# Detection, Pose Estimation and Segmentation for Multiple Bodies: Closing the Virtuous Circle

Miroslav Purkrabek and Jiri Matas
Visual Recognition Group
Czech Technical University in Prague
purkrmir@fel.cvut.cz

(a1) Detection (b1) Pose estimation (c1) Mask refinement (d1) Mask-out (a2) Detection Final masks

Figure 1. The BBox-Mask-Pose (BMP) method. Steps (a) – (d) repeat until no new detections in step (a) are found. The loop can start from a bounding box (a.), pose (b.), or segmentation mask (c.). In this example, the background player is first undetected in step (a1). BMP correctly fits the foreground player's pose (b1) and corrects his segmentation (c1). After masking the foreground player (d1) the background player is detected (a2) and his body segemented (final masks).

## Abstract

*Human pose estimation methods work well on separated people but struggle with multi-body scenarios. Recent work has addressed this problem by conditioning pose estimation with detected bounding boxes or bottom-up-estimated poses. Unfortunately, all of these approaches overlooked segmentation masks and their connection to estimated keypoints. We condition pose estimation model by segmentation masks instead of bounding boxes to improve instance separation. This improves top-down pose estimation in multi-body scenarios but does not fix detection errors. Consequently, we develop BBox-Mask-Pose (BMP), integrating detection, segmentation and pose estimation into self-improving feedback loop. We adapt detector and pose estimation model for conditioning by instance masks and use Segment Anything as pose-to-mask model to close the circle. With only small models, BMP is superior to top-down methods on OCHuman dataset and to detector-free methods on COCO dataset, combining the best from both approaches and matching state of art performance in both settings. Code is available on the project website* [1].

[1] MiraPurkrabek.github.io/BBox-Mask-Pose/

## 1. Introduction

Human pose estimation (HPE) plays a crucial role in tasks like action detection and gesture recognition. It is a challenging problem, especially in multi-body scenes where people overlap, leading to issues such as merged bounding boxes or collapsed poses. Results on multi-body datasets are far from saturated, with state-of-the-art below 50% [37].

Top-down and detector-free methods are the two primary approaches in HPE. Top-down approaches, e.g. [12, 33], estimate a pose for each bounding box provided by a detector; inaccurate or missing detections are one of their main failure modes. Detector-free (bottom-up, single-stage, and hybrid) [28, 37] methods generate poses directly from the image without relying on bounding boxes. Top-down methods perform better on datasets like COCO, however, they struggle in multi-body scenarios where detection errors lead to degraded performance, giving detector-free methods an advantage in crowded scenes, e.g. in Fig. 1.

Bounding boxes, masks, and poses represent the human body in different ways and are often trained on different datasets. Bounding boxes are easy to annotate and effective for detecting small people in large scenes, but they lack detail and may merge instances in crowded scenes. Pose estimation models provide anatomical detail but are less effec-

tive in detecting instances, which is addressed in top-down methods by an external detector.

BUTCD [37] uses the concept of *conditioning*. Top-down methods estimate poses on image crops defined by bounding boxes, producing one skeleton per crop centered on the bounding box. Thus, top-down methods are conditioned by bounding boxes. Detector-free methods are conditioned only by an image, while pose-refining methods rely on prior pose estimates to iteratively refine their output.

The proposed BBox-Mask-Pose (BMP) method extends conditioning to masks and integrates bounding boxes, masks, and poses into a feedback loop (Fig. 1). BMP uses three specialized models that iteratively refine each other's output, allowing detection, segmentation, and pose estimation to achieve a consistent results and performance gains, especially in multi-body scenarios. Specifically, the models are:

- Enhanced RTMDet [18]: An adapted detector that ignores masked instances. It runs iteratively, avoiding duplicating detections and adding missed instances.
- MaskPose: a pose estimation model that is conditioned by instance segmentation masks rather than bounding boxes. Its pose estimation is more robust in dense scenes.
- SAM2 (Segment Anything Model) [21], conditioned by carefully selected pose keypoints, which enhances segmentation capabilities and facilitates information passing between bounding box locations and pose estimates.

BMP, with no dataset-specific parameter adjustment or fine-tuning, matches state-of-the-art (SOTA) results of detector-free approaches on the OCHuman dataset, while also achieving SOTA performance of top-down methods on the COCO dataset. Conditioning pose on masks, detection on prior detections, and segmentation on poses creates a cycle that improves the accuracy of all components. To keep BMP efficient, we used moderately-sized transformers (ViT-b [8], RTMDet-l) not specialized on multi-body scenarios. Larger models (ViT-h, RTMDet-x) would boost performance even more.

The BMP method combines an object detector with a model that understands the object structure and could generalize to tasks where specialized models interpret the structure, as HPE models do for human anatomy.

In summary, the main contributions are:

1. A detector that ignores masked instances, enabling iterative detection and retrieval of previously missed detections.
2. MaskPose, a pose estimation model conditioned by segmentation masks instead of bounding boxes, boosting performance in dense scenes without adding parameters.
3. BBox-Mask-Pose (BMP) method linking bounding boxes, segmentation masks, and poses to simultaneously address multi-body detection, segmentation and pose estimation.

## 2. Related work

The most widely used dataset for 2D human pose estimation is COCO [15], with MPII [3] being a less common alternative. Another dataset used for training is AIC [31]. Datasets like OCHuman [35] and CrowdPose [14] focus on multibody problems such as occlusion and self-occlusion. OCHuman is too small for large-scale training and is traditionally used only for evaluation. CrowdPose is big enough for training but is unsuitable for evaluation in multi-dataset setup as it mixes train and test sets of COCO, MPII and AIC. For COCO and related datasets, the evaluation metric is Object Keypoint Similarity (OKS), while Percentage of Correct Keypoints (PCKh) is used for MPII. In addition to pose estimation dataset, CrowdHuman [22] focuses on person detection in crowds.

There are two main approaches to 2D human pose estimation: top-down and detector-free. Detector-free can further be divided into single-stage [23, 24, 28, 32], bottom-up [6, 9, 20] and hybrid [37].

Top-down methods [12, 16, 25, 33, 34] use person detector to detect bounding boxes and estimate one skeleton for each bounding box. They leverage big progress in human detection and specialize on understanding of human structure. Top-down methods are the most successful in datasets like COCO, MPII or AIC but struggle on crowded datasets like OCHuman due to low-quality detections. Most notably, ViTPose [33] combines multiple datasets into one strong backbone and set a strong baseline, setting up state of the art performance on most datasets.

On the other hand, detector-free models do not achieve SOTA performance on COCO but are superior to top-down methods on OCHuman as they are specialized on decoupling close-interaction instances. The most successful model, BUCTD [37], conditions top-down pose estimation by previously estimated keypoints (from bottom-up methods). It is a pose-refinement method which has state-of-the-art results on OCHuman datasets due to its strong ability to decouple close-interaction people.

The latest direction in modelling of human body are foundational models [7, 10, 13, 30]. They try to learn general features describing human body that could be used for all human-related tasks like segmentation, pose estimation etc. Most notably, Sapiens 2b [13] was trained on staggering 2M images and with 2B parameters is almost four times bigger than ViTPose-h. Even with this size, foundational models perform comparatively or worse than much smaller specialized models.

Object (or person) detection is one of the most researched problems in computer vision. Huge models like InternImage [29] or Co-DETR [38] holds SOTA performance on multiple datasets. In our comparison, we use smaller almost real-time models RTMDet [18], ConvNeXt [17] and HRNet [25] which have slightly lower perfor-

mance but run much faster. To the best of our knowledge, conditioning object detection by (previously processed) segmentation masks was not researched. The only exception is PoseNMS [19], which is used to verify or reject bounding boxes.

The idea of segmentation conditioned by human pose is not new. Many models [1, 2, 27, 35, 36] estimate instance segmentation from either ground truth pose or estimated keypoints. Other methods like [5] use pose for test-time adaptation in instance segmentation. The latest segmentation foundational model SAM2 [21] is conditioned not only by human pose but by any point(s). Conditioning mask by pose is not new but the other direction (conditioning pose by mask) remains unsolved.

## 3. Method

The following sections detail the components of the BBox-Mask-Pose (BMP) method. To create an iterative process involving detection, segmentation, and pose estimation, each component must be conditioned by the others. We adapt the detector and pose estimation model for mask conditioning and use the Segment Anything Model 2 [21] to condition masks with bounding boxes and keypoints.

In BMP, the loop starts with the detector; in general, it could start from any of the three representations.

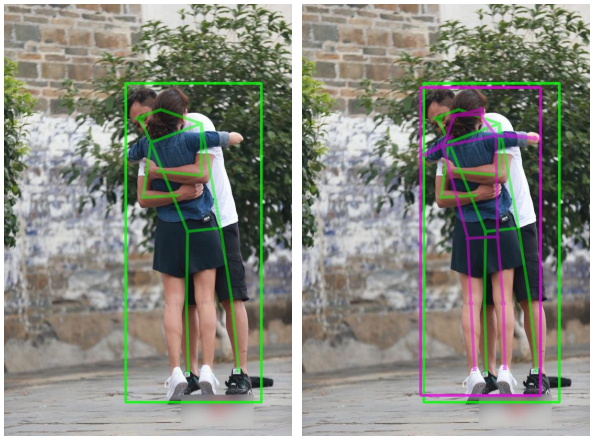
### 3.1. Detection

Conditioning the detector with a mask involves detecting new instances in an image while ignoring already processed instances. We condition the detector through foreground masking as shown in Fig. 1. We use *instance-removal* data augmentation to make the detector robust against masked foreground instances. During training, some instances in the image are masked out, and the model learns not to predict them.

A major challenge for the detector is handling background instances split into multiple disconnected parts by masking. The detector often detects these parts as separate instances instead of grouping them. Instance-removal augmentation simulates this divisive occlusion by randomly masking out patches in addition to selected instances. This approach improves detection, but does not fully resolve the problem. Examples of such failure cases are included in the supplementary material.

The detector retains its ability to detect instances in unmasked images. We suspect that instance-removal augmentation may reduce performance in low-light conditions, although this has not been verified. Practically, we can use the same detector model to initiate the iterative loop and in subsequent iterations.

Our detector (RTMDet [18]) estimates not only bounding boxes but also segmentation masks. We use these masks directly to guide pose estimation. If your detector of choice



(a) **Missed instance** which is detected in the second iteration of BMP. Left – RTMDet [18]+MaskPose, right – BMP.

(b) **Two instances in one detection** are resolved by refining segmentation masks with SAM [21] prompted by the detected pose. Left – RTMDet [18], right – BMP. Note that the detection of the woman is improved, but the right leg is still wrong.

(c) **Collapse of pose estimates into one** for two instances with correctly detected overlapping bboxes. Left – ViTPose-b conditioned by bounding box, right – MaskPose-b conditioned by masks.

Figure 2. The BBox-Mask-Pose resolves detection errors (top and middle) and pose errors (bottom). Quantitative results in Tab. 3.

outputs only bounding boxes or if you want to refine segmentation masks before estimating pose, you could refine or estimate masks by SAM before pose estimation.

The next chapter introduces MaskPose and explains how it is conditioned by estimated masks. We experimented with the same semi-transparent masking for the detector as used for MaskPose, but this impaired the detector's ability to distinguish foreground from background instances. When a segmentation mask is incorrect, we mask out that part of the image, resulting in lost information for the detector. Despite this instability, BBox-Mask-Pose improves upon the top-down baseline.

### 3.2. Pose Estimation

Traditional top-down methods rely solely on bounding boxes, cropping an image patch centered on the bounding box. If multiple people appear in the same crop, the model estimates the pose of the central person but often merges body parts from others into a single skeleton. We introduce MaskPose, which builds on ViTPose [33] and adapts it to use segmentation masks for conditioning.

ViTPose trained in multi-dataset setup generalizes well across datasets, leveraging the strength of the ViT [8] backbone. ViTPose use specialized head for each dataset with shared backbone. We also train MaskPose on the COCO, MPII, and AIC datasets, but MaskPose has a single head for all datasets. This head predicts all 21 keypoints defined across COCO, AIC, and MPII, resulting in negligible performance loss compared to using separate heads. MaskPose can thus be evaluated directly on any dataset without switching heads.

Like the detector, MaskPose is conditioned by masking parts of the image, but here we mask the background. MaskPose uses image crops with a darker background to emphasize the foreground instance. Unlike the detector, MaskPose uses semi-transparent masking, blending 20% of the masked-out image with 80% of the original. We experimented with different transparency ratios and observed no performance difference, as long as both components were present. Fully masking the background causes loss of contextual information, impairing MaskPose's recovery from inaccurate masks, while no masking reverts to a traditional bounding-box-based approach. Semi-transparent masking provides a balance, making MaskPose robust to mask errors by preserving background context.

To enhance robustness to inaccurate masks, we randomly deform ground truth and pseudo-ground truth masks during training, allowing the model to predict keypoints outside the mask.

MaskPose has approximately the same number of parameters as ViTPose, differing only in head architecture and preprocessing. These small changes enable MaskPose to perform similarly on standard datasets (COCO, AIC, MPII) while improving performance in multi-body scenarios. Mask conditioning adapts the top-down method for multi-body cases, allowing instance specification in densely overlapping scenes. MaskPose is highly robust to incorrect masks, proving to be the strongest component of the BMP loop.

### 3.3. Segmentation

We use Segment Anything Model v2 (SAM) [21] for mask generation, conditioned by estimated bounding boxes and poses. SAM is inherently a conditioned segmentor, so no adaptations are needed. The key challenge is SAM prompting.

Prompting SAM automatically is complex. It was trained with a maximum of 8 point prompts and fails with more, such as all 17 keypoints from COCO pose (Fig. 3a). The challenge is twofold: determining the number of keypoints and selecting them. This chapter outlines our prompting method for a successful BBox-Mask-Pose loop and analyzes hyper-parameter effects on the loop. For more on segmentation conditioned by pose with SAM, see the supplementary material.

**Visibility**. Detected keypoints present a visibility challenge. Pose estimation models estimate both visible and occluded keypoints but typically do not distinguish between them (with some exceptions, like [26]). SAM can handle occluded keypoints if they are on the instance border but struggles if they're within another instance. We approximate visibility using confidence, selecting only keypoints above a certain confidence threshold.

**Spread**. To segment disconnected parts of an instance (for example, the legs of the background player in Fig. 1), we maximize keypoints spread. Selecting keypoints along the bounding box border provides a good spread, but SAM still needs at least one unambiguous keypoint to specify the instance. We mimic human annotation by first choosing the most confident keypoint (analogous to a human's initial click in the center) and then selecting keypoints to maximize spread. To avoid redundancy, we select at most one facial keypoint (an eye or the nose).

**Bounding box**. Another question is whether to use bounding boxes (Fig. 3b). Experiments with ground truth boxes show that bounding boxes improve mask quality, but the situation changes with detected bounding boxes, especially in multi-body scenarios. The detector may only capture part of an instance or merge two instances. Prompting SAM with detected boxes restricts it to the detected area, limiting its ability to correct detection errors. Conversely, SAM without a bounding box can "explore" undetected areas but loses precision within the bounding box. Since detection correction is critical for BMP success, we prompt SAM without a detected bounding box. Prompting with bounding box would be useful for final mask refinement af-

(a) Different **number of keypoints** for prompting SAM. Prompting with too many points hinders performance.
Left – 6 keypoint prompts, right – 13 correct prompts.

(b) Prompting **with and without bounding box**. Prompting with bounding box prevents SAM from fixing body parts outside of bounding box.
Left – RTMDet, middle – with bbox, right – without bbox.

Figure 3. Influence of prompting parameters on SAM masks.

ter the BMP loop when bounding boxes are stable.

The keypoint selection algorithm is summarized in Alg. 1. It maximizes keypoint spread similar to KMeans++ initialization [4], factoring in keypoint confidence. We used 6 positive keypoints for each instance ($N_{max}$) and confidence threshold $T_c = 0.5$.

Our experiments suggest that automatically selected keypoints have a different distribution from human-annotated prompts. Human annotators intuitively understand the scene, and SAM generally performs better with human prompts than with automated keypoint selection. Pose keypoints tend to lie on the borders and extremes of the instance, whereas humans often click in the middle of the instance. By choosing visible, high-confidence, and spread keypoints, we partially simulate human prompting. Although automated prompts do not match human effectiveness, the BBox-Mask-Pose loop still improves segmentation, and pose-prompted SAM outperforms bounding-box-prompted SAM.

**Pose-Mask consistency**. When incorrect keypoints are selected during prompting, SAM's segmentation mask may be worse than the original detector mask. After mask generation, we measure the *pose-mask consistency* of both the original detector mask and the mask refined by SAM. Pose-mask consistency (*P-Mc*) is defined as:

**Algorithm 1:** Keypoints selection for SAM

**Inputs :** Set ot detected keypoints $K$,
Confidence threshold $T_c$,
Max number of keypoint $N_{max}$

**Output:** Set of selected keypoints $S$

```
Select keypoints from K with confidence ≥ T_c
Sort keypoints in K by confidence
S ← ∅
Select the most confident keypoint into S
while len(S) < N_max do
    k ← keypoint from K furthest to S
    Add k to S
end
return S
```

$$\text{P-Mc} = \frac{\sum k_p^+}{\sum k_p} + \frac{\sum k_n^-}{\sum k_n} \tag{1}$$

where $k_p$ represents the positive keypoints of the instance, and $k_n$ represents negative keypoints (those from other instances in the image). $k_p^+$ are positive keypoints inside the mask, while $k_p^-$ are negative keypoints outside the mask. Thus, pose-mask consistency measures the proportion of keypoints (both positive and negative) that are consistent with the mask. If the refined mask has a lower *P-Mc* than the previous mask, we discard it. BBox-Mask-Pose discard approximately 15% of SAM-refined masks.

Prompting with ground truth data behaves differently than with noisy estimated data. As mentioned, the ground truth bounding box consistently improves the predicted mask. Similarly, ground truth data includes annotated visibility, allowing us to use only visible keypoints. We prompted SAM with ground truth bounding boxes and poses when generating pseudo ground truth for AIC and MPII to train MaskPose. For further details on prompting with ground truth, see the supplementary material.

### 3.4. Closing the circle

With all three models adapted for mutual conditioning, we establish a closed iterative loop. As shown in Fig. 1, the detector conditions MaskPose, which in turn conditions SAM2 segmentation. The loop completes by masking out processed instances and rerunning the detector.

The loop is not infinite. Each BBox-Mask-Pose iteration masks out more of the image, and when all instances are masked, the detector no longer identifies new instances, ending the loop. In practice, the user can set the number

of iterations manually, as later iterations yield diminishing performance gains but are computationally expensive.

To minimize duplicate detections, we use two forms of non-maximum suppression (NMS): bounding box NMS in the detector and pose NMS in the pose estimator. We apply both with standard settings. Bounding box NMS with intersection-over-union (IoU) at 0.3 and pose NMS with object-keypoint-similarity (OKS) at 0.9. If valid detections are mistakenly suppressed, they are re-detected in the next BMP loop iteration.

# 4. Results

## 4.1. Implementation details

The BMP loop uses RTMDet-l [18] as the detector. We fine-tuned RTMDet with instance-removal augmentation for 10 epochs on COCO, covering all 80 classes, to enable it to ignore already-processed instances. The same detector was used in top-down model experiments for fair comparison.

MaskPose builds on ViTPose [33], so we use the same training setup: 210 epochs with three learning rate steps. We employ a multi-dataset training strategy across COCO, AIC, and MPII. Since MPII and AIC lack ground truth segmentation, we generate pseudo ground truth using SAM2, prompted with ground truth bounding boxes and annotated visible keypoints.

The Segment Anything Model (SAM) is used without fine-tuning to prevent catastrophic forgetting, as its training set is unavailable. We use version *sam2-hiera-base+* with post-processing settings: max_hole_area at 10 and max_sprinkle_area at 50. Each instance is processed independently, which yields slightly better results than batch processing.

## 4.2. Comparison with SOTA

**Pose estimation**. Tab. 1 compares pose estimation performance on the OCHuman and COCO datasets. All top-down methods except MIPNet are evaluated with the same detections from RTMDet-l [18] for a fair comparison. MaskPose improves the ViTPose [33] baseline from 42.6 to 45.0 AP through mask conditioning. BMP 1× and MaskPose yield identical results, as BMP 1× essentially runs MaskPose with an additional mask refinement step, which does not affect pose. BBox-Mask-Pose 2× further increases MaskPose performance from 45.0 to 48.2 AP through iterative conditioning between masks and poses. BMP matches the state-of-the-art performance of BUTCD [37] on OCHuman and ViTPose [33] in COCO, combining the strengths of the top-down and hybrid approaches. BMP and MaskPose perform similarly on COCO, as the detector captures nearly all instances in the first pass, with only a few additional detections in the second iteration.

Additionally, BUCTD could be integrated into the BBox-Mask-Pose loop. BUCTD conditions pose estimation on (bottom-up) poses, while BMP conditions pose on masks. BUCTD could either refine MaskPose's keypoints or replace MaskPose within the BMP loop. Furthermore, BMP provides not only pose estimates but also bounding boxes and segmentation masks.

Although BMP could run until no new detections occur in the *detect* phase, we find that performance plateaus after two iterations, similar to BUCTD. Further iterations add computational cost without notable performance gains.

**Detection and segmentation**. Tab. 2 compares BMP detection and segmentation with previous work on the OCHuman dataset. BMP 1× improves the RTMDet pipeline by refining bounding boxes and segmentation masks using pose-prompted SAM, as illustrated in Fig. 1. BMP

| Model | OCHuman val AP | OCHuman test AP | COCO val AP |
|---|---|---|---|
| DEKR [9] | 37.9 | 36.5 | 71.0 |
| CID-w48 [28] | 46.1 | 45.0 | 69.8 |
| BUCTD [37] | 48.3 | 47.4 | 74.8 |
| Sapiens 0.3b [13] | 42.0 | 41.3 | 66.1 |
| MIPNet† [12] | 42.0 | 42.5 | 76.3 |
| ViTPose-b [33] | 42.5 | 42.6 | <u>76.3</u> |
| **MaskPose-b** | 45.3 | 45.0 | **76.4** |
| BUCTD 2× [37] | **48.8** | **48.3** | —‡ |
| **BBox-Mask-Pose** 1× | 45.3 | 45.0 | **76.4** |
| **BBox-Mask-Pose** 2× | <u>48.6</u> | <u>48.2</u> | **76.4** |

Table 1. **Pose estimation – comparison** with prior art. Best results are in bold, second best underlined. The top part of the table shows detection-free, middle top-down and the bottom iterative methods. Top-down methods are compared using detections from RTMDet-l [18] apart from MIPNet† which reports results from [12]. MaskPose improves ViTPose and BMP further improves MaskPose, matching SOTA performance.
‡ Result for BUCTD 2× on COCO was not reported in [37].

| Model | OCHuman test bbox AP | OCHuman test mask AP |
|---|---|---|
| ConvNeXt [17] | 29.4 | 20.4 |
| HRNet [25] | 27.1 | 19.4 |
| RTMDet-l [18] | <u>30.0</u> | 26.5 |
| **BBox-Mask-Pose** 1× | <u>30.0</u> | <u>31.1</u> |
| **BBox-Mask-Pose** 2× | **31.3** | **32.4** |

Table 2. **Detection and instance segmentation – comparison** with prior art. BMP 1× improves the RTMDet baseline through SAM prompted by estimated keypoints. BMP 2× further improves detection by detecting background instances in masked-out images.

| bbox AP @ max_IoU | 0.0 – 0.2 | 0.2 – 0.4 | 0.4 – 0.6 | 0.6 – 0.8 | 0.8 – 1.0 | 0.0 – 1.0 |
|---|---|---|---|---|---|---|
| RTMDet-l | 22.8 | 0.3 | 46.8 | 52.8 | 49.0 | 31.1 |
| BBox-Mask-Pose 2× | 22.8 (+0.0) | 0.4 (+0.1) | 46.0 (-0.8) | 52.7 (-0.1) | 52.9 (+3.9) | 32.1 (+1.0) |

Table 3. **Ablation study of detection** by maximum IoU on OCHuman-val. BBox-Mask-Pose improves detection especially in multi-body scenarios where bbox overlap is huge. Traditional detectors like RTMDet often merge two individuals into one instance or ignore the background individual. BMP resolves the issues with instance understanding through pose estimation. See e.g. the detection errors in Fig. 2.

| pose | SAM | pose | loops | bbox | pose | params |
|---|---|---|---|---|---|---|
| ✓ | ✓ | ✗ | 1× | 31.1 | 45.3 | 225 M |
| ✓ | ✓ | ✗ | 2× | 32.1 | 48.6 | 369 M |
| ✓ | ✓ | ✓ | 1× | 31.1 | 46.4 | 312 M |
| ✗ | ✓ | ✗ | 2× | 31.9 | 47.3 | 282 M |
| ✓ | ✗ | ✗ | 2× | 30.8 | 47.0 | 201 M |

Table 4. **Ablation study** of BBox-Mask-Pose components evaluated on OCHuman-val. Bbox and pose evaluated with AP. The sum of trainable parameters approximates computational complexity. First row corresponds to BMP 1×, second to BMP 2×.

2× further improves detection and segmentation through re-detection of background instances in images with masked-out instances, as shown in Fig. 2.

### 4.3. Ablation study

**Detection accuracy in multi-body scenarios**. To analyze BMP's impact on bounding box quality, we assess detection performance on the OCHuman validation split by Max_IoU. For each instance, we calculate its highest IoU with other instances and split the dataset accordingly. Tab. 3 shows that detection is largely unaffected across all Max_IoU values except for the highest. BMP improves detection accuracy in high overlap scenarios by capturing missed instances or splitting merged instances, as illustrated in Fig. 2.

**Looping SAM and pose estimation**. The third row of Tab. 4 shows a slight improvement in pose estimation when re-running pose on SAM-refined masks. This pipeline, detect-pose-SAM-pose, is comparable to one BMP iteration as it cannot re-detect previously missed instances. SAM mask refinement improves MaskPose keypoint predictions, suggesting that an SAM-pose-SAM loop could further enhance the results. However, the additional computational cost outweighs the gains, so we exclude it to keep BMP efficient.

**Prompting SAM only with bounding box**. This approach effectively omits the pose estimation model from the loop, as SAM is prompted solely by the bounding box detected in the first step. SAM refines the segmentation mask and updates the bounding box accordingly. Tab. 4 shows that SAM alone improves performance over omitting SAM entirely (second-last and last rows). Adding keypoints as prompts further boosts detection from 31.9 to 32.1 AP and pose estimation from 47.3 to 48.6 AP.

**Omitting SAM**. When SAM is omitted from BMP, segmentation masks are provided only by the detector. This causes the detector to loop with itself without conditioning from masks or poses, often resulting in un-segmented body parts, like missed limbs. For example, in Fig. 1, un-segmented legs of a background player could be detected as separate instances, as shown in Fig. 4. In practice, omitting SAM resembles running a detector with a low non-maxima suppression (NMS) threshold, resulting in many false-positive bounding boxes. This hinders detection performance, but slightly boosts pose accuracy. Low-confidence poses minimally impact the COCO evaluation, as they do not deform the precision-recall curve in the AP computation. That explain why looping the detector with itself still improves the pose. However, using SAM improves detection from 30.8 to 32.1 AP and pose estimation from 47.0 to 48.6 AP, as shown in Tab. 4.

**Computational complexity estimation**. Comparing the complexity of iterative approaches with previous work is challenging. Runtime per frame favors optimized code, which is prioritized in industry but less so in research. Here, we approximate computational complexity by summing trainable parameters that each image passes through. For example, combining the detector (RTMDet-l) with 57M parameters and the pose model (ViTPose-b) with 87M parameters results in 144M trainable parameters. This method was used to compute the values in Tab. 4, with 57M for RTMDet-l [18], 87M for MaskPose-b (ViTPose-b [33]), and 81M for SAM (sam2-hiera-base+ [21]).

While summing parameters provides an estimate, it does not account for the difference between models that process entire images and those that work per instance. To adjust for this, one could multiply by the average instance count per image, but we omit this detail for simplicity, as we only use it for ablation and comparisons with other top-down models.

Omitting SAM from the loop reduces parameters significantly, but also sharply decreases performance. Running the pose estimation again after SAM mask refinement increases parameter usage by 40%, from 225M to 312M.

For comparison, the Sapiens 0.3b model, a recent foundational model with 336M parameters [13], combined with

RTMDet-l totals 393M parameters, surpassing the 369M of two BMP iterations. BMP outperforms Sapiens 0.3b on both COCO and OCHuman datasets.

## 5. Conclusions

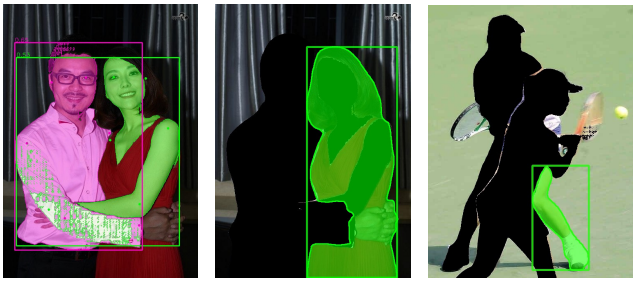
We presented BBox-Mask-Pose (BMP), a method for detection, segmentation, and pose estimation in multi-body scenarios. Unlike prior approaches, BMP conditions pose estimation on predicted instance masks through MaskPose. BMP integrates MaskPose and uses the Segment Anything Model (SAM) as a conditioned segmentation model, forming a self-improving loop integrating detection, segmentation, and pose estimation. By conditioning each model on outputs from the others, BMP improves detection, segmentation, and pose estimation simultaneously on the OCHuman dataset. BMP achieves state-of-the-art (SOTA) performance on both the OCHuman and COCO datasets.

Key findings:

1. Conditioning the top-down pose model with masks instead of bounding boxes improves performance, especially in crowded scenes.
2. The BMP method demonstrates that explicit mutual conditioning between the detector, segmentator, and pose estimation models enhances performance. However, adapting these models for mutual conditioning is non-trivial.
3. BMP's effectiveness diminishes after two iterations, with additional iterations offering little performance gain while increasing computational cost.
4. BMP matches the SOTA performance of detector-free approaches on OCHuman while also matching the SOTA performance of top-down models on COCO.
5. Surprisingly, the Segment Anything Model proved the least effective component in BMP. Although pose-based prompting improves SAM's masks, automated prompting falls short compared to human interaction.
6. The modular structure of BMP enables further performance gains by integrating improved models or adding BUCTD [37] to the loop.

**Limitations** of BMP primarily involve imperfect SAM mask refinement. When SAM is prompted with inaccurate keypoints (e.g., occluded or mislocalized), it has limited recovery ability, which can lead to masking out the wrong instances, preventing the detector from retrieving them. We experimented with semi-transparent masking, as used in MaskPose, but found it ineffective for this issue.

A second limitation occurs when detecting in masked-out images. If a foreground instance divides a background instance into disconnected parts, the detector often fails to connect these, generating multiple small bounding boxes for each segment. Although pose NMS suppresses redundant detections, disconnected body parts remain separate.



(a) Segmenting only skin (green)

(b) Re-detection of unsegmented clothes.

(c) Missed limb is re-detected

Figure 4. Characteristic errors in the BMP loop. The weakest part is SAM and its prompting with correct keypoints.

Attempts to use data augmentation to improve detector robustness in such cases were unsuccessful. Examples of these errors are included in Fig. 4.

**Future work**. MaskPose has demonstrated robustness to incorrect masks due to training augmentations; extending this robustness to the detector and pose-to-seg models could significantly enhance BMP performance.

Beyond robustness, improving the efficiency of interactions between bounding boxes, masks, and poses is an area for exploration. Foundational models aim to unify body representations at a feature level but lack the explicit constraints offered by different representations. Although foundational models are non-iterative, their large size often results in longer inference times compared to smaller, specialized models. Our findings indicate that explicit constraints within specialized models could improve performance while keeping the models smaller and faster.

## References


[1] Niaz Ahmad, Jawad Khan, Jeremy Yuhyun Kim, and Youngmoon Lee. Multiposeseg: Feedback knowledge transfer for multi-person pose estimation and instance segmentation. *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2086–2092, 2022. 3

[2] Niaz Ahmad, Jawad Khan, Jeremy Yuhyun Kim, and Youngmoon Lee. Joint human pose estimation and instance segmentation with poseplusseg. In *AAAI Conference on Artificial Intelligence*, 2022. 3

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

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006. 5

[5] Kambiz Azarian, Debasmit Das, Hyojin Park, and Fatih Murat Porikli. Test-time adaptation vs. training-time generalization: A case study in human instance segmentation using keypoints estimation. *2023 IEEE/CVF Winter Conference*

*on Applications of Computer Vision Workshops (WACVW)*, pages 411–420, 2022. 3

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2

[7] Yuanzheng Ci, Yizhou Wang, Meilin Chen, Shixiang Tang, Lei Bai, Feng Zhu, Rui Zhao, Fengwei Yu, Donglian Qi, and Wanli Ouyang. Unihcp: A unified model for human-centric perceptions. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17840–17852, 2023. 2

[8] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4

[9] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14676–14686, 2021. 2, 6

[10] Seyed Abolfazl Ghasemzadeh, Gabriel Van Zandycke, Maxime Istasse, Niels Sayez, Amirafshar Moshtaghpour, and Christophe De Vleeschouwer. Deepsportlab: a unified framework for ball detection, player instance segmentation and pose estimation in team sports scenes. *ArXiv*, abs/2112.00627, 2021. 2

[11] Kerui Gu, Rongyu Chen, and Angela Yao. On the calibration of human pose estimation. *arXiv preprint arXiv:2311.17105*, 2023. 13

[12] Rawal Khirodkar, Visesh Chari, Amit Agrawal, and Ambrish Tyagi. Multi-instance pose networks: Rethinking top-down pose estimation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3102–3111, 2021. 1, 2, 6

[13] Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Zhaoen Su, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. In *European Conference on Computer Vision*, 2024. 2, 6, 7

[14] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. *arXiv preprint arXiv:1812.00324*, 2018. 2

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 2

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[17] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6

[18] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *ArXiv*, abs/2212.07784, 2022. 2, 3, 6, 7, 11, 12, 13, 14, 16

[19] George Papandreou, Tyler Lixuan Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Christoph Bregler, and Kevin P. Murphy. Towards accurate multi-person pose estimation in the wild. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3711–3719, 2017. 3

[20] George Papandreou, Tyler Lixuan Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin P. Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *European Conference on Computer Vision*, 2018. 2

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 3, 4, 7, 11, 12, 13, 14, 15

[22] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *ArXiv*, abs/1805.00123, 2018. 2

[23] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11069–11078, 2022. 2

[24] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-end trainable multi-instance pose estimation with transformers. *arXiv preprint arXiv:2103.12115*, 2021. 2

[25] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2, 6

[26] Pengzhan Sun, Kerui Gu, Yunsong Wang, Linlin Yang, and Angela Yao. Rethinking visibility in human pose estimation: Occluded pose reasoning via transformers. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5891–5900, 2024. 4

[27] Subarna Tripathi, Maxwell D. Collins, Matthew A. Brown, and Serge J. Belongie. Pose2instance: Harnessing keypoints for person instance segmentation. *ArXiv*, abs/1704.01152, 2017. 3

[28] Dongkai Wang and Shiliang Zhang. Contextual instance decoupling for robust multi-person pose estimation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11050–11058, 2022. 1, 2, 6

[29] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14408–14419, 2023. 2

[30] Yizhou Wang, Yixuan Wu, Shixiang Tang, Weizhen He, Xun Guo, Feng Zhu, Lei Bai, Rui Zhao, Jian Wu, Tong He, and

Wanli Ouyang. Hulk: A universal knowledge translator for human-centric tasks. *ArXiv*, abs/2312.01697, 2023. 2

[31] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, Yizhou Wang, and Yonggang Wang. Ai challenger : A large-scale dataset for going deeper in image understanding. *ArXiv*, abs/1711.06475, 2017. 2

[32] Yabo Xiao, Xiaojuan Wang, Dongdong Yu, Kai Su, Lei Jin, Mei Song, Shuicheng Yan, and Jian Zhao. Adaptivepose++: A powerful single-stage network for multi-person pose regression. *arXiv preprint arXiv:2210.04014*, 2022. 2

[33] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 4, 6, 7, 14

[34] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *Advances in neural information processing systems*, 34:7281–7293, 2021. 2

[35] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 889–898, 2019. 2, 3

[36] Desen Zhou and Qian He. Poseg: Pose-aware refinement network for human instance segmentation. *IEEE Access*, 8: 15007–15016, 2020. 3

[37] Mu Zhou, Lucas Stoffl, Mackenzie W. Mathis, and Alexander Mathis. Rethinking pose estimation in crowds: overcoming the detection information bottleneck and ambiguity. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14643–14653, 2023. 1, 2, 6, 8

[38] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 2

# Detection, Pose Estimation and Segmentation for Multiple Bodies: Closing the Virtuous Circle

Supplementary Material

## A. Prompting SAM ablation study

### A.1. Setup

Here, we describe the ablation study on prompting SAM. The study evaluates three metrics: detection improvement (bounding box; bbox), segmentation improvement (segm), and pose improvement (pose). For all experiments, we use bounding boxes and segmentation masks from RTMDet-l and pose estimates from MaskPose as the baseline pipeline. The experimental pipeline remains consistent throughout.

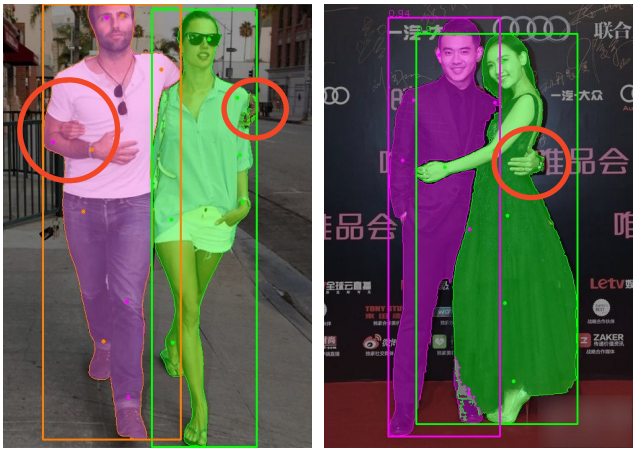
Detection and segmentation changes are evaluated on bounding boxes and segmentation masks refined by SAM, following the det-pose-SAM pipeline. Pose estimation is assessed by re-running MaskPose on refined masks, forming a det-pose-SAM-pose pipeline, similar to the setup in Tab. 4.

All experiments use *RTMDet-l* [18] as the detector, *MaskPose-b* as the pose estimator, and *sam2-hiera-base+* as the SAM2 [21] model. Each experiment is assigned a specific name, listed in the leftmost column of the tables, for clear referencing. When experiments appear in multiple tables for comparison, their names remain consistent for easier cross-referencing. Each result is highlighted in green or red depending on whether it improves or hinders performance compared to the RTMDet+MaskPose baseline.

**Detection vs. segmentation**. Before analyzing the results of the ablation study, we address a counterintuitive observation. When refining masks on OCHuman, segmentation and detection often conflict; improvement in one can lead to a decrease in the other. This is due to the focus on people with high overlap in the OCHuman dataset. Many examples consist of a large area representing the main body and smaller, disconnected body parts. Examples are shown in Fig. 5.

When mask refinement focuses heavily on the main segment, segmentation scores improve, as missing disconnected parts has little impact on mask IoU. Conversely, overly general prompting can cause SAM to merge both instances into one mask, creating a bounding box that may be more accurate than the original. Large masks merge instances, while small masks often miss disconnected body parts.

We prioritize detection, even though the goal is to improve all three metrics. The mask refinement step in BBox-Mask-Pose must ensure that segmented masks adequately remove limbs during the mask-out step, as shown in Figs. 4c and 9. However, excessively large masks prevent decoupling of merged instances, as seen in Fig. 2b. Thus, our aim is to improve detection without significantly hindering segmentation performance.



(a) (b)

Figure 5. Segmentation error involving a small number of pixels, like the circled hands, may have a large impact on detection accuracy measured by bounding box IoU. A detector returning correct bounding boxes, which would be nearly identical for both persons especially in (a), can make segmentation of the two people very challenging. Improving detection may thus lead to decrease in segmentation performance. Keypoints used for SAM prompting are marked (best viewed in zoom).

### A.2. Results

**Bounding box**. The question of whether to prompt SAM with a bounding box is addressed in Tab. 5, with examples provided in Fig. 3b. When the bounding box is accurate, or nearly so, it significantly improves segmentation quality. However, when the bounding box is incorrect, such as missing parts of an occluded person (Fig. 4c), prompting restricts mask refinement to the given bounding box, reducing the chance of recovery.

In the final version of BBox-MaskPose, we do not use bounding box prompting, as we prioritize SAM's ability to explore and detect previously missed body parts (Fig. 11). However, when bounding boxes are reliable, prompting with them can further refine segmentation and pose estimation, yielding improved results, as shown in Tab. 4 in Sec. 4.3. Bounding box prompting is also advantageous when ground truth bounding boxes are available.

**Number of positive keypoints** (⊕). Tab. 5 evaluates the effect of using different numbers of keypoints for prompt-

| name | batch | bbox | ⊕ | ⊖ | bbox | segm | pose |
|---|---|---|---|---|---|---|---|
| RTMDet [18] + MaskPose | | | | | 31.1 | 27.1 | 45.3 |
| A1 | ✗ | ✓ | 0 | 0 | 27.5 | **31.6** | 44.2 |
| A2 | ✗ | ✓ | 2 | 0 | 28.5 | **31.6** | **44.3** |
| A3 | ✗ | ✓ | 4 | 0 | 29.3 | 30.9 | 44.0 |
| A4 | ✗ | ✓ | 6 | 0 | 30.4 | 29.0 | 43.6 |
| A5 | ✗ | ✓ | 8 | 0 | **31.4** | 26.9 | 43.5 |
| B1 | ✗ | ✗ | 1 | 0 | 2.5 | 2.8 | 12.6 |
| B2 | ✗ | ✗ | 2 | 0 | 20.5 | 20.6 | 39.8 |
| B3 | ✗ | ✗ | 4 | 0 | 31.6 | **29.1** | **43.5** |
| B4 | ✗ | ✗ | 6 | 0 | 32.2 | 27.3 | 42.7 |
| B5 | ✗ | ✗ | 8 | 0 | **32.5** | 26.0 | 42.1 |
| B6 | ✗ | ✗ | 10 | 0 | 32.2 | 24.2 | 41.4 |

Table 5. Ablation study on prompting SAM [21] with varying positive keypoints (⊕) on OCHuman-val. Best results for each metric highlighted in **bold**; best method for BMP highlighted in blue. Green text indicates improvement over the baseline, red text indicates a decline. Detection and segmentation often conflict (Fig. 5). More keypoints improve segmentation (including incorrect masks) and bounding box detection, but increase segmentation errors. Pose remains stable but suffers from both wrong segmentation (guidance errors) and wrong detection (crop errors).

| name | batch | bbox | ⊕ | ⊖ | bbox | segm | pose |
|---|---|---|---|---|---|---|---|
| RTMDet [18] + MaskPose | | | | | 31.1 | 27.1 | 45.3 |
| A3 | ✗ | ✓ | 4 | 0 | 29.3 | **30.9** | 44.0 |
| C1 | ✗ | ✓ | 4 | 1 | 29.5 | 30.5 | 44.3 |
| C2 | ✗ | ✓ | 4 | 3 | **29.8** | 28.2 | **44.2** |
| C3 | ✓ | ✓ | 4 | – | 29.3 | **30.9** | 44.0 |
| B4 | ✗ | ✗ | 6 | 0 | **32.2** | **27.3** | 42.7 |
| C4 | ✗ | ✗ | 6 | 1 | 29.9 | 23.8 | 43.6 |
| C5 | ✗ | ✗ | 6 | 3 | 27.5 | 19.2 | **44.1** |
| C6 | ✓ | ✗ | 6 | – | **32.2** | **27.3** | 42.7 |

Table 6. Ablation study on prompting SAM [21] with varying negative keypoints (⊖) on OCHuman-val. Best results for each metric in **bold**; best method for BMP highlighted in blue. Green text indicates improvement over the baseline, red text indicates a decline. Adding negative keypoints to bounding boxes hinders segmentation but slightly improves detection. Without bounding boxes, negative keypoints degrade both detection and segmentation. Processing all image instances simultaneously (batch) gives the same or worse results.

ing.

In the top section, which includes bounding box prompts, using more keypoints increases the likelihood of confusing the model, leading to a drop in segmentation quality. However, more keypoints also increase the chance of expanding the mask beyond the bounding box, which improves detection. In particular, using 8 keypoints as positive prompts slightly outperforms the original baseline in detection.

The second section, without bounding box prompts, highlights that too few keypoints fail to define the instance adequately, causing both detection and segmentation to fail catastrophically. The best segmentation results occur with 4 keypoints, while detection performs best with 8. We chose 6 keypoints as a middle ground, balancing strong detection performance with slightly improved segmentation.

**Number of negative keypoints** (⊖). SAM2 provides two methods for negative prompting: explicit negative prompts and batch processing of all instances in the image. For explicit negative prompts, we identify the closest keypoint from other instances in the same image, provided it has confidence above a specified threshold.

Tab. 6 evaluates the impact of negative keypoint prompts. The top section examines adding negative prompts to 4 positive prompts and a bounding box. Negative prompts slightly improve detection quality, but significantly reduce segmentation quality. Given the trade-off, the decrease in segmentation outweighs the minor improvement in detection, so we avoid using negative keypoints in this setup.

The bottom section evaluates the effect of negative prompts without a bounding box prompting. Here, adding negative keypoints decreases both detection and segmentation performance, making it ineffective for this configuration.

**Batch processing**. Tab. 6 also evaluates the impact of batch processing, where SAM is prompted with multiple instances simultaneously. In this approach, SAM outputs non-overlapping masks for each prompted instance, ensuring that no mask is a subset of another. Although this behavior is logical, batch processing consistently produced the same or slightly lower results compared to single-instance processing in all our experiments.

We chose to stick with single-instance processing, as it likely allows the model to optimize better for one instance at a time, even if the resulting masks may overlap. Overlaps could be resolved in a post-processing step using pose information.

**Confidence threshold** ($T_c$). The top part of Tab. 7 examines the effect of varying the confidence threshold $T_c$ for selecting keypoints as prompts. Lower thresholds select keypoints with greater variability but increase the risk of using incorrectly estimated keypoints. The best results are achieved with a threshold of $T_c = 0.3$, which aligns with its common use in heatmap-based pose estimation models.

Interestingly, a lower threshold ($T_c = 0.1$) outperforms a higher threshold ($T_c = 0.8$), suggesting that variability is more important than strictly ensuring keypoint correctness. This may indicate that SAM is either robust to incor-

| name | batch | bbox | $\oplus$ | $\ominus$ | $T_c$ | sel. | ext. bbox | P-Mc | bbox by IoU | bbox | segm | pose |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTMDet [18] + MaskPose | | | | | | | | | | 31.1 | 27.1 | 45.3 |
| Confidence threshold $T_c$ | | | | | | | | | | | | |
| D1 | ✗ | ✗ | 6 | 0 | 0.8 | c+d | — | ✗ | ✗ | 29.9 | 27.2 | 42.1 |
| B4 | ✗ | ✗ | 6 | 0 | 0.5 | c+d | — | ✗ | ✗ | 32.2 | 27.3 | 42.7 |
| D2 | ✗ | ✗ | 6 | 0 | 0.4 | c+d | — | ✗ | ✗ | 32.4 | 27.6 | 43.1 |
| D3 | ✗ | ✗ | 6 | 0 | 0.3 | c+d | — | ✗ | ✗ | **32.7** | 27.9 | 43.3 |
| D4 | ✗ | ✗ | 6 | 0 | 0.2 | c+d | — | ✗ | ✗ | 32.5 | **28.3** | **43.6** |
| D5 | ✗ | ✗ | 6 | 0 | 0.1 | c+d | — | ✗ | ✗ | 32.5 | 28.2 | **43.6** |
| Selection method | | | | | | | | | | | | |
| D3 | ✗ | ✗ | 6 | 0 | 0.3 | c+d | — | ✗ | ✗ | **32.7** | **27.9** | 43.3 |
| E1 | ✗ | ✗ | 6 | 0 | 0.3 | c | — | ✗ | ✗ | 29.7 | 26.2 | **45.0** |
| E2 | ✗ | ✗ | 6 | 0 | 0.3 | d | — | ✗ | ✗ | 34.6 | 20.6 | 36.8 |
| Extended bounding box | | | | | | | | | | | | |
| F1 | ✗ | ✓ | 4 | 0 | 0.3 | c+d | ✗ | ✗ | ✗ | 29.3 | **31.1** | 44.1 |
| F2 | ✗ | ✓ | 4 | 0 | 0.3 | c+d | ✓ | ✗ | ✗ | 29.7 | 31.0 | 44.1 |
| Pose-Mask consistency | | | | | | | | | | | | |
| D3 | ✗ | ✗ | 6 | 0 | 0.3 | c+d | — | ✗ | ✗ | **32.7** | 27.9 | 43.3 |
| G1 | ✗ | ✗ | 6 | 0 | 0.3 | c+d | — | ✓ | ✗ | 30.9 | **31.1** | **45.0** |
| Bounding box by max_IoU | | | | | | | | | | | | |
| D3 | ✗ | ✗ | 6 | 0 | 0.3 | c+d | — | ✗ | ✗ | **32.7** | 27.9 | 43.3 |
| F1 | ✗ | ✓ | 4 | 0 | 0.3 | c+d | ✗ | ✗ | ✗ | 29.3 | **31.1** | **44.1** |
| H1 | ✗ | ✗/✓ | 6/4 | 0 | 0.3 | c+d | ✗ | ✗ | ✓ | 29.7 | 30.1 | 43.9 |
| Final methods | | | | | | | | | | | | |
| D3 | ✗ | ✗ | 6 | 0 | 0.3 | c+d | — | ✗ | ✗ | **32.7** | 27.9 | 43.3 |
| J1 | ✗ | ✗/✓ | 6/4 | 0 | 0.5 | c+d | ✓ | ✓ | ✓ | 29.2 | 31.1 | 46.3 |

Table 7. Ablation study on prompting SAM [21] with varying confidence thresholds ($T_c$), keypoint selection methods (sel.), and additional techniques on OCHuman-val. Best results for each metric in **bold**; best method for BMP highlighted in blue. Green text indicates improvement over the baseline, red text indicates a decline. Final methods used in BBox-Mask-Pose are highlighted in green. Two different methods used: one for the BMP loop, another for mask and pose refinement.

rect prompts (which we find unlikely) or that confidence is not a reliable metric for evaluating keypoint accuracy. As human pose estimation models are often overconfident, using self-estimated OKS from [11] could likely yield better results than relying on confidence.

**Selection method** (sel.). We compare three methods for selecting keypoints as prompts. The first method, confidence-only (c), sorts keypoints by confidence and selects the top N most confident ones. The second, distance-only (d), selects the N keypoints farthest from the center of the bounding box. The third method, described in Sec. 3.3, combines confidence and distance (c+d).

The second part of Tab. 7 shows that combining confidence and distance (c+d) outperforms either approach alone, providing superior results.

**Extending bounding box**. Experiment F2 in Tab. 7 explores the idea of extending the bounding box when using it for prompting. If selected keypoints fall outside the bounding box, it is extended to include all prompt keypoints. This ensures that no positive prompt lies outside the bounding box.

The results show that extending the bounding box slightly improves the detection accuracy while maintaining segmentation and pose estimation performance when using the bounding box. This approach is not applicable when prompting without a bounding box.

**Pose-Mask consistency** (P-Mc). Experiment G1 in Tab. 7 evaluates the effect of Pose-Mask Consistency (P-Mc), as described in Sec. 3.3. P-Mc significantly improves segmentation and pose estimation, but reduces detection performance. As a result, it is highly effective for refining masks and poses when the bounding box is approximately correct but not suitable for use in the iterative BBox-Mask-Pose loop.

**Bounding box depending on max_IoU**. The last experiment (H1) involves prompting with a bounding box only

Figure 6. Multiple background instances may merge into a single mask when no bounding box is provided as a prompt. The yellow mask was refined and covers all spectators. Foreground instances are omitted in the left image for clarity.
Left – RTMDet [18], right – BMP.

for instances with $max\_IoU > 0.5$. The rationale is that bounding boxes are typically accurate for isolated instances, where bounding box prompting improves results. However, for highly overlapping instances, the bounding box is often inaccurate and degrades detection performance. The results of this experiment are in Tab. 7.

As expected, the results fall between always prompting with bounding boxes and never using them. While this approach significantly improves segmentation compared to prompting without bounding boxes, the improvement in detection over always prompting with bounding boxes is minor. A qualitative analysis reveals that this method is primarily beneficial for low-resolution background instances, such as spectators in sports images. Without bounding box prompting, SAM often segments the entire background, leading to inaccuracies. This phenomenon is not well captured in the evaluation, as background instances rarely have pose annotations and have limited detection and segmentation labels. An example is shown in Fig. 6.

### A.3. Summary

The ablation study on automated SAM prompting is extensive and may seem overwhelming. To provide a clear summary, the last rows of Tab. 7 present two prompting methods used in BBox-Mask-Pose (BMP).

**D3**: This method is used in the BMP loop to balance refined masks with improved detection. It primarily enhances detection accuracy while slightly improving segmentation. Although it does not achieve the best standalone results, it performs best when used within the closed BMP loop with re-detections.

**J1**: This method is designed to refine masks and poses to produce high-quality estimates. It is used, for instance, in BMP ablations (Sec. 4.3) to loop SAM and MaskPose without re-detection. It significantly improves segmentation and pose estimation but is not part of the reported BMP results. J1 could be applied after the BMP loop terminates to further refine masks and bounding boxes, but we avoided this because it introduces additional overhead by requiring extra SAM (and possibly MaskPose) iterations. While such micro-loops and adjustments could further improve the reported results, our focus is on maintaining clarity, showing that two simple loops are sufficient to improve detection, segmentation, and pose estimation.

(a) Two people in matching coats.

(b) Two boys in one pair of pants, wearing matching shirts.

(c) Two players with matching jerseys.

Figure 7. Instances not split even after mask refinement by SAM [21], typically due to similar or identical textures.

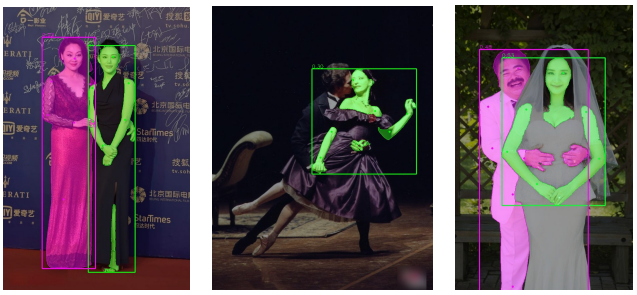
**Pose estimation robustness**. Pose estimation demonstrates notable robustness to the quality of estimated masks. MaskPose consistently produces accurate poses, even with low-quality masks (e.g., experiment C5 in Tab. 6), and almost always outperforms the ViTPose [33] baseline conditioned by the bounding box. However, achieving the MaskPose-SAM-MaskPose self-improving loop requires employing several hand-crafted tweaks. Among these, the Pose-Mask Consistency, as used in experiment J1 in Tab. 7, is particularly critical. Overall, BMP's pose estimation benefits more from refined detections and re-detection of background instances than from refining masks through SAM. This highlights the importance of robust detection to improve overall performance within the BMP framework.

## B. Failure cases analysis

Here, we provide a detailed analysis of BMP failure cases. While the most common issues are discussed in the paper, particularly in Sec. 5 and Fig. 4, this section offers additional examples and introduces a previously unmentioned type of error, instance merging.

**Merging instances**. Even though BMP is designed to decouple instances merged by the detector, and MaskPose performs well in such cases, SAM can mistakenly merge instances if it is incorrectly prompted or if the instances have similar textures. Prominent examples of these failures are shown in Fig. 7.

BMP struggles to address these issues because bounding box prompting would also fail, given that the detected bounding box already merges the instances. Furthermore, Pose-Mask Consistency (P-Mc) does not help in such cases, as only one instance is detected. Without negative key-



(a) (b) (c)

Figure 8. Oversegmentation. Green instances have incorrect masks – only the skin is segmented, excluding the clothes. This issue commonly occurs with clothing that exposes bare shoulders, such as dresses or jerseys. Keypoints used for SAM prompting are marked (best viewed in zoom).

points, a large mask that merges multiple instances (or even covers the entire image) would still achieve $P - Mc = 1.0$, since all positive keypoints fall within the mask and no negative keypoints are present to penalize the score.

**Segmenting clothes** instead of the whole person. This issue, illustrated in Fig. 8, is particularly common in OCHuman, where many individuals wear specific clothing. The problem frequently arises when a person has bare shoulders, such as in an evening dress or basketball jersey. In such cases, shoulder, facial, knee, elbow, and wrist keypoints, which are on the skin rather than clothing, prompt SAM to segment only the skin, leaving the clothing unsegmented. Hip and sometimes ankle keypoints could help refine segmentation, but these are typically low-confidence predictions and are often not selected.

Unsegmented clothing causes downstream issues as the masking-out step leaves the clothes visible. In subsequent BMPiterations, the detector identifies these as separate instances, as shown in Fig. 4.

We suggest two potential solutions. The first is to improve SAM prompting to include clothing in the segmentation. The bounding box prompt could address this specific case, but it hinders performance in other scenarios, as detailed in Fig. 3b and Appendix A. The second is to fine-tune the detector to ignore clothing when the skin is masked out. However, this approach risks reducing the detector's generalizability and causing overfitting to scenarios with visible skin and faces, which we believe is not a viable long-term solution.

**Missing body parts**. When SAM fails to segment a body part, it remains unmasked and may be redetected in the next stage, as shown in Figs. 4 and 9. This issue is even more pronounced when prompting with a bounding box, as detected bounding boxes often exclude disconnected limbs, leaving SAM unable to recover them. For this reason, we avoid prompting with the bounding box in the BMP loop.

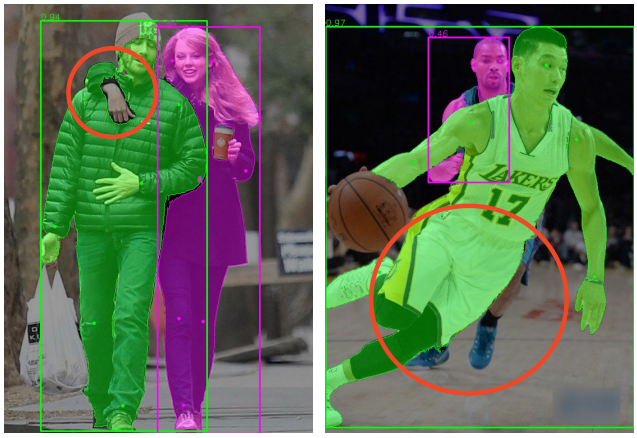


Figure 9. Images where SAM [21] successfully decoupled instances but failed to segment a disconnected body part. These parts remain unmasked and risk being re-detected, as illustrated in Fig. 4c. Keypoints used for SAM prompting are marked (best viewed in zoom).

Missed limbs could potentially be addressed by better alignment between pose and mask. If the refined mask is inconsistent with the prompted pose, SAM could be restarted with different prompts to minimize missed limbs. However, if the limb is also missed by MaskPose, BMP cannot resolve the issue.

**Correct examples**. BMP performs reliably in most cases, as demonstrated by the quantitative results. Figs. 10 and 11 showcase examples of successful detection and segmentation in challenging multi-body scenarios, including cases where a person is upside down.

In particular, Fig. 11 highlights the ability of BMP to balance segmentation and detection, as discussed in Fig. 5. The improvements are significant, with more precise segmentation and accurate instance counts in the scene. Some small body parts may occasionally be assigned to the wrong instance, but overall performance remains strong.

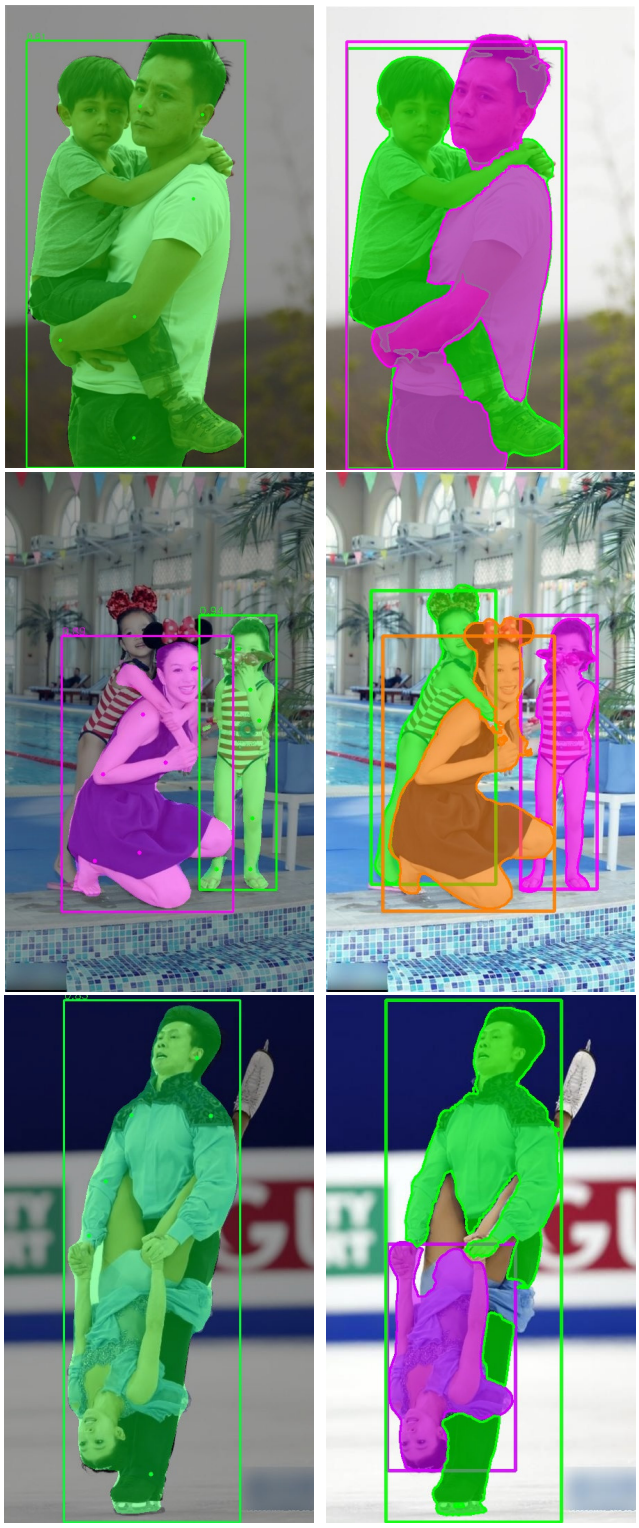


Figure 10. Two iterations of BMP successfully decouple merged instances, even in challenging images with upside-down people. Left – RTMDet [18], right – BMP.

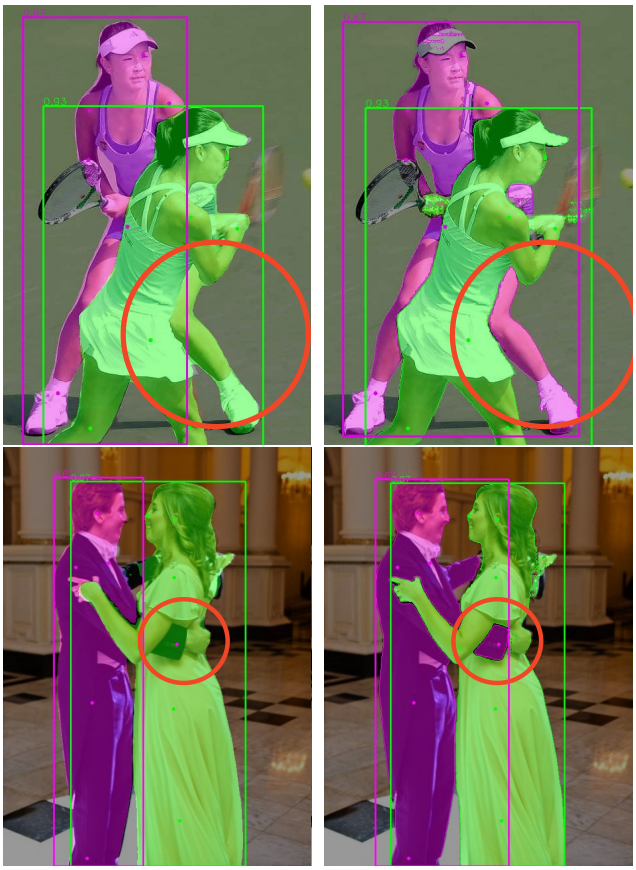


Figure 11. Images where BMP improves detection and segmentation using its pose estimates and SAM prompting with selected keypoint. Bounding box prompting did not lead to comparable results. Keypoints used for SAM prompting are marked (best viewed in zoom). Left – RTMDet [18], right – BMP.