The Value of Out-of-Distribution Data

Ashwin De Silva[1,†], Rahul Ramesh[2,†], Carey E. Priebe[1], Pratik Chaudhari[2], Joshua T. Vogelstein[1]
[1]Johns Hopkins University, [2]University of Pennsylvania
[†]Equal Contribution
ldesilv2@jhu.edu, rahulram@seas.upenn.edu, cep@jhu.edu, pratikac@seas.upenn.edu, jovo@jhu.edu

Abstract

More data helps us generalize to a task. But real datasets can contain out-of-distribution (OOD) data; this can come in the form of heterogeneity such as intra-class variability but also in the form of temporal shifts or concept drifts. We demonstrate a counter-intuitive phenomenon for such problems: generalization error of the task can be a non-monotonic function of the number of OOD samples; a small number of OOD samples can improve generalization but if the number of OOD samples is beyond a threshold, then the generalization error can deteriorate. We also show that if we know which samples are OOD, then using a weighted objective between the target and OOD samples ensures that the generalization error decreases monotonically. We demonstrate and analyze this issue using linear classifiers on synthetic datasets and medium-sized neural networks on CIFAR-10.

1. Introduction

We procure more data with the goal of improving the generalization error. The central assumption of doing so—which is baked into learning theory [1]—is that this data comes from the desired task. But real data is often highly heterogeneous [2]; this heterogeneity can arise from nuisances, geometric ones such as viewpoint or semantic ones such as chairs of different shapes. Datasets curated at the Internet-scale [3] may also be susceptible to data poisoning attacks [4]. Such "out-of-distribution" (OOD) data, i.e., data that does not come from our desired task can be detrimental to performance.

We demonstrate a counter-intuitive phenomenon: generalization error on the target task can be a non-monotonic function of the number of OOD samples. In other words, there exist situations when a small number of OOD samples can improve the generalization error but if the number of OOD samples is beyond a threshold, then the generalization error deteriorates. **If we know which samples are OOD**, e.g., using a two-sample test to check for changes in the distribution [5], then the non-monotonic nature of the generalization error may be mitigated by simply ignoring the OOD samples. We show how one can do better: **using a weighted objective between the target and OOD samples, we can ensure that the generalization error on the target task decreases** monotonically with the number of OOD samples.

To be presented as a short paper at the Out-of-Distribution Generalization in Computer Vision (OOD-CV) workshop, ECCV 2022.

2. Generalization error is non-monotonic in the number of OOD samples

We define a task P as a joint distribution over the input domain X and the output domain Y. We model the heterogeneity in the dataset as two distributions: n samples drawn from a target task P_t and m samples drawn from an out-of-distribution (OOD) task P_o. We would like to minimize the generalization error $e_t(h) = \mathbb{E}_{(x,y)\sim P_t}\left[h(x) \neq y\right]$ on the target task. In order to do so, we may find a hypothesis that minimizes the empirical loss

$$\hat{e}(h) = \frac{1}{n+m}\sum_{i=1}^{n+m} \ell\left(h(x_i), y_i\right), \tag{1}$$

using the dataset $\{(x_i, y_i)\}_{i=1}^{n+m}$; here ℓ measures the discrepancy between the prediction $h(x_i)$ and the label y_i. If $P_t = P_o$, then classical results in learning theory show that $e_t(h) - \hat{e}(h) = \mathcal{O}((n+m)^{-1/2})$. But if $P_t \neq P_o$, then we should expect that the error on P_t of a hypothesis obtained by minimizing the average empirical loss can be sub-optimal, especially in cases when the number of OOD samples $m \gg n$.

2.1. An example using Fisher's Linear Discriminant

Consider a binary classification problem with one-dimensional inputs in Fig. 1. Target samples come from a Gaussian mixture model (with means $\{-\mu, \mu\}$ for the two

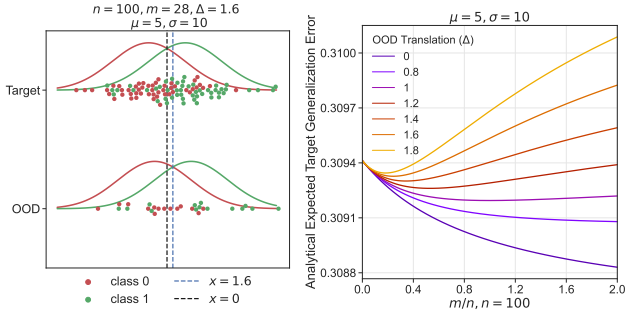


Figure 1. Left: A schematic of the Gaussian mixture model corresponding to the target task (top) and the OOD task (bottom). The OOD sample size ($m = 28$) at which the target generalization error is minimized at $\Delta = 1.6$ is indicated in the diagram. **Right:** For $n = 100$, we plot the generalization error of FLD on the target task as a function of the ratio of OOD and target samples m/n, for different OOD tasks corresponding to different values of Δ. This plot was computed using the analytical expression for the generalization error in (2); see Appendix B.6 for a numerical simulation study. For small values of Δ, when the two tasks are similar to each other, the generalization error $e_t(h)$ decreases monotonically. However, beyond a certain value of Δ, the generalization error is non-monotonic in the number of OOD samples. The optimal value of m/n which leads to the best generalization error is a function of the relatedness between the two tasks, as governed by Δ in this example. This non-monotonic behavior can be explained in terms of a bias-variance tradeoff with respect to the target task: a large number of OOD samples reduces the variance but also results in a bias with respect to the optimal hypothesis of the target task.

classes) and OOD samples are drawn from a Gaussian mixture with means $\{-\mu + \Delta, \mu + \Delta\}$; also see Appendix B.1. We can use Fisher's linear discriminant (FLD) to get an analytical expression for the generalization error of the target task

$$e_t(\hat{h}) = \frac{1}{2}\left[\Phi\left(\frac{m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right) + \Phi\left(\frac{-m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right)\right] \tag{2}$$

where Φ is the CDF of the standard normal distribution; see Appendices B.2 and B.3 for the derivation.

We can capture this discussion as a theorem, and our FLD example provides the proof.

Theorem 1. There exist target and OOD tasks P_t and P_o respectively such that the generalization error on the target task of the hypothesis that minimizes the empirical risk in (1) is non-monotonic in the number of OOD samples.

2.2. Demonstrating non-monotonic generalization error in neural networks

A non-monotonic trend in the generalization error can arise from geometric intra-class nuisances, which are very common in curated datasets [6]. We constructed sub-tasks from CIFAR-10 to study this aspect (Appendix C.1). We consider a binary classification problem (Bird vs. Cat) as the target task and introduce different kinds of OOD samples as images rotated by an angle between 0°-135°. Fig. 2 (left) shows that the generalization error decreases monotonically for small rotations but it is non-monotonic for larger angles.

Large datasets can contain categories whose appearance evolves in time (e.g., a typical laptop in 2022 looks very different from that of 1992), or categories can have semantic intra-class nuisances (e.g., chairs of different shapes). We split CIFAR-10 into five binary classification tasks to study this phenomenon (see Appendix C.1) and evaluated the trend in generalization error for all 20 distinct pairs of tasks. Fig. 2 (middle, red curves) illustrates non-monotonic trends for 3 such pairs; see Appendix C for more details.

3. Exploiting the non-monotonic nature of generalization error

Suppose we knew which samples in our dataset were OOD for the target task. It stands to reason that we should be able to not only mitigate the non-monotonic nature of the generalization error but also exploit it, as suggested below.

Theorem 2 (Paraphrased from [7]). For tasks P_t and P_o, let $\hat{h}_\alpha$ be the minimizer of the α-weighted empirical risk $\hat{e}_\alpha(h) = \alpha\hat{e}_t(h) + (1-\alpha)\hat{e}_o(h)$. The generalization error

$$e_t(\hat{h}) \leq e_t(h_t^*) + 4\sqrt{\left(\frac{\alpha^2}{n} + \frac{(1-\alpha)^2}{m}\right)\left(V_H - \log\delta\right)} + 2(1-\alpha)d_H(P_t, P_o),$$

with probability at least $1-\delta$. Here $h_t^* = \operatorname{argmin}_{h\in H} e_t(h)$ is the target error minimizer; V_H is a constant proportional to the VC-dimension of the hypothesis class H and $d_H(P_t, P_o)$ is a notion of relatedness between the tasks P_t and P_o.

In simple words, if we use an appropriate value of α that makes the second and third terms on the right-hand side small, then we can mitigate the deterioration of generalization error due to OOD samples. If the OOD samples are very different from those of the target task, i.e., if $d(P_t, P_o)$ is large, then this theorem suggests that we should pick an $\alpha \approx 1$. Doing so effectively ignores the OOD samples and the generalization error then decreases monotonically as $\mathcal{O}(n^{-1/2})$.

3.1. Choosing the optimal α^*

If we define $\rho = \frac{\sqrt{V_H - \log\delta}}{d_H(P_t, P_o)}$ to be, roughly speaking, the ratio of the capacity and the distance between tasks, then a short calculation shows that for $\alpha \in [0, 1]$

$$\alpha^*(n, m) = \begin{cases} 1 & \text{if } n \geq 4\rho^2, \\ \frac{n}{n+m}\left(1 + \sqrt{\frac{m^2}{4\rho^2(n+m) - nm}}\right) & \text{else.} \end{cases}$$

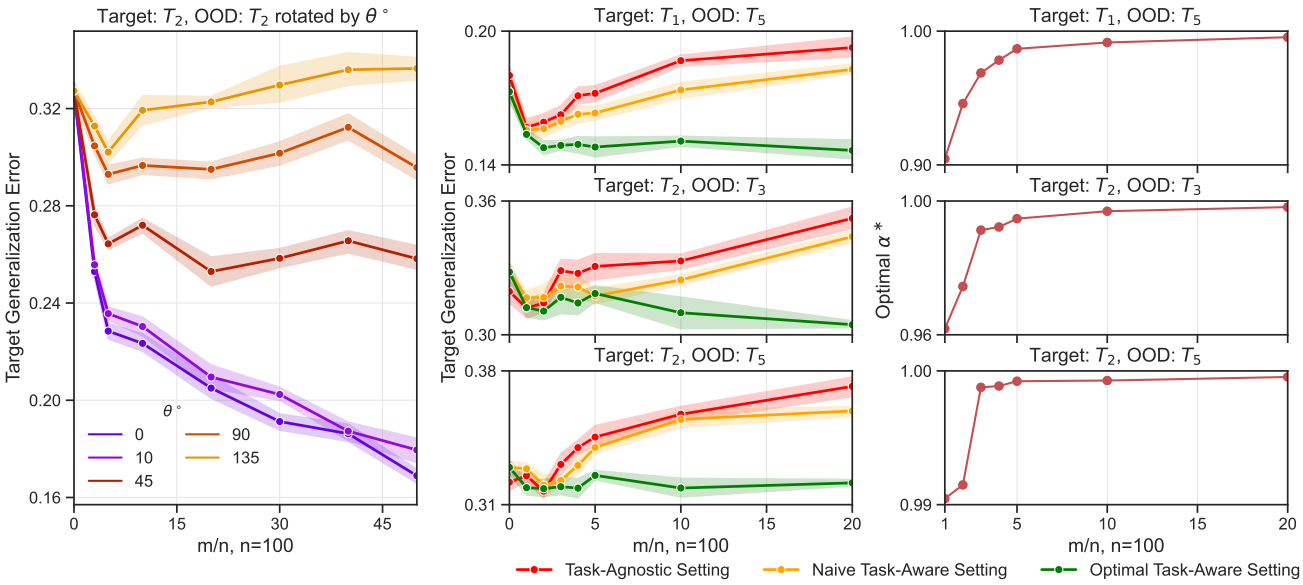


Figure 2. Left: A binary classification problem (Bird vs. Cat) is the target task and images of these classes rotated by different angles θ° are the OOD task. We see non-monotonic curves for larger values of θ°. For 135° in particular, the generalization error at $m/n = 50$ is worse than the generalization error with no OOD samples, i.e. OOD samples actively hurt generalization. **Middle:** Generalization error on the target task is plotted against the number of samples from the OOD task for 3 different pairs of target-OOD tasks constructed from CIFAR-10 for three settings: task-agnostic ERM where we minimize the total average risk over both tasks (red), an objective which minimizes the sum of the average loss of the target and OOD tasks which corresponds to $\alpha = 1/2$ (task-aware, yellow) and an objective which minimizes an optimally weighted convex combination of the target and OOD empirical loss (green). **Right:** The optimal α^* obtained via grid search for the three problems in the middle column plotted against different number of OOD samples. Note that the appropriate value of α lies very close to 1 but it is never exactly 1. In other words the OOD samples always benefit if we use the weighted objective in Theorem 2, even if this benefit is marginal in cases when OOD samples are very different from those of the target. See Appendix C.2 for experimental details and Appendix C.6 for experiments on more task pairs.

This suggests that if we have a hypothesis space with small VC-dimension or if the OOD samples and target samples come from very different distributions, then we should train only on the target samples to obtain optimal error. Otherwise, including the OOD samples after appropriately weighing them using α^* can give a better generalization error.

It is not easy to estimate ρ because it depends upon the VC-dimension of the hypothesis class [7, 8]. But in general, we can treat α as a hyper-parameter and use validation data to search for its optimal value. For our FLD example we can do slightly better: since we know the exact expression for the generalization error for the hypothesis that minimizes the α-weighted empirical loss (see Appendix B.4 and Appendix B.5), we can calculate α^* by numerically sweeping over $\alpha \in [0, 1]$.

Fig. 3 shows that regardless of the number of OOD samples (m) and the relatedness between the OOD and the target tasks (Δ), we can obtain a generalization error that is always better than that of a hypothesis trained without any OOD samples. In other words, if we choose α^* appropriately (the example in Fig. 1 corresponds to choosing $\alpha = 1/2$), then we do not suffer from non-monotonic generalization error on the target task.

3.2. Training neural networks using a weighted objective

In §2.2 we found that for some pairs of tasks, the generalization error is non-monotonic in the number of OOD samples. In this section, we show that if we knew which samples were OOD, then we can rectify this phenomenon by using an appropriate value of α^* to weigh the target and OOD samples. In Fig. 2 (middle), we track the test error of the target task for three settings: training is agnostic to the presence of OOD data (task agnostic setting in red), naively uses $\alpha = 1/2$ (yellow) and training uses an optimal value of α selected using a grid-search (green). We observe that searching over α improves the test error on all 3 pairs of target-OOD tasks. We discuss how data augmentation can also lead to non-monotonic behavior of the risk in Fig. 4.

Remark 3 (Sampling mini-batches during training). For $m \gg n$, mini-batches that are sampled uniformly randomly from the dataset will be dominated by OOD samples. As a result, the gradient even if it is still unbiased, is computed

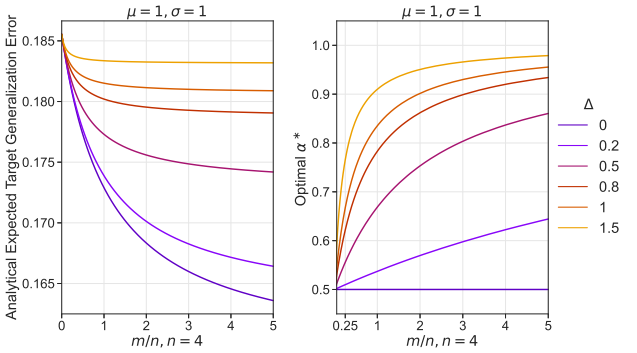


Figure 3. Left: Generalization error of the target task for the Gaussian mixture model using a weighted objective from Theorem 2 in place of classical FLD; see Appendix B.4. Note that unlike in Fig. 1, the generalization error monotonically decreases with the number of OOD samples m. **Right:** The optimal α^* that yields the smallest target generalization error as a function of the number of OOD samples. Note that α^* increases as the number of OOD samples m increases; this increase is more drastic for large values of Δ and is more gradual for small values of Δ. Observe that $\alpha^* = 1/2$ for all values of m if $\Delta = 0$. See Appendix B.6 for a numerical simulation study.

using very few samples from the target task. This leads to an increase in the test-error, which is particularly noticeable with α^* chosen appropriately after grid search. We therefore use a biased sampling procedure where each mini-batch contains a fraction β samples from the target task and the remainder $1 - \beta$ consists of OOD samples. This parameter controls the bias and variance of the gradient of the target task ($\beta = \frac{n}{n+m}$ gives unbiased gradients with respect to the unweighted total objective and high variance with respect to the target task when $m \gg n$, see Appendix C.4). We evaluated $\beta = \{0.5, 0.75\}$ and find that both improve the test error; see Fig. 7.

Remark 4 (Weighted objective for over-parameterized networks). It has been argued previously that weighted objectives are not effective for over-parameterized models such as deep networks because both surrogate losses $\hat{e}_t(h)$ and $\hat{e}_o(h)$ are zero when the model fits the training dataset [9]. It may therefore seem that the weighted objective in Theorem 2 cannot help us mitigate the non-monotonic nature of the generalization error; indeed the minimizer of $\alpha\hat{e}_t(h) + (1-\alpha)\hat{e}_o(h)$ is the same for any α if the minimum is exactly zero. Our experiments suggest otherwise: the value of α does impact the generalization error—even for deep networks. This is perhaps because even if the cross-entropy loss is near-zero for a deep network towards the end of training, it is never exactly zero.

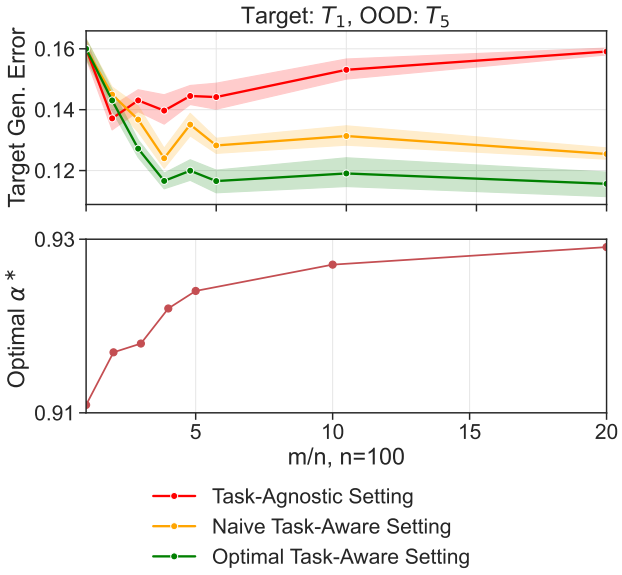


Figure 4. Data augmentation (padding with random cropping and random left/right flipping) during training has an interesting effect in the presence of OOD samples. Although the network trained in the task-agnostic setting (red) continues to perform poorly with lots of OOD samples, even a naive weighing of the target and OOD loss ($\alpha = 1/2$) is enough to provide a monotonically decreasing error (yellow). This suggests that data augmentation can mitigate some of the anomalies that arise from OOD data, although we can do better by addressing them specifically using, for instance, the weighted objective (green).

4. Discussion

While we are compelled to use generic supervised learning algorithms under the i.i.d assumption if we do not know about the presence of OOD samples, there are several different ways to find a hypothesis that generalizes well if we know which samples are OOD. For example, pre-training on all data and fine-tuning on the target task, e.g., doubly-robust estimation. We next plan to broaden our empirical study and explore the influence of multiple OOD tasks on the generalization error of the target task, provide a theoretical explanation for the non-monotonic trend in the generalization error of the target task, and develop efficient ways of determining the optimal weighing parameter α.

References

[1] V. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.

[2] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. Mit Press, 2008.

[3] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, *et al.*, "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models," *arXiv preprint arXiv:2206.04615*, 2022.

[4] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," *Advances in neural information processing systems*, vol. 30, 2017.

[5] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[6] G. R. Van Horn, *Towards a Visipedia: Combining Computer Vision and Communities of Experts*. PhD thesis, California Institute of Technology, 2019.

[7] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[8] R. Vedantam, D. Lopez-Paz, and D. J. Schwab, "An empirical investigation of domain generalization with empirical risk minimizers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28131–28143, 2021.

[9] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?," in *International Conference on Machine Learning*, pp. 872–881, 2019.

[10] C. Zhang, L. Zhang, and J. Ye, "Generalization bounds for domain adaptation," *Advances in neural information processing systems*, vol. 25, 2012.

[11] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," *Advances in neural information processing systems*, vol. 20, 2007.

[12] Y. Bu, G. Aminian, L. Toni, G. W. Wornell, and M. Rodrigues, "Characterizing and understanding the generalization error of transfer learning with gibbs algorithm," in *International Conference on Artificial Intelligence and Statistics*, pp. 8673–8699, PMLR, 2022.

[13] S. Hanneke and S. Kpotufe, "On the value of target data in transfer learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[14] I. Redko, A. Habrard, and M. Sebban, "Theoretical analysis of domain adaptation with optimal transport," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 737–753, Springer, 2017.

[15] B. Wang, J. Mendez, M. Cai, and E. Eaton, "Transfer learning via minimizing the performance gap between domains," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, 2006.

[17] F. Zenke, B. Poole, and S. Ganguli, "Continual Learning Through Synaptic Intelligence," in *International Conference on Machine Learning*, pp. 3987–3995, 2017.

A. Related Work

There is a large body of work that has used weighted-ERM based methods [7, 10–16]; this is either done to address domain shift or to address different distributions of tasks in a transfer or multi-task problem. The main idea is to obtain a single hypothesis h by minimizing an α-weighted combination of the empirical error of the target and source task $\alpha\hat{e}_t(h) + (1-\alpha)\hat{e}_s(h)$ with the goal of generalizing to new data from the target task. This body of work forms the primary motivation for our paper; in our case, the "source" task is the OOD samples.

It has been argued recently [8] that the generalization bounds of such approaches, building primarily upon [7] are loose. In this paper, we identify an unusual non-monotonic trend in the generalization error of the target task. First note that the calculations in [7] can be used directly for the case when we do not know the identity of the OOD samples by setting $\alpha = \frac{n}{n+m}$. One can get an insight into the non-monotonic trend of the target error from Theorem 3 in [7] because the second term $4\sqrt{\left(\frac{\alpha^2}{n} + \frac{(1-\alpha)^2}{m}\right)}\sqrt{V_H - \log\delta}$ decreases with increasing m while the third term $2(1-\alpha)d_H(P_t, P_o)$ increases because $\alpha \to 0$. While such a trend in the upper bound does not directly imply a non-monotonic trend in the error itself, this discussion suggests that the results from our experiments are not inconsistent with existing theory. There is a discrepancy here, e.g., we notice that [7]'s upper bound for naively weighted empirical error ($\alpha = 1/2$) does not have a non-monotonic trend (again, this is only an upper bound on the target error).

A more recent paper [12] presents an exact characterization of the target generalization error using conditional symmetrized Kullback-Leibler information between the output hypothesis and target samples given the source samples. While this work does not discuss the non-monotonic trends in target generalization error, it would be of interest to leverage this exact characterization to provide a theoretical explanation behind it.

B. Fisher's Linear Discriminant (FLD)

B.1. Synthetic Tasks

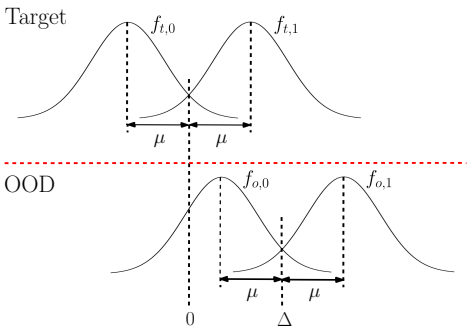


Figure 5. A schematic of the synthetic tasks

The target task P_t and the OOD task P_o are both binary classification problems with one-dimensional inputs. In both tasks, each class is sampled from a univariate Gaussian distribution. The OOD task is the target task translated by Δ. In summary, the target task has the class conditional densities,

$$f_{t,0} \overset{d}{=} \mathcal{N}(-\mu, \sigma^2)$$
$$f_{t,1} \overset{d}{=} \mathcal{N}(+\mu, \sigma^2),$$

while the OOD task distribution has the class conditional densities,

$$f_{o,0} \overset{d}{=} \mathcal{N}(\Delta - \mu, \sigma^2)$$
$$f_{o,1} \overset{d}{=} \mathcal{N}(\Delta + \mu, \sigma^2).$$

We also assume that both the target and OOD tasks have the same label distribution with equal class prior probabilities, i.e. $p(y_t = 1) = p(y_o = 1) = \pi = \frac{1}{2}$. Fig. 5 depicts the P_t and P_o pictorially.

B.2. Task-Agnostic Fisher's Linear Discriminant

In this section, we derive FLD when we have samples from a single task – which is also applicable to the task-agnostic setting. Consider a binary classification problem with $D_t = \{(x_i, y_i)\}_{i=1}^n \sim P_t$ where $x_i \in X \subseteq \mathbb{R}^d$ and $y_i \in Y = \{0, 1\}$.

Let f_k and π_k be the conditional density and prior probability of class k ($k \in \{0, 1\}$) respectively. The probability that x belongs to class k is

$$p(y = k \mid x) = \frac{\pi_k f_k(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)},$$

and the *maximum a posteriori* estimate of the class label is

$$h(x) = \underset{k \in \{0,1\}}{\text{argmax}}\ p(y = k \mid x) = \underset{k \in \{0,1\}}{\text{argmax}}\ \log(\pi_k f_k(x)). \tag{3}$$

Fisher's linear discriminant (FLD) assumes that each f_k is a multivariate Gaussian distribution with the same covariance matrix Σ, i.e,

$$f_k(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\bigg(-\frac{1}{2}(x - \mu_k)^\top \Sigma^{-1}(x - \mu_k)\bigg).$$

Under this assumption, the joint-density f of (x, y) becomes,

$$f(x, y) \propto \prod_{k=0}^{1} \left[\frac{\pi_k}{|\Sigma|^{1/2}} \exp\bigg(-\frac{1}{2}(x - \mu_k)^\top \Sigma^{-1}(x - \mu_k)\bigg)\right]^{\mathbf{1}[y=k]}$$

Therefore, the log-likelihood $l(\mu_0, \mu_1, \Sigma, \pi_0, \pi_1)$ over D_t is given by,

$$l(\mu_0, \mu_1, \Sigma, \pi_0, \pi_1) = \sum_{k=0}^{1} \sum_{(x,y) \in D_{t,k}} \left[\log \pi_k - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x - \mu_k)^\top \Sigma^{-1}(x - \mu_k)\right] + \text{const.}$$

where $D_{t,k}$ is the set of samples of D_t that belongs to class k. Based on the likelihood function above, we can obtain the maximum likelihood estimates $\hat{\mu}_k, \hat{\Sigma}, \hat{\pi}_k$. The expression for the estimate $\hat{\mu}_k$ is

$$\hat{\mu}_k = \frac{1}{|D_{t,k}|} \sum_{(x,y) \in D_{t,k}} x. \tag{4}$$

Plugging these estimates into (3), we get,

$$\begin{aligned}
\hat{h}(x) &= \underset{k \in \{0,1\}}{\text{argmax}} \left[\log \hat{\pi}_k - \frac{1}{2}\log|\hat{\Sigma}| - \frac{1}{2}(x - \hat{\mu}_k)^\top \hat{\Sigma}^{-1}(x - \hat{\mu}_k)\right] \\
&= \underset{k \in \{0,1\}}{\text{argmax}} \left[\log \hat{\pi}_k - \frac{1}{2}\log|\hat{\Sigma}| + x^\top \hat{\Sigma}^{-1}\hat{\mu}_k - \frac{1}{2}\hat{\mu}_k^\top \hat{\Sigma}^{-1}\mu_k\right]
\end{aligned}$$

Therefore, $\hat{h}(x) = 1$ iff,

$$\begin{aligned}
x^\top \hat{\Sigma}^{-1}\hat{\mu}_1 - \frac{1}{2}\hat{\mu}_1^\top \hat{\Sigma}^{-1}\mu_1 + \log \hat{\pi}_1 &> x^\top \hat{\Sigma}^{-1}\hat{\mu}_0 - \frac{1}{2}\hat{\mu}_0^\top \hat{\Sigma}^{-1}\mu_0 + \log \hat{\pi}_0 \\
x^\top \hat{\Sigma}^{-1}\hat{\mu}_1 - x^\top \hat{\Sigma}^{-1}\hat{\mu}_0 &> \frac{1}{2}\hat{\mu}_1^\top \hat{\Sigma}^{-1}\mu_1 - \frac{1}{2}\hat{\mu}_0^\top \hat{\Sigma}^{-1}\mu_0 + \log \hat{\pi}_0 - \log \hat{\pi}_1 \\
(\hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0))^\top x &> (\hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0))^\top \left(\frac{\hat{\mu}_0 + \hat{\mu}_1}{2}\right) + \log \frac{\hat{\pi}_0}{\hat{\pi}_1}
\end{aligned}$$

Hence the FLD decision rule $\hat{h}(x)$ is

$$\hat{h}(x) = \begin{cases} 1, & \omega^\top x > c \\ 0, & \text{otherwise} \end{cases}$$

where $\omega = \hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0)$ is a projection vector and $c = \omega^\top \left(\frac{\hat{\mu}_0 + \hat{\mu}_1}{2}\right) + \log \frac{\hat{\pi}_0}{\hat{\pi}_1}$ is a threshold. When $d = 1$ and $\pi_0 = \pi_1$, the decision rule reduces to

$$\hat{h}(x) = \begin{cases} 1, & x > \frac{\hat{\mu}_0 + \hat{\mu}_1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

B.3. Deriving the Generalization Error of the Target Task for Synthetic Tasks with FLD

We would like to derive an expression for the average generalization error of the target task, when we consider the synthetic tasks described in Appendix B.1. For simplicity, we set the variance σ^2 of the class conditional densities of the synthetic tasks to 1.

In the task-agnostic setting, the learning algorithm sees a single dataset $D = D_t \cup D_o$ of size $n+m$ which is a combination of both target and OOD samples. We can estimate μ_k using (4) to obtain

$$\begin{aligned}\hat{\mu}_k = \frac{1}{|D_k|}\sum_{(x,y)\in D_k} x &= \frac{\sum_{(x,y)\in D_{t,k}} x + \sum_{(x,y)\in D_{o,k}} x}{n_k + m_k} \\ &= \frac{n_k \bar{x}_{t,k} + m_k \bar{x}_{o,k}}{n_k + m_k} \\ &= \frac{n\bar{x}_{t,k} + m\bar{x}_{o,k}}{n+m}.\end{aligned} \tag{6}$$

where D_k is the set of samples of D that belongs to class k, $n_k = |D_{t,k}|$ and $m_k = |D_{o,k}|$ for $k \in \{0,1\}$. $\bar{x}_{t,k}$ and $\bar{x}_{o,k}$ denote the sample means of class k in target and OOD datasets respectively. We assume that $\pi = \frac{1}{2}$ from which it follows that $n_k = n\pi_k = \frac{n}{2}$ and $m_k = m\pi_k = \frac{m}{2}$. We cannot explicitly compute $\bar{x}_{t,k}$ and $\bar{x}_{o,k}$ in the task-agnostic setting, because we cannot separate target samples from OOD samples in D.

Since the samples are drawn from Gaussians, their averages also follow Gaussian distributions. Hence, the threshold $\hat{c} = \frac{\hat{\mu}_0 + \hat{\mu}_1}{2}$ of the hypothesis $\hat{h}$, estimated using FLD, is a random variable with a Gaussian distribution i.e., $\hat{c} \sim \mathcal{N}(\mu_h, \sigma_h^2)$ where

$$\begin{aligned}\mu_h &= \mathbb{E}[\hat{c}] = \frac{m\Delta}{n+m}, \\ \sigma_h^2 &= \text{Var}[\hat{c}] = \frac{1}{n+m}.\end{aligned}$$

The target error of a hypothesis $\hat{h}$ is

$$\begin{aligned}p(\hat{h}(x) \neq y \mid x, \hat{c}) &= \frac{1}{2}p_{x\sim f_{t,1}}[x < \hat{c}] + \frac{1}{2}p_{x\sim f_{t,0}}[x > \hat{c}] \\ &= \frac{1}{2} + \frac{1}{2}p_{x\sim f_{t,1}}[x < \hat{c}] - \frac{1}{2}p_{x\sim f_{t,0}}[x < \hat{c}] \\ &= \frac{1}{2}\big[1 + \Phi(\hat{c} - \mu) - \Phi(\hat{c} + \mu)\big]\end{aligned} \tag{7}$$

Using (7), the expected error on the target task $e_t(\hat{h}) = \mathbb{E}_{\hat{c}\sim\mathcal{N}(\mu_h,\sigma_h^2)}[p(\hat{h}(x) \neq y \mid x, \hat{c})]$ is given by,

$$\begin{aligned}e_t(\hat{h}) &= \int_{-\infty}^{\infty} \frac{1}{2}\big[1 + \Phi(\hat{c} - \mu) - \Phi(\hat{c} + \mu)\big]\frac{1}{\sigma_h}\phi\left(\frac{\hat{c} - \mu_h}{\sigma_h}\right)d\hat{c} \\ &= \int_{-\infty}^{\infty} \frac{1}{2}\big[1 + \Phi(y\sigma_h + \mu_h - \mu) - \Phi(y\sigma_h + \mu_h + \mu)\big]\phi(y)dy \\ &= \frac{1}{2}\left[\Phi\left(\frac{\mu_h - \mu}{\sqrt{1+\sigma_h^2}}\right) + \Phi\left(\frac{-\mu_h - \mu}{\sqrt{1+\sigma_h^2}}\right)\right]\end{aligned}$$

In the last equality, we make use of the identity $\int_{-\infty}^{\infty}\Phi(cx+d)\phi(x)dx = \Phi\big(\frac{d}{\sqrt{1+c^2}}\big)$ where ϕ and Φ are the PDF and CDF of the standard normal. Substituting the expressions for μ_h, σ_h^2 into the above equation, we get

$$e_t(\hat{h}) = \frac{1}{2}\left[\Phi\left(\frac{m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right) + \Phi\left(\frac{-m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right)\right] \tag{8}$$

For synthetic tasks with $\sigma^2 \neq 1$, the target generalization error can be obtained by simply replacing μ and Δ with $\frac{\mu}{\sigma}$ and $\frac{\Delta}{\sigma}$ respectively in (8).

B.4. Task-Aware Weighted Fisher's Linear Discriminant

We consider a target dataset $D_t = \{(x_i, y_i)\}_{i=1}^n$ and an OOD dataset $D_o = \{(x_i, y_i)\}_{i=1}^m$, which are samples from the synthetic tasks from Appendix B.1. This setting differs from Appendix B.3 since we know whether each sample from $D = D_t \cup D_o$ is OOD or not. This difference allows us to consider a log-likelihood function that weights the target and OOD samples differently, i.e. we consider

$$l(\mu_0, \mu_1, \sigma_0^2, \sigma_1^2) = \sum_{k=0}^{1} \left(\alpha \cdot \sum_{(x,y)\in D_{t,k}} \left[-\log \sigma_k - \frac{(x-\mu_k)^2}{2\sigma_k^2} \right] + (1-\alpha) \cdot \sum_{(x,y)\in D_{o,k}} \left[-\log \sigma_k - \frac{(x-\mu_k)^2}{2\sigma_k^2} \right] \right) + \text{const.} \quad (9)$$

α is a weight that controls the contribution of the OOD samples in the log-likelihood function. Under the above log-likelihood, the maximum likelihood estimate for μ_k is

$$\hat{\mu}_k = \frac{\alpha \sum_{(x,y)\in D_{t,k}} x + (1-\alpha) \sum_{(x,y)\in D_{o,k}} x}{\alpha |D_{t,k}| + (1-\alpha)|D_{o,k}|}. \quad (10)$$

We can make use of the above $\hat{\mu}_k$ to get a weighted FLD decision rule using (5).

B.5. Deriving the Generalization Error of the Target Task for Synthetic Tasks with Weighted FLD

We consider the synthetic tasks in Appendix B.1 with $\sigma^2 = 1$. We re-write $\hat{\mu}_k$ from (10) using notation from Appendix B.3:

$$\hat{\mu}_k = \frac{n\alpha \bar{x}_{t,k} + m(1-\alpha)\bar{x}_{o,k}}{n\alpha + m(1-\alpha)}.$$

We can explicitly compute $\bar{x}_{t,k}$ and $\bar{x}_{o,k}$ in the task-aware setting since we can separate target samples from OOD samples. For the synthetic dataset, the threshold $\hat{c}_\alpha = \frac{\hat{\mu}_0 + \hat{\mu}_1}{2}$ of the hypothesis $\hat{h}_\alpha$ follows a normal distribution $\mathcal{N}(\mu_{h\alpha}, \sigma_{h\alpha}^2)$ where

$$\mu_{h\alpha} = \mathbb{E}[\hat{c}_\alpha] = \frac{m(1-\alpha)\Delta}{n\alpha + m(1-\alpha)}$$

$$\sigma_{h\alpha}^2 = \text{Var}[\hat{c}_\alpha] = \frac{\alpha^2 n + (1-\alpha)^2 m}{(\alpha n + (1-\alpha)m)^2}$$

Similar to the Appendix B.3, we derive an analytical expression for the expected target risk of the weighted FLD, which is

$$e_t(\hat{h}_\alpha) = \frac{1}{2}\left[\Phi\left(\frac{\mu_{h\alpha} - \mu}{\sqrt{1+\sigma_{h\alpha}^2}} \right) + \Phi\left(\frac{-\mu_{h\alpha} - \mu}{\sqrt{1+\sigma_{h\alpha}^2}} \right) \right] \quad (11)$$

B.6. Additional Experiments using FLD

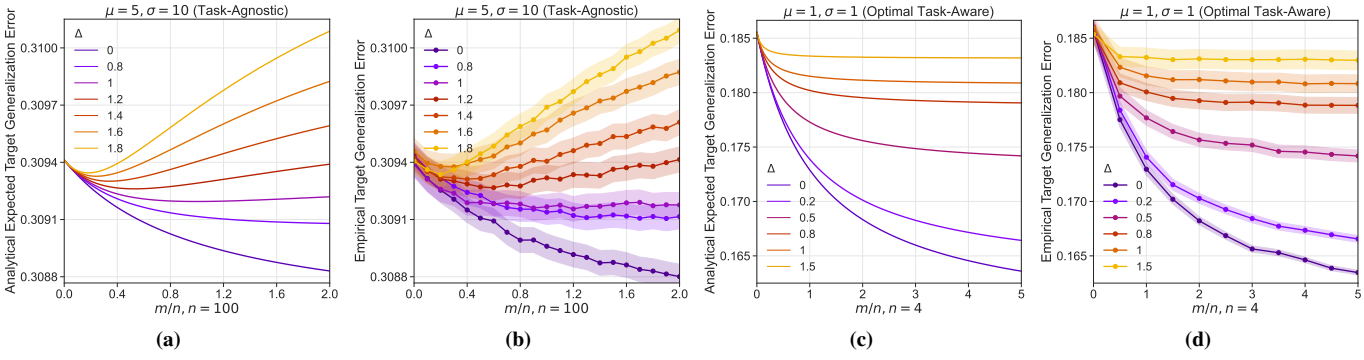


Figure 6. The FLD generalization error (Y-axis) on the target task is plotted against the ratio of OOD samples to target samples (X-axis). Figures (a) and (c) are plotted using the analytical expressions in (8) and (11) respectively while figures (b) and (d) are the corresponding plots from Monte-carlo simulations. The Monte-carlo simulations agree with the plots from the analytical expression, which validates its correctness. **(a) and (b):** The figure is identical to Fig. 1 and considers synthetic tasks with $n = 100$, $\mu = 5$ and $\sigma = 10$ in the task-agnostic setting. While a small number of OOD samples improves generalization on the target task, lots of samples increase the generalization error on the target task. **(c) and (d):** The figures consider synthetic tasks with $n = 4$, $\mu = 1$ and $\sigma = 1$ in the task-aware setting. If we consider the weighted FLD trained with optimal α^*, then the average generalization error monotonically decreases with more OOD samples.

C. Experiments with Neural Networks

C.1. Image Classification Dataset

Experiments with neural nets make use of tasks from Split-CIFAR10 [17] which are five binary classification tasks constructed by grouping consecutive labels of CIFAR-10. Each task has 10,000 training images and 2000 test images. The 5 tasks are airplane vs. automobile (T_1), bird vs. cat (T_2), deer vs. dog (T_3), frog vs. horse (T_4) and ship vs truck (T_5).

We consider two sets of tasks to study the impact of OOD data. The first set considers geometric intra-class nuisances which we study using a classification task as the target task and rotated versions of the same task as different OOD tasks. The second set studies category shifts and concept drifts using two different target and OOD classification problems.

The two sets of tasks are constructed as follows:

1. **T_2 as Target and Rotated T_2 as OOD:** We choose the bird vs. cat (T_2) task as the target task. We then rotate the images of T_2 by an angle θ° counter-clockwise around their centers to form a new task denoted by θ-T_2, which we consider as the OOD task. For each random seed, we randomly select a fixed sample of size $n = 100$ from the target task. Next, we select samples from the OOD task of varying sizes $m = \{0, 100, 200, 300, 400, 500, 1000, 2000\}$ such that each progressive sample is a subset of the next sample. The samples from both target and OOD tasks preserve the ratio of the classes. When selecting multiple sets of OOD samples, the OOD images that correspond to the 100 selected target images are disregarded. We perform the experimental routine for $\theta = \{0^\circ, 10^\circ, 45^\circ, 90^\circ, 135^\circ\}$.

2. **T_i as Target and T_j as OOD:** We choose a pair of distinct tasks and consider one as the target task and the other as the OOD task. First, we randomly select a fixed set of size $n = 100$ from the target task. Like in the previous set of tasks, we select samples from the OOD task of varying sizes $m = \{0, 100, 200, 300, 400, 500, 1000, 2000\}$ such that each progressive sample is a subset of the next sample. The samples from the target and OOD tasks preserve the ratio of the two classes. We perform experiments for all pairs of tasks (20 in total) in Split-CIFAR10.

C.2. Experimental Details

For both the task-agnostic and task-aware settings, at each m-value, we construct a combined dataset containing the n sized target set and m sized OOD set. We use a CNN for experiments in the task-agnostic and task-aware settings. We experiment with α fixed to 0.5 (naive task-aware model) and with the optimal α^*. We average the runs over 10 random seeds and evaluate on a test set of size 2000.

In the optimal task-aware setting, we use a grid-search to find the optimal α^* for each value of m. We use an adaptive equally-spaced α search set of size 10 such that it ranges from α^*_{prev} to 1.0 (excluding 1.0) where α^*_{prev} is the optimal value of α corresponding to the previous value of m. We use this search space since we expect α^* to be an increasing function of m.

C.3. Neural Architectures and Training

We use a small convolutional neural network (0.12M weights) with 3 convolution layers (kernel size 3 and 80 filters) interleaved with max-pooling, ReLU, batch-norm layers, with a fully-connected classifier layer in our experiments. The networks are trained using stochastic gradient descent (SGD) with Nesterov's momentum and cosine-annealed learning rate. The hyper-parameters used for the training are, learning rate of 0.01, mini-batch size of 128, and a weight-decay of 10^{-5}. All the images are normalized to have mean 0.5 and standard deviation 0.25. In the task-agnostic setting, we use sampling without replacement to construct the mini-batches. In the task-aware settings, we construct mini-batches with a fixed ratio of target and OOD samples. See Appendix C.4 and Fig. 7 for more details.

C.4. Construction of Mini-Batches

Consider a mini-batch $\{(x_{b_i}, y_{b_i})\}_{i=1}^{B}$ of size B. Let the randomly chosen mini-batch contains B_t target samples and B_o OOD samples ($B = B_t + B_o$). Let $\hat{e}_{B,t}(h)$ and $\hat{e}_{B,o}(h)$ denote the average mini-batch surrogate losses for the B_t target samples and B_o OOD samples respectively.

In the task-aware setting, $\hat{e}_{B,t}(h)$ and $\hat{e}_{B,o}(h)$ can be computed explicitly for each mini-batch resulting in the mini-batch gradient

$$\hat{\nabla}\hat{e}_B(h) = \alpha\hat{\nabla}\hat{e}_{B,t}(h) + (1-\alpha)\hat{\nabla}\hat{e}_{B,o}(h). \tag{12}$$

If we were to sample without replacement, we expect the fraction of the target samples in every mini-batch to approximately equal $\frac{n}{n+m}$ on average. However, if $m >> n$, we run into a couple of issues. First, we observe that most mini-batches have no

target samples, making it impossible to compute $\hat{\nabla}\hat{e}_{B,t}(h)$. Next, even if the mini-batch does have some target samples, there are very few of them, resulting in high variance in the estimate $\hat{\nabla}\hat{e}_{B,t}(h)$.

Hence, we find it beneficial to consider alternative sampling schemes for the mini-batch. Independent of the values of n and m, we use a sampler which ensures that every mini-batch has a fixed fraction of target samples, which we denote by β. For example if the mini-batch size B is 20 and if $\beta = 0.5$, then every mini-batch has 10 target samples and 10 OOD samples regardless of n and m. Note that this sampling biases the gradient, but results in reduced variance estimates. In practice, we observe improved test errors when we set β to either 0.5 or 0.75.

C.5. Comparing the Effect of Using Conventional and Custom Batches

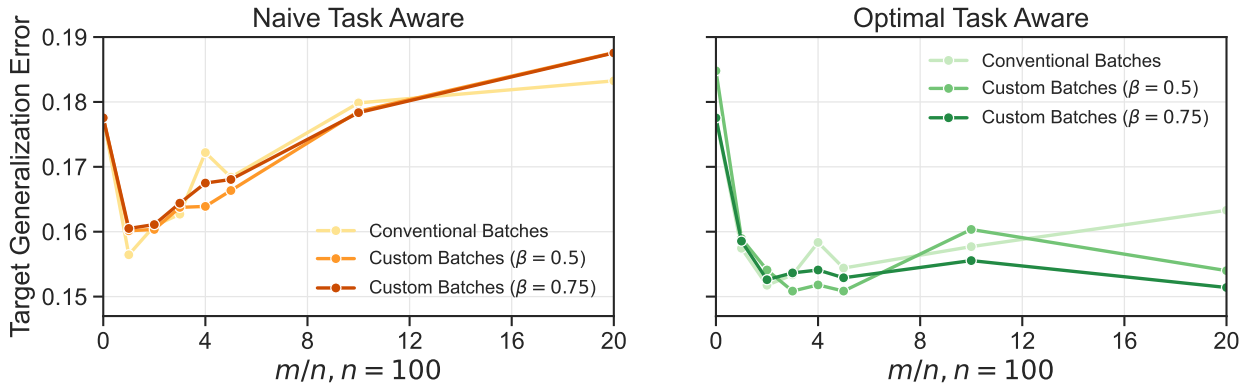
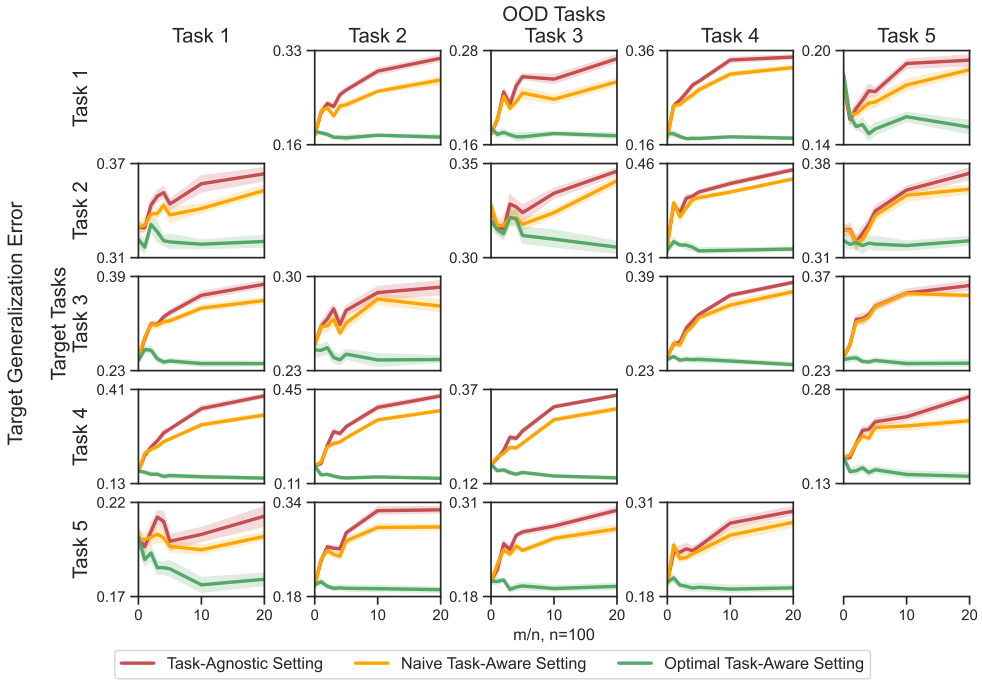
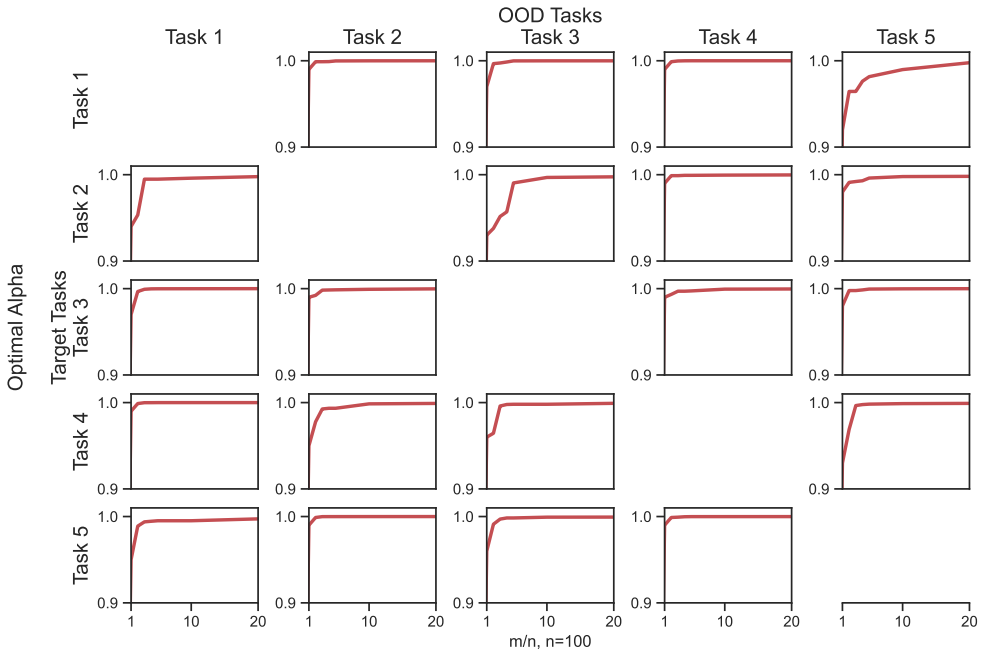


Figure 7. The test error of a neural network on the target task (Y-axis) is plotted against the number of samples from the OOD task (X-axis) for the target-OOD task pair of T_1 and T_5. One set of curves (lightest shade of green and yellow) considers mini-batches which are constructed using sampling without replacement; This is the conventional strategy used in supervised learning. The other curves consider $\beta = 0.5$ (intermediate shades of orange and green) and $\beta = 0.75$ (darkest shade of red and green). All plots are in the task-aware setting. **Left:** If we consider $\alpha = 0.5$, then the choice of β has little effect on the generalization error. **Right:** However, if we use the α^* to weight the OOD and target losses, then the generalization error depends on the the choice of β with $\beta = 0.75$ having the lowest test error.

C.6. Target Generalization Error Curves for all the SplitCIFAR-10 Task Pairs



(a)



(b)

Figure 8. (a) We plot the test error on the target task (Y-axis) against the ratio of number of samples from the OOD task to the number of samples on the target task (X-axis), for all target-OOD task pairs from Split-CIFAR10. A neural net trained with a loss weighted by α^* is able to leverage OOD data to improve the networks ability to generalize on the target task. **(b)** The optimal α^* (Y-axis) is plotted against the number of OOD samples (X-axis) for the optimally weighted task-aware setting. As we increase the number of OOD samples, we see that α^* increases. This allows us to balance the variance from few target samples and the bias from using OOD samples from a different disitribution.