

A Supplementary real-world results

Table 3 reports the per-task success rates of CoA, ACT, and DP across 8 real-world kitchen manipulation tasks. CoA consistently achieves the highest average performance.

Table 3: Per-task success rate in real-world experiments.

Task	CoA	ACT	DP
close cabinet	0.80	1.00	0.90
close fridge lower	0.20	0.50	0.60
close fridge upper	0.70	0.40	0.80
open cabinet	0.50	0.40	0.10
open fridge lower	0.70	0.40	0.00
open microwave	0.80	0.30	0.50
place in cabinet	0.70	0.10	0.00
place in fridge	0.50	0.60	0.00
Avg.	0.613	0.463	0.363

B Per-task success rates on RLBench

To complement the summary figure in the main paper, which visualizes the performance gap between CoA and baseline methods, we provide the full success rates on all 60 RLBench tasks in Table 4. This table lists the per-task success rate of CoA, ACT, and DP, along with the gap of baselines over CoA. Tasks are ordered by the maximum improvement CoA achieves over either baseline, highlighting where our method provides the most substantial gains.

Table 4: Detailed results of the overall comparison on RLBench. The simplified names used in Figure 4 are matched with their corresponding original task names. The success gap between ACT, DP and CoA is shown as superscripts.

Simplified name	Original name	CoA	ACT	DP
pick up cup	pick_up_cup	0.80	0.20 ^{-0.60}	0.00 ^{-0.80}
phone on base	phone_on_base	0.80	0.04 ^{-0.76}	0.04 ^{-0.76}
reach target	reach_target	0.84	0.88 ^{+0.04}	0.08 ^{-0.76}
remove meat	meat_off_grill	0.88	0.32 ^{-0.56}	0.16 ^{-0.72}
push button	push_button	0.76	0.08 ^{-0.68}	0.12 ^{-0.64}
put money in safe	put_money_in_safe	0.80	0.36 ^{-0.44}	0.24 ^{-0.56}
move hanger	move_hanger	0.88	0.68 ^{-0.20}	0.32 ^{-0.56}
slide block	slide_block_to_target	0.52	0.32 ^{-0.20}	0.00 ^{-0.52}
remove toilet roll	take_toilet_roll_off_stand	0.56	0.40 ^{-0.16}	0.08 ^{-0.48}
lamp off	lamp_off	0.68	0.68 ^{-0.00}	0.20 ^{-0.48}
lamp on	lamp_on	0.48	0.44 ^{-0.04}	0.00 ^{-0.48}
open door	open_door	0.92	0.44 ^{-0.48}	0.60 ^{-0.32}
open drawer	open_drawer	0.88	0.52 ^{-0.36}	0.44 ^{-0.44}
remove frame	take_frame_off_hanger	0.64	0.44 ^{-0.20}	0.24 ^{-0.40}
open washer	open_washing_machine	0.76	0.44 ^{-0.32}	0.60 ^{-0.16}
remove pan lid	take_lid_off_saucepan	0.80	0.40 ^{-0.40}	0.60 ^{-0.20}
unplug charger	unplug_charger	0.60	0.56 ^{-0.04}	0.20 ^{-0.40}
buzz game	beat_the_buzz	0.36	0.12 ^{-0.24}	0.00 ^{-0.36}
remove umbrella	take_umbrella_out_of_umbrella_stand	0.52	0.16 ^{-0.36}	0.20 ^{-0.32}
drag to target	reach_and_drag	0.64	0.36 ^{-0.28}	0.28 ^{-0.36}
get ice	get_ice_from_fridge	0.60	0.32 ^{-0.28}	0.24 ^{-0.36}

Continued on next page

Simplified name	Original name	CoA	ACT	DP
open box	open_box	0.32	0.16 ^{-0.16}	0.32 ^{-0.00}
place knife	place_knife_on_chopping_board	0.04	0.04 ^{-0.00}	0.00 ^{-0.04}
play jenga	play_jenga	1.00	1.00 ^{-0.00}	0.72 ^{-0.28}
place plate	put_plate_in_colored_dish_rack	0.32	0.12 ^{-0.20}	0.04 ^{-0.28}
put bottle in fridge	put_bottle_in_fridge	0.28	0.00 ^{-0.28}	0.00 ^{-0.28}
remove plate	take_plate_off_colored_dish_rack	0.40	0.40 ^{-0.00}	0.12 ^{-0.28}
turn tap	turn_tap	0.56	0.36 ^{-0.20}	0.32 ^{-0.24}
remove from scale	take_off_weighing_scales	0.84	0.44 ^{-0.40}	0.64 ^{-0.20}
stack wine	stack_wine	0.80	0.56 ^{-0.24}	0.56 ^{-0.24}
close drawer	close_drawer	1.00	0.96 ^{-0.04}	0.76 ^{-0.24}
close box	close_box	1.00	0.96 ^{-0.04}	0.76 ^{-0.24}
set clock	change_clock	0.40	0.28 ^{-0.12}	0.20 ^{-0.20}
hang frame	hang_frame_on_wall	0.16	0.08 ^{-0.08}	0.00 ^{-0.16}
open microwave	open_microwave	0.44	0.40 ^{-0.04}	0.40 ^{-0.04}
close fridge	close_fridge	0.92	0.84 ^{-0.08}	0.76 ^{-0.16}
remove USB	take_usb_out_of_computer	0.60	0.48 ^{-0.12}	0.72 ^{+0.12}
change channel	change_channel	0.12	0.00 ^{-0.12}	0.00 ^{-0.12}
insert USB	insert_usb_in_computer	0.92	0.80 ^{-0.12}	0.88 ^{-0.04}
seat down	toilet_seat_down	1.00	0.96 ^{-0.04}	0.88 ^{-0.12}
close grill	close_grill	0.56	0.48 ^{-0.08}	0.68 ^{+0.12}
lift block	lift_numbered_block	0.08	0.00 ^{-0.08}	0.08 ^{-0.00}
seat up	toilet_seat_up	0.84	0.76 ^{-0.08}	0.88 ^{+0.04}
take out shoes	take_shoes_out_of_box	0.08	0.00 ^{-0.08}	0.16 ^{+0.08}
take out money	take_money_out_safe	0.76	0.80 ^{+0.04}	0.68 ^{-0.08}
screw nail	screw_nail	0.08	0.12 ^{+0.04}	0.00 ^{-0.08}
water plants	water_plants	0.48	0.40 ^{-0.08}	0.56 ^{+0.08}
hockey	hockey	0.08	0.04 ^{-0.04}	0.00 ^{-0.08}
open wine	open_wine_bottle	0.36	0.28 ^{-0.08}	0.40 ^{+0.04}
hit ball	hit_ball_with_cue	0.08	0.00 ^{-0.08}	0.00 ^{-0.08}
put groceries	put_groceries_in_cupboard	0.08	0.04 ^{-0.04}	0.00 ^{-0.08}
turn on oven	turn_oven_on	0.36	0.32 ^{-0.04}	0.28 ^{-0.08}
set checkers	setup_checkers	0.04	0.00 ^{-0.04}	0.04 ^{-0.00}
basketball	basketball_in_hoop	0.76	0.72 ^{-0.04}	0.72 ^{-0.04}
hang hanger	place_hanger_on_rack	0.32	0.04 ^{-0.28}	0.00 ^{-0.32}
open grill	open_grill	0.24	0.00 ^{-0.24}	0.00 ^{-0.24}
straighten rope	straighten_rope	0.00	0.04 ^{+0.04}	0.00 ^{-0.00}
sweep dust	sweep_to_dustpan	0.92	1.00 ^{+0.08}	1.00 ^{+0.08}
press switch	press_switch	0.44	0.52 ^{+0.08}	0.56 ^{+0.12}
close microwave	close_microwave	0.72	0.80 ^{+0.08}	0.80 ^{+0.08}

C Hyperparameters for RL Bench

We provide the training and evaluation hyperparameters for CoA and all baseline methods used in the simulation experiments. To ensure a fair comparison, the hyperparameters for ACT are largely aligned with those of CoA, allowing us to isolate and assess the impact of our proposed modeling paradigm. For DP, we observe slower convergence relative to CoA and ACT, and thus extend its training duration to 100,000 iterations. In addition, we incorporate temporal ensembling into DP following the implementation in ACT. Octo converges substantially faster, and we find that 2,000 training iterations are sufficient. Given that Octo is primarily pretrained on single-camera data, we finetune it using only the front camera, while increasing the image resolution to enhance visual fidelity. All models are trained on a single NVIDIA H100 GPU per task.

Table 5: Hyperparameters for CoA

Backbone	ImageNet-trained ResNet18 [10]
Action dimension	8 (3 position + 4 quaternion + 1 gripper)
Cameras	wrist, front, right shoulder, left shoulder
Learning rate	$1e^{-4}$
Weight decay	$1e^{-4}$
Image size	128×128
Execution horizon	1
Observation horizon	1
# encoder layers	4
# decoder layers	7 (6 + 1 multi-token prediction layer)
# heads	8
Feedforward dimension	3200
Hidden dimension	512
Dropout	0.1
Iteration	20000
Batch size	128
Temporal ensembling	true (reverse temporal ensemble)
Action normalization	$[-1, 1]$

Table 6: Hyperparameters for ACT

Backbone	ImageNet-trained ResNet18 [10]
Action dimension	8 (3 position + 4 quaternion + 1 gripper)
Cameras	wrist, front, right shoulder, left shoulder
Learning rate	$1e^{-4}$
Weight decay	$1e^{-4}$
Image size	128×128
Action sequence	20
Execution horizon	1
Observation horizon	1
# encoder layers	4
# decoder layers	6
# heads	8
Feedforward dimension	3200
Hidden dimension	768
Dropout	0.1
Iteration	20000
Batch size	128
Temporal ensembling	true
Action normalization	$[-1, 1]$

Table 7: Hyperparameters for DP

Backbone	ImageNet-trained ResNet18 [10]
Noise predictor	UNet [29]
Action dimension	8 (3 position + 4 quaternion + 1 gripper)
Cameras	wrist, front, right shoulder, left shoulder
Learning rate	$1e^{-4}$
Weight decay	$1e^{-6}$
Image size	128×128
Observation horizon	1
Action sequence	20
Execution horizon	1
Train, test diffusion steps	50, 50
Hidden dimension	512
Iteration	100000
Batch size	128
Temporal ensembling	true (following ACT’s)
Scheduler	DDPM [12]
Action normalization	[-1, 1]

Table 8: Hyperparameters for Octo

Pretrained model	Octo-small [28]
Action dimension	8 (7 delta joints + 1 gripper)
Cameras	front
Learning rate	$3e^{-4}$
Weight decay	$1e^{-2}$
Image size	256×256
Observation horizon	1
Action sequence	4
Execution horizon	1
Iteration	2000
Batch size	128
Temporal ensembling	false
Action normalization	mean 0, std 1
Finetuning head	linear head
Image augmentation	resized crop, brightness, contrast, saturation, hue

D ACT variant with keyframe action

To further examine the impact of keyframe action on action sequence modeling, we conduct an additional ablation by modifying the ACT baseline. Specifically, we introduce a variant, ACT+KF, in which an extra keyframe action is appended to ACT’s original action chunk.

As shown in Table 9, ACT+KF achieves a higher average success rate (0.516) compared to the original ACT (0.488), indicating that injecting keyframe actions yields marginal improvements. However, the overall gain remains limited.

This result suggests that while keyframe actions may provide some global guidance, they do not substantially improve the final action quality when introduced in this manner. A similar trend is observed in the poor performance of *Hybrid* (Table 2), a variant of CoA that incorporates both keyframe supervision and causal decoding but lacks trajectory continuity. The limited effectiveness of both ACT+KF and Hybrid underscores a key insight: merely injecting keyframe signals and enforcing an autoregressive structure is not sufficient. Instead, it is crucial to model the entire trajectory holistically with temporal continuity, which is explicitly realized in our CoA formulation.

Table 9: Comparison of ACT vs. ACT+KF (with keyframe action) on 10 RLBench tasks.

Task	ACT	ACT+KF
Stack Wine	0.56	0.56
Turn Tap	0.36	0.32
Open Drawer	0.52	0.76
Push Button	0.08	0.16
Pick Up Cup	0.20	0.36
Take Lid	0.40	0.40
Press Switch	0.52	0.28
Reach Target	0.88	0.72
Sweep Dust	1.00	0.96
Open Box	0.36	0.64
Avg.	0.488	0.516