# READ: Retrieval-Enhanced Asymmetric Diffusion for Motion Planning

Supplementary Material

Takeru Oba[†], Matthew Walter[‡], Norimichi Ukita[†]
[†] Toyota Technological Institute, ‡ Toyota Technological Institute at Chicago
sd21502@toyota-ti.ac.jp, mwalter@ttic.edu, ukita@toyota-ti.ac.jp

## 6. Task details

Figure 11 visualizes demonstration sequences from the 16 tasks, which we describe below. Unless specified otherwise, objects are randomly placed on the table.

**Pick up Cup (PC)** This task involves grasping and lifting a red cup placed on the table. In addition to the red cup, there is a distractor cup of various colors placed randomly. The prediction is labeled as a failure if the robot grasps the distractor cup. Since the red cup is always picked in demonstrations, motion planning models learn that the red cup is the target.

**Reach Target (RT)** The task involves directing the robot end-effector to reach a red sphere placed in the workspace. Unlike other tasks, the target (red sphere) is suspended in the air and does not make contact with the table. In addition to the red sphere, two randomly colored spheres are also placed at random locations, and the challenge is to ignore these distractors while precisely reaching the red target.

**Push Button (PB)** The task is to press a button placed on the table.

**Open Wine (OW)** The task involves removing the cap of a vertical wine bottle placed on the table.

**Close Box (CB)** The task is to close an open box placed on the table.

**Put Rubbish (PR)** The task is to dispose of rubbish by putting it into a trash bin. In addition to the rubbish, two tomatoes are placed on the table as distractors, requiring the successful identification and disposal of only the rubbish while ignoring the tomatoes.

**Stack Wine (SW)** The task is to lay a wine bottle on its side on a rack. The positions of racks and bottles are random, while where the bottles should be placed on the racks is fixed across episodes. If the bottle is not placed in the predetermined position, it is labeled a failure.

**Place Hunger (PH)** The task involves hanging hangers on a rack. The hangers are randomly positioned, while the location of the rack is fixed across episodes.

**Water Plants (WP)** The task involves using a watering can to provide water to a plant. The watering can is already filled with water.

**Beat the Buzz (BB)** In this task, the robot needs to move a wand with a ring from one end of a pole to the other without making contact.

**Put Knife (PK)** The task involves removing a knife from a container and placing it on a cutting board. To extract the knife, it is necessary to pull it directly upward.

**Take Plate (TP)** The task is to take a plate from a container and place it on top of a blue cube.

**Straighten Rope (SR)** The task involves straightening a rope bent in an S-shape.

**Put Umbrella (PU)** The task is to grab the handle of an umbrella placed on top of a container and put the umbrella into the container.

**Scoop with Spatula (SS)** The task is to place a square cube on top of a spatula. Rather than gripping and directly placing the cube onto the spatula, the challenge involves grabbing the spatula, smoothly maneuvering it under the cube, and scooping the cube with momentum.

**Stack Blocks (SB)** The task is to stack any two out of four randomly placed red cubes onto a marked spot in the center of the table. In addition to the red cubes, cubes of random colors are placed as distractors, but stacking these distractors will not lead to task success.

## 7. Implementation details

In this section, we provide the details of the baseline methods that we compare to (i.e., Deterministic, VPSDE, and VPSDE+CG). Note we provide implementations of these methods in the code that we make available at https://github.com/Obat2343/READ. Unless specified otherwise, we use the Adam optimizer and optimize the models until the training loss converges.

**Deterministic:** Figure 12(a) shows the architecture of the Deterministic baseline. Given an RGB-D image $I$, the network generates a feature map using an encoder with a Resnet18 backbone. To accommodate the four channels of the RGB-D image, we modify the input channel size of the first layer of Resnet18. It then feeds a flattened image feature map into a two-layer perception with the Gelu activation function to predict the motion $\hat{M}(0)$. We train this model to minimize the mean square error (MSE) between the predicted $\hat{M}(0)$ and ground-truth $M(0)$ motions. Note that we train the model from scratch.

**VPSDE:** Our implementation of VPSDE adapts that of Song et al. [50], which is designed for image synthesis tasks, to perform motion planning. Figure 12(b) shows the model architecture of VPSDE that we implemented. We adopt the same architecture as READ. $\text{Enc}^I$ is ConvNext-

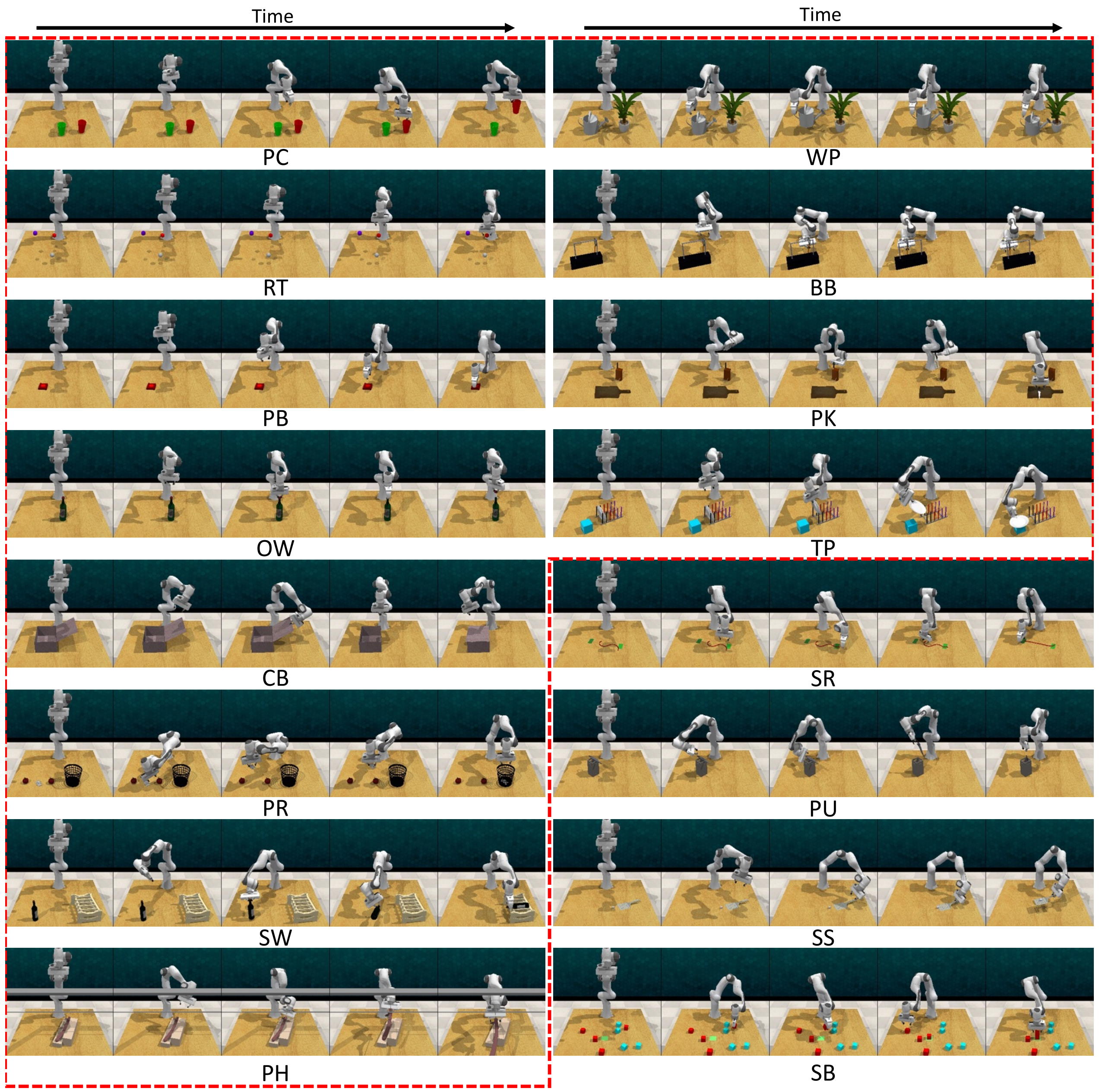


Figure 11. Demonstration sequences for the 16 different tasks. The red dashed line encloses tasks included in Avg12.

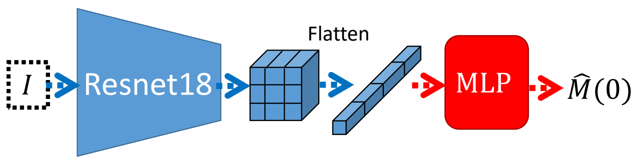
based UNet [27, 44]. $\mathrm{Enc}^p, \mathrm{Enc}^t$ and $\mathrm{Dec}^e$ are three-layer perceptrons with a Gelu activation function [17]. $\mathrm{Enc}^f$ is a multi-head transformer encoder [54]. PE denotes sinusoidal positional encoding [54]. While READ predicts the motion $M(0)$ directly, VPSDE predicts the noise $\epsilon_\theta(I, M(t), t)$ from $I$, $M(t)$, and $t$, where $M(t)$ is the perturbed motion at diffusion step $t$ obtained by forward SDE. While READ obtains $M(t)$ by a latent-space forward SDE, VPSDE obtains $M(t)$ by the following a different forward SDE:

$$dx = -\frac{1}{2}\beta_t x dt + \sqrt{\beta_t} dw, \tag{6}$$

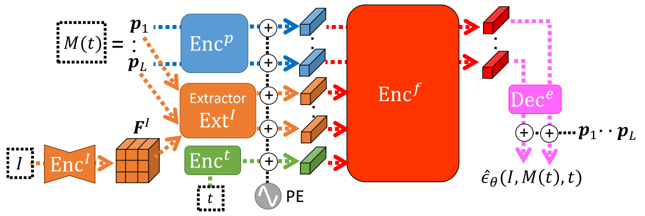
where $x = M$ in our scenario. We linearly schedule $\beta_t$ from 1.0 to 20.0. VPSDE is trained to minimize the following MSE loss:

$$\mathcal{L} = \|\epsilon - \epsilon_\theta(I, M(t), t)\|. \tag{7}$$

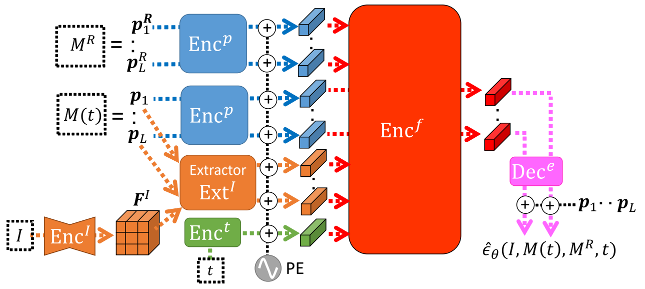
**VPSDE+CG:** We apply Classifier-free Guidance (CG) to VPSDE for a retrieval-based diffusion model baseline. Figure 12(b) shows the model architecture of VPSDE+CG. The



(a) Model architecture for the Deterministic baseline.



(b) Model architecture for the VPSDE baseline.



(c) Model architecture of VPSDE+CG.

Figure 12. Baseline model architectures.

model architecture is almost the same as the VPSDE except that the retrieved motion $M^R$ is additionally conditioned. $M^R$ is embedded by the pose encoder Enc$^p$ that is shared with $M(t)$ and the embedded feature is fed into Enc$^f$ (i.e., a transformer). During inference, VPSDE+CG predicts the noise as follows:

$$\hat{\epsilon} = w\epsilon_\theta(I, M(t), M^R, t) - (w-1)\epsilon_\theta(I, M(t), t). \quad (8)$$

The parameter $w$ controls the strength of guidance. The unconditional term $\epsilon_\theta(I, M(t), t)$ is implemented just by masking the condition $M^R$. To strongly guide the reverse process by $M^R$, we set $w = 1.0$ We train the model by minimizing the following MSE loss:

$$\mathcal{L} = \|\epsilon - \epsilon_\theta(I, M(t), M^R, t)\|. \quad (9)$$

To learn the unconditional prediction, we stochastically mask $M^R$ with a masking probability of 10%. Forward SDE and $\beta_t$ are the same as VPSDE.

## 8. Full tables and figures

In the main paper, we only provide Avg12 and Avg16 in Tables 2(a), 2(b), 2(c), and 3 due to the page limitation. In this section, we provide the full version of the tables. Talbes 4, 5, 6, and 7 corresponding to Tables 2(a), 2(b), 2(c), and 3, respectively. Moreover, we additionally provide Cheat ($i = 3$) and Cheat ($i = 10$) in Table 5. These results show the success rates when the $i^{\text{th}}$ nearest neighbor to the ground-truth motion is executed without refinement. Note that Cheat ($i = 1$) in Table 5 is the same as Cheat in Table 2(b).

The READ ($i = 1$) in Table 4 represents the performance when the motion retrieved by Cheat ($i = 1$) is refined. The success rates of Cheat ($i = 1$) are shown in Table 5. Across most tasks, READ ($i = 1$) outperforms Cheat ($i = 1$), with a notable increase in success rates, particularly in the cases of OW and PK. Figure 13 visualizes how READ refines the retrieved motion over iterations. Visualizations of OW and PK in Fig. 13 show that motions, which initially deviate from the ground-truth (GT) motion, gradually converge towards their GT motion. Thanks to the robust refinement ability of READ, the predictive motion succeeds even when the retrieved motion is far from the GT motion. A comparison between READ ($i = 10$) and Cheat ($i = 10$) further demonstrates the effectiveness of READ. In the case of $i = 10$, READ ($i = 10$) improves Avg12 and Avg16 by more than 40% compared to Cheat ($i = 10$), reinforcing our conclusion that READ is robust to retrieval failure.

## 9. Robustness to poor retrieval

Here, we evaluate the performance of READ when retrieved motions are far from correct motions. In additional experiments with $i$=500-th nearest motion (i.e., poor retrieval) shown in Table 8, Avg12=1.3, which is too low, if the retrieved motion is used without refinement. The retrieved motion is far from the correct one. With one-step refinement ($N$=1), READ improves performance to Avg12=47.5, and with further refinement ($N$=100), to Avg12=88.0, comparable to READ's Avg12=88.8 in Table 1.

## 10. Realistic scenarios

We evaluated the success rates when adding random noise and changes in overall brightness to the test images. For images corrupted by noise and brightness changes, Avg12 is 88.5 and 87.8 as shown in Table 9, respectively, which are comparable to READ in Table 1.

## 11. Limitations and future work

We found limitations of our method through experiments in difficult tasks in the Avg16 tasks. In the SR (straighten rope) task, a challenge arises from the configuration space. To grasp the rope, the robot's arms must be extended to their near maximum limit, making it difficult to find an appropriate pose due to reaching the limits of the robot's range of motion. To address this issue, it may be possible to combine our latent space implicit kinematics constraints with explicit constraints.

Table 4. Refinement.

| | i | Avg16 | Avg12 | PC | RT | PB | OW | CB | PR | SW | PH | WP | BB | PK | TP | SR | PU | SS | SB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| READ | 1 | 70.4 | 88.6 | 97 | 100 | 73 | 87 | 100 | 97 | 98 | 42 | 94 | 89 | 86 | 100 | 21 | 38 | 0 | 4 |
| READ-O | 1 | 61.4 | 78.3 | 95 | 98 | 84 | 79 | 100 | 97 | 91 | 48 | 57 | 31 | 65 | 95 | 2 | 37 | 3 | 3 |
| READ-L | 1 | 33.6 | 44.8 | 65 | 2 | 49 | 47 | 85 | 65 | 34 | 20 | 31 | 35 | 32 | 72 | 1 | 0 | 0 | 0 |
| READ | 3 | 70.6 | 89.3 | 97 | 100 | 75 | 88 | 100 | 96 | 98 | 55 | 89 | 88 | 86 | 100 | 19 | 35 | 3 | 2 |
| READ-O | 3 | 61.1 | 78.8 | 95 | 97 | 88 | 81 | 98 | 100 | 91 | 44 | 57 | 31 | 64 | 99 | 3 | 27 | 2 | 1 |
| READ-L | 3 | 34.1 | 45.3 | 65 | 4 | 56 | 44 | 87 | 61 | 34 | 19 | 34 | 41 | 32 | 67 | 0 | 1 | 0 | 0 |
| READ | 10 | 71.0 | 89.8 | 97 | 100 | 75 | 90 | 99 | 97 | 98 | 55 | 92 | 88 | 86 | 100 | 19 | 36 | 2 | 3 |
| READ-O | 10 | 60.1 | 77.3 | 95 | 98 | 86 | 82 | 100 | 95 | 91 | 41 | 60 | 25 | 58 | 97 | 3 | 23 | 3 | 4 |
| READ-L | 10 | 34.9 | 46.5 | 81 | 4 | 60 | 37 | 85 | 69 | 37 | 19 | 34 | 40 | 29 | 63 | 0 | 0 | 0 | 0 |

Table 5. Retrieval.

| | Avg16 | Avg12 | PC | RT | PB | OW | CB | PR | SW | PH | WP | BB | PK | TP | SR | PU | SS | SB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cheat (i=1) | 60.6 | 79.5 | 85 | 96 | 96 | 57 | 86 | 99 | 85 | 42 | 87 | 76 | 51 | 94 | 1 | 13 | 0 | 1 |
| Cheat (i=3) | 48.7 | 64.1 | 67 | 81 | 83 | 35 | 71 | 92 | 70 | 30 | 86 | 38 | 36 | 80 | 0 | 7 | 2 | 1 |
| Cheat (i=10) | 30.7 | 40.4 | 56 | 21 | 47 | 17 | 64 | 58 | 50 | 9 | 72 | 16 | 19 | 56 | 0 | 2 | 3 | 1 |
| READ | 56.6 | 73.5 | 73 | 89 | 86 | 37 | 97 | 97 | 77 | 37 | 91 | 73 | 34 | 91 | 0 | 19 | 3 | 1 |
| READ-O | 57.1 | 75.5 | 95 | 82 | 100 | 70 | 93 | 98 | 80 | 31 | 79 | 48 | 37 | 93 | 0 | 5 | 0 | 2 |
| READ-L | 52.4 | 68.9 | 92 | 11 | 100 | 46 | 99 | 100 | 79 | 29 | 80 | 69 | 30 | 92 | 0 | 9 | 1 | 2 |

In the PU task, which involves placing an umbrella into an umbrella stand, a problem arises from even slight deviations in the angle of holding the umbrella causing significant changes in the position of its tip. Incorporating a mechanism to provide feedback on the grasped result could allow fine adjustments to the grasping position.

In the SS task, where a cube is scooped with a spatula, requiring deliberate and forceful sliding motion, the importance of force control for future work is emphasized, as most planning methods, including ours, currently predict only the trajectory.

In the SB task, the original motions consisted of approximately 350 to 400 frames, which have been aligned to 100 frames through interpolation. This interpolation results in the loss of crucial frame information (e.g., grasping frame). To mitigate this issue, it would be intriguing to explore the incorporation of dynamic prediction length models [12, 34] or keyframe-based motion planning techniques [40, 47] to prevent such information loss.

Table 6. EM vs. CD with READ-O

| | Avg16 | Avg12 | PC | RT | PB | OW | CB | PR | SW | PH | WP | BB | PK | TP | SR | PU | SS | SB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| w/EM | 60.1 | 78.5 | 85 | 96 | 96 | 57 | 86 | 99 | 85 | 42 | 87 | 76 | 51 | 94 | 1 | 13 | 0 | 1 |
| w/CD | 58.8 | 77.4 | 94 | 75 | 49 | 73 | 81 | 97 | 98 | 50 | 89 | 66 | 66 | 91 | 0 | 9 | 3 | 2 |

Table 7. Success rates of READ with different hyperparameters.

| Setting | N | K | L | T | Avg16 | Avg12 | PC | RT | PB | OW | CB | PR | SW | PH | WP | BB | PK | TP | SR | PU | SS | SB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 100 | 3 | 0.5 | 2.0 | 70.2 | 88.8 | 96 | 99 | 72 | 88 | 100 | 98 | 97 | 57 | 91 | 86 | 82 | 100 | 16 | 36 | 4 | 1 |
| B | 10 | 3 | 0.5 | 2.0 | 69.9 | 88.0 | 96 | 99 | 76 | 88 | 100 | 93 | 99 | 50 | 91 | 83 | 83 | 98 | 20 | 34 | 4 | 4 |
| C | 1 | 3 | 0.5 | 2.0 | 68.8 | 87.0 | 97 | 98 | 54 | 91 | 100 | 97 | 100 | 52 | 88 | 90 | 78 | 100 | 18 | 37 | 1 | 0 |
| D | 100 | 10 | 0.5 | 2.0 | 67.1 | 85.9 | 98 | 97 | 64 | 86 | 99 | 97 | 99 | 31 | 90 | 90 | 80 | 100 | 3 | 37 | 3 | 0 |
| E | 100 | 1 | 0.5 | 2.0 | 67.1 | 85.5 | 97 | 97 | 64 | 87 | 100 | 95 | 98 | 35 | 93 | 90 | 71 | 99 | 7 | 35 | 2 | 5 |
| F | 100 | 3 | 1.0 | 2.0 | 68.9 | 87.4 | 98 | 81 | 85 | 89 | 99 | 97 | 100 | 40 | 95 | 97 | 69 | 99 | 4 | 47 | 2 | 0 |
| G | 100 | 3 | 0.1 | 2.0 | 59.5 | 76.6 | 86 | 71 | 66 | 67 | 100 | 94 | 95 | 47 | 92 | 68 | 38 | 95 | 6 | 17 | 2 | 8 |
| H | 100 | 3 | 0.5 | 20.0 | 65.4 | 84.3 | 96 | 95 | 62 | 89 | 100 | 95 | 100 | 31 | 91 | 94 | 60 | 98 | 5 | 28 | 2 | 0 |
| I | 100 | 3 | 0.5 | 0.1 | 67.4 | 85.9 | 95 | 95 | 68 | 95 | 100 | 92 | 100 | 38 | 92 | 82 | 75 | 99 | 6 | 32 | 1 | 8 |

Table 8. Refinement from poor retrieval.

| | i | N | Avg16 | Avg12 | PC | RT | PB | OW | CB | PR | SW | PH | WP | BB | PK | TP | SR | PU | SS | SB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| READ | 500 | 0 | 1.0 | 1.3 | 0 | 2 | 0 | 0 | 4 | 0 | 1 | 0 | 3 | 5 | 0 | 1 | 0 | 0 | 0 | 0 |
| READ | 500 | 1 | 36.4 | 47.5 | 75 | 73 | 32 | 38 | 55 | 43 | 80 | 15 | 66 | 24 | 36 | 33 | 0 | 13 | 0 | 0 |
| READ | 500 | 100 | 69.2 | 88.0 | 95 | 99 | 77 | 85 | 100 | 96 | 97 | 55 | 89 | 80 | 83 | 100 | 17 | 27 | 5 | 2 |

Table 9. Planning from perturbed images.

| | type | Avg16 | Avg12 | PC | RT | PB | OW | CB | PR | SW | PH | WP | BB | PK | TP | SR | PU | SS | SB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| READ | noise | 69.3 | 88.5 | 96 | 99 | 74 | 90 | 100 | 95 | 99 | 51 | 86 | 89 | 83 | 100 | 15 | 29 | 1 | 1 |
| READ | brightness | 69.9 | 87.8 | 94 | 98 | 79 | 88 | 100 | 97 | 99 | 40 | 91 | 87 | 81 | 100 | 19 | 37 | 7 | 2 |

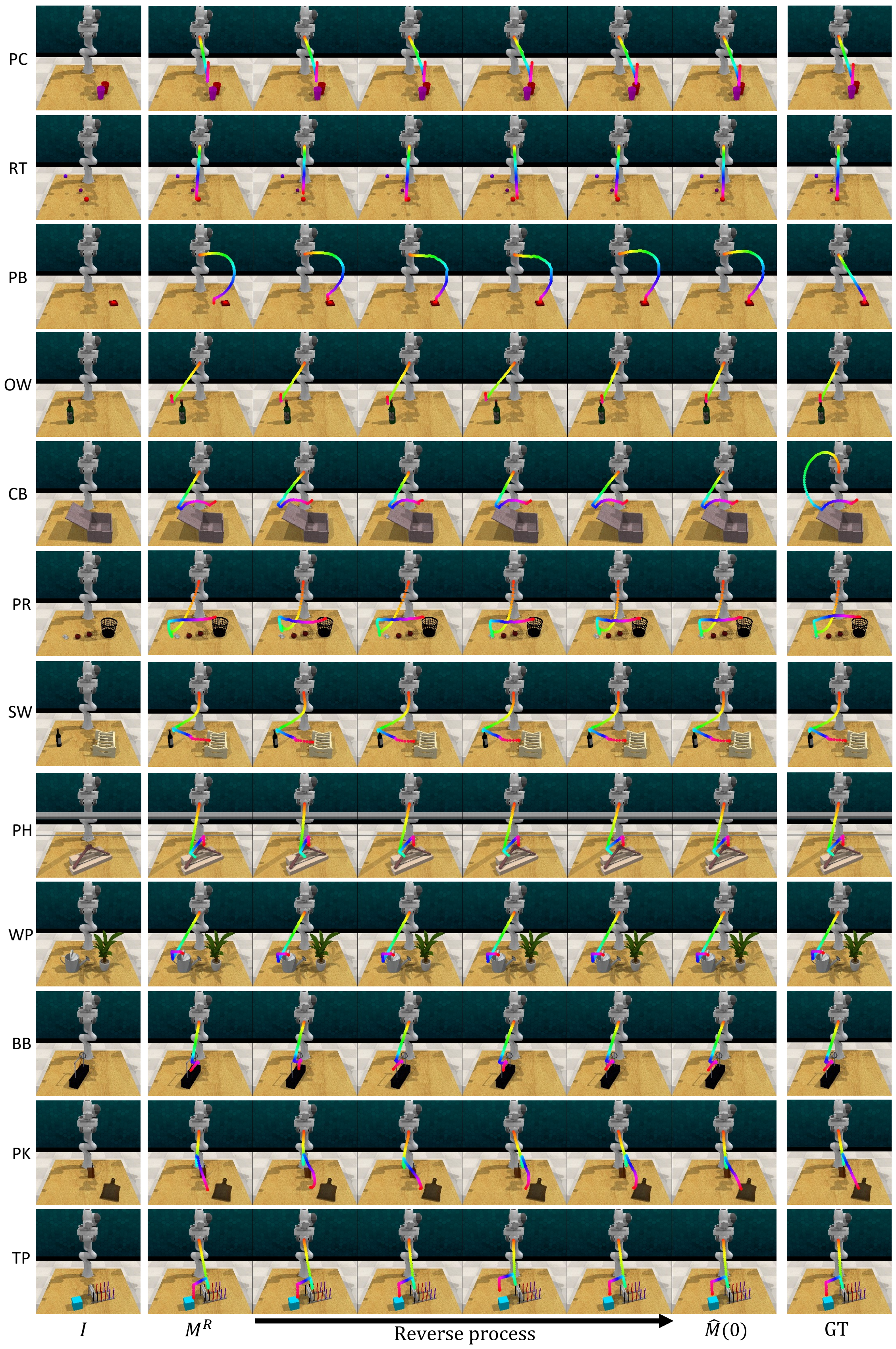


Figure 13. Visualization of motion refinement by READ.