A Reinforcement Learning Approach to MIMO Radar Calibration



TEXAS A&M UNIVERSITY
Engineering

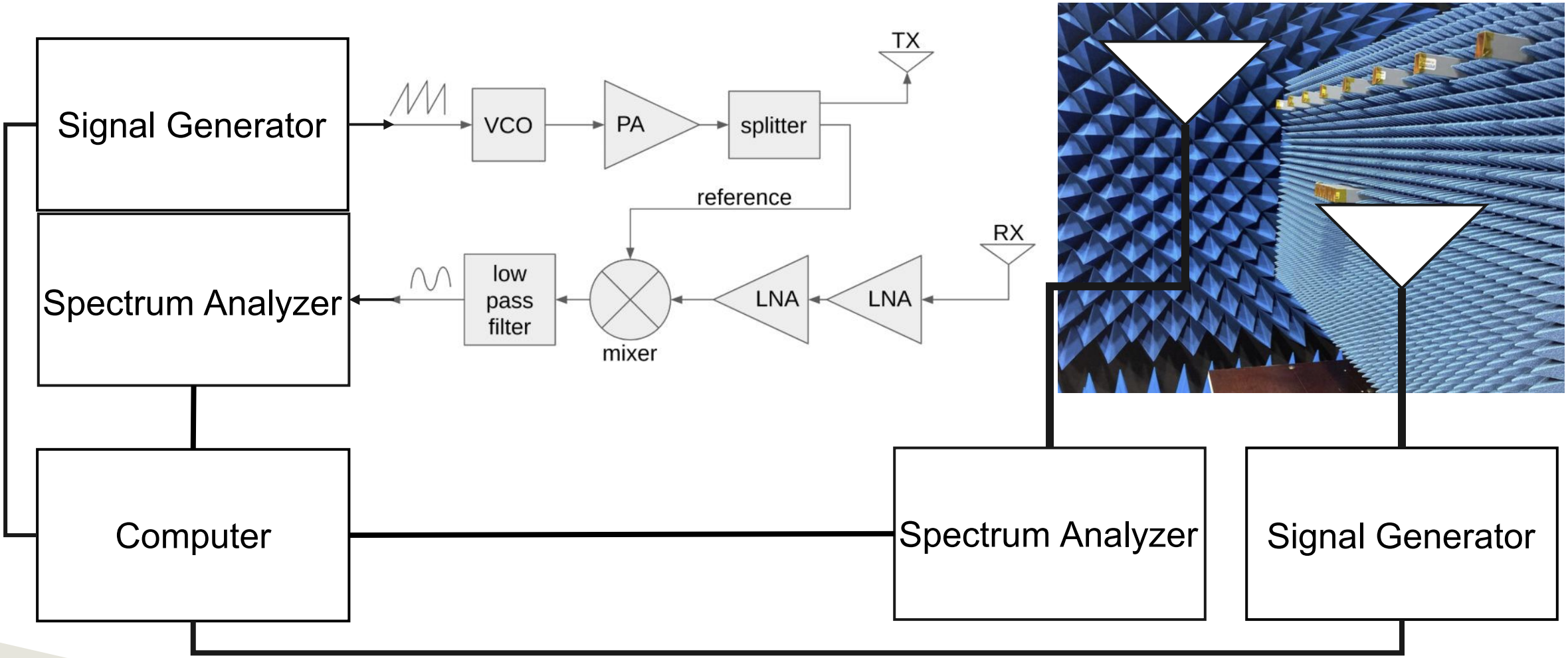
MiWiRa Lab

Table of Contents

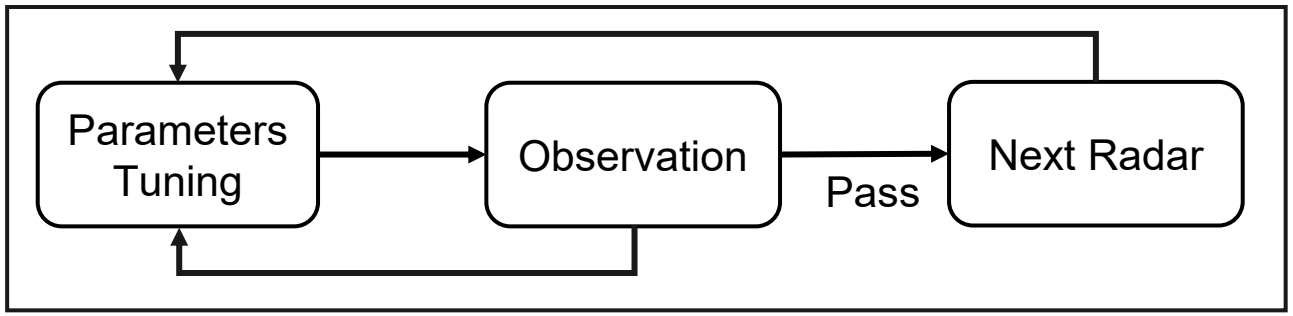
1. Background and Motivation
2. Detailed Motivation
3. Problem Description
4. Advanced Reinforcement Learning: Offline RL
5. Results and Analysis
6. Conclusion

Background and Motivation

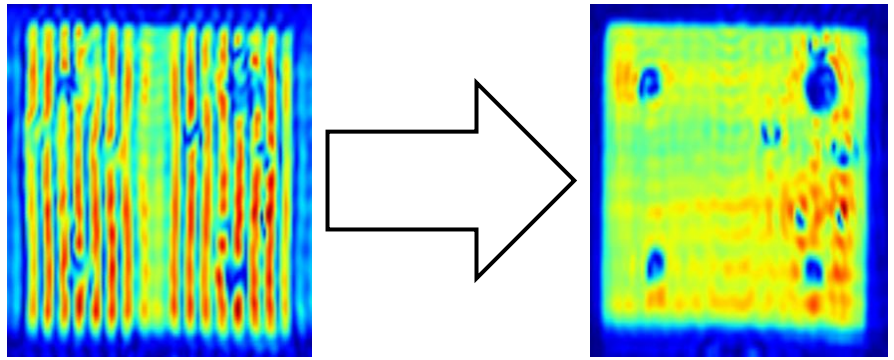
Traditional Calibration Methods for MIMO Radar



Calibration using MIMO-SAR Imaging



Workflow



Observation

$1e^{0}$
$a_1e^{\theta_1}$
$a_2e^{\theta_2}$
$a_3e^{\theta_3}$

Control parameters

Human

Dataset for Tx1 → Rx1

Dataset for Tx1 → Rx2

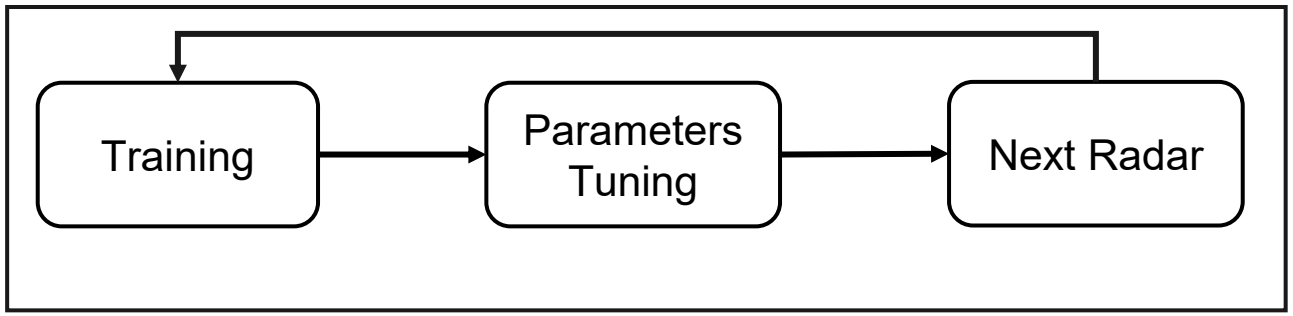
Dataset for Tx2 → Rx1

Dataset for Tx2 → Rx2

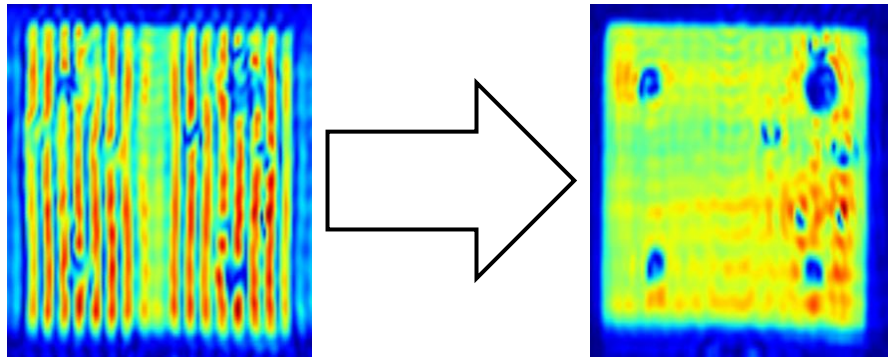
MIMO-SAR Imaging Algorithm

Target Image

Calibration using Supervised Learning



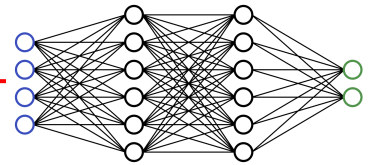
Workflow



Observation

Control parameters

$1e^{0}$
$a_1 e^{\theta_1}$
$a_2 e^{\theta_2}$
$a_3 e^{\theta_3}$



Supervised Learning

Dataset for Tx1 → Rx1

Dataset for Tx1 → Rx2

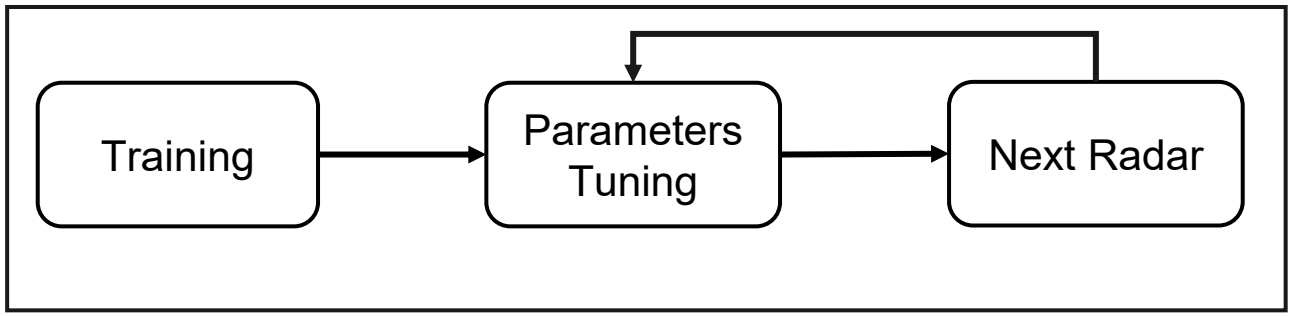
Dataset for Tx2 → Rx1

Dataset for Tx2 → Rx2

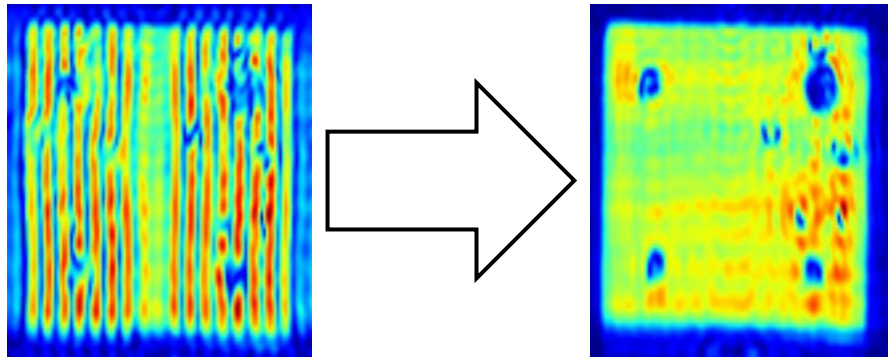
MIMO-SAR Imaging Algorithm

Target Image

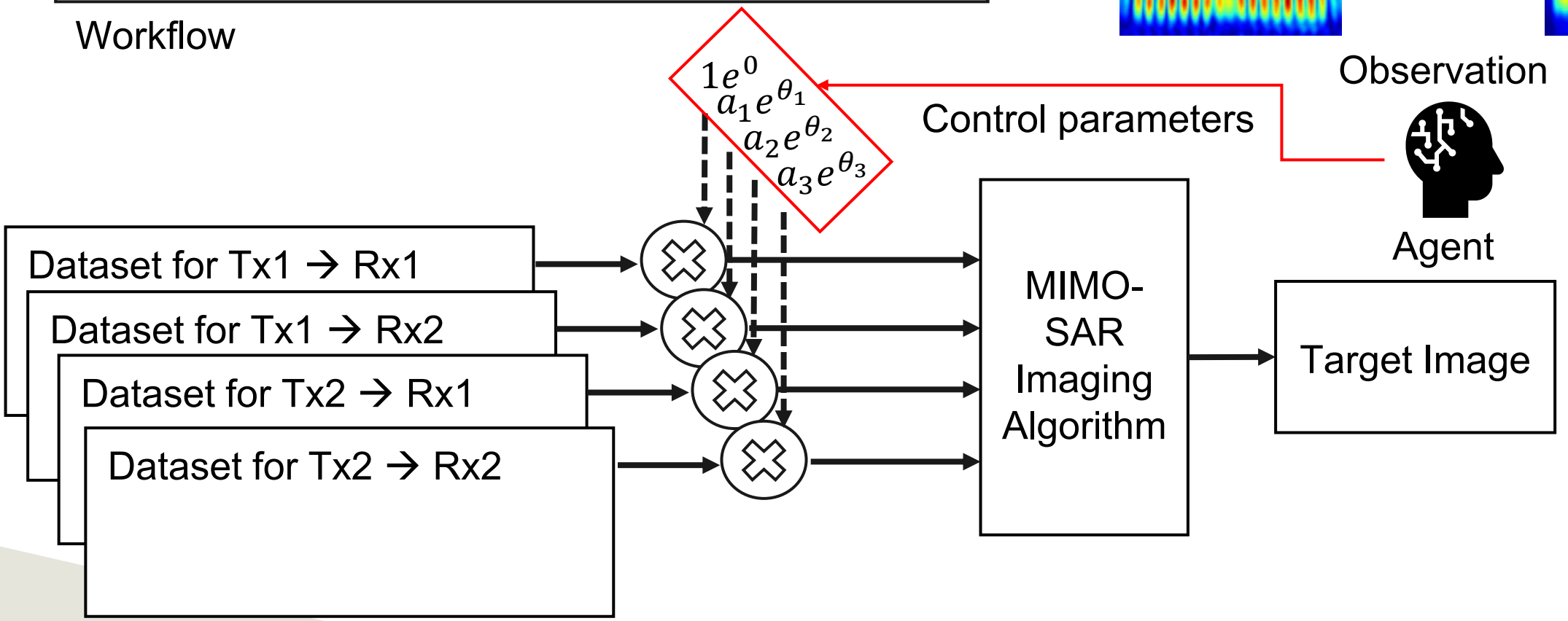
Calibration using Reinforcement Learning



Workflow



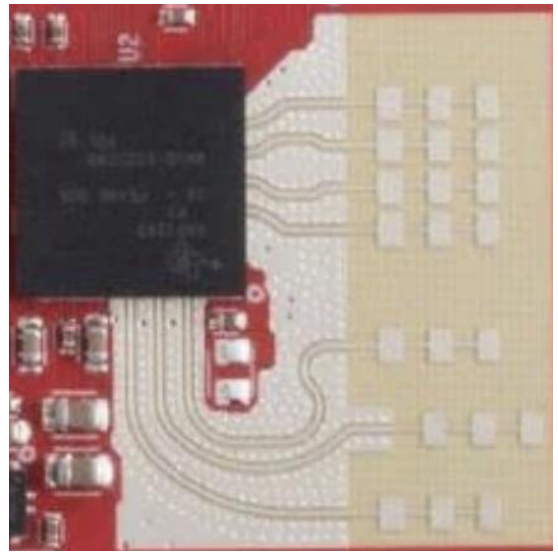
Observation



Detailed Motivation

Calibration Parameters for MIMO-SAR Imaging

The radars manufactured in the factory have different calibration parameters due to various process-related factors and performance optimization.



$$\underline{a} \triangleq \{a_1, a_2, a_3\}$$
$$\underline{\theta} \triangleq \{\theta_1, \theta_2, \theta_3\}$$

$1e^{0}$
$a_1 e^{\theta_1}$
$a_2 e^{\theta_2}$
$a_3 e^{\theta_3}$

Dataset for Tx1 → Rx1

Dataset for Tx1 → Rx2

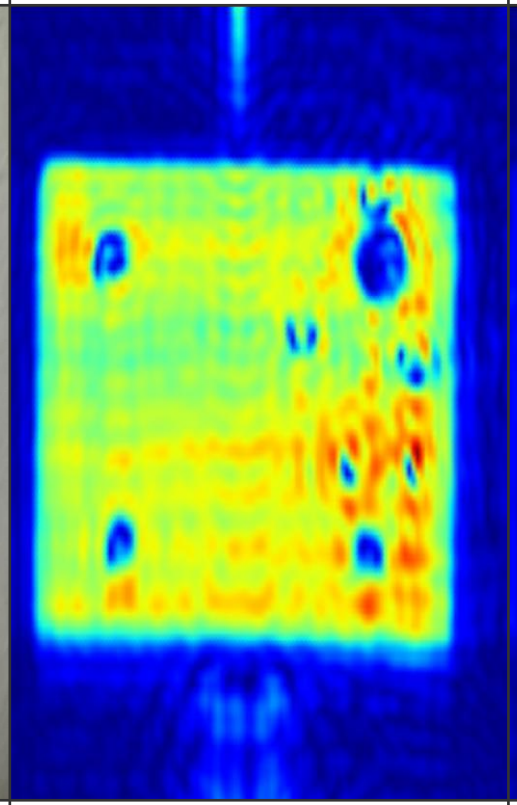
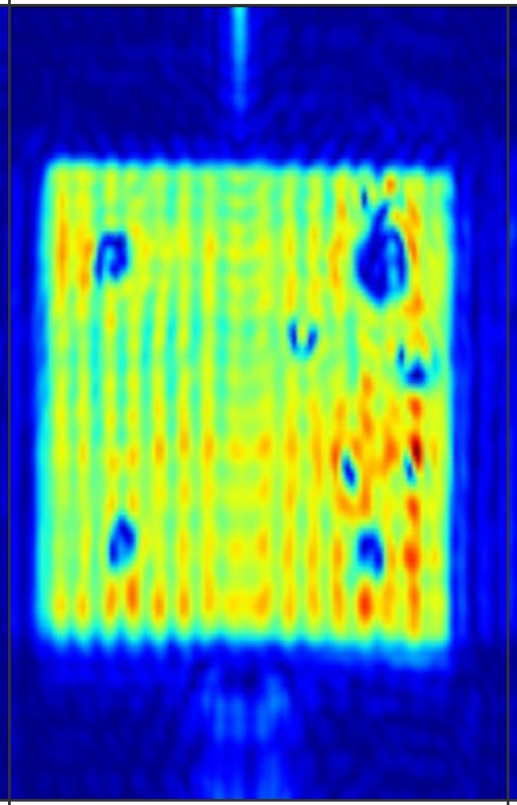
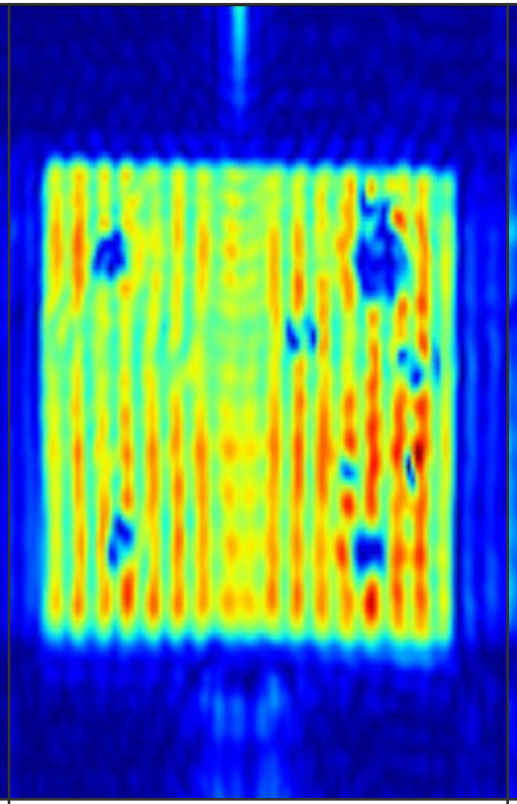
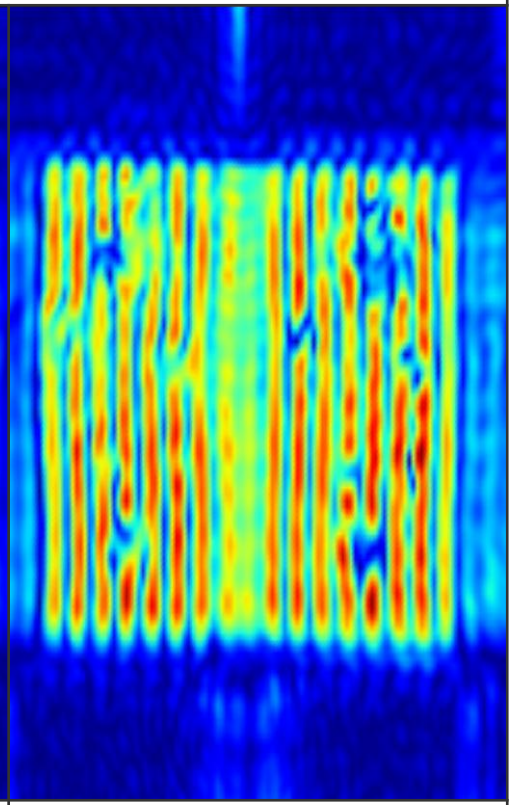
Dataset for Tx2 → Rx1

Dataset for Tx2 → Rx2

MIMO-SAR Imaging Algorithm

Target Image

Emulation Example Based on Calibrated Radar Data

Optical data	Image from calibrated radar	Images from intentionally miscalibrated virtual radars @ $\underline{a} = \{1, 1, 1\}$ & $\underline{\theta} = \{0^\circ, 0^\circ, 0^\circ\}$		
				
Optimal Calibration Parameters	$\underline{a}_{opt} = \{1, 1, 1\}$ $\underline{\theta}_{opt} = \{0^\circ, 0^\circ, 0^\circ\}$	$\underline{a}_{opt} = \{0.8, 0.9, 1.2\}$ $\underline{\theta}_{opt} = \{0^\circ, -30^\circ, 5^\circ\}$	$\underline{a}_{opt} = \{1, 1, 1000\}$ $\underline{\theta}_{opt} = \{0^\circ, 0^\circ, 0^\circ\}$	$\underline{a}_{opt} = \{1, 1, 1\}$ $\underline{\theta}_{opt} = \{0^\circ, 0^\circ, 180^\circ\}$

Limitations of Exhaustive Search for Calibration

$Assume, a_1, a_2, a_3 \in \{0.5, 1, 2\}$ and $\theta_1, \theta_2, \theta_3 \in \{-90^\circ, 0^\circ, 90^\circ, 180^\circ\}$

Image Generation @
$\underline{a} = \{1, 1, 1\}$
$\underline{\theta} = \{0^\circ, 0^\circ, 0^\circ\}$

Mapping

Generated Images	Optimal Calibration Parameters
Image Data 1	$\underline{a}_{opt} = \{1, 1, 1\}$ $\underline{\theta}_{opt} = \{0^\circ, 0^\circ, 0^\circ\}$
Image Data 2	$\underline{a}_{opt} = \{1, 1, 2\}$ $\underline{\theta}_{opt} = \{0^\circ, 0^\circ, 0^\circ\}$
⋮	⋮
Image Data $3^3 \times 4^3$	$\underline{a}_{opt} = \{2, 2, 2\}$ $\underline{\theta}_{opt} = \{180^\circ, 180^\circ, 180^\circ\}$

Optimal Calibration Parameters

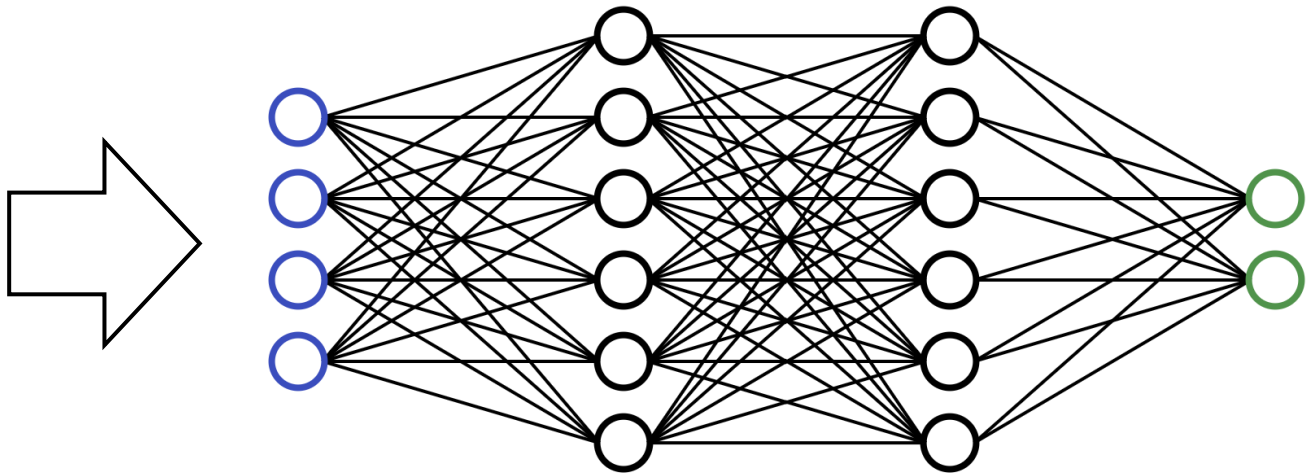
We can construct this table through emulation (Training) before the actual calibration (Test).

⇨

However, if the parameter values are continuous or the number of MIMO channels increases, the size of the table becomes excessively large.

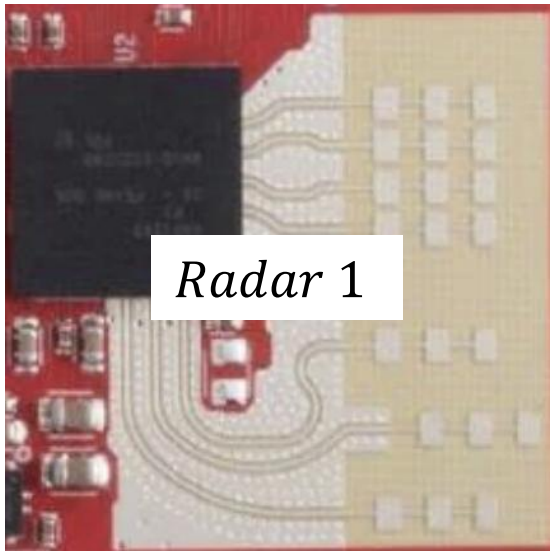
Limitations of Supervised Learning for Calibration

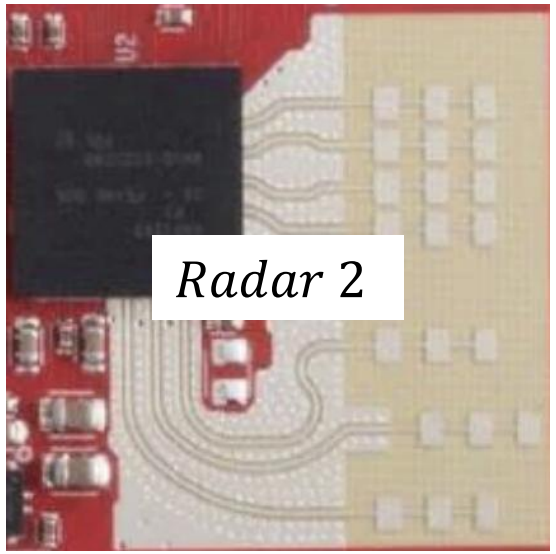
Generated Images	Optimal Calibration Parameters
Image Data 1	$\underline{a}_{opt} = \{1, 1, 1\}$ $\underline{\theta}_{opt} = \{0^{\circ}, 0^{\circ}, 0^{\circ}\}$
Image Data 2	$\underline{a}_{opt} = \{1, 1, 2\}$ $\underline{\theta}_{opt} = \{0^{\circ}, 0^{\circ}, 0^{\circ}\}$
⋮	⋮
Image Data $n^3 \times m^3$	$\underline{a}_{opt} = \{2, 2, 2\}$ $\underline{\theta}_{opt} = \{180^{\circ}, 180^{\circ}, 180^{\circ}\}$



If the table is represented as a function using a neural network model or other models, the issue of table size can be resolved.

Limitations of Supervised Learning for Calibration

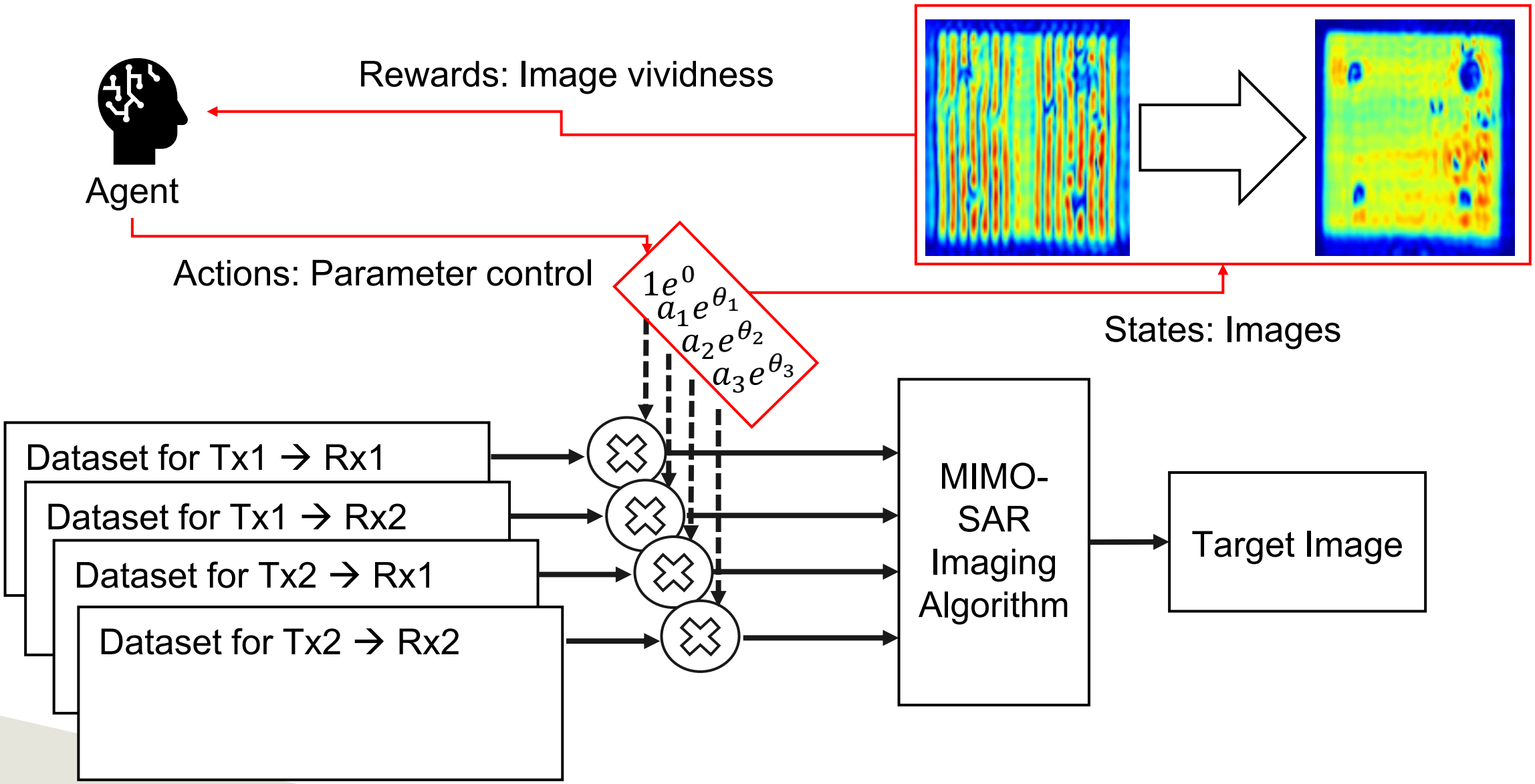




$$\{Emulation\ table\ for\ Radar\ 1\} \neq \{Emulation\ table\ for\ Radar\ 2\}$$

However, we cannot guarantee that a model trained on Radar 1 can be applied to all other radars. For example, the optimal image for Radar 2 may be better than the optimal image for Radar 1. If, during the calibration of Radar 2, the image measured at the initial condition is mapped to the optimal image of Radar 1, no further calibration will be performed, and the initial value will be used as the optimal calibration parameters. Therefore, in Radar 2, it may become impossible to find its own optimal image.

Formulating the problem as a Markov Decision Process



Problem Description

Design of the Action Space

$$10\log_{10} a_i \in \{-2 + k\Delta_a | k = 0, 1, \cdots, N_a\}$$

$$\theta_i \in \{-20° + k\Delta_\theta | k = 0, 1, \cdots, N_\theta\}$$

$$where\ \Delta_a = \frac{4}{N_a}, \Delta_\theta = \frac{40°}{N_\theta}, i = 1, 2, 3$$

$$\mathcal{A} = \left\{ a_j^{\pm}, \theta_j^{\pm} \middle| j = 1, 2, 3 \right\}$$

$$where\ a_j^{\pm}:\ a_j \leftarrow a_j \pm \Delta_a, \theta_j^{\pm}:\ \theta_j \leftarrow \theta_j \pm \Delta_\theta$$

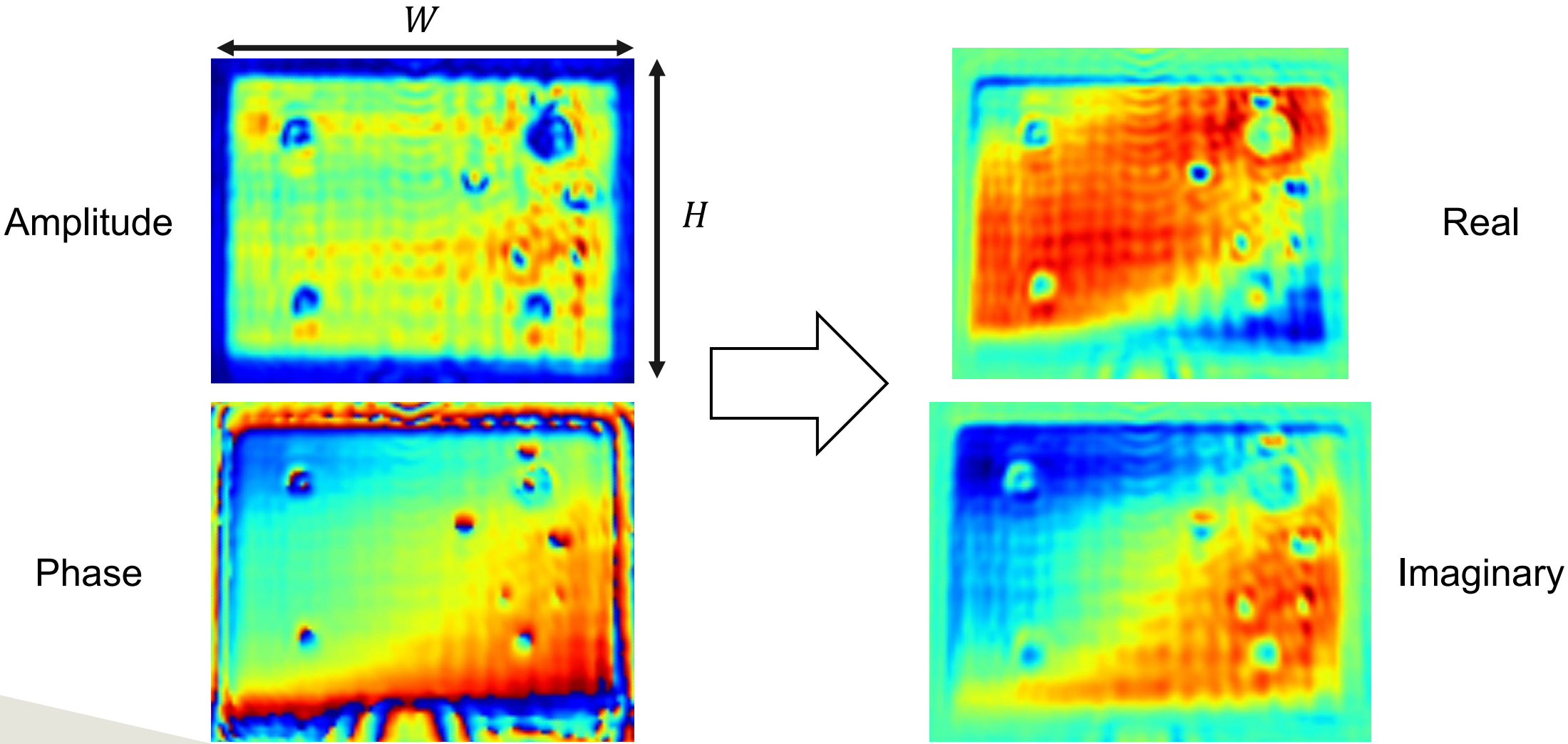
$Example$

$$\begin{bmatrix} a_1 = 2 \\ \theta_1 = 20° \\ a_2 = 0 \\ \theta_2 = -20° \\ a_3 = -2 \\ \theta_3 = 0° \end{bmatrix} a_1^- \rightarrow \begin{bmatrix} a_1 = 1 \\ \theta_1 = 20° \\ a_2 = 0 \\ \theta_2 = -20° \\ a_3 = -2 \\ \theta_3 = 0° \end{bmatrix} \theta_2^+ \rightarrow \begin{bmatrix} a_1 = 1 \\ \theta_1 = 20° \\ a_2 = 0 \\ \theta_2 = -10° \\ a_3 = -2 \\ \theta_3 = 0° \end{bmatrix} a_3^+ \rightarrow \begin{bmatrix} a_1 = 1 \\ \theta_1 = 20° \\ a_2 = 0 \\ \theta_2 = -10° \\ a_3 = -1 \\ \theta_3 = 0° \end{bmatrix} \theta_3^- \rightarrow \begin{bmatrix} a_1 = 1 \\ \theta_1 = 20° \\ a_2 = 0 \\ \theta_2 = -10° \\ a_3 = -1 \\ \theta_3 = -10° \end{bmatrix} a_2^+ \rightarrow \begin{bmatrix} a_1 = 1 \\ \theta_1 = 20° \\ a_2 = 1 \\ \theta_2 = -10° \\ a_3 = -1 \\ \theta_3 = -10° \end{bmatrix}$$

Design of the State Space

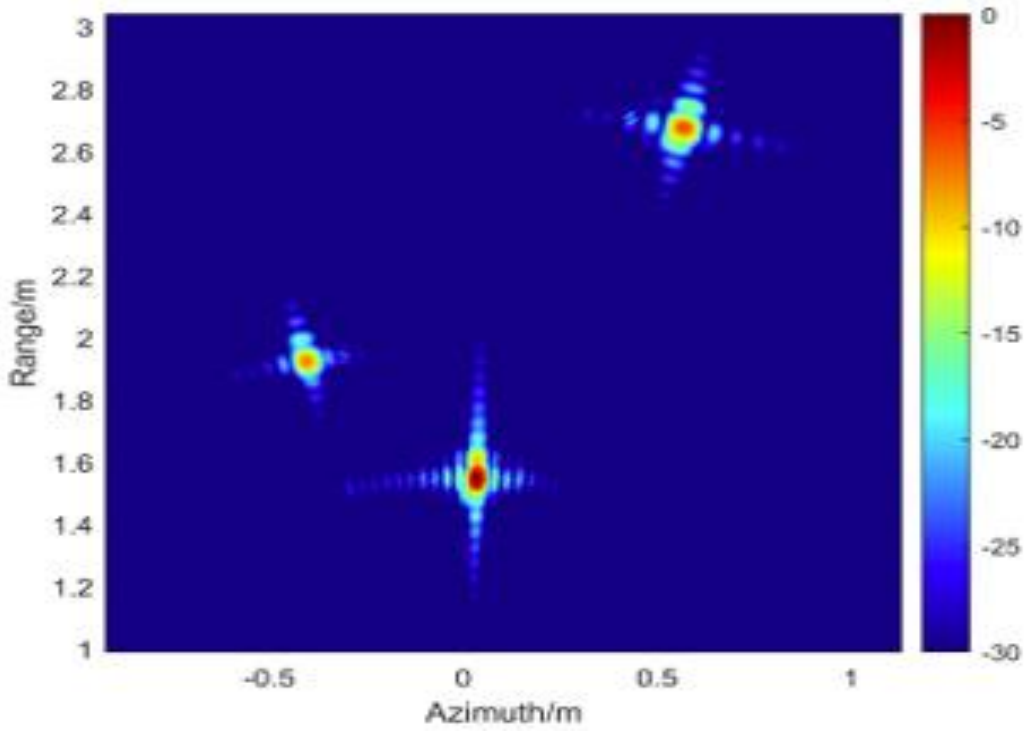
$$\mathcal{S} = \{s_t \in \mathbb{C}^{H \times W} | t \in \mathbb{N}\}$$

$$\mathcal{S} = \{s_t \in \mathbb{R}^{H \times W \times 2} | t \in \mathbb{N}\}$$



Calibration Quality Metric

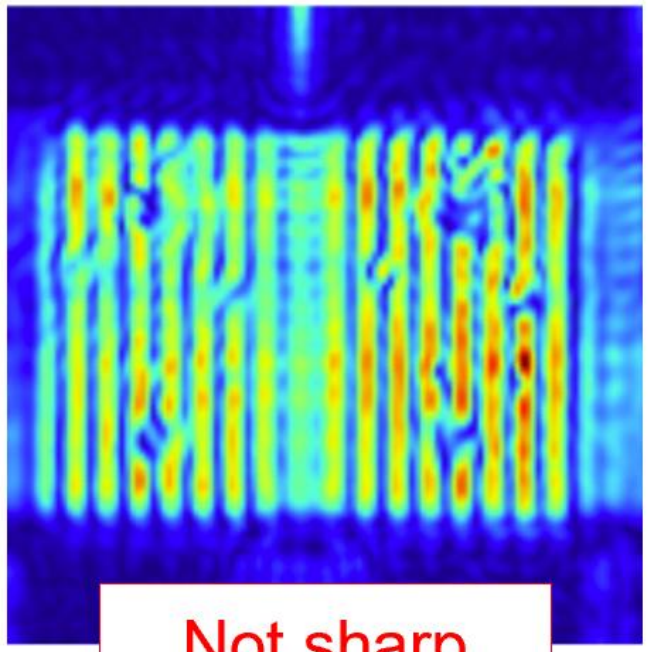
- To define a reward function, we first need to establish what constitutes a well-calibrated image.
- Most radar calibration and autofocus studies focus on sparse point-mass targets.
- Commonly used metrics, such as entropy and sharpness, work well in these cases because they favor images with high-intensity peaks and low energy elsewhere.



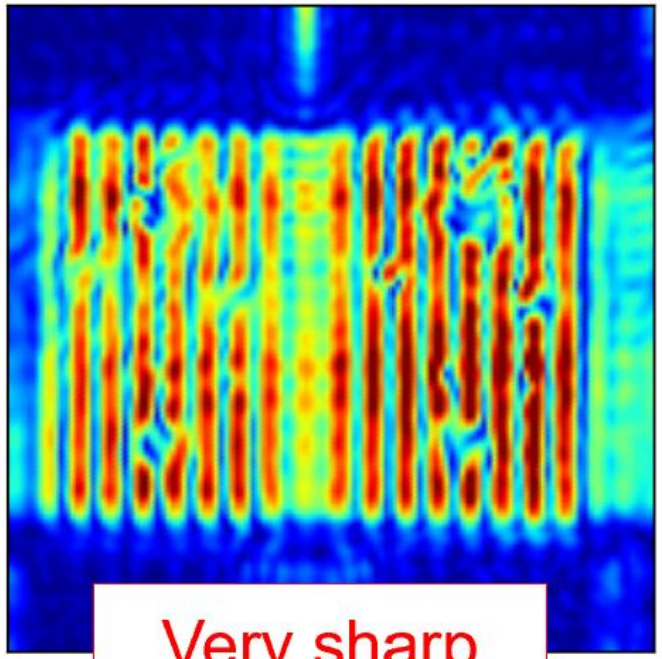
Calibration Quality Metric

- Standard metrics are designed for sparse targets.
- These fail for extended targets.

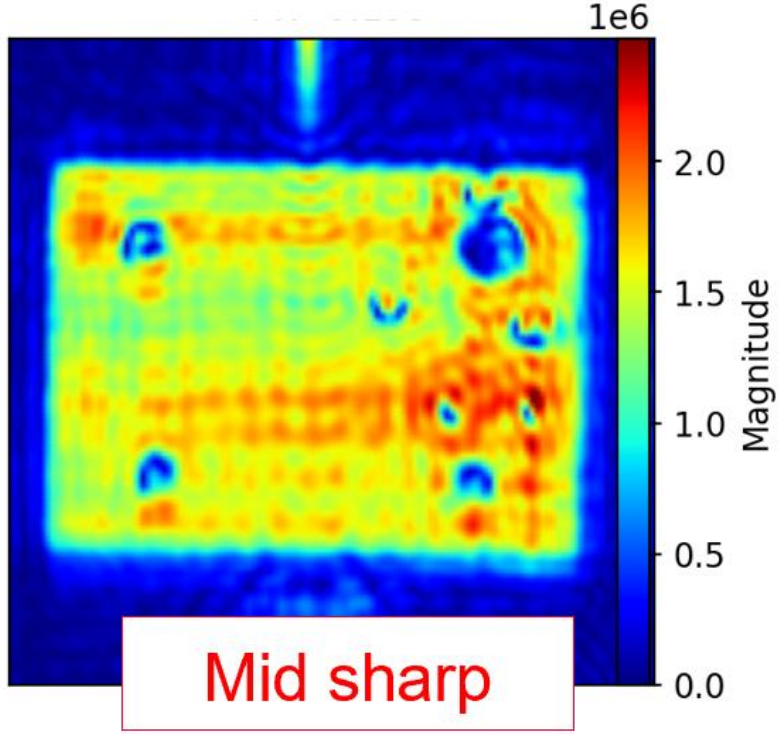
Miscalibrated Radar



Miscalibrated Radar



Well-Calibrated Radar



$$\text{Sharpness} = \sum I_{i,j}^2$$

Calibration Quality Metric

Anisotropic Total Variation (TV):

$$\mathrm{TV}_{\mathrm{L1}}(u) = \sum_{i,j} \left(\left| (\mathrm{D_x} u)_{\mathrm{i,j}} \right| + \left| \left(\mathrm{D_y} u \right)_{\mathrm{i,j}} \right| \right)$$

Isotropic Total Variation (TV):

$$\mathrm{TV}_{\mathrm{L2}}(u) = \sum_{i,j} \sqrt{(\mathrm{D_x} u)_{\mathrm{i,j}}^2 + \left(\mathrm{D_y} u \right)_{\mathrm{i,j}}^2}$$

Calibration Quality Metric

Anisotropic Total Variation (TV):

$$\mathrm{TV}_{\mathrm{L1}}(u) = \sum_{i,j} \left(\left| (\mathrm{D_x} u)_{\mathrm{i,j}} \right| + \left| \left(\mathrm{D_y} u \right)_{\mathrm{i,j}} \right| \right)$$

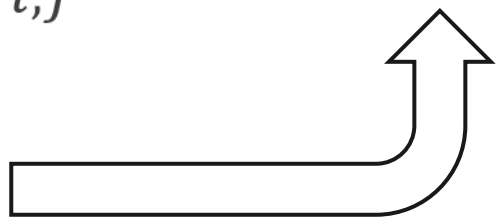
Isotropic Total Variation (TV):

$$\mathrm{TV}_{\mathrm{L2}}(u) = \sum_{i,j} \sqrt{(\mathrm{D_x} u)_{\mathrm{i,j}}^2 + \left(\mathrm{D_y} u \right)_{\mathrm{i,j}}^2}$$

Finite Differences

$$(\mathrm{D_x} u)_{\mathrm{i,j}} = u_{\mathrm{i+1,j}} - u_{\mathrm{i,j}}$$

$$\left(\mathrm{D_y} u \right)_{\mathrm{i,j}} = u_{\mathrm{i,j+1}} - u_{\mathrm{i,j}}$$

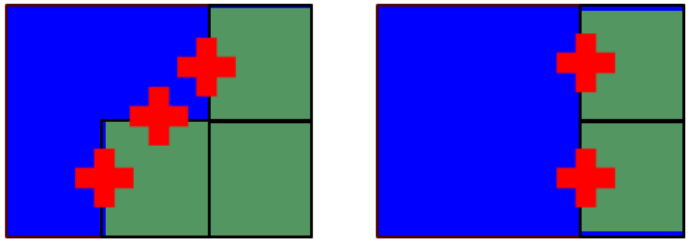


Calibration Quality Metric

Anisotropic Total Variation (TV):

$$\mathrm{TV}_{\mathrm{L1}}(u) = \sum_{i,j} \left(\left| (\mathrm{D_x} u)_{\mathrm{i,j}} \right| + \left| \left(\mathrm{D_y} u \right)_{\mathrm{i,j}} \right| \right)$$

Penalizes curves more



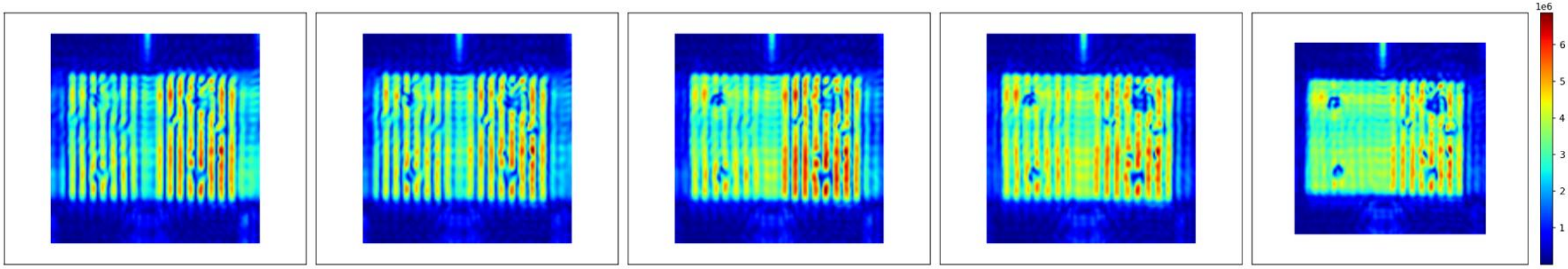
Isotropic Total Variation (TV):

$$\mathrm{TV}_{\mathrm{L2}}(u) = \sum_{i,j} \sqrt{(\mathrm{D_x} u)_{\mathrm{i,j}}^2 + \left(\mathrm{D_y} u \right)_{\mathrm{i,j}}^2}$$

Rotation Invariance

Calibration Quality Metric

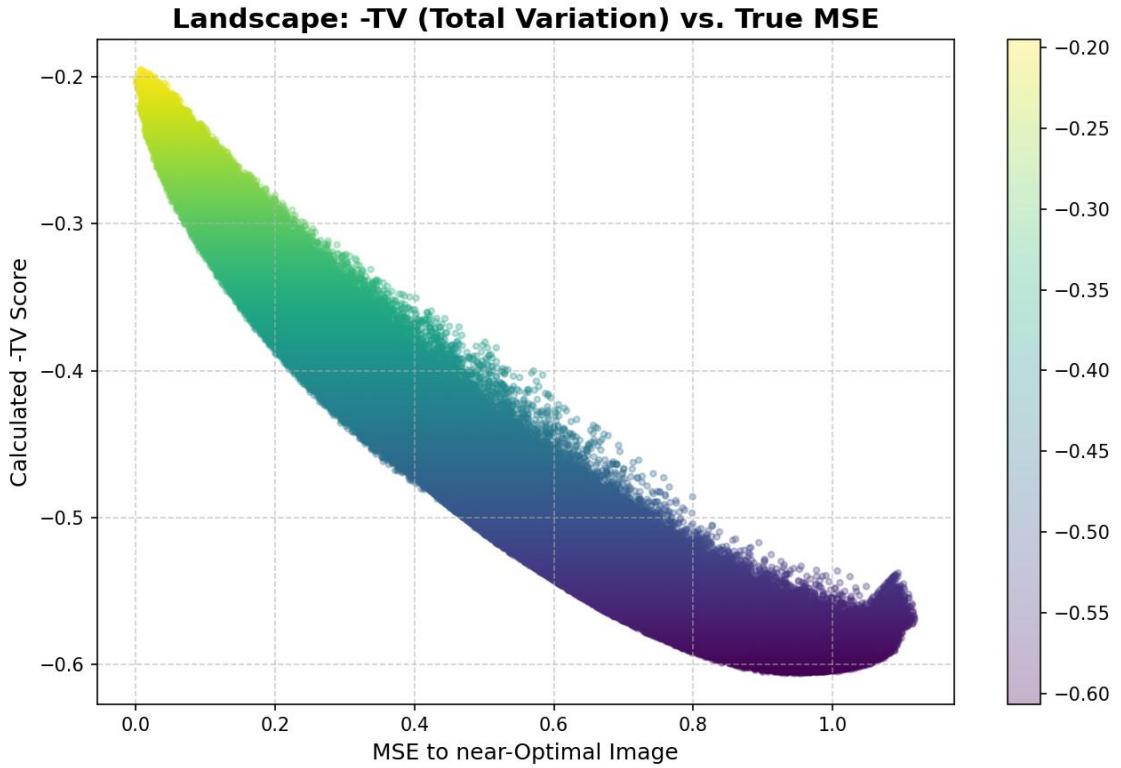
- Minimizing Total Variation (TV) ⇒ Better Quality Calibration

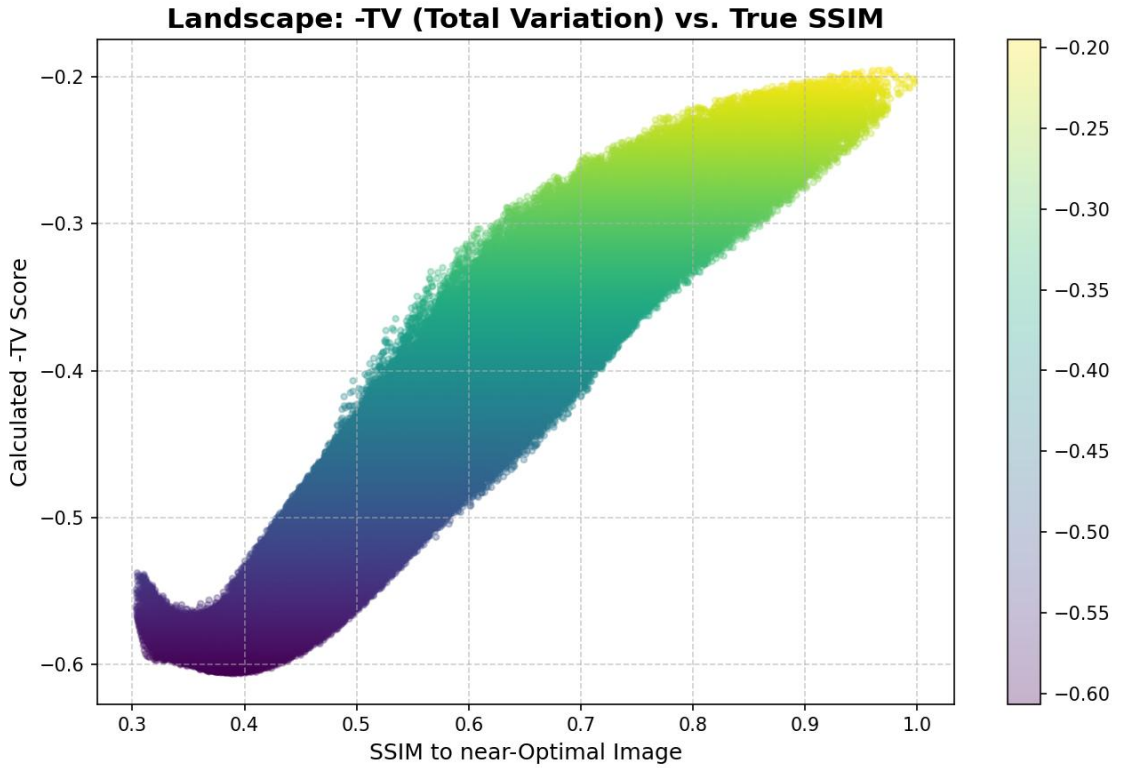


Lower TV (→)

Calibration Quality Metric

- Minimizing Total Variation (TV) ⇒ Better Quality Calibration
- Provides a near-monotonic landscape





Naive Reward Formulation

Dense reward:

$$r_t = \begin{cases} \mathrm{TV}(s_t) - \mathrm{TV}(s_{t+1}), & a_t = \text{move} \\ 0, & a_t = \text{stop} \\ 0, & a_t = \text{wall} \end{cases}$$

Sparse reward:

$$r_t = \begin{cases} 0, & a_t = \text{move} \\ -\mathrm{TV}(s_t), & a_t = \text{stop} \\ 0, & a_t = \text{wall} \end{cases}$$

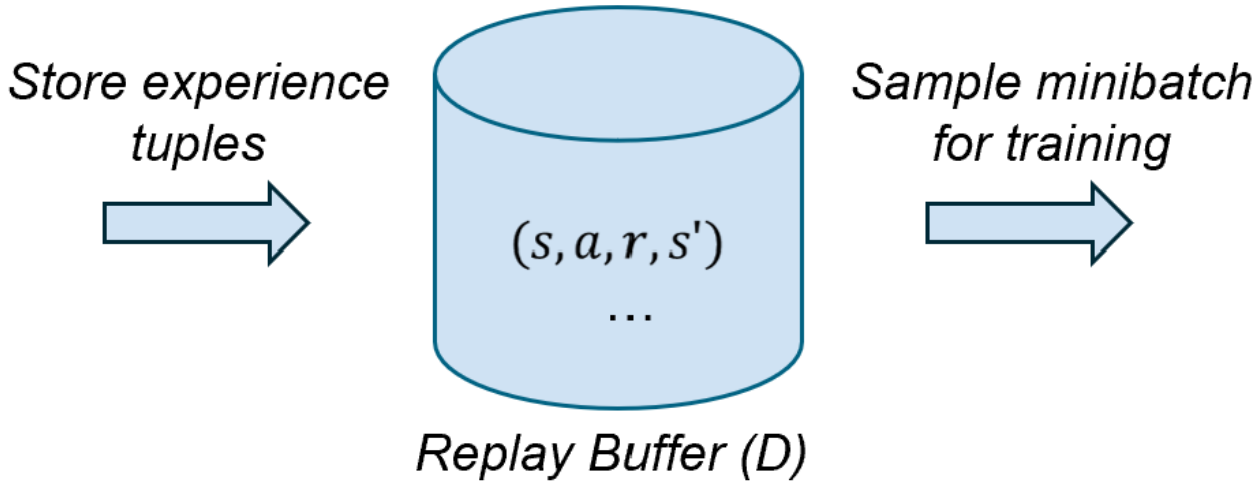
MDP Formulation

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$$

State	The radar image together with the current calibration parameter settings.
Action	A discrete calibration adjustment applied to one of the amplitude or phase parameters. Each action increases or decreases one parameter by one discretization step.
Transition function	The state transition induced by applying a calibration action to the current radar image and parameter setting.
Reward function	The reward evaluates the quality of the calibrated radar image using Total Variation (TV)-based image quality metrics. We introduce different forms of rewards in testing aiming.
Discount factor	A discount factor that controls the importance of future rewards.

Advanced Reinforcement Learning: Offline RL

Offline Learning



Policy evaluation:

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}' \sim \mathcal{D}} \left[\left((r(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}' \sim \hat{\pi}^k(\mathbf{a}'|\mathbf{s}')} [\hat{Q}^k(\mathbf{s}',\mathbf{a}')]) - Q(\mathbf{s},\mathbf{a}) \right)^2 \right]$$

Distributional mismatch

$$\pi(a|s) \neq \pi_{\mathcal{D}}(a|s)$$

Overestimation from out-of-distribution actions

$$\mathbb{E}\left[\max_a \hat{Q}(s,a)\right] \geq \max_a \mathbb{E}\left[\hat{Q}(s,a)\right]$$

Conservative Q-Learning Framework

- Conservative for out-of-distribution actions
- Stable value estimation

$$\mathcal{L}_{TD} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q(s,a) - \left(r + \gamma \max_{a'} Q(s',a')\right)\right)^2\right]$$

$$\mathcal{L}_{CQL} = \mathcal{L}_{TD} + \alpha\left(\mathbb{E}_{s\sim\mathcal{D}}\left[\log \sum_{a} \exp(Q(s,a))\right] - \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[Q(s,a)\right]\right)$$

Preserve dataset actions

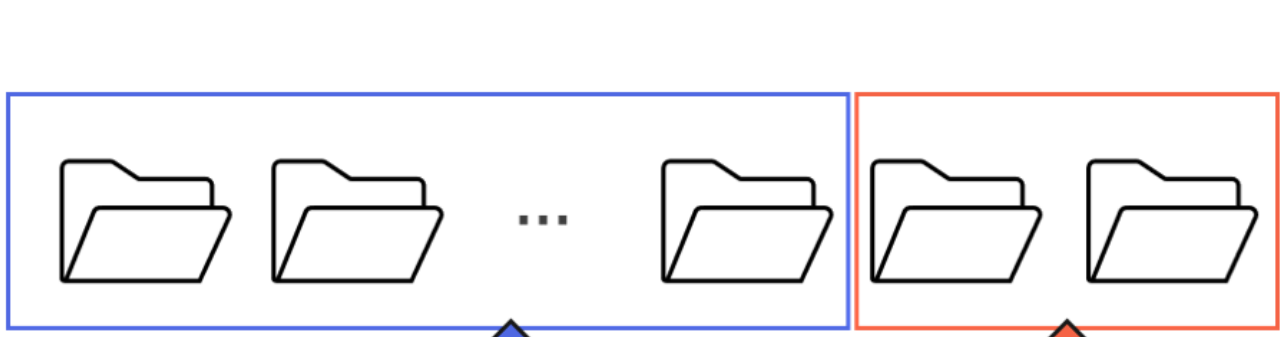
Penalize overly large Q-values

Results and Analysis

Training/Testing Environment

$$Parameter\ span = [-2{:}\,2, -2{:}\,2, -2{:}\,2, -20{:}\,20, -20{:}\,20, -20{:}\,20]$$

$$Action\ space = \mathcal{A} \cup \{Stop\} \quad where\ \mathcal{A} = \left\{a_j^{\pm}, \theta_j^{\pm} \middle| j = 1,2,3\right\}, a_j^{\pm}:\ a_j \leftarrow a_j \pm \Delta_a, \theta_j^{\pm}:\ \theta_j \leftarrow \theta_j \pm \Delta_\theta$$



Training
8 files

Testing
2 files

Each file

- $5^6 = 15{,}625$ unique parameters + images
- Build (s, a, s')−pairs $= 5^6 \cdot 13 = 203{,}125$

Training/Testing Environment

$$Parameter\ span = [-2{:}\,2, -2{:}\,2, -2{:}\,2, -20{:}\,20, -20{:}\,20, -20{:}\,20]$$

$$Action\ space = \mathcal{A} \cup \{Stop\} \quad where\ \mathcal{A} = \left\{ a_j^{\pm}, \theta_j^{\pm} \middle| j = 1, 2, 3 \right\}, a_j^{\pm} :\ a_j \leftarrow a_j \pm \Delta_a, \theta_j^{\pm} :\ \theta_j \leftarrow \theta_j \pm \Delta_\theta$$

Training: d3rlpy Discrete CQL package with custom CNN/Encoder

Main Hyperparameters:

- Number of steps per file: 10,000
- Batch size: 256
- Learning rate: 3e-4
- Discount factor (gamma): 0.99 ~ 0.995
- Conservative penalty weight : 0.1 ~ 1.0

Evaluation: Online-style RL evaluation using comprehensive testing dataset.

Run learned policies 100 times from uniformly random initial states.

Offline RL Configurations

Table 1: Reward configurations.

Config	Description	$r_t \mid a_t = \text{move}$	$r_t \mid a_t = \text{stop}$	$r_t \mid a_t = \text{wall}$	α
Main progression					
1	Dense reward	$-\text{TV}(s_{t+1}) + \text{TV}(s_t)$	0.0	$0.0 + \text{END}$	0.1
2	Sparse reward	0.0	$-\text{TV}(s_t)$	$0.0 + \text{END}$	0.1
3	Shifted (positive) sparse reward	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	0.1
4	Increased α	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	1.0
Ablations from Config 3					
5	+ Wall penalty	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$-1.0 + \text{END}$	0.5
6	Exponential stop reward	0.0	$\left(\frac{\max_s \text{TV}(s) - \text{TV}(s_t)}{\max_s \text{TV}(s) - \min_s \text{TV}(s)}\right)^6$	$0.0 + \text{END}$	0.1

Offline RL Configurations

Table 1: Reward configurations.

Config	Description	$r_t \mid a_t = \text{move}$	$r_t \mid a_t = \text{stop}$	$r_t \mid a_t = \text{wall}$	α
Main progression					
1	Dense reward	$-\text{TV}(s_{t+1}) + \text{TV}(s_t)$	0.0	$0.0 + \text{END}$	0.1
2	Sparse reward	0.0	$-\text{TV}(s_t)$	$0.0 + \text{END}$	0.1
3	Shifted (positive) sparse reward		$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	0.1
4	Increased α		$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	1.0
Ablations from Config 3					
5	+ Wall penalty	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$-1.0 + \text{END}$	0.5
6	Exponential stop reward	0.0	$\left(\frac{\max_s \text{TV}(s) - \text{TV}(s_t)}{\max_s \text{TV}(s) - \min_s \text{TV}(s)}\right)^6$	$0.0 + \text{END}$	0.1

Naive formulations

Offline RL Configurations

Table 1: Reward configurations.

Config	Description	$r_t \mid a_t = \text{move}$	$r_t \mid a_t = \text{stop}$	$r_t \mid a_t = \text{wall}$	α
Main progression					
1	Dense reward	$-\text{TV}(s_{t+1}) + \text{TV}(s_t)$	0.0	$0.0 + \text{END}$	0.1
2	Sparse reward	0.0	$-\text{TV}(s_t)$	$0.0 + \text{END}$	0.1
3	Shifted (positive) sparse reward	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	0.1
4	Increased α	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	1.0
Ablations from Config 3					
5	+ Wall penalty	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$-1.0 + \text{END}$	0.5
6	Exponential stop reward	0.0			

Shifted to “prevent” foreseen reward hacking

Offline RL Configurations

Table 1: Reward configurations.

Config	Description	$r_t \mid a_t = \text{move}$	$r_t \mid a_t = \text{stop}$	$r_t \mid a_t = \text{wall}$	α
Main progression					
1	Dense reward	$-\text{TV}(s_{t+1}) + \text{TV}(s_t)$	0.0	$0.0 + \text{END}$	0.1
2	Sparse reward	0.0	$-\text{TV}(s_t)$	$0.0 + \text{END}$	0.1
3	Shifted (positive) sparse reward	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	0.1
4	Increased α	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	1.0
Ablations from Config 3					
5	+ Wall penalty	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$-1.0 + \text{END}$	0.5
6	Exponential stop reward	0.0	$\left(\frac{\max_s \text{TV}(s) - \text{TV}(s_t)}{\max_s \text{TV}(s) - \min_s \text{TV}(s)}\right)^e$		

Stay in valid states

Offline RL Configurations

Table 1: Reward configurations.

Config	Description	$r_t \mid a_t = \text{move}$	$r_t \mid a_t = \text{stop}$	$r_t \mid a_t = \text{wall}$	α
Main progression					
1	Dense reward	$-\text{TV}(s_{t+1}) + \text{TV}(s_t)$	0.0	$0.0 + \text{END}$	0.1
2	Sparse reward	0.0	$-\text{TV}(s_t)$	$0.0 + \text{END}$	0.1
3	Shifted (positive) sparse reward	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	0.1
4	Increased α	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$0.0 + \text{END}$	1.0
Ablations from Config 3					
5	+ Wall penalty	0.0	$\max_s \text{TV}(s) - \text{TV}(s_t)$	$-1.0 + \text{END}$	0.5
6	Exponential stop reward	0.0	$\left(\frac{\max_s \text{TV}(s) - \text{TV}(s_t)}{\max_s \text{TV}(s) - \min_s \text{TV}(s)}\right)^6$	$0.0 + \text{END}$	0.1

Place strong emphasis on near-optimal calibrations

Training Results

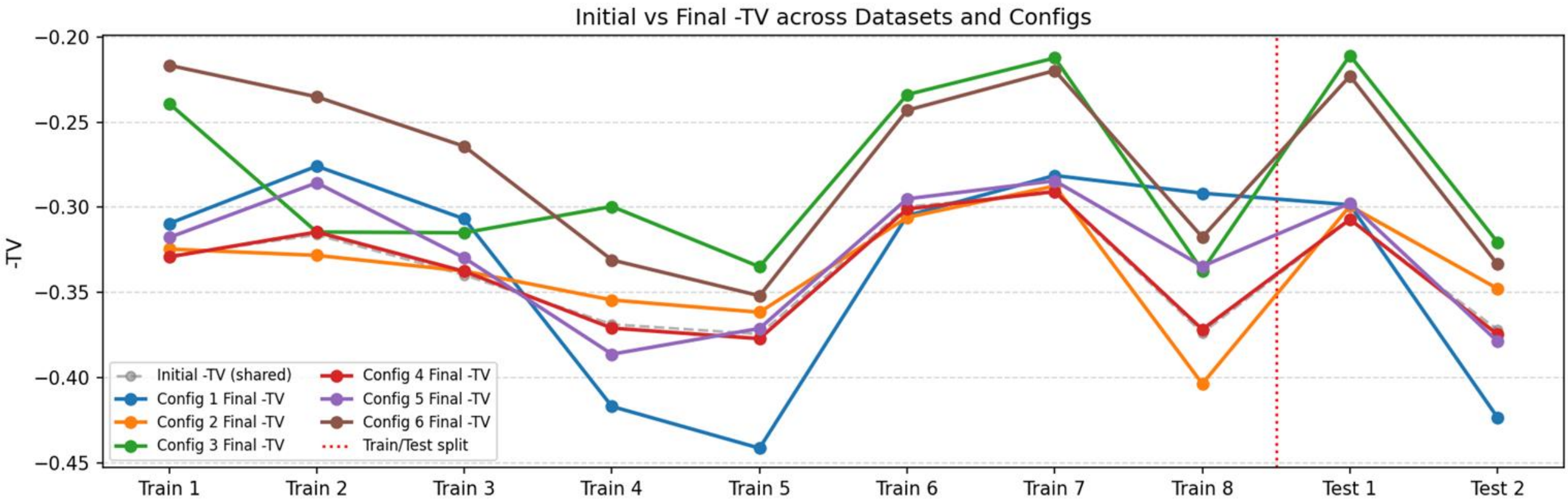




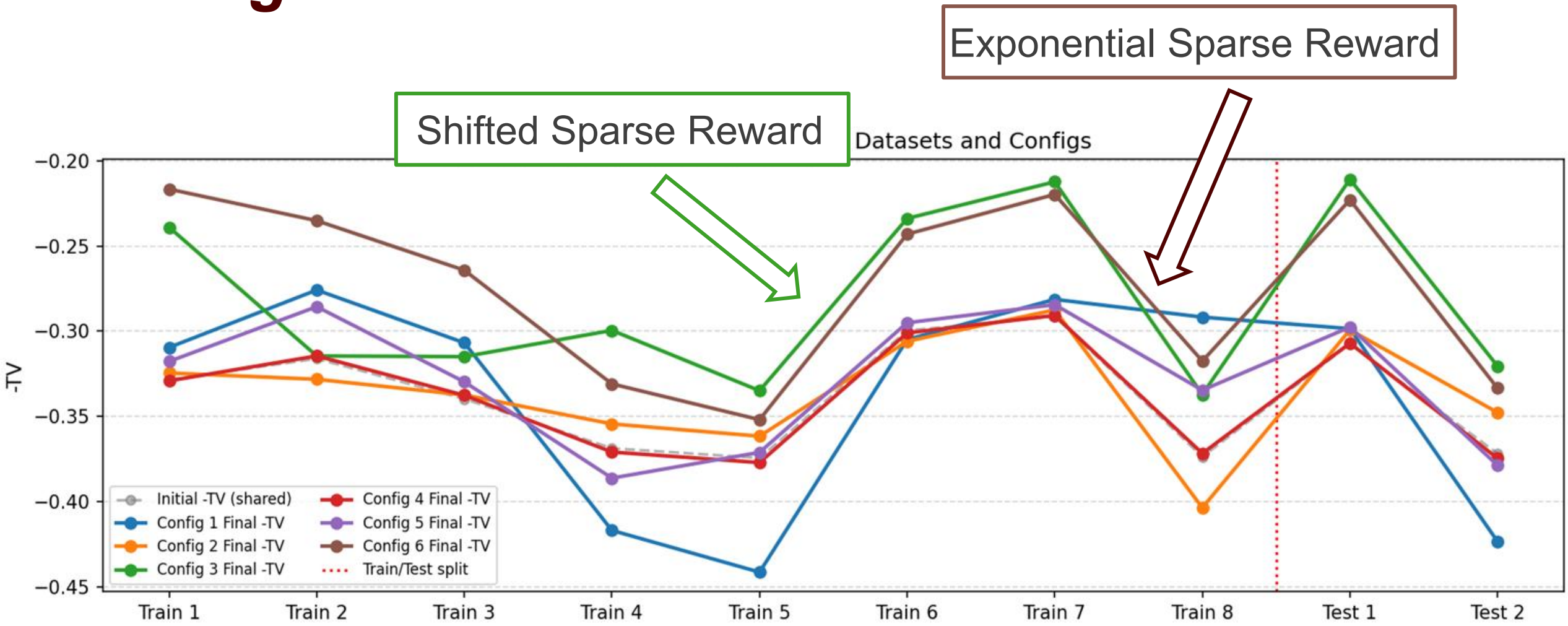
Example training curves from Config 4



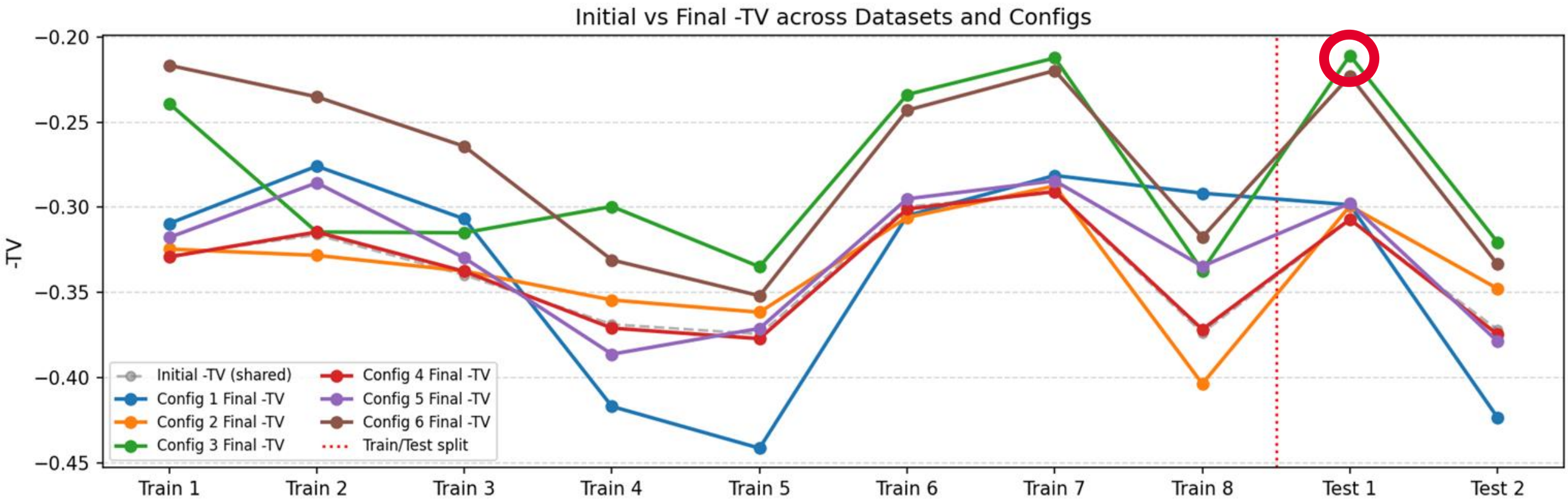
Testing Results



Testing Results



Testing Results



Conclusion

Conclusion

- Through this work, we propose a reinforcement learning-based framework to compensate for errors introduced during the MIMO radar manufacturing process.
- To achieve this, we propose using image quality metrics as reward functions in reinforcement learning.
- Implemented an offline RL pipeline for MIMO radar calibration using pre-collected transition data.
- Designed tailored quality evaluation metric and shifted sparse reward to improve training stability and reduce oscillatory behavior.

Future Work

- This study serves as an initial investigation into the feasibility of reinforcement learning-based MIMO radar calibration and demonstrates the viability of the proposed framework.
- However, this study has two main limitations: the action and state spaces are discrete, and the framework has only been validated through emulation.
- Future work is needed to address these limitations by extending the framework to continuous action and state spaces and validating its performance in real-world experiments.



TEXAS A&M UNIVERSITY

Engineering

MiWiRa Lab

THANK YOU

References

Paper Title	Category	Link
OPTICAL AND SAR IMAGE REGISTRATION WITH DEEP REINFORCEMENT LEARNING	Applying reinforcement learning to radar imaging field	Optical_and_SAR_ ge_Registration_w
Reinforcement Learning based Waveform Design for Cognitive Imaging Radar	Applying reinforcement learning to radar imaging field	Reinforcement_Le g_based_Waveforı
Radar Waveform Parameter Optimization with Offline Reinforcement Learning Strategy	Applying reinforcement learning to radar imaging field	Radar_Waveform_ meter_Optimizatio
Non-linear redundancy calibration	Radar calibration techniques	Non_linear_redun dancy_calibration
Fast Dynamic Calibration of Millimeter-Wave Radar External Parameters: Research and Applications	Radar calibration techniques	Fast_Dynamic_Cal ıtion_of_Millimeteı
Automatic delay and phase mismatch calibration in FMCW MIMO radar	Radar calibration techniques	Automatic_delay_ _phase_mismatch_
An FMCW MIMO Radar Calibration and Mutual Coupling Compensation Approach	Radar calibration techniques	An_FMCW_MIMO _calibration_and_r