# C-SURE: Shrinkage Estimator and Prototype Classifier for Complex-Valued Deep Learning

Rudrasis Chakraborty[1*] Yifei Xing[1*] Minxuan Duan[2] Stella X. Yu[1]

[1] UC Berkeley / ICSI [2] Peking University

## Abstract

*The James-Stein (JS) shrinkage estimator is a biased estimator that captures the mean of Gaussian random vectors. While it has a desirable statistical property of dominance over the maximum likelihood estimator (MLE) in terms of mean squared error (MSE), not much progress has been made on extending the estimator onto manifold-valued data.*

*We propose C-SURE, a novel Stein's unbiased risk estimate (SURE) of the JS estimator on the manifold of complex-valued data with a theoretically proven optimum over MLE. Adapting the architecture of the complex-valued SurReal classifier, we further incorporate C-SURE into a prototype convolutional neural network (CNN) classifier.*

*We compare C-SURE with SurReal and a real-valued baseline on complex-valued MSTAR and RadioML datasets. C-SURE is more accurate and robust than SurReal, and the shrinkage estimator is always better than MLE for the same prototype classifier. Like SurReal, C-SURE is much smaller, outperforming the real-valued baseline on MSTAR (RadioML) with less than $1\%$ ($3\%$) of the baseline size.*

## 1. Introduction

Deep learning has been widely adopted in computer vision, often assuming data that follow vector-space properties. However, there are plenty of natural non-Euclidean manifold-valued data. Complex-valued data such as medical images, radio signals, and nuclear covariances can all be modeled as Riemannian manifolds. Even for real-valued signals, their manifold-valued representations could be more informative of underlying signals, such as Fourier transforms and spectrum-based techniques [14, 20, 30].

The earliest relevant research for manifold-valued deep learning can be traced back to [26], which regards images as manifolds and applies differential geometry for structural analysis. More recent works have explored preserving the inherent geometry of graphs [15, 24], or achieving group equivariance and invariance [3, 8]. However, these works do not offer an extension to naturally manifold-valued data.

The intrinsic geometric structures of non-Euclidean manifold data are addressed in [4, 11, 7]. In particular, with convolution on the manifold defined as weighted Fréchet mean (wFM) filtering [4, 5], significant performance gain can be achieved on manifold-valued data along with drastic reduction in the model parameter count.

We focus on the manifold of complex-valued data such as synthetic aperture radar (SAR) images, magnetic resonance (MR) images, and radio frequency (RF) signals. Per the polar form of complex numbers, the complex plane can be treated as a product space of scaling and planar rotations. This representation allows [6] to develop an efficient CNN classifier based on wFM filtering on the specific manifold.

The wFM is only one way to compute the mean of samples on a manifold. Recently, [29] suggests an alternative by defining a James-Stein (JS) estimator that outperforms the Fréchet mean in terms of MSE over the field of semi-positive definite matrices (SPD). Here we extend the idea to data lying on the field of complex numbers and prove the dominance of the JS estimator over the Fréchet mean.

The JS estimator arises from simultaneously estimating the mean of a multivariate homoscedastic normal distribution. Let $X$ denote a random vector whose $p$ components are independent and normally distributed with mean $\theta_i$ and variance $\sigma^2$, $i = 1, \ldots, p$. When $p > 2$, it can be shown that the JS estimator $\theta^{\mathrm{JS}}$ dominates the maximum likelihood estimator (MLE) $\theta^{\mathrm{MLE}}$ [17]:

$$\theta^{\mathrm{MLE}} = X \tag{1}$$

$$\theta^{\mathrm{JS}} = \left(1 - \frac{(p-2)\sigma^2}{\|X\|^2}\right) X. \tag{2}$$

When $(p-2)\sigma^2 < \|X\|^2$, the JS estimator simply takes the natural estimator $X$ (i.e., $\theta^{\mathrm{MLE}}$) and shrinks it towards the origin 0. The JS estimator can be viewed as an empirical Bayes method, where $\theta$ itself is a random variable with prior distribution and needs to be estimated [10].

The JS estimator can be viewed as a special case of a hierarchical Bayesian model [2], where the unknown mean $\theta_i$

*Authors of equal contributions.

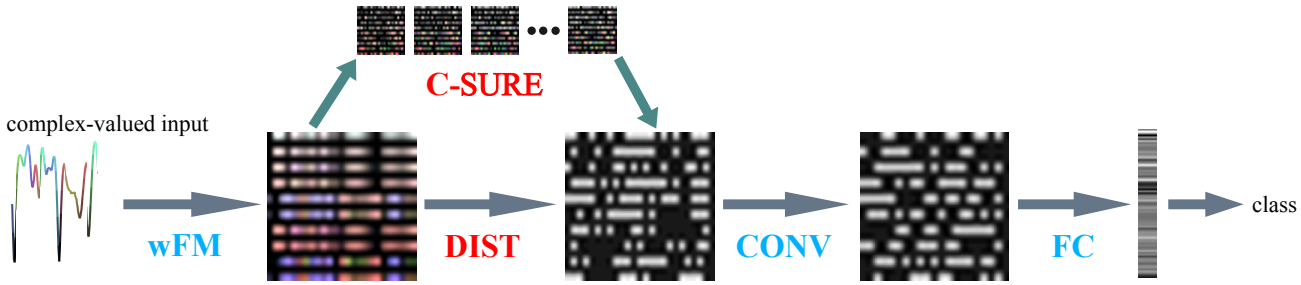


Figure 1: Workflow of our model-based C-SURE CNN classifier. Our model is based on the SurReal complex-valued CNN [6], which consists of wFM convolution layers (wFM), distance transformation layers (DIST), standard convolution layers (CONV), and finally fully connected layers (FC) for softmax classification. We incorporate C-SURE into the distance transformation layer: During training, the statistical mean of the wFM features per class is estimated using C-SURE, and the minimum distances between the wFM features and the set of class means become the real-valued output; during testing, only the distances between the wFM features and the saved class means need to be calculated. The real-valued distances go through standard CONV and FC layers for the final classification into semantic categories.

follows a normal prior distribution with mean $\mu$ and variance $\tau^2$. Given data variance $\sigma^2$, for any prior setting $(\mu, \tau)$, the maximum a posterior (MAP) estimation of $\theta$ is a weighted sum of the data and the prior:

$$\hat{\theta}(\mu, \tau; \sigma) = \frac{\tau^2}{\tau^2 + \sigma^2} X + \frac{\sigma^2}{\tau^2 + \sigma^2} \mu. \tag{3}$$

The best JS estimator can be solved by choosing $(\mu, \tau)$ that minimizes the Stein's unbiased risk estimate (SURE) [27]:

$$\hat{\mu}^{\text{SURE}}, \hat{\tau}^{\text{SURE}} = \arg\min_{\mu,\tau} \text{SURE}(\mu, \tau) \tag{4}$$

$$\text{SURE}(\mu, \tau) = -p\sigma^2 + \|\hat{\theta} - X\|^2 + 2\sigma^2 \sum_{i=1}^{p} \frac{\partial \hat{\theta}}{\partial X_i} \tag{5}$$

The importance of SURE is that it does not depend on the unknown $\theta$, and yet it is an unbiased estimate of the mean-squared error (MSE) between $\hat{\theta}$ and $\theta$. Minimizing SURE can thus act as a surrogate for minimizing the MSE and the optimal estimation setting can be obtained without $\theta$.

It has been a challenge to generalize the JS shrinkage estimator to non-Euclidean spaces. The idea of shrinking is natural only on certain manifolds; there is no formula for the shrinkage estimator on manifolds in general. For instance, a shrinkage estimator for covariance matrices is designed in [19, 9] and then generalized to the Riemannian manifold of symmetric positive definite (SPD) matrices [29].

Our goal is not only to extend the shrinkage estimator to the manifold of complex numbers, but also to use it in learning the classification of signals or images, an under-explored application of the JS estimator among various machine learning settings [21, 28, 12].

We propose C-SURE, a novel SURE of the JS estimator on the manifold of complex-valued data with a theoretically proven optimum over MLE. We incorporate it into learning a convolutional neural network (CNN) classifier (Fig.1). Instead of learning a purely discriminative classifier, we learn a nearest prototype-based classifier based on the feature distribution mean of each class.

Compared to popular real-valued CNN classifiers and the SurReal CNN classifier [6] based on the Fréchet mean on the complex-valued manifold, our C-SURE prototype classifiers achieve better performance with faster convergence on two complex-valued datasets: MSTAR and RadioML. Our model is also much smaller, outperforming the real-valued CNN on MSTAR with less than $1\%$ of the model size.

## 2. Shrinkage Estimator of Complex Numbers

We view the field of complex numbers $\mathbf{C}$ as a product group of two smooth Riemannian manifolds, or more specifically Lie groups. On each of these two manifolds, we define Gaussian distributions and construct the JS shrinkage estimator of the Fréchet Mean (FM) on the manifold. We show that our shrinkage estimator on $\mathbf{C}$ yields a uniformly smaller risk than the MLE estimator, i.e., FM.

**Manifold View of Complex Plane C.** We represent $\mathbf{C}$ as a product space of two Riemannian manifolds [6]. Utilizing the polar form of a complex number $\mathbf{c} = re^{i\theta}$, we have:

$$\mathbf{C} = \{re^{i\theta}\} \simeq \mathbf{R}^+ \times \mathbf{S^1} \simeq \mathbf{P_1} \times \mathsf{SO(2)} \tag{6}$$

where the manifold of $1 \times 1$ semi-positive definite matrices $\mathbf{P_1}$ is topologically the space of non-negative numbers $\mathbf{R}^+$, and the manifold of $2 \times 2$ rotation matrices $\mathsf{SO(2)}$ is topologically a circle $\mathbf{S^1}$. With this decomposition, designing the JS shrinkage estimator on $\mathbf{C}$ is reduced to designing the estimator on $\mathbf{P_1}$ and $\mathsf{SO(2)}$ separately.

The shrinkage estimator on $\mathbf{P_1}$ has been dealt with in [29], where they propose a novel shrinkage estimator on the

FM of SPD matrices, with proven dominance over the MLE of the FM in terms of the MSE risk.

In order to develop the shrinkage estimator on SO(2), we choose a Riemannian metric and define our FM. Using the Lie algebra of SO(2), we can apply the procedure in [29] and derive our shrinkage estimator on SO(2).

**Fréchet Mean on a Riemannian manifold.** Let $\mathcal{M}$ be a topological manifold equipped with a Riemannian metric, and let $d : \mathcal{M} \times \mathcal{M} \to \mathbf{R}$ denote the associated distance. Given a collection of $n$ points $\{X_i, i = 1, \ldots, n\}$ on the manifold, their Fréchet Mean [13] is defined as:

$$\bar{X} = \arg\min_{X \in \mathcal{M}} \sum_{i=1}^{n} d^2(X, X_i). \tag{7}$$

In general, $\bar{X}$ may not be unique, but can be made unique under certain constraints.

**Log-Euclidean Metric on SO(2).** We endow $\mathsf{SO}(2)$ with the Log-Euclidean (LE) metric, for it is computationally efficient with closed-form solutions. We extend the Log-Euclidean metric in [1] and define the induced geodesic distance $d_{LE} : \mathsf{SO(2)} \times \mathsf{SO(2)} \to \mathbf{R}$ as:

$$d_{LE}(X_1, X_2) = \| \log(X_1) - \log(X_2) \|_F \tag{8}$$

$$X_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \in \mathsf{SO(2)} \tag{9}$$

$$\log(X_i) = (\theta_i + 2\pi k) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \; k = \pm 1, \pm 2, \ldots. \tag{10}$$

Here $\log$ denotes the matrix logarithm. The logarithm of a rotation matrix is not unique; we fix $k = 0$ here and obtain an isomorphism between $\mathfrak{so}(2)$ and the interval $(-\pi, \pi]$. This particular logarithm is called *principal*.

We establish a mapping $\widetilde{X}$ from point $X$ on the manifold $\mathcal{M}$=SO(2) to a number in the real domain $\mathbf{R}$ that indicates the size of rotation directly. Since the Lie algebra $\mathfrak{so}(2)$ of SO(2) is the space of $2 \times 2$ skew-symmetric matrices, we define $\widetilde{X}$ through the mapping $\Phi : \mathfrak{so}(2) \to \mathbf{R}$.

$$\widetilde{X} = \Phi(\log(X)) = \sqrt{2}\,\theta \tag{11}$$

$$\log(X) = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \in \mathfrak{so}(2). \tag{12}$$

The distance between points $X_1$ and $X_2$ on SO(2) is simply:

$$d_{LE}(X_1, X_2) = \min\{|\widetilde{X_1} - \widetilde{X_2}|, 2\sqrt{2}\pi - |\widetilde{X_1} - \widetilde{X_2}|\}, \tag{13}$$

ensuring the uniqueness of the FM on SO(2).

**Gaussian Distributions on SO(2).** Gaussian distributions on the manifold of positive definite matrices become Log-Normal distributions [25]. We follow the same procedure and extend it to other matrix Lie groups such as SO(2).

**Definition 1.** *We say $X$ follows a Log-Normal distribution with mean $M$ and covariance matrix $\Sigma \in \mathbf{R}_{m \times m}$, or $X \sim \mathrm{LN}(M, \Sigma)$ if*

$$\widetilde{X} \sim N(\widetilde{M}, \Sigma) \tag{14}$$

We can further define the mixture of Gaussians in order to capture multi-modal distributions in real-world data.

**Definition 2.** *We say that $X$ follows a mixture of $K$ Log-Normal distributions each with mean $M_k \in G$ and covariance matrix $\Sigma_k \in \mathbf{R}^{m \times m}$, or $X \sim \mathrm{MLN}(w, \boldsymbol{M}, \boldsymbol{\Sigma})$ if*

$$\widetilde{X} \sim \sum_{k=1}^{K} w_k N(\widetilde{M_k}, \Sigma_k) \tag{15}$$

$$\sum_{\mathrm{k}=1}^{\mathrm{K}} w_{\mathrm{k}} = 1, \quad 0 \le w_{\mathrm{k}} \le 1, \forall k. \tag{16}$$

We refer to each $N(\widetilde{M_k}, \Sigma_k)$ as the $k$-th component density in the Gaussian mixture model.

Calculating these distributions requires the composition of logarithmic and exponential maps. On SO(2), we have

$$\exp(\log X + \log Y) = XY \tag{17}$$

since SO(2) has a trivial Lie algebra with zero Lie brackets.

**C-SURE Shrinkage Estimator on $\mathsf{SO(2)}^p$.** Let $X$ denote a $p$-dimensional complex-valued random variable; for the $i$-th dimension, the value $X_i$ is modeled as a point on the manifold SO(2). We are going to estimate the $p$-dimensional mean vector $M$ from a collection of these manifold-valued observations, using the JS estimator derived from a hierarchical Bayesian approach.

For $X \in \mathsf{SO(2)}^p$, we assume that $X_i$ is independently distributed according to the Log-Normal with individual mean $M_i$ and equal variance $vI$, and the means $\{M_i, i = 1, \ldots, p\}$ are independently and identically distributed according to a Log-Normal mixture:

$$X_i | M_i \overset{ind}{\sim} \mathrm{LN}(M_i, vI), i = 1, \ldots, p \tag{18}$$

$$M_i \overset{i.i.d}{\sim} \mathrm{MLN}(w, \mu, D). \tag{19}$$

We assume that $v$ is known and $w$ is fixed, whereas $\mu$ and $D = \mathrm{Diag}(\lambda_1 I, \cdots, \lambda_K I)$ are unknown and can be optimized by minimizing the SURE risk.

Since $w$ is fixed, we can first calculate the JS estimator for each of the $k$ component densities independently, and then combine them with their respective weights in $w$ to obtain the JS estimator of the Log-Normal mixture.

Specifically, using the derivations for the Gaussian distribution on SO(2), we extend the MAP estimate in Eqn(3) to

SO(2) for the $k$-th component density $M_{i,k} \sim \mathrm{LN}(\mu_k, D_k)$:

$$\widehat{M}_{i,k}^{\mu,D}(w) = \exp\left(\frac{\lambda_k}{\lambda_k+v}\log\overline{X}_i^{\mathrm{LE}}(w)+\frac{v}{\lambda_k+v}\log\mu_k\right) \tag{20}$$

where $\overline{X}_i^{\mathrm{LE}}(w)$ denotes the mean of $X_i$ over a total of $N$ sample observations, according to the Log-Euclidean metric and given the mixture weights $w$.

We can then extend the MSE to our manifold by defining the empirical loss function $l$ as:

$$l\left(\widehat{M}_k^{\mu,D}, M_k\right) = \sum_{i=1}^{p} d_{\mathrm{LE}}^2\left(\widehat{M}_{i,k}^{\mu,D}, M_{i,k}\right) \tag{21}$$

and the corresponding risk $R$ as $\mathbb{E}[l]$:

$$\begin{aligned} R\left(\widehat{M}_k^{\mu,D}, M_k\right) &= \mathbb{E}\left[l\left(\widehat{M}_k^{\mu,D}, M_k\right)\right] \\ &= \sum_{i=1}^{p}\frac{v}{(\lambda_k+v)^2}\left(v\left\|\log\mu_k-\log M_{i,k}\right\|^2+\frac{p\lambda_k^2}{N}\right). \end{aligned} \tag{22}$$

The SURE estimate in Eqn(5), $\mathrm{SURE}(\mu_k, \lambda_k)$, becomes:

$$\sum_{i=1}^{p}\frac{v}{(\lambda_k+v)^2}\left(v\left\|\log\overline{X}_i^{\mathrm{LE}}-\log\mu_k\right\|^2+\frac{p\left(\lambda_k^2-v^2\right)}{N}\right). \tag{23}$$

Our SURE estimate of the $k$-th component mean and variance on the manifold of complex values is thus:

$$\hat{\mu}_k^{\mathrm{SURE}}, \hat{\lambda}_k^{\mathrm{SURE}} = \arg\min_{\mu_k,\lambda_k}\mathrm{SURE}(\mu_k,\lambda_k) \tag{24}$$

$$\begin{aligned} &= \arg\min_{\mu_k,\lambda_k}\sum_{i=1}^{p}\frac{v}{(\lambda_k+v)^2} \\ &\left(v\left\|\log\overline{X}_i^{\mathrm{LE}}-\log\mu_k\right\|^2+\frac{p\left(\lambda_k^2-v^2\right)}{N}\right). \end{aligned} \tag{25}$$

We propose our C-SURE shrinkage estimator for $M_i$ as a weighted sum of its components:

$$\widehat{M}_i^{\mathrm{SURE}}(w) = \sum_{k=1}^{K}\exp\left(w_k\left(\frac{\hat{\lambda}_k^{\mathrm{SURE}}}{\hat{\lambda}_k^{\mathrm{SURE}}+v}\log\overline{X}_i^{\mathrm{LE}}+\frac{v}{\hat{\lambda}_k^{\mathrm{SURE}}+v}\log\hat{\mu}_k^{\mathrm{SURE}}\right)\right). \tag{26}$$

**Optimality of C-SURE Shrinkage over MLE.** We show that $\left(\hat{\mu}_k^{\mathrm{SURE}}, \hat{\lambda}_k^{\mathrm{SURE}}\right)$ minimizes the actual risk $R\left(\widehat{M}_k^{\mu,D}, M_k\right)$. We follow the approach in [29]: For each component density, we have $M_i \sim \mathrm{LN}(\mu, D)$ where $D = \lambda I$, $\mathrm{SURE}(\mu,\lambda)$ is a good approximation of $l\left(\widehat{M}^{\mu,D}, M\right)$.

**Theorem 1.** *Assume that*

$$\begin{aligned} &(A)\ v^2 < \infty \\ &(B)\ \limsup_{p\to\infty}\tfrac{1}{p}\textstyle\sum_i\left\|\log M_i\right\|^2 < \infty \\ &(C)\ \limsup_{p\to\infty}\tfrac{1}{p}\textstyle\sum_i\left\|\log M_i\right\|^{2+\delta} < \infty \textit{ for some } \delta > 0 \end{aligned}.$$

*Then the following holds in probability as $p \to \infty$:*

$$\sup_{\substack{\lambda>0 \\ \|\log\mu\|<\max_i\left\|\log\overline{X}_i^{LE}\right\|}}\left|\mathit{SURE}(\mu,\lambda)-l\left(\widehat{M}^{\mu,D}, M\right)\right| \to 0. \tag{27}$$

We can now show that for each component density, our proposed shrinkage estimator is asymptotically optimal, compared with MLE of the FM of Log-Normal distribution on SO(2) in terms of risk.

**Theorem 2.** *If (A), (B), (C) in Theorem 1 hold, then,*

$$\lim_{p\to\infty}\left[R\left(\widehat{M}^{\mathit{SURE}}, M\right)-R\left(\widehat{M}^{\mu,D}, M\right)\right] \le 0 \tag{28}$$

A proof of the theorems above for SPD can be found in [29]. Since the key is the trivial Lie algebra, which holds for both SPD and SO(2), we omit the similar proof.

**Weight Update.** After we obtain the class-wise means $\widehat{M}^{\mathrm{SURE}}(w)$ using Eqn(26) for a fixed $w$, we update $w$ by some learning method. While there are a plethora of statistical algorithms readily available to update $w$, e.g., Bayesian methods, EM algorithm etc., we find that gradient descent is more stable and produces more optimized values.

## 3. Prototype Classifier with C-SURE

We incorporate our C-SURE shrinkage estimator into a nearest prototype CNN classifier. We model each class as a mixture of Gaussians and learn their prototypes using our C-SURE estimator. Instead of assigning an instance to the nearest prototype, we use their minimal distance as a feature for discriminative classification.

**C-SURE Classifier Architecture.** Specifically, we build our classifier based on the SurReal complex-valued CNN [6], which consists of wFM convolution layers (wFM), distance transformation layers (DIST), and standard convolution layers (CONV), and finally fully connected layers (FC) for softmax classification (Fig.1).

We incorporate C-SURE into the distance transformation layer: During training, the statistical mean of the wFM features per class is estimated using C-SURE, and the minimum distances between the wFM features and the set of class means become the real-valued output; during testing, only the distances between the wFM features and

**Algorithm 1:** C-SURE Prototype Feature Layer.

**Input:** data $X_{\text{all}}$, variances $\{v\}$
**Output:** class means $\{M_i\}$, distance features $\{O_i\}$, $i$ out of $p$ refers to the $i$-th dimension of the data.

**for** *each class* **do**
  Gather instances of this class in $X$
  Calculate a running estimate of $\overline{X}^{\text{LE}}$
  **if** *training* **then**
    **for** *each mixture component* $k$ **do**
      $$\left(\hat{\mu}_k^{\text{SURE}}, \hat{\lambda}_k^{\text{SURE}}\right) = \arg\min_{\mu_k, \lambda_k} \sum_{i=1}^{p} \frac{v}{(\lambda_k + v)^2} \left( v \left\| \log \overline{X}_i^{\text{LE}} - \log \mu_k \right\|^2 + \frac{p(\lambda_k^2 - v^2)}{N} \right)$$
    **end**
  **end**
  $$\widehat{M}_i^{\text{SURE}}(w) = \sum_{k=1}^{K} \exp\left( w_k \left( \frac{\hat{\lambda}_k^{\text{SURE}}}{\hat{\lambda}_k^{\text{SURE}} + v} \log \overline{X}_i^{\text{LE}} + \frac{v}{\hat{\lambda}_k^{\text{SURE}} + v} \log \hat{\mu}_k^{\text{SURE}} \right) \right)$$
  Compute $d\left(\widehat{M}_i^{\text{SURE}}(w), \overline{X}_i^{\text{LE}}\right)$
**end**
Compute $O_i$ as the minimal distance between $X_i$ and all the class means $\left\{\widehat{M}_i^{\text{SURE}}(w)\right\}$
Update $w$ with SGD, to reduce the classification loss

the saved class means need to be calculated. The real-valued distances go through standard CONV and FC layers for the final classification into semantic categories.

**C-SURE Prototype Feature Layer.** This layer consists two parts, C-SURE and DIST in Fig.1. The feature from the wFM convolutional layer becomes the input $X$, and it is processed per class as well as per mixture component $k$. The output is the minimal distance between each instance and the C-SURE estimate of all the class means (Algorithm 1).

**Discussions.** We have proposed a novel JS shrinkage estimator on the manifold of complex values and used the SURE estimate to compute the FM on the manifold. This method is used to calculate the class-specific distribution mean on the manifold and implement a prototype-based classifier.

The dominance of the JS estimator over MLE is a statistical property for some fixed data. When the shrinkage estimator is incorporated into the loop of deep learning for classification, it is unclear whether the shrinkage estimator has any practical advantage over the simpler MLE.

There are two major changes to the fixed data mean estimation setting: **1)** Both the data and the estimator are changing during learning; **2)** The final task performance is not critically dependent on how well the estimator fits the data mean, but on how well the feature derived from the distance to data prototypes separates classes.

Therefore, while JS dominates MLE, it is unclear whether a prototype CNN classifier with a built-in JS estimator would be theoretically superior to a purely discriminative classifier such as the SurReal CNN classifier [6].

We turn to experiments on complex-valued data classification to test our ideas and validate our approach in practice.

## 4. Experimental Results

We compare our C-SURE prototype classifier against two baselines. **1)** The first baseline is a real-valued CNN classifier which ignores the geometry of complex numbers and treats each complex value as two independent real numbers. **2)** The second baseline is the complex-valued SurReal discriminative classifer our model is built on.

We experiment on two complex-valued datasets: MSTAR and RadioML. Our results demonstrate that our model-based classifier is more accurate (Table 1), more stable and robust (Fig. 5). Like SurReal, our model is also much smaller, outperforming the real-valued CNN on MSTAR with less than $1\%$ of the model size (Fig. 2).

Table 1: Comparison of Classification Accuracies.

| Dataset | Real-Valued | SurReal | C-SURE |
|---|---|---|---|
| MSTAR-L | 99.1% | **99.2%** | **99.2%** |
| MSTAR-S | 97.4% | 97.7% | **98.1%** |
| RadioML | 75.8% | 78.4% | **81.6%** |

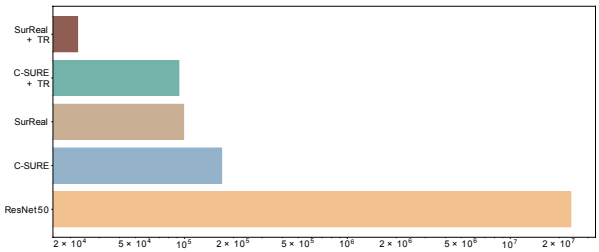


Figure 2: MSTAR model size comparison. Each horizontal bar indicates the total number of parameters on a *log* scale. There are 5 models: two-channel real-valued baseline ResNet50, the complex-valued baseline SurReal, our complex-valued C-SURE, and the latter two models implemented with the tensor ring trick [22] and marked by +TR. Our C-SURE classifier has more parameters than the SurReal model that it adapts from. Like SurReal, C-SURE is smaller than $1\%$ of the real-valued CNN baseline ResNet50.

All the experiments are trained on a GeForce RTX 2080 GPU for a total of 120 epochs, using Adam optimizer and cross-entropy loss. The batch size is 100 for MSTAR and 400 for RadioML. The learning rate is 0.015 for MSTAR and 0.03 for RadioML.

### 4.1. MSTAR Target Classification

**MSTAR Data.** The dataset contains complex-valued SAR images of 11 classes [18]. We create two random subsets, large (L) and small (S), from the original MSTAR dataset. The small set is entirely contained in the large set. We center-crop the SAR images into $100 \times 100$ pixels, and convert the complex values into the polar form.

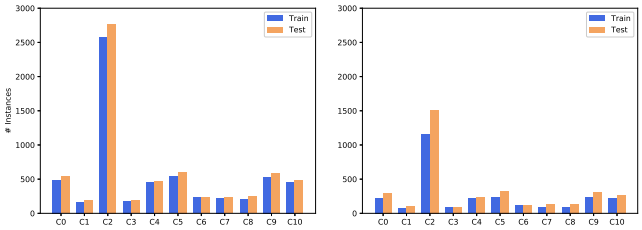


Figure 3: MSTAR large (L) and small (S) subsets have highly imbalanced classes. C2 is the largest class, C0, C4, C5, C9, C10 come next at about 20% of the size of C2, and C1, C3, C6, C7, C8 are at about 8% of the size of C2.

**Real-valued CNN Baseline.** We use ResNet50 [16] and feed the complex-valued image as a (real,imaginary) two-channel real-valued image. Fig. 2 compares the model size among different approaches and implementations. Even with the need to store the class prototypes, our C-SURE classifier remains light-weight like SurReal, with less than 1% of the ResNet50 size.

**Accuracy and Robustness.** Table 1 shows that C-SURE is overall more accurate than SurReal and ResNet50, and the gain is larger for the small dataset, with least confusion between classes (Fig. 4). This slight effect is consistent with the idea of using prototypes for few-shot recognition.

Fig. 5 compares how the training and testing accuracy evolves during training. C-SURE seems not only more stable and fast converging as the training accuracy plateaus sooner, but also more robust as it has the least performance gap between training and testing.

### 4.2. RadioML Modulation Classification

**RadioML Data.** They are synthetically generated radio signals with modulation operating over both voice and text data. Noise is added further for channel effects. Each signal is tagged with a signal-to-noise ratio (SNR), in the range of $[-20, 18]$ with an increment step of 2. There are 11 types of modulations; each type has $20,000$ instances. The data is split $50/50$ between training and testing.

**Real-valued CNN Baseline.** We use the ÓShea's model [23] and feed the complex-valued RF time series of length 128 as a (real,imaginary) two-channel signal. Our C-SURE classifier is less than 3% of the size of ÓShea's model.

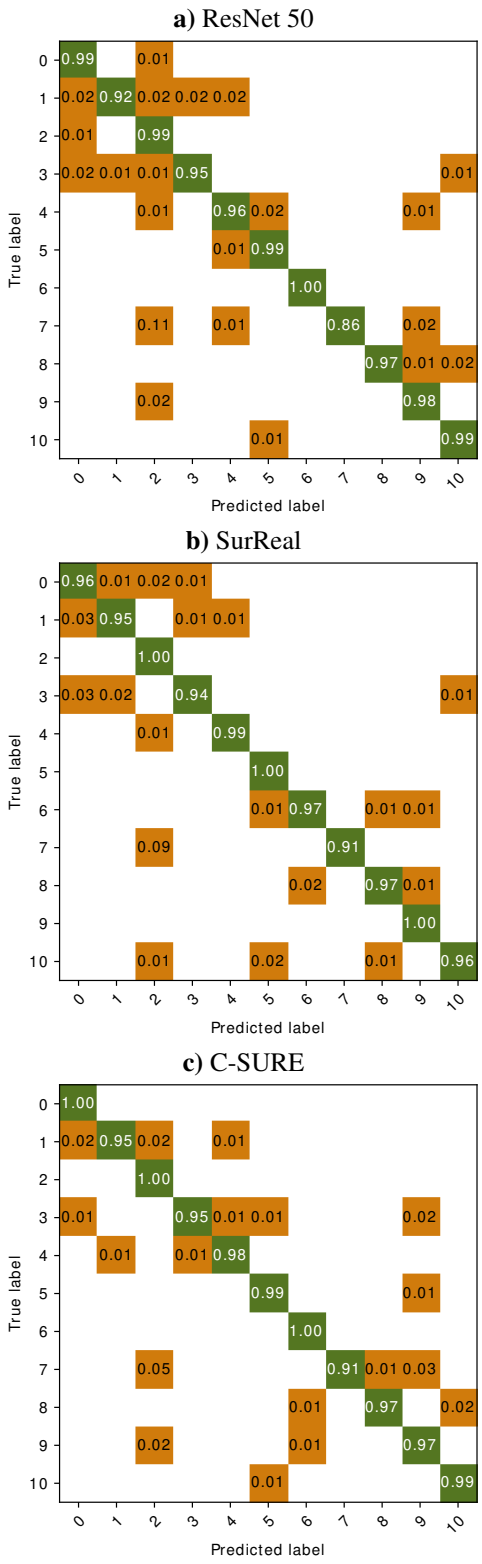


Figure 4: C-SURE has the least confusion between classes on our MSTAR small set. Among the three classifiers, the confusion matrix for our C-SURE has the largest values on the diagonal and the smallest values off the diagonal.

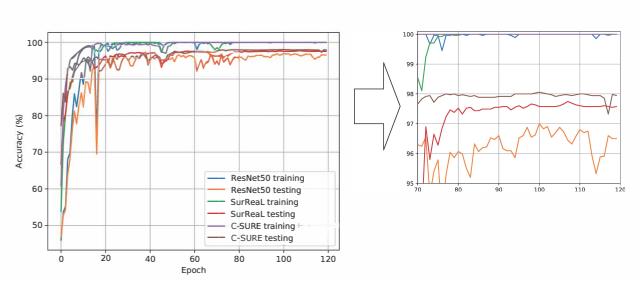


Figure 5: C-SURE is more robust, stable and fast converging. Shown here is the classification accuracy over training epochs on the MSTAR small set, for ResNet50, SurReal, and C-Sure. All three models have a significant performance gap between training and testing. However, C-SURE has the least gap and is more robust. C-SURE is also more stable and fast converging, as the training accuracy plateaus sooner.

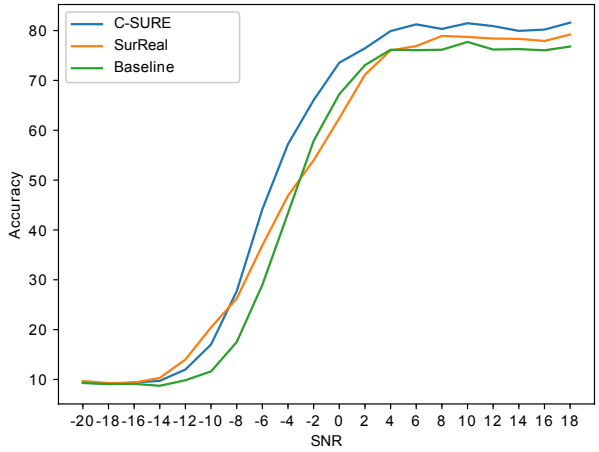


Figure 6: C-SURE has a higher test accuracy than baselines overall. C-SURE outperforms the real-valued baseline at every SNR; C-SURE outperforms SurReal when SNR$> -8$, and the gain is larger when SNR$\in [-8, 8]$.

**Accuracy over SNR.** Fig. 6 compares the test accuracy at various SNR's. When the SNR is too low or too high, all three models become almost equally poor or good. Nevertheless, C-SURE is more accurate than the real-valued baseline at every SNR, than SurReal when SNR$> -8$, with a larger gain in the middle SNR range of $[-8, 8]$.

### 4.3. Shrinkage or MLE in A Prototype Classifier?

Our C-SURE classifier differs from SurReal, the complex-valued classifier baseline, on two aspects: It models class prototypes explicitly and uses the shrinkage estimator on the manifold of complex values. If we fix the model as a prototype classifier and vary the amount of shrinkage, we can tease out the contribution of the shrinkage estimator against the standard MLE in our C-SURE classifier.

Table 2: C-SURE with Varying Hyperparameter $v$

| **Dataset** | **Variance** $v$ | **Accuracy** (%) |
|---|---|---|
| **MSTAR** | (MLE) 0 | 98.8 |
| | 1 | **99.2** |
| | 10 | 97.5 |
| **RadioML** | (MLE) 0 | 80.7 |
| | 1 | **81.6** |
| | 10 | 78.6 |

The C-SURE shrinkage estimator has a hyperparameter $v$ specifying the data variance in the hierarchical Bayesian model. When $v = 0$, there is no shrinkage adjustment from the prior distribution, and the estimator is reduced to MLE.

Table 2 lists the test accuracies on both MSTAR and RadioML tasks as we vary the hyperparameter $v$. There is always a shrinkage estimator at $v > 0$ better than MLE – the shrinkage estimator at $v = 0$, validating the benefit of utilizing a shrinkage estimator in a prototype CNN classifier.

While C-SURE seems to be able to outperform SurReal, the size of gain remains small. More controlled and careful experimentation would be needed to clarify which data classification scenarios C-SURE would be best at.

## 5. Summary

Most existing deep learning approaches assume data lying in a vector space. We consider the complex-valued data, where the range of the data is no longer in the Euclidean space. The SurReal complex-valued classifier outperforms the real-valued CNN baseline with a significantly reduced model size [6], based on computing the geometric mean on the manifold (i.e., FM) for complex-valued data.

We propose C-SURE, a novel shrinkage estimator on the complex-valued manifold with provably smaller MSE than FM. We further incorporate it into learning a nearest prototype CNN classifier, by adapting SurReal's model architecture and with only a slight increase in the model size.

On complex-valued MSTAR and RadioML datasets, our experimental results suggest that our C-SURE classifier tends to be more accurate and robust than SurReal, and the shrinkage estimator is always better than MLE for the same prototype classifier.

More experimentation would help clarify the strengths and weaknesses of the C-SURE classification approach.

## Acknowledgements

This research was supported, in part, by Berkeley Deep Drive and DARPA.

## References


[1] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007. 3

[2] James O Berger, William E Strawderman, et al. Choice of hierarchical priors: admissibility in estimation of normal means. *The Annals of Statistics*, 24(3):931–951, 1996. 1

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 1

[4] Rudrasis Chakraborty, Monami Banerjee, and Baba C Vemuri. H-cnns: Convolutional neural networks for riemannian homogeneous spaces. *arXiv preprint arXiv:1805.05487*, 2, 2018. 1

[5] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. Manifoldnet: A deep network framework for manifold-valued data. *arXiv preprint arXiv:1809.06211*, 2018. 1

[6] Rudrasis Chakraborty, Jiayun Wang, and Stella X Yu. Surreal: Frechet mean and distance transform for complex-valued deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 4, 5, 7

[7] Rudrasis Chakraborty, Chun-Hao Yang, Xingjian Zhen, Monami Banerjee, Derek Archer, David Vaillancourt, Vikas Singh, and Baba Vemuri. A statistical recurrent model on the manifold of symmetric positive definite matrices. In *Advances in Neural Information Processing Systems*, pages 8883–8894, 2018. 1

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016. 1

[9] Michael J Daniels and Robert E Kass. Shrinkage estimators for covariance matrices. *Biometrics*, 57(4):1173–1184, 2001. 2

[10] Bradley Efron and Carl Morris. Stein's estimation rule and its competitors—an empirical bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973. 1

[11] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. In *International Conference on Learning Representations*, 2018. 1

[12] Greg M Fleishman, P Thomas Fletcher, Boris A Gutman, Gautam Prasad, Yingnian Wu, and Paul M Thompson. Geodesic refinement using james-stein estimators. *Mathematical Foundations of Computational Anatomy*, 60, 2015. 2

[13] Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310, 1948. 3

[14] William T. Freeman and Edward H Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):891–906, 1991. 1

[15] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] William James and Charles Stein. Estimation with quadratic loss. In *Breakthroughs in statistics*, pages 443–460. Springer, 1992. 1

[18] Eric R Keydel, Shung Wu Lee, and John T Moore. Mstar extended operating conditions: A tutorial. In *Algorithms for Synthetic Aperture Radar Imagery III*, volume 2757, pages 228–242. International Society for Optics and Photonics, 1996. 6

[19] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004. 2

[20] Michael Maire, Takuya Narihira, and Stella X Yu. Affinity cnn: Learning pixel-centric pairwise relations for figure/ground embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 174–182, 2016. 1

[21] Jonathan H Manton, Vikram Krishnamurthy, and H Vincent Poor. James-stein state filtering algorithms. *IEEE Transactions on Signal Processing*, 46(9):2431–2447, 1998. 2

[22] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. 5

[23] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. Convolutional radio modulation recognition networks. In *International conference on engineering applications of neural networks*, pages 213–226. Springer, 2016. 6

[24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 1

[25] Armin Schwartzman. *Random ellipsoids and false discovery rates: Statistics for diffusion tensor imaging data*. PhD thesis, Stanford University, 2006. 3

[26] Nir Sochen, Ron Kimmel, and Ravi Malladi. A general framework for low level vision. *IEEE transactions on image processing*, 7(3):310–318, 1998. 1

[27] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981. 2

[28] Yue Wu, Brian Tracey, Premkumar Natarajan, and Joseph P Noonan. James–stein type center pixel weights for non-local means image denoising. *IEEE Signal Processing Letters*, 20(4):411–414, 2013. 2

[29] Chun-Hao Yang and Baba C Vemuri. Shrinkage estimation on the manifold of symmetric positive-definite matrices with applications to neuroimaging. In *International Conference on Information Processing in Medical Imaging*, pages 566–578. Springer, 2019. 1, 2, 3, 4

[30] Stella Yu. Angular embedding: A robust quadratic criterion. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):158–173, 2011. 1