CAUSAL COVARIATE SHIFT CORRECTION USING FISHER INFORMATION PENALTY

Anonymous authors
Paper under double-blind review

A APPENDIX

In this section, we describe the relationship between relative entropy and fisher information. We also present the baselines, datasets details, C^3 batchwise performance λ selection details, and experimental setup.

A.1 REPRESENTING THE CURRENT DERIVATIVE WITH THE FISHER INFORMATION MATRIX

Let us consider having a model with parameter θ and a likelihood function $p(X \mid \theta)$, where X is observed data. The estimate of true parameter θ can be found by using estimator $\hat{\theta}$. The Fisher information $I(\theta)$ can be defined as the expected value of the negative hessian of the log-likelihood function.

$$I(\theta) = \mathbb{E}\left[-\frac{\partial^2 \log p(X \mid \theta)}{\partial\theta\partial\theta^T}\right] \tag{1}$$

The Cramér-Rao Lower Bound (CRLB) states that for any unbiased estimator $\hat{\theta}$, the variance-covariance matrix $V(\hat{\theta})$ satisfies the inequality property:

$$V(\hat{\theta}) \succeq I^{-1}(\theta) \tag{2}$$

The symbol $\succeq$ represents the following matrix inequality$V(\hat{\theta}) - I^{-1}(\theta)$ positive and semi-definite.

Now let us assume $q(\hat{\theta})$ as Gaussian distribution function to be estimated around parameter θ mean and variance-covriance matrix $V(\hat{\theta})$ in such manner that:

$$q(\hat{\theta}) \approx \mathcal{N}(\theta, V(\hat{\theta})) \tag{3}$$

We have a model $f(x)$ which outputs a target distribution $Q(y|x)$ for each given input x. The D_{KL} divergence for source to target distribution can be found by $D_{KL}(P\,||\,Q) = \sum_y P(y|x) \log\left(\frac{P(y|x)}{Q(y|x)}\right)$.
Considering y as continuous target variable, $P(y|x)$ and $Q(y|x)$ as Gaussian distributions with means μ_P and μ_Q and variances σ_P^2 and σ_Q^2.
Relative entropy can be computed in closed form using mean-variance of source and target distribution as follows:

$$D_{KL}(P\,||\,Q) = \frac{1}{2}\left[\log\left(\frac{\sigma_Q^2}{\sigma_P^2}\right) + \frac{\sigma_P^2 + (\mu_P - \mu_Q)^2}{\sigma_Q^2} - 1\right] \tag{4}$$

The D_{KL} can be approximated as:

$$D_{KL}(p(\theta) \parallel q(\hat{\theta})) \approx \int p(\theta) \log\left(\frac{p(\theta)}{\mathcal{N}(\theta, V(\hat{\theta}))}\right) d\theta \tag{5}$$

With the help of CRLB we can replace $V(\hat{\theta})$ with $I^{-1}(\theta)$ as $V(\hat{\theta}) \succeq I^{-1}(\theta)$, we get:

$$D_{KL}(p(\theta) \parallel q(\hat{\theta})) \approx \int p(\theta) \log\left(\frac{p(\theta)}{\mathcal{N}(\theta, I^{-1}(\theta))}\right) d\theta \tag{6}$$

which is the estimation of relative entropy by using a variance-covariance matrix of estimated parameters with the help of FIM.

A.2 THE FISHER INFORMATION MATRIX AS AN APPROXIMATION OF VARIATIONAL POSTERIORS

Before introducing the penalty term which is one of our contributions we investigated the relation between relative entropy (D_{KL}) and FIM. Lets assume that θ is estimated parameter for given input data folds i.e (X_1,X_2,X_3,..., X_n) with a probability function $P(x;\theta)$. By using an unbiased estimator $\hat{\theta}(X_1, X_2, \ldots, X_n)$ of θ, the variance estimator satisfies the following CRLB property.

$$\sigma^2(\hat{\theta}) \geq \frac{1}{nI(\theta)} \tag{7}$$

where $I(\theta)$ is Fisher information and n is sample size which can be described as:

$$I(\theta) = -\mathbb{E}\left[\frac{\partial^2 \log P(X;\theta)}{\partial \theta^2}\right] \tag{8}$$

It is crucial to understand the relation of FIM to D_{KL} (D_{KL}). We can find D_{KL} between source to target distributions $P(x)$ and $Q(x)$ with the same support set of X with K number of folds by:

$$D_{KL}(P\|Q) = \int_X P(x) \log\left(\frac{Q(x)}{P(x)}\right) dx \tag{9}$$

If we assume $P(x;\theta)$ as true distribution for given input X with parameter θ and $Q(x;\hat{\theta})$ as arbitrary target distribution with parameter $\hat{\theta}$ then we can rewrite D_{KL} as:

$$D_{KL}(P(\cdot;\theta) \parallel Q(\cdot;\hat{\theta})) = \mathbb{E}_{X\sim P(\cdot;\theta)}\left[\log\left(\frac{Q(X;\hat{\theta})}{P(X;\theta)}\right)\right] \tag{10}$$

Consider the special case of $Q(x;\hat{\theta})$ parameterized by $\hat{\theta}$ whereby we want to minimize D_{KL} w.r.t $\hat{\theta}$. For this case D_{KL} is at minimum if we have $Q(x;\hat{\theta})$ = $P(x;\hat{\theta})$. Thus we get:

$$D(P(\cdot;\theta) \parallel P(\cdot;\hat{\theta})) \geq 0 \tag{11}$$

By applying Taylor expansion up to second-order to the log $P(x;\hat{\theta})$ for true parameter θ we have:

$$\begin{aligned} \log P(X;\hat{\theta}) = \log P(X;\theta) \\ + (\hat{\theta}-\theta)\frac{\partial \log P(X;\theta)}{\partial \theta} \\ - \frac{1}{2}(\hat{\theta}-\theta)^2 \frac{\partial^2 \log P(X;\theta)}{\partial \theta^2} + O((\hat{\theta}-\theta)^3) \end{aligned} \tag{12}$$

By taking expectation w.r.t X we have:

$$\begin{aligned} \mathbb{E}_{X\sim P(\cdot;\theta)}\left[\log P(X;\hat{\theta}) - \log P(X;\theta)\right] = \\ (\hat{\theta}-\theta)\mathbb{E}_{X\sim P(\cdot;\theta)}\left[\frac{\partial \log P(X;\theta)}{\partial \theta}\right] \\ -\frac{1}{2}(\hat{\theta}-\theta)^2 I(\theta) + O((\hat{\theta}-\theta)^3) \end{aligned} \tag{13}$$

The left-hand in above mentioned equation is D_{KL} i.e $D(P(\cdot;\theta) \parallel P(\cdot;\hat{\theta}))$. As we know that D_{KL} is always non-negative, as so the right-hand side must also be non-negative. Thus we get:

$$\left(\frac{\hat{\theta}-\theta}{2}\right) I(\theta) \geq 0 \tag{14}$$

It will hold for any $\hat{\theta}$, from this we can conclude that:

$$I(\theta) \geq 0 \tag{15}$$

which is Fisher information. The following algorithm 1 provides an overview of our proposed method C^3.

Algorithm 1 Dataset fragmentation and causal covariate shift correction

Require: model $f(\theta)$ parameterized by θ;
training dataset $\mathcal{D}_{tr}$;
validation data $\mathcal{D}_v$;
number of batches K
number of epochs T

procedure SHIFTCORRECTION($\mathcal{D}_{tr}$, $\mathcal{D}_v$)
split $\mathcal{D}_{tr}$ into K batches
initialize $f(\theta)$ and $\mathcal{L}(x, y; \theta)$
  for epoch $\leftarrow 1$ **to** T **do**
    for $i \leftarrow 1$ **to** K **do**
      for $j \leftarrow i + 1$ **to** K **do**
        $\mathrm{D}_{KL}(D_i, D_j)$
        for each pair (D_i, D_j): $\mathcal{L}(x, y; \theta) = -\int P(y(x)) \log(P(y|x; \theta))d\theta - \lambda \times \int \frac{\partial^2 \log p(X|\theta)}{\partial\theta\partial\theta^T} d\theta$
      end for
    end for
    update $f(\theta)$ using $\mathcal{L}(x, y; \theta)$
  end for
  return $f(\theta)$
end procedure

B EXPERIMENTS

In this section, we demonstrate the efficacy of C^3 against multiple baseline settings for causal covariate shift and on the benchmarks for natural covariate shift as a surrogate.

1. **Baselines:**
   There are five baselines in our experiment:
   - **B1:** *Clean* Verifying the effectiveness of C^3 on datasets without any covariate shift.
   - **B2:** *Natural shift Consequences* Analyzing the performance of the C^3 in the presence of natural covariate shift.
   - **B3:** *Causal shift Consequences* Analyzing C^3 performance in the presence of causal shift caused by dataset fragmentation.
   - **B4:** *Loss Recaliberation* Recaliberating the loss function and then measure the performance of C^3.
   - **B5:** *Correction* correction of natural covariate shift via proxy with C^3.
2. **Model architecture:** We used a five-layer convolutional neural network (CNN) with softmax cross-entropy loss. Our CNN model consists of 2 convolutional layers with pooling, and 3 fully connected layers. The model architecture for all image-based benchmarks remains consistent, for tabular datsets the model architecture differs from image-based but remains the same for all tabular datasets. We used a multi-layer perceptron network for tabular data with a hidden layer with 4 neurons, relu as an activation function, and Adam optimizer. We set the hyper-parameter λ value within the range (0.01, 0.04, 0.07, 0.1) in all of our experiments. We present λ = 0.1 results in this paper for all of our experiments. All of our baselines are implemented in TensorFlow 2.11 [1] and the code is anonymously available at [2].
3. **Machine Specification:** We run all of our experiments on RTX 3090 Ti with 24 GB GPU memory and 128 GB system memory.
4. **Benchmarks:** We compare the performance of C^3 with standard cross validation and significant importance based methods like: importance weighting (IW) Huang et al. (2006),

[1] www.tensorflow.org
[2] https://anonymous.4open.science/r/C3-C908/MNIST-Batchwise

importance weighting cross-validation (IWCV) Sugiyama et al. (2007), kernel mean matching (KMM) Gretton et al. (2009) and dynamic importance weighting (DIW) Fang et al. (2020). They were strategically chosen to represent landmark literature and current state-of-the-art.

5. **Datasets:** To compare the effectiveness of our developed method C^3 we used 40 real-world benchmarking datasets. To evaluate our method we used 13 image-based datasets benchmarks and 27 binary datasets from KEEL repository as benchmarks Alcalá-Fdez et al. (2011). The used image-based benchmarks, comprising: MNIST LeCun (1998), Fashion-MNIST Xiao et al. (2017), Kuzushiji-MNIST Clanuwat et al. (2018), Permuted-MNIST Goodfellow et al. (2013), MNIST-C Mu & Gilmer (2019) , SVHN Netzer et al. (2011), Caltech101 Fei-Fei et al. (2004), Tiny ImageNet Krizhevsky et al. (2009), STL-10 Coates et al. (2011) CIFAR-10 and CIFAR-100 Krizhevsky et al. (2009), CIFAR10-C and CIFAR100-C Hendrycks & Dietterich (2019).
6. **Calibrating the penalty:** One of our contributions to the paper is introducing the penalty term as above mentioned. We calibrate the penalty term (λ) with different values in batch/fold setup and present the result in Fig 1. We report $\lambda = 0.1$ results in our paper because our method C^3 is more robust to covariate shift. In figure 1 we can observe that C^3 performs better when we set $\lambda = 0.1$ as compared to other values.

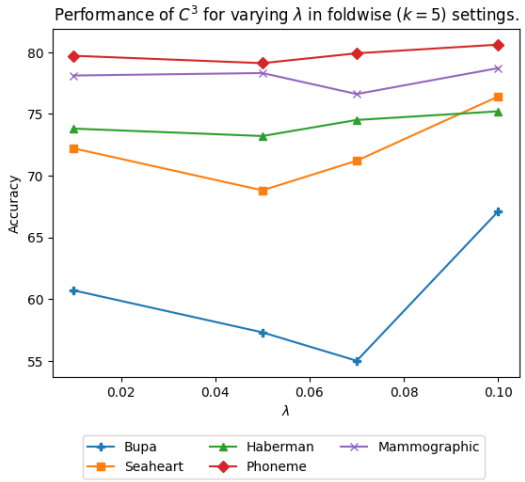

Figure 1: Performance of C^3 for varying λ in foldwise ($k = 5$) settings.

7. **Experimental design:** To study the effect of causal covariate shift caused by fragmentation, we perform evaluations on datasets with natural covariate shift and also on clean (free of covariate shift) datasets. We use accuracy as the first and direct evaluation metric in all experiments. We run each experiment 5 times and report average results due to spatial constraints.

C Discussion

To verify the effectiveness of C^3, we perform batchwise experiments for causal covariate shift whose results are presented in Table 1 which also validates **B4 & B5**. We consider batchwise holdout cross-validation as a baseline in comparison to C^3. To ensure better performance of C^3 we compare the mean accuracy over all batches μ_1 of Table 2 and μ_2 of Table 1. We report accuracy for each single batch as well in all experimental settings to verify C^3 performance. We then consider C^3 with the whole dataset as a baseline for our C^3 batchwise method.
The Δ_3 of Table 1 presents the difference between μ_2 of Table 1 and μ_1 of Table 2. The Δ_3 shows improvement in accuracy and provides support to our claim of causal covariate shift correction **B5**. To verify **B5** we executed C^3 in batchwise settings on all dataset which results are reported in Table

1.

Our proposed method C^3, shows improvement in accuracy in almost every batchwise setting and for each batch also as compared to the baseline. To validate the adaptive nature of C^3 to natural shift correction, we perform experiments on above mentioned datasets with natural shift. We notice that C^3 is able to correct natural shift when it tries to correct causal shift. C^3 shows improvement in accuracy for almost all benchmarks, like it shows 5%, 13.9%, and 8.6% improvement for Kuzushiji-MNIST, CIFAR10-C, and Fashion-MNIST with 20 batch split. C^3 also adapts to natural shift when it tries to correct causal covariate shift.
C^3's accuracy improves as the number of batches decreases, due to statistics getting more robust with larger supports. It is shown in Table 1. C^3 7.5% for Fashion-MNIST and 6.9% improvement in accuracy in 10 batch setup as compared to CV with the same batch setup. C^3 improves in accuracy with 7.2%, 7.2%, and 2.1% for Fashion-MNIST, Kuzushiji-MNIST, and Permuted-MNIST when batch size is 6. For the batch size 5, the improvement is 9.7%,8.1%, and 7.3% for CIFAR100, Kuzushiji-MNIST, and Fashion-MNIST. In 4 batches scenario we report 11.3%, 6.9%, 6.1%, and 5.8% improvement in accuracy for CIFAR-100, Khushiji-MNIST, Fashion-MNIST, and CIFAR100-C. In the case of 2 batches the improvement in accuracy is 20.3%, 15.5%, 6.6%, and 6.3% for CIFAR-10, CIFAR-10, CIFAR100-C, and Khushiji-MNIST. Overall, C^3 outperforms in the batchwise case and in the case where a complete dataset is provided with other benchmarking methods, and results are discussed ahead in the comparison with SOTA.

REFERENCES

Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. *Advances in neural information processing systems*, 33: 11996–12007, 2020.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop*, 2004.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, Bernhard Schölkopf, et al. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3 (4):5, 2009.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Dataset	Baseline		Batchwise accuracy						Mean	Variance	$\Delta_3 = \mu_2 - \mu_1$
	CV	**C^3**	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	μ_2	σ_2^2	$\Delta_3(\%)$
Training data = 5% , Number_of_Batches = 20											
MNIST	94.8	97.9	90.7	90.6	91	91.7	91.4	91.8	91.2	0.09	$\uparrow 2.5$
Permuted-MNIST	95.1	97.6	91.1	89.8	90.2	90.4	91.5	90.7	90.3	0.26	$\uparrow 1.9$
Fashion-MNIST	83.1	88.4	81.5	81.7	81.2	81.4	81.5	81.9	81.5	0.058	$\uparrow 8.6$
Kuzushiji-MNIST	75.4	89.2	68.5	69.4	67.2	68	67.8	66.9	68.4	0.63	$\uparrow 5.0$
CIFAR-10	71.5	88.7	50.9	51.4	52.2	48.9	50.3	57.4	51.8	7.18	$\uparrow 1.9$
CIFAR-100	38.2	58.7	23.9	18.2	18.5	17.8	23.9	18.3	20.1	7.26	$\uparrow 1.8$
CIFAR10-C	63.9	73.3	46.4	54.3	57.8	61.1	61.5	61.8	57.2	30.1	$\uparrow 13.9$
CIFAR100-C	28.8	39.4	11.9	17.2	18.5	21.3	22.1	24.9	19.3	17.1	$\uparrow 0.9$
Training data = 10% , Number_of_Batches = 10											
MNIST	94.8	97.9	91.9	91.7	91.2	91.8	91.3	91.8	91.7	0.08	$\uparrow 0.6$
Permuted-MNIST	95.1	97.6	91.6	91.9	91.3	91.6	90.1	91.2	91.5	0.31	0
Fashion-MNIST	83.1	88.4	79.5	82.4	81.6	79.5	82.3	81.9	81.2	1.21	$\uparrow 7.5$
Kuzushiji-MNIST	75.4	89.2	71.4	70.4	71.7	70.7	70.5	70.9	70.9	0.71	$\uparrow 6.9$
CIFAR-10	71.5	88.7	52.1	53.1	48.5	59.3	52.5	55.7	53.5	11.09	$\uparrow 0.9$
CIFAR-100	38.2	58.7	27.2	25.8	20.4	17.0	21.9	22.8	22.5	11.3	$\uparrow 1.0$
CIFAR10-C	63.9	73.3	52.7	59.9	61.9	64.4	66.1	65.7	61.7	21.1	$\uparrow 39.1$
CIFAR100-C	28.8	39.4	16.2	22.1	24.8	27.2	26.8	27.3	21.1	15.6	$\downarrow 1.7$
Training data = 15% , Number_of_Batches = 6											
MNIST	94.8	97.9	93.2	92.6	93.2	93.5	93.1	93.3	93.2	0.09	$\uparrow 1.7$
Permuted-MNIST	95.1	97.6	93.1	93	92.7	93.7	93.5	93.1	93.2	0.13	$\uparrow 2.1$
Fashion-MNIST	83.1	88.4	81.4	81.9	82.9	80.9	82.3	82.2	81.9	0.49	$\uparrow 7.2$
Kuzushiji-MNIST	75.4	89.2	73.8	73.9	74.4	74.6	73.9	74.2	74.2	0.13	$\uparrow 7.2$
CIFAR-10	71.5	88.7	53.4	57.8	55.9	54.1	56.1	58.2	55.9	3.7	$\uparrow 2.0$
CIFAR-100	38.2	58.7	25.8	22.4	26.6	23.4	22.9	24.7	24.3	2.34	$\uparrow 1.7$
CIFAR10-C	63.9	73.3	58.4	61.1	63.2	64.3	67.4	66.9	63.5	9.87	$\uparrow 42.4$
CIFAR100-C	28.8	39.4	21.9	25.5	27.2	28.7	28.5	29.5	26.8	6.6	$\uparrow 1.2$
Training data = 20% , Number_of_Batches = 5											
MNIST	94.8	97.9	93.6	93.8	94.3	93.7	93.8	–	93.8	0.07	$\uparrow 1.9$
Permuted-MNIST	95.1	97.6	94.1	93.9	93.6	94.1	94.3	–	94	0.07	$\uparrow 2.2$
Fashion-MNIST	83.1	88.4	82.8	83.1	82.1	81.4	82.6	–	82.4	0.44	$\uparrow 7.3$
Kuzushiji-MNIST	75.4	89.2	75.4	76.3	75.8	75.1	75.6	–	75.6	0.21	$\uparrow 8.1$
CIFAR-10	71.5	88.7	50.3	56.2	53.5	57.8	59.9	–	55.5	11.3	$\uparrow 18$
CIFAR-100	38.2	58.7	34.2	35.4	33.9	34.9	34.7	–	34.6	11.7	$\uparrow 9.7$
CIFAR10-C	63.9	73.3	58.4	63.3	64.3	66.5	66.2	–	63.7	8.53	$\uparrow 1.7$
CIFAR100-C	28.8	39.4	22.1	25.4	27.4	28.9	29.7	–	26.7	7.43	$\uparrow 2.9$
Training data = 25% , Number_of_Batches = 4											
MNIST	94.8	97.9	94.4	94.4	94.3	94.4	–	–	94.4	0.003	$\uparrow 2.0$
Permuted-MNIST	95.1	97.6	94.4	94.3	94.5	94.5	–	–	94.4	0.009	$\uparrow 2.8$
Fashion-MNIST	83.1	88.4	82.8	83.2	83.6	83.5	–	–	83.3	0.13	$\uparrow 6.1$
Kuzushiji-MNIST	75.4	89.2	77.2	75.5	77.6	75.3	–	–	76.4	1.37	$\uparrow 6.9$
CIFAR-10	71.5	88.7	56.8	57.3	62.2	63.4	–	–	59.9	8.47	$\uparrow 5.0$
CIFAR-100	38.2	58.7	33.9	34.1	34.7	33.5	–	–	34.1	0.18	$\uparrow 11.3$
CIFAR10-C	63.9	73.3	60.9	64.4	66.8	68.3	–	–	65.1	7.81	$\uparrow 3.7$
CIFAR100-C	28.8	39.4	24.7	28.2	29.7	31.9	–	–	28.6	6.86	$\uparrow 5.8$
Training data = 50% , Number_of_Batches = 2											
MNIST	94.8	97.9	95.9	96.1	–	–	–	–	96	0.02	$\uparrow 2.5$
Permuted-MNIST	95.1	97.6	95.7	96.1	–	–	–	–	95.9	0.08	$\uparrow 2.4$
Fashion-MNIST	83.1	88.4	84.2	84.4	–	–	–	–	84.3	0.02	$\uparrow 4.5$
Kuzushiji-MNIST	75.4	89.2	79.3	80.4	–	–	–	–	79.8	0.61	$\uparrow 6.3$
CIFAR-10	71.5	88.7	76.3	80.6	–	–	–	–	78.4	4.62	$\uparrow 20.3$
CIFAR-100	38.2	58.7	39.8	39.9	–	–	–	–	39.85	.002	$\uparrow 15.5$
CIFAR10-C	63.9	73.3	65.5	68.6	–	–	–	–	67.1	2.4	$\uparrow 2.8$
CIFAR100-C	28.8	39.4	31.2	34.8	–	–	–	–	33	3.24	$\uparrow 6.6$

Table 1: C^3 Batchwise Accuracy

Dataset	Baseline		Batchwise accuracy						Mean	Variance
	CV	C^3	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	μ_1	σ_1^2
Training data = 5% , Number_of_Batches = 20										
MNIST	94.8	97.9	88.1	89.3	87.9	89.9	88.9	88.8	88.7	0.49
Permuted-MNIST	95.1	97.6	88	86.1	88.9	88.7	87.2	88.3	88.4	1.41
Fashion-MNIST	83.1	88.4	72.7	73.7	74.2	72.5	74	70.6	72.9	1.94
Kuzushiji-MNIST	75.4	89.2	63.6	63.4	62.2	66.7	58.3	63.4	63.4	5.59
CIFAR-10	71.5	88.7	44.1	49.0	50.3	50.7	51.2	54.5	49.9	9.67
CIFAR-100	38.2	58.7	13.1	16.5	18.1	19.3	20.4	22.7	18.3	9.17
CIFAR10-C	63.9	73.3	19.3	20.1	16.3	16.1	14.9	10.2	16.2	10.4
CIFAR100-C	28.8	39.4	11.1	16.3	19.1	19.7	21.6	22.3	18.4	14.2
Training data = 10% , Number_of_Batches = 10										
MNIST	94.8	97.9	90.7	91.7	91.1	90.2	89.3	91.5	91.1	1.06
Permuted-MNIST	95.1	97.6	91.9	92.8	91.8	91.2	91.4	91.3	91.5	0.62
Fashion-MNIST	83.1	88.4	69.4	69.9	75.2	73.4	74.4	71.7	73.7	9.22
Kuzushiji-MNIST	75.4	89.2	64	63.6	64.1	64.9	65.2	64.5	64.0	1.15
CIFAR-10	71.5	88.7	47.9	52.8	52.9	53.7	54.5	54.1	52.6	4.87
CIFAR-100	38.2	58.7	17.3	20.3	22.2	23.3	24.7	21.2	21.5	5.51
CIFAR10- C	63.9	73.3	22.8	17.6	18.4	12.2	17.4	12.9	22.6	16.8
CIFAR100-C	28.8	39.4	15.5	21.9	25.2	26.6	27.1	20.5	22.8	16.3
Training data = 15% , Number_of_Batches = 6										
MNIST	94.8	97.9	90.8	90.7	91.7	92.2	92.2	91.6	91.5	0.43
Permuted-MNIST	95.1	97.6	91.2	90.2	91.2	92.1	90.6	91.1	91.1	0.41
Fashion-MNIST	83.1	88.4	74.4	75.7	77.7	75	70.6	74.7	74.7	5.40
Kuzushiji-MNIST	75.4	89.2	66.9	67	66.4	69.9	64.8	67.2	67.0	2.73
CIFAR-10	71.5	88.7	50.9	51.2	53.9	53.8	57.1	56.9	53.9	5.91
CIFAR-100	38.2	58.7	18.2	21.1	22.5	23.4	24.1	24.4	22.6	4.57
CIFAR10- C	63.9	73.3	18.3	25.1	23.9	20.7	20.9	21.1	21.6	4.99
CIFAR100- C	28.8	39.4	18.2	19.9	21.2	21.9	25.6	27.1	22.3	9.64
Training data = 20% , Number_of_Batches = 5										
MNIST	94.8	97.9	92.7	92.2	93.4	91.2	90.2	–	91.9	1.59
Permuted-MNIST	95.1	97.6	91	91.7	91.4	91.2	93.5	–	91.8	1.01
Fashion-MNIST	83.1	88.4	76.7	74.1	72	75.3	77.2	–	75.1	4.40
Kuzushiji-MNIST	75.4	89.2	68.3	69.2	67.7	65.5	66.8	–	67.5	2.02
CIFAR-10	71.5	88.7	36.9	38.5	37.5	36.8	37.9	–	37.5	0.50
CIFAR-100	38.2	58.7	19.7	22.3	23.5	24.4	24.9	–	22.9	3.43
CIFAR10- C	63.9	73.3	58.1	62.0	61.8	60.7	67.5	–	62.0	9.43
CIFAR100- C	28.8	39.4	20.0	23.2	23.2	26.7	26.1	–	23.8	5.77
Training data = 25% , Number_of_Batches = 4										
MNIST	94.8	97.9	92.1	93.6	92.5	91.3	–	–	92.4	0.92
Permuted-MNIST	95.1	97.6	92.1	92.2	90.5	91.6	–	–	91.6	0.61
Fashion-MNIST	83.1	88.4	74.6	77.5	78.1	78.5	–	–	77.2	3.12
Kuzushiji-MNIST	75.4	89.2	70.6	69.2	69.5	68.8	–	–	69.5	0.60
CIFAR-10	71.5	88.7	51.9	52.7	56.9	58.1	–	–	54.9	7.02
CIFAR-100	38.2	58.7	20.3	22.4	23.9	24.8	–	–	22.8	2.7
CIFAR10- C	63.9	73.3	57.6	62.1	63.1	62.6	–	–	61.4	4.81
CIFAR100- C	28.8	39.4	19.7	23.4	23.8	24.5	–	–	22.8	3.64
Training data = 50% , Number_of_Batches = 2										
MNIST	94.8	97.9	93.3	93.7	–	–	–	–	93.5	0.08
Permuted-MNIST	95.1	97.6	93.3	93.7	–	–	–	–	93.5	0.08
Fashion-MNIST	83.1	88.4	80	79.7	–	–	–	–	79.8	0.04
Kuzushiji-MNIST	75.4	89.2	73.6	73.3	–	–	–	–	73.5	0.04
CIFAR-10	71.5	88.7	56.2	60.1	–	–	–	–	58.1	3.81
CIFAR-100	38.2	58.7	23.2	25.4	–	–	–	–	24.3	1.21
CIFAR10- C	63.9	73.3	63.2	65.5	–	–	–	–	64.3	1.32
CIFAR100- C	28.8	39.4	25.3	27.5	–	–	–	–	26.4	1.21

Table 2: CV batch-wise accuracy