MVISTA-4D

View-Consistent 4D World Model with Test-Time Action Inference for Robotic Manipulation

Jiaxu Wang*, Yicheng Jiang*, Tianlun He, Jingkai Sun, Qiang Zhang

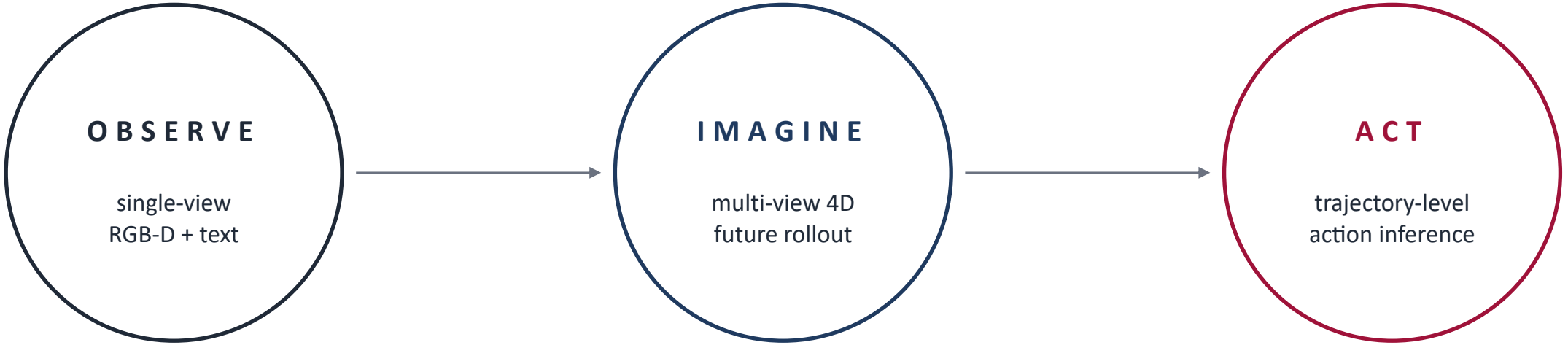
Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao†, Xiangyu Yue†

CUHK MMLab · HKUST · HKU · X-Humanoid · Tsinghua

ICML 2026

Imagine, then act.

Instead of mapping observations directly to actions, forecast a future and pick actions that realize it.



$$\hat{o}_{1:T} = p_\vartheta(o_o, l)$$

PROBLEM

Existing world models cannot predict complete 4D scenes.

2D Video

appearance only

— no geometry

3D Point Models

sparse geometry

— weak semantics

Single-View 4D

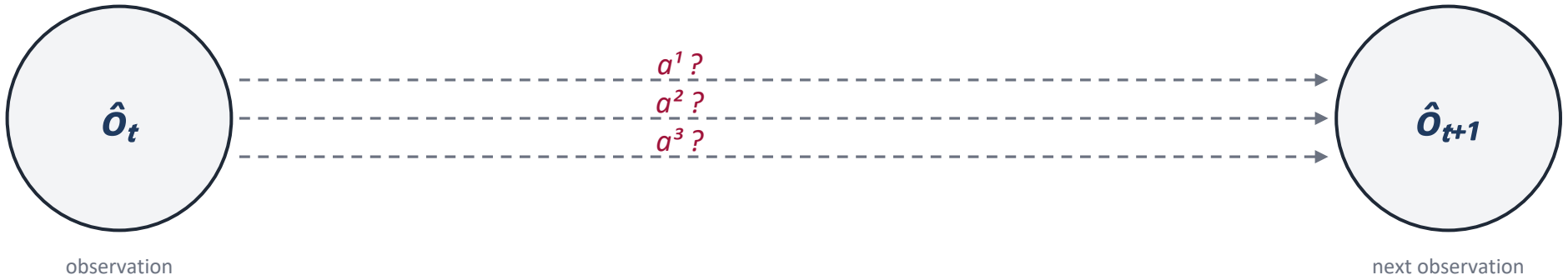
RGB-D forecast

— incomplete under occlusion

Need: strong visual priors + geometry-consistent, multi-view 4D predictions.

And turning imagination into actions is ill-posed.

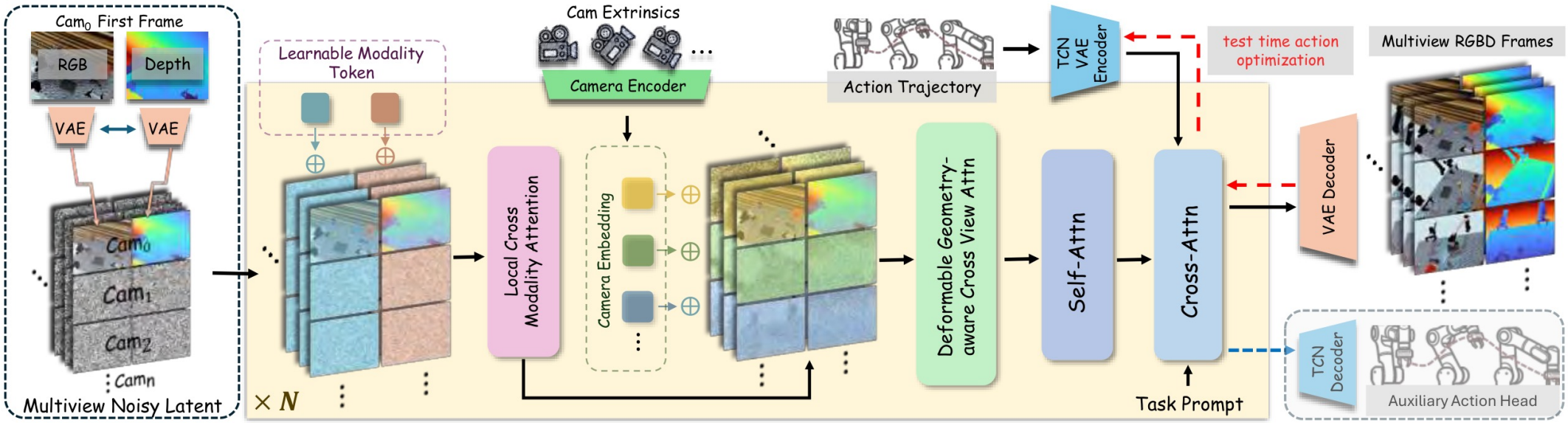
Many actions can explain the same transition.



Our remedy:

encode the entire trajectory as a low-dim latent and recover it via backprop.

A trajectory-conditioned multi-view 4D world model.



01

Cross-view + cross-modal fusion

02

Trajectory latent style code

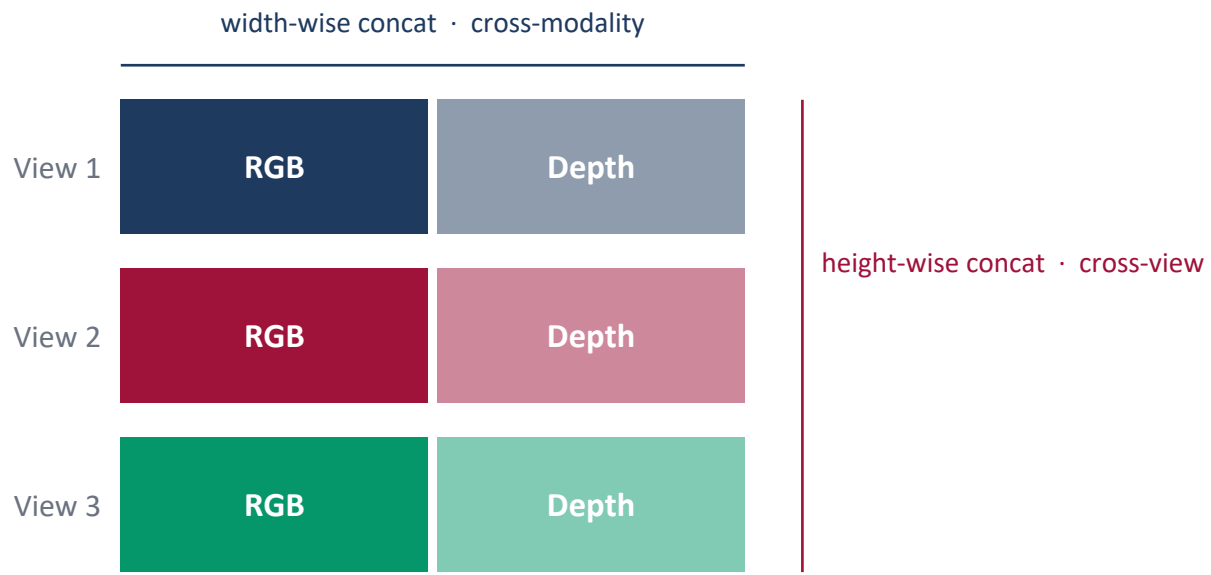
03

Test-time action inference

04

Residual inverse dynamics

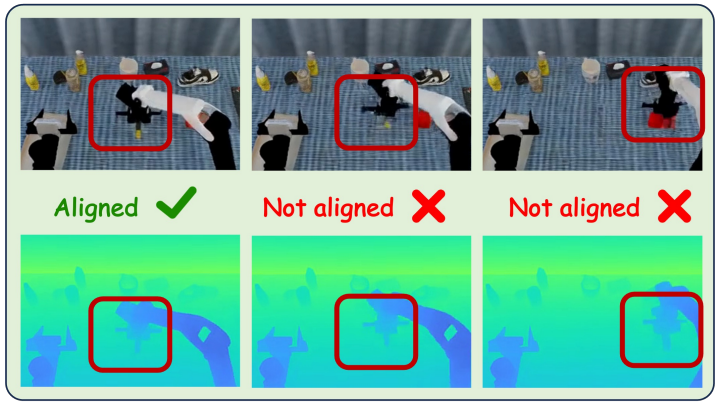
Structured tokenization for multi-view RGB-D.



Reduces token distance between key dependencies — Transformers model correlations more effectively.

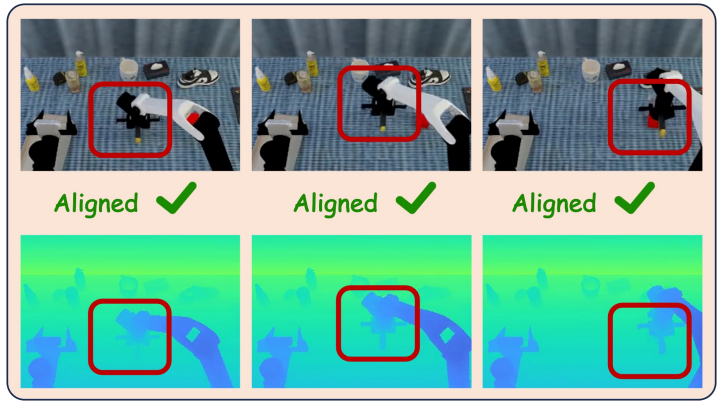
Local attention aligns RGB and depth.

Without modality fusion



(a) Without cross modality modeling

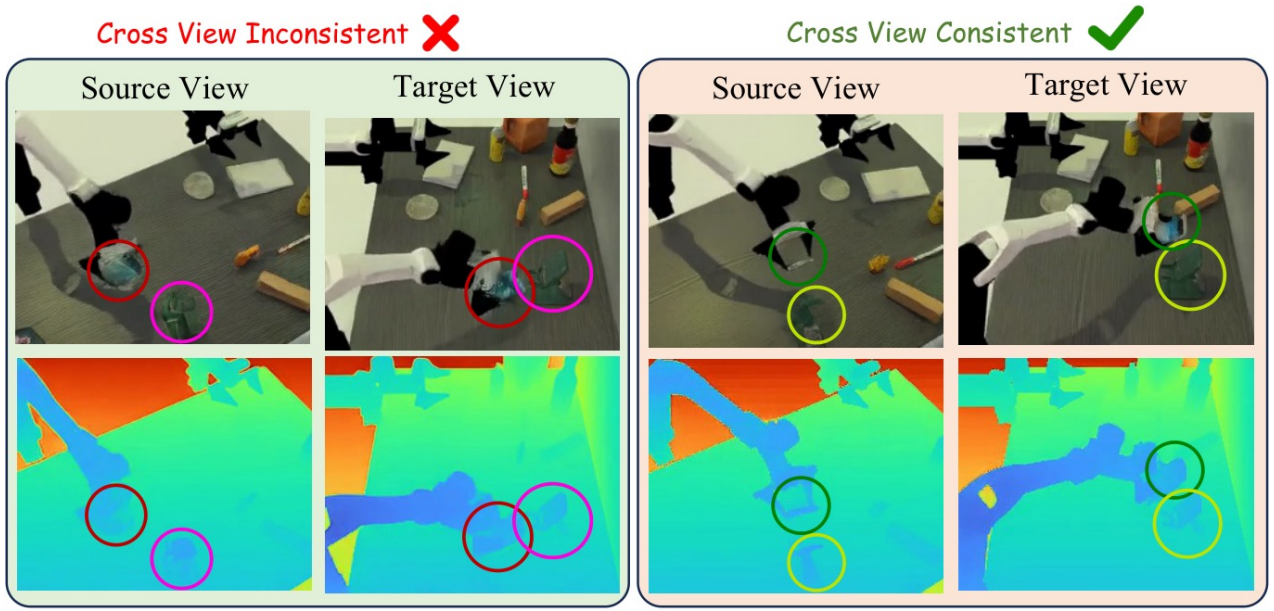
Ours



(b) Ours

Predicted RGB stays aligned with depth boundaries across views and time.

Geometry-aware deformable cross-view attention.



(a) Without Cross View Modeling

(b) Ours Cross View Modeling

spherical camera embedding

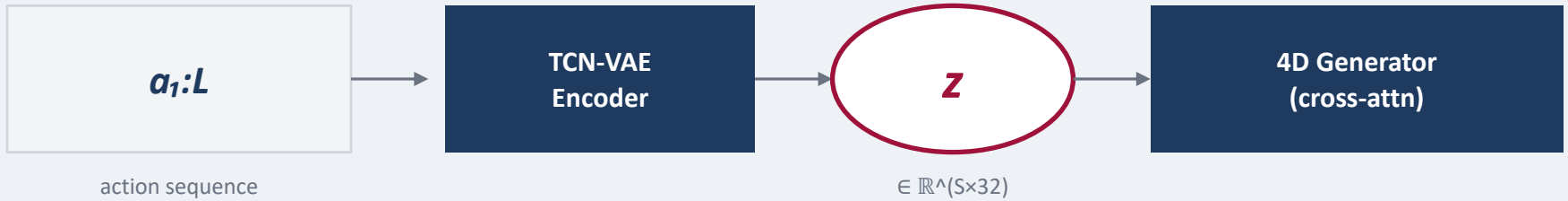
epipolar sparse sampling

deformable refinement

Sparse epipolar candidates + small offsets → geometry-aligned cross-view features.

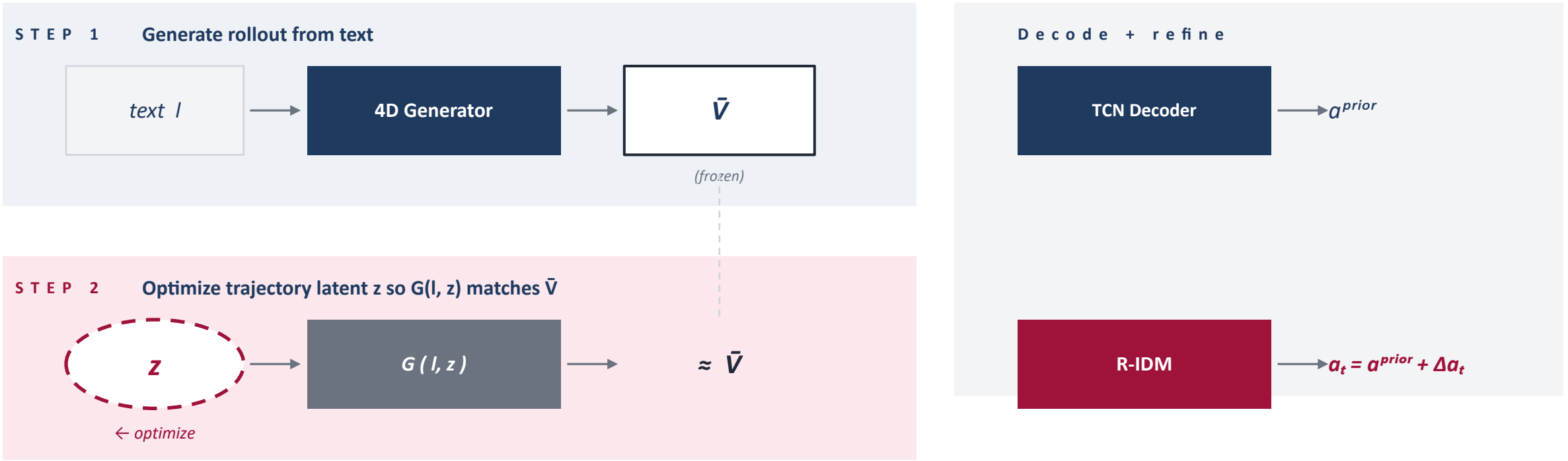
Encode the action sequence into a style code.

Trajectory acts as a style signal — temporal rhythm, smoothness, coarse phase.



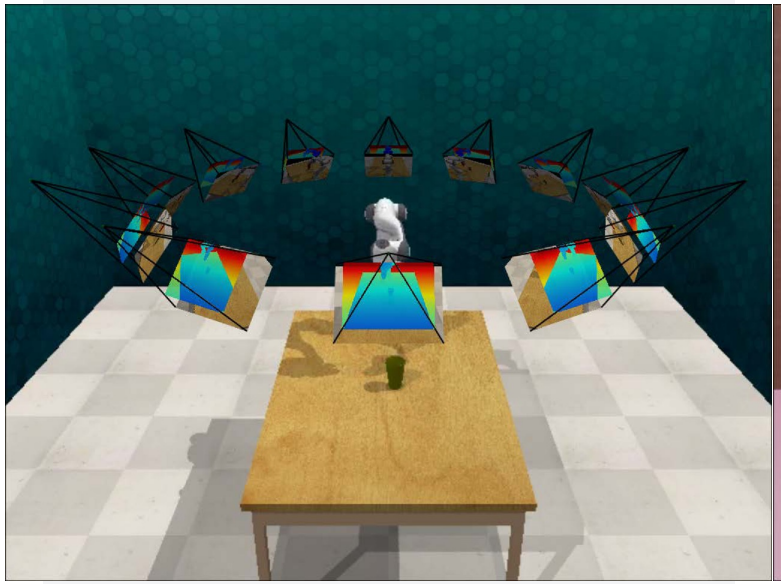
$\mathcal{L}_{traj} = ||\hat{z} - z||^2$ *(latent-consistency, training only)*

Recover actions via test-time optimization.



$$z^* = \arg\min \; D(G(l, z) , \bar{V}) + \lambda \, ||z||^2$$

Datasets and setups.

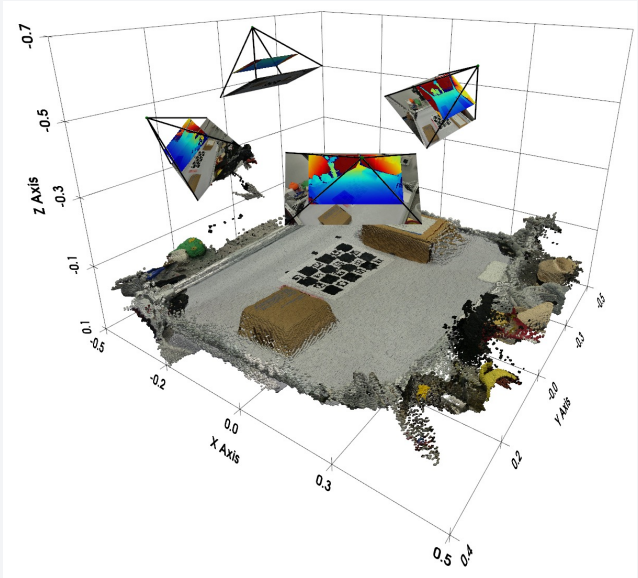


RLBench

10 tasks · 8K+ trajs · 12 cams · 320×240

RoboTwin 2

10 tasks · 10K+ trajs · 12 cams · 320×240



Real Robot

14 tasks · 4 RGB-D cams · tele-op

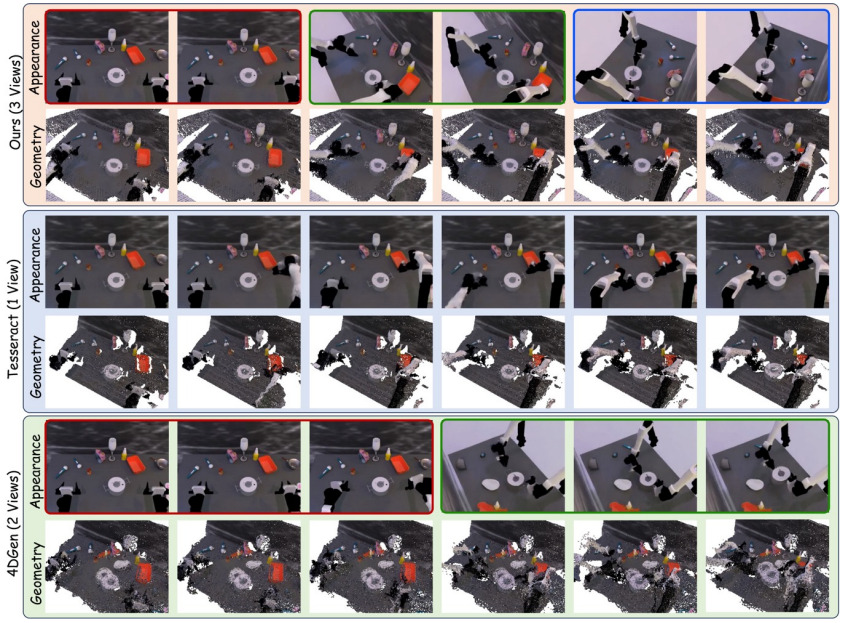
Metrics · PSNR / SSIM / FVD · AbRel / RMSE / δ_1 · CD / EMD · Success Rate

4D generation — quantitative.

Method	PSNR↑	SSIM↑	FVD↓	AbRel↓	RMSE↓	δ_1↑	CD↓	EMD↓
RLBench								
UniPi*	23.88	91.7	19.94	117.9	43.2	91.6	15.0	20.6
4DGen	22.25	87.1	20.51	91.8	29.3	94.1	10.9	16.0
TesserAct	23.86	92.8	27.77	91.8	29.4	96.8	11.0	16.3
Ours	**23.31**	**90.8**	**18.57**	**90.5**	**29.1**	**97.1**	**9.6**	**15.3**
RoboTwin								
UniPi*	22.98	89.2	22.18	5.52	18.13	95.1	9.88	20.53
4DGen	22.18	85.2	24.61	3.00	13.90	96.6	7.18	10.62
TesserAct	22.65	89.8	27.29	3.71	15.07	97.3	7.11	10.28
Ours	**22.91**	**90.2**	**21.93**	**2.60**	**12.30**	**97.4**	**6.51**	**9.90**
Real-World								
UniPi*	22.53	90.62	28.62	39.95	42.69	67.55	58.41	63.22
4DGen	21.34	89.75	25.60	23.36	29.61	79.59	17.32	15.61
TesserAct	22.27	91.50	50.79	30.56	33.17	34.16	38.47	34.65
Ours	**21.82**	**89.98**	**23.08**	**20.79**	**25.11**	**82.18**	**13.06**	**14.37**

Best geometry across all three datasets · best FVD on RoboTwin and Real-World.

Multi-view 4D generation comparison.

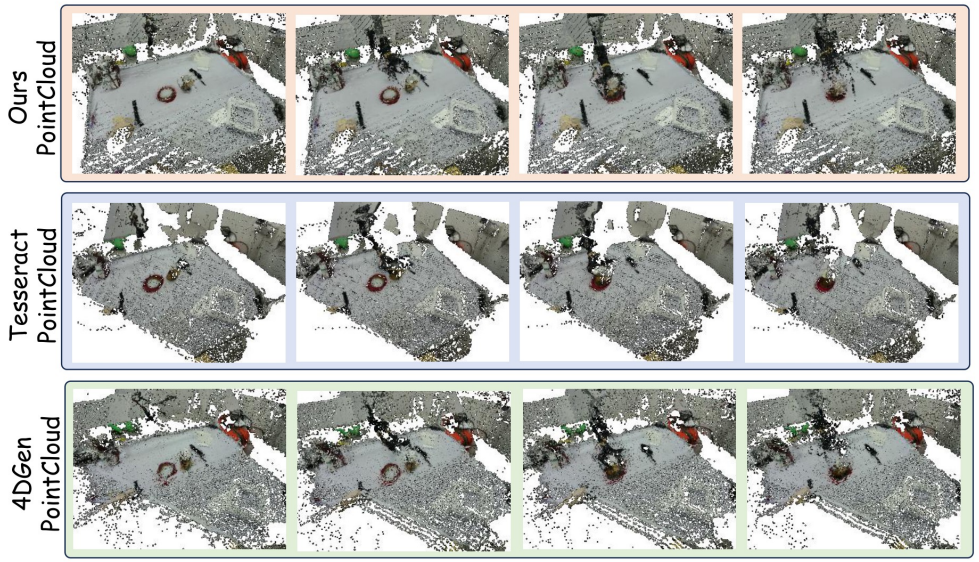


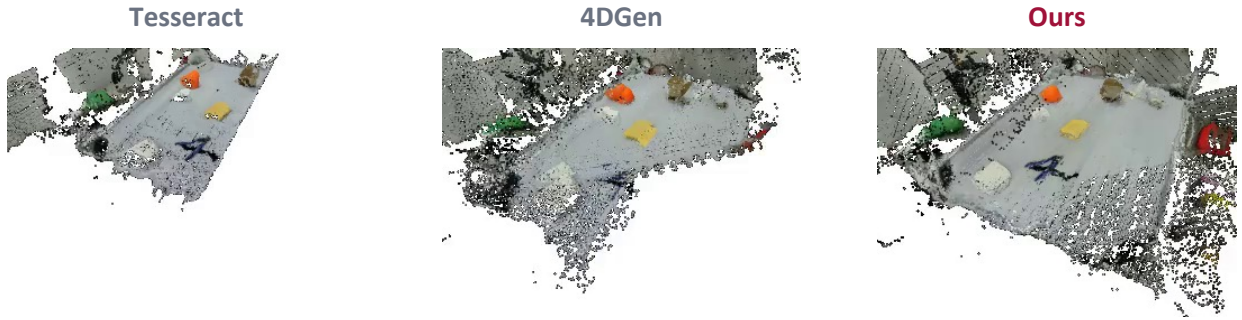
Tesseract

4DGen

Ours

Generated geometry on our real-robot dataset.



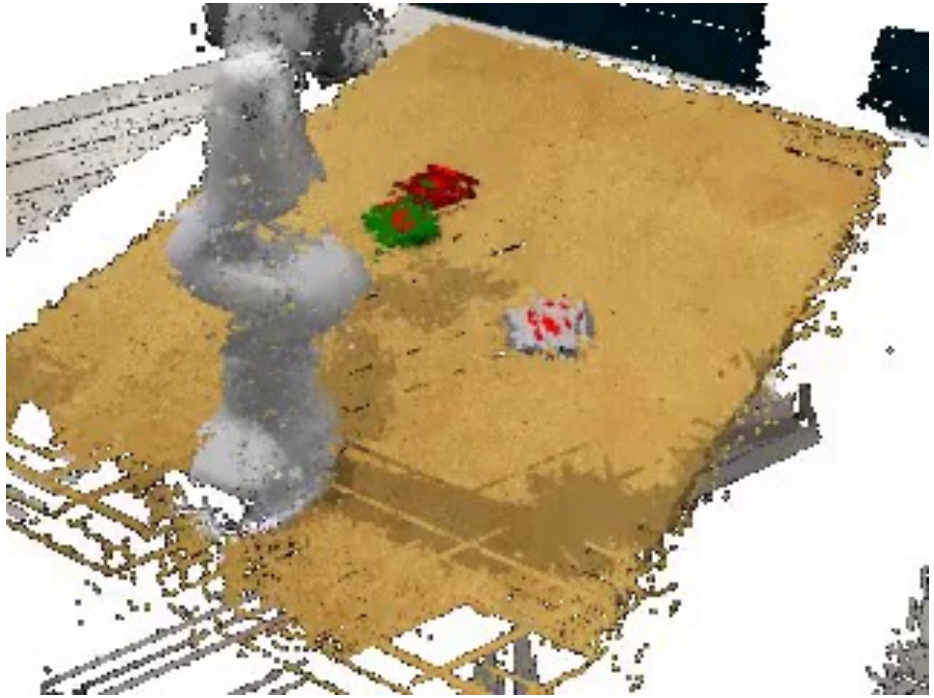


Multi-step manipulation rollout.

Push maroon → push green → push gray.

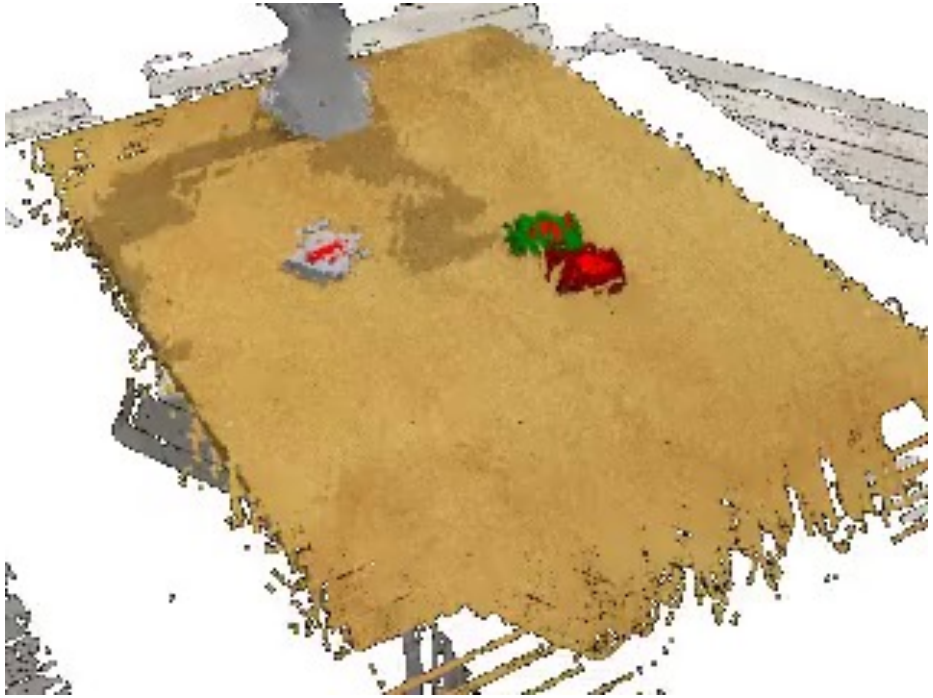
4DGen

baseline



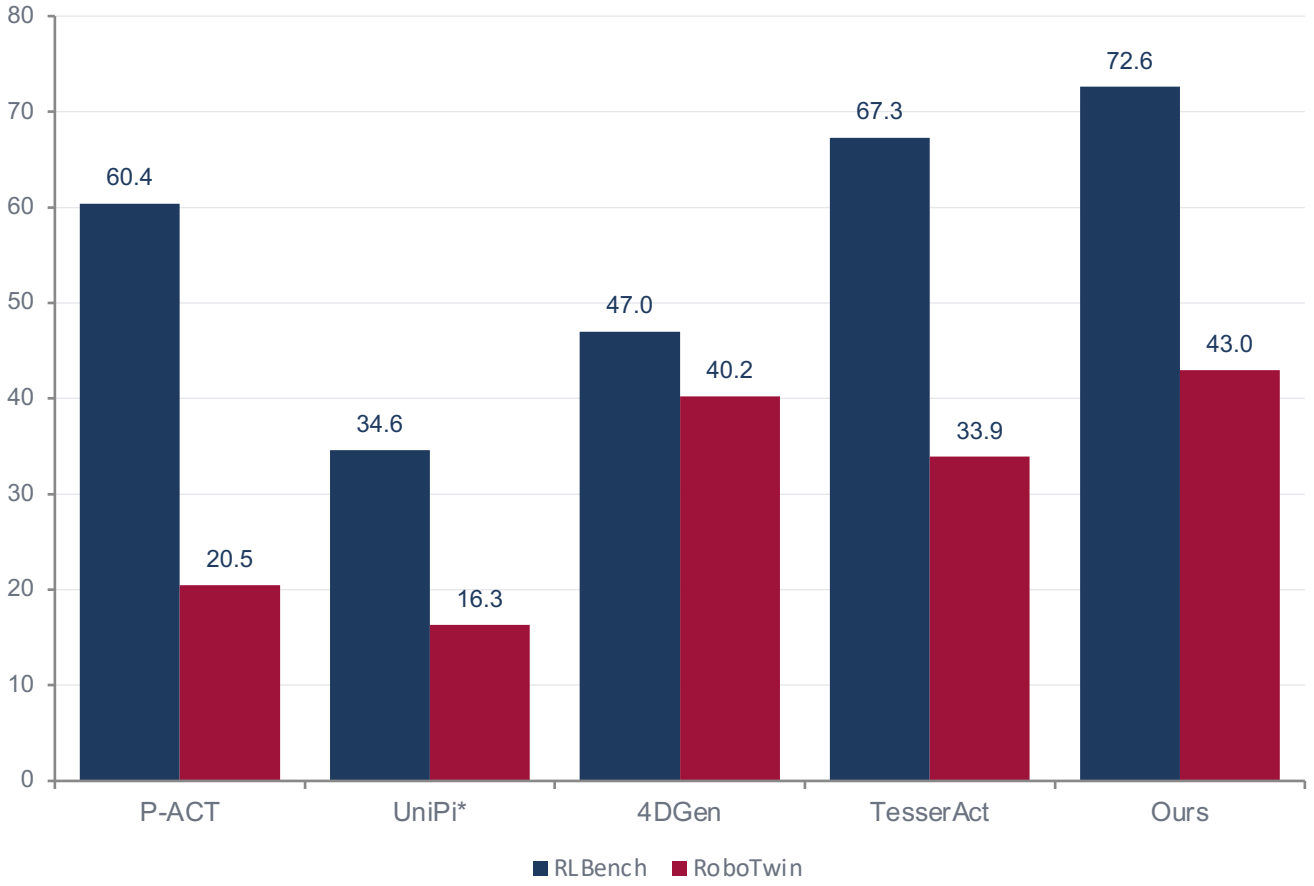
Ours

multi-view 4D + R-IDM



Denser, better-aligned point clouds during multi-step sequencing.

Success rates across platforms.



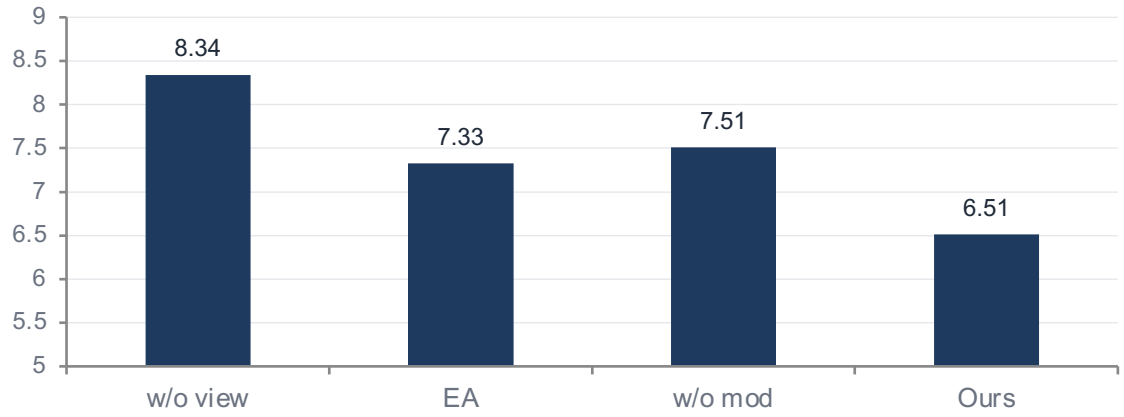
Real robot · per-task SR (%)

Task	TesserAct	Ours
Arrange Boxes	7	**15**
Cap Bottle	27	**33**
Open Drawer	37	**56**
Place Fruits	17	**23**
Put Orange	66	63
Stack Cubes	45	**50**

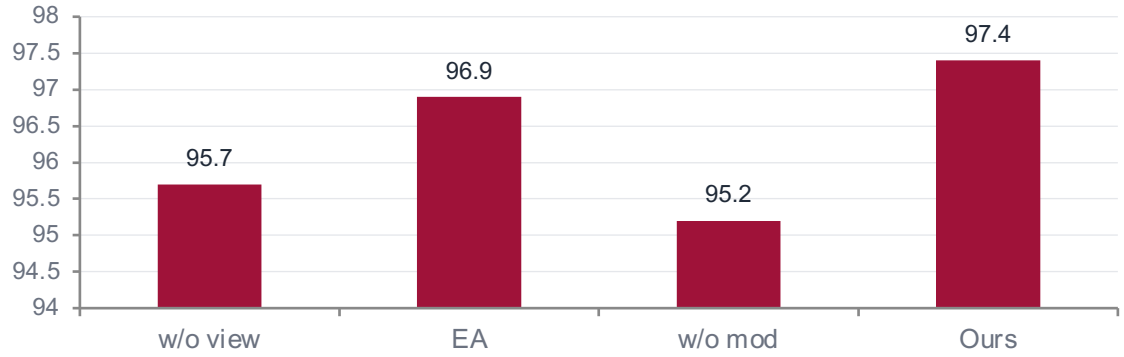
Multi-view geometry + trajectory-latent optimization + R-IDM.

Each design choice carries its weight.

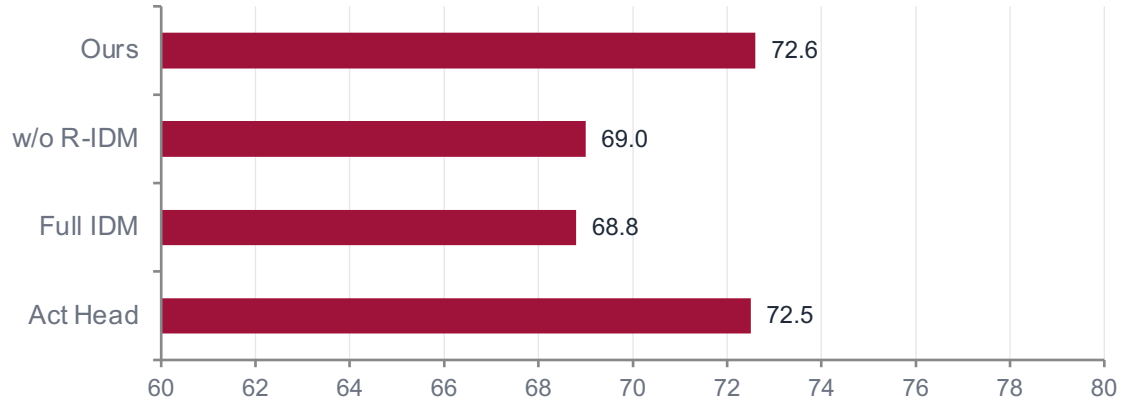
Chamfer ↓ (RoboTwin)



δ_1 ↑ (RoboTwin)



Action inference · RLBench SR (%)



Manipulation fusion ablation · SR (%)

Dataset	w/o view	w/o mod	cat depth	w/o TCN-VAE	Full
RLBench	68.8	67.2	66.1	68.1	**72.6**
RoboTwin	38.0	38.6	39.4	41.5	**43.0**

Limitations.

1 Test-time optimization latency

Iterative latent optimization needs multiple gradient steps at inference, raising latency and reducing real-time responsiveness for high-frequency control loops.

2 Calibration sensitivity

The method relies on accurate camera extrinsics and robot kinematics — changes in camera pose or robot configuration can degrade multi-view consistency and action quality.

3 Two-stage inference

Rollout generation and trajectory optimization are decoupled — errors in the imagined future propagate into action inference without a global feedback loop.

Future work

higher-resolution geometry · contact-aware constraints · closed-loop correction · faster amortized inference

Conclusion

- Embodied multi-view 4D generative world model with explicit cross-modality and cross-view fusion
- Trajectory-level action conditioning via compact latent codes enables test-time optimization
- Residual inverse dynamics for reliable action refinement around a strong trajectory prior
- Strong performance on 34 manipulation tasks across simulation and real robots

Thank you — Questions & Discussion

mercerai.github.io/MVISTA-4D/