Supplemental materials for “PourIt!: Weakly-supervised Liquid Perception from a Single Image for Visual Closed-Loop Robotic Pouring”

Haitao Lin Yanwei Fu[†] Xiangyang Xue[†]
Fudan University

This supplemental material is organized as follows. In Section A, we give the details of pre-processing the Liquid dataset [7]. In Section B, we discuss the limitation of the Liquid dataset as referred to in the main paper. The data collection efficiency is compared in Section C. The details of the robotic experiments are illustrated in Section D. More qualitative results are shown and discussed in Section E. Additional examples are shown in our supplemental video.

A. Pre-processing of the Liquid Dataset

As the sequences of each trial on the Liquid dataset are generated by the robot pouring the container, thus it contains images of liquid and non-liquid. We thus design a strategy to utilize the provided mask annotations for automatically separating each sequence into images with liquid and non-liquid. Concretely, we visualize the sum of annotated pixels of liquid in each frame as in Fig. 1. We observe that the number of pixels belonging to liquid will grow during continuous pouring. Thus, we search the turning points (orange and green ones in Fig. 1) on the curve. Finally, the sequence is dived into two parts of liquid and non-liquid for training and testing.

B. Limitation of Liquid Dataset

The UW Liquid Pouring dataset [7] is an useful dataset in tasks of estimating liquid volume like the excellent works [6, 7]. Nevertheless, it is not completely suitable for our task, *i.e.*, perceiving the liquid out-flowed from bottleneck. The reasons comes from two folds: (1) The annotations in Liquid dataset contain liquid out-flowed from bottleneck, and liquid stayed in the container. In our task, we only focus and model the liquid out-flowed from bottleneck for visual closed-up control. (2) The annotation quality of some images are imperfect as in Fig 2. Some liquid regions are unlabeled in original dataset, but our method can well discover and focus on the such unlabeled regions of liquid, thus this will influence the final evaluation of performance.

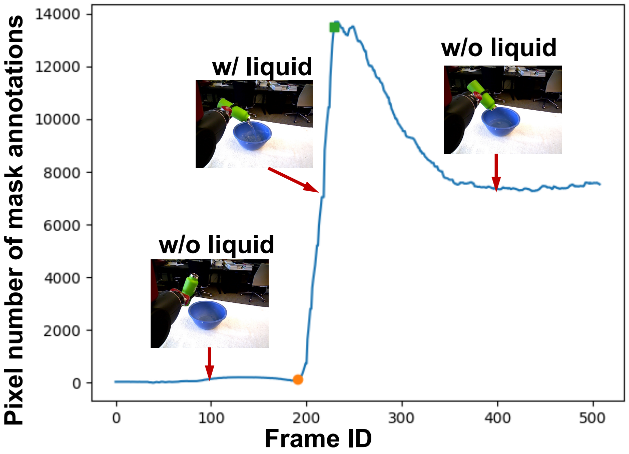


Figure 1: The curve of pixel number of annotations per frame in UW Liquid Pouring dataset [7]. The turning points on the curve helps divide the sequences into two parts of images with liquid and non-liquid.

C. Efficiency of Data Collection

We compare the efficiency of our image-level labeled data collection pipeline and that of pixel-wise annotation pipeline. We labeled the coarse pixels annotation using the PaddleLabel tools [2], and then using the Apple pencil [1] (the mouse is hard to use as the liquid is long and thin) to refine the labeled pixels. It takes nearly $2.5h$ for 200 images. In contrast, we easily obtain the image-level labeled data in a semi-automatic manner within $10min$, which only demands the human to put containers on the workspace of the robot. The efficacy of our pipeline is **15** times faster than that of pixel-wise annotation pipeline. Although these works [5, 6, 7] utilize the thermal camera and heating water to accelerate the efficacy off annotation, but it needs extra calibration of thermal camera aligned to the RGB camera and heating water. Additionally, the generated annotations sometimes are imperfect as shown in Fig 2 and discussed in Sec. B. Thus, our data collection pipeline is easy and efficient, which do not rely on tedious human labor and additional equipment.

[†] indicates corresponding author.

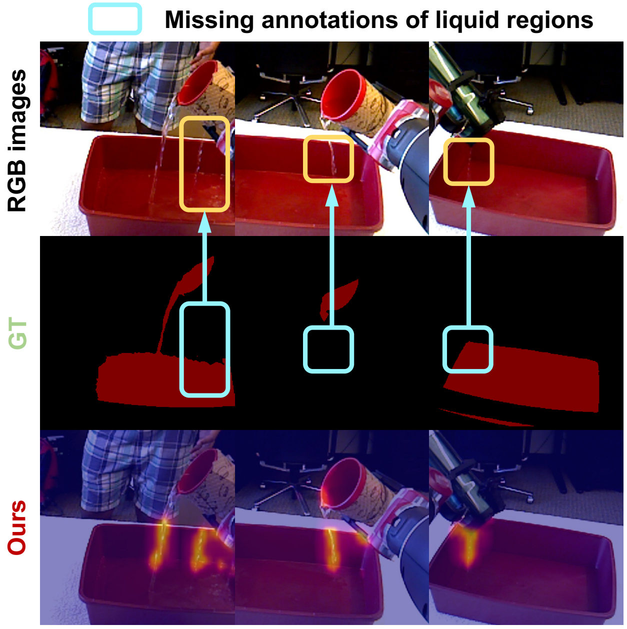


Figure 2: Illustration of the limitation in the Liquid dataset [7]. Some of the actual liquid regions are unlabeled in the original provided annotations, while our method can discover such small regions of liquid.

D. Robotics Experiment Setup

We use six groups of source-target containers for robotic experiments, and the as shown in Fig. 3. For each trial, the source and target containers are placed randomly within the camera's field of view.

Pose Tracking. When the source container is rotated horizontally to the ground, the pre-processing stage of object segmentation in the method of Lin *et al.* [3] cannot detect the object, rendering further pose estimation unavailable. To address this, we employ a pseudo-tracking strategy to calculate the pose of the source container using the robot's forward kinematics. Specifically, we assume that when the source container is grasped by the robot, the object is attached to the robot's gripper. Therefore, the source container's pose is written as,

$$\Delta T_n^{grip} \cdot T_0^{obj} = T_n^{obj} \tag{1}$$

where T_0^{obj} is the initial object pose when the gripper exactly touches and grasps this object at time t_0, and ΔT_n^{grip} is the relative transformation of gripper at time t_n versus that at time t_0. Finally, the tracking pose of the object attached to the gripper (at time t_n) is denoted as T_n^{obj}.

Initial Pouring Point Calculation. The robot first grasps the source container by using the pose estimator proposed by Lin *et al.* [3]. Then the initial pouring point is calculated by using the estimated pose and size as in Fig. 4.

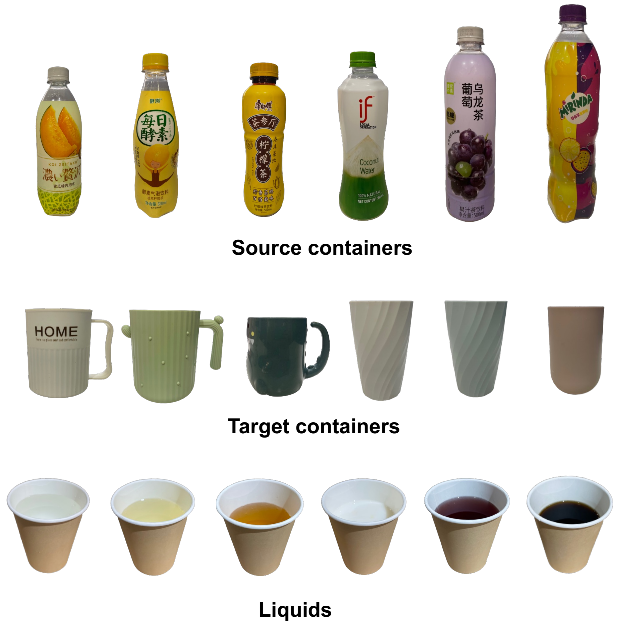


Figure 3: Different containers and liquids were used in our experiments. We use six groups of source containers, target containers, and liquids, individually.

Given the poses and sizes of both containers in the robot's base frame, *i.e.*, $\{\boldsymbol{R}^{src}, \boldsymbol{t}^{src}; \boldsymbol{s}^{src}\}$ and $\{\boldsymbol{R}^{tgt}, \boldsymbol{t}^{tgt}; \boldsymbol{s}^{tgt}\}$, we simplify the calculation by only considering the 3-DoF translation and 3D size of containers for in our experiments. The initial pouring point $\boldsymbol{t}^p$ is calculated by,

$$\begin{cases} t_x^p = t_x^{tgt} \\ t_y^p = t_y^{tgt} - s_x^{tgt}/2 \\ t_z^p = t_z^{tgt} + s_x^{src}/2 + s_y^{tgt}/2 + h \end{cases} \tag{2}$$

where h is a margin of safety pouring height to avoid the bottleneck of the source container colliding with that of the target container if the containers are too close when pouring. We set h as $3.5cm$ in our experiment. t_x^p, t_y^p and t_z^p are components of vectors $\boldsymbol{t}^p$.

Details of Dynamic Scenes. For the dynamic scenes, the experimenter holds and moves the containers in linear motion and random motion as illustrated in Fig. 5. The dynamic scenes are very challenging, as the out-flowing liquid has a certain speed in moving direction when the robot continuously adjusts the source containers to track the target one. This demands real-time visual feedback to control the position of the source container. Our experimental results show our capability of real-time visual feedback to guarantee the liquid dropped into the moving target containers.

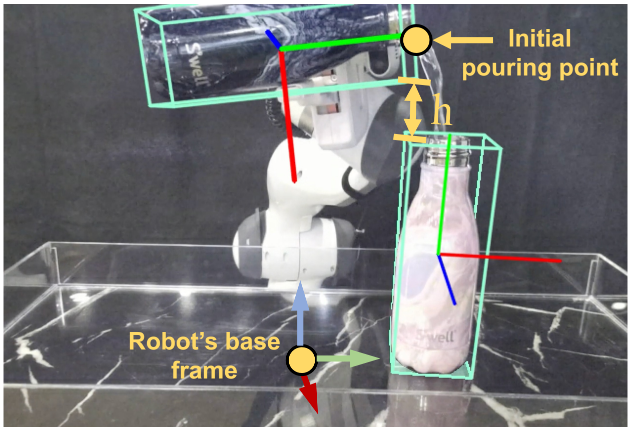
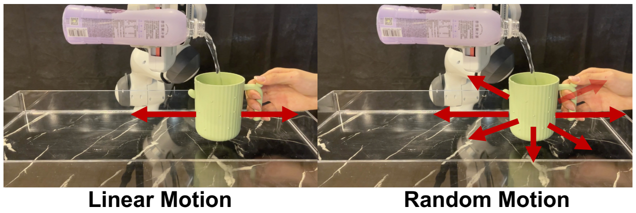


Figure 4: Illustration of initial pouring point calculation. The narrows with red, green, and blue colors means the x, y, and z axis, individually.



Linear Motion **Random Motion**

Figure 5: Different motion types of the target container used in our robotic experiment of dynamic scenes.

E. Qualitative Results

Here we provide more qualitative examples on the dataset to show the qualitative evaluation. We show the input RGB images and CAM results overplayed on the RGB images on the UW Liquid Pouring (Fig. 6) and *PourIt!* (Fig. 7) datasets.

References

[1] Apple Pencil. `https://www.apple.com/hk/en/apple-pencil/`. 1

[2] PaddlePaddle Authors. Paddlelabel, an effective and flexible tool for data annotation. `https://github.com/PaddleCV-SIG/PaddleLabel`, 2022. 1

[3] Haitao Lin, Zichang Liu, Chilam Cheang, Yanwei Fu, Guodong Guo, and Xiangyang Xue. Sar-net: Shape alignment and recovery network for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2022. 2

[4] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: end-to-end weakly-supervised semantic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16846–16855, 2022. 4, 5, 6

[5] Connor Schenck and Dieter Fox. Towards learning to perceive and reason about liquids. In *International Symposium on Experimental Robotics*, pages 488–501. Springer, 2016. 1, 4

[6] Connor Schenck and Dieter Fox. Visual closed-loop control for pouring liquids. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2629–2636. IEEE, 2017. 1

[7] Connor Schenck and Dieter Fox. Perceiving and reasoning about liquids using fully convolutional networks. *The International Journal of Robotics Research*, 37(4-5):452–471, 2018. 1, 2

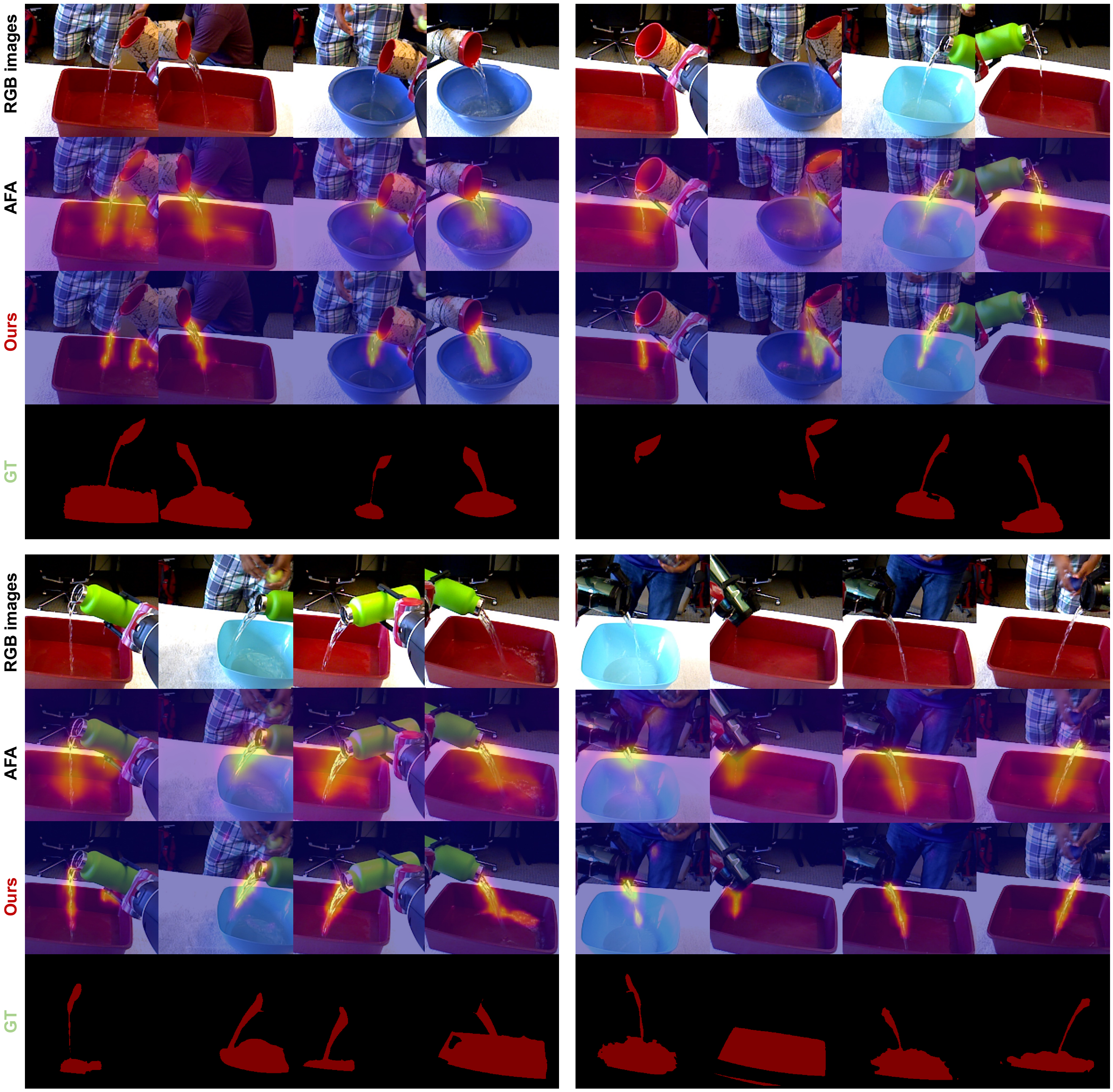


Figure 6: More CAM visualization results of different methods on the Liquid dataset [5]. Note that the quality of AFA's CAM is similar to its segmented mask results, thus we only visualize CAM but not AFA segmented masks [4] for consistency of visual contrast. The comparison results show that our CAM results focus more tightly on complete liquid region, with well alignment of low-level boundaries.

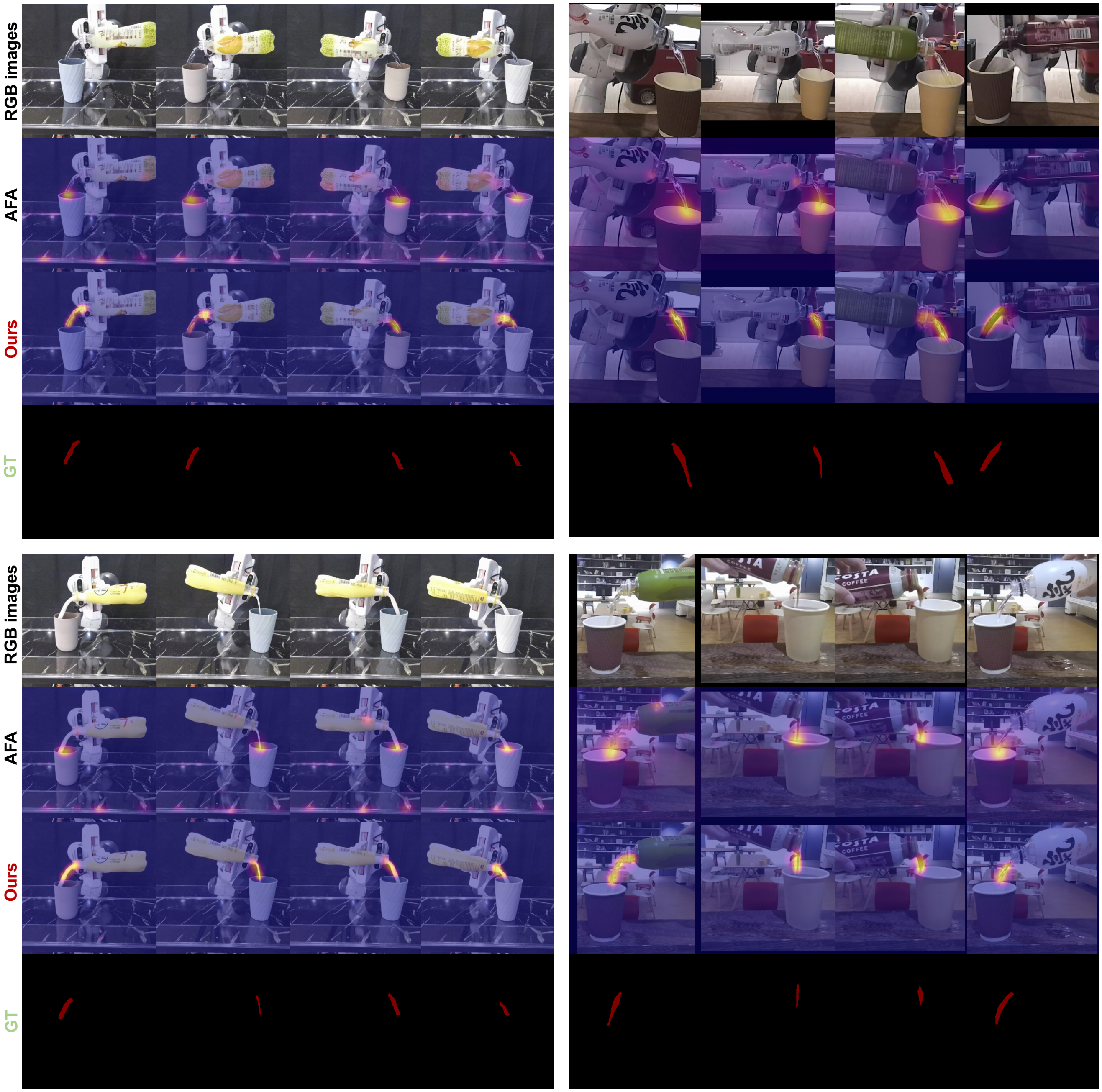


Figure 7a: More CAM visualization results of different methods on the PourIt! dataset. Note that the quality of AFA's CAM is similar to its segmented mask results, thus we only visualize CAM but not AFA segmented masks [4] for consistency of visual contrast. The comparison results show that our CAM results focus more tightly on complete liquid region, with well alignment of low-level boundaries and generalization across novel scenes.

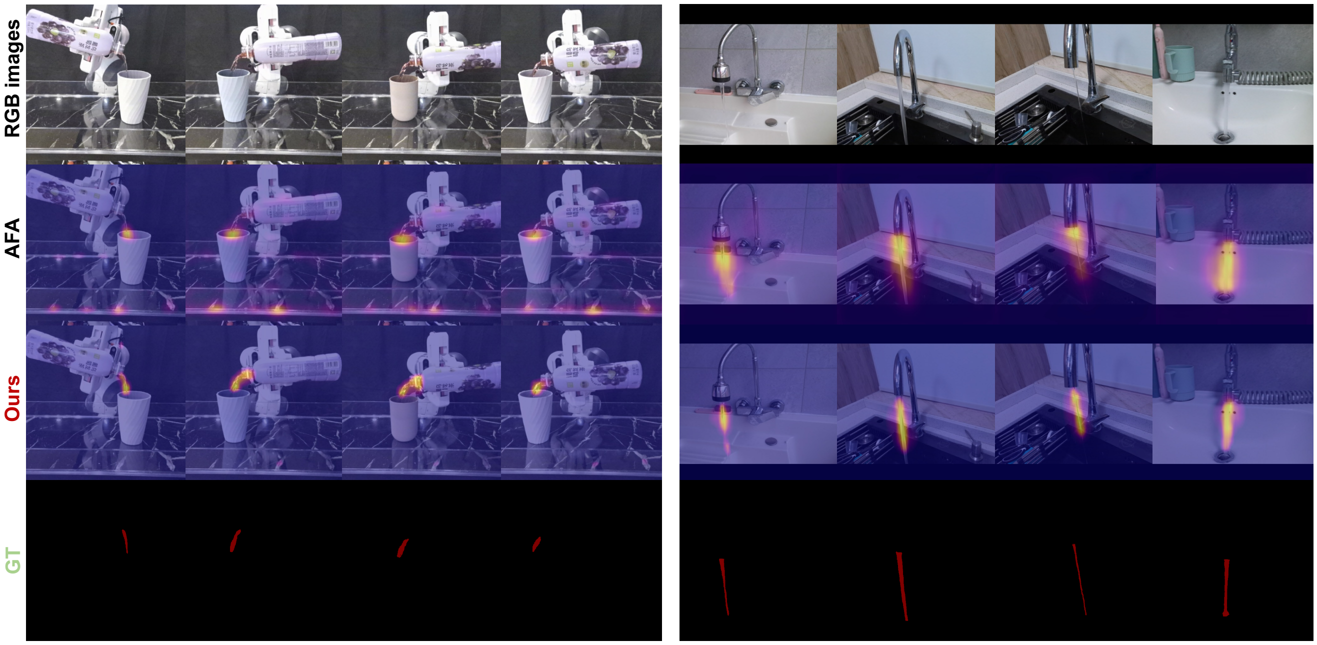


Figure 7b: More CAM visualization results of different methods on the PourIt! dataset (following the figure above). Note that the quality of AFA's CAM is similar to its segmented mask results, thus we only visualize CAM but not AFA segmented masks [4] for consistency of visual contrast. The comparison results show that our CAM results focus more tightly on complete liquid region, with well alignment of low-level boundaries and generalization across novel scenes.