Perception Stitching: Zero-Shot Perception Encoder Transfer for Visuomotor Robot Policies

Pingcheng Jian[1] **Easop Lee**[1] **Zachary Bell**[2] **Michael M. Zavlanos**[1] **Boyuan Chen**[1]

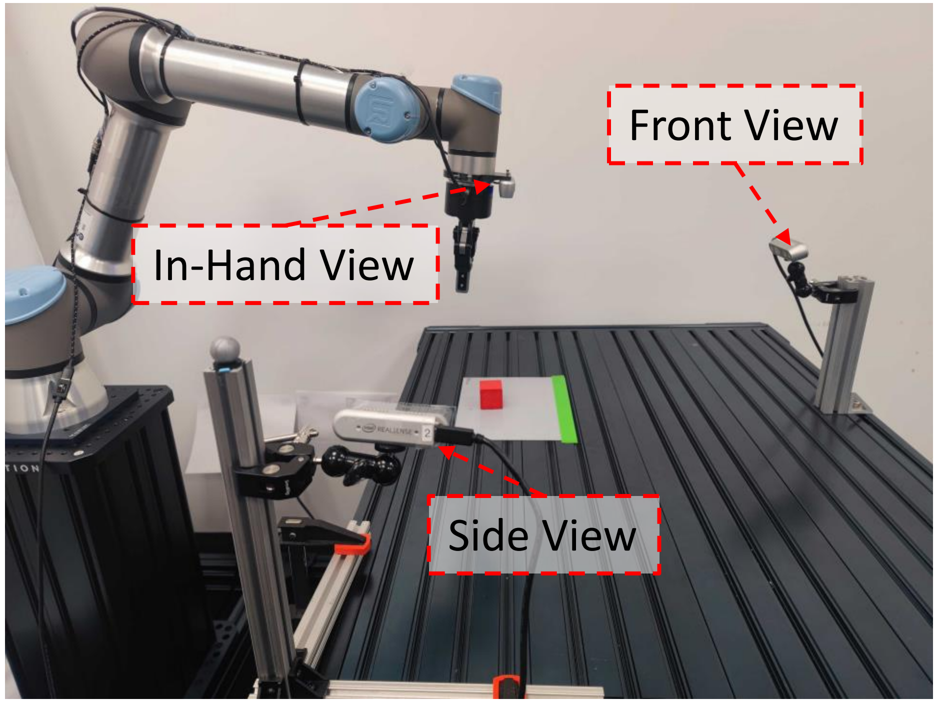
[1]Duke University [2]Air Force Research Laboratory

Duke UNIVERSITY

AFRL THE AIR FORCE RESEARCH LABORATORY

Motivation

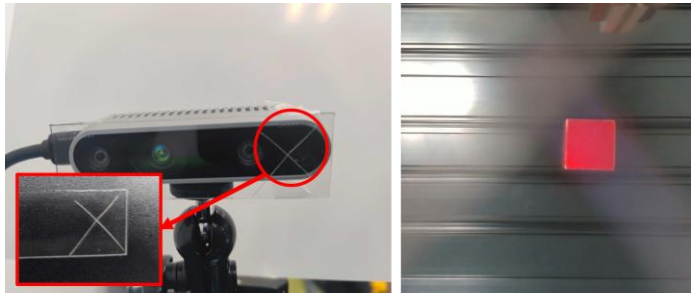
- How to share the learned knowledge of the same task under different visual observations?




Fisheye Camera

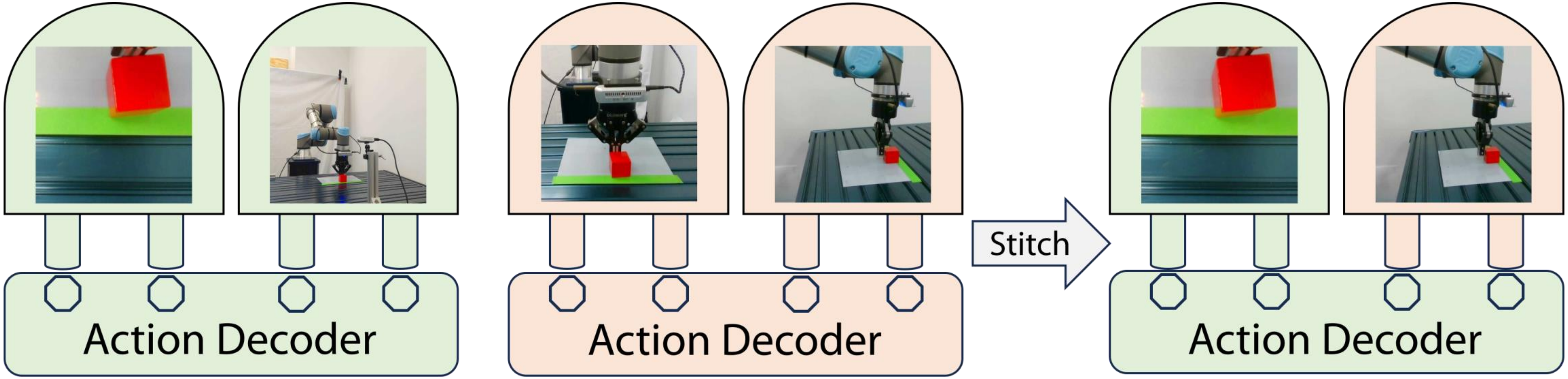

Masked Lens


RGBD Camera


Broken Lens

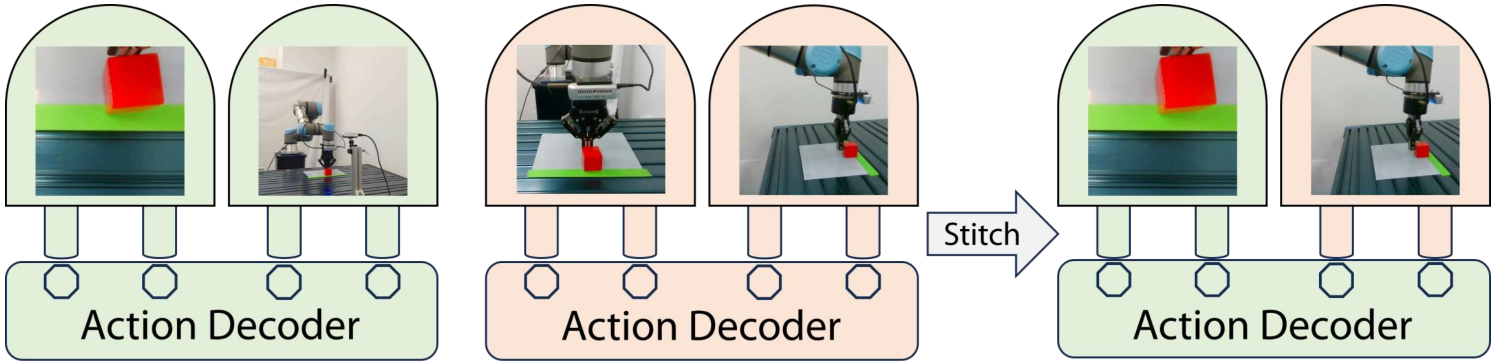
Motivation

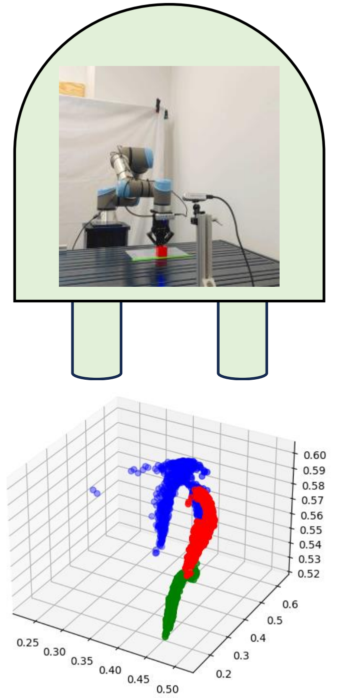
- Directly stitch the perception encoder to another visuomotor policy.
- Zero-shot transfer of the trained visuomotor policies to a novel combination of perceptual configurations.



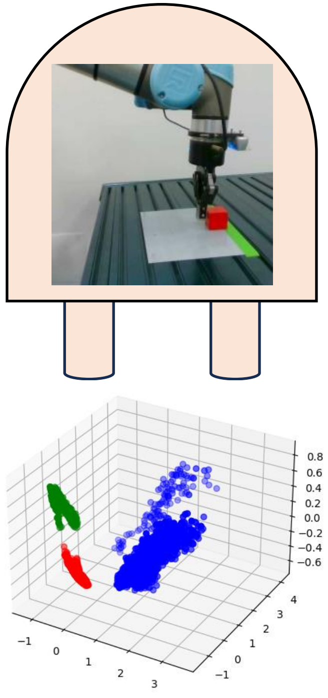
Challenge

- How to align the latent representations of different visual encoders?





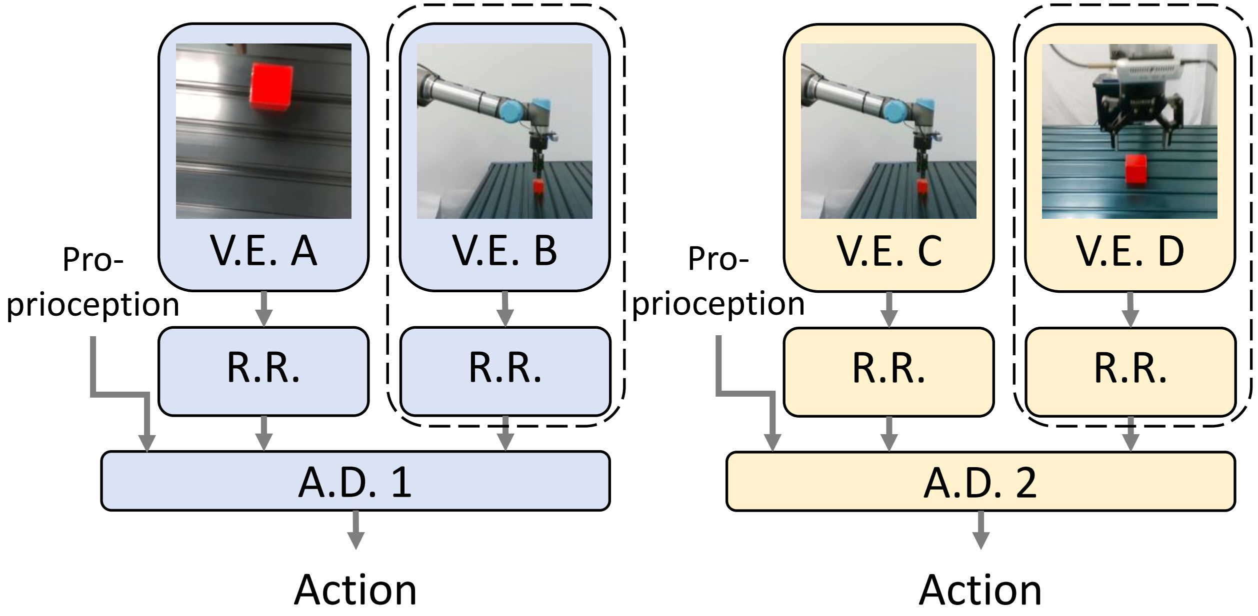
Latent Representation



Latent Representation

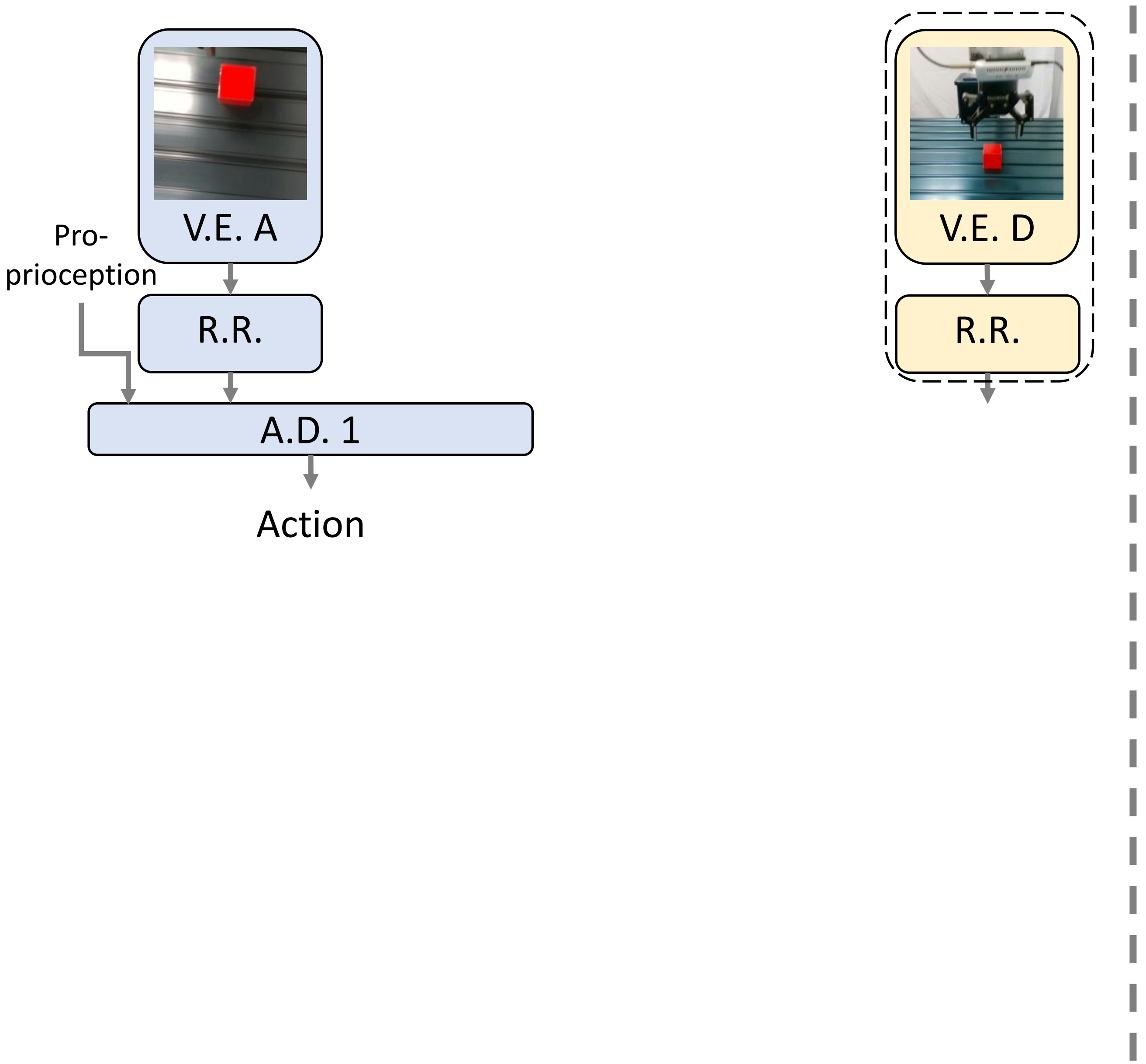
Method

V.E. : Visual Encoder A.D. : Action Decoder R.R.: Relative Representation



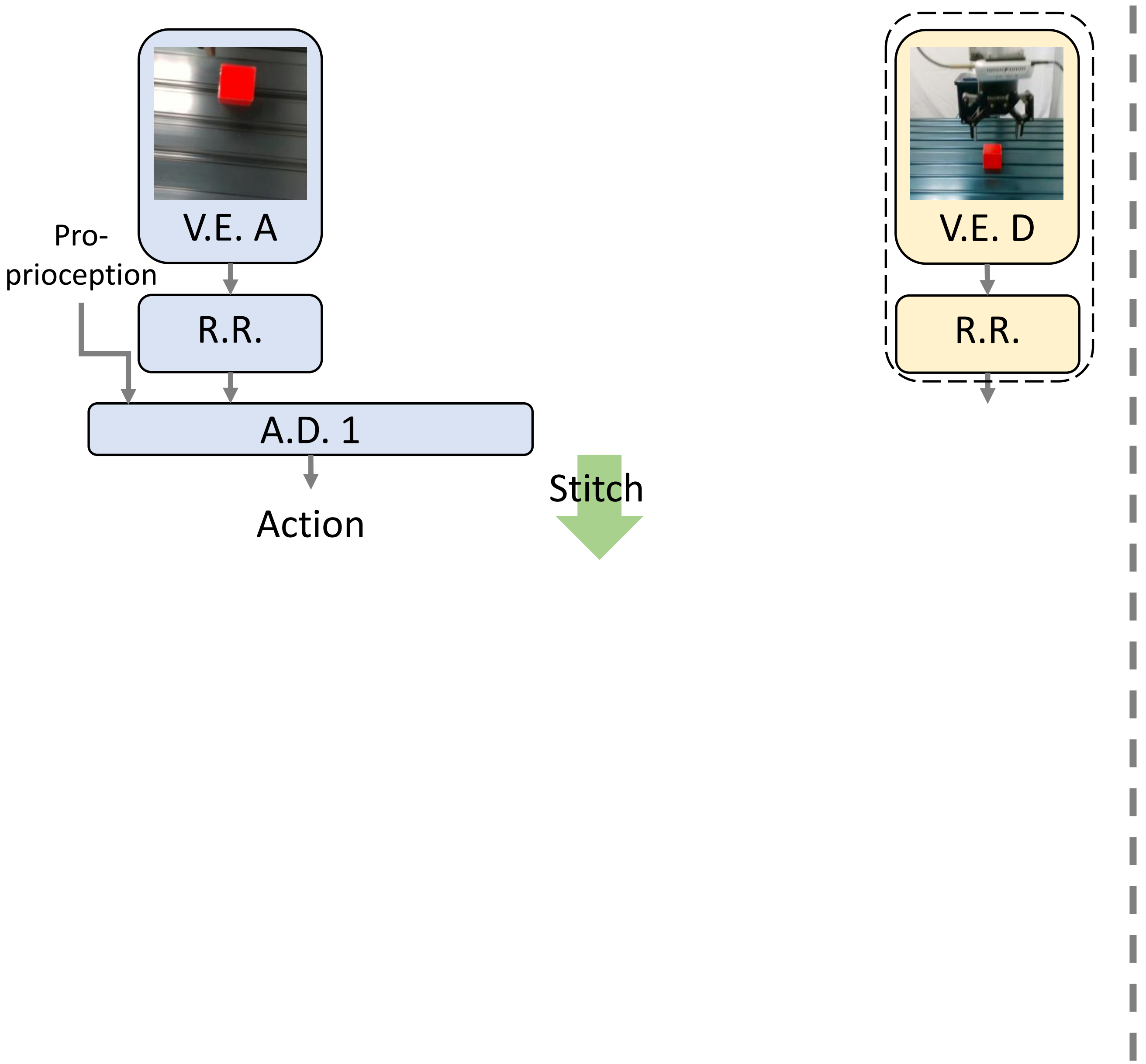
Method

V.E. : Visual Encoder   A.D. : Action Decoder   R.R.: Relative Representation



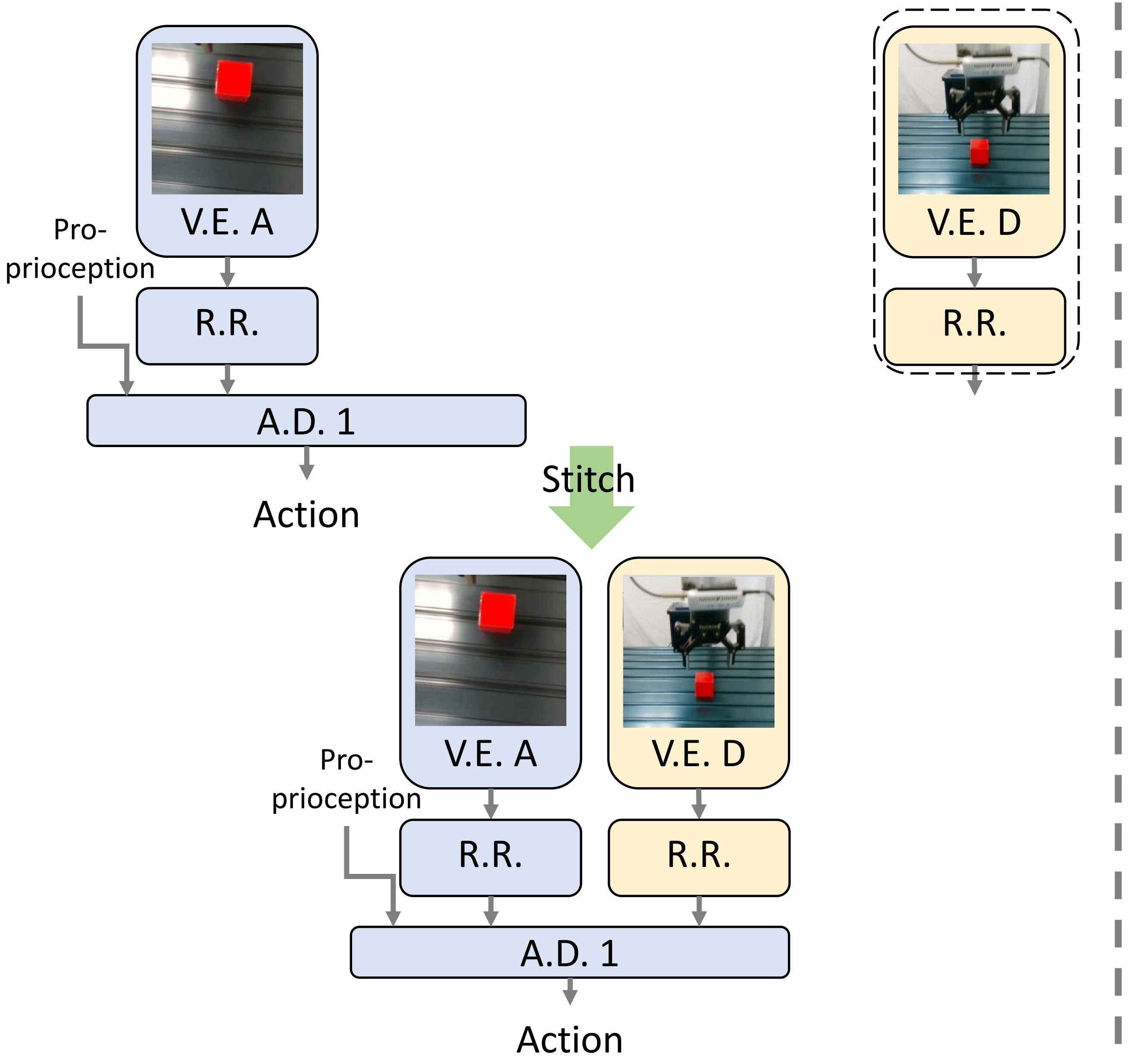
Method

V.E. : Visual Encoder   A.D. : Action Decoder   R.R.: Relative Representation



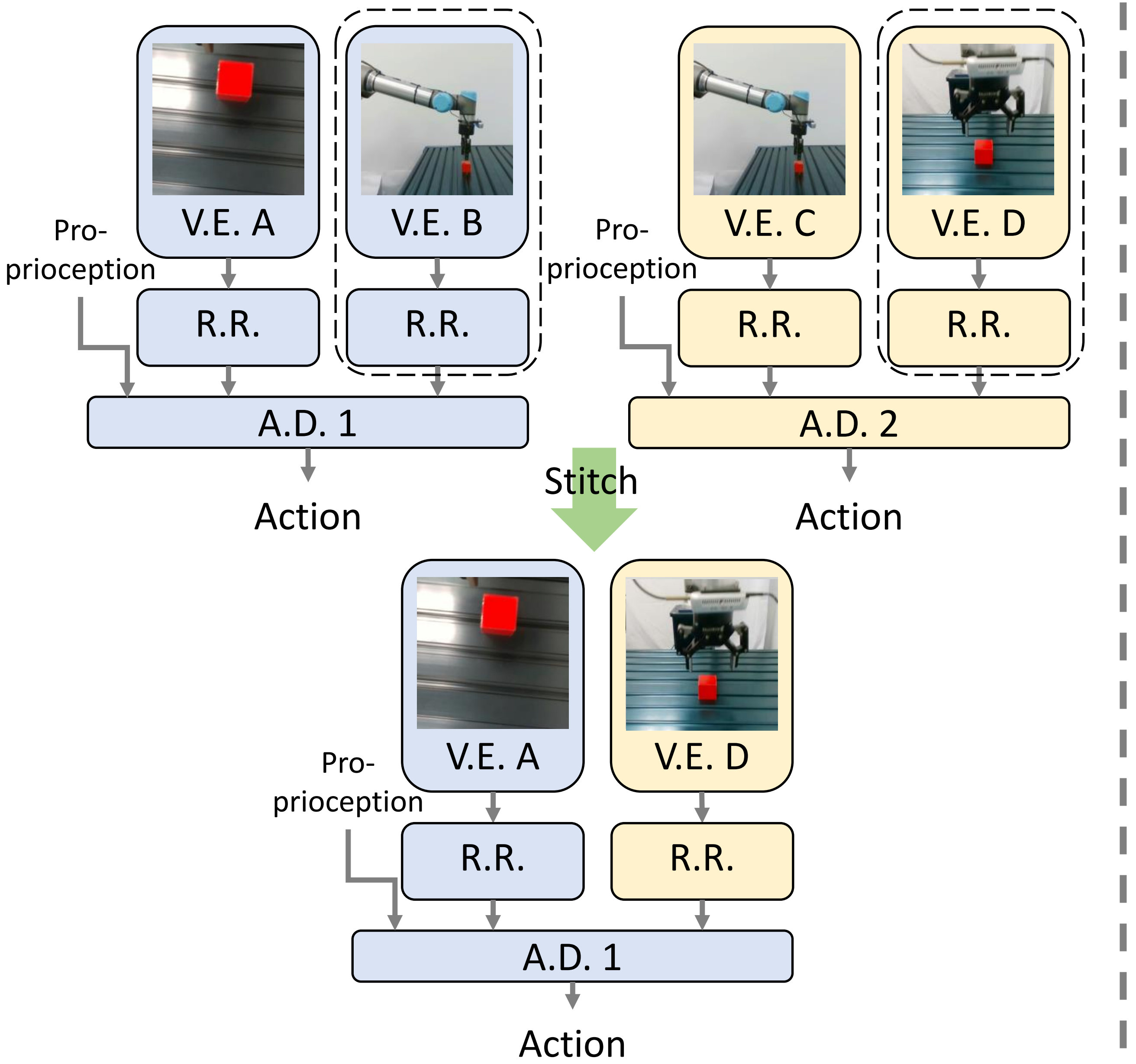
Method

V.E. : Visual Encoder   A.D. : Action Decoder   R.R.: Relative Representation

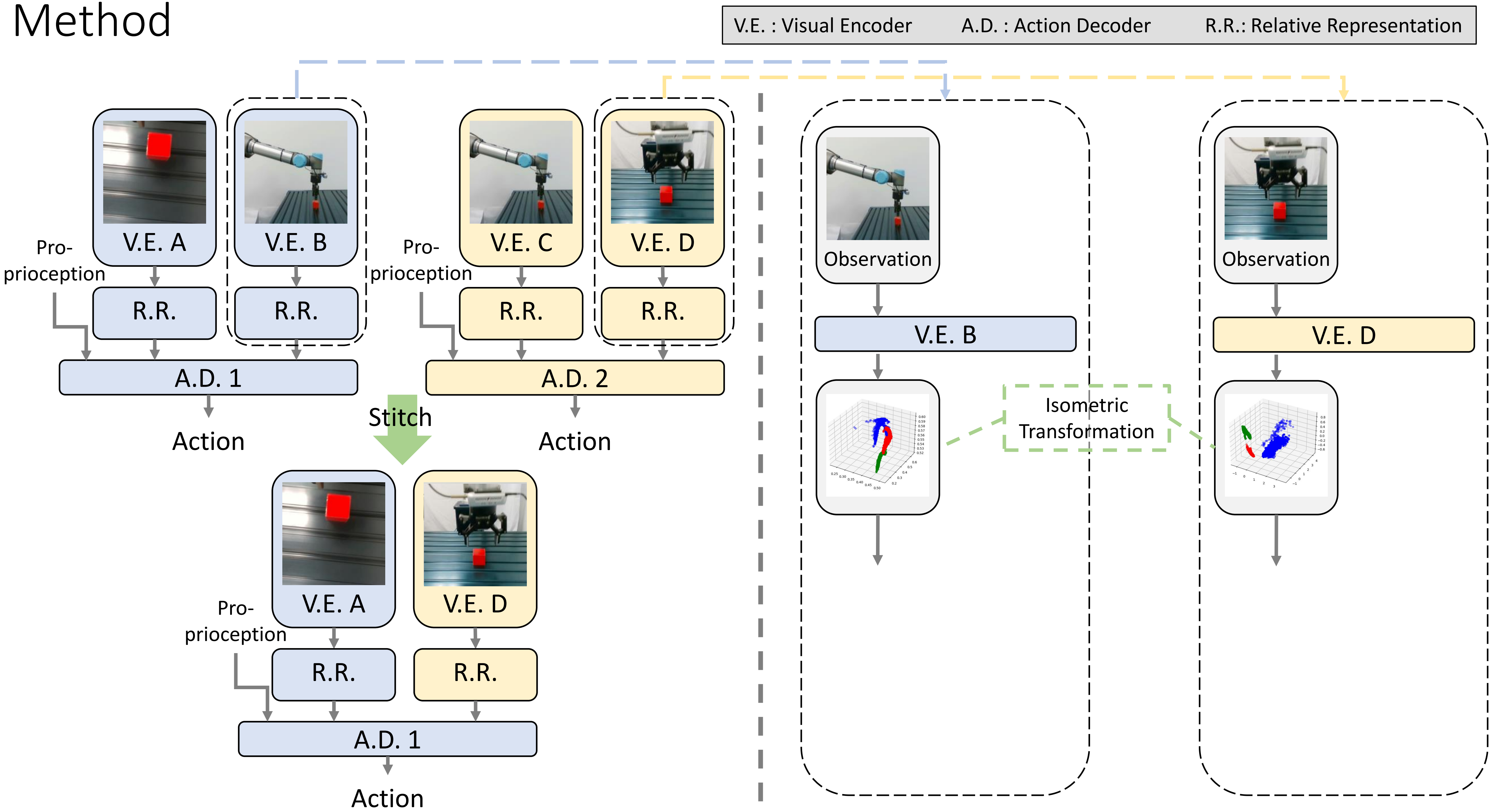


Method

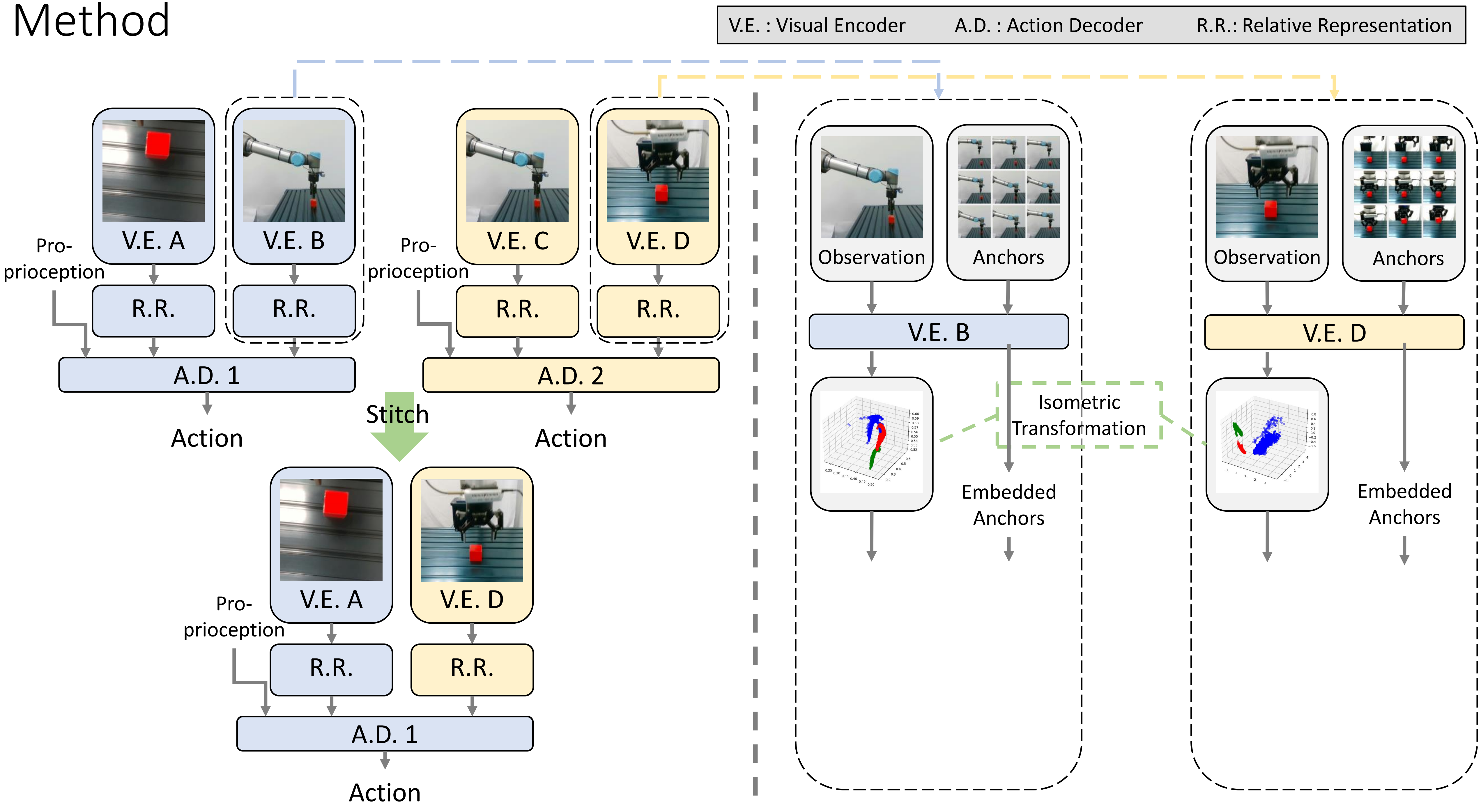
V.E. : Visual Encoder   A.D. : Action Decoder   R.R.: Relative Representation



Method


V.E. : Visual Encoder
A.D. : Action Decoder
R.R.: Relative Representation
V.E. A
V.E. B
V.E. C
V.E. D
Pro-
prioception
R.R.
A.D. 1
A.D. 2
Action
Stitch
Observation
Isometric
Transformation

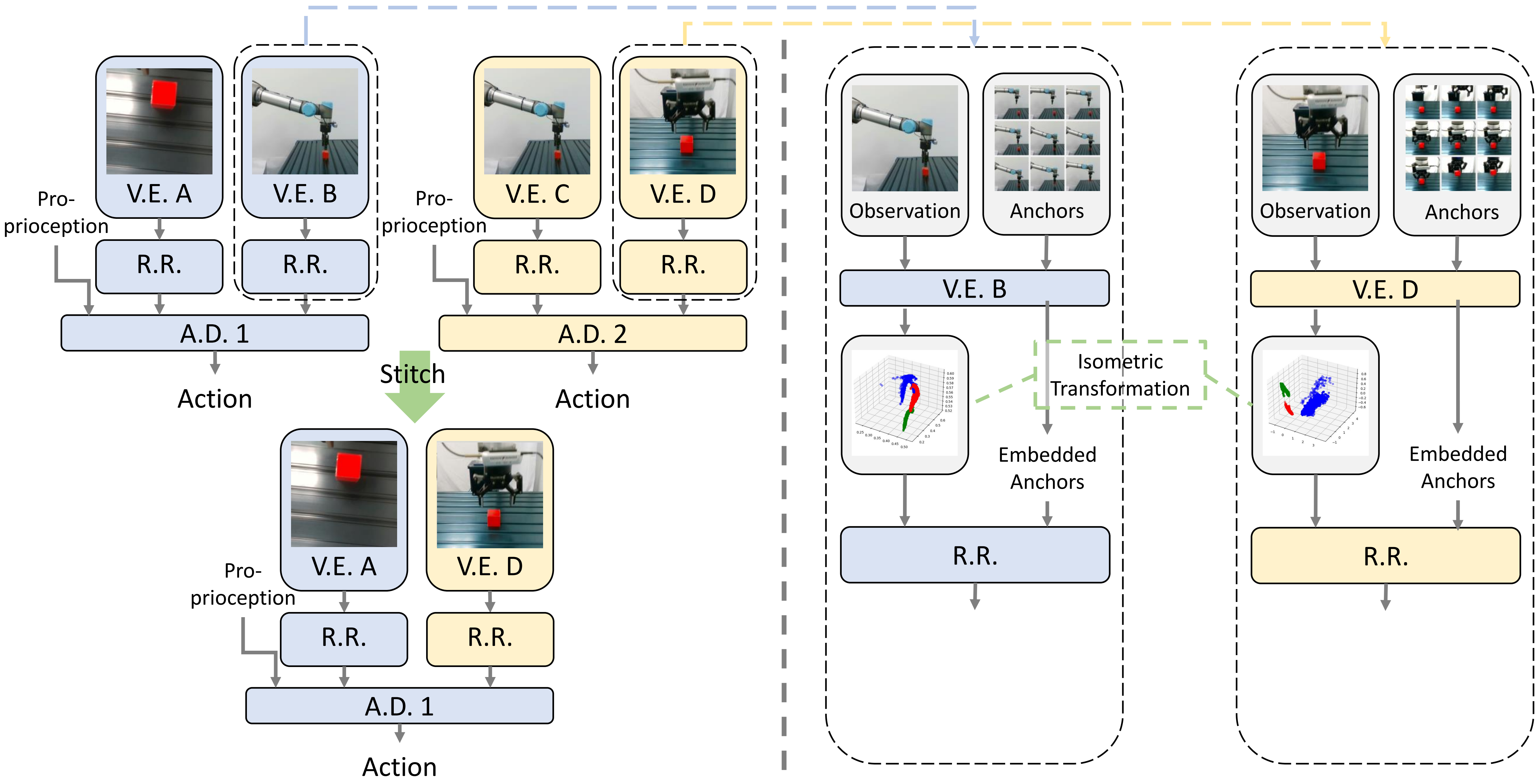
Method



V.E. : Visual Encoder
A.D. : Action Decoder
R.R.: Relative Representation
V.E. A
V.E. B
V.E. C
V.E. D
Pro-
prioception
R.R.
A.D. 1
A.D. 2
Action
Stitch
Observation
Anchors
Isometric
Transformation
Embedded
Anchors

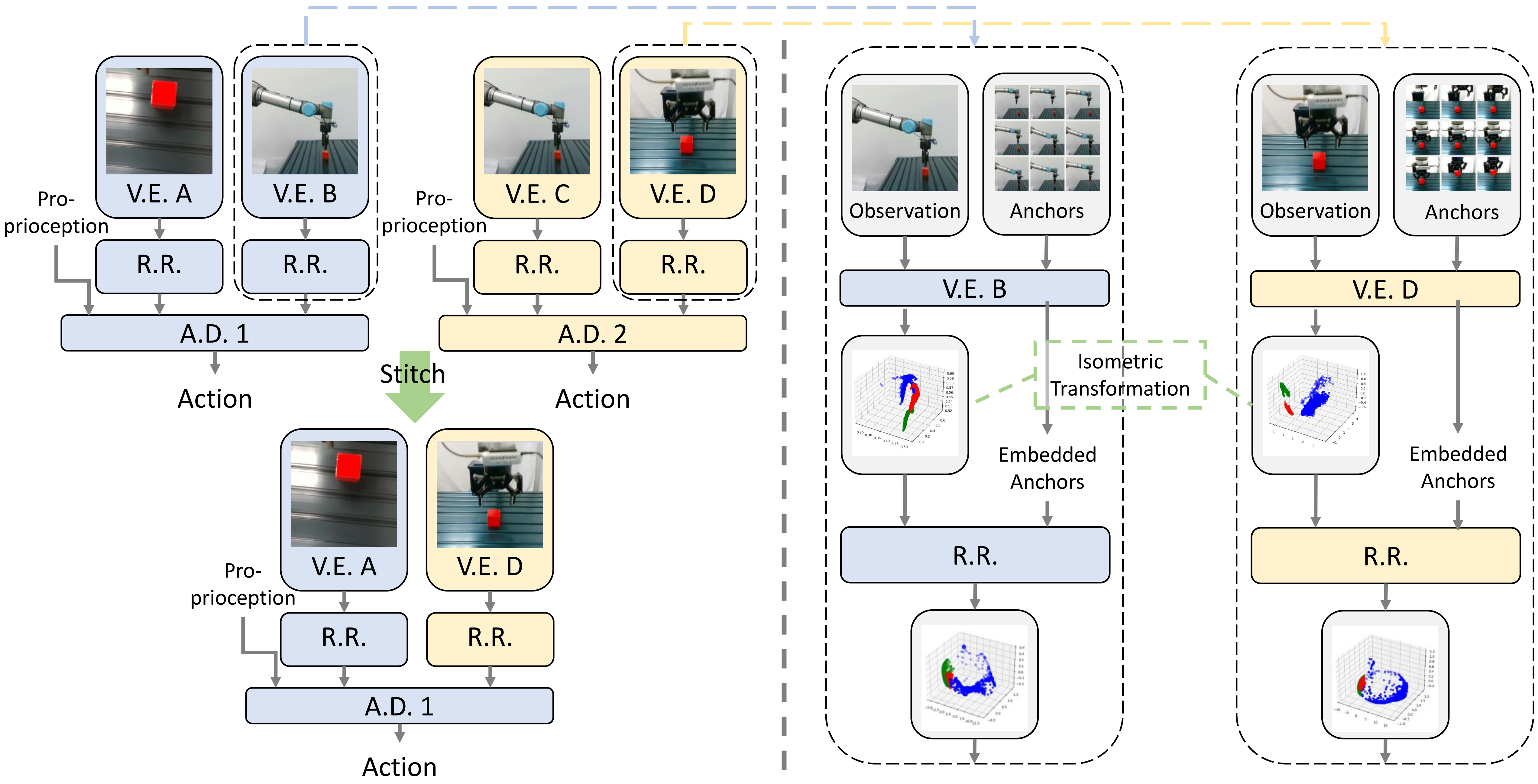
Method

V.E. : Visual Encoder   A.D. : Action Decoder   R.R.: Relative Representation

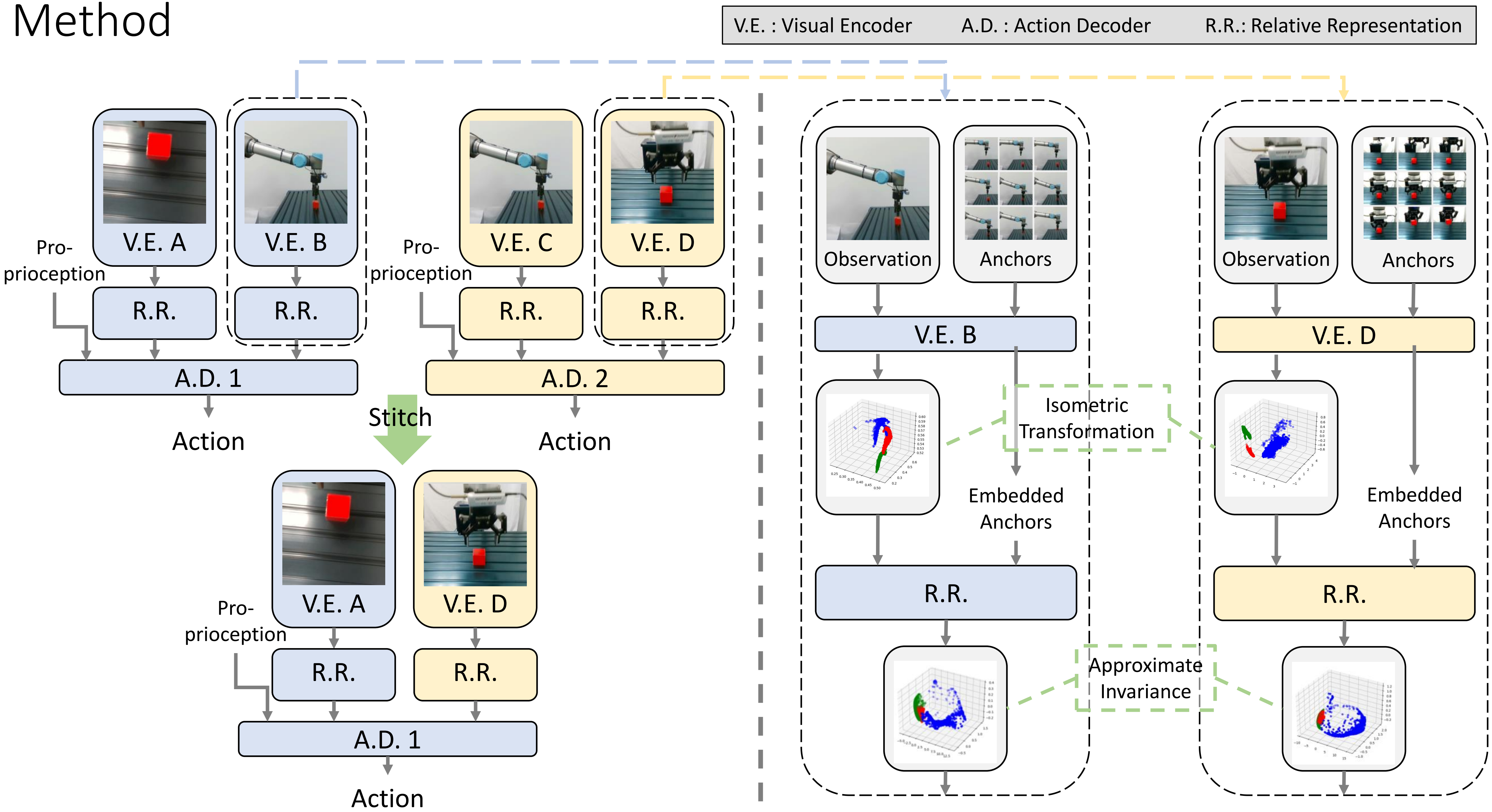


Method

V.E. : Visual Encoder A.D. : Action Decoder R.R.: Relative Representation

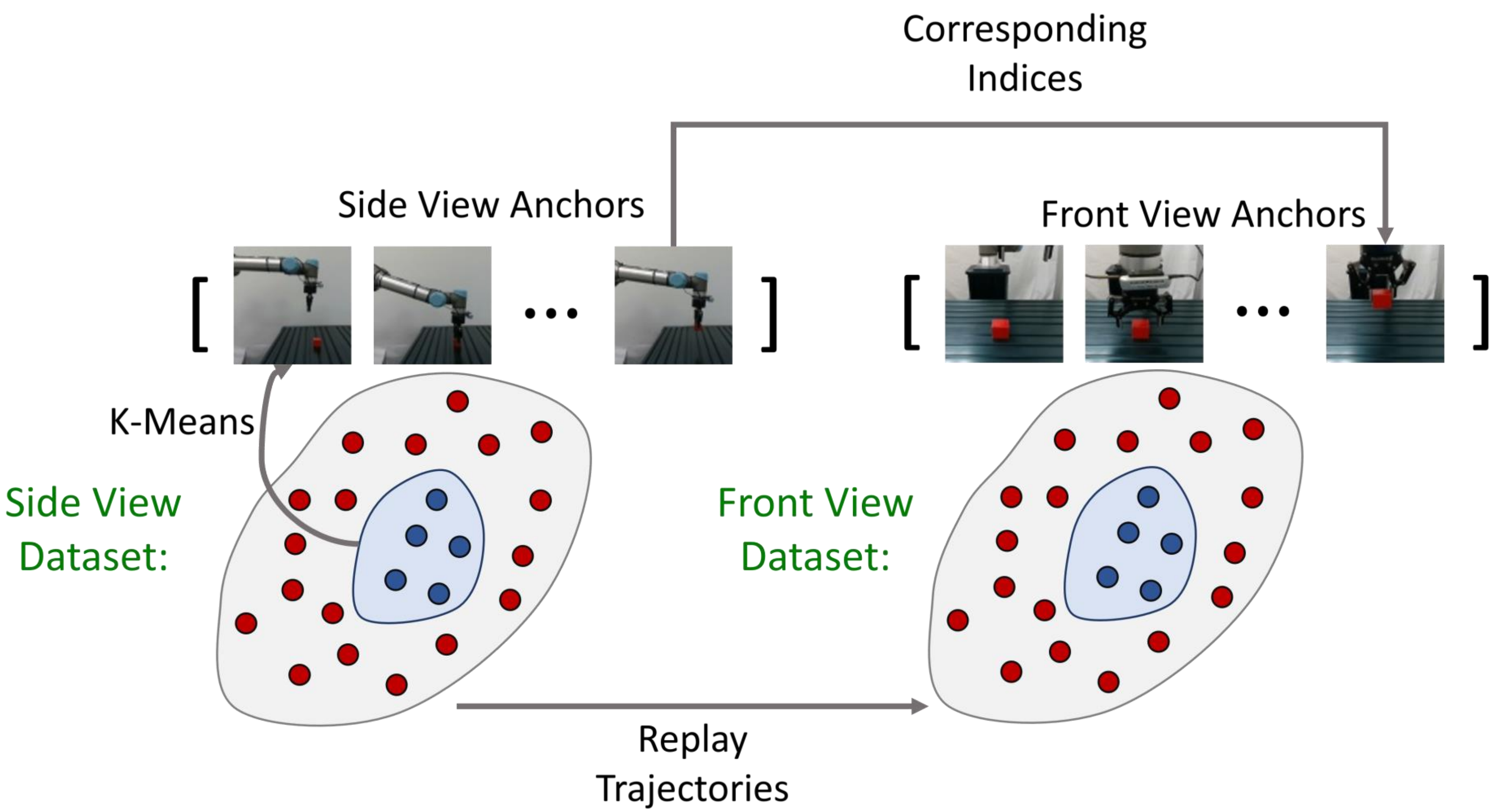


Method

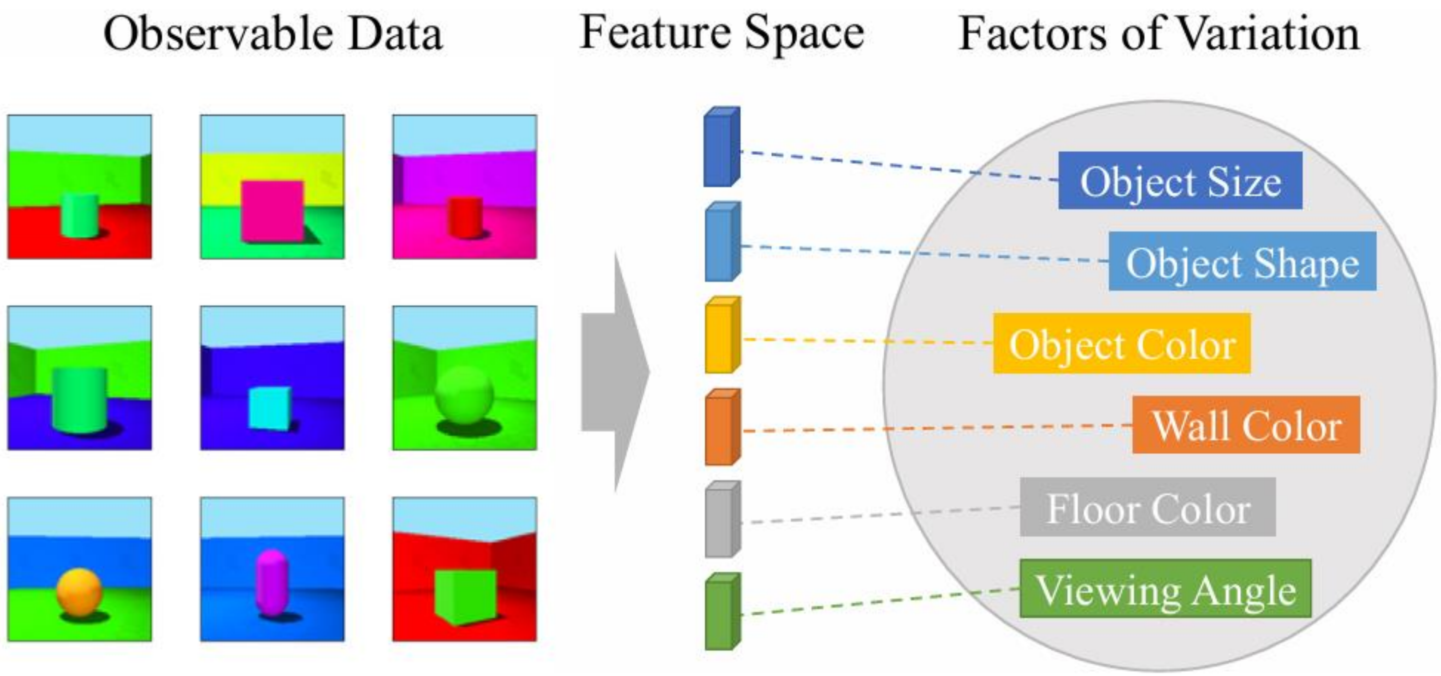


V.E. : Visual Encoder
A.D. : Action Decoder
R.R.: Relative Representation
V.E. A
V.E. B
V.E. C
V.E. D
Pro-
prioception
R.R.
A.D. 1
A.D. 2
Action
Stitch
Observation
Anchors
Isometric
Transformation
Embedded
Anchors
Approximate
Invariance

Method: Anchor Images Selection



Method: Disentanglement Regularization



- Encode the distinct factors with independent latent variables in the latent feature space.

Xin Wang, et al. "Disentangled Representation Learning." arXiv Preprint arXiv: 2211.11695 (2022)

Method: Disentanglement Regularization

- Calculate the covariance of the k^{th} and l^{th} dimension of the batch of embedded representations with:

$$\operatorname{cov}\left(z_k, z_l\right)=\frac{1}{N-1} \sum_{i=1}^N\left(z_{ik}-\bar{z}_k\right) \cdot\left(z_{il}-\bar{z}_l\right),$$

where $\bar{z}_k$ is the mean of the k^{th} dimension feature across all N data points in the batch, calculated as $\bar{z}_k = \frac{1}{N}\sum_{i=1}^{N} z_{ik}$.

- Then the disentanglement loss is calculated by:

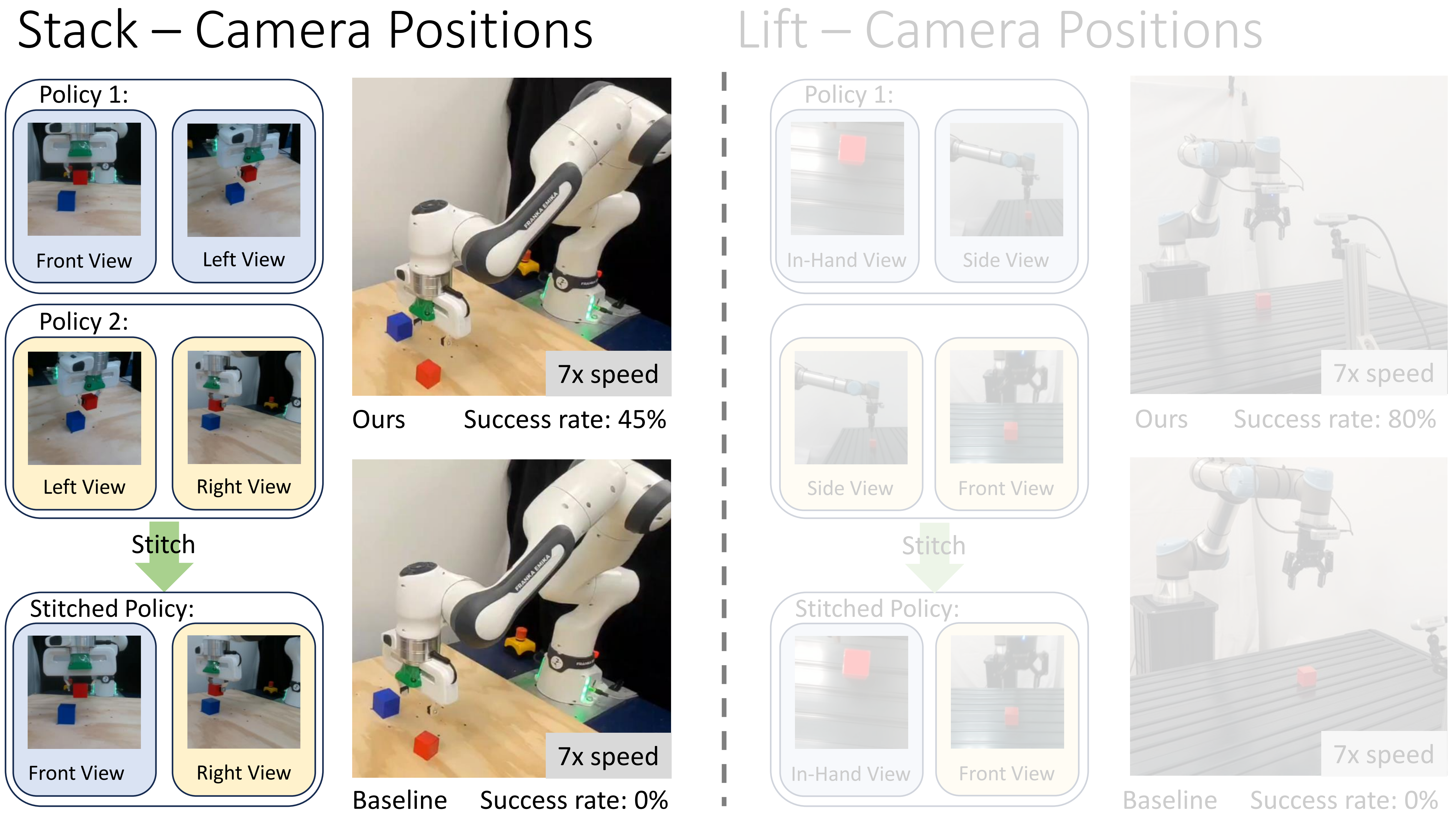
$$L_{\text{disent}}=\frac{1}{Z(Z-1)} \sum_{k=1}^Z \sum_{l=1, l \neq k}^Z\left|\operatorname{cov}\left(z_k, z_l\right)\right|,$$

- The final loss function we adopt for our PeS method is:

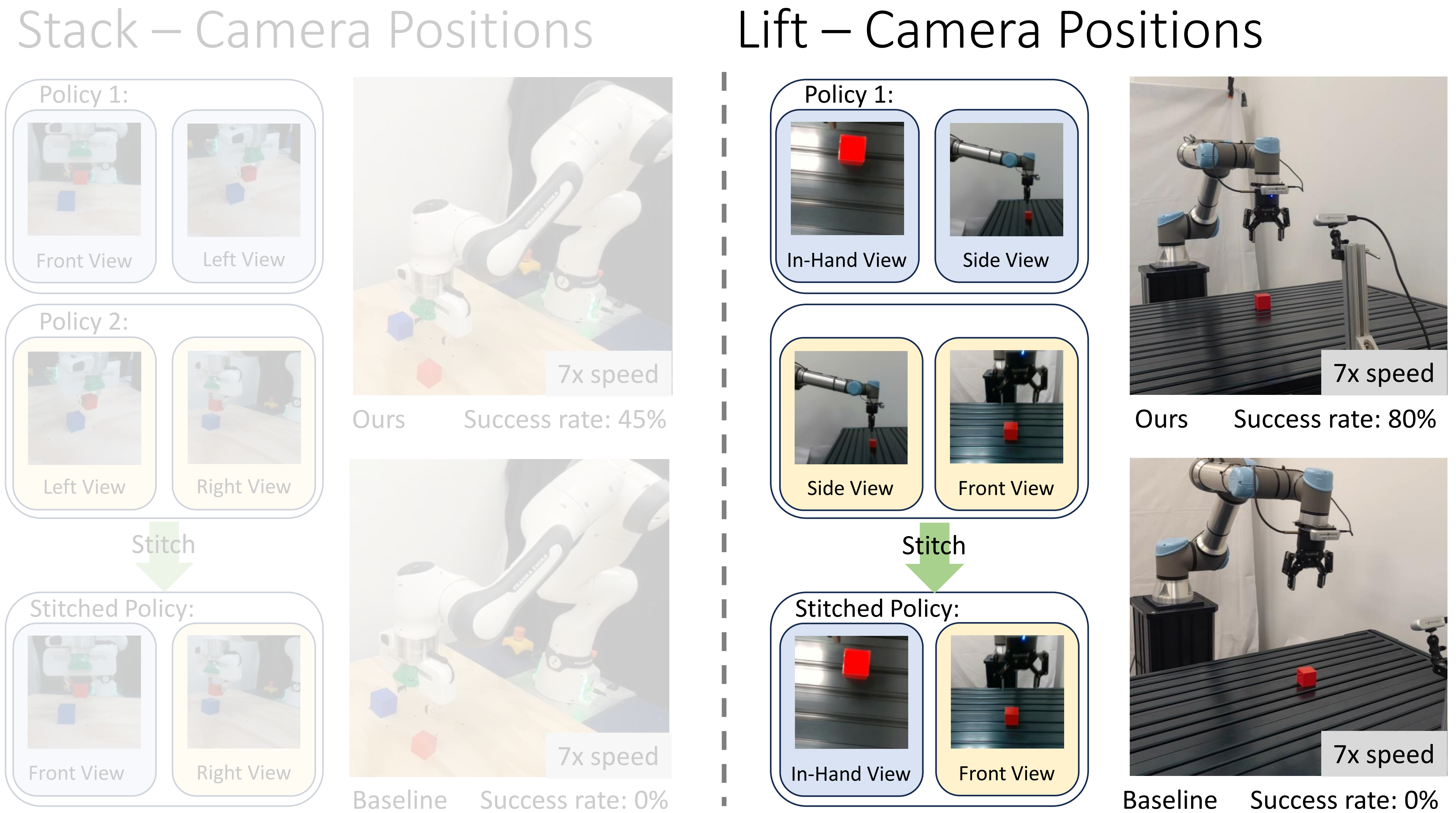
$$L_{PeS} = L_{BC} + \lambda L_{disent}$$

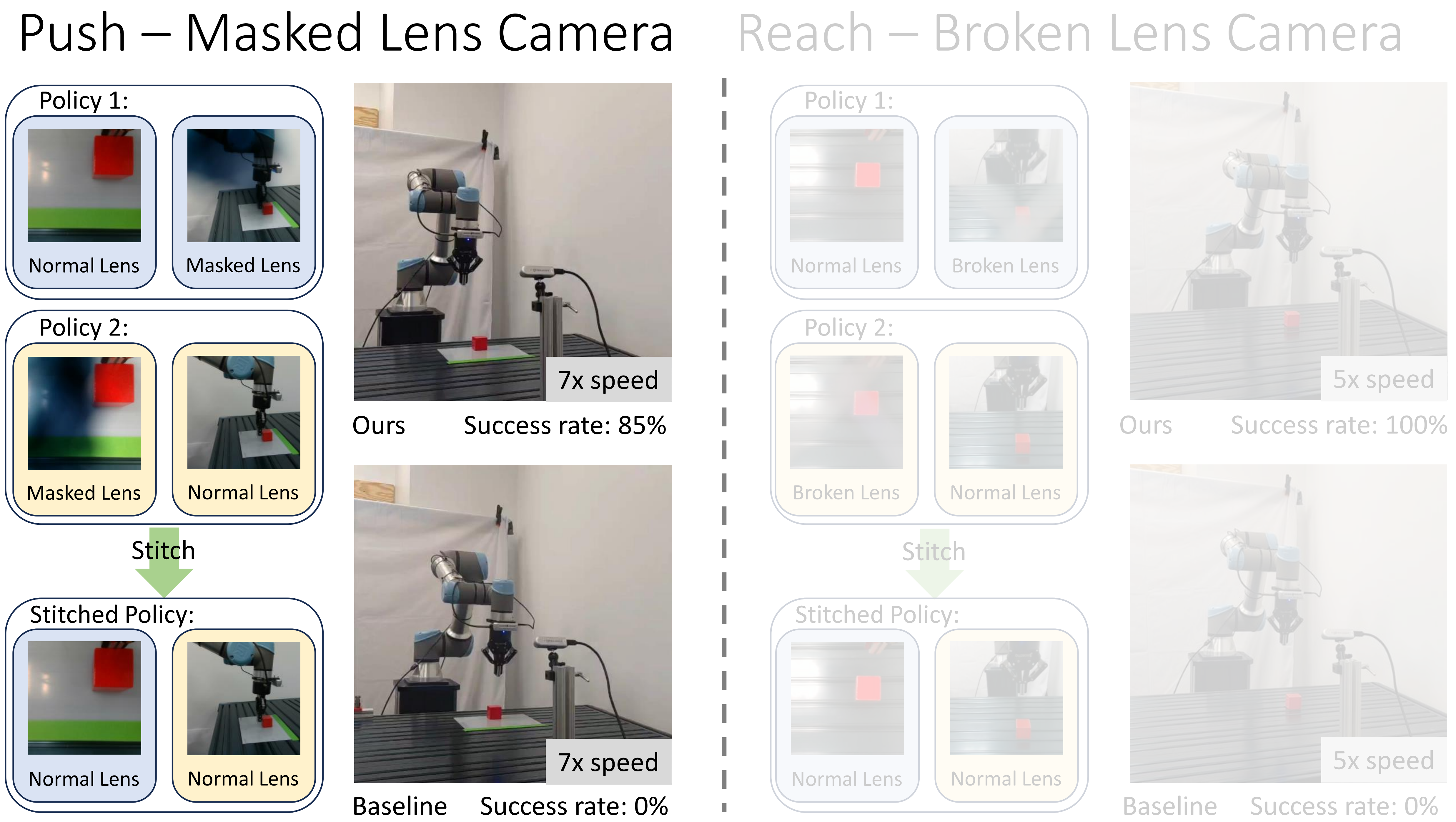
- This loss encourage the features at the latent space to be independent with each other. Therefore, it disentangles the underlying factors hidden in the observable data in representation form.

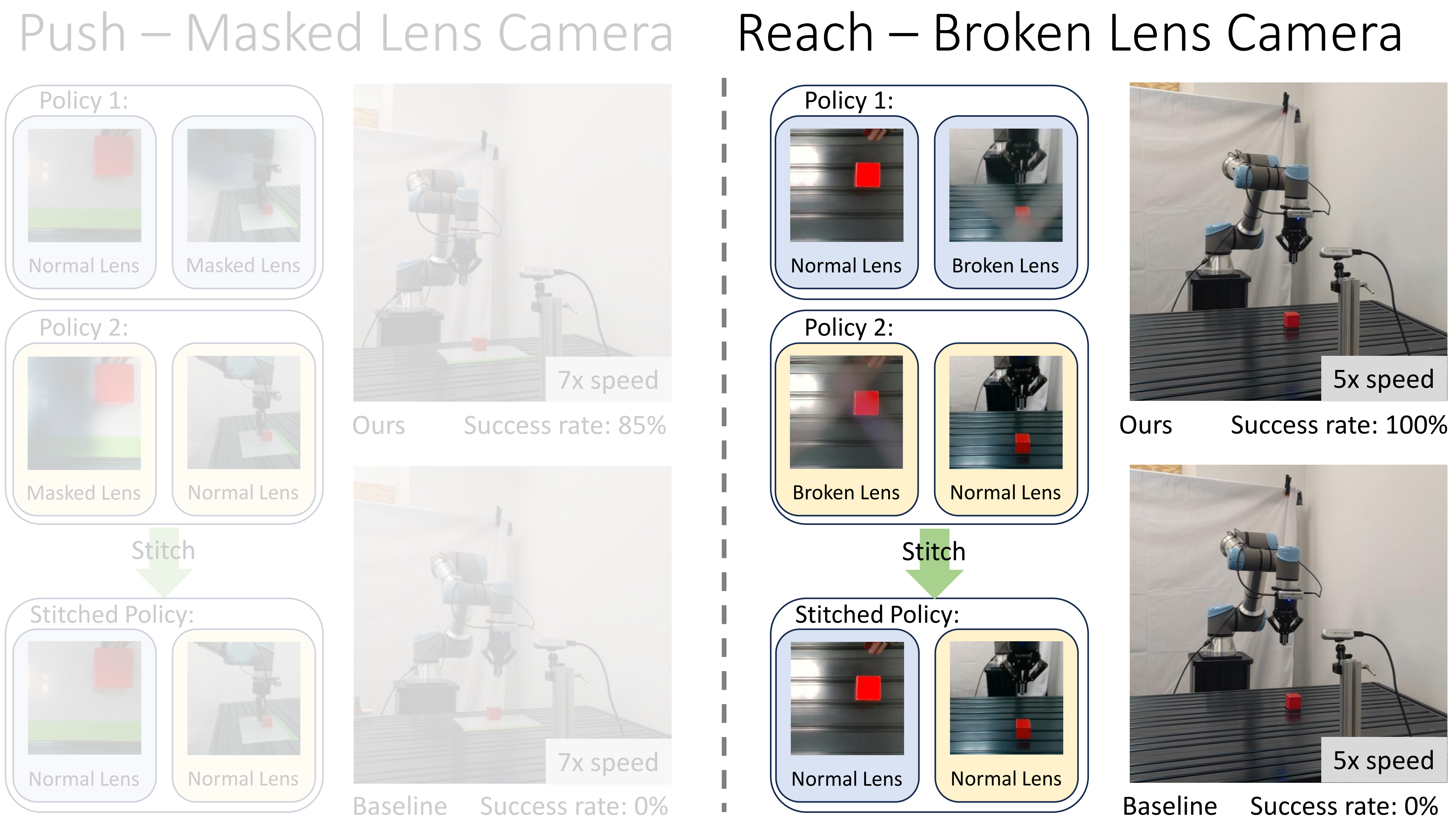
Real-World Experiments



Stack – Camera Positions
Policy 1:
Front View
Left View
Policy 2:
Left View
Right View
Stitch
Stitched Policy:
Front View
Right View
7x speed
Ours
Success rate: 45%
7x speed
Baseline
Success rate: 0%
Lift – Camera Positions
Policy 1:
In-Hand View
Side View
Side View
Front View
Stitch
Stitched Policy:
In-Hand View
Front View
7x speed
Ours
Success rate: 80%
7x speed
Baseline
Success rate: 0%


Stack – Camera Positions
Policy 1:
Front View
Left View
Policy 2:
Left View
Right View
Stitch
Stitched Policy:
Front View
Right View
7x speed
Ours
Success rate: 45%
7x speed
Baseline
Success rate: 0%
Lift – Camera Positions
Policy 1:
In-Hand View
Side View
Side View
Front View
Stitch
Stitched Policy:
In-Hand View
Front View
7x speed
Ours
Success rate: 80%
7x speed
Baseline
Success rate: 0%


Push – Masked Lens Camera
Policy 1:
Normal Lens
Masked Lens
Policy 2:
Masked Lens
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
7x speed
Ours
Success rate: 85%
7x speed
Baseline
Success rate: 0%
Reach – Broken Lens Camera
Policy 1:
Normal Lens
Broken Lens
Policy 2:
Broken Lens
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
5x speed
Ours
Success rate: 100%
5x speed
Baseline
Success rate: 0%



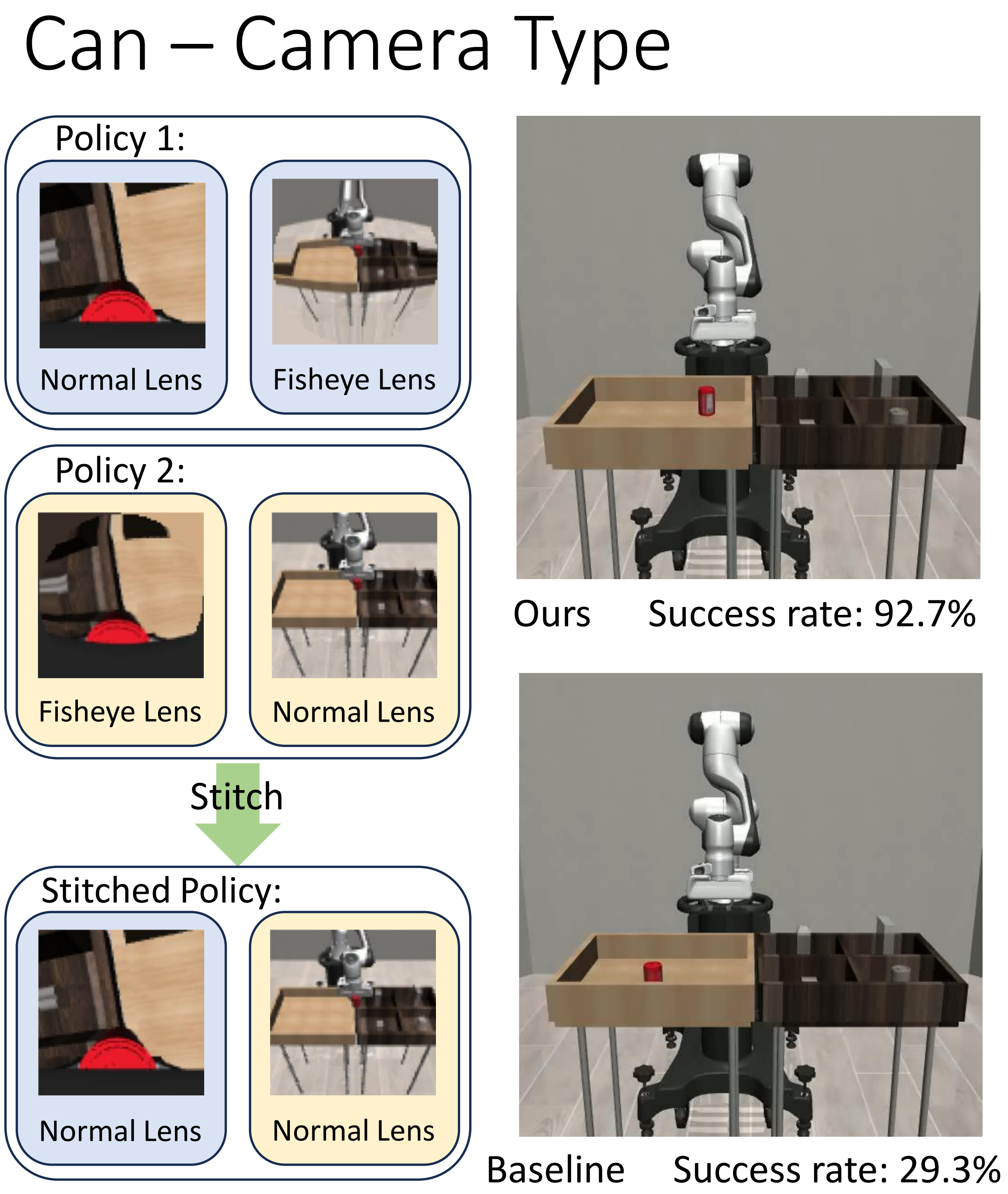
Push – Masked Lens Camera
Policy 1:
Normal Lens
Masked Lens
Policy 2:
Masked Lens
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
7x speed
Ours
Success rate: 85%
7x speed
Baseline
Success rate: 0%
Reach – Broken Lens Camera
Policy 1:
Normal Lens
Broken Lens
Policy 2:
Broken Lens
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
5x speed
Ours
Success rate: 100%
5x speed
Baseline
Success rate: 0%

Real-World Experiments Results

	Reach **broken lens**	**Push** **masked lens**	**Lift** **different positions**	**Stack** **different positions**
PeS	**100.0**	**85.0**	**80.0**	**45.0**
Devin et al. 2017	0.0	0.0	0.0	0.0

Zero-Shot Transfer Success Rates in Real World

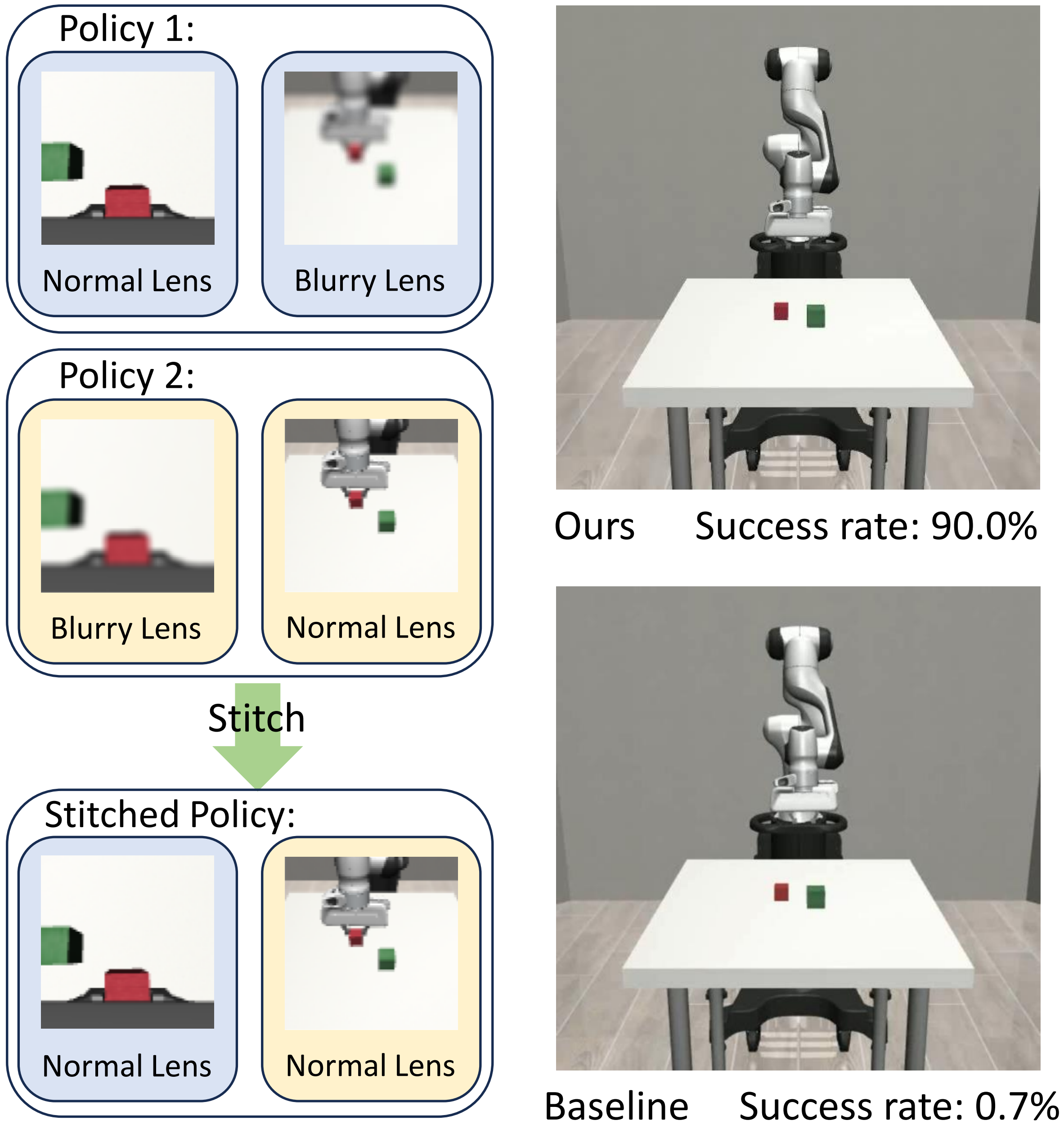
Simulation Experiments


Can – Camera Type
Policy 1:
Normal Lens
Fisheye Lens
Policy 2:
Fisheye Lens
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
Ours
Success rate: 92.7%
Baseline
Success rate: 29.3%

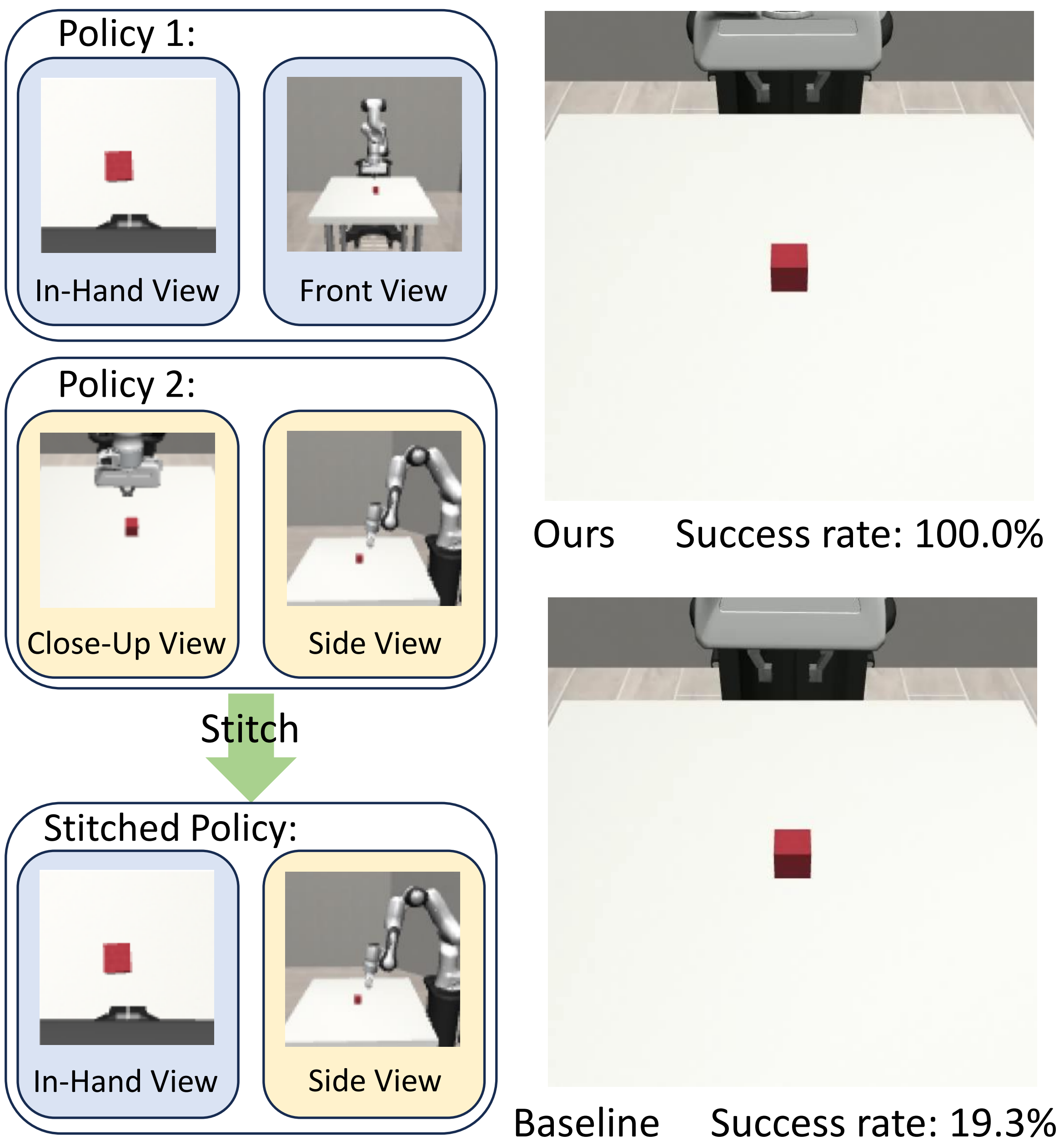
Stack – Blurry Camera
Policy 1:
Normal Lens
Blurry Lens
Policy 2:
Blurry Lens
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
Ours
Success rate: 90.0%
Baseline
Success rate: 0.7%

Can – Camera Type

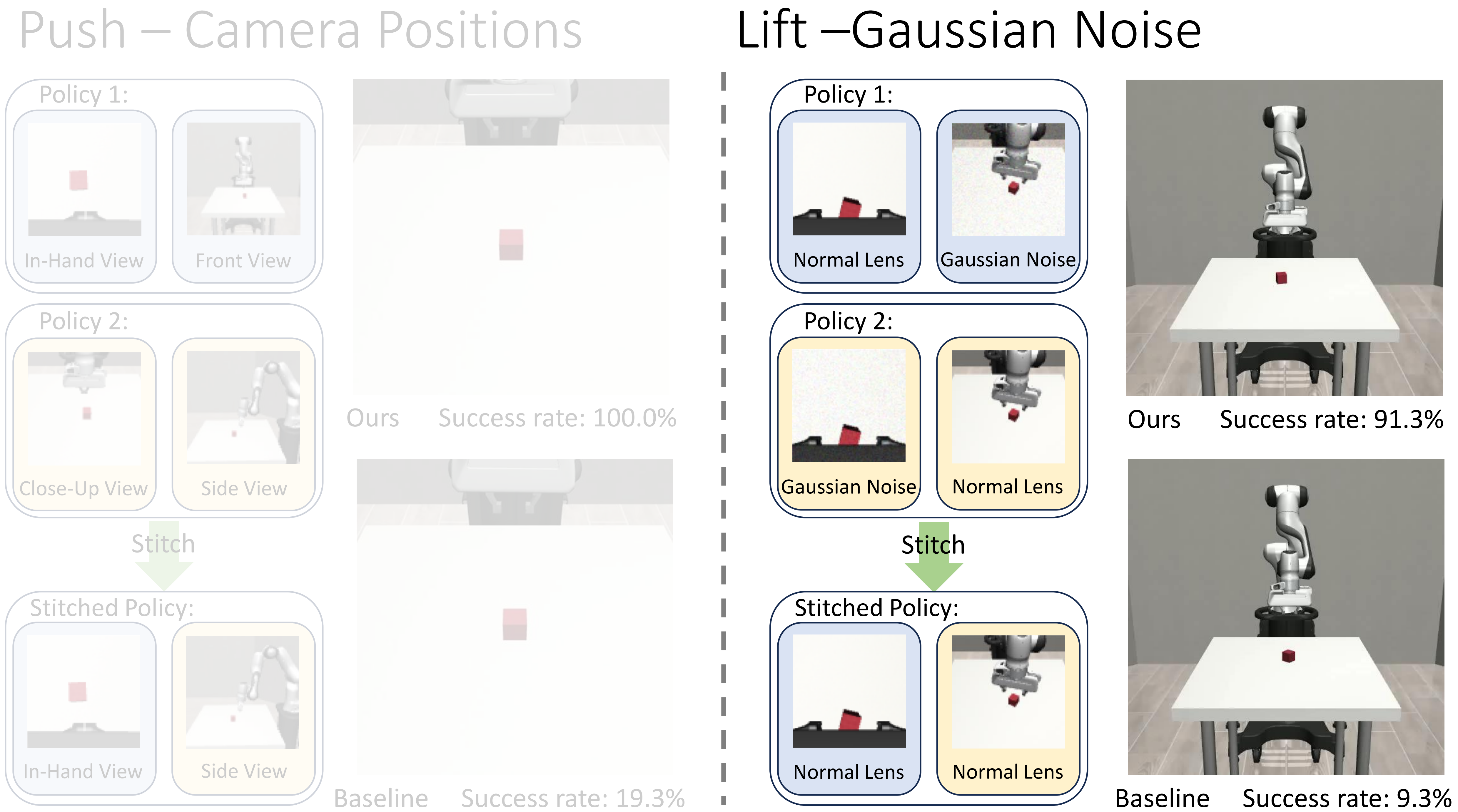
Stack – Blurry Camera



Push – Camera Positions

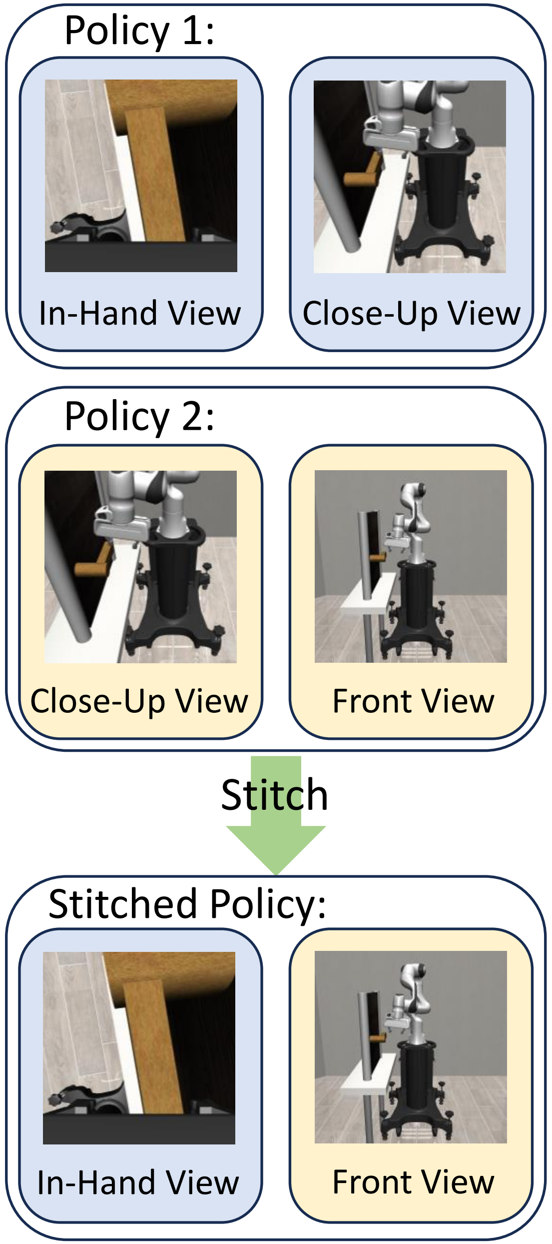


Lift –Gaussian Noise



Push – Camera Positions
Policy 1:
In-Hand View
Front View
Policy 2:
Close-Up View
Side View
Stitch
Stitched Policy:
In-Hand View
Side View
Ours
Success rate: 100.0%
Baseline
Success rate: 19.3%
Lift –Gaussian Noise
Policy 1:
Normal Lens
Gaussian Noise
Policy 2:
Gaussian Noise
Normal Lens
Stitch
Stitched Policy:
Normal Lens
Normal Lens
Ours
Success rate: 91.3%
Baseline
Success rate: 9.3%

Door Open – Camera Positions



Ours Success rate: 48.7%

Baseline Success rate: 0.0%

Simulation Experiments Results

		Mask	Zoom in	Blurred	Noise	Fisheye	Camera Position	Average
Push	Devin et al. 2017	60.7±10.6	8.7±4.99	16.7±3.77	59.3±6.80	29.3±7.36	19.3±5.73	32.3
	Cannistraci et al. 2024 (linear)	89.3±4.11	94.0±2.83	64.7±1.89	74.7±6.18	74.0±2.83	78.7±2.49	79.2
	Cannistraci et al. 2024 (non-linear)	12.7±1.89	18.7±4.99	42.8±3.27	23.3±0.94	6.0±4.32	5.3±2.49	18.1
	PeS (w/o disent. loss)	**100.0±0.0**	86.0±2.83	80.7±9.84	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	94.5
	PeS (w. l1 & l2 loss)	88.7±4.99	95.3±1.89	90.0±5.66	**100.0±0.0**	93.3±0.94	80.7±4.99	91.3
	PeS	**100.0±0.00**	**100.0±0.00**	**95.3±0.94**	**100.0±0.0**	92.7±2.50	**100.0±0.00**	**98**
Lift	Devin et al. 2017	0.0±0.00	5.3±2.49	48.0±5.89	9.3±4.11	14.7±4.99	36.0±1.63	18.9
	Cannistraci et al. 2024 (linear)	72.7±3.77	64.0±2.83	86.0±4.32	68.7±1.88	88.7±1.88	57.3±2.49	72.9
	Cannistraci et al. 2024 (non-linear)	89.3±2.49	36.0±3.27	52.7±3.40	93.3±2.49	16.7±2.49	21.3±0.94	51.6
	PeS (w/o disent. loss)	83.3±6.60	80.7±5.73	**93.3±0.94**	91.3±5.73	79.3±2.49	**93.3±2.49**	86.9
	PeS (w. l1 & l2 loss)	**97.3±2.49**	85.3±0.94	90.7±0.94	86.0±4.32	88.0±1.63	84.7±3.77	88.7
	PeS	92.7±2.50	**94.7±1.89**	89.3±4.11	**96.0±1.63**	**88.7±0.94**	93.0±0.03	**92.4**

Zero-Shot Transfer Success Rates in basic Simulation tasks

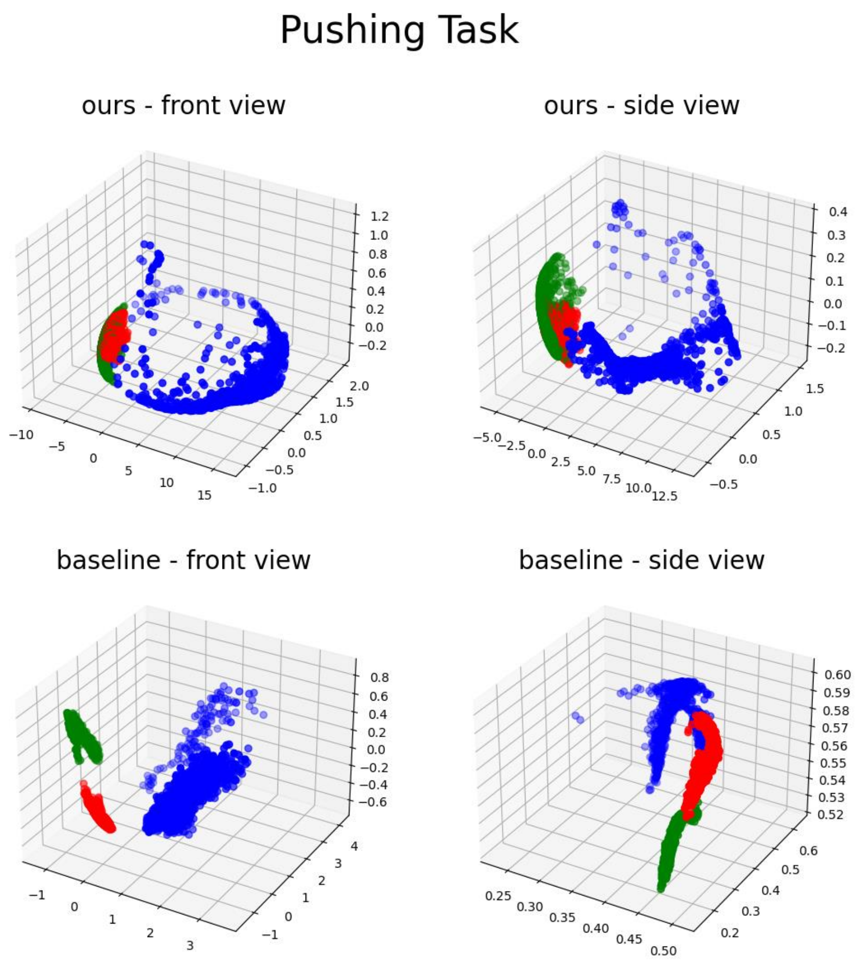
		Mask	Zoom in	Blurred	Noise	Fisheye	Camera Position	Average
Can	Devin et al. 2017	19.3±5.25	24.7±1.89	2.67±1.89	6.0±4.32	29.3±3.40	1.3±1.89	13.9
	Cannistraci et al. 2024 (linear)	33.3±0.94	48.0±1.63	48.7±2.49	65.3±0.94	26.7±3.77	34.7±3.77	42.8
	Cannistraci et al. 2024 (non-linear)	72.7±0.94	24.7±2.49	37.3±4.99	42.7±3.40	8.7±1.89	39.3±1.89	37.6
	PeS (w/o disent. loss)	44.7±8.06	**89.3±4.11**	34.7±4.11	30.7±6.80	**92.7±2.50**	44.7±3.40	56.1
	PeS (w. l1 & l2 loss)	47.3±0.94	58.7±1.88	54.0±8.64	36.0±7.12	58.7±1.88	64.7±6.60	53.2
	PeS	**83.3±5.24**	**89.3±2.49**	**74.0±2.83**	**78.7±4.11**	56.0±2.83	**78.7±2.49**	**76.7**
Stack	Devin et al. 2017	0.7±0.94	8.0±1.63	0.7±0.94	24.0±2.83	0.0±0.00	14.0±3.27	7.9
	Cannistraci et al. 2024 (linear)	47.3±0.94	62.0±4.32	32.7±3.77	30.7±0.94	54.0±8.64	14.7±6.18	40.2
	Cannistraci et al. 2024 (non-linear)	10.0±1.63	12.0±0.00	0.0±0.00	3.3±0.94	0.0±0.00	0.7±0.94	4.3
	PeS (w/o disent. loss)	34.0±11.43	10.7±4.11	62.0±10.71	34.0±7.12	22.7±3.77	26.0±4.32	31.6
	PeS (w. l1 & l2 loss)	92.7±0.94	**98.0±0.00**	62.7±6.60	24.0±4.90	59.3±7.36	58.7±1.88	65.9
	PeS	**94.7±0.94**	96.7±0.94	**90.0±1.63**	**96.7±1.89**	**97.3±2.49**	**80.0±4.90**	**92.6**
Door Open	Devin et al. 2017	9.3±4.11	5.3±0.94	0.0±0.00	4.0±1.63	0.7±0.94	0.0±0.00	3.2
	Cannistraci et al. 2024 (linear)	0.0±0.00	1.3±0.94	10.7±2.49	10.7±4.99	2.0±1.63	47.3±9.29	12
	Cannistraci et al. 2024 (non-linear)	26.0±2.83	31.3±4.99	49.3±8.22	48.0±5.89	62.7±3.40	44.7±3.40	43.7
	PeS (w/o disent. loss)	24.7±7.71	44.0±2.83	34.7±3.77	0.7±0.94	36.7±0.94	23.3±3.40	27.4
	PeS (w. l1 & l2 loss)	4.0±1.63	**78.0±5.66**	3.3±0.94	2.0±1.63	42.7±4.99	6.0±3.26	22.7
	PeS	**58.7±4.11**	68.7±0.94	**70.7±0.94**	**52.7±3.40**	**64.7±4.99**	**48.7±3.40**	**60.7**

Zero-Shot Transfer Success Rates in difficult Simulation tasks

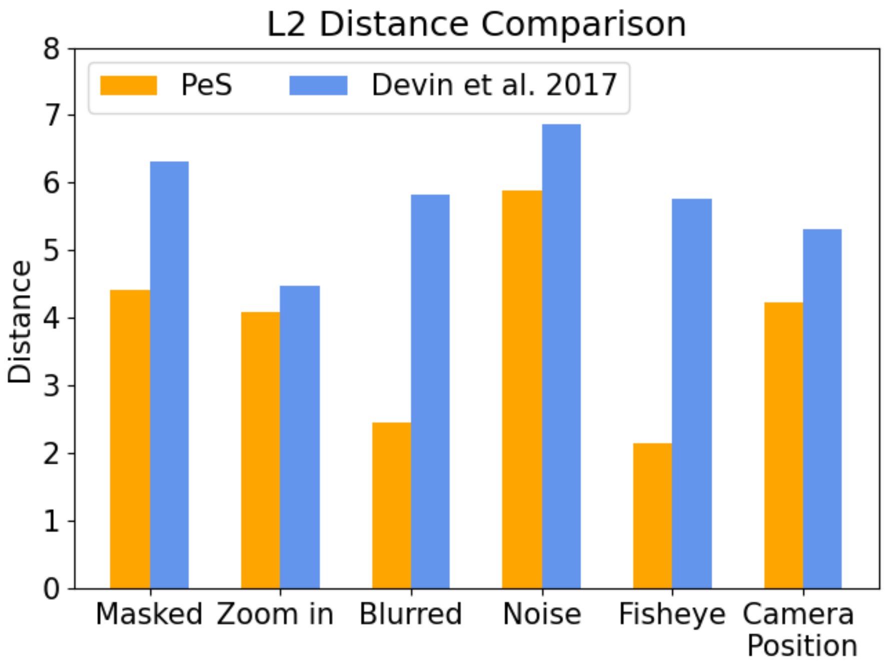
Latent Space at Module Interface

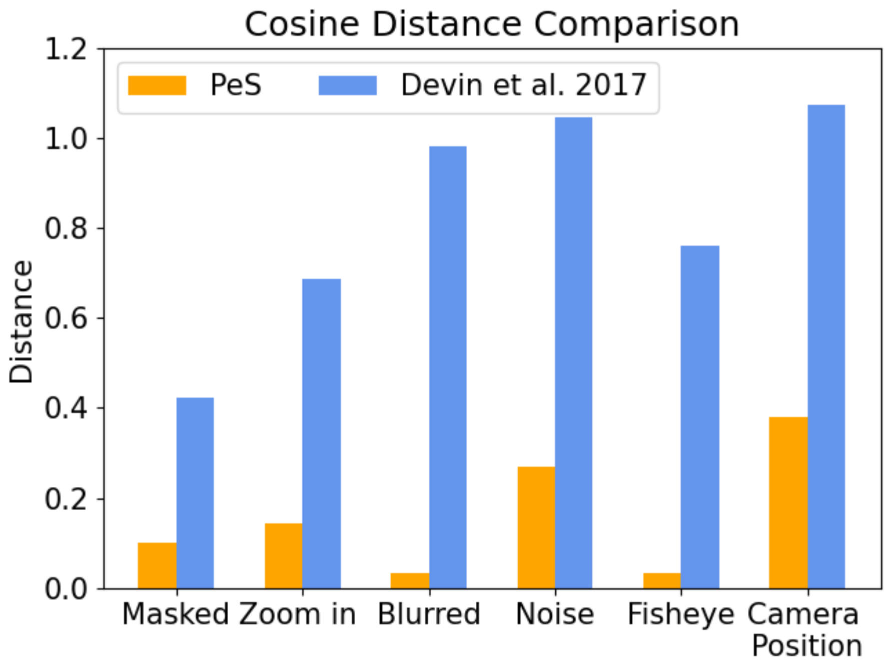
Latent Space at Module Interface

- Red dots: robot's end effector is at higher positions
- Green dots: medium heights
- Blue dots: lower positions near the cube.
- The 256D representations are reduced to 3D with PCA.
- PeS: similar latent representation shapes with each other.
- Devin baseline: approximately isometric transformation (rotation in this case) relationship with each other.



Latent Space at Module Interface



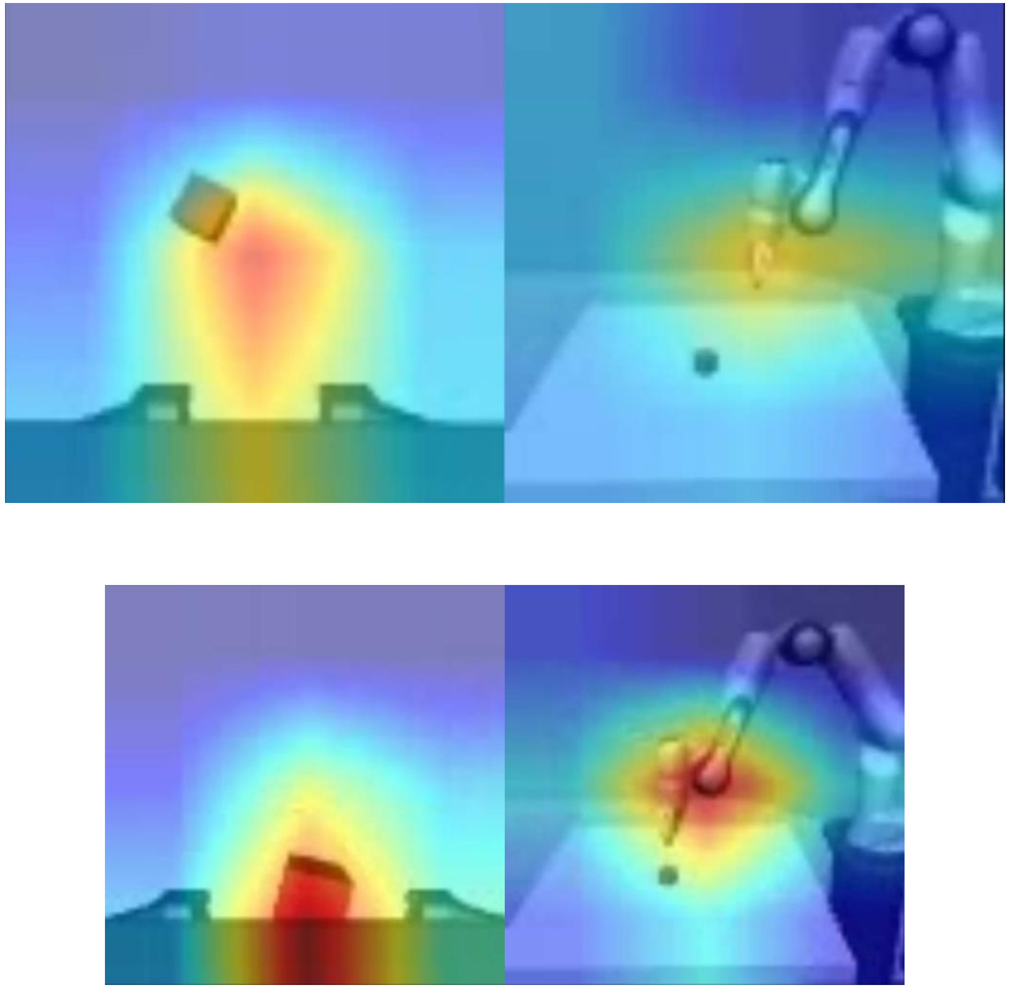


- One representation is from the second view encoder of policy 1 and the other is from the second view encoder of policy 2.
- Cosine distance: PeS significantly smaller than Devin baseline.
- L2 distances: PeS smaller than Devin baseline, but the differences are not pronounced in some cases.

Attention Heatmap with Grad-CAM

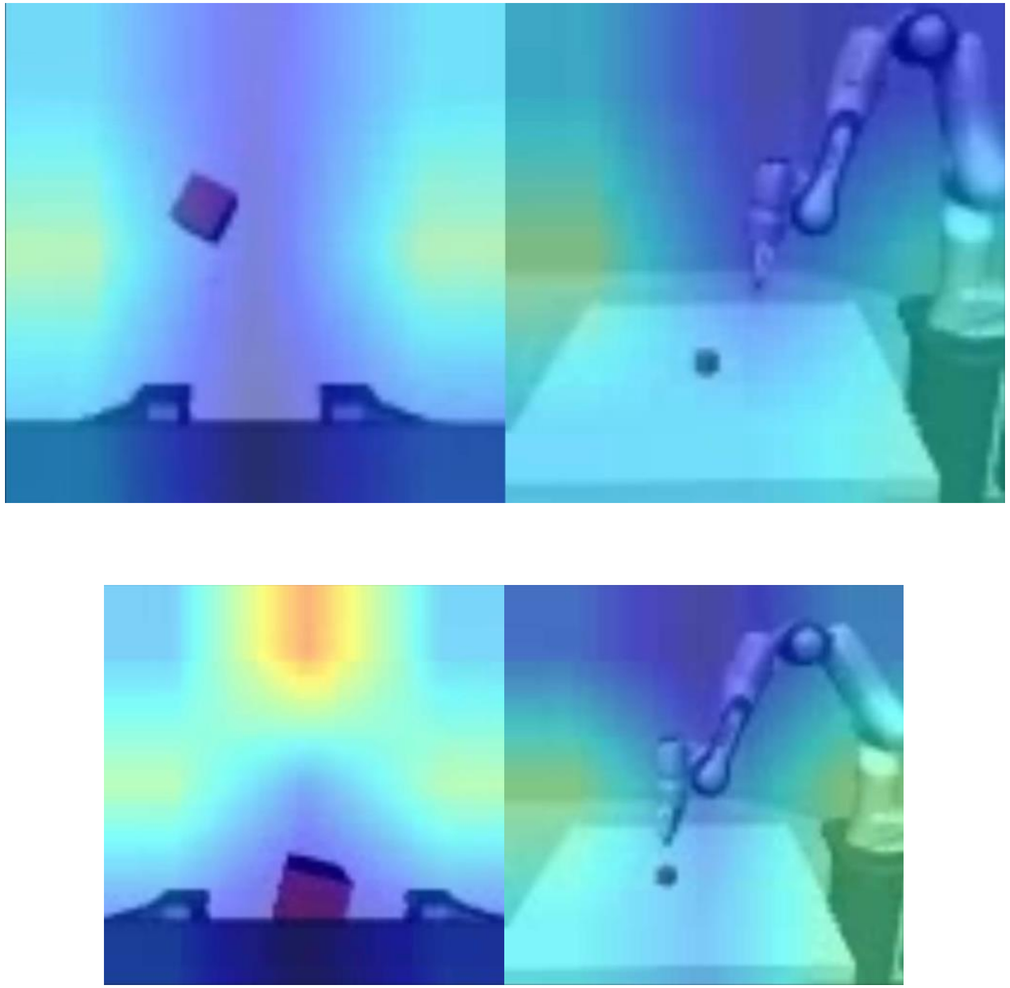
Attention Heatmap with Grad-CAM

Ours



In-Hand View Side View

Baseline



In-Hand View Side View

Conclusion

- **Perception Stitching (PeS)** is a method for zero-shot visuomotor policies transfer via latent spaces alignment.
- **Aligns** the **latent spaces** of different visual encoders and allows the trained visual encoders to be reused in a plug-and-go manner.
- Evaluation on 30 simulation experiments and 4 real-world experiments shows the pronounced advantage of PeS, and our analysis further reveals the mechanism of its superior performance.

Thank You!

Attention Heatmap with Grad-CAM

- We modify the Gradient-weighted Class Activation Mapping (Grad-CAM) approach to highlight the regions that the policies pay attention to.

- Replace the before-softmax score y^c for class c of the image classification networks with the log-likelihood $l(a)$ of the robot action a in the training dataset.

Attention Heatmap with Grad-CAM

- Denote the k^{th} feature map activation output from the last convolutional layer as A^k.
- The backpropagated gradient of $l(a)$ with respect to A^k is computed as $\frac{\partial l(a)}{\partial A^k}$.
- do global average pooling of these gradients over the width (indexed by i) and height (indexed by j) dimensions of the feature map to get the neuron importance weight α_k^a:

$$\alpha_k^a = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial l(a)}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

- This weight α_k^a captures the 'importance' of feature map k for robot action a.

Attention Heatmap with Grad-CAM

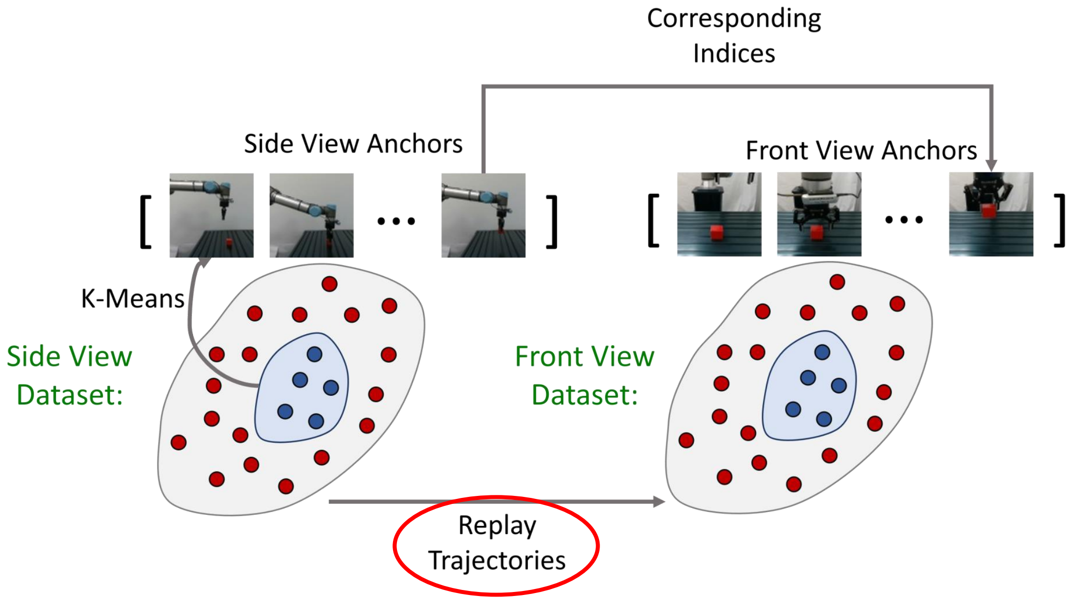
- Then, the attention map Grad-CAM is calculated as the weighted combination of forward activation maps followed by a ReLU:

$$L^a_{\text{Grad-CAM}} = \text{ReLU}\underbrace{\left(\sum_k \alpha^a_k A^k\right)}_{\text{linear combination}}$$

- We apply ReLU because we are only interested in the features that have a positive influence on the actions.
- This $L^a_{Grad-CAM}$ is a heatmap of the same size as the convolutional feature maps A^k. We upsample it to the input image size with bilinear interpolation to get the final attention heatmap of the input image.
- A larger value on this heatmap means this pixel contributes to a larger gradient of the log-likelihood of the robot action.

Limitations and Future Work

Limitations and Future Work



- Limitations:
- Replaying the trajectories takes about twice the time as collecting data with random sampling.