boldsymbol{\beta}$ and $\boldsymbol{\psi}$ control body shape and facial deformation. The camera parameters determine the placement or projection of the reconstructed geometry. Because non-root rotations are defined locally, preserving a joint's local rotation does not in general preserve its absolute three-dimensional position when an ancestor rotation changes.

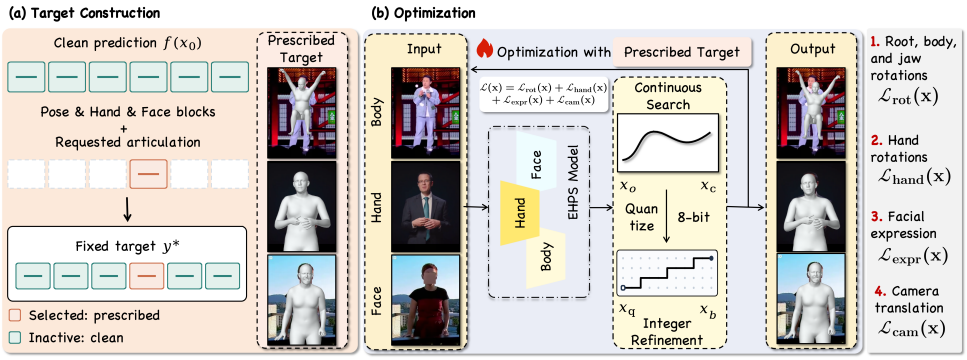



Figure 2: Overview of CTRLRECON. (a) Image-specific target construction specifies prescribed changes while preserving non-target components. (b) Representation-aware, continuous-to-8-bit optimization steers EHPS outputs toward the prescribed target.

### 3.2 ADVERSARIAL ATTACK PIPELINE

An image-space adversarial attack modifies a clean input while keeping the estimator $f$ fixed. Representing images as vectors in $\mathbb{R}^d$, with $d = 3 \times H \times W$, we define the perturbed input and feasible set under a budget $\epsilon > 0$ as

$$\mathbf{x} = \Pi_{[0,1]}(\mathbf{x}_0 + \boldsymbol{\delta}), \qquad \|\boldsymbol{\delta}\|_\infty \le \epsilon, \tag{4}$$

$$\mathcal{C} = \{\mathbf{x} \in [0,1]^d : \|\mathbf{x} - \mathbf{x}_0\|_\infty \le \epsilon\}, \tag{5}$$

where $\mathbf{x}_0$ is the clean image, $\mathbf{x}$ is a candidate input, and $\Pi_{[0,1]}$ clips pixel values to their valid range. The attack objective guides the search over $\mathcal{C}$. Untargeted attacks seek deviation from a reference prediction or reconstruction, whereas targeted attacks seek agreement with a prescribed output. We study structured targeted attacks on EHPS that modify selected articulations while preserving the remaining pose and expression components.

## 4 METHOD

CTRLRECON first constructs a fixed image-specific target by replacing selected pose or expression components with prescribed values while retaining clean predictions for the remaining components. It then optimizes a representation-aware objective through bounded continuous search followed by refinement on the feasible 8-bit lattice. Figure 2 illustrates the overall framework.

### 4.1 PROBLEM FORMULATION

Given a fixed EHPS estimator $f$ under white-box access and a clean image $\mathbf{x}_0$, we seek an adversarial image $\mathbf{x} \in \mathcal{C}$, where $\mathcal{C}$ is the bounded perturbation set defined in Section 3.2. Let $\mathcal{P}$ index the individual joint-rotation blocks and the expression block. An active set $\mathcal{A} \subseteq \mathcal{P}$ specifies which blocks should match the requested values $\bar{\mathbf{y}}_k$, while the remaining blocks should preserve their clean predictions. Rotation preservation follows the estimator's coordinate convention: global orientation for the root and local rotations for non-root joints.

### 4.2 TARGET CONSTRUCTION

Writing $f_k$ for output block $k$, we define the image-specific target as

$$\mathbf{y}_k^\star = \begin{cases} \bar{\mathbf{y}}_k, & k \in \mathcal{A}, \\ f_k(\mathbf{x}_0), & k \in \mathcal{P} \setminus \mathcal{A}. \end{cases} \tag{6}$$

We also set $\mathbf{c}^\star = \mathbf{c}(\mathbf{x}_0)$ to anchor camera translation to its clean value. All targets are computed once and remain fixed throughout optimization.

### 4.3 CTRLRECON OBJECTIVE

To account for the geometry and scale of each output representation, we minimize

$$\mathcal{L}(\mathbf{x}) = \mathcal{L}_{\text{rot}}(\mathbf{x}) + \mathcal{L}_{\text{hand}}(\mathbf{x}) + \mathcal{L}_{\text{expr}}(\mathbf{x}) + \mathcal{L}_{\text{cam}}(\mathbf{x}). \tag{7}$$

All pose and expression blocks contribute through their prescribed or clean targets. Body shape is excluded from the objective. Below, we omit $\mathbf{x}$ from predicted quantities. Targets and normalization scales are fixed, with the scales determining relative penalties without additional group weights.

**Root, body, and jaw rotations.** Let $\mathcal{B}$ index root, body, and jaw joints. We measure orientation discrepancy using the prediction and target rotation matrices $\mathbf{R}_j, \mathbf{R}_j^\star \in \mathrm{SO}(3)$ in the estimator's native pose coordinates:

$$\mathcal{L}_{\text{rot}}(\mathbf{x}) = \frac{1}{2\kappa_R^2} \sum_{j \in \mathcal{B}} \|\mathbf{R}_j - \mathbf{R}_j^\star\|_F^2. \tag{8}$$

For a relative rotation angle $\vartheta$, the Frobenius discrepancy is $\sqrt{8}\sin(\vartheta/2)$. Thus $\kappa_R = \sqrt{8}\sin(\tau_R/2)$, with $0 < \tau_R \leq \pi$, calibrates the loss to a reference angle $\tau_R$ (Appendix A.1).

**Hand rotations.** The Gram–Schmidt decoder $G$ maps multiple six-dimensional outputs to the same rotation and can become sensitive to small perturbations near rank-deficient representations (Appendix A.2). For finger joints, we use a loss that admits multiple target representations while penalizing insufficient positive pivots relative to the clean output.

Let $\mathcal{H}$ index finger joints, with raw outputs $\mathbf{A}_j \in \mathbb{R}^{3\times 2}$. We express each target as $\mathbf{R}_{\text{reg},j}^\star$ in the coordinates of $G(\mathbf{A}_j)$, before conversion to native pose parameters. For each full-column-rank clean output $\mathbf{A}_j(\mathbf{x}_0)$, let $\mathbf{T}_{0,j}$ be the fixed positive-diagonal upper-triangular factor in its thin QR decomposition. The transformation $\mathbf{Z}_j = (\mathbf{R}_{\text{reg},j}^\star)^\top \mathbf{A}_j \mathbf{T}_{0,j}^{-1}$ expresses the prediction in the target frame relative to the clean scale and shear. For $0 < \mu < 1$, define

$$\mathbf{T}_\mu(\mathbf{Z}) = \begin{bmatrix} \max(Z_{11}, \mu) & Z_{12} \\ 0 & \max(Z_{22}, \mu) \end{bmatrix}. \tag{9}$$

With $\mathbf{I}_{3,2}$ denoting the first two columns of $\mathbf{I}_3$ and $\tau_h > 0$, the hand loss is

$$\mathcal{L}_{\text{hand}}(\mathbf{x}) = \frac{1}{2\tau_h^2} \sum_{j \in \mathcal{H}} \left\| \mathbf{Z}_j \mathbf{T}_\mu(\mathbf{Z}_j)^{-1} - \mathbf{I}_{3,2} \right\|_F^2. \tag{10}$$

At zero loss, $G(\mathbf{A}_j) = \mathbf{R}_{\text{reg},j}^\star$ for all $j \in \mathcal{H}$, with multiple raw representations admissible.

**Facial expression.** Jaw rotation and expression coefficients control different aspects of facial configuration. We match the expression coefficients to their prescribed or clean targets through

$$\mathcal{L}_{\text{expr}}(\mathbf{x}) = \frac{\|\boldsymbol{\psi} - \boldsymbol{\psi}^\star\|_2^2}{2 d_\psi \tau_\psi^2}, \tag{11}$$

where $d_\psi$ is the coefficient dimension and $\tau_\psi > 0$ is the reference scale. Division by $d_\psi$ normalizes the squared discrepancy per coefficient.

**Camera translation.** To preserve reconstruction placement alongside the requested articulation, we penalize displacement from the clean camera target:

$$\mathcal{L}_{\text{cam}}(\mathbf{x}) = \frac{\|\mathbf{c} - \mathbf{c}^\star\|_2^2}{2\tau_c^2}, \tag{12}$$

where $\tau_c > 0$ sets the scale of this soft preservation penalty.

### 4.4 OPTIMIZATION

We minimize $\mathcal{L}$ in two stages: continuous search over $\mathcal{C}$ followed by refinement on the feasible eight-bit lattice. The second stage addresses changes in target agreement introduced by quantization.

**Continuous optimization.** Starting from $\mathbf{x}_0$, the search uses a limited history of image and gradient differences to approximate local curvature. It identifies active pixel bounds along a projected-gradient path, refines the free coordinates, and performs a feasible line search (Byrd et al., 1995). Gradients include all image-dependent paths, including those through predicted crop coordinates. Throughout the search, we retain the lowest-loss evaluated feasible image $\mathbf{x}_c$, including line-search trials.

**Quantization and integer refinement.** We quantize $\mathbf{x}_c$ to its nearest feasible eight-bit image $\mathbf{x}_q \in \mathcal{C}_8$, where

$$\mathcal{C}_8 = \mathcal{C} \cap \{0, 1/255, \ldots, 1\}^d. \tag{13}$$

At an integer iterate $\mathbf{x}$, let $\mathcal{S}_K(\mathbf{x})$ contain feasible one-level changes to at most $K$ pixel channels. A nonnegative Gauss–Newton diagonal $\mathbf{h}$, computed once from the componentwise residuals at $\mathbf{x}_q$ (Nocedal & Wright, 2006), defines the proposal

$$\mathbf{s}^\star \in \operatorname*{arg\,min}_{\mathbf{s} \in \mathcal{S}_K(\mathbf{x})} \left[ \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x})^\top \mathbf{s} + \frac{1}{2} \sum_{i=1}^{d} h_i s_i^2 \right]. \tag{14}$$

By separability, the proposal selects up to $K$ feasible changes with the largest predicted reductions, while the curvature captures local output sensitivity, including that of currently matched preservation terms. Each proposal is then evaluated using $\mathcal{L}$ and accepted only under strict descent with sufficient agreement between predicted and observed reduction, which also controls the adjustment of $K$. Independently of acceptance, we retain the best evaluated lattice image $\mathbf{x}_b$, export it as a PNG, and reevaluate the reloaded image with the original estimator. Appendix B provides the full derivation and implementation details.

## 5 Experiments

We investigate the effectiveness of CtrlRecon in controlling body pose, hand articulation, and facial expression while preserving non-target components. Evaluation covers geometric target agreement, joint attack success, visual matching, and ablations on the objective design and perturbation budget.

### 5.1 Experimental Settings

**Datasets and Models.** We evaluate SMPLer-X-H32 (H32) (Cai et al., 2023) and OSX-L (Lin et al., 2023b) on UBody (Lin et al., 2023b) and PW3D (Von Marcard et al., 2018) using their official checkpoints, with all model parameters fixed. UBody provides 2,351 held-out inputs after reserving three of its 2,354 test images for development, while PW3D uses a fixed subset of 2,010 test images.

**Baselines.** We compare with targeted variants of FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017), and C&W (Carlini & Wagner, 2017). All optimize the same unweighted raw-parameter $\ell_2$ target loss, while C&W additionally penalizes input distortion. They use complete input gradients, identical evaluation inputs, and the same target construction. Appendix C.3 details these baselines.

**Implementation Details.** We consider three targets: a raised-arm sitting pose (Body), a right-hand peace gesture (Hand), and a grimacing expression (Face), while preserving all inactive components (Appendix C.1). Unless stated otherwise, attacks use $\|\boldsymbol{\delta}\|_\infty \leq 0.06$ on pixels in $[0, 1]$ and are reevaluated after PNG export and reload. For each model–dataset–target setting, PGD and C&W use the ceiling of CtrlRecon's mean number of accepted continuous updates, including unsuccessful attacks. Appendices B.4 and C.2 detail numerical settings and budget calibration.

### 5.2 Targeted Attack Performance

**Settings.** We evaluate targeted attacks on H32 and OSX-L using the full UBody test set and a selected subset of PW3D. The attacks target a prescribed body pose, right-hand gesture, or facial expression under the configuration described in Section 5.1.

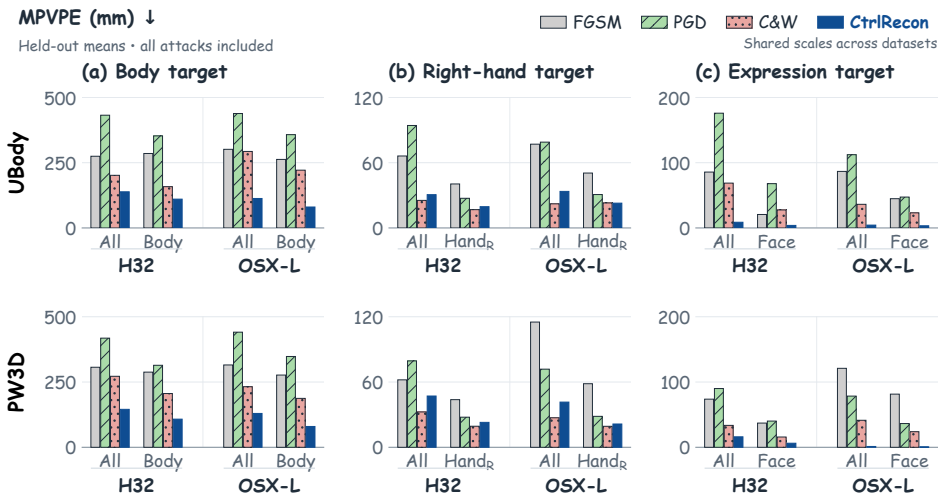



Figure 3: Quantitative comparison of target matching on UBody and PW3D. MPVPE (↓) measures agreement with the prescribed body, right-hand, and expression targets.

Table 1: Attack success rate (%) comparison with baselines on UBody and PW3D using H32 and OSX-L.

| Model | Method | UBody | | | PW3D | | |
|---|---|---|---|---|---|---|---|
| | | Body | Hand | Face | Body | Hand | Face |
| H32 | FGSM | 0.17 | 0.00 | 0.00 | 0.95 | 0.00 | 0.00 |
| | PGD | 11.95 | 65.04 | 45.98 | 10.80 | 48.98 | 16.43 |
| | C&W | 85.96 | 89.45 | 75.54 | 69.89 | 92.14 | 73.42 |
| | CTRLRECON | **91.96** | **91.45** | **95.32** | **95.37** | **94.03** | **95.07** |
| OSX-L | FGSM | 0.34 | 0.00 | 0.04 | 0.50 | 0.00 | 0.00 |
| | PGD | 27.26 | 33.35 | 73.16 | 14.72 | 32.49 | 69.16 |
| | C&W | 78.43 | 15.48 | 89.75 | 69.96 | 52.05 | 91.02 |
| | CTRLRECON | **97.24** | **95.53** | **99.49** | **97.06** | **95.16** | **100.00** |

**Evaluation Metrics.** We report target-relative mean per-vertex position error (MPVPE) and attack success rate (ASR) on both datasets. MPVPE measures the average geometric deviation between the reconstructed mesh and its target, in millimeters, for the whole mesh (All) and the corresponding target region: Body, right Hand, or Face. ASR measures the percentage of inputs that simultaneously satisfy thresholds on rotation, expression, and camera translation deviations. Appendices C.4–C.5 define the metrics and thresholds; Appendices D.1–D.2 give extended results.

**Results.** Figure 3 shows that CTRLRECON achieves the lowest whole-body MPVPE for body targets and the lowest face MPVPE for expression targets on both H32 and OSX-L across UBody and PW3D. For example, on UBody, CTRLRECON reduces whole-body MPVPE for body targets from C&W's 201.68 to 138.55 mm on H32 and from 293.07 to 112.97 mm on OSX-L. For hand targets, however, C&W achieves lower whole-body MPVPE on both models and datasets. Despite this variation in geometric performance, Table 1 shows that CTRLRECON achieves the highest joint ASR in all 12 model–dataset–target combinations, ranging from 91.45% to 99.49% on UBody and from 94.03% to 100.00% on PW3D. These results demonstrate consistent advantages in jointly satisfying the rotation, expression, and camera requirements, although the ranking by mean geometric error depends on the target.

Table 2: VLM vote share (%, ↑) on UBody for H32 and OSX-L. Higher values indicate more frequent selection as the closest visual match to the prescribed target.

| Model | Method | Body target | | | | Hand target (right hand) | | | | Face target | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GPT | Q3.5 | Q3.8 | DS | GPT | Q3.5 | Q3.8 | DS | GPT | Q3.5 | Q3.8 | DS |
| H32 | FGSM | 0.00 | 0.09 | 0.43 | 0.14 | 0.00 | 0.04 | 0.00 | 0.00 | 0.04 | 0.57 | 0.17 | 0.06 |
| | PGD | 28.78 | 27.60 | 27.39 | 26.03 | **35.74** | **35.79** | **34.85** | 38.26 | 5.51 | 5.84 | 5.37 | 5.22 |
| | C&W | 18.64 | 21.17 | 20.10 | 14.71 | 31.91 | 33.05 | 32.55 | 18.26 | 10.38 | 9.97 | 9.84 | 5.45 |
| | CTRLRECON | **52.58** | **51.14** | **52.09** | **59.12** | 32.34 | 31.12 | 32.60 | **43.48** | **84.06** | **83.62** | **84.62** | **89.28** |
| OSX-L | FGSM | 0.04 | 0.21 | 0.09 | 0.04 | 0.00 | 0.04 | 0.00 | 0.04 | 0.04 | 0.39 | 0.21 | 0.56 |
| | PGD | 14.98 | 17.43 | 17.35 | 16.60 | 23.82 | 28.55 | 24.63 | 28.86 | 28.43 | 26.36 | 27.17 | 31.57 |
| | C&W | 0.98 | 1.76 | 1.87 | 1.28 | 13.53 | 24.76 | 17.18 | 23.84 | 19.95 | 13.97 | 20.44 | 26.71 |
| | CTRLRECON | **84.00** | **80.60** | **80.69** | **82.08** | **62.65** | **46.66** | **58.19** | **47.25** | **51.58** | **59.28** | **52.17** | **41.16** |

### 5.3 VISUAL EVALUATION

**Settings.** We compare CTRLRECON with the three baselines on H32 and OSX-L over the full UBody test set. Visual similarity is independently assessed by four vision-language models (VLMs): GPT-5.6 Sol (GPT), Qwen3.5-122B-A10B (Q3.5), Qwen3.8-27B-FP8 (Q3.8), and DeepSeek-V4.1-Flash (DS). For each comparison, a judge observes three views of all candidate reconstructions together with a shared target reference and selects the candidate that best matches the target over the specified active blocks, while disregarding the remaining components. Candidate identities are anonymized, and their display positions are randomly permuted for each comparison. See Appendices C.6 and D.3.

**Evaluation Metrics.** We use vote share as the evaluation metric, defined as the percentage of votes received by a candidate over all evaluated samples in the UBody test set.

**Results.** Table 2 shows that CTRLRECON achieves the highest vote share in 21 of 24 evaluation settings, with an overall mean of $60.10\%$, exceeding PGD, the strongest baseline, by $36.68$ percentage points. All four judges assign the highest vote share to CTRLRECON for the body and expression targets on both models, as well as for the hand target on OSX-L. The H32 hand target is the only consistent exception, where PGD ranks first under GPT and both Qwen judges. However, as reported in Table 1, CTRLRECON attains a substantially higher ASR in this setting, reflecting stronger joint satisfaction of the target and preservation constraints. Figure 4 provides qualitative comparisons for all three target types.

### 5.4 ABLATION STUDIES

**Settings.** We examine the contributions of the objective terms and perturbation budget on H32/UBody, using the full CTRLRECON objective at $\epsilon = 0.06$ as the reference configuration. Loss ablations remove individual objective terms, while budget ablations vary $\epsilon$ with the full objective retained. Articulation terms are ablated on their corresponding targets, whereas the camera term and perturbation budget are evaluated across all three targets. See Appendix C.7.

**Evaluation Metrics.** We report regional Procrustes-aligned mean per-vertex position error (PA-MPVPE) for geometric target agreement, together with camera translation error and joint ASR as defined in Section 5.2.

**Results.** Table 3 shows that removing an articulation term substantially degrades success on its associated target, confirming the target-specific role of each articulation objective. Removing the camera term markedly increases camera translation error while leaving aligned geometric error relatively small, indicating that PA-MPVPE does not by itself capture camera preservation because Procrustes alignment removes global translation. Increasing the perturbation budget improves success mainly in the low-budget regime, with only marginal gains beyond $\epsilon = 0.06$. These results




Figure 4: Qualitative comparison on UBody using H32. Columns show Clean, FGSM, PGD, C&W, CTRLRECON, and the prescribed target; rows correspond to body, right-hand, and expression targets.

Table 3: Loss-term and perturbation-budget ablations on UBody using H32. Evaluation metrics include regional PA-MPVPE (PA-V), camera translation error (Cam.), and attack success rate (ASR).

| | Body target | | | Right-hand target | | | Face target | | |
|---|---|---|---|---|---|---|---|---|---|
| Configuration | PA-V↓ | Cam.↓ | ASR↑ | PA-V↓ | Cam.↓ | ASR↑ | PA-V↓ | Cam.↓ | ASR↑ |
| Full ($\epsilon = 0.06$) | 68.66 | 11.54 | 91.96 | 9.51 | 5.13 | 91.45 | 0.56 | 1.53 | 95.32 |
| (a) Loss terms $\epsilon = 0.06$ | | | | | | | | | |
| w/o body | 163.31 | 0.13 | 1.87 | — | — | — | — | — | — |
| w/o hands | — | — | — | 31.44 | 0.05 | 0.09 | — | — | — |
| w/o expression | — | — | — | — | — | — | 2.09 | 0.22 | 0.00 |
| w/o camera | 65.82 | 876.20 | 1.23 | 9.57 | 106.36 | 12.72 | 0.49 | 328.55 | 1.06 |
| (b) Perturbation budget all losses enabled | | | | | | | | | |
| $\epsilon = 0.01$ | 146.27 | 32.63 | 28.46 | 21.82 | 18.33 | 38.24 | 2.88 | 11.04 | 0.51 |
| $\epsilon = 0.03$ | 80.02 | 19.08 | 76.99 | 10.96 | 6.11 | 89.58 | 1.11 | 4.95 | 92.17 |
| $\epsilon = 0.09$ | 66.17 | 11.02 | 92.09 | 9.13 | 5.31 | 91.49 | 0.50 | 0.96 | 95.53 |

indicate complementary contributions from the articulation and camera terms, together with diminishing returns from larger perturbation budgets.

## 6 CONCLUSION AND DISCUSSION

We introduced CTRLRECON to study targeted adversarial attack of expressive human pose and shape estimation. Experiments across multiple EHPS models and datasets show that body pose, hand articulation, and facial expression can be steered toward prescribed states while largely preserving non-target components. This shifts the demonstrated EHPS threat from arbitrary reconstruction degradation to attacker-directed control, with potential consequences for avatar behavior in digital-human systems. Our evaluation is limited to a controlled attack setting and a finite set of prescribed targets; broader targets, transferability, and defenses remain open directions.

## AI USE STATEMENT

Generative AI tools were used in two roles in this work. First, they assisted with language polishing, translation, drafting and revising portions of the manuscript, standardizing mathematical notation and expressions, and correcting LaTeX syntax and formatting. Second, vision-language models were used as experimental judges in the visual target-matching evaluation described in Section 5.3 and Appendix C.6, and their outputs were aggregated according to the predefined evaluation protocol to obtain the reported vote-share results. The research methodology, attack formulation, and experimental protocol were determined by the authors. All AI-assisted manuscript content and model-generated evaluation outputs were reviewed and verified by the authors, who take full responsibility for the final content and reported results.

## ETHICS STATEMENT

This work investigates the adversarial robustness of expressive human pose and shape estimation (EHPS) systems by studying whether their reconstructed outputs can be steered toward prescribed body poses, hand gestures, or facial expressions. Such capabilities are inherently dual-use. While the purpose of this study is to expose a previously underexplored security risk and support the development of more robust digital-human systems, similar techniques could in principle be misused to manipulate reconstructed human behavior or generate outputs that misrepresent the content of an input image.

Our experiments are conducted on fixed research models and established benchmark datasets, and are limited to controlled robustness evaluation. The prescribed targets are predefined pose, gesture, and expression templates used to measure targeted control, rather than content designed to imitate or target specific individuals. We do not deploy the attack against real-world services or interactive systems, nor do we use it to alter the behavior of identifiable persons. The study focuses on characterizing model vulnerabilities under bounded image perturbations and on measuring their impact on reconstructed outputs.

We believe that documenting this attack surface is important for improving the security of EHPS-based applications, particularly in digital-human, avatar, motion-capture, and interactive systems where reconstructed pose and expression can directly determine downstream behavior. We therefore present the method together with its assumptions, perturbation constraints, and evaluation protocol so that its security implications can be studied in a controlled and reproducible setting. All datasets, model checkpoints, and third-party assets used in this work remain subject to their respective licenses and terms of use.

## REPRODUCIBILITY STATEMENT

We provide detailed specifications of the proposed method and experimental protocol to facilitate reproducibility. Sections 3 and 4 define the EHPS output representation, target construction, representation-aware objective, perturbation constraints, and two-stage optimization procedure. Appendix A provides the supporting objective analysis, and Appendix B details the optimization procedure, gradient treatment, integer refinement, numerical settings, and computational-cost accounting.

Appendix C specifies the complete experimental protocol, including target definitions and estimator conventions (Appendix C.1), evaluation subsets and budget calibration (Appendix C.2), baseline configurations (Appendix C.3), geometric and joint-success metrics (Appendices C.4 and C.5), the VLM evaluation protocol (Appendix C.6), and the ablation settings (Appendix C.7). Numerical constants required for reproduction, including loss scales, optimization budgets, solver settings, hardware configuration, and random seed, are reported in Appendix B.4. Extended quantitative results are provided in Appendix D.

All reported attacks are evaluated after feasible 8-bit quantization, PNG export, and reload with the original estimator. Unsuccessful attacks are retained in the reported means and success rates unless explicitly stated otherwise. We intend to make the implementation and experiment configurations available to support reproduction of the reported results, subject to the licenses and redistribution terms of the datasets, pretrained checkpoints, and third-party model assets used in this work.

## REFERENCES


Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Zhongang Cai, Wanqi Yin, Ailing Zeng, Chen Wei, Qingping Sun, Wang Yanjun, Hui En Pang, Haiyi Mei, Mingyuan Zhang, Lei Zhang, Chen Change Loy, Lei Yang, and Ziwei Liu. Smpler-x: Scaling up expressive human pose and shape estimation. *Advances in Neural Information Processing Systems*, 36:11454–11468, 2023.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pp. 39–57, 2017.

Rongyu Chen, Li'an Zhuo, Linlin Yang, Qi Wang, Liefeng Bo, Bang Zhang, and Angela Yao. Extpose: Robust and coherent pose estimation by extending vits. In *International Conference on Machine Learning*, volume 267, 2025a.

Zeyuan Chen, Hongyi Xu, Guoxian Song, You Xie, Chenxu Zhang, Xin Chen, Chao Wang, Di Chang, and Linjie Luo. X-dancer: Expressive music to human dance video generation. In *International Conference on Computer Vision*, pp. 10602–10611, 2025b.

Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *European Conference on Computer Vision*, pp. 20–40, 2020.

Théo Combe, Rebecca Fribourg, Lucas Detto, and Jean-Marie Normand. Exploring the influence of virtual avatar heads in mixed reality on social presence, performance and user experience in collaborative tasks. *IEEE Transactions on Visualization and Computer Graphics*, 30(5):2206–2216, 2024.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Computer Vision and Pattern Recognition*, pp. 13137–13146, 2022.

Zhiying Li, Guanggang Geng, Yeying Jin, Shuyuan Lin, Fengyuan Ma, Zhaoxin Fan, and Lili Wang. Latentstealth: Unnoticeable and efficient adversarial attacks on expressive human pose and shape estimation. *arXiv preprint arXiv:2505.12009*, 2025.

Zhiying Li, Yeying Jin, Fan Shen, Zhi Liu, Weibin Chen, Pengju Zhang, Xiaomei Zhang, Boyu Chen, Michael Shen, Kejian Wu, Zhaoxin Fan, and Jin Dong. Unveiling hidden vulnerabilities in digital human generation via adversarial attacks. *Pattern Recognition*, 170:112042, 2026.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. In *Advances in Neural Information Processing Systems*, 2023a.

Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-stage 3d whole-body mesh recovery with component aware transformer. In *Computer Vision and Pattern Recognition*, pp. 21159–21168, 2023b.

Yuanxun Lu, Jinxiang Chai, and Xun Cao. Live speech portraits: real-time photorealistic talking-head animation. *ACM Transactions on Graphics*, 40(6):1–17, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3d hand pose estimation for whole-body 3d human mesh estimation. In *Computer Vision and Pattern Recognition Workshops*, pp. 2307–2316, 2022.

J Nocedal and S Wright. *Numerical Optimization*. Springer, 2006.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Computer Vision and Pattern Recognition*, pp. 10975–10985, 2019.

István Sárándi and Gerard Pons-Moll. Neural localizer fields for continuous 3d human pose and shape estimation. In *Advances in Neural Information Processing Systems*, 2024.

Kaiyue Shen, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Julien Valentin, Jie Song, and Otmar Hilliges. X-avatar: Expressive human avatars. In *Computer Vision and Pattern Recognition*, pp. 16911–16921, 2023.

Qingping Sun, Yanjun Wang, Ailing Zeng, Wanqi Yin, Chen Wei, Wenjia Wang, Haiyi Mei, Chi-Sing Leung, Ziwei Liu, Lei Yang, and Zhongang Cai. Aios: All-in-one-stage expressive human pose and shape estimation. In *Computer Vision and Pattern Recognition*, pp. 1834–1843, 2024.

Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision*, pp. 614–631, 2018.

Xuanyu Wang, Weizhan Zhang, Christian Sandor, and Hongbo Fu. Real-and-present: Investigating the use of life-size 2d video avatars in hmd-based AR teleconferencing. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):5626–5641, 2025.

Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Computer Vision and Pattern Recognition*, pp. 10965–10974, 2019.

Jun Xiang, Yudong Guo, Leipeng Hu, Boyang Guo, Yancheng Yuan, and Juyong Zhang. Expressive talking human from single-image with imperfect priors. In *International Conference on Computer Vision*, pp. 10398–10409, 2025.

Fu Chia Yang, Harrison Jesse Smith, Christos Mousas, and Michael Neff. The motion is the message: Evaluating motion tracking quality for VR avatars. *IEEE Transactions on Visualization and Computer Graphics*, 32(5):3615–3625, 2026.

Wanqi Yin, Zhongang Cai, Ruisi Wang, Ailing Zeng, Chen Wei, Qingping Sun, Haiyi Mei, Yanjun Wang, Hui En Pang, Mingyuan Zhang, Lei Zhang, Chen Change Loy, Atsushi Yamashita, Lei Yang, and Ziwei Liu. Smplest-x: Ultimate scaling for expressive human pose and shape estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Hongwen Zhang, Yating Tian, Yuxiang Zhang, Mengcheng Li, Liang An, Zhenan Sun, and Yebin Liu. Pymaf-x: Towards well-aligned full-body model regression from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12287–12303, 2023.

Qitao Zhao and Shubham Tulsiani. Sparse-view pose estimation and reconstruction via analysis by generative synthesis. In *Advances in Neural Information Processing Systems*, 2024.

This appendix provides the analysis and experimental details supporting the main text. Appendix A analyzes the objective in Section 4.3; Appendix B expands the optimization procedure in Section 4.4. Appendix C specifies the targets, evaluation cohorts, baselines, and metrics used in Section 5. Appendix D extends the geometric, joint-success, and visual comparisons in Sections 5.2 and 5.3.

# A Analysis of the Objective

This section supplements the objective in Section 4.3. We first relate the matrix loss to rotation angle, then characterize the hand loss's admissible representations and positive-pivot margin. Targets and clean-reference factors are fixed throughout, and all derivations use exact real arithmetic.

## A.1 Rotation-Loss Calibration

The matrix loss for root, body, and jaw rotations has a direct angular interpretation. For $\mathbf{R}, \mathbf{Q} \in \mathrm{SO}(3)$ with relative angle $\vartheta \in [0, \pi]$, the identity $\mathrm{tr}(\mathbf{R}^\top \mathbf{Q}) = 1 + 2\cos\vartheta$ gives

$$\|\mathbf{R} - \mathbf{Q}\|_F^2 = 6 - 2\,\mathrm{tr}(\mathbf{R}^\top \mathbf{Q}) = 8\sin^2(\vartheta/2). \tag{15}$$

Consequently, the normalization $\kappa_R = \sqrt{8}\sin(\tau_R/2)$, with $0 < \tau_R \le \pi$, yields

$$\frac{\|\mathbf{R} - \mathbf{Q}\|_F^2}{\kappa_R^2} = \frac{\sin^2(\vartheta/2)}{\sin^2(\tau_R/2)}. \tag{16}$$

Thus, the normalized squared discrepancy equals one at the reference angle $\vartheta = \tau_R$ and increases monotonically over $[0, \pi]$. Because it depends only on the relative rotation, it assigns the same value to different axis-angle representations of the same orientation.

Expression and camera losses normalize coefficient-wise mean squared error and squared translation distance by their respective reference scales. These scales set optimization penalties; the separate thresholds used to declare attack success are specified in Appendix C.5.

## A.2 Analysis of the Hand Representation

For one finger joint, omit the joint index and write the raw output as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2] \in \mathbb{R}^{3\times 2}$. The target $\mathbf{R}^\star_{\mathrm{reg}}$ is expressed in the coordinates of $G(\mathbf{A})$, before conversion to native pose parameters (Appendix C.1). We characterize the admissible target representations, the margin, and the loss's local behavior in turn.

**Rotation preimages.** Gram–Schmidt decoding maps a full-column-rank raw output to a rotation:

$$\begin{aligned} \mathbf{q}_1 &= \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2}, \qquad \mathbf{v}_2 = \mathbf{a}_2 - \mathbf{q}_1(\mathbf{q}_1^\top \mathbf{a}_2), \\ \mathbf{q}_2 &= \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|_2}, \quad G(\mathbf{A}) = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_1 \times \mathbf{q}_2]. \end{aligned} \tag{17}$$

Let $\mathcal{T}_+$ denote the upper-triangular $2 \times 2$ matrices with positive diagonal, and let $\mathbf{I}_{3,2} = [\mathbf{e}_1, \mathbf{e}_2]$. Right multiplication by $\mathbf{T} \in \mathcal{T}_+$ preserves the first normalized column and positively scales the second orthogonal component. Conversely, any output decoding to $\mathbf{R}$ has a positive-diagonal thin QR factorization with orthonormal factor $\mathbf{R}\mathbf{I}_{3,2}$. Thus,

$$G(\mathbf{A}\mathbf{T}) = G(\mathbf{A}), \qquad G^{-1}(\mathbf{R}) = \{\mathbf{R}\mathbf{I}_{3,2}\mathbf{T} : \mathbf{T} \in \mathcal{T}_+\}. \tag{18}$$

A target rotation therefore specifies an entire family of six-dimensional outputs. The positive diagonal is essential: a negative pivot changes a decoded axis, and a zero pivot makes the ideal decoder singular.

**Clean-reference normalization.** For a full-column-rank clean output, write $\mathbf{A}_0 = \mathbf{Q}_{0,2}\mathbf{T}_0$, where $\mathbf{T}_0 \in \mathcal{T}_+$. Its diagonal entries are denoted by $t_{11}^0, t_{22}^0 > 0$. The target-frame transformation in Section 4.3 then satisfies

$$\mathbf{Z} = (\mathbf{R}^\star_{\mathrm{reg}})^\top \mathbf{A}\mathbf{T}_0^{-1}, \qquad \mathbf{Z}(\mathbf{x}_0) = (\mathbf{R}^\star_{\mathrm{reg}})^\top G(\mathbf{A}_0)\mathbf{I}_{3,2}. \tag{19}$$

Multiplication by $\mathbf{T}_0^{-1}$ removes the clean scale and shear at initialization. In particular, a preservation target gives $\mathbf{Z}(\mathbf{x}_0) = \mathbf{I}_{3,2}$. Using the positive-floor matrix $\mathbf{T}_\mu$ in Equation (9), write $\bar{\mathbf{Z}} = \mathbf{Z}\mathbf{T}_\mu(\mathbf{Z})^{-1}$ and $\ell_h(\mathbf{A}) = \|\bar{\mathbf{Z}} - \mathbf{I}_{3,2}\|_F^2/(2\tau_h^2)$. This is the single-joint contribution to $\mathcal{L}_{\mathrm{hand}}$.

**Positive-pivot margin and decoder sensitivity.** The margin discourages raw representations near a singular decoder even when their decoded rotation matches the target. For a normalized vector $\mathbf{q} = \mathbf{v}/\|\mathbf{v}\|_2$ with $\mathbf{v} \neq \mathbf{0}$,

$$\frac{\partial \mathbf{q}}{\partial \mathbf{v}} = \frac{\mathbf{I} - \mathbf{q}\mathbf{q}^\top}{\|\mathbf{v}\|_2}. \tag{20}$$

Thus, small normalization denominators in Equation (17) amplify changes transverse to the normalized direction. A loss on the decoded orientation alone does not separately penalize proximity to such representations.

A simple example isolates the effect of the margin. Set $\mathbf{R}^\star_{\mathrm{reg}} = \mathbf{I}_3$, $\mathbf{T}_0 = \mathbf{I}_2$, and $\mathbf{A}_s = [\mathbf{e}_1, \mathbf{e}_1 + s\mathbf{e}_2]$. For every $s > 0$, $G(\mathbf{A}_s) = \mathbf{I}_3$, although the second normalization denominator tends to zero as $s \downarrow 0$. For $s < 0$, the decoded rotation is $\mathrm{diag}(1, -1, -1)$, while the decoder is undefined at $s = 0$. The preimage loss remains defined and becomes

$$\ell_h(\mathbf{A}_s) = \frac{1}{2\tau_h^2}\left(\frac{s}{\max(s, \mu)} - 1\right)^2. \tag{21}$$

For $s < \mu$, the derivative is $(s - \mu)/(\tau_h^2 \mu^2)$, so increasing $s$ toward $\mu$ decreases the loss. Positivity prevents an incorrect axis from being canceled by a signed ratio, while the positive margin also penalizes target-decoding representations whose normalized pivots are too small. This example establishes a preference among raw representations; it does not establish a hand-specific geometric property or an empirical advantage over a uniform rotation loss.

**Zero-loss target recovery.** The zero set consists exactly of target representations satisfying the pivot margin. Define this subset in normalized coordinates as

$$\mathcal{F}_\mu = \left\{ \begin{bmatrix} t_{11} & t_{12} \\ 0 & t_{22} \\ 0 & 0 \end{bmatrix} : t_{11}, t_{22} \geq \mu,\ t_{12} \in \mathbb{R} \right\}. \tag{22}$$

The positive floors make $\mathbf{T}_\mu(\mathbf{Z})$ invertible for every $\mathbf{Z}$, including rank-deficient raw predictions.

**Proposition 1** (Target recovery)**.** *For a full-column-rank clean reference and* $0 < \mu < 1$*,*

$$\ell_h(\mathbf{A}) = 0 \quad \Longleftrightarrow \quad \mathbf{Z} \in \mathcal{F}_\mu \quad \Longrightarrow \quad G(\mathbf{A}) = \mathbf{R}^\star_{\mathrm{reg}}. \tag{23}$$

*Proof.* Since $\mathbf{T}_\mu(\mathbf{Z})$ is invertible, zero loss is equivalent to $\bar{\mathbf{Z}} = \mathbf{I}_{3,2}$ and hence to $\mathbf{Z} = \mathbf{I}_{3,2}\mathbf{T}_\mu(\mathbf{Z})$. On the diagonal, this equality requires $Z_{11}, Z_{22} \geq \mu$. The other entries require $Z_{21} = Z_{31} = Z_{32} = 0$ and leave $Z_{12}$ unrestricted. These conditions are exactly $\mathbf{Z} \in \mathcal{F}_\mu$, proving the equivalence.

At any such point, $\mathbf{A} = \mathbf{R}^\star_{\mathrm{reg}}\mathbf{I}_{3,2}\mathbf{T}_\mu(\mathbf{Z})\mathbf{T}_0$. The final two factors have a positive-diagonal upper-triangular product, so Equation (18) gives $G(\mathbf{A}) = \mathbf{R}^\star_{\mathrm{reg}}$. □

For preservation targets, $\mathbf{Z}(\mathbf{x}_0) = \mathbf{I}_{3,2}$ and $0 < \mu < 1$ place the clean prediction in $\mathcal{F}_\mu$, giving zero clean loss. The zero set admits multiple raw representatives, although loss values away from this set need not be invariant under positive upper-triangular transformations.

**Local bounds on the decoder denominators.** The next proposition bounds the two normalization denominators in Equation (17) away from zero when the hand residual is small.

**Proposition 2** (Local denominator bounds)**.** *Under the same clean-reference assumption, if* $\|\bar{\mathbf{Z}} - \mathbf{I}_{3,2}\|_F \leq \xi < 1$*, then*

$$\|\mathbf{a}_1\|_2 \geq \mu t_{11}^0(1 - \xi), \qquad \mathrm{dist}(\mathbf{a}_2, \mathrm{span}(\mathbf{a}_1)) \geq \mu t_{22}^0(1 - \xi). \tag{24}$$

*Proof.* Set $\mathbf{B} = \bar{\mathbf{Z}} = [\mathbf{b}_1, \mathbf{b}_2]$. The residual assumption implies that, for every $\mathbf{z} \in \mathbb{R}^2$, $\|\mathbf{B}\mathbf{z}\|_2 \geq (1 - \xi)\|\mathbf{z}\|_2$, since $\|\mathbf{B} - \mathbf{I}_{3,2}\|_2 \leq \xi$. Let $\mathbf{U} = \mathbf{T}_\mu(\mathbf{Z})\mathbf{T}_0$. Its diagonal satisfies $U_{11} \geq \mu t_{11}^0$ and $U_{22} \geq \mu t_{22}^0$. Expanding $\mathbf{A} = \mathbf{R}^\star_{\mathrm{reg}}\mathbf{B}\mathbf{U}$ gives $\mathbf{a}_1 = \mathbf{R}^\star_{\mathrm{reg}}\mathbf{b}_1 U_{11}$ and $\mathbf{a}_2 = \mathbf{R}^\star_{\mathrm{reg}}(\mathbf{b}_1 U_{12} + \mathbf{b}_2 U_{22})$.

The first identity, together with $\|\mathbf{b}_1\|_2 \geq 1-\xi$, proves the first bound. For the second, orthogonal projection removes the component parallel to $\mathbf{a}_1$, giving

$$\begin{aligned}\operatorname{dist}(\mathbf{a}_2, \operatorname{span}(\mathbf{a}_1)) &= U_{22} \inf_{\alpha\in\mathbb{R}} \|\mathbf{b}_2 - \alpha \mathbf{b}_1\|_2 \\ &\geq U_{22}(1-\xi) \inf_{\alpha\in\mathbb{R}} \|(-\alpha, 1)^\top\|_2 \geq \mu t_{22}^0 (1-\xi).\end{aligned} \tag{25}$$

□

These bounds apply locally to the decoder and depend on $t_{11}^0$ and $t_{22}^0$. They do not establish uniform conditioning across clean references or for the full image-to-output mapping.

**Local interpretation of the hand scale.** To interpret $\tau_h$ near the unit reference, consider $\mathbf{Z} = \operatorname{Exp}([\boldsymbol{\omega}]_\times)\mathbf{I}_{3,2}$, with $[\boldsymbol{\omega}]_\times$ denoting the cross-product matrix. The floors are inactive in a neighborhood of this reference, and first-order expansion yields

$$\bar{\mathbf{Z}} - \mathbf{I}_{3,2} = \begin{bmatrix} 0 & 0 \\ \omega_z & 0 \\ -\omega_y & \omega_x \end{bmatrix} + O(\|\boldsymbol{\omega}\|_2^2). \tag{26}$$

Hence $\tau_h\sqrt{2\ell_h} = \|\boldsymbol{\omega}\|_2 + O(\|\boldsymbol{\omega}\|_2^2)$ at this reference. This local relation does not extend to arbitrary finite rotations or shears. When differentiating the hand loss, gradients include the dependence of $\mathbf{T}_\mu(\mathbf{Z})$ on $\mathbf{Z}$ on each smooth branch.

## B Optimization and Implementation

This section expands the optimization procedure in Section 4.4: residuals and curvature, gradients through predicted crops, the two-stage search, and numerical settings. Both stages use the same fixed targets and objective.

### B.1 Residuals and Gauss–Newton Curvature

The componentwise residuals determine the Gauss–Newton approximation used in integer refinement. To express the four losses in Section 4.3 in this form, let vec stack matrix columns and col concatenate vectors, and define

$$\begin{aligned}\mathbf{r}_{R,j} &= \kappa_R^{-1} \operatorname{vec}(\mathbf{R}_j - \mathbf{R}_j^\star), \\ \mathbf{r}_{h,j} &= \tau_h^{-1} \operatorname{vec}(\mathbf{Z}_j \mathbf{T}_\mu(\mathbf{Z}_j)^{-1} - \mathbf{I}_{3,2}), \\ \mathbf{r}_\psi &= (\sqrt{d_\psi}\tau_\psi)^{-1}(\boldsymbol{\psi} - \boldsymbol{\psi}^\star), \qquad \mathbf{r}_c = \tau_c^{-1}(\mathbf{c} - \mathbf{c}^\star).\end{aligned} \tag{27}$$

Their concatenation gives the equivalent least-squares objective

$$\mathbf{r}(\mathbf{x}) = \operatorname{col}\big(\{\mathbf{r}_{R,j}\}_{j\in\mathcal{B}}, \{\mathbf{r}_{h,j}\}_{j\in\mathcal{H}}, \mathbf{r}_\psi, \mathbf{r}_c\big), \qquad \mathcal{L}(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{r}(\mathbf{x})\|_2^2. \tag{28}$$

The identity $\|\operatorname{vec}(\mathbf{M})\|_2^2 = \|\mathbf{M}\|_F^2$ shows that this representation preserves the scales and aggregation of all four losses. The residual dimension is $m = 9|\mathcal{B}| + 6|\mathcal{H}| + d_\psi + d_c$.

The residual Jacobian $\mathbf{J}_r = \partial\mathbf{r}/\partial\mathbf{x}$ gives $\nabla\mathcal{L} = \mathbf{J}_r^\top\mathbf{r}$ wherever the residuals are differentiable. On a twice continuously differentiable branch, the Hessian is

$$\nabla^2\mathcal{L} = \mathbf{J}_r^\top\mathbf{J}_r + \sum_{a=1}^{m} r_a \nabla^2 r_a. \tag{29}$$

The Gauss–Newton approximation retains the first term. We compute its diagonal at the initial quantized image $\mathbf{x}_q$ and keep it fixed during integer refinement:

$$h_i = \|\mathbf{J}_r(\mathbf{x}_q)_{:,i}\|_2^2, \qquad i = 1, \ldots, d. \tag{30}$$

Replacing the residual entries with square roots of group losses can preserve the scalar loss while changing this Gauss–Newton curvature.

**Residual ordering in H32.** The H32 implementation uses an equivalent ordering: it flattens non-finger rotation matrices by rows and stores each hand vector as $\tau_h^{-1}(\bar{Z}_{21}, \bar{Z}_{31}, \bar{Z}_{32}, \bar{Z}_{11} - 1, \bar{Z}_{22} - 1, \bar{Z}_{12})^\top$, where $\bar{\mathbf{Z}} = \mathbf{Z}\mathbf{T}_\mu(\mathbf{Z})^{-1}$. After concatenation with the expression and camera entries, the resulting vector differs from Equation (28) by a fixed permutation, $\widetilde{\mathbf{r}} = \mathbf{P}\mathbf{r}$. Since $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$, both the loss gradient and $\mathbf{J}_r^\top\mathbf{J}_r$ remain unchanged.

### B.2 Input Gradients through Predicted Crops

Predicted crop coordinates provide an additional path from the image to the sampled features. For image-dependent features $\mathbf{F}(\mathbf{x})$, crop coordinates $\mathbf{b}(\mathbf{x})$, and sampling operator $\mathcal{V}$, the full input derivative contains two terms:

$$\frac{d\mathcal{V}}{d\mathbf{x}} = \frac{\partial\mathcal{V}}{\partial\mathbf{F}}\frac{d\mathbf{F}}{d\mathbf{x}} + \frac{\partial\mathcal{V}}{\partial\mathbf{b}}\frac{d\mathbf{b}}{d\mathbf{x}}. \tag{31}$$

**Coordinate-gradient adapter.** When the original ROIAlign operator provides only the feature derivative, we add the coordinate derivative through an auxiliary implementation, $\mathcal{V}_{\text{aux}}$. It evaluates the same piecewise bilinear sampling expression, including its averaging and boundary conventions. Let sg denote stop-gradient, and set $\mathbf{U} = \mathcal{V}_{\text{aux}}(\text{sg}(\mathbf{F}), \mathbf{b})$ and

$$\widetilde{\mathcal{V}}(\mathbf{F}, \mathbf{b}) = \mathcal{V}(\mathbf{F}, \text{sg}(\mathbf{b})) + [\mathbf{U} - \text{sg}(\mathbf{U})]. \tag{32}$$

The bracketed term is zero in the forward pass, preserving the original sampling output. The adapter recovers Equation (31) under local conditions: sampling counts, interpolation cells, and clipping branches must remain fixed; both operators must represent the same differentiable function; and the original operator must supply the correct feature derivative. Stop-gradient then assigns the feature derivative to the original operator and the coordinate derivative to $\mathbf{U}$, avoiding duplicate contributions. At branch boundaries, a classical derivative need not exist; differentiation follows the selected branch.

**Implementation and numerical checks.** The H32 adapter uses `aligned=False`, adaptive average sampling, and the original boundary rules. Float64 intermediates reproduce float32 fused multiply-add rounding at coordinate boundaries, while the network remains in float32. A runtime guard checks the auxiliary and original sampling values with normalized maximum tolerance $2 \times 10^{-5}$.

Numerical validation covers 18 development points: three inputs, each evaluated for three targets at the clean and initially quantized images. Predicted parameters and residuals agree exactly between the original and adapted forward passes at these points. Table 4 summarizes the numerical comparisons. Of the 34 above-tolerance comparisons without a monitored branch change, follow-up checks attribute the largest discrepancy component to forward precision in 33 cases and to an activation-branch contribution in one case. These checks assess sampled derivatives; they do not exhaustively verify the Jacobian or curvature.

Table 4: Numerical gradient checks for H32. The first three rows partition the 356 comparisons without a monitored branch change; the 148 comparisons crossing monitored branches are reported separately.

| Comparison category | Count |
|---|---|
| No monitored branch change, within tolerance | 306 |
| No monitored branch change, above tolerance | 34 |
| No monitored branch change, small signal | 16 |
| All comparisons without a monitored branch change | 356 |
| Comparisons crossing monitored branches | 148 |

### B.3 Two-Stage Optimization

Continuous optimization searches within the pixel bounds; integer refinement then starts from the nearest feasible eight-bit image. In both stages, the accepted iterate determines where the search

proceeds, while a separate record retains the best feasible candidate evaluated so far. A trial can improve this record without being accepted as the next iterate.

**Bounded continuous search.** Write the feasible box as $\mathcal{C} = [\boldsymbol{\ell}, \mathbf{u}]$, with bounds $\ell_i = \max(0, x_{0,i} - \epsilon)$ and $u_i = \min(1, x_{0,i} + \epsilon)$. Starting from $\mathbf{x}_0$, L-BFGS-B (Byrd et al., 1995) identifies active bounds through a generalized Cauchy point along a projected-gradient path, refines the free coordinates by subspace minimization, and performs a feasible line search. Image and gradient differences provide its limited-memory curvature approximation. The best-candidate record $\mathbf{x}_c$ tracks every evaluated feasible point, including line-search trials.

**Continuation after a solver return.** The controller tests the best retained point using the projected-gradient mapping

$$\mathbf{p}_{\mathcal{C}}(\mathbf{x}) = \mathbf{x} - \Pi_{\mathcal{C}}\big(\mathbf{x} - \nabla\mathcal{L}(\mathbf{x})\big). \tag{33}$$

At differentiable points, a zero mapping is equivalent to first-order stationarity on the box. If the numerical stationarity test fails and budget remains, the first continuation check permits a new solver call from $\mathbf{x}_c$ with cleared history. Each later continuation additionally requires a loss improvement since the start of the preceding call greater than $\varepsilon_{\text{float32}} \max(1, \mathcal{L}(\mathbf{x}_c))$, where $\varepsilon_{\text{float32}}$ is float32 machine epsilon. Checks and line-search evaluations consume the continuous budget. No further check is issued once that budget is exhausted, and attack success does not trigger early stopping.

**Feasible eight-bit quantization.** Quantization must satisfy both the pixel bounds and the eight-bit constraint. The feasible integer codes have bounds $L_i = \lceil 255\ell_i \rceil$ and $U_i = \lfloor 255u_i \rfloor$. If $L_i \le U_i$ for every coordinate, we set

$$(\mathbf{x}_q)_i = \frac{1}{255}\,\mathrm{clip}\big(\mathrm{round}(255x_{c,i}), L_i, U_i\big). \tag{34}$$

This coordinatewise nearest-code selection minimizes $\|\mathbf{z} - \mathbf{x}_c\|_2^2$ over $\mathcal{C}_8$, because squared distance is separable across the integer intervals. Bounds are computed in float64, and ties are rounded to the nearest even code. The preprocessed clean input need not itself be eight-bit. An empty interval is recorded as a failure, with no change to the perturbation budget.

**Curvature-guided integer refinement.** The curvature accounts for both targeted and preserved outputs, even when a preserved output currently matches its target. For preserved residuals $\mathbf{r}_p$, define $\mathcal{L}_p = \frac{1}{2}\|\mathbf{r}_p\|_2^2$ and $\mathbf{J}_p = \partial\mathbf{r}_p/\partial\mathbf{x}$. On a twice continuously differentiable branch with $\mathbf{r}_p(\mathbf{x}_\circ) = \mathbf{0}$, Equation (29) gives

$$\nabla\mathcal{L}_p(\mathbf{x}_\circ) = \mathbf{0}, \qquad \nabla^2\mathcal{L}_p(\mathbf{x}_\circ) = \mathbf{J}_p(\mathbf{x}_\circ)^\top\mathbf{J}_p(\mathbf{x}_\circ). \tag{35}$$

Thus a matched preservation term can contribute curvature while its gradient is zero. The fixed diagonal in Equation (30) captures this sensitivity at $\mathbf{x}_q$.

**One-code proposals.** At an accepted integer iterate $\mathbf{x} \in \mathcal{C}_8$, each coordinate may remain unchanged or move by one feasible intensity code. Let $D_i(\mathbf{x}) = \{0, \pm 1/255\} \cap [\ell_i - x_i, u_i - x_i]$. With $\mathbf{g} = \nabla\mathcal{L}(\mathbf{x})$, define

$$\mathcal{S}_K(\mathbf{x}) = \{\mathbf{s} \in \mathbb{R}^d : s_i \in D_i(\mathbf{x}),\ \|\mathbf{s}\|_0 \le K\}, \qquad q_{\mathbf{x}}(\mathbf{s}) = \sum_{i=1}^{d}\big(g_i s_i + \tfrac{1}{2}h_i s_i^2\big). \tag{36}$$

Here $K \in \{1, \dots, d\}$ limits the number of changed coordinates. The proposal minimizes $q_{\mathbf{x}}$ over $\mathcal{S}_K(\mathbf{x})$; separability gives the following exact solution.

**Proposition 3** (Optimal separable one-code proposal)**.** *For $h_i \ge 0$, let $\nu_i = \max_{v \in D_i(\mathbf{x})}(-g_i v - \frac{1}{2}h_i v^2)$. Select the $K$ coordinates with the largest positive $\nu_i$, or all coordinates with positive $\nu_i$ if fewer than $K$ exist. At each selected coordinate, choose a nonzero change attaining $\nu_i$; leave all others unchanged. This proposal minimizes $q_{\mathbf{x}}$ over $\mathcal{S}_K(\mathbf{x})$.*

*Proof.* Since $0 \in D_i(\mathbf{x})$, every $\nu_i \ge 0$. For a fixed set $S$ of coordinates allowed to change, the minimum is $-\sum_{i \in S}\nu_i$. Replacing a selected smaller benefit with an omitted larger benefit decreases the model value. An optimal support therefore contains the $K$ largest positive benefits, or all positive benefits if fewer than $K$ exist. Zero-benefit changes can be omitted. Ties are resolved by coordinate order. □

With float32 decoding, adjacent intensity codes can differ by slightly more or less than $1/255$. The implementation scores these actual differences, and the same argument applies to the resulting finite coordinate choices.

**Acceptance and support adjustment.** Both the accepted iterate $\mathbf{x}$ and the best-candidate record $\mathbf{x}_b$ start at $\mathbf{x}_q$, with $K = 1$. For a proposal with $q_{\mathbf{x}}(\mathbf{s}) < 0$, we evaluate the nonlinear loss and compute the agreement between observed and predicted reduction:

$$\rho = \frac{\mathcal{L}(\mathbf{x}) - \mathcal{L}(\mathbf{x} + \mathbf{s})}{-q_{\mathbf{x}}(\mathbf{s})}. \tag{37}$$

A trial becomes the next accepted iterate only if it strictly decreases the loss and $\rho \geq \rho_{\min} = 0.1$. Independently, any feasible trial whose loss is lower than the best recorded loss updates $\mathbf{x}_b$, including a trial that fails the ratio test. Rejection or $\rho < 0.25$ halves $K$, with a lower bound of one. An accepted trial with $\rho > 0.75$ doubles $K$, up to $d$; all other accepted trials leave $K$ unchanged. The gradient is updated at accepted integer images, without differentiating through rounding.

**Stopping and exported output.** Refinement ends when no positive predicted decrease exists, the integer budget is exhausted, or a proposal is rejected with $K = 1$ before the support adjustment. Strict descent prevents repeated accepted lattice points, and the best recorded image has an evaluated loss no greater than that of $\mathbf{x}_q$. Proposition 3 guarantees optimality for the separable proposal model. It does not establish optimality for the nonlinear objective, because the model omits residual Hessians, cross-coordinate curvature, and curvature changes after $\mathbf{x}_q$. Finally, $\mathbf{x}_b$ is checked for feasibility, exported as a PNG, and reloaded for evaluation with the original estimator. Algorithm 1 summarizes the complete procedure.

**Algorithm 1:** Two-stage optimization of CTRLRECON
**Input:** Clean image $\mathbf{x}_0$, fixed estimator, requested blocks and values, pixel budget, loss scales, and optimization budgets.

1. Construct the target $\mathbf{y}^\star$ and the clean QR factors; keep both fixed throughout optimization.
2. Run bounded continuous search with the continuation rule, retaining the lowest-loss evaluated feasible image $\mathbf{x}_c$.
3. Form $\mathbf{x}_q$ using Equation (34), evaluate its loss and gradient, and compute $\mathbf{h}$. Initialize $\mathbf{x} = \mathbf{x}_b = \mathbf{x}_q$ and $K = 1$.
4. While the integer budget and stopping conditions permit, construct the proposal in Proposition 3 and evaluate its loss. Update $\mathbf{x}_b$ if the trial improves the best recorded loss. Update $\mathbf{x}$ and its gradient only on acceptance, then adjust $K$ using the agreement rule.
5. Export $\mathbf{x}_b$ as a PNG and evaluate the reloaded $\widehat{\mathbf{x}}$ with the original estimator.

**Output:** The exported image and its feasibility and success records.

### B.4 NUMERICAL SETTINGS AND COMPUTATIONAL COST

The H32 configuration uses inputs in $\mathbb{R}^{3\times512\times384}$ and predicts 53 native axis-angle rotations, ten shape coefficients, ten expression coefficients, and three translation coordinates. Thus $|\mathcal{B}| = 23$, $|\mathcal{H}| = 30$, $d_\psi = 10$, and $d_c = 3$, giving $m = 400$. Eye rotations are fixed to zero by the decoder. Table 5 lists the numerical configuration for H32; these settings are not assumed to be identical across estimators.

The continuous solver uses the SciPy 1.10.1 reverse-communication interface and caches objective–gradient pairs for repeated queries at the same point. The network runs in float32, while optimization variables, bounds, and solver state use float64. An empty integer interval or a clean QR pivot of at most $10^{-6}$ raises an error. Failed workers retain the active-request record, and interrupted attempts are explicitly recorded.

**Cost accounting.** Curvature computation has a separate derivative cost. We accumulate the diagonal from scalar residual gradients without storing the full Jacobian:

$$\mathbf{h} = \sum_{a=1}^{m} \nabla_{\mathbf{x}} r_a(\mathbf{x}_q) \odot \nabla_{\mathbf{x}} r_a(\mathbf{x}_q). \tag{38}$$

Table 5: Numerical settings for H32. Loss scales and query budgets are fixed across the three targets.

| Quantity | Setting |
|---|---|
| Checkpoint / processed RGB input | SMPLer-X-H32 / $3 \times 512 \times 384$ |
| Pixel constraint | $\epsilon = 0.06$, range $[0, 1]$ |
| Rotation / expression scales $\tau_R, \tau_\psi$ | $5\pi/180$ / 0.1 |
| Hand margin / scale $\mu, \tau_h$ | 0.1 / $5\pi/180$ |
| Camera scale $\tau_c$ | 0.1 m |
| Residual dimension $m$ | 400 |
| Continuous / integer / combined query caps | 1,800 / 200 / 2,000 |
| Initial curvature | 400 scalar residual VJPs, separately counted |
| L-BFGS-B history / line-search limit | `maxcor`=10 / `maxls`=20 |
| L-BFGS-B tolerances | `ftol`=$10^{-12}$, `gtol`=$10^{-8}$ |
| Solver guard limits | `maxfun`=`maxiter`=2,000 |
| Conditional history reset | From the best feasible candidate |
| Acceptance threshold / initial support | $\rho_{\min} = 0.1$ / $K = 1$ |
| Arithmetic | Network float32, optimizer float64, TF32 off |
| Batch capacity / hardware | 16 / NVIDIA L20 |
| Seed in the H32 campaign | 20260909 |

Here $\odot$ denotes elementwise multiplication. Computing the sum requires $m$ scalar residual vector–Jacobian products (VJPs), counted separately from attack queries.

The query budgets cover evaluations performed by the two search stages. All continuous objective and gradient evaluations, including line-search trials and continuation checks, count toward $Q_c = 1{,}800$. The integer cap $Q_d = 200$ includes the initial quantized evaluation. Unused continuous queries are not transferred to integer refinement, and accepted updates are counted separately from queries.

We record initial curvature VJPs, preparation and evaluation forward passes, loss-gradient backward passes, and elapsed time in separate categories. Batched execution additionally records speculative gradients for rejected trials, effective batch size, and amortized optimization time. The combined 2,000-query cap therefore does not bound total derivative computation. Comparisons with matched update counts need not have matched execution times or computational costs.

# C Experimental Protocol

This section specifies target construction, evaluation subsets, baseline configurations, and ablation controls. It defines geometric error, joint attack success, and visual voting separately, retaining the conventions of the recorded experiments.

## C.1 Target Definitions and Estimator Conventions

Body, Hand, and Expression are three separate target families. For each estimator and input, target construction starts from the estimator's clean prediction: the requested blocks are replaced by the corresponding template, and all other blocks retain their clean values. The clean shape $\boldsymbol{\beta}^\star = \boldsymbol{\beta}(\mathbf{x}_0)$ is used to construct the reference mesh and is included in the baselines' raw-parameter loss. It is not penalized by the objective of CtrlRecon. H32 fixes eye rotations to zero.

**Body target.** The Body template specifies a sitting pose with raised arms by replacing 19 local rotations in the 21-joint body block, which excludes the pelvis. Table 6 lists the replacement axis-angle vectors; indices 11 and 14 retain their clean values. The body block includes the wrists, while finger rotations form separate blocks. The template specifies articulation without enforcing chair contact or other scene constraints.

**Hand target.** The Hand template replaces all 15 right-hand finger rotations with a peace gesture. Table 7 lists the decoded axis-angle values in SMPL-X joint order, rounded to six decimal places. Before assignment, these values are converted to radians and the estimator's actual decoder mean is

subtracted, as detailed below. The resulting parameters replace the right-hand prediction; they are not increments to the clean hand. The left hand and all other inactive blocks retain their clean values.

**Expression target.** The Expression template, labeled Face in the tables, replaces the jaw axis-angle vector with $(14, 0, 5)$ degrees and sets all ten expression coefficients to

$$\boldsymbol{\psi}^\star = (2.7, 0, -1.2, 2.5, 0, 1.2, 0, 0, 0, 0)^\top. \tag{39}$$

All ten coefficients are prescribed, including the zero entries.

Table 6: Body target constants, local axis-angle vectors in degrees. Vectors are not Euler-angle triplets.

| Index | Axis-angle | Index | Axis-angle |
|---|---|---|---|
| 0 | $(-95, 0, 10)$ | 1 | $(-95, 0, -10)$ |
| 2 | $(10, 0, 0)$ | 3 | $(105, 0, 0)$ |
| 4 | $(105, 0, 0)$ | 5 | $(10, 0, 0)$ |
| 6 | $(-10, 0, 0)$ | 7 | $(-10, 0, 0)$ |
| 8 | $(5, 0, 0)$ | 9 | $(0, 0, 0)$ |
| 10 | $(0, 0, 0)$ | 12 | $(0, 0, 10)$ |
| 13 | $(0, 0, -10)$ | 15 | $(0, 0, 55)$ |
| 16 | $(0, 0, -55)$ | 17 | $(0, -25, 0)$ |
| 18 | $(0, 25, 0)$ | 19 | $(0, 0, 0)$ |
| 20 | $(0, 0, 0)$ | | |

Table 7: Right-hand target constants in degrees, shown to six decimal places. The zero-based indices follow the 15-joint right-hand block, and the vectors specify decoded axis-angle rotations before mean subtraction.

| Index | $\theta_x$ | $\theta_y$ | $\theta_z$ |
|---|---|---|---|
| 0 | 0.000000 | 15.000000 | 0.000000 |
| 1 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.000000 | -10.000000 | 0.000000 |
| 4 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 0.000000 | 0.000000 | 0.000000 |
| 6 | -52.742348 | 15.410321 | 67.573909 |
| 7 | -39.770791 | -32.516082 | 61.419143 |
| 8 | -38.155167 | -15.482101 | 67.219550 |
| 9 | -16.090765 | 1.609077 | 85.813645 |
| 10 | -20.971972 | -10.901484 | 70.087897 |
| 11 | -21.635886 | -19.629683 | 75.248370 |
| 12 | 80.000000 | -43.489839 | 24.316309 |
| 13 | -5.780990 | -12.529146 | 1.216748 |
| 14 | 18.343505 | -25.000000 | 19.227032 |

**Coordinate and decoder conventions.** Target assignment requires distinguishing the regressor representation, the native output, and the decoded rotation. For raw output $\mathbf{A}_j$, the regressor rotation is $\mathbf{R}_{\mathrm{reg},j} = G(\mathbf{A}_j)$, while the native rotation is $\mathbf{R}_j = \mathrm{Exp}([\boldsymbol{\theta}_j]_\times)$. H32 predicts six-dimensional root, body, finger, and jaw representations, then orthogonalizes them and converts them to native axis-angle parameters. The non-finger loss uses matrices reconstructed from these native parameters; the hand loss operates on the raw outputs before conversion.

H32 additionally restores its mirrored left-hand crop by negating the second and third native axis-angle coordinates. Thus, a native left-hand target $(\theta_x^\star, \theta_y^\star, \theta_z^\star)$ must be mapped to $(\theta_x^\star, -\theta_y^\star, -\theta_z^\star)$ before conversion to $\mathbf{R}^\star_{\mathrm{reg},j}$. Right-hand targets require no sign change.

For native finger parameters $\boldsymbol{\theta}_j$ and decoder mean $\bar{\boldsymbol{\theta}}_j$, the decoded rotation and template conversion are

$$\mathbf{R}_j^{\mathrm{dec}} = \mathrm{Exp}([\boldsymbol{\theta}_j + \bar{\boldsymbol{\theta}}_j]_\times), \qquad \boldsymbol{\theta}_j^\star = \boldsymbol{\theta}_j^{\mathrm{full},\star} - \bar{\boldsymbol{\theta}}_j. \tag{40}$$

The decoder adds the mean to the axis-angle vector in radians rather than composing two rotation matrices. Consequently, equivalent native axis-angle representatives can produce different decoded rotations after mean addition. Evaluation therefore retains each estimator's conversion routine and checks both native and mean-adjusted hand rotations.

Round-trip tests on three H32 development inputs give maximum discrepancies of $1.81 \times 10^{-7}$ rad for native rotations and $2.39 \times 10^{-7}$ rad for mean-adjusted rotations, both below the $10^{-4}$ rad tolerance. These checks validate the H32 conversions on those inputs; the conversion conventions remain estimator-specific.

### C.2 Evaluation Subsets and Budget Calibration

**Evaluation cohorts and exclusions.** The held-out UBody cohort contains 2,351 inputs per estimator and target. The PW3D geometric evaluation contains 2,009 H32 inputs and 2,004 inputs for each of OSX-L and Hand4Whole after the recorded exclusions. The excluded samples are original index 2 for H32 and IDs 32666, 52491, 66626, 51905, 42749, and 56271 for OSX-L/Hand4Whole. All methods use the same exclusions within an estimator, and unsuccessful attacks remain in the reported means and rates.

**Development-based update budgets.** PGD and C&W receive an update budget calibrated separately for each model–dataset–target group:

$$N = \left\lceil \frac{1}{|\mathcal{D}_{\text{dev}}|} \sum_{i \in \mathcal{D}_{\text{dev}}} n_i^{\text{cont}} \right\rceil, \tag{41}$$

Here, $n_i^{\text{cont}}$ is the number of continuous updates accepted by CTRLRECON on development input $i$. Calibration includes completed unsuccessful attacks; incomplete attempts are not counted as zero-update runs. Table 8 gives the resulting budgets. Each C&W run shares its total budget across three stages, as specified in Appendix C.3, while FGSM always takes one step.

Table 8: PGD and C&W update budgets $N$ from development calibration. Each C&W run shares one budget across its three stages.

| Dataset | Estimator | Body | Hand | Expression |
|---|---|---|---|---|
| UBody | H32 | 1,070 | 1,105 | 1,754 |
| | OSX-L | 504 | 650 | 1,746 |
| | Hand4Whole | 1,513 | 1,162 | 1,561 |
| PW3D | H32 | 848 | 1,283 | 1,187 |
| | OSX-L | 658 | 1,558 | 1,765 |
| | Hand4Whole | 1,090 | 1,207 | 1,569 |

The UBody development set uses indices $0, 784, 1569$, corresponding to IDs 36421, 56886, and 90633 in the OSX-L/Hand4Whole exports. For H32, the recorded accepted-update counts are $(1223, 983, 1004)$ for Body, $(1539, 1077, 699)$ for Hand, and $(1777, 1731, 1752)$ for Expression.

### C.3 Baseline Configurations

FGSM, PGD, and C&W all use the same unweighted raw-parameter matching loss:

$$L_{\text{raw}}(\mathbf{x}; \mathbf{y}^\star) = \left\| \text{col}_{k \in \mathcal{K}} \, \text{vec}\big(f_k(\mathbf{x}) - \mathbf{y}_k^\star\big) \right\|_2, \tag{42}$$

where $\mathcal{K}$ contains every output parameter block, including shape. For H32, the vector has $53 \cdot 3 + 10 + 10 + 3 = 182$ coordinates; poses are native axis-angle values in radians before hand-mean addition, and translation is in meters. Equation (42) uses the unsquared L2 norm without balancing units or weighting parameter groups. It contains no rotation-matrix, decoded-geometry, or positive-preimage terms. It therefore does not define a physically uniform distance across output blocks.

**FGSM and PGD.** Both initialize at $\mathbf{x}^{(0)} = \mathbf{x}_0$ and use

$$\mathbf{x}^{(k+1)} = \Pi_{\mathcal{C}} \left( \mathbf{x}^{(k)} - \alpha \, \text{sign} \, \nabla_{\mathbf{x}} L_{\text{raw}}(\mathbf{x}^{(k)}; \mathbf{y}^\star) \right). \tag{43}$$

FGSM takes one step with $\alpha = 0.06$. PGD uses $\alpha = 0.001$ for the $N$ updates calibrated in Appendix C.2. Both pass their final iterate to the feasible quantizer. The recorded H32 configuration uses no random initialization, success-based early stopping, or best-of-restart selection.

**C&W regression adaptation.** For the pixel bounds in Appendix B.3, define $\mathbf{x}_{\mathrm{mid}} = (\mathbf{u} + \boldsymbol{\ell})/2$ and $\mathbf{r}_{\mathrm{box}} = (\mathbf{u} - \boldsymbol{\ell})/2$. The bounded parameterization $\mathbf{x}(\mathbf{w}) = \mathbf{x}_{\mathrm{mid}} + \mathbf{r}_{\mathrm{box}} \odot \tanh(\mathbf{w})$ yields

$$L_{\mathrm{CW}}(\mathbf{w}; \lambda_{\mathrm{CW}}) = \frac{\|\mathbf{x}(\mathbf{w}) - \mathbf{x}_0\|_2^2}{d} + \lambda_{\mathrm{CW}} L_{\mathrm{raw}}(\mathbf{x}(\mathbf{w}); \mathbf{y}^\star). \tag{44}$$

The three stages use $\lambda_{\mathrm{CW}} \in \{0.1, 1, 10\}$. Each starts from a fresh clean-derived $\mathbf{w}_0$ and an independent Adam state with learning rate $0.01$. To initialize $\mathbf{w}_0$, we clip $(\mathbf{x}_0 - \mathbf{x}_{\mathrm{mid}})/\mathbf{r}_{\mathrm{box}}$ to $[-0.999999, 0.999999]$ before applying $\mathrm{atanh}$, using a $10^{-12}$ radius floor in the division. The objective is a bounded regression adaptation of C&W and does not use a classification-margin loss.

The three stages divide a single total update budget $N$. Writing $N = 3n_0 + r_N$ with $r_N \in \{0, 1, 2\}$, the first $r_N$ stages receive $n_0 + 1$ updates and the remaining stages receive $n_0$. Optimization continues for the allocated updates even after a candidate passes the internal selection test.

C&W selects among evaluated pre-update iterates, excluding the unevaluated final state. Among candidates passing its implementation-specific pose–expression test, it selects the one with minimum pixel distortion. If none passes, it selects the candidate with minimum raw target loss. This internal test is distinct from the joint success criterion used for reporting in Appendix C.5. All three baselines use the same feasible quantizer and are evaluated with the original estimator.

### C.4 Geometric Metrics

Geometric metrics measure the discrepancy between decoded reconstructions and the reference target within specified regions.

**Target-relative geometry.** For corresponding predicted and reference joints and vertices expressed in the same coordinate convention, we use

$$\begin{aligned} \mathrm{MPJPE}(\mathbf{J}, \mathbf{J}^{\mathrm{ref}}) &= \frac{1}{N_J} \sum_{j=1}^{N_J} \|\mathbf{J}_j - \mathbf{J}_j^{\mathrm{ref}}\|_2, \\ \mathrm{MPVPE}(\mathbf{V}, \mathbf{V}^{\mathrm{ref}}) &= \frac{1}{N_V} \sum_{v=1}^{N_V} \|\mathbf{V}_v - \mathbf{V}_v^{\mathrm{ref}}\|_2. \end{aligned} \tag{45}$$

Here, $N_J$ and $N_V$ count the corresponding joints and vertices. Table 9 defines the point sets and anchors for each region. For non-PA vertex errors, each mesh is centered at its own regional anchor $\mathbf{a}_r(\mathbf{y})$, regressed from the posed mesh using $\mathbf{J}_{\mathrm{regressor}}\mathbf{V}(\mathbf{y})$. With $\widetilde{\mathbf{V}}_{r,v}(\mathbf{y}) = \mathbf{V}_v(\mathbf{y}) - \mathbf{a}_r(\mathbf{y})$ and coordinates in meters, the regional error reported in millimeters is

$$\mathrm{MPVPE}_r(\mathbf{y}, \mathbf{y}^\star) = \frac{1000}{|\mathcal{V}_r|} \sum_{v \in \mathcal{V}_r} \left\| \widetilde{\mathbf{V}}_{r,v}(\mathbf{y}) - \widetilde{\mathbf{V}}_{r,v}(\mathbf{y}^\star) \right\|_2. \tag{46}$$

The body vertex set excludes both hands and the face. Body joints are obtained with the J14 mesh regressor, and the right-hand joint set includes the wrist and fingertips. Every selected point is evaluated, including points outside the image. This protocol therefore differs from visibility-masked, bilateral-hand UBody reporting: hand metrics cover the right hand only and use all selected points.

**Procrustes alignment.** PA-MPVPE and PA-MPJPE first fit a separate similarity transformation for each image, region, and point type:

$$(a_{\mathrm{PA}}, \mathbf{R}_{\mathrm{PA}}, \mathbf{t}_{\mathrm{PA}}) \in \underset{a>0,\, \mathbf{R} \in \mathrm{SO}(3),\, \mathbf{t} \in \mathbb{R}^3}{\arg\min} \sum_i \|a\mathbf{R}\mathbf{P}_i + \mathbf{t} - \mathbf{P}_i^\star\|_2^2. \tag{47}$$

The fit uses centering, SVD, and reflection correction to obtain a proper rotation. It rejects a predicted-cloud squared norm at most $10^{-20}$ or a nonpositive fitted scale. The aligned metrics then compute the corresponding mean Euclidean error and multiply by 1,000 to convert meters to

Table 9: Point sets for the reported geometry. A dash denotes a point type or anchor not used by the selected metrics. Non-PA anchors are regressed from the posed mesh, not taken from the legacy LBS pelvis.

| Region | Joints/keypoints | Vertices | Non-PA anchor |
|---|---|---|---|
| All (whole body) | — | 10,475 | Pelvis |
| Hand (right only) | 21 | 778 | Right wrist |
| Face | — | 5,023 | Neck |
| Body | 14 | 3,896 | Pelvis |

millimeters. Because the fit minimizes squared error rather than mean Euclidean error, an aligned metric can occasionally exceed its unaligned counterpart.

The complete comparisons report MPVPE on All, Body, HandR, and Face; PA-MPVPE on All, HandR, and Face; and PA-MPJPE on Body and HandR. The ablations additionally report body-vertex PA-MPVPE.

**Decoding and reference targets.** Both datasets use the regional evaluator `target-regional-ubody-rhand-v1` with a frozen neutral decoder, non-flat hand means, float32 parameters, zero eye poses, and no camera translation. The predicted shape is retained and contributes to geometric error. Camera drift is reported separately as $100\|\mathbf{c} - \mathbf{c}^\star\|_2$ cm.

Geometric evaluation retains the original per-method targets, including small differences in the archived UBody targets for OSX-L and Hand4Whole. On PW3D, targets are bitwise identical across methods for each estimator and input–target pair. Visual voting uses a shared display target, as specified in Appendix C.6.

### C.5 Joint Attack Success

Joint attack success requires both target matching and preservation of inactive outputs in the reloaded image. For rotations $\mathbf{R}, \mathbf{Q} \in \mathrm{SO}(3)$, define the angular error

$$\mathrm{ang}(\mathbf{R}, \mathbf{Q}) = \arccos\left[\mathrm{clip}\left(\frac{\mathrm{tr}(\mathbf{R}^\top \mathbf{Q}) - 1}{2}, -1, 1\right)\right]. \tag{48}$$

For target family $t$, the reloaded image $\widehat{\mathbf{x}}$ is successful only when every constraint below is satisfied:

$$S_t^{\mathrm{joint}}(\widehat{\mathbf{x}}) = \mathbf{1}\begin{bmatrix} \max\limits_{j \in \mathcal{B} \cup \mathcal{H}} \mathrm{ang}(\mathbf{R}_j(\widehat{\mathbf{x}}), \mathbf{R}_j^\star) \le \eta_R^t, \\ \max\limits_{j \in \mathcal{H}} \mathrm{ang}(\mathbf{R}_j^{\mathrm{dec}}(\widehat{\mathbf{x}}), \mathbf{R}_j^{\mathrm{dec},\star}) \le \eta_R^t, \\ \|\boldsymbol{\psi}(\widehat{\mathbf{x}}) - \boldsymbol{\psi}^\star\|_2 / \sqrt{d_\psi} \le \eta_\psi^t, \\ \|\mathbf{c}(\widehat{\mathbf{x}}) - \mathbf{c}^\star\|_2 \le \eta_c^t. \end{bmatrix} \tag{49}$$

**Threshold configuration.** ASR threshold selection should reflect how much deviation from the target an application can tolerate. Because body posture, finger articulation, and facial configuration differ in their required level of detail, thresholds should be chosen for the target family and the output being assessed. Finer control requires tighter tolerances on the relevant outputs, whereas coarser tasks may permit larger deviations. Table 10 lists the thresholds fixed before evaluation and shared across estimators, methods, and ablations. The rotation constraints check every active and preserved rotation individually, including mean-adjusted rotations in both hands. Shape is not part of this success criterion; pixel feasibility is verified separately.

**Aggregation.** For the fixed evaluation set $\mathcal{I}_t$, the attack success rate is

$$\mathrm{ASR}_t = \frac{100}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} S_t^{\mathrm{joint}}(\widehat{\mathbf{x}}_i). \tag{50}$$

The denominator includes unsuccessful attacks and inputs whose clean predictions already meet the target. The evaluation subsets in Appendix C.2 are selected independently of attack success.

Table 10: Target-specific joint success thresholds. Translation is in meters. All native and mean-adjusted hand rotation constraints use the same angular threshold within each target family.

| Target | Rotation $\eta_R^t$ | Expression RMSE $\eta_\psi^t$ | Translation $\eta_c^t$ |
|---|---|---|---|
| Body | $90^\circ$ | 0.50 | 0.30 |
| Hand | $50^\circ$ | 0.30 | 0.20 |
| Expression | $15^\circ$ | 0.80 | 0.20 |

### C.6 VLM Evaluation Protocol

Visual voting assesses how closely the active output blocks match the target. It does not test joint satisfaction of target and preservation constraints. The evaluation uses four judges: GPT-5.6 Sol, Qwen3.5-122B-A10B, Qwen3.8-27B-FP8, and DeepSeek-V4.1-Flash. The evaluation cohort contains 2,351 held-out UBody inputs per target and estimator. The main-text comparison uses H32 and OSX-L, giving 24 estimator–target–judge groups. Table 14 adds Hand4Whole, yielding three estimators and four judges across all three target families. Request/response identifiers are provided with that table.

**Ballot construction and judgment.** Each ballot asks the judge to choose among four attack methods: FGSM, PGD, C&W, and CtrlRecon. A shared target serves as the reference. Front, $-60^\circ$, and $+60^\circ$ views show the candidates and target, with the target in the top-left tile. Anonymous candidate numbers are randomly permuted once per ballot and held consistent across views.

The judge must select the closest candidate even if none matches exactly. The instructions focus on whole-body posture for Body, anatomical right-hand fingers in a wrist-local frame for Hand, and facial configuration in a head-local frame for Expression. They exclude rendering aesthetics and inferred numerical parameters from the judgment.

**Vote aggregation.** The main-text mean weights its 24 groups equally. Within each estimator–target–judge group, method $m$ receives a vote share of $n_m/N$, where $n_m$ is the number of votes for that method and $N$ is the total number of votes across the four attack methods. Table 14 reports these shares for the extended comparison. Vote shares express relative visual preference and do not measure absolute attack success.

### C.7 Ablation Protocol

**Loss and budget variants.** A loss-ablation variant retains the groups indexed by $\mathcal{G}_v \subseteq \{\mathrm{rot}, \mathrm{hand}, \mathrm{expr}, \mathrm{cam}\}$ and optimizes $\mathcal{L}_v = \sum_{a \in \mathcal{G}_v} \mathcal{L}_a$ without rescaling the remaining terms. In H32, the four groups contain 207 non-finger rotation residuals, 180 hand residuals, ten expression residuals, and three camera residuals. Removing each group leaves 193, 220, 390, or 397 residuals, respectively, compared with 400 in the full objective. Curvature is computed from the retained residuals, so these dimensions also specify the initial numbers of scalar residual VJPs.

Budget ablations retain the full objective and use $\epsilon \in \{0.01, 0.03, 0.06, 0.09\}$, reusing the full-method run at $0.06$ as the reference. Table 3 reports each articulation-term removal on its corresponding target; camera removal and budget changes are reported for all three targets. Cross-target cells omitted from that table are intentional omissions rather than unfinished experiments.

**Controlled settings and reporting.** The targets, input images, complete gradients, continuous solver, continuation rule, integer acceptance rule, and query caps remain unchanged across variants. All variants are evaluated with the same target-specific joint criterion: removing a loss term does not remove its corresponding success requirement. Regional PA-MPVPE and camera distance follow Appendix C.4, and joint ASR follows Appendix C.5.

The full held-out cohort is evaluated, including unsuccessful attacks. The ablation implementation splits effective batches of three into $2 + 1$; the reused full-method control executes batches of three together. This numerical difference limits the extent to which small changes can be attributed solely to the ablated factor.

# D Additional Results

## D.1 Geometric Target Agreement

Tables 11 and 12 extend Figure 3 to three estimators and all nine geometric metrics defined in Appendix C.4. Unsuccessful attacks remain in every mean. Across both datasets, CtrlRecon leads whole-mesh MPVPE for Body and face MPVPE for Expression; C&W leads all nine Hand4Whole Hand-target metrics.

Table 11: Target-relative geometry on UBody (mm, ↓). Each estimator–target group contains 2,351 inputs, including unsuccessful attacks. All denotes the whole mesh; Body excludes hands and face; HandR denotes the right hand; H4W denotes Hand4Whole. Bold marks the lowest error within each estimator–target–metric group, including ties at the displayed precision.

| | | MPVPE | | | | PA-MPVPE | | | PA-MPJPE | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | Method | All | Body | HandR | Face | All | HandR | Face | Body | HandR |
| *Body target* | | | | | | | | | | |
| H32 | FGSM | 274.67 | 285.24 | 147.92 | 50.15 | 227.52 | 10.00 | 2.23 | 213.60 | 9.87 |
| | PGD | 431.85 | 352.65 | 84.57 | 139.42 | **76.06** | 2.59 | **1.84** | **80.44** | 2.61 |
| | C&W | 201.68 | 158.17 | 89.77 | 44.06 | 135.01 | 2.36 | 2.07 | 129.71 | 2.35 |
| | CtrlRecon | **138.55** | **110.43** | **61.05** | **36.19** | 96.79 | **2.15** | 2.07 | 93.25 | **2.16** |
| OSX-L | FGSM | 301.21 | 262.55 | 150.77 | 62.73 | 226.29 | 9.45 | 2.88 | 222.03 | 9.50 |
| | PGD | 438.37 | 357.40 | 88.50 | 139.49 | 89.40 | 2.89 | 2.34 | 95.63 | 2.90 |
| | C&W | 293.07 | 221.37 | 123.58 | 50.97 | 170.79 | 2.46 | 2.35 | 171.07 | 2.35 |
| | CtrlRecon | **112.97** | **80.32** | **39.92** | **38.09** | **71.06** | **1.58** | **1.99** | **72.45** | **1.58** |
| H4W | FGSM | 270.60 | 270.50 | 147.96 | 43.27 | 223.38 | 6.25 | 2.07 | 216.69 | 6.24 |
| | PGD | 489.28 | 364.68 | **104.49** | 153.51 | **130.77** | 3.50 | 2.14 | **136.55** | 3.53 |
| | C&W | 246.30 | 184.83 | 109.32 | 39.96 | 171.54 | **1.75** | 1.75 | 173.09 | **1.63** |
| | CtrlRecon | **227.41** | **170.52** | 114.98 | **34.37** | 185.35 | 1.90 | **1.64** | 187.23 | 1.79 |
| *Right-hand target* | | | | | | | | | | |
| H32 | FGSM | 66.16 | 57.25 | 40.46 | 19.02 | 28.16 | 30.25 | 1.37 | 26.84 | 32.96 |
| | PGD | 94.20 | 77.68 | 27.28 | 37.03 | 17.12 | **8.44** | 0.74 | 17.76 | **8.78** |
| | C&W | **25.35** | **20.00** | **16.84** | **10.98** | **11.31** | 8.95 | **0.63** | **11.28** | 9.36 |
| | CtrlRecon | 30.63 | 24.01 | 19.57 | 12.46 | 12.80 | 9.51 | 0.68 | 12.55 | 9.95 |
| OSX-L | FGSM | 77.02 | 68.19 | 50.48 | 31.60 | 34.47 | 28.94 | 1.91 | 32.69 | 31.45 |
| | PGD | 78.90 | 64.24 | 30.72 | 30.04 | 14.10 | 14.97 | 0.58 | 14.25 | 15.94 |
| | C&W | **22.19** | **15.95** | 23.01 | **8.79** | **10.37** | 15.75 | **0.47** | **9.90** | 16.83 |
| | CtrlRecon | 33.62 | 25.07 | **22.83** | 14.45 | 15.34 | **12.13** | 0.92 | 15.24 | **12.84** |
| H4W | FGSM | 72.64 | 66.05 | 48.01 | 26.15 | 31.15 | 31.25 | 1.45 | 30.26 | 34.00 |
| | PGD | 197.55 | 163.27 | 50.09 | 75.37 | 28.34 | 12.91 | 1.10 | 28.19 | 13.42 |
| | C&W | **15.41** | **11.26** | **18.04** | **6.64** | **7.16** | **11.08** | **0.28** | **6.31** | **11.38** |
| | CtrlRecon | 29.15 | 23.79 | 30.35 | 11.12 | 13.21 | 17.03 | 0.52 | 11.78 | 17.97 |
| *Expression target* | | | | | | | | | | |
| H32 | FGSM | 85.64 | 72.34 | 29.84 | 20.66 | 31.05 | 8.91 | 3.68 | 30.66 | 8.82 |
| | PGD | 175.84 | 144.83 | 35.27 | 67.89 | 17.48 | 0.98 | 2.24 | 19.22 | 1.00 |
| | C&W | 68.73 | 58.78 | 17.07 | 27.62 | 17.65 | 1.16 | 2.15 | 19.92 | 1.18 |
| | CtrlRecon | **8.77** | **6.83** | **2.34** | **3.93** | **2.84** | **0.32** | **0.56** | **3.19** | **0.32** |
| OSX-L | FGSM | 86.65 | 73.95 | 32.59 | 44.86 | 35.35 | 6.79 | 3.72 | 32.29 | 6.90 |
| | PGD | 112.25 | 91.30 | 22.16 | 47.35 | 11.66 | 0.66 | 0.95 | 11.99 | 0.67 |
| | C&W | 36.23 | 27.57 | 9.01 | 23.17 | 11.03 | 0.55 | 1.20 | 10.51 | 0.56 |
| | CtrlRecon | **4.39** | **2.72** | **0.97** | **3.37** | **1.60** | **0.10** | **0.23** | **1.44** | **0.10** |
| H4W | FGSM | 70.68 | 63.36 | 31.14 | 27.47 | 30.51 | 5.80 | 3.08 | 29.86 | 5.97 |
| | PGD | 158.96 | 128.66 | 33.10 | 63.57 | 21.23 | 1.31 | 2.25 | 20.82 | 1.34 |
| | C&W | 10.26 | 7.56 | 3.01 | 5.36 | 3.42 | **0.30** | 1.16 | 3.07 | **0.31** |
| | CtrlRecon | **4.41** | **3.44** | **1.94** | **2.11** | **1.90** | 0.37 | **0.70** | **1.72** | 0.38 |

Table 12: Target-relative geometry on PW3D (mm, ↓). Each target uses 2,009 inputs for H32 and 2,004 for OSX-L and Hand4Whole (H4W), including unsuccessful attacks. All denotes the whole mesh; Body excludes hands and face; HandR denotes the right hand. Bold marks the lowest error within each estimator–target–metric group, including ties at the displayed precision.

| | | MPVPE | | | | PA-MPVPE | | | PA-MPJPE | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | Method | All | Body | HandR | Face | All | HandR | Face | Body | HandR |
| *Body target* | | | | | | | | | | |
| H32 | FGSM | 306.60 | 287.92 | 150.92 | 50.91 | 238.50 | 9.12 | 2.36 | 238.73 | 9.03 |
| | PGD | 418.07 | 314.12 | 93.01 | 136.59 | 96.29 | 3.12 | 2.26 | 102.13 | 3.10 |
| | C&W | 271.98 | 205.91 | 108.26 | 67.05 | 162.28 | 2.55 | **2.12** | 167.32 | 2.48 |
| | CTRLRECON | **145.43** | **107.86** | **53.46** | **43.07** | **86.33** | **2.26** | 2.37 | **88.45** | **2.25** |
| OSX-L | FGSM | 315.39 | 276.70 | 139.27 | 76.11 | 230.86 | 7.21 | 2.76 | 227.54 | 7.23 |
| | PGD | 440.95 | 347.84 | 94.24 | 153.41 | 91.79 | 2.84 | 2.41 | 98.74 | 2.85 |
| | C&W | 232.07 | 187.37 | 93.12 | **56.28** | 150.94 | 2.72 | 2.35 | 152.20 | 2.64 |
| | CTRLRECON | **129.72** | **79.81** | **39.22** | 56.98 | **71.72** | **1.42** | **1.58** | **73.94** | **1.42** |
| H4W | FGSM | 296.87 | 282.99 | 141.12 | 63.73 | 233.30 | 6.23 | 2.18 | 229.55 | 6.28 |
| | PGD | 528.06 | 406.30 | 112.70 | 180.25 | **144.97** | 3.30 | 2.81 | **150.27** | 3.31 |
| | C&W | 234.78 | 188.10 | 111.88 | 48.82 | 192.44 | **1.94** | **2.14** | 191.62 | **1.81** |
| | CTRLRECON | **213.31** | **168.81** | **106.32** | **42.98** | 184.68 | 1.98 | 2.29 | 184.02 | 1.87 |
| *Right-hand target* | | | | | | | | | | |
| H32 | FGSM | 62.01 | 48.64 | 43.81 | 29.94 | 31.28 | 31.99 | 1.57 | 32.01 | 34.77 |
| | PGD | 79.41 | 66.31 | 27.69 | 34.83 | 18.86 | 8.76 | 0.64 | 20.12 | 8.99 |
| | C&W | **32.50** | **23.68** | **19.37** | **13.32** | **15.42** | 9.76 | **0.55** | **15.96** | 10.10 |
| | CTRLRECON | 47.15 | 35.11 | 22.93 | 18.57 | 19.38 | **8.64** | 0.91 | 20.00 | **8.95** |
| OSX-L | FGSM | 115.11 | 92.24 | 58.45 | 65.18 | 48.99 | 30.34 | 2.22 | 50.41 | 33.09 |
| | PGD | 71.83 | 57.79 | 28.46 | 30.68 | 14.56 | 15.01 | 0.58 | 15.31 | 15.94 |
| | C&W | **27.14** | **19.17** | **19.31** | **11.20** | **12.77** | 14.45 | **0.55** | **13.29** | 15.36 |
| | CTRLRECON | 41.56 | 29.72 | 21.43 | 18.29 | 18.46 | **12.52** | 0.89 | 19.59 | **13.24** |
| H4W | FGSM | 86.48 | 69.66 | 53.32 | 45.39 | 40.84 | 31.49 | 1.61 | 42.63 | 34.44 |
| | PGD | 188.17 | 155.57 | 56.74 | 85.49 | 31.99 | 13.94 | 1.06 | 33.59 | 14.54 |
| | C&W | **17.00** | **11.51** | **20.11** | **7.48** | **9.60** | **11.90** | **0.34** | **8.82** | **12.25** |
| | CTRLRECON | 27.26 | 20.00 | 30.57 | 11.96 | 14.64 | 18.67 | 0.49 | 13.76 | 19.77 |
| *Expression target* | | | | | | | | | | |
| H32 | FGSM | 73.76 | 57.25 | 33.89 | 37.12 | 35.05 | 8.75 | 3.70 | 37.01 | 8.73 |
| | PGD | 90.03 | 76.85 | 23.95 | 40.18 | 16.11 | 1.16 | 2.71 | 17.68 | 1.17 |
| | C&W | 33.60 | 26.64 | 11.31 | 15.79 | 15.83 | 1.16 | 2.76 | 17.90 | 1.17 |
| | CTRLRECON | **16.08** | **13.67** | **4.16** | **6.26** | **5.35** | **0.53** | **1.52** | **5.90** | **0.51** |
| OSX-L | FGSM | 120.99 | 91.79 | 49.51 | 81.45 | 52.91 | 6.34 | 4.56 | 50.45 | 6.46 |
| | PGD | 78.45 | 62.24 | 16.73 | 36.39 | 10.13 | 0.59 | 0.74 | 10.93 | 0.60 |
| | C&W | 41.40 | 29.79 | 9.17 | 23.89 | 12.71 | 0.58 | 1.09 | 13.27 | 0.59 |
| | CTRLRECON | **1.29** | **0.94** | **0.34** | **0.64** | **0.48** | **0.04** | **0.06** | **0.53** | **0.04** |
| H4W | FGSM | 81.42 | 64.80 | 39.15 | 44.85 | 38.97 | 5.75 | 3.11 | 40.56 | 5.94 |
| | PGD | 160.02 | 130.51 | 41.09 | 77.22 | 25.07 | 1.48 | 2.14 | 26.13 | 1.52 |
| | C&W | 11.44 | 7.93 | 3.83 | 6.76 | 4.25 | **0.36** | 1.23 | 4.14 | **0.37** |
| | CTRLRECON | **3.20** | **2.35** | **1.93** | **1.71** | **1.68** | 0.51 | **0.44** | **1.59** | 0.52 |

### D.2 JOINT ATTACK SUCCESS

Table 13 extends the joint-ASR comparison in Section 5.2 to all three estimators on both UBody and PW3D, using the same success criteria defined in Appendix C.5.

CTRLRECON achieves the highest ASR for every target on H32 and OSX-L across both datasets. Results on Hand4Whole are more mixed: C&W attains the highest ASR for all three targets on UBody, whereas on PW3D, CTRLRECON performs best on the Body target and C&W performs best on the Hand and Expression targets. These results indicate that the consistent ASR advantage observed on H32 and OSX-L does not uniformly extend to Hand4Whole.

Table 13: Joint ASR on UBody and PW3D (%, ↑). Results cover H32, OSX-L, and Hand4Whole under the same target-specific success criteria. Each target uses 2,351 inputs on UBody, 2,009 on PW3D/H32, and 2,004 on PW3D/OSX-L and PW3D/Hand4Whole. Bold marks the highest rate within each estimator and target.

| Estimator | Method | Body | Hand | Expression |
|---|---|---|---|---|
| *UBody* | | | | |
| H32 | FGSM | 0.17 | 0.00 | 0.00 |
| | PGD | 11.95 | 65.04 | 45.98 |
| | C&W | 85.96 | 89.45 | 75.54 |
| | CTRLRECON | **91.96** | **91.45** | **95.32** |
| OSX-L | FGSM | 0.34 | 0.00 | 0.04 |
| | PGD | 27.26 | 33.35 | 73.16 |
| | C&W | 78.43 | 15.48 | 89.75 |
| | CTRLRECON | **97.24** | **95.53** | **99.49** |
| Hand4Whole | FGSM | 0.38 | 0.00 | 0.00 |
| | PGD | 28.07 | 39.05 | 53.68 |
| | C&W | **83.58** | **83.75** | **97.02** |
| | CTRLRECON | 78.99 | 71.16 | 90.81 |
| *PW3D* | | | | |
| H32 | FGSM | 0.95 | 0.00 | 0.00 |
| | PGD | 10.80 | 48.98 | 16.43 |
| | C&W | 69.89 | 92.14 | 73.42 |
| | CTRLRECON | **95.37** | **94.03** | **95.07** |
| OSX-L | FGSM | 0.50 | 0.00 | 0.00 |
| | PGD | 14.72 | 32.49 | 69.16 |
| | C&W | 69.96 | 52.05 | 91.02 |
| | CTRLRECON | **97.06** | **95.16** | **100.00** |
| Hand4Whole | FGSM | 0.40 | 0.00 | 0.00 |
| | PGD | 14.77 | 25.00 | 47.16 |
| | C&W | 73.50 | **59.33** | **97.16** |
| | CTRLRECON | **78.04** | 57.88 | 93.31 |

### D.3 VISUAL TARGET MATCHING

Table 14 extends the visual comparison in Section 5.3 to three estimators and four judge labels, covering 36 estimator–target–judge groups. Each entry is the percentage of votes received by one of the four attack methods, following the protocol in Appendix C.6.

**Target-specific preferences.** CTRLRECON receives the highest share for Body and Face in every reported group. For Hand, CTRLRECON leads on OSX-L and C&W leads on Hand4Whole under all four judge labels. H32 shows a split: PGD leads under GPT, Q3.5, and Q3.8, while CTRLRECON leads under DS. Visual preference evaluates matching of the active blocks, whereas joint ASR also tests preservation. Accordingly, PGD can receive more Q3.8 votes for H32 Hand even when CTRLRECON has higher joint ASR.

Table 14: VLM vote shares on UBody (%, ↑). GPT, Q3.5, Q3.8, and DS label GPT-5.6 Sol, Qwen3.5-122B-A10B, Qwen3.8-27B-FP8, and DeepSeek-V4.1-Flash, respectively; see the note for request/response identifiers. Judges choose among the four attack methods. Vote shares $n_m/N$ within each group are reported as percentages. Bold marks the largest share within each estimator–target–judge group.

| Model | Method | Body target | | | | Hand target (right hand) | | | | Face target | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GPT | Q3.5 | Q3.8 | DS | GPT | Q3.5 | Q3.8 | DS | GPT | Q3.5 | Q3.8 | DS |
| H32 | FGSM | 0.00 | 0.09 | 0.43 | 0.14 | 0.00 | 0.04 | 0.00 | 0.00 | 0.04 | 0.57 | 0.17 | 0.06 |
| | PGD | 28.78 | 27.60 | 27.39 | 26.03 | **35.74** | **35.79** | **34.85** | 38.26 | 5.51 | 5.84 | 5.37 | 5.22 |
| | C&W | 18.64 | 21.17 | 20.10 | 14.71 | 31.91 | 33.05 | 32.55 | 18.26 | 10.38 | 9.97 | 9.84 | 5.45 |
| | CTRLRECON | **52.58** | **51.14** | **52.09** | **59.12** | 32.34 | 31.12 | 32.60 | **43.48** | **84.06** | **83.62** | **84.62** | **89.28** |
| OSX-L | FGSM | 0.04 | 0.21 | 0.09 | 0.04 | 0.00 | 0.04 | 0.00 | 0.04 | 0.04 | 0.39 | 0.21 | 0.56 |
| | PGD | 14.98 | 17.43 | 17.35 | 16.60 | 23.82 | 28.55 | 24.63 | 28.86 | 28.43 | 26.36 | 27.17 | 31.57 |
| | C&W | 0.98 | 1.76 | 1.87 | 1.28 | 13.53 | 24.76 | 17.18 | 23.84 | 19.95 | 13.97 | 20.44 | 26.71 |
| | CTRLRECON | **84.00** | **80.60** | **80.69** | **82.08** | **62.65** | **46.66** | **58.19** | **47.25** | **51.58** | **59.28** | **52.17** | **41.16** |
| Hand4Whole | FGSM | 0.30 | 0.86 | 0.81 | 1.46 | 0.00 | 0.04 | 0.09 | 0.04 | 1.68 | 1.70 | 1.21 | 1.84 |
| | PGD | 30.03 | 22.98 | 23.34 | 27.84 | 37.30 | 33.29 | 35.19 | 32.98 | 1.94 | 1.83 | 2.37 | 3.63 |
| | C&W | 31.43 | 36.20 | 35.97 | 32.36 | **50.53** | **52.32** | **55.15** | **48.35** | 35.31 | 29.17 | 30.81 | 37.47 |
| | CTRLRECON | **38.25** | **39.97** | **39.87** | **38.34** | 12.17 | 14.35 | 9.57 | 18.63 | **61.07** | **67.31** | **65.62** | **57.06** |