Data-Driven Target Localization Using Adaptive Radar Processing and Convolutional Neural Networks

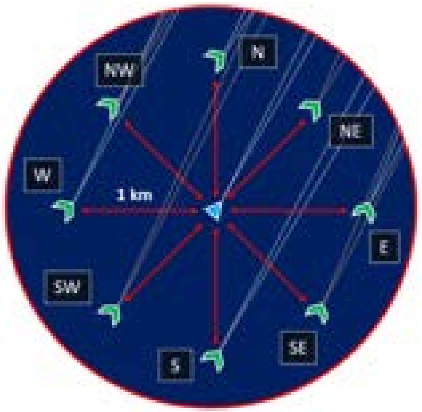
Shyam Venkatasubramanian[1], Sandeep Gogineni[2], Bosung Kang[3], Ali Pezeshki[4], Muralidhar Rangaswamy[5], Vahid Tarokh[1]

[1] Duke University
[2] Information Systems Laboratories, Inc.
[3] University of Dayton Research Institute
[4] Colorado State University
[5] U.S. Air Force Research Laboratory

This work is supported by the AFOSR under award FA9550-21-1-0235. Dr. Rangaswamy, and Dr. Kang are supported by the AFOSR under project 20RYCOR051. Dr. Gogineni is supported by the AFOSR under project 20RYCOR052.

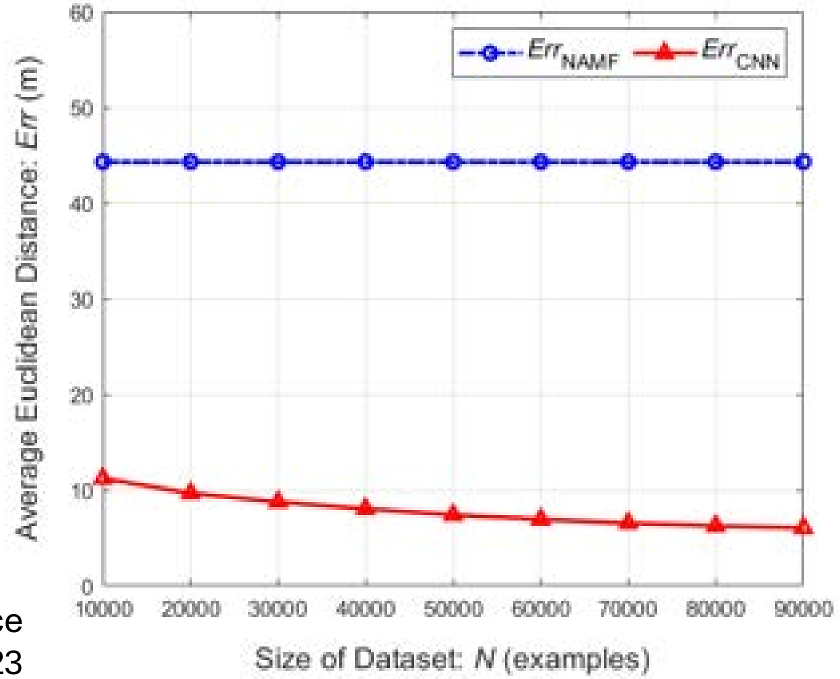
Summary of Recent Work and Benchmarking Radar STAP Dataset

- Benchmarked CNN vs. Normalized Adaptive Matched Filter (gold standard of model-based approaches)
  - **No mismatches in radar scenario**: CNN has **10-fold gain in localization accuracy** over NAMF (**45 m to 5 m**)
- Showed that CNN can be made **robust to mismatches** (perturbations) in radar scenarios
  - Validated the efficacy of chordal distance as a task affinity measure to benchmark CNN gain over NAMF
  - Developed a **transfer learning (TL) approach** to make CNN robust to mismatches in the radar scenario
    - Using transfer learning, the CNN recovers up to a **4-fold gain** over NAMF (**45 m to 11 m**)
- Began construction of **Benchmarking Radar STAP Dataset** (ImageNet of radar STAP) using RFView®
  - **10,000 total scenarios** in dataset, each consisting of **10,000 clutter + noise realizations**
  - LBG algorithm finds **100 representative scenarios** (pre-trained CNN provided) from 10,000 total scenarios
    - **To use dataset:** User selects pre-trained CNN from scenario with minimal task affinity measure with respect to their own scenario
    - User can **fine-tune** this CNN via transfer learning w/ few samples



D	N	NW	W	SW	S	SE	E	NE
$\hat{d}_{\text{chordal}}$	0.31	0.31	0.34	0.45	0.51	0.55	0.54	0.47
CNN Gain	1.65	1.63	1.41	1.02	0.91	0.84	0.86	1.01
CNN Gain w/ TL	4.05	3.97	3.92	3.87	3.85	3.8	3.81	3.88

Radar platform displaced in each cardinal & intermediate direction
- **Pairwise Agreement** between chordal distance and CNN Gain
- **Transfer Learning (TL)** with few samples recovers CNN Gain



S. Venkatasubramanian, S. Gogineni, B. Kang, A. Pezeshki, M. Rangaswamy, and V. Tarokh, "Subspace Perturbation Analysis for Data-Driven Radar Target Localization," *IEEE Radar Conference*, pp. 1–5, 2023

Outline

1) Introduction and Background
   - Data-Driven Target Localization
   - Problem Statement

2) Overview of Methods
   - Robust Mismatched Case Example Scenario
   - Subspace Perturbation Analysis

3) Empirical Results
   - Regression CNN Framework
   - Robust Mismatched Case Results

4) Conclusion

5) Addendum: Doppler Processing Example

Outline

1) Introduction and Background
   - Data-Driven Target Localization
   - Problem Statement
2) Overview of Methods
   - Robust Mismatched Case Example Scenario
   - Subspace Perturbation Analysis
3) Empirical Results
   - Regression CNN Framework
   - Robust Mismatched Case Results
4) Conclusion
5) Addendum: Doppler Processing Example

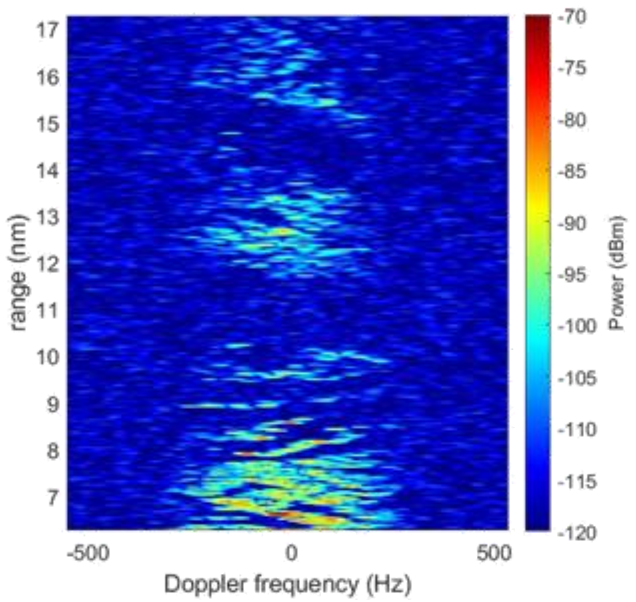
Target Localization

- Target Localization is crucial to the design of modern radar systems
  - Foundational to surveillance, navigation, and military operations
  - Defined post radar STAP detection

Limitations of Radar STAP Detection & Localization:

- Possible target/clutter subspace overlap
  - Projecting out clutter removes target response [1]

- Clutter interference unknown a priori
  - Clutter covariance estimated with limited data [2]
  - Lends to suboptimal detection (adaptive case)
  - Weakens optimality claims regarding target localization



Emergence of Big Data in Radar

- Follows from recent emergence of big data in computer vision
  - Several publicly available databases for natural images:
    - CIFAR-10/100 [3], COCO [4], ImageNet [5]
    - Allow for the training and testing of computer vision algorithms

<u>Big Data in Radar STAP Applications</u>:

- Detailed satellite topographic maps are now readily available.

- Can we use the side-information provided by this data – instead of idealized mathematical models – to improve radar STAP performance?

- Need <u>simulators</u> that can use the data for modeling – i.e., **RFView®**

RFView® Digital Twin Software

- Developed by ISL Inc.
  - Splatter Clutter and Target Signal engine has been in commercial use since 1989
  - Supports efficient real-time instantiation of RF environments[1]
  - Extensively validated against measured datasets
- User-defined simulation scenarios
  - Specify parameters and targets + land cover data provided
- High-fidelity data generation [6]
  - Simulate radar return & generate target, clutter, and noise data

[1] Training the regression convolutional neural networks for our analysis requires massive noise-free datasets, which we generate by rapidly instantiating dynamically varying environments in RFView®

Heatmap Tensor Generation – Signal Model

- Signal model derived from radar STAP detection problem
  - Hypothesis testing on radar return (matched filtered to range bin ρ):

$$H_0 : \mathbf{Z}_\rho = \bar{\mathbf{C}}_\rho + \bar{\mathbf{N}}_\rho \qquad\qquad H_1 : \mathbf{Y}_\rho = \mathbf{X}_\rho + \mathbf{C}_\rho + \mathbf{N}_\rho$$

$\mathbf{Z}_\rho$ and $\mathbf{Y}_\rho$ consist of K realizations of the radar return and clutter-plus-noise data
$\bar{\mathbf{C}}_\rho, \mathbf{C}_\rho$ are unique and consist of K realizations of the clutter data
$\bar{\mathbf{N}}_\rho, \mathbf{N}_\rho$ are unique and consist of K realizations of the noise data
$\mathbf{a}_\rho(\theta, \phi^*)$ is the steering vector associated w/ coordinates $\left(r_\rho, \theta, \phi^*\right)$

Notation: $\boldsymbol{r_\rho}$ = range (of range bin ρ), $\boldsymbol{\theta}$ = azimuth, $\boldsymbol{\phi}^*$ = elevation of target (known)

Heatmap Tensor Generation

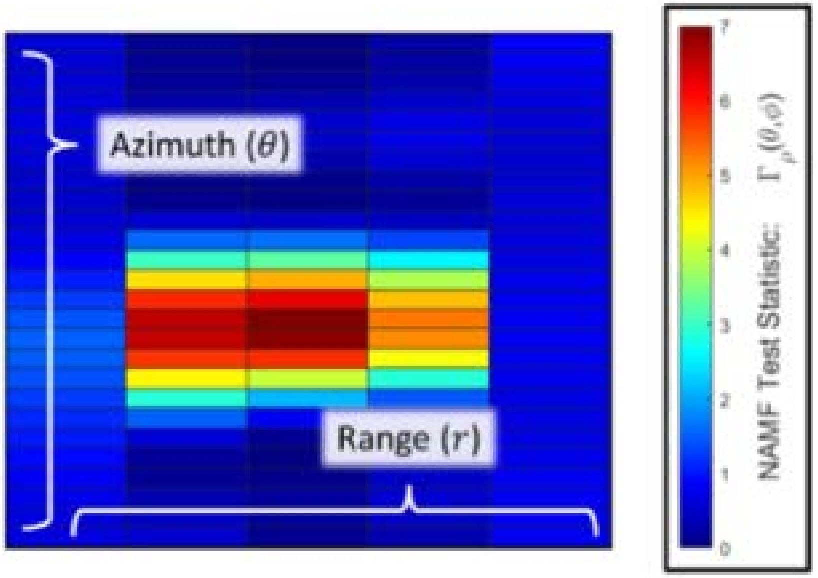
- The NAMF Test Statistic, $\Gamma_\rho(\theta) \in \mathbb{R}^+$, for coordinates $\left(r_\rho, \theta, \phi^*\right)$ is given by:

$$\Gamma_\rho(\theta) = \frac{\left\|\tilde{\mathbf{a}}_\rho(\theta,\phi^*)^H \boldsymbol{\Sigma}^{-1} \tilde{\mathbf{Y}}_\rho\right\|_2^2}{\left[\tilde{\mathbf{a}}_\rho(\theta,\phi^*)^H \boldsymbol{\Sigma}^{-1} \tilde{\mathbf{a}}_\rho(\theta,\phi^*)\right] \left\|\mathrm{diag}\left(\tilde{\mathbf{Y}}_\rho^H \tilde{\mathbf{Y}}_\rho\right)\right\|}$$

where: $\Sigma = \left(\mathbf{Z}_\rho \mathbf{Z}_\rho^H\right)/\mathrm{K}$

<u>Heatmap Tensor Generation</u>:

- For each target configuration (N total):
    - For each range bin (ρ) from constrained area:
        - Sweep across azimuth with step size $(\Delta\theta)$
            - Record NAMF test statistic, $\Gamma_\rho(\theta)$



<u>Result</u>: For $(\Delta r, \Delta\theta) = (30\ \mathrm{m}, 0.4°)$

- 2D Heatmap Tensor of size 5×26

9

Average Euclidean Distance Metric

- Comparing Regression CNN with traditional method (NAMF):
  - **Metric**: Euclidean distance (averaged across test dataset)

$$Err_{\text{CNN}} = \frac{\sum_{j=1}^{N_{eval}} \|(x_j^*, y_j^*) - (\hat{x}_j, \hat{y}_j)\|_2}{N_{eval}}$$

$$Err_{\text{NAMF}} = \frac{\sum_{j=1}^{N_{eval}} \|(x_j^*, y_j^*) - (\bar{x}_j, \bar{y}_j)\|_2}{N_{eval}}$$

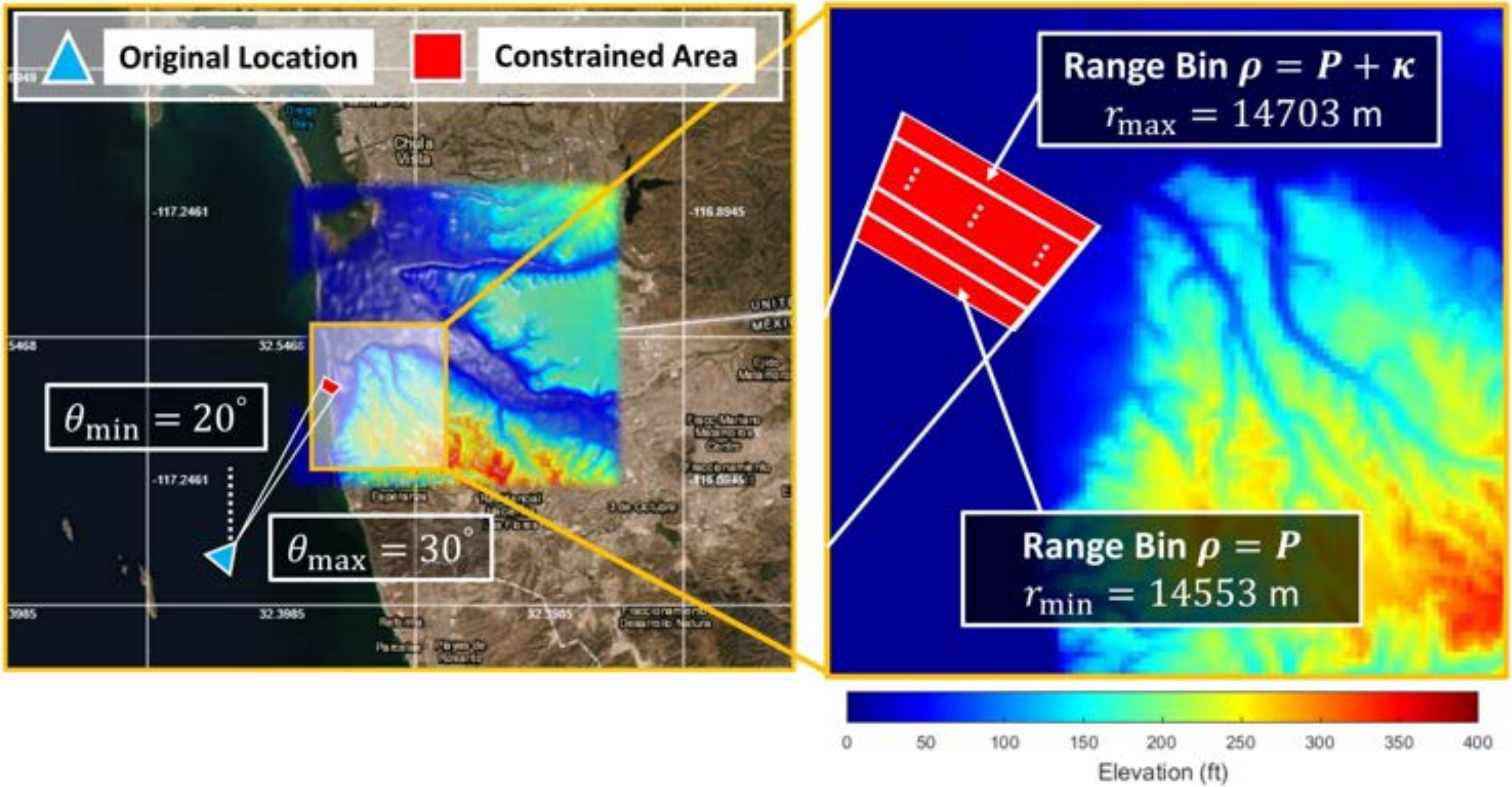
<u>Regression CNN</u>: Err_{CNN}

- Estimated target location vs. ground truth target location

<u>Traditional Method</u>: Err_{NAMF}

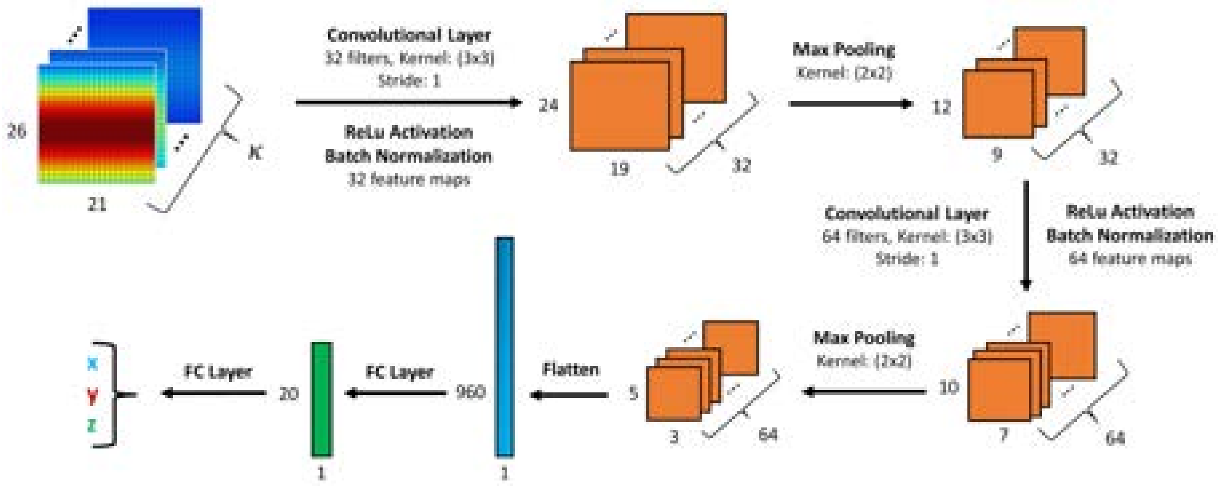
- Midpoint of cell with peak test statistic in each heatmap tensor vs. ground truth target location

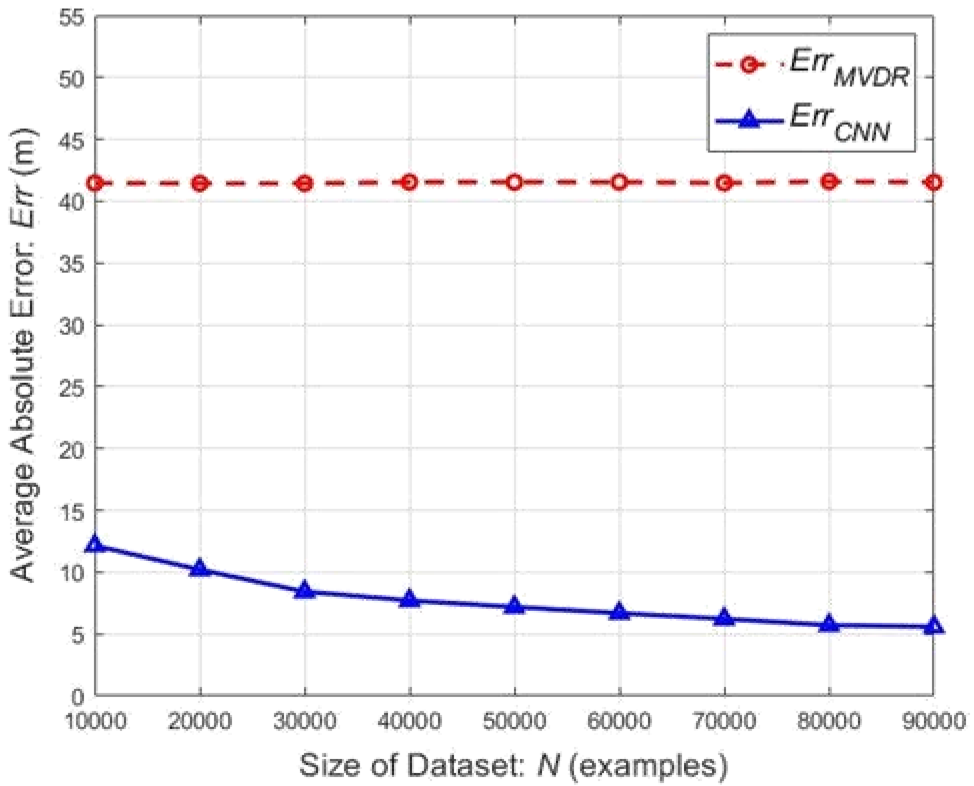
Matched Case RFView® Example Scenario (Previous Result)



Regression CNN vs. Traditional Method (Previous Result)

$0.9N$ training examples, $0.1N$ test examples, SCNR = 20 dB





12

Problem Statement

- Detection statistics observe **suboptimal performance** when the clutter covariance matrix is estimated from limited data (radar returns)
  - **Diminishes target localization accuracy** post radar STAP detection

<u>Data-Driven Approach</u>:

- Can we use **deep learning** to improve target localization performance?
  - **Traditional approach**: Estimating the peak location from a detection test statistic
  - Baseline CNN [7] achieves significant gains over traditional approach in matched settings. **Can CNN performance across mismatched settings be predetermined**?
    - Perform a **Subspace Perturbation Analysis** to benchmark CNN performance
  - Consider robust mismatched case scenario in RFView® for further evaluation

Outline

1) Introduction and Background
   - Data-Driven Target Localization
   - Problem Statement

2) Overview of Methods
   - Robust Mismatched Case Example Scenario
   - Subspace Perturbation Analysis

3) Empirical Results
   - Regression CNN Framework
   - Robust Mismatched Case Results

4) Conclusion

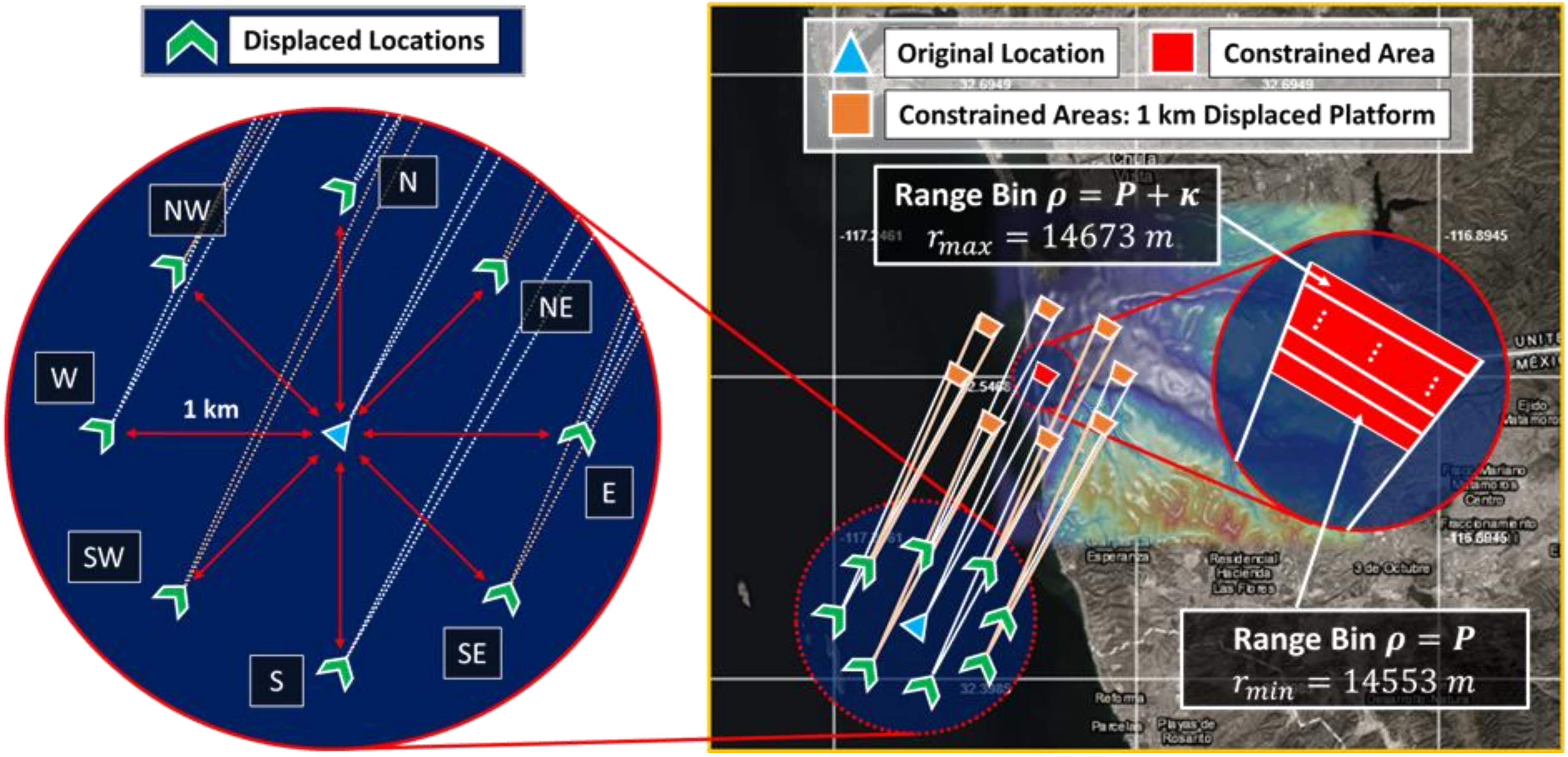
5) Addendum: Doppler Processing Example

Mismatched RFView® Example Scenarios

- Airborne radar platform system flying over coastal Southern California
  - We consider a single-channel transmitter and an $L = 16$-channel receiver
- $D = 2$ dimensions exploited for processing: Range, $\boldsymbol{r}$, and Azimuth, $\boldsymbol{\theta}$
- Each radar return is produced using $\Lambda = 1$ transmitted pulses
  - RFView® aggregates topography & radar parameters to simulate radar return
  - Targets are stationary ($v = 0$ m/s) and target elevation, $\boldsymbol{\phi}^*$, is known beforehand

Parameters	Values
Carrier frequency	$10,000$ MHz
Bandwidth	5 MHz
PRF & Duty Factor	1100 Hz & 10
Receiving antenna	16×1 (horizontal $\times$ vertical elements)
Transmitting antenna	48×5 (horizontal $\times$ vertical elements)
Antenna element spacing	0.015 m
Platform height	1000 m
Area latitude (min, max)	$(32.4611°, 32.6399°)$
Area longitude (min, max)	$(-117.1554°, -116.9433°)$

Mismatched RFView® Example Scenarios



Mismatched RFView® Example Scenarios

<u>Original Platform Location Instance</u> (O):

- Target randomly placed within red constrained area
  - RCS randomly chosen from uniform distribution w/ set mean (range = 10 dBsm)
  - Generate $\boldsymbol{K = 100}$ **independent radar returns** using RFView ($L = 16$ channels, $\Lambda = 1$ pulses) and <u>transform into heatmap tensor</u> via the NAMF test statistic
  - **Repeat $\boldsymbol{N}$ times** to obtain $\boldsymbol{N}$ **heatmap tensors** (examples in our dataset).
    - Heatmap tensor = 2D representation of red constrained area

- Approximate constrained area specifications:
  - Contains κ range bins, where $\kappa =$ **Depth Parameter**
  - Default grid resolution: $(\Delta r, \Delta\theta)$, where $\Delta r =$ **Chip Size**

Original Location (O) – Parameters	Values
Platform latitude, longitude	$32.4275°, -117.1993°$
Constrained area range $(r_{\min}, r_{\max})$	$(14553\ \mathrm{m}, 14673\ \mathrm{m})$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

Mismatched RFView® Example Scenarios

<u>Displaced Platform Location Instances</u> (D):

- **(D)** ∈ **{**1 km North **(N),** 1 km Northwest **(NW),** 1 km West **(W),** 1 km Southwest **(SW),** 1 km South **(S),** 1 km Southeast **(SE),** 1 km East **(E)**, 1 km Northeast **(NE)}:**

- Generate **$0.1N$ heatmap tensors** for each **(D)**
  - <u>Repeat the following $0.1N$ times</u>:
  - Target randomly placed within **orange** constrained area
  - RCS randomly chosen from uniform distribution w/ set mean (range = 10 dBsm)
    - Generate $\boldsymbol{K = 100}$ **independent radar returns** using RFView ($L = 16,\ \Lambda = 1$) and <u>transform into heatmap tensor</u> via the NAMF test statistic
      - Heatmap tensor = 2D representation of orange constrained area

<u>**Notation**</u>: $\boldsymbol{\Lambda}$ = number of pulses, $\boldsymbol{L}$ = number of channels, $\boldsymbol{K}$ = number of realizations

Mismatched RFView® Example Scenarios

Displaced Platform Location Instance (D):

- Approximate constrained area specifications:
  - Contains κ range bins, default grid resolution: $(\Delta r, \Delta\theta)$

1 km North (N) – Parameters	Values
Platform latitude, longitude	$32.4095°, -117.1993°$
Constrained area range $(r_{\min}, r_{\max})$	$(13800$ m$, 13920$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km South (S) – Parameters	Values
Platform latitude, longitude	$32.3915°, -117.1993°$
Constrained area range $(r_{\min}, r_{\max})$	$(15321$ m$, 15441$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km Northwest (NW) – Parameters	Values
Platform latitude, longitude	$32.4070°, -117.1920°$
Constrained area range $(r_{\min}, r_{\max})$	$(14467$ m$, 14587$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km Southeast (SE) – Parameters	Values
Platform latitude, longitude	$32.3940°, -117.1920°$
Constrained area range $(r_{\min}, r_{\max})$	$(14680$ m$, 14800$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km West (W) – Parameters	Values
Platform latitude, longitude	$32.4005°, -117.2099°$
Constrained area range $(r_{\min}, r_{\max})$	$(15207$ m$, 15327$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km East (E) – Parameters	Values
Platform latitude, longitude	$32.4005°, -117.1887°$
Constrained area range $(r_{\min}, r_{\max})$	$(13921$ m$, 14041$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km Southwest (SW) – Parameters	Values
Platform latitude, longitude	$32.3940°, -117.2066°$
Constrained area range $(r_{\min}, r_{\max})$	$(15544$ m$, 15664\ )$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

1 km Northeast (NE) – Parameters	Values
Platform latitude, longitude	$32.4070°, -117.1920°$
Constrained area range $(r_{\min}, r_{\max})$	$(13558$ m$, 13678$ m$)$
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	$(20°, 30°)$

Proof-of-concept – Subspace Perturbation Analysis

- **Goal**: Measure similarity between clutter-plus-noise (clutter) subspaces of $(\mathbf{O})$ and $(\mathbf{D}) \in \{\mathbf{N}, \mathbf{NW}, \mathbf{W}, \mathbf{SW}, \mathbf{S}, \mathbf{SE}, \mathbf{E}, \mathbf{NE}\}$

- Consider clutter-plus-noise data matrices of $(\mathbf{O})$ & $(\mathbf{D})$:
  - $\mathbf{Z}_\rho^{(\mathbf{O})} = \bar{\mathbf{C}}_\rho^{(\mathbf{O})} + \bar{\mathbf{N}}_\rho^{(\mathbf{O})} \in \mathbb{C}^{(\Lambda \cdot L)\times K}$ $\mathbf{Z}_\rho^{(\mathbf{D})} = \bar{\mathbf{C}}_\rho^{(\mathbf{D})} + \bar{\mathbf{N}}_\rho^{(\mathbf{D})} \in \mathbb{C}^{(\Lambda \cdot L)\times K}$

- The chordal distance between $\bar{\mathbf{C}}_\rho^{(\mathbf{O})}$ and $\bar{\mathbf{C}}_\rho^{(\mathbf{D})}$ is denoted: $\hat{\mathbf{d}}_{\mathbf{chordal}}$
  - $\hat{\mathbf{d}}_{\mathbf{chordal}}$ measures the similarity between $(\mathbf{O})$ and $(\mathbf{D})$

- Derived $\hat{\mathbf{d}}_{\mathbf{chordal}}$ between $\bar{\mathbf{C}}_\rho^{(\mathbf{O})}$ and $\bar{\mathbf{C}}_\rho^{(\mathbf{D})}$ for $\Lambda = 1, L = 16, K = 100$:

D	**N**	**NW**	**W**	**SW**	**S**	**SE**	**E**	**NE**
$\hat{\mathrm{d}}_{\mathrm{chordal}}$	0.31	0.31	0.34	0.45	0.51	0.55	0.54	0.47

Notation: $\boldsymbol{\Lambda}$ = number of pulses, $\boldsymbol{L}$ = number of channels, $\boldsymbol{K}$ = number of realizations

Outline

1) Introduction and Background
   - Data-Driven Target Localization
   - Problem Statement

2) Overview of Methods
   - Robust Mismatched Case Example Scenario
   - Subspace Perturbation Analysis

3) Empirical Results
   - Regression CNN Framework
   - Robust Mismatched Case Results

4) Conclusion

5) Addendum: Doppler Processing Example

Regression CNN Framework

Regression CNN for target localization $(\Lambda = 1, L = 16, K = 100)$

Original Platform Location Instance $(\mathbf{O})$:

- The original platform location instance dataset has: $\boxed{N = 9 \times 10^4 \text{ training examples}}$

Displaced Platform Location Instances $(\mathbf{D}) \in \{\mathbf{N}, \mathbf{NW}, \mathbf{W}, \mathbf{SW}, \mathbf{S}, \mathbf{SE}, \mathbf{E}, \mathbf{NE}\}$:

- Each displaced platform location instance has a dataset with $\boxed{0.1N \text{ test examples}}$

- Fit Regression CNN with training dataset
  - Learn target locations $(r, \theta) \rightarrow (x, y)$ from the $0.9N$ heatmap tensors

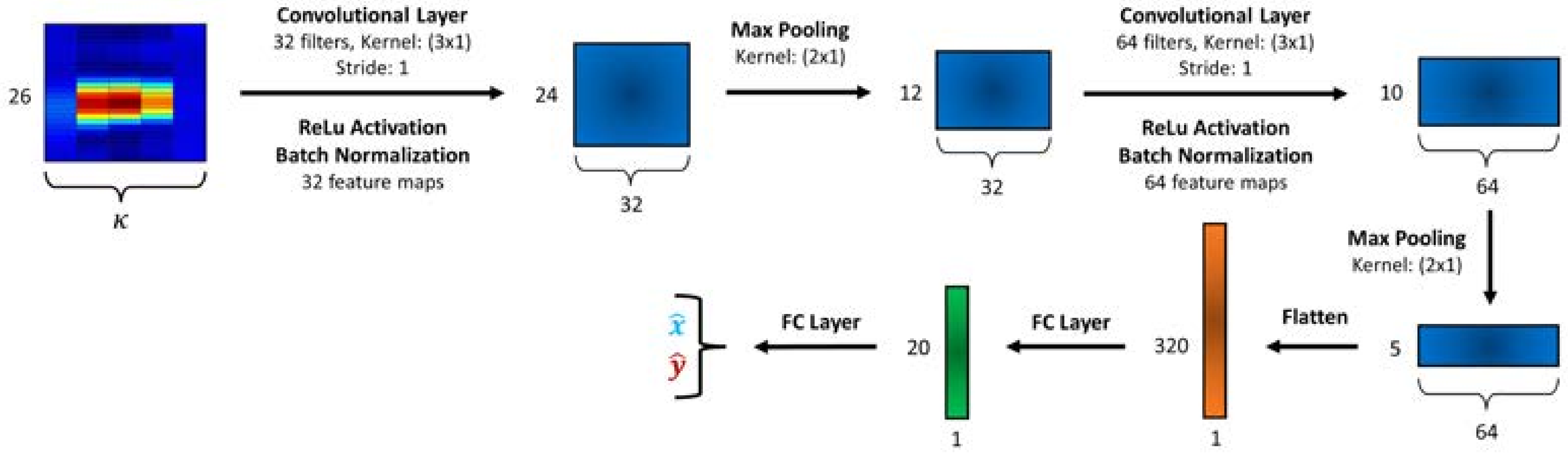
- Evaluate Regression CNN on test datasets
  - Estimate target location $(\hat{r}, \hat{\theta}) \rightarrow (\hat{x}, \hat{y})$ for each of the $0.1N$ heatmap tensors

Regression CNN Framework – Baseline CNN

- Used for evaluations with default grid resolution: $(\Delta r, \Delta\theta) = (30\ \mathrm{m}, 0.4°)$

  Tensor dimensions $= \mathbf{5 \times 26}$ Output $= (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ **coordinates of target**

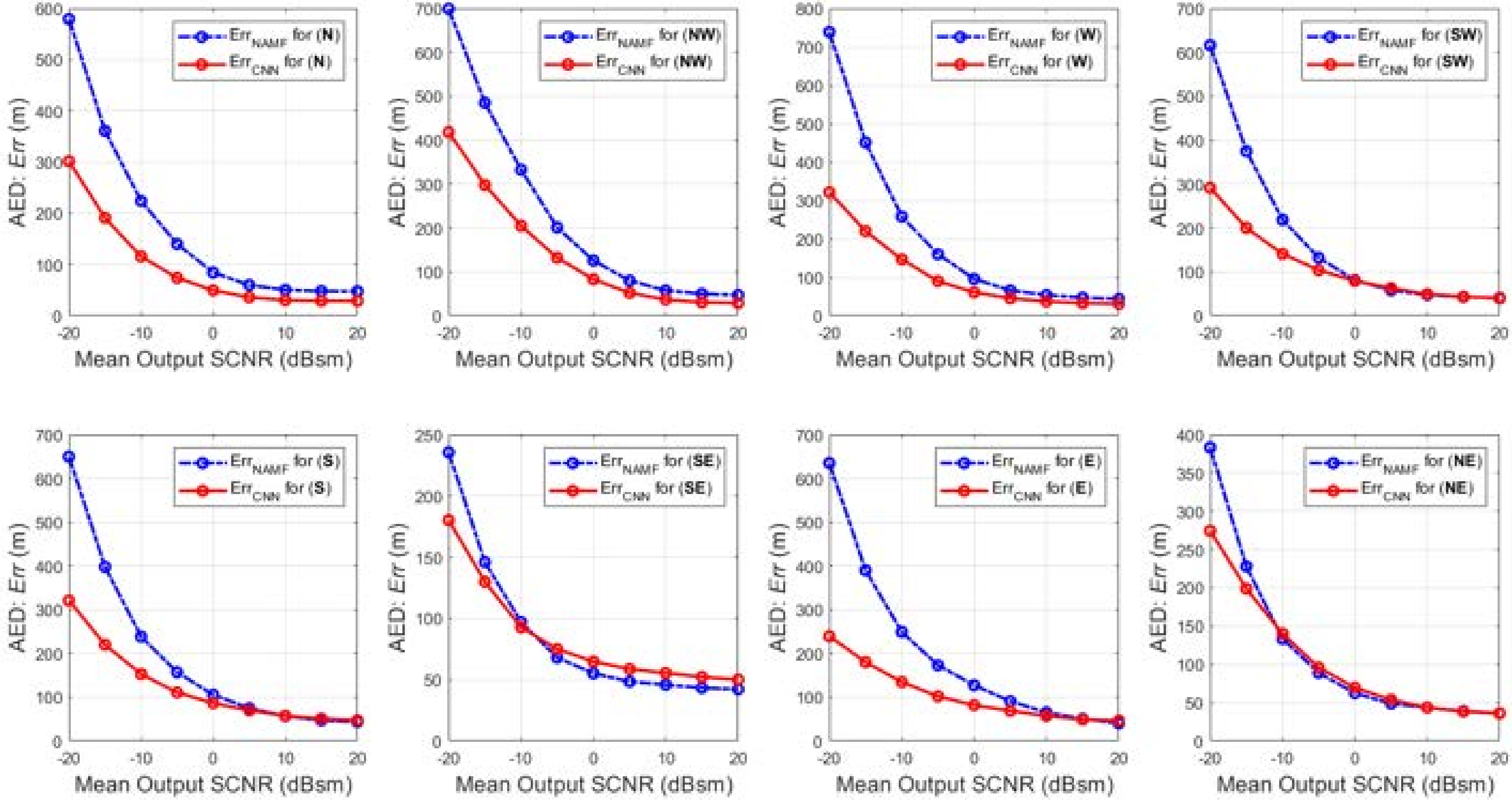
  Inference Time $= 2.25 \times 10^{-3}$ s [1] No. of Trainable Parameters $= 13{,}374$



[1] All network inference times were measured using an NVIDIA GeForce RTX 3090 GPU and averaged across all examples (heatmap tensors) in the respective dataset.

Mismatched Case Empirical Results

Evaluating Regression CNN Framework for **Variable SCNR (Baseline CNN)**



Mismatched Case Empirical Results

- Recall chordal distance ($\hat{\mathbf{d}}_{\mathbf{chordal}}$) between $(\mathbf{O})$ and $(\mathbf{D})$:
  - For <u>mean output SCNR</u> **above** the SCNR-independent breakdown threshold of the NAMF test statistic ($\boldsymbol{c = -4}$ $\mathbf{dB}$) [8], we **expect the following ordering** of the displaced platform location instances (in terms of localization accuracy):
  - Err_{CNN} for $(\mathbf{N}), (\mathbf{NW}), (\mathbf{W})$ will see the <u>greatest improvement</u> over Err_{NAMF}
  - Err_{CNN} for $(\mathbf{SW}), (\mathbf{NE})$ will see a <u>diminished improvement</u> over Err_{NAMF}
  - Err_{CNN} for $(\mathbf{S}), (\mathbf{E}), (\mathbf{SE})$ will see the <u>lowest improvement</u> over Err_{NAMF}

- For <u>mean output SCNR</u> $= 20$ dB, the gain afforded by the CNN is:

D	N	NW	W	SW	S	SE	E	NE
$\hat{\mathrm{d}}_{\mathrm{chordal}}$	0.31	0.31	0.34	0.45	0.51	0.55	0.54	0.47
Gain	1.65	1.63	1.41	1.02	0.91	0.84	0.86	1.01

Pairwise Agreement

Mismatched Case Empirical Results – Transfer Learning

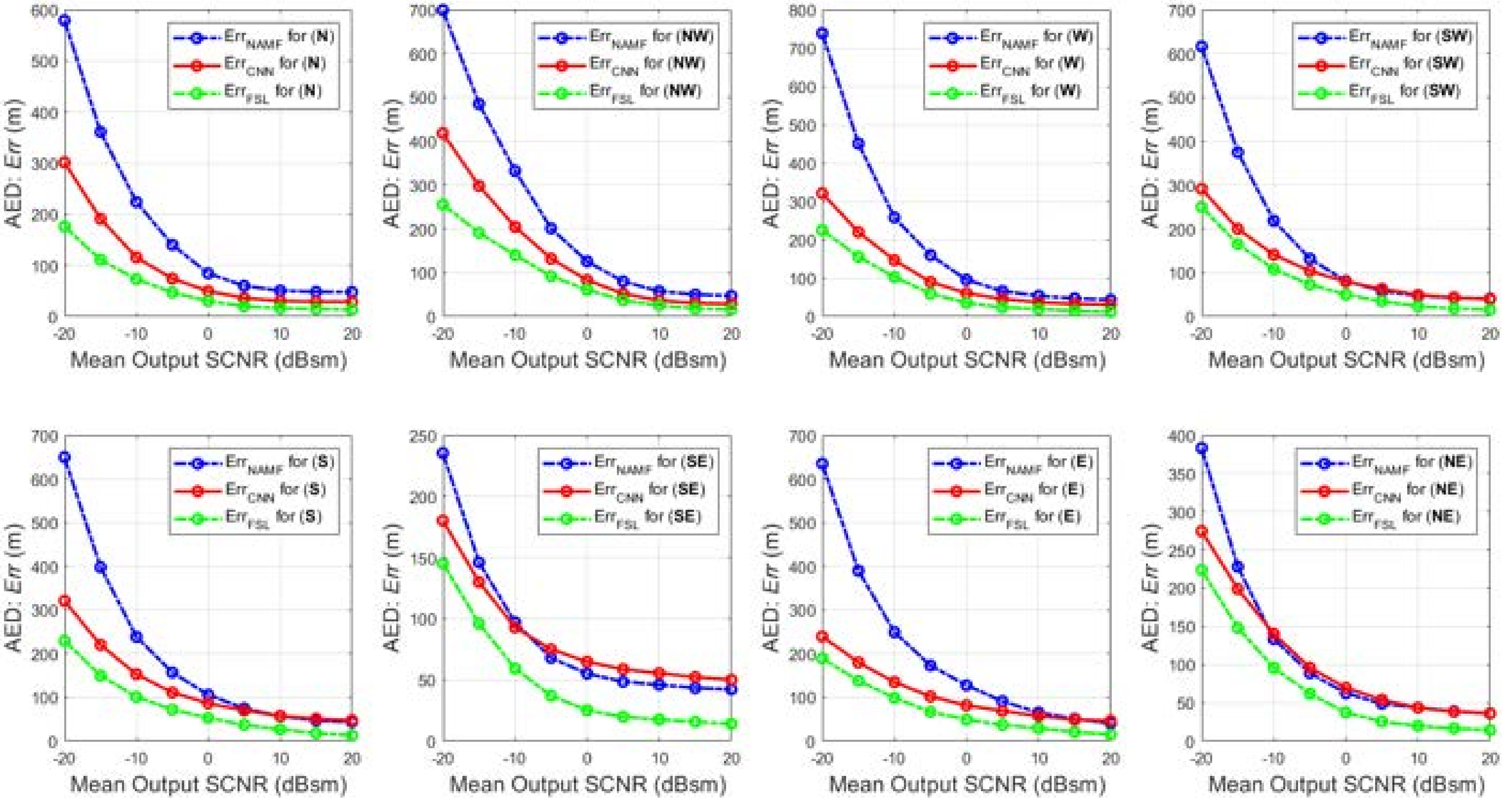
- We observe that the gains afforded by our regression CNN framework are diminished across the displaced platform location instances
- How can we improve this diminished gain? A: **Transfer Learning**.

Transfer Learning

- Applying model trained on original platform location instance $(\mathbf{O})$ to displaced platform location instances $(\mathbf{D}) \in \{\mathbf{N}, \mathbf{NW}, \mathbf{W}, \mathbf{SW}, \mathbf{S}, \mathbf{SE}, \mathbf{E}, \mathbf{NE}\}$:
  - Instead of generating 9×10^4 new heatmap tensors for each displaced platform location instance to re-train the network from scratch, what if we **fine-tuned** the pre-trained CNN using only 64 new examples for each displaced platform location instance?

Mismatched Case Empirical Results – Transfer Learning

Augmenting Regression CNN Framework with **Transfer Learning (Baseline CNN)**



Mismatched Case Empirical Results

- For mean output SCNR $= 20\ \mathrm{dB}$, the gain afforded by the CNN after transfer learning is (Gain [TL]):

D	N	NW	W	SW	S	SE	E	NE
$\hat{\mathrm{d}}_{\mathrm{chordal}}$	0.31	0.31	0.34	0.45	0.51	0.55	0.54	0.47
Gain	1.65	1.63	1.41	1.02	0.91	0.84	0.86	1.01
Gain (TL)	4.05	3.97	3.92	3.87	3.85	3.8	3.81	3.88

Outline

1) Introduction and Background
   - Data-Driven Target Localization
   - Problem Statement

2) Overview of Methods
   - Robust Mismatched Case Example Scenario
   - Subspace Perturbation Analysis

3) Empirical Results
   - Regression CNN Framework
   - Robust Mismatched Case Results

4) Conclusion

5) Addendum: Doppler Processing Example

Conclusion

- Benchmarking target localization performance of neural networks across mismatched scenarios has been an open problem
    - **In this work**: Benchmarked target localization performance of regression CNN framework by performing a subspace perturbation analysis
        - **Chordal distance metric** presents a **pairwise agreement** with the gains afforded by regression CNN framework over more traditional approach
    - To ameliorate the reduced target localization accuracies observed by our CNN (for large $\hat{\mathbf{d}}_{\mathbf{chordal}}$), we augmented our approach with transfer learning
        - 3-fold gain is now observed across all displaced platform location instances

Future Directions

- Complex-Valued Siamese CNN for target localization in development
- Benchmarking radar STAP database in development

Outline

1) Introduction and Background
   - Data-Driven Target Localization
   - Problem Statement

2) Overview of Methods
   - Robust Mismatched Case Example Scenario
   - Subspace Perturbation Analysis

3) Empirical Results
   - Regression CNN Framework
   - Robust Mismatched Case Results

4) Conclusion

5) Addendum: Doppler Processing Example

Addendum – Doppler Processing Example

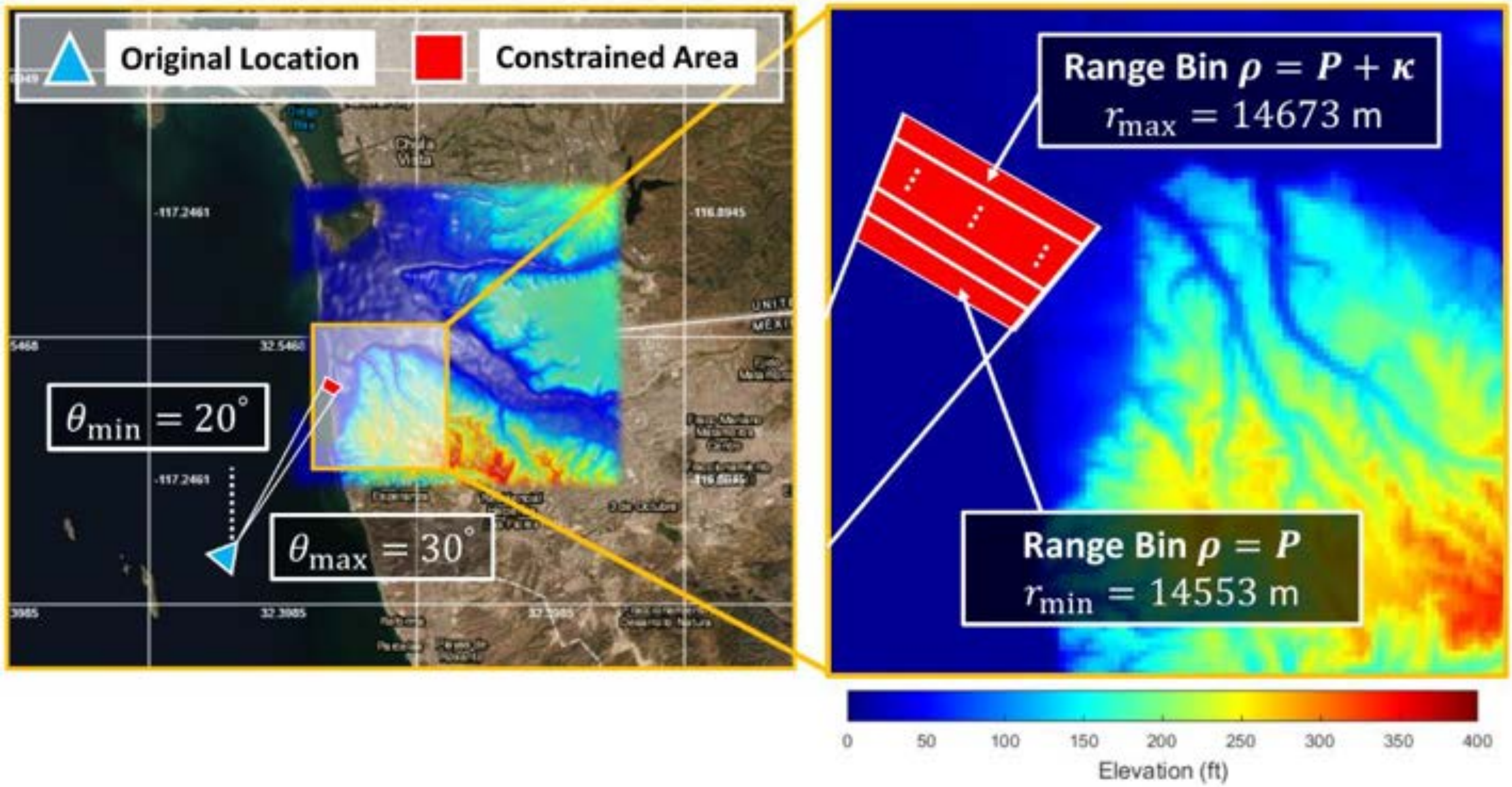
- Previous papers: **Deep learning** improves target localization performance
  - **More classical approach**: Estimating the peak location from a detection test statistic
  - Baseline CNN [7] achieves significant gains over more classical approach in matched settings. **Can we extend this to the Doppler case? (Position & Velocity Estimation)**
    - Need to revisit heatmap tensor generation procedure introduced in [7]
  - Consider Doppler example scenario in RFView® for further evaluation

<u>Doppler Example Scenario</u>:

- $D = 3$ dimensions exploited for processing: Range, $\boldsymbol{r}$, Azimuth, $\boldsymbol{\theta}$, Velocity, $\boldsymbol{v}$
- Each radar return is produced using Λ transmitted pulses
  - RFView® aggregates topography & radar parameters to simulate radar return

Addendum – Doppler Example Scenario

- Consider a single airborne radar platform within the following scene:



Addendum – Doppler Example Scenario

- Moving target randomly placed within red constrained area
  - RCS randomly chosen from uniform distribution w/ set mean (range = 10 dBsm)
  - Generate $\boldsymbol{K}$ **independent radar returns** $\left(\mathbf{Y}_{\boldsymbol{\rho}} \in \mathbb{R}^{\Lambda \times L \times K}\right)$ using RFView (Λ transmitted pulses, L-channel receiver) and <u>transform into heatmap tensor</u>
  - **Repeat** $\boldsymbol{N = 9 \times 10^4}$ **times** to obtain $\boldsymbol{N}$ **heatmap tensors** (examples in our dataset)
    - Heatmap tensor = 3D representation of constrained area

- Approximate constrained area specifications: ($\kappa = 5$ range bins)
  - **Default grid step size**: $(\Delta r, \Delta\theta, \Delta v) = (30\text{ m}, 0.4°, 0.5\text{ m/s})$

Original Location – Parameters	Values
Platform latitude, longitude	32.4275°, −117.1993°
Constrained area range $(r_{\min}, r_{\max})$	(14553 m, 14673 m)
Constrained area azimuth $(\theta_{\min}, \theta_{\max})$	(20°, 30°)
Constrained area velocity $(v_{\min}, v_{\max})$	(175 m/s, 190 m/s)

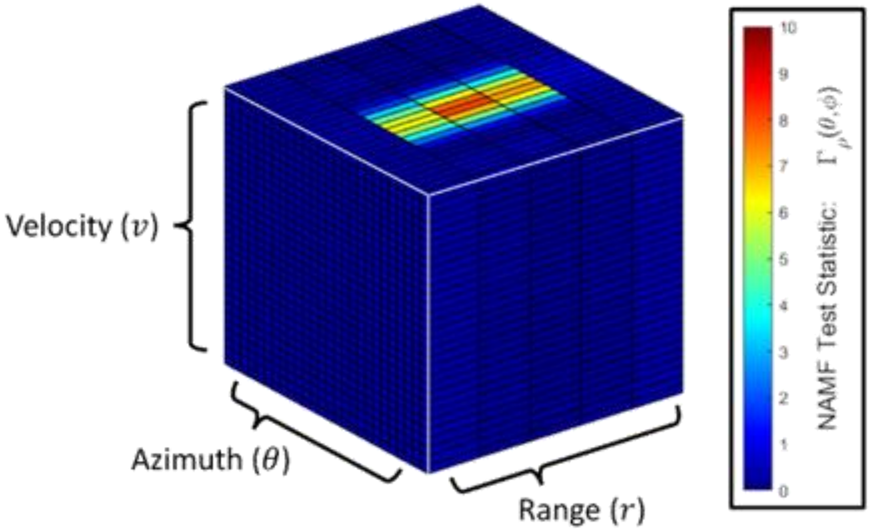
Addendum – Doppler Example Scenario

Heatmap Tensor Generation:

- For each example in our dataset ($N = 9 \times 10^4$ total):
  - For each range bin ($r = 98, \ldots, 102$) from constrained area:
    - Sweep across azimuth and velocity with step size $(\Delta\theta, \Delta v)$
      - Record NAMF Test Statistic, $\Gamma_\rho(\theta, v)$, at each location

Result:

- Heatmap Tensor of size $5 \times 26 \times 31$



Addendum – Doppler CNN Framework

<u>Doppler CNN</u> for target position and velocity estimation $(\Lambda = 4, L = 16, K = 100)$

- **<u>Dataset</u> — $\boldsymbol{N = 9 \times 10^4}$ heatmap tensors**, produced using NAMF test statistic
  - Randomly partition N examples such that we have:
    $\boxed{N_{train} = 0.9N \text{ training examples}}$ $\boxed{N_{eval} = 0.1N \text{ validation examples}}$
- Fit Regression CNN with training dataset
  - Learn target locations $(r^*, \theta^*, v^*) \rightarrow (x^*, y^*, v^*)$ from the $0.9N$ heatmap tensors
- Evaluate Regression CNN on validation dataset
  - Estimate target location $(\hat{r}, \hat{\theta}, \hat{v}) \rightarrow (\hat{x}, \hat{y}, \hat{v})$ for each of the $0.1N$ heatmap tensors

Addendum – Doppler CNN Architecture

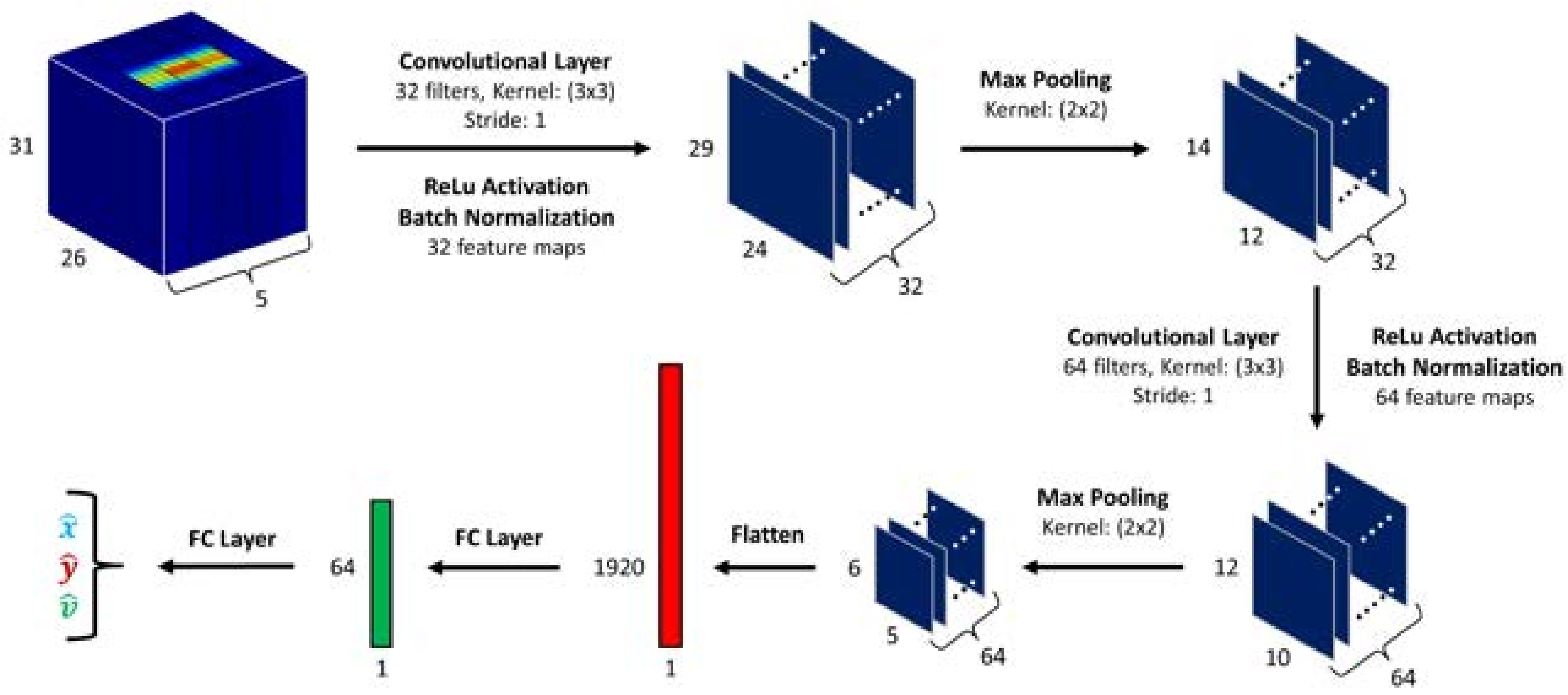
- Used for evaluations with <u>default grid step size</u>: $(\Delta r, \Delta\theta, \Delta v) = (30\ \mathrm{m}, 0.4°, 0.5\ \mathrm{m/s})$

  Tensor dimensions $= \mathbf{5 \times 26 \times 31}$

  Output $= (\hat{x}, \hat{y}, \hat{v})$ coordinates of target

  Inference Time $= 4.35 \times 10^{-3}$ s [1]

  No. of Trainable Parameters $= 143{,}299$



Addendum – Average Euclidean Distance Metric

- Comparing Doppler CNN with traditional methods (NAMF):
  - **Metric**: Euclidean distance (averaged across validation dataset)

- <u>Doppler CNN</u>:
- Estimated target attributes vs. ground truth target attributes

$$Err_{\text{CNN}} = \frac{\sum_{j=1}^{N_{eval}} \|(x_j^*, y_j^*) - (\hat{x}_j, \hat{y}_j)\|_2}{N_{eval}}$$

$$(Err_{\text{CNN}})_v = \frac{\sum_{j=1}^{N_{eval}} \|v_j^* - \hat{v}_j\|_2}{N_{eval}}$$

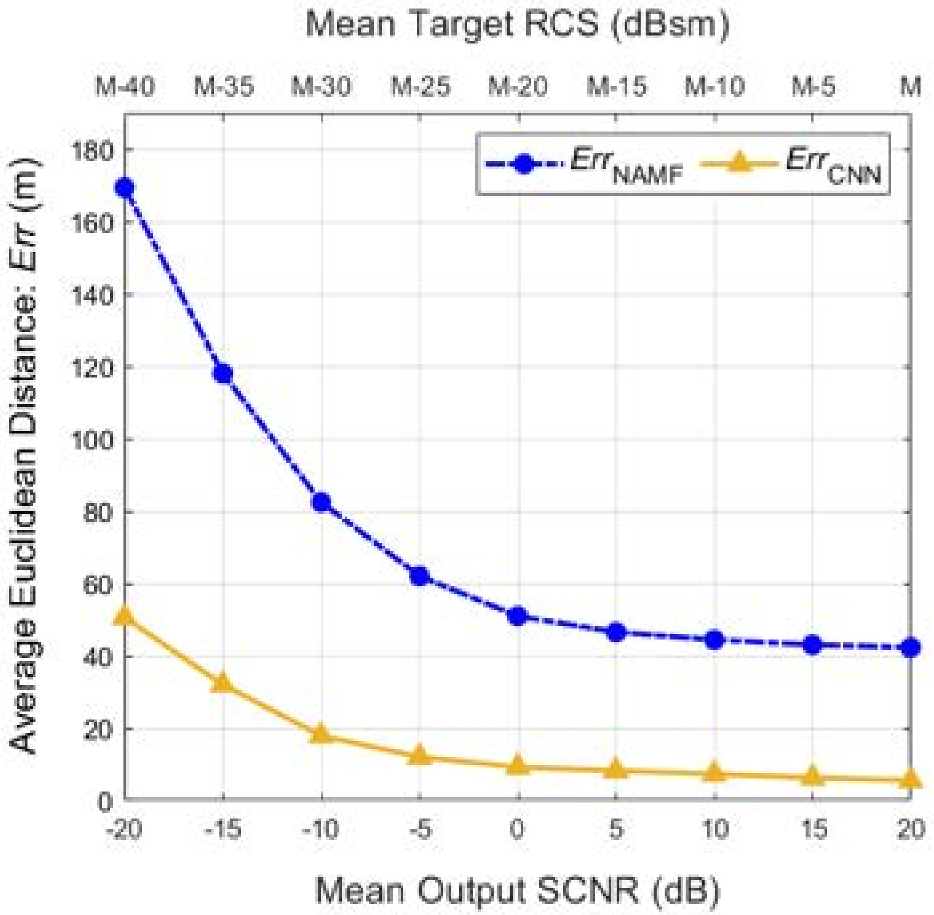
- <u>Traditional Method</u>:
- Midpoint of cell with peak test statistic in each heatmap tensor vs. ground truth target attributes

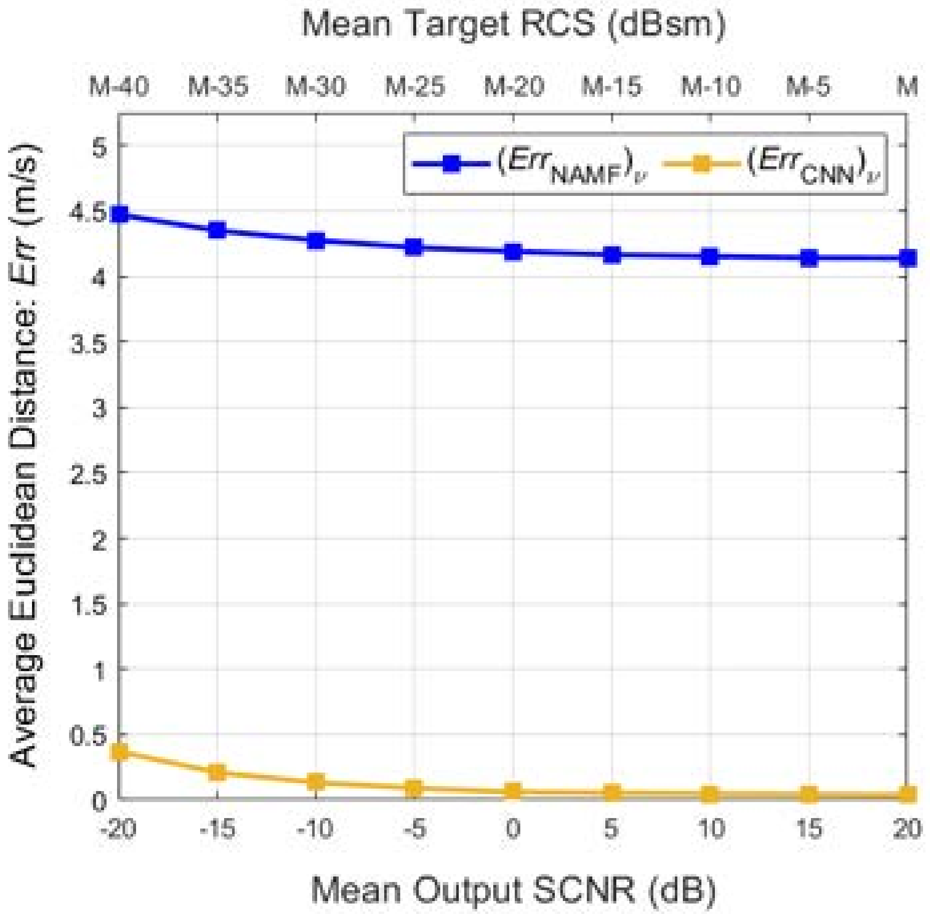
$$Err_{\text{NAMF}} = \frac{\sum_{j=1}^{N_{eval}} \|(x_j^*, y_j^*) - (\bar{x}_j, \bar{y}_j)\|_2}{N_{eval}}$$

$$(Err_{\text{NAMF}})_v = \frac{\sum_{j=1}^{N_{eval}} \|v_j^* - \bar{v}_j\|_2}{N_{eval}}$$

Addendum – Doppler Example Empirical Results

<u>Doppler CNN</u> $[Err_{CNN}$ & $(Err_{CNN})_v]$ vs. <u>NAMF Test Statistic</u> $[Err_{NAMF}$ & $(Err_{NAMF})_v]$





Summary of Accomplishments

- Benchmarked CNN w.r.t Normalized Adaptive Matched Filter (gold standard of model-based approaches)
  - **No mismatches in radar scenario**: CNN yields 10-fold improvement in localization accuracy over NAMF, improving localization accuracy (Euclidean distance) from 45 m to 4.5 m (range resolution = 30 m)
- Showed that CNN can be made **robust to mismatches** between the train and test data in radar scenarios (e.g., sensitivity of target localization by changing platform location and clutter response)
  - **Proof-of-concept**: Displace the radar scenario by 1 km in each cardinal and intermediate direction, which results in a markedly different clutter response (highly variable terrain)
    - Showed that we can use chordal distance as a task affinity measure (between original radar scenario and displaced scenarios) to help benchmark CNN gain over NAMF (yields pairwise agreement)
    - Developed a Transfer Learning approach to robustify CNN to mismatches in the radar scenario, with only small sample overhead for training (only 64 new samples instead of 100,000 training samples)
      - Using transfer learning, the CNN recovers up to a 4-fold gain over NAMF (10 m vs. 40 m)
- Outlined and began construction of our **benchmarking radar STAP dataset** (stored in tree data structure)
  - **10,000 total scenarios** in dataset, each consisting of **10,000 clutter + noise realizations** (clutter response)
  - LBG algorithm determines **100 representative scenarios** from 10,000 total scenarios (using Energy distance)
    - For each representative scenario, we provide **100,000 radar data cubes** (target + clutter + noise data), which the user can use to train their own neural network. We also provide our pre-trained CNN.
    - To use the dataset, the user stores in memory the CNN that has been trained on the representative scenario with the minimal task affinity measure (Energy distance) w.r.t their own scenario.
    - User can fine-tune CNN via. transfer learning by generating a few new samples from their own scenario

References

[1] J. H. Michels, M. Rangaswamy, and B. Himed, “Performance of parametric and covariance based STAP tests in compound-gaussian clutter,” Digital Signal Processing, vol. 12, no. 2, pp. 307–328, 2002.

[2] W. Melvin, M. Wicks, and R. Brown, “Assessment of multichannel airborne radar measurements for analysis and design of space-time processing architectures and algorithms,” in Proceedings of the 1996 IEEE National Radar Conference, 1996, pp. 130–135.

[3] A. Krizhevsky, G. Hinton et al., “Learning multiple layers of features from tiny images,” Citeseer, 2009

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft COCO: common objects in context,” in European conference on computer vision. Springer, 2014, pp. 740–755.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[6] S. Gogineni, J. R. Guerci, H. K. Nguyen, J. S. Bergin, D. R. Kirk, B. C. Watson, and M. Rangaswamy, “High fidelity rf clutter modeling and simulation,” IEEE Aerospace and Electronic Systems Magazine, vol. 37, no. 11, pp. 24-43, 2022.

[7] S. Venkatasubramanian, C. Wongkamthong, M. Soltani, B. Kang, S. Gogineni, A. Pezeshki, M. Rangaswamy, and V. Tarokh, “Toward Data-Driven STAP Radar,” 2022 IEEE Radar Conference (RadarConf22), pp. 1-5.

[8] S. Venkatasubramanian, S. Gogineni, B. Kang, A. Pezeshki, M. Rangaswamy, and V. Tarokh, “Data-Driven Target Localization Using Adaptive Radar Processing and Convolutional Neural Networks,” arXiv, 2022.