# 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.

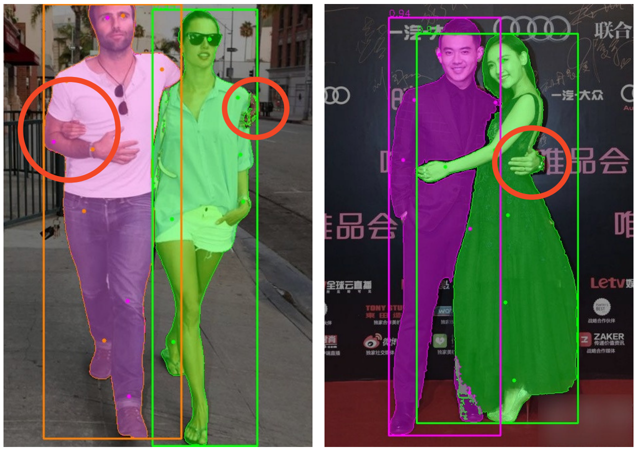
Detection and segmentation changes are evaluated on bounding boxes and segmentation masks refined by SAM, following the det-pose-SAM pipeline. Pose estimation is assessed by re-running MaskPose on refined masks, forming a det-pose-SAM-pose pipeline, similar to the setup in Tab. 4.

All experiments use *RTMDet-l* [22] as the detector, *MaskPose-b* as the pose estimator, and *sam2-hiera-base+* as the SAM2 [25] 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. 6, 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. 6 evaluates the effect of using different numbers of keypoints for prompting.

| name | batch | bbox | ⊕ | ⊖ | bbox | segm | pose |
|---|---|---|---|---|---|---|---|
| RTMDet [22] + 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 6. Ablation study on prompting SAM [25] 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 [22] + 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 7. Ablation study on prompting SAM [25] 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.

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. 7 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. 7 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. 8 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 incorrect prompts (which we find unlikely) or that confidence is not a reliable metric for evaluating keypoint accuracy. As

| name | batch | bbox | ⊕ | ⊖ | $T_c$ | sel. | ext. bbox | P-Mc | bbox by IoU | bbox | segm | pose |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTMDet [22] + 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 8. Ablation study on prompting SAM [25] 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.

human pose estimation models are often overconfident, using self-estimated OKS from [13] 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. 8 shows that combining confidence and distance (c+d) outperforms either approach alone, providing superior results.

**Extending bounding box**. Experiment F2 in Tab. 8 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. 8 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 for instances with $max\_IoU > 0.5$. The rationale is that bounding boxes are typically accurate for isolated instances,

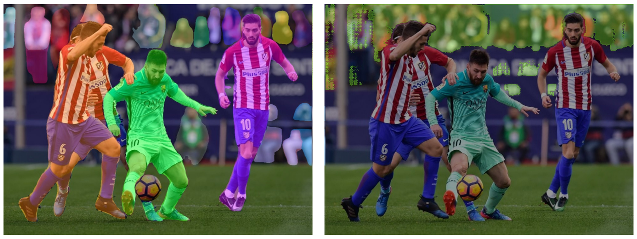


Figure 6. Multiple background instances may merge into a single mask when no bounding box is provided as a prompt. The yellow mask was refined and covers all spectators. Foreground instances are omitted in the left image for clarity.
Left – RTMDet [22], right – BMP.

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. 8.

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. 8 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.

Table 9. **Detection results on CIHP [12].** BMP brings a small improvement; CIHP is more similar to COCO than to OCHuman.

| | det AP | mask AP |
|---|---|---|
| RTMDet-l | 69.5 | 63.9 |
| BMP 1× | 69.4 -0.1 | 65.7 +1.8 |
| BMP 2× | **69.7** +0.2 | **65.9** +2.0 |

**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. 7), and almost always outperforms the ViTPose [38] 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. 8, 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. Additional results

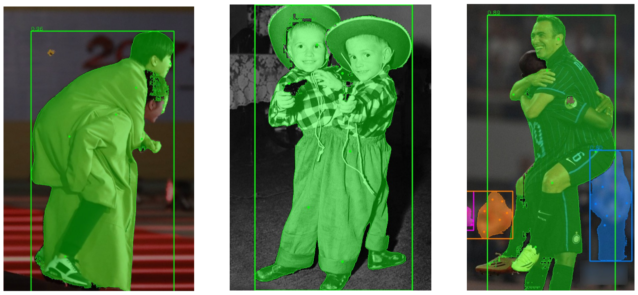
Tab. 9 shows results on CIHP dataset [12]. BMP is the most effective in scenarios with max IoU between 0.5 and 1.0 (see also Tab. 3). The improvement in non-crowd scenes (e.g. COCO) is negligible. Note that not all crowd datasets are equal. COCO, CIHP and CrowdPose feature group photos with many bboxes tightly squeezed next to each other. On the other hand, OCHuman and part of CIHP feature entangled people with highly overlapping bboxes. BMP excels in the most difficult scenes with overlapping bboxes, while not harming performance on group photos.

## C. 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-

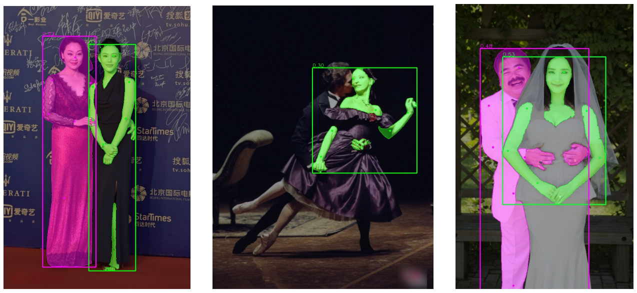


(a) Two people in matching coats.

(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 [25], typically due to similar or identical textures.



(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 BMP iterations, the detector identifies these as separate instances, as shown in Fig. 4.

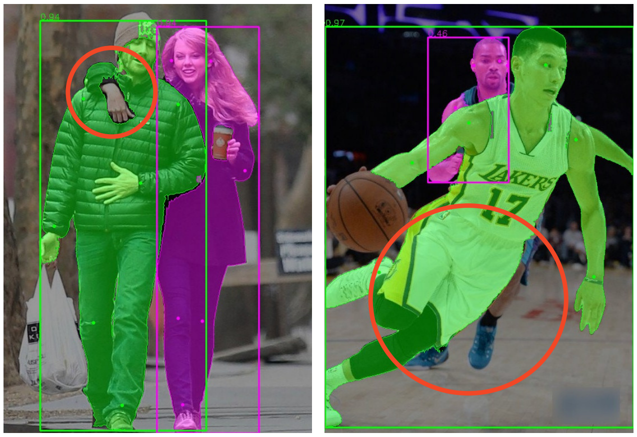


Figure 9. Images where SAM [25] successfully decoupled instances but failed to segment a disconnected body part. These parts remain unmasked and risk being re-detected, as illustrated in Fig. 4c. Keypoints used for SAM prompting are marked (best viewed in zoom).

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.

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. 11 and 12 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.

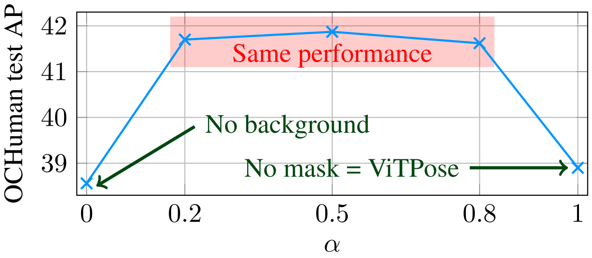


Figure 10. MaskPose performance with **different values of** $\alpha$. Fine-tuning for 5 epochs on 10% of dataset, masks detected.

| 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× | <u>31.9</u> | <u>47.3</u> | 282 M |
| ✓ | ✗ | ✗ | 2× | 30.8 | 47.0 | 201 M |

Table 10. **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 $1\times$, second to BMP $2\times$.

## D. Additional ablation Study

### D.1. Semi-transparency for MaskPose

Fig. 10 shows preliminary experiments on the $\alpha$ values in MaskPose. When $\alpha = 0$, the model loses the background context and becomes sensitive to detected mask quality. For $\alpha \in [0.2, 0.8]$, the model combines the foreground and the background and exhibits good and stable performance.

### D.2. Number of parameters of BMP

Tab. 4 in Sec. 4.3 shows the performance change with and without various BMP components. For clarity, we also present Tab. 10, which shows the same result along with the number of trainable parameters of the whole loop. For example, combining the detector (RTMDet-l) with 57M parameters and the pose model (ViTPose-b) with 87M parameters results in 144M trainable parameters.

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

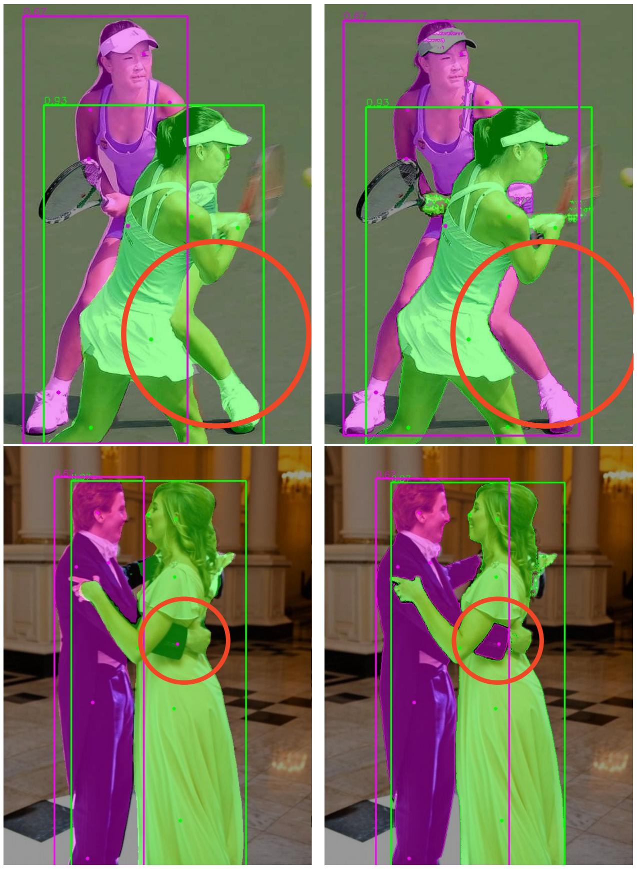


Figure 11. Images where BMP improves detection and segmentation using its pose estimates and SAM prompting with selected keypoint. Bounding box prompting did not lead to comparable results. Keypoints used for SAM prompting are marked (best viewed in zoom). Left – RTMDet [22], right – BMP.

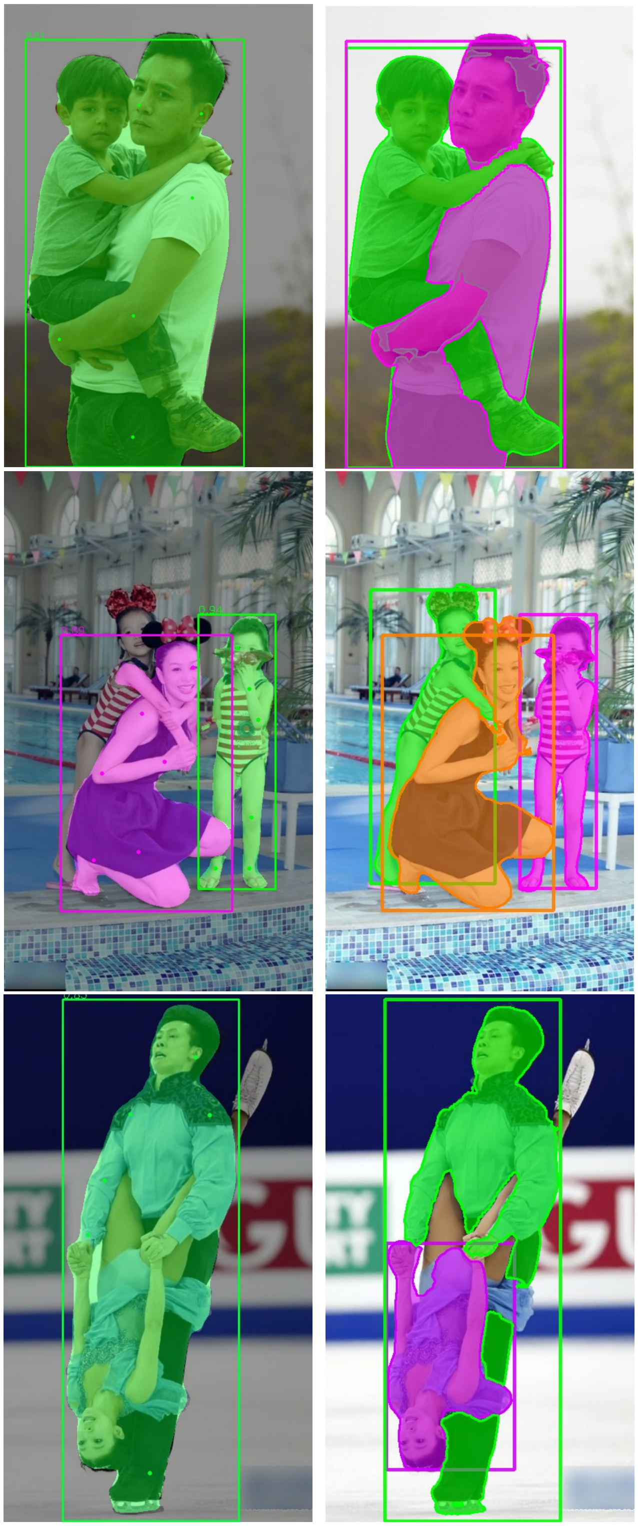


Figure 12. Two iterations of BMP successfully decouple merged instances, even in challenging images with upside-down people. Left – RTMDet [22], right – BMP.

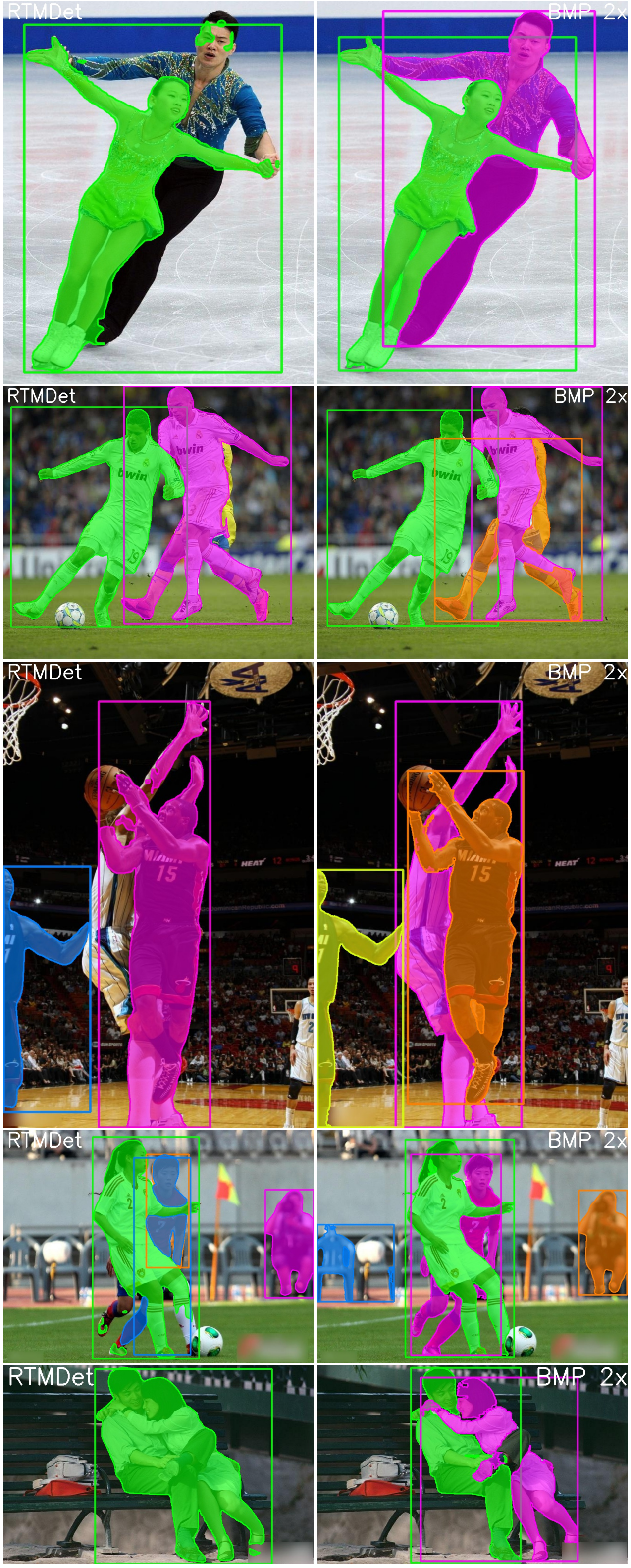


Figure 13. Qualitative results on the OCHuman dataset. Left – RTMDet [22], right – BMP 2×.

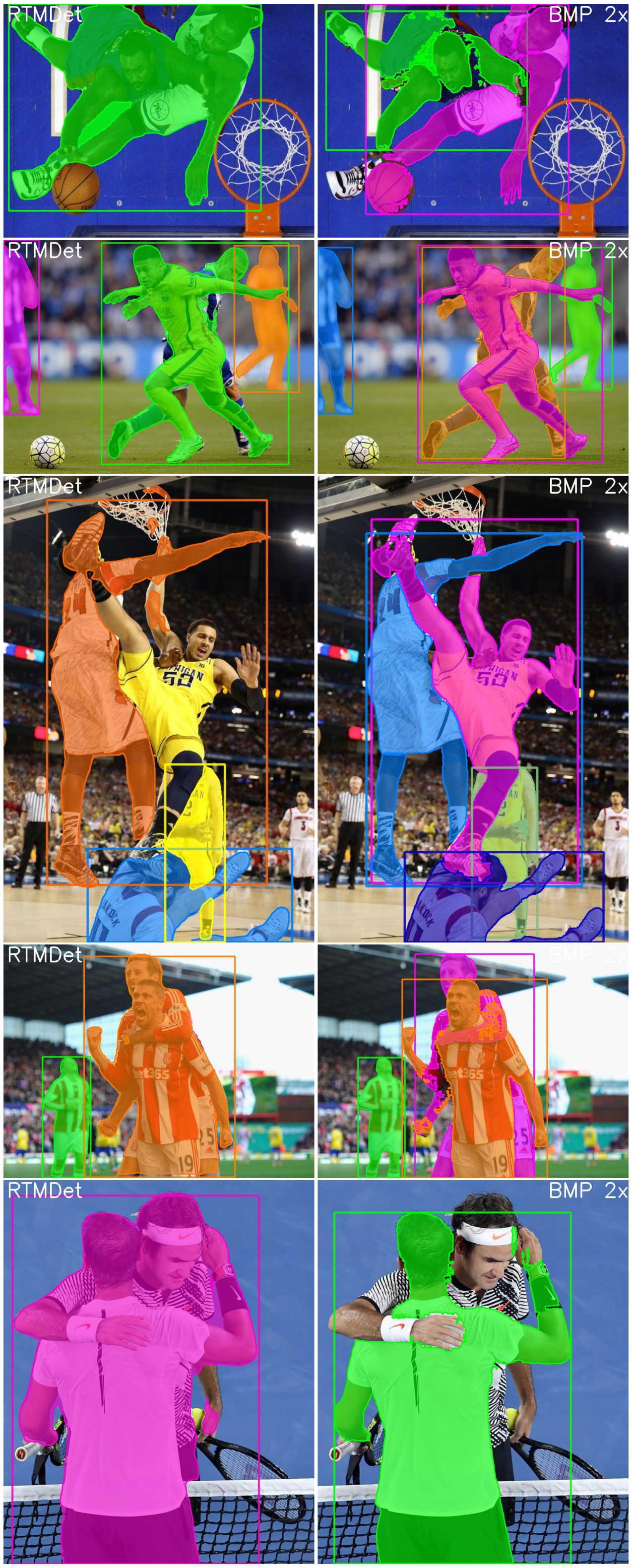


Figure 14. More qualitative results on the OCHuman dataset. Left – RTMDet [22], right – BMP 2×.