RiskyZoo: A Library for Risk-Sensitive Supervised Learning

William Wong [1] **Audrey Huang** [2] **Liu Leqi** [1] **Kamyar Azizzadenesheli** [3] **Zachary C. Lipton** [1]

Abstract

Supervised learning models are increasingly used in algorithmic decision-making. The traditional assumption on the training and testing data being independently and identically distributed is often violated in practical learning settings, due to distribution shifts. To mitigate the effects of such nonstationarities, risk-sensitive learning is proposed to train models under different (risk) functionals beyond the expected loss. For example, learning under the conditional value-at-risk of the losses is equivalent to training a model under a particular type of worst-case distribution shift. While many risk functionals and learning procedures have been proposed, their implementations are either nonexistent or in individualized repositories. With no common implementations and baseline test beds, it is difficult to decide which risk functionals and learning procedures to use. To address this, we introduce a library (RiskyZoo) for risk-sensitive supervised learning. The library contains implementations of risk-sensitive learning objectives and optimization procedures that can be used as add-ons to the PyTorch library. We also provide datasets to compare these learning methods. We demonstrate usage of our library through comparing models learned under different risk objectives, optimization performances of different methods for a single objective, and risk assessments of pretrained ImageNet models.

1. Introduction

Predictions made by supervised learning models assist responsible decision making in a variety of domains, including healthcare (Patel et al., 2019; Rajpurkar et al., 2020; Tschandl et al., 2020; Bien et al., 2018), credit lending (Bussmann et al., 2021; Kruppa et al., 2013), and employment (Raghavan et al., 2020; Hoffman et al., 2017). In credit lending, for example, decisions to grant loans take into account the predicted risk of the applicant defaulting.

Traditionally, these supervised learning models are trained to minimize the average loss, which may result in undesirable model behavior in two ways. First, training methods may fail to account for real-world dynamics, such as distribution shift between training data and the deployment environment. Consequently, the model that minimizes the training average loss may differ from the one that minimizes the test average loss. Second, the average loss may be poorly aligned with real-world desiderata, such as robustness or risk aversion. For example, in finance, international banking law requires financial institutions to evaluate their models (BCBS, 2013) in terms of the conditional value-at-risk (CVaR), a risk-averse functional.

Recognizing that optimization of the average loss fails to address these real-world dynamics and desiderata, a plethora of risk-sensitive supervised learning works instead optimize other *risk functionals* of the loss (Duchi & Namkoong, 2018; Duchi et al., 2020; Leqi et al., 2019; Li et al., 2020; Khim et al., 2020; Lee et al., 2020). These functionals produce models that align with the risk preferences of the decision maker and possess responsible decision making properties such as robustness to distribution shifts and noisy labels.

Existing risk-sensitive learning literature has proposed optimization methods for a diversity of risk functionals. However, these methods are implemented in individualized repositories under different structures, often for a single risk functional and application at a time. As a result, it is difficult for researchers to compare the effects of optimizing different risks, or to compare different optimization methods for the same risk, on problems of interest. This hinders studies on providing guidelines for choosing the desirable risk functional under different setups.

In light of this, we introduce a library for risk-sensitive supervised learning (RiskyZoo[1]) that contains implementations of a comprehensive set of risk functionals, optimiza-

[1]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA., USA [2]Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL., USA [3]Department of Computer Science, Purdue University, West Lafayette, IN., USA. Correspondence to: William Wong <wwong3@andrew.cmu.edu>, Liu Leqi <leqil@cs.cmu.edu>.

ICML 2022 Workshop on Responsible Decision Making in Dynamic Environments, Baltimore, Maryland, USA, 2022.

[1]Library code is sourced at github.com/w07wong/RiskyZoo.

tion procedures, and datasets. The library design enables one to easily adjust a traditional supervised learning training pipeline in PyTorch to perform risk-sensitive learning. To demonstrate the usage of our library, we conduct a study on comparing models learned through different risk functionals under various distribution shifts. These experiments demonstrate that even if the learning objective is to minimize the expected test loss, the choice of risk functional (during training time) depends on the learning setup (e.g., the type of distribution shifts between training and testing data).

2. Risk-Sensitive Supervised Learning

Risk-sensitive supervised learning, in contrast to traditional supervised learning, is built to minimize not just the expected loss but other functionals of the loss.

Notation Given training data $\{X_i, Y_i\}_{i=1}^n$ independently and identically drawn from a training data distribution $\mathbb{P}_{\text{train}}(X, Y)$ where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, a risk functional ρ that maps a random variable to a real value, and a hypothesis class $\mathcal{F}$, we aim to find

$$f^\star(\rho, \mathbb{P}_{\text{train}}) \in \min_{f \in \mathcal{F}} \rho(\ell_f(X, Y)),$$

where $\ell_f(X, Y) := \ell(f(X), Y)$ is the loss random variable under model f. We use $F_f(\ell_f(X, Y))$ to denote the CDF of $\ell_f(X, Y)$. In the traditional supervised learning setting, the risk functional is the expected value, i.e., $\rho = \mathbb{E}$. We use $f^\star(\rho, \mathbb{P}_{\text{train}})$ to emphasize that the model is learned under functional ρ and data from $\mathbb{P}_{\text{train}}$. When context is clear, we may omit the arguments and use $f^\star$. At test time, the data are sampled from $\mathbb{P}_{\text{test}}$, which may deviate from $\mathbb{P}_{\text{train}}$.

2.1. Risk Functionals

Below we describe common risk functionals utilized in machine learning literature and applications, and summarize their related works in Table 2 (Appendix A).

Expected Value $\rho_{\mathbb{E}}(\ell_f(X, Y)) = \mathbb{E}[\ell_f(X, Y)]$ is the learning objective in traditional supervised learning.

CVaR $\rho_{\text{CVaR},\alpha}(\ell_f(X, Y)) = \mathbb{E}[\ell_f(X, Y) | \ell_f(X, Y) \geq \text{VaR}_\alpha(\ell_f(X, Y))]$ where $\alpha \in [0, 1]$ and VaR_α is the $100\times\alpha$-percentile of the losses. CVaR quantifies the expected value of losses that exceed VaR_α.

Entropic Risk $\rho_{\text{Ent, t}}(\ell_f(X, Y)) = \frac{1}{t} \log \mathbb{E}[e^{t\ell_f(X,Y)}]$. As $t \to \infty$, the entropic risk focuses on minimizing tail losses. As $t \to -\infty$, the entropic risk ignores outliers in the data.

Human-aligned Risk $\rho_{\text{H, a,b}}(\ell_f(X, Y)) = \mathbb{E}[\ell_f(X, Y) w(F_f(\ell_f(X, Y)))]$ where $w(t) = \frac{3-3b}{a^2-a+1}(3t^2 - 2(a + 1)t + a) + 1$ is a weighting function inspired by cumulative prospective theory that overweights extreme losses.

Inverted CVaR $\rho_{\overline{\text{CVaR}},\alpha}(\ell_f(X, Y)) = \mathbb{E}[\ell_f(X, Y) | \ell_f(X, Y) \leq \text{VaR}_\alpha(\ell_f(X, Y))]$, where $\alpha \in [0, 1]$. Inverted CVaR quantifies the expected value of losses below VaR_α.

Mean-Variance $\rho_{\text{MV, c}}(\ell_f(X, Y)) = \mathbb{E}[\ell_f(X, Y)] + c \cdot \text{Variance}[\ell_f(X, Y)]$.

Trimmed Risk $\rho_{\text{Trim},c}(\ell_f(X, Y)) = \mathbb{E}[\ell_f(X, Y) | F_f(\ell_f(X, Y)) \in [\alpha, 1 - \alpha]]$, where $\alpha \in [0, 0.5)$. Trimmed risk ignores extreme (high and low) losses.

2.2. Algorithmic Properties

Distribution shifts, i.e. $\mathbb{P}_{\text{train}}(X, Y) \neq \mathbb{P}_{\text{test}}(X, Y)$, occur in many real-world settings. Below, we summarize existing literatures that use risk-sensitive learning to deal with different types of distribution shifts. In other words, these risk-sensitive models $f^\star(\rho, \mathbb{P}_{\text{train}})$ are close to $f^\star(\mathbb{E}, \mathbb{P}_{\text{test}})$ (which we call algorithmic properties).

General distribution shift: $\mathbb{P}_{\text{train}}(X, Y) \neq \mathbb{P}_{\text{test}}(X, Y)$. Duchi & Namkoong (2018) provides a distributionally robust optimization (DRO) method for dealing with generic distribution shifts and connects $f^\star(\rho_{\text{CVaR}}, \mathbb{P}_{\text{train}})$ with the model under a particular type of worst case shift.

Covariate Shift: $\mathbb{P}_{\text{train}}(X) \neq \mathbb{P}_{\text{test}}(X)$ but $\mathbb{P}_{\text{train}}(Y|X) = \mathbb{P}_{\text{test}}(Y|X)$. In Fan et al. (2017), $\rho_{\overline{\text{CVaR}}}$ is proposed to deal with class imbalance at training time but balanced classes at test time.

Label Shift: $\mathbb{P}_{\text{train}}(Y) \neq \mathbb{P}_{\text{test}}(Y)$ while $\mathbb{P}_{\text{train}}(X|Y) = \mathbb{P}_{\text{test}}(X|Y)$. Garg et al. (2021) proposes to use inverted CVaR as the learning objective and show that under certain settings $f^\star(\rho_{\overline{\text{CVaR}}}, \mathbb{P}_{\text{train}}) = f^\star(\mathbb{E}, \mathbb{P}_{\text{test}})$. We note that a particular type of label shift is **noisy labels** where $\mathbb{P}_{\text{train}}(Y)$ is a mixture of $\mathbb{P}_{\text{test}}(Y)$ and the uniform distribution.

In addition to the aforementioned distribution shifts, risk sensitive learning has also been used in treating **outliers and heavy tails** in the training distribution, e.g., using trimmed risk (Tukey & McLaughlin, 1963).

3. RiskyZoo

Our library (RiskyZoo) provides a general framework for learning and assessing models under different risk functionals. RiskyZoo is built upon PyTorch and consists of three modules: (i) Risk Functionals, (ii) Optimizers, and (iii) Datasets (Figure 1). The modular design of RiskyZoo allows it to be easily integrated into existing PyTorch pipelines for tasks such as image classification, risk prediction, etc.

Module: Risk Functionals Risk functionals can serve as both the training objectives for learning a model and evalaution metrics for assessing the performance of a model. RiskyZoo implements all risk functionals mentioned in Sec-

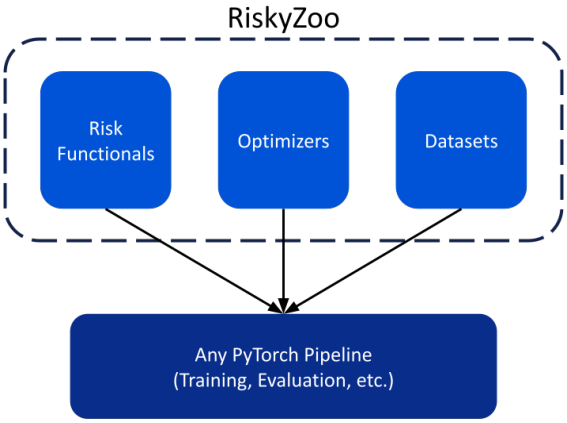


Figure 1. RiskyZoo consists of three modules that can be plugged in individually into existing PyTorch pipelines.

tion 2.1. Each risk functional takes in an array of losses for the model and outputs the risks for the model by applying the risk function on the empirical CDF of the losses (Leqi et al., 2022). In cases for expected value, the output is the same as the average loss. Users can specify customizable parameters for the functional. For example, for CVaR, the user can specify its parameter α.

Module: Optimizers The library's standard optimization procedures are first-order methods (Leqi et al., 2022). That is, one may use standard first-order optimizers provided in PyTorch to minimize the risks since our implementations of risk functionals are differentiable. In contrast to other risks, CVaR has many proposed specialized optimization methods: TruncCVaR (Curi et al., 2020), Soft-CVaR (Curi et al., 2020), an adaptive sampling method (Curi et al., 2020), and two distributionally robust optimization methods (Duchi & Namkoong, 2018; Duchi et al., 2020). Further details and a comparison of CVaR optimizers are detailed in Appendix C. RiskyZoo contains implementations of all aforementioned CVaR optimization methods. These optimization methods can be used in existing PyTorch training pipelines with minimal modifications. For example, users may replace optimizers such as SGD or Adam with one of the CVaR optimizers. We note that the usage of these optimizers is similar to a PyTorch optimizer, since these optimizers have a function (zero_grad) to zero out the gradients and a function (step) to take an optimization step.

Module: Datasets To compare the behavior of models learned under different risk functionals and understand the trade-offs of the risk functionals, we create standardized datasets for risk-sensitive learning. Existing risk-sensitive learning works often test the proposed methods on different datasets, making it hard to interpret the results. For better reproducibility, the dataset module is designed to test (and compare) the algorithmic properties of different risk functionals. Specifically, it contains five datasets for settings including covariate shift, label shift (noisy labels), and outliers. We plan to add more large-scale datasets for these settings in the future. Details of the datasets and demonstrations of usage are presented in Section 4.1.

4. RiskyZoo In Action

In the following, we illustrate the usage of RiskyZoo through two angles: (i) Risk assessment: assessing learned models through different risk functionals; and (ii) Risk-sensitive learning: training risk-sensitive models and comparing them.

Risk assessment is the task of evaluating learned models through multiple perspectives, including inspecting the train and test loss distributions of a given model (Appendix B) and assessing the train and test risks in terms of different risk functionals. Each risk captures some aspects of the model. CVaR, by focusing on the losses that exceed some threshold, quantifies the upper tail performance of a model. Inverted CVaR on the other hand, exposes how well a model performs on its best data points. Trimmed risk gives a more robust estimate of the mean performance. Human-aligned risk is the opposite of trimmed risk, focusing more on both tail ends of performance. Mean-variance includes the variance of losses. Entropic risk assesses a model's performance against outliers. We perform risk assessments of all learned models and pretrained ImageNet models below.

For risk-sensitive learning, we use datasets provided in RiskyZoo to compare the algorithmic properties of different risk functionals, and train risk-sensitive CIFAR-10 models under noisy labels.

4.1. Risk-Sensitive Learning on the Dataset Module

We provide usage examples of RiskyZoo's Datasets module (Section 3) for comparing the algorithmic properties of

Model	Accuracy	$\rho_{\mathbb{E}}$	$\rho_{\text{CVaR},\alpha=0.05}$	$\rho_{\text{Ent},t=-1}$	$\rho_{\text{H},a=0.4,b=0.3}$	$\rho_{\overline{\text{CVaR}},\alpha=0.95}$	$\rho_{\text{MV},c=1}$	$\rho_{\text{Trim},\alpha=0.05}$
GoogLeNet	0.70	1.28	1.35	0.60	**2.04**	1.00	**4.38**	1.06
Inception	0.70	1.83	1.92	**0.39**	3.51	1.22	14.42	1.29
ResNet-18	0.70	1.25	**1.31**	0.50	2.15	**0.91**	5.35	**0.96**
ShuffleNet	0.69	1.36	1.43	0.46	2.42	0.97	6.72	1.03
VGG-11	0.69	**1.26**	1.33	0.52	2.13	0.93	5.21	0.98

Table 1. **Risk Assessment of ImageNet Models.** Validation accuracy and the risk-sensitive performance of each model.

different risk functionals. These experiments showcase that even if one aims to obtain a model that performs well in expectation at test time, depending on the setups, different risk functionals should be chosen as the learning objective at training time. Full experimental details and results can be found in Appendix B.

Noisy Labels To illustrate robustness to noisy labels, we explore two settings: when the training labels are corrupted, and when both the training and test labels are corrupted. We apply noise by randomly sampling $20\%, 30\%, 40\%$, and 80% of the data and flipping labels. Full details and results can be found in Appendices B.1 and B.2. As seen in Table 5, when only the training labels are corrupted, trimmed risk and entropic risk achieve the lowest average test loss.

Covariate Shift We explore a covariate shift setting described in Appendix B.3. Visualized in Figure 11, the dataset contains one majority class (green) with 2000 points and a minority class (blue) with 400 points. We represent a covariate shift with the training set containing more minority class data points but the test set following the original data distribution. Summarized in Table 10, CVaR, compared to expected loss, achieves $3\times$ greater test accuracy, $2\times$ smaller average test loss, and $2.75\times$ lower test CVaR.

Minority Group Performance Under nonuniform distributions, we still wish to achieve similar performance for all subpopulations. We follow the setup from Duchi & Namkoong (2018); Leqi et al. (2019). In addition to risk assessment, we quantify model performance under majority and minority risk achieved. Full details are found in Appendix B.4. Shown in Table 11, CVaR achieves the lowest minority risk by focusing on minimizing the loss for the worst $\alpha = 0.1$ percentile of losses. However, this leads to the greatest majority risk. Entropic risk, human-aligned risk, and mean-variance also achieve lower minority risks than expected loss, but achieve lower majority risks than CVaR.

Label Shift We follow the same setup as the Minority Group Performance dataset but apply a label shift to the test data. Label shift details and results are summarized in Appendix B.5. We find that inverted CVaR, entropic risk, and trimmed risk outperform expected loss on all test metrics except mean-variance (Table 13).

4.2. Large Scale Examples

We demonstrate usage of our library on larger tasks.

ImageNet: Risk Assessment Model performance is frequently quantified using accuracy or mean squared error. However, models which achieve similar accuracies may perform differently under risk notions. To demonstrate this, we conduct risk assessments of pretrained PyTorch ImageNet classifiers on the ImageNet validation set (Russakovsky et al., 2015). We select models with similar validation accuracies: GoogLeNet (Szegedy et al., 2015), Inception (Szegedy et al., 2016), ResNet-18 (He et al., 2016), ShuffleNet (Ma et al., 2018), and VGG-11 (Simonyan & Zisserman, 2014). Results are summarized in Table 1. Despite similar accuracies, tail performances of the loss distributions of the models differ. For example, Inception has over $2\times$ higher mean-variance and $1.3\times$ greater CVaR than other models. In situations where worst-case performance and predictability is important, Inception may not be as well suited as other models.

CIFAR-10: Learning under Noisy Labels We replicate the robust classification setup in Jiang et al. (2018); Li et al. (2020). We take the CIFAR-10 dataset and uniformly at random corrupt 80% of the training labels. The test set remains clean. The performance of the VGG-11 models learned under different risk functionals are are summarized in Appendix D. As a sanity check, in our experiment, among all learned models, the one trained with a particular risk functional minimizes that objective under the training data (Table 15). At test time (when there is clean data), the model learned under entropic risk produces 5% greater accuracy than the model trained with the expected loss (Table 16).

5. Conclusion and Future Work

We present Risky Zoo, a risk-sensitive supervised learning library which provides implementations of a comprehensive list of risk functionals, their optimization procedures, and datasets for inspecting their algorithmic properties. For future work, we will explore real-world applications, such as finance, which require risk-averse or risk-seeking models. We also plan to extend the library to support risk-sensitive reinforcement learning.

References

BCBS. Fundamental review of the trading book: A revised market risk framework. *Consultative Document, October*, 2013.

Bien, N., Rajpurkar, P., Ball, R., Irvin, J., Park, A., Jones, E., Bereket, M., Patel, B., Yeom, K., Shpanskaya, K., Halabi, S., Zucker, E., Fanton, G., Amanatullah, D., Beaulieu, C., Riley, G., Stewart, R., Blankenberg, F., Larson, D., Jones, R., Langlotz, C., Ng, A., and Lungren, M. Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of mrnet. *PLoS Medicine*, 15(11), November 2018. ISSN 1549-1277. doi: 10.1371/journal.pmed.1002699. Publisher Copyright: © 2018 Bien et al. http://creativecommons.org/licenses/by/4.0/.

Bussmann, N., Giudici, P., Marinelli, D., and Papenbrock, J. Explainable machine learning in credit risk management. *Computational Economics*, 57, 01 2021. doi: 10.1007/s10614-020-10042-0.

Curi, S., Levy, K. Y., Jegelka, S., and Krause, A. Adaptive sampling for stochastic risk-averse learning. *Advances in Neural Information Processing Systems*, 33:1036–1047, 2020.

Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

Duchi, J., Hashimoto, T., and Namkoong, H. Distributionally robust losses for latent covariate mixtures. *arXiv preprint arXiv:2007.13982*, 2020.

Fan, Y., Lyu, S., Ying, Y., and Hu, B. Learning with average top-k loss. *Advances in neural information processing systems*, 30, 2017.

Garg, S., Wu, Y., Smola, A. J., Balakrishnan, S., and Lipton, Z. Mixture proportion estimation and pu learning: A modern approach. *Advances in Neural Information Processing Systems*, 34, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hoffman, M., Kahn, L. B., and Li, D. Discretion in hiring. *The Quarterly Journal of Economics*, 133(2):765–800, 2017.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

Khim, J., Leqi, L., Prasad, A., and Ravikumar, P. Uniform convergence of rank-weighted learning. In *International Conference on Machine Learning*, pp. 5254–5263. PMLR, 2020.

Kruppa, J., Schwarz, A., Arminger, G., and Ziegler, A. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 40(13):5125–5131, 2013. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2013.03.019.

Lee, J., Park, S., and Shin, J. Learning bounds for risk-sensitive learning. *Advances in Neural Information Processing Systems*, 33:13867–13879, 2020.

Leqi, L., Prasad, A., and Ravikumar, P. K. On human-aligned risk minimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Leqi, L., Huang, A., Lipton, Z. C., and Azizzadenesheli, K. Supervised learning with general risk functionals. In *To appear in International Conference on Machine Learning*, 2022.

Li, T., Beirami, A., Sanjabi, M., and Smith, V. Tilted empirical risk minimization. *arXiv preprint arXiv:2007.01162*, 2020.

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Nemirovski, A. and Shapiro, A. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

Patel, B., Rosenberg, L., Willcox, G., Baltaxe, D., Lyons, M., Irvin, J., Rajpurkar, P., Amrhein, T., Gupta, R., Halabi, S., Langlotz, C., Lo, E., Mammarappallil, J., Mariano, A., Riley, G., Seekins, J., Shen, L., Zucker, E., and Lungren, M. Human–machine partnership with artificial intelligence for chest radiograph diagnosis. *NPJ Digital Medicine*, 2(1), December 2019. ISSN 2398-6352. doi: 10.1038/s41746-019-0189-7. Publisher Copyright: © 2019, The Author(s).

Raghavan, M., Barocas, S., Kleinberg, J., and Levy, K. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 469–481, 2020.

Rajpurkar, P., O'Connell, C., Schechter, A., Asnani, N., Li, J., Kiani, A., Ball, R., Mendelson, M., Maartens, G., Van Hoving, D., Griesel, R., Ng, A., Boyles, T., and Lungren, M. CheXaid: deep learning assistance for physician diagnosis of tuberculosis using chest x-rays in patients

with HIV. *NPJ Digital Medicine*, 3:115, 09 2020. doi: 10.1038/s41746-020-00322-2.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Tschandl, P., Codella, N., Halpern, A., Puig, S., Apalla, Z., Rinner, C., Soyer, P., Rosendahl, C., Malvehy, J., Zalaudek, I., Argenziano, G., Longo, C., and Kittler, H. Human–computer collaboration for skin cancer recognition. *Nature Medicine*, 26, 08 2020. doi: 10.1038/s41591-020-0942-0.

Tukey, J. W. and McLaughlin, D. H. Less vulnerable confidence and significance procedures for location based on a single sample: Trimming/winsorization 1. *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 331–352, 1963.

A. Discussion on Risk Functionals

Below is a table that summarizes existing supervised learning works that cover these risk functionals.

Objective	Related Works
CVaR	Duchi & Namkoong (2018); Duchi et al. (2020); Khim et al. (2020); Lee et al. (2020)
Entropic Risk	Li et al. (2020); Lee et al. (2020)
Human-Aligned Risk	Leqi et al. (2019)
Inverted CVaR	Lee et al. (2020); Garg et al. (2021)
Mean-Variance	Lee et al. (2020)
Trimmed Risk	Tukey & McLaughlin (1963); Khim et al. (2020)

Table 2. Risk-sensitive objectives and their related works.

B. Risk-Sensitive Learning on the Dataset Module

We compare the performance of the risk-sensitive objectives described in Section 2.1 against expected loss. In the sections below, we describe the datasets and present experimental results on the datasets contained in the RiskyZoo Dataset module. We train logistic regression classifiers using full batch gradient descent with learning rate 1e-2 and no momentum. Models are trained till convergence which we define as when the current risk is within the average of the past 5 risks by 1e-4. We use cross entropy loss for classification and mean squared error for regression. For classification, if the number of training iterations exceed 3000, we stop training. For regression, we set the iteration limit to 5000. Training and testing use a 70:30 dataset train-test split unless stated otherwise.

B.1. Classification: Noisy Labels, Label Shift

We replicate a scenario where training data is corrupted from Jiang et al. (2018); Li et al. (2020). We simulate a label shift with the test labels remaining clean. There are two classes of data drawn from 2D Gaussian distributions $\mathcal{N}(0, 0.16)$ and $\mathcal{N}(1, 0.16)$. We draw 2000 training data points and 2000 test data points. Given some overall noise level $\mathcal{K}$, we distribute the noise non-uniformly: one class has $0.7 * \mathcal{K}$ noise and the other has $0.3 * \mathcal{K}$ noise. Noise is applied by flipping the labels for each class. We evaluate the accuracy and performance under each risk functional for $\mathcal{K} = 0.2, 0.3, 0.4$, and 0.8. Visualizations of the data are shown in Figure 2. For CVaR, inverted CVaR, and trimmed risk, we choose α values to study the effects of optimizing using the worst α percentile of losses, best α percentile of losses, and discarding both the highest and lowest α percentile of losses respectively. Train and test accuracies of models learned under each risk-sensitive objective are shown in Table 3.

No Noise	20% Noise	30% Noise	40% Noise	80% Noise

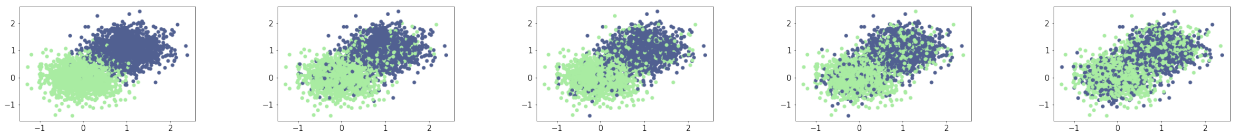


Figure 2. Noisy labels classification datasets with varying level of noise $\mathcal{K}$.

Aside from CVaR, all other objectives achieve similar performance across all noise levels. Data points whose labels are flipped lie on the wrong side of the decision boundary. As a result, they can incur high losses. CVaR, by focusing on high loss data points is uncertain where the decision boundary is and produces prediction probabilities close to uniform. Decision boundaries are illustrated in Figure 3.

Risk assessment of models are summarized in Table 4 and Table 5 respectively. Models optimizing for a risk functional during training minimizes that objective during training time. Under the test data, the better a model's decision boundary, the better it performs under all risk notions.

The train and test loss distributions are illustrated in Figures 4 and 5. Although CVaR has a shorter tail than all other risk functionals, its losses are concentrated around a higher value. Inverted CVaR and trimmed risk exhibit longer, albeit short, tails due to the objectives ignoring samples with high losses.

Finally, we compare the average number of training epochs each risk functional takes till convergence in Figure 6. Mean-variance requires the least number of epochs on average across all levels of noise. Trimmed risk and inverted CVaR require more epochs but achieve better performance.

Objective	Parameters	20% Noise		30% Noise		40% Noise		80% Noise	
		Train	Test	Train	Test	Train	Test	Train	Test
Expected Loss	-	0.87	0.95	0.83	0.95	0.78	0.94	0.64	0.78
CVaR	$\alpha = 0.3$	0.83	0.91	0.61	0.62	0.56	0.59	0.50	0.56
Entropic Risk	$t = -0.5$	0.87	0.95	0.83	0.95	0.79	0.95	0.64	0.78
Human-Aligned Risk	$a = 0.4, b = 0.8$	0.87	0.95	0.82	0.95	0.78	0.93	0.63	0.76
Inverted CVaR	$\alpha = 0.7$	0.83	0.91	0.80	0.92	0.77	0.94	0.63	0.77
Mean-Variance	$c = 1$	0.86	0.95	0.82	0.94	0.78	0.93	0.64	0.78
Trimmed Risk	$\alpha = 0.3$	0.85	0.94	0.82	0.95	0.78	0.95	0.64	0.77

Table 3. **Classification: Noisy Labels, Label Shift.** Train and test accuracies for varying levels of noise when only the training data is corrupted with noisy labels. CVaR, by focusing on high loss data points around the decision boundary, suffers from a uniform prediction model and lower accuracies.

B.2. Classification: Noisy Labels, No Label Shift

This study replicates the Classification: Noisy Labels, Label Shift except both training and test labels are corrupted. The same amount of noise is applied to the training and test datasets.

Train and test accuracies of models are listed in Table 6. Decision boundaries are illustrated in Figure 7. Train and test performances under each risk functional are listed in Tables 7 and 8. Loss distributions are illustrated in Figures 8 and 9. Optimizing under an objective produces a model that performs well on that objective under the training data. However, test performance is tied to the accuracy of the learned decision boundary. In this case, the tighter the boundary, the better performance a model achieves under all the risk functionals. Trimmed risk and inverted CVaR produce the most confident decision boundaries while the CVaR model has close to uniform class probabilities for the data points. As a result, trimmed risk and inverted CVaR achieve better risk-sensitive metrics compared to CVaR. The average number of training epochs till convergence is shown in Figure 10. Similar to the Classification: Noisy Labels, Label Shift example, trimmed risk and inverted CVaR require more epochs for convergence.

B.3. Classification: Covariate Shift

We explore covariate shift by drawing data from an XOR distribution created by four Gaussians with standard deviations of 0.4. Two classes are created: a majority class (green) containing 2000 data points and a minority class (blue) containing 400 samples (Figure 11). We randomly sub-sample the majority class in the training set to obtain a new dataset with a 10%/90% class imbalance with the majority and minority classes swapped. The test set distribution remains unchanged. While a model minimizing expected loss can achieve low risk on the training set, it performs poorly on the training minority class. Consequently at test time, the model will perform poorly on the original majority class.

Results on the training set are summarized in Table 9. The model trained with expected loss achieves low risk, but has a CVaR over $2\times$ greater than the models minimizing CVaR, entropic risk, and mean-variance. Results on the test set are summarized in Table 10. CVaR achieves the highest accuracy and lowest average loss. For large values of c, mean-variance performs similarly. By heavily penalizing variance, the mean-variance model may be forced to achieve relatively uniform performance on both the majority and minority classes. Decision boundaries, train loss distributions, and test loss distributions are shown in Figure 13. The average number of training epochs is summarized in Figure 12.

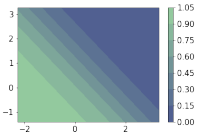
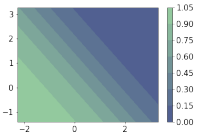
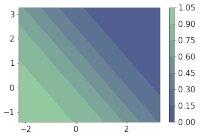
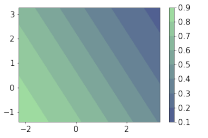
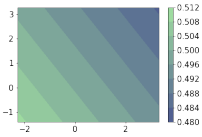
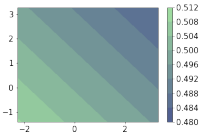
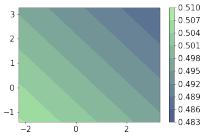
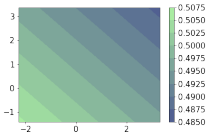
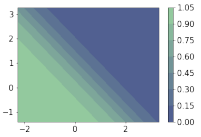
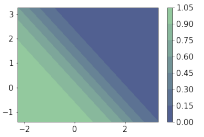
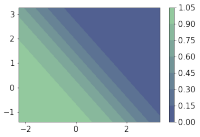
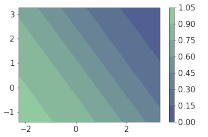
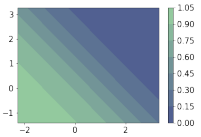
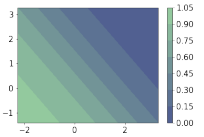
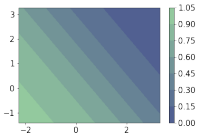
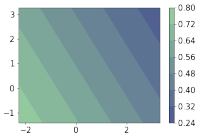
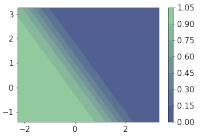
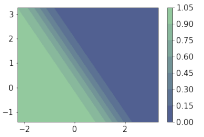
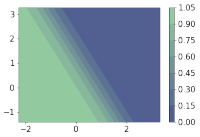
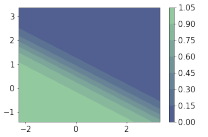
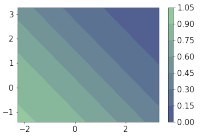
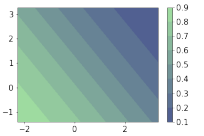
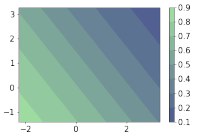
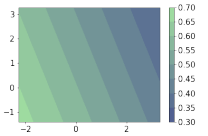
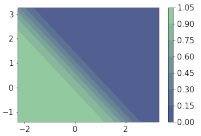
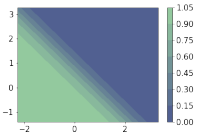
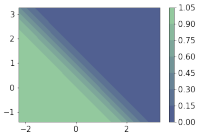
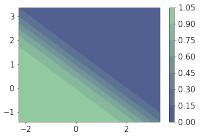
Objective	20% Noise	30% Noise	40% Noise	80% Noise
Expected Loss				
CVaR				
Entropic Risk				
Human-Aligned Risk				
Inverted CVaR				
Mean-Variance				
Trimmed Risk				

Figure 3. **Classification: Noisy Labels, Label Shift.** The decision boundaries of models learned under each objective and different levels of noise. The color bar indicates the predicted likelihood of each class: **blue** means higher probability of the blue class, and **green** means higher probability of the green class. The less uniform the decision boundary, the lower average loss a model has for this dataset. CVaR incurs high average loss due to its uncertainty across all points.

Training Objective	Noise	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.2	0.41	0.87	0.37	**0.46**	0.19	0.86	0.23
CVaR	0.2	0.67	**0.70**	0.66	0.67	0.59	0.69	0.63
Entropic Risk	0.2	**0.40**	1.00	0.33	0.47	0.12	1.21	0.17
Human-Aligned Risk	0.2	0.44	0.83	0.41	0.48	0.24	0.75	0.28
Inverted CVaR	0.2	0.49	1.42	0.35	0.61	0.08	2.37	0.13
Mean-Variance	0.2	0.51	0.76	0.50	0.53	0.36	**0.65**	0.38
Trimmed Risk	0.2	0.47	1.41	**0.31**	0.60	**0.06**	2.66	**0.09**
Expected Loss	0.3	0.48	0.95	0.45	**0.53**	0.25	0.88	0.30
CVaR	0.3	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.3	**0.47**	1.14	0.40	0.54	0.16	1.28	0.21
Human-Aligned Risk	0.3	0.51	0.88	0.48	0.54	0.31	0.76	0.35
Inverted CVaR	0.3	0.58	1.70	0.39	0.72	0.09	2.94	0.14
Mean-Variance	0.3	0.56	0.80	0.55	0.58	0.42	**0.68**	0.44
Trimmed Risk	0.3	0.58	1.76	**0.37**	0.73	**0.06**	3.37	**0.10**
Expected Loss	0.4	0.54	0.98	0.51	**0.58**	0.31	0.85	0.35
CVaR	0.4	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.4	**0.52**	1.19	0.46	0.59	0.21	1.19	0.26
Human-Aligned Risk	0.4	0.56	0.90	0.54	**0.58**	0.37	0.76	0.40
Inverted CVaR	0.4	0.66	1.95	0.44	0.81	0.10	3.27	0.15
Mean-Variance	0.4	0.60	0.81	0.59	0.61	0.46	**0.70**	0.49
Trimmed Risk	0.4	0.68	2.08	**0.43**	0.85	**0.07**	3.84	**0.11**
Expected Loss	0.8	0.64	0.91	0.63	**0.66**	0.48	0.78	0.52
CVaR	0.8	0.69	**0.70**	0.69	0.69	0.62	**0.71**	0.66
Entropic Risk	0.8	**0.63**	1.02	**0.61**	**0.66**	0.42	0.87	0.48
Human-Aligned Risk	0.8	0.65	0.86	0.65	0.67	0.52	0.75	0.55
Inverted CVaR	0.8	0.95	2.54	0.68	1.09	**0.23**	3.43	0.36
Mean-Variance	0.8	0.66	0.79	0.66	0.67	0.56	0.72	0.59
Trimmed Risk	0.8	0.86	2.24	0.64	0.98	**0.23**	2.90	**0.34**

Table 4. **Classification: Noisy Labels, Label Shift.** Comparison of **train** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. We observe that optimizing for a risk functional improves performance for that objective under the training data distribution.

Training Objective	Noise	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.2	0.30	0.53	0.29	0.32	0.18	0.39	0.21
CVaR	0.2	0.66	0.68	0.66	0.66	0.59	0.68	0.63
Entropic Risk	0.2	0.23	**0.48**	0.22	0.26	0.11	**0.35**	0.14
Human-Aligned Risk	0.2	0.35	0.56	0.34	0.36	0.23	0.43	0.26
Inverted CVaR	0.2	0.24	0.63	0.21	0.28	0.06	0.53	0.09
Mean-Variance	0.2	0.45	0.61	0.45	0.46	0.35	0.51	0.37
Trimmed Risk	0.2	**0.19**	0.51	**0.17**	**0.22**	**0.04**	0.42	**0.07**
Expected Loss	0.3	0.35	0.56	0.34	0.36	0.23	0.43	0.27
CVaR	0.3	0.69	0.70	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.3	0.26	0.50	0.25	0.28	0.14	**0.36**	0.17
Human-Aligned Risk	0.3	0.41	0.60	0.40	0.42	0.29	0.47	0.32
Inverted CVaR	0.3	0.22	0.57	0.20	0.26	0.06	0.47	0.09
Mean-Variance	0.3	0.50	0.64	0.50	0.51	0.40	0.55	0.42
Trimmed Risk	0.3	**0.17**	**0.47**	**0.16**	**0.21**	**0.04**	0.37	**0.06**
Expected Loss	0.4	0.40	0.59	0.39	0.41	0.28	0.47	0.31
CVaR	0.4	0.69	0.70	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.4	0.30	0.53	0.29	0.32	0.18	0.39	0.21
Human-Aligned Risk	0.4	0.46	0.63	0.45	0.47	0.34	0.52	0.37
Inverted CVaR	0.4	0.20	0.51	0.18	0.23	0.06	0.40	0.09
Mean-Variance	0.4	0.54	0.66	0.53	0.54	0.44	0.58	0.47
Trimmed Risk	0.4	**0.16**	**0.43**	**0.15**	**0.19**	**0.04**	**0.33**	**0.06**
Expected Loss	0.8	0.57	0.75	0.56	0.58	0.44	0.64	0.48
CVaR	0.8	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.8	0.52	0.77	0.51	0.53	0.37	**0.62**	0.41
Human-Aligned Risk	0.8	0.60	0.75	0.59	0.61	0.49	0.65	0.52
Inverted CVaR	0.8	0.67	1.40	0.53	0.72	0.31	1.63	0.47
Mean-Variance	0.8	0.63	0.72	0.62	0.63	0.54	0.66	0.57
Trimmed Risk	0.8	**0.48**	0.97	**0.43**	**0.52**	**0.24**	0.88	**0.35**

Table 5. **Classification: Noisy Labels, Label Shift.** Comparison of **test** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Entropic risk, inverted CVaR, and trimmed risk are better at ignoring noisy labels during training time than expected loss. As a result, those former risk functionals induce models which achieve lower average loss (besides inverted CVaR with 80% noise) on a non-corrupted test dataset.

Objective	Parameters	20% Noise		30% Noise		40% Noise		80% Noise	
		Train	Test	Train	Test	Train	Test	Train	Test
Expected Loss	-	0.87	0.87	0.83	0.83	0.79	0.78	0.64	0.63
CVaR	$\alpha = 0.3$	0.85	0.85	0.62	0.61	0.56	0.56	0.46	0.47
Entropic Risk	$t = -1$	0.87	0.86	0.83	0.82	0.79	0.79	0.64	0.63
Human-Aligned Risk	$a = 0.4, b = 0.8$	0.87	0.87	0.83	0.82	0.78	0.78	0.63	0.62
Inverted CVaR	$\alpha = 0.7$	0.82	0.82	0.80	0.80	0.75	0.75	0.63	0.62
Mean-Variance	$c = 1$	0.87	0.86	0.83	0.82	0.78	0.77	0.63	0.63
Trimmed Risk	$\alpha = 0.3$	0.85	0.85	0.82	0.82	0.78	0.78	0.63	0.63

Table 6. **Classification: Noisy Labels, No Label Shift.** Train and test accuracies for varying levels of noise when both the training and test data is corrupted with noisy labels. Since the train and test dataset follow the same distribution, train and test accuracies are expected to be similar.

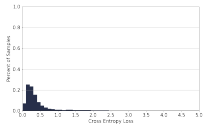
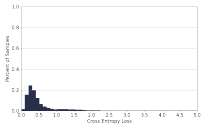
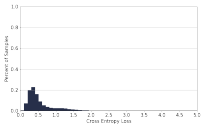
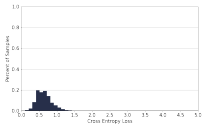
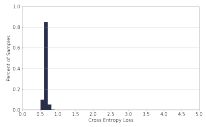
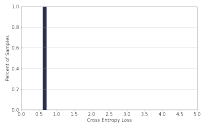
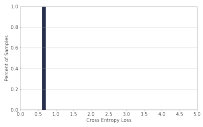
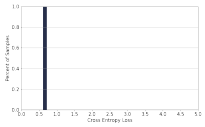
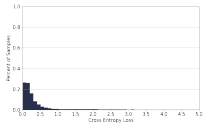
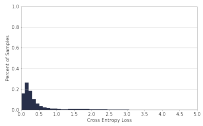
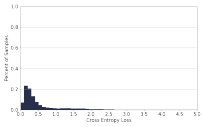
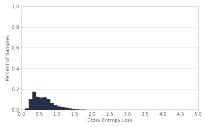
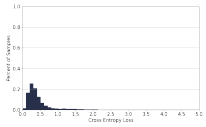
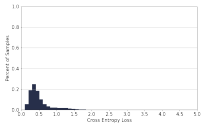
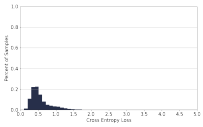
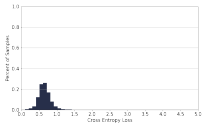
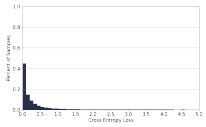
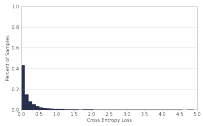
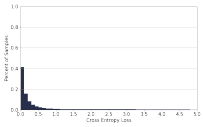
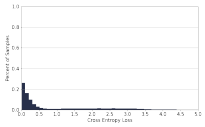
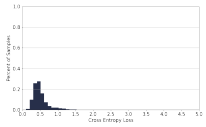
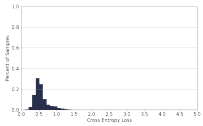
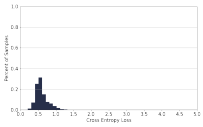
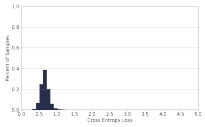
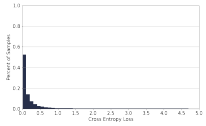
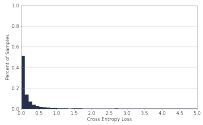
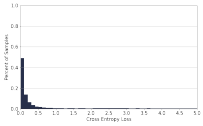
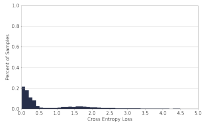
Objective	20% Noise	30% Noise	40% Noise	80% Noise
Expected Loss				
CVaR				
Entropic Risk				
Human-Aligned Risk				
Inverted CVaR				
Mean-Variance				
Trimmed Risk				

Figure 4. **Classification: Noisy Labels, Label Shift.** The **training** loss distributions of models learned under each objective and different levels of noise. Risk functionals which achieve lower expected loss on the test set than the expected loss risk functional are entropic risk, inverted CVaR, and trimmed risk. Models learned with these functionals have a greater proportion of smaller losses than expected loss. CVaR achieves uniform performance across all data points but incurs high loss for all.

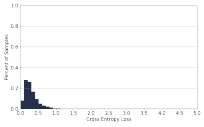
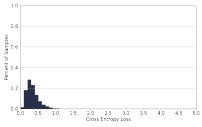
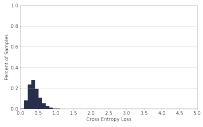
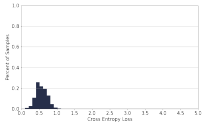
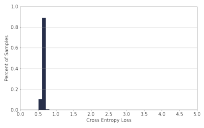
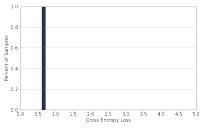
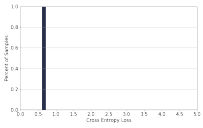
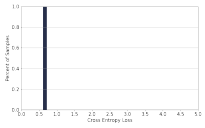
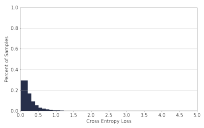
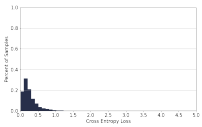
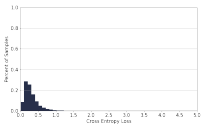
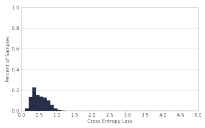
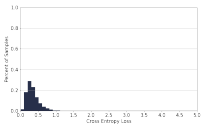
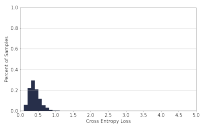
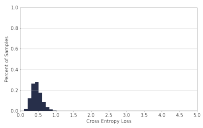
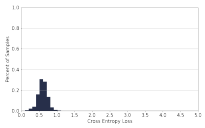
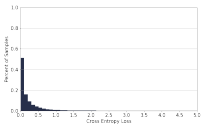
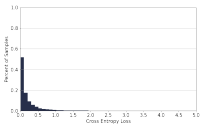
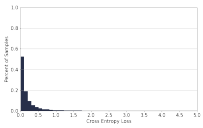
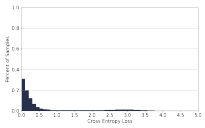
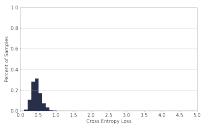
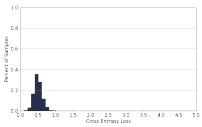
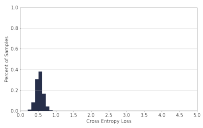
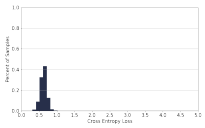
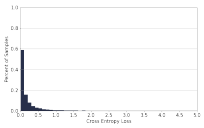
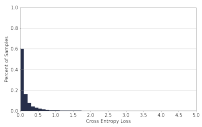
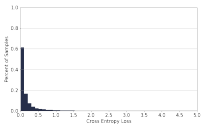
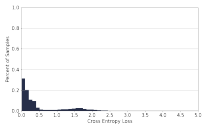
Objective	20% Noise	30% Noise	40% Noise	80% Noise
Expected Loss				
CVaR				
Entropic Risk				
Human-Aligned Risk				
Inverted CVaR				
Mean-Variance				
Trimmed Risk				

Figure 5. **Classification: Noisy Labels, Label Shift.** The **test** loss distributions of models learned under each objective and different levels of noise. Risk functionals which achieve the lowest expected loss have heavier concentrations of losses towards the left side of the graph. While CVaR has a shorter tail distribution than all other risk functionals, it has high loss for all data points representing worse performance.

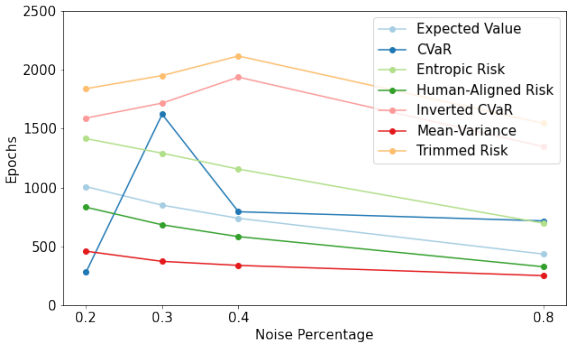


Figure 6. **Classification: Noisy Labels, Label Shift.** Average number of training epochs till convergence criteria is met when training under each risk functional.

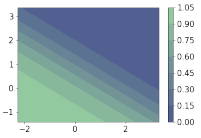
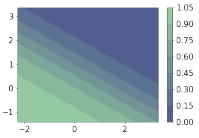
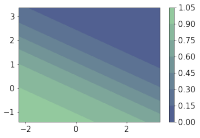
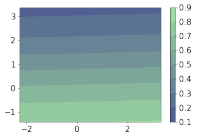
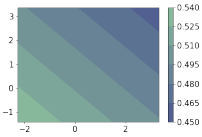
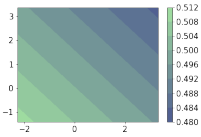
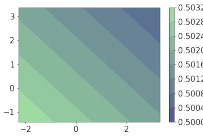
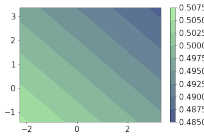
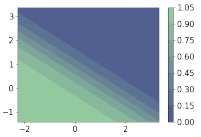
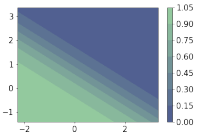
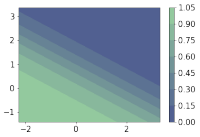
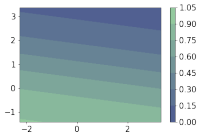
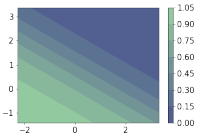
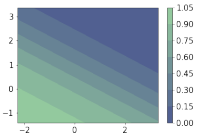
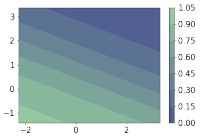
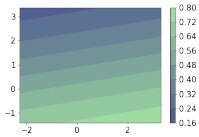
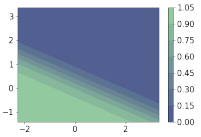
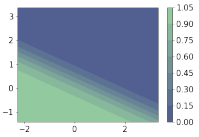
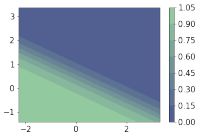
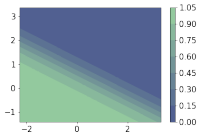
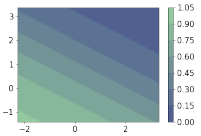
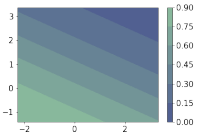
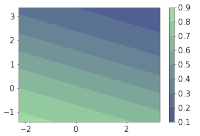
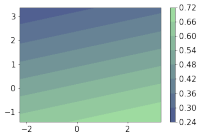
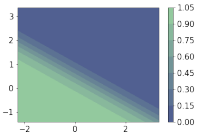
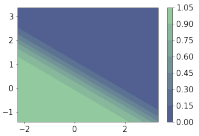
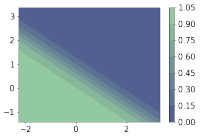
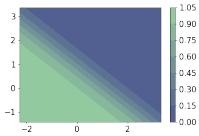
Objective	20% Noise	30% Noise	40% Noise	80% Noise
Expected Loss				
CVaR				
Entropic Risk				
Human-Aligned Risk				
Inverted CVaR				
Mean-Variance				
Trimmed Risk				

Figure 7. **Classification: Noisy Labels, No Label Shift.** The decision boundaries of models learned under each objective and different levels of noise. The color bar indicates the predicted likelihood of each class: **blue** means higher probability of the blue class, and **green** means higher probability of the green class. Similar to the Classification: Noisy Labels, Label Shift experiment, CVaR has a uniform decision boundary since both classes of data exist on both sides of the decision boundary during training.

Training Objective	Noise	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.2	0.42	0.88	0.38	**0.47**	0.20	0.87	0.24
CVaR	0.2	0.67	**0.70**	0.67	0.67	0.59	0.69	0.63
Entropic Risk	0.2	**0.40**	1.01	0.33	0.48	0.12	1.24	0.17
Human-Aligned Risk	0.2	0.44	0.83	0.42	0.48	0.25	0.76	0.29
Inverted CVaR	0.2	0.49	1.43	0.35	0.61	0.08	2.40	0.13
Mean-Variance	0.2	0.51	0.77	0.50	0.53	0.36	**0.65**	0.39
Trimmed Risk	0.2	0.47	1.42	**0.31**	0.60	**0.06**	2.70	**0.09**
Expected Loss	0.3	0.48	0.95	0.44	**0.52**	0.25	0.88	0.29
CVaR	0.3	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.3	**0.46**	1.12	0.39	0.54	0.16	1.27	0.21
Human-Aligned Risk	0.3	0.50	0.88	0.48	0.53	0.30	0.76	0.34
Inverted CVaR	0.3	0.57	1.67	0.39	0.71	0.09	2.89	0.14
Mean-Variance	0.3	0.56	0.80	0.55	0.58	0.41	**0.68**	0.43
Trimmed Risk	0.3	0.57	1.72	**0.36**	0.72	**0.06**	3.30	**0.10**
Expected Loss	0.4	0.53	0.98	0.50	**0.57**	0.30	0.85	0.35
CVaR	0.4	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.4	**0.52**	1.18	0.45	0.58	0.21	1.19	0.26
Human-Aligned Risk	0.4	0.55	0.90	0.53	0.58	0.36	0.76	0.40
Inverted CVaR	0.4	0.68	1.94	0.47	0.82	0.13	3.11	0.19
Mean-Variance	0.4	0.60	0.81	0.59	0.61	0.46	**0.70**	0.48
Trimmed Risk	0.4	0.67	2.06	**0.42**	0.84	**0.07**	3.78	**0.11**
Expected Loss	0.8	0.65	0.91	0.63	**0.66**	0.48	0.78	0.52
CVaR	0.8	0.69	**0.70**	0.69	0.69	0.62	**0.71**	0.66
Entropic Risk	0.8	**0.64**	1.02	**0.61**	**0.66**	0.42	0.87	0.49
Human-Aligned Risk	0.8	0.66	0.86	0.65	0.67	0.52	0.75	0.55
Inverted CVaR	0.8	0.95	2.52	0.69	1.08	**0.24**	3.33	0.37
Mean-Variance	0.8	0.67	0.79	0.66	0.67	0.56	0.72	0.59
Trimmed Risk	0.8	0.85	2.18	0.65	0.96	0.25	2.72	**0.35**

Table 7. **Classification: Noisy Labels, No Label Shift.** Comparison of **train** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Optimizing for a risk functional improves a models performance under that objective.

Training Objective	Noise	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Value	0.2	0.42	0.88	0.38	**0.47**	0.19	0.86	0.24
CVaR	0.2	0.67	**0.70**	0.67	0.67	0.59	0.69	0.63
Entropic Risk	0.2	**0.40**	1.01	0.33	**0.47**	0.12	1.21	0.17
Human-Aligned Risk	0.2	0.44	0.83	0.41	0.48	0.24	0.74	0.28
Inverted CVaR	0.2	0.50	1.44	0.35	0.62	0.08	2.40	0.13
Mean-Variance	0.2	0.51	0.76	0.50	0.53	0.36	**0.65**	0.39
Trimmed Risk	0.2	0.47	1.42	**0.31**	0.60	**0.06**	2.68	**0.09**
Expected Value	0.3	0.48	0.97	0.45	**0.53**	0.25	0.90	0.29
CVaR	0.3	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.3	**0.47**	1.15	0.40	0.55	0.16	1.31	0.21
Human-Aligned Risk	0.3	0.50	0.90	0.48	0.54	0.30	0.78	0.34
Inverted CVaR	0.3	0.58	1.71	0.40	0.72	0.09	2.92	0.14
Mean-Variance	0.3	0.56	0.80	0.55	0.58	0.41	**0.69**	0.44
Trimmed Risk	0.3	0.58	1.77	**0.37**	0.73	**0.06**	3.35	**0.10**
Expected Value	0.4	0.53	0.98	0.50	**0.57**	0.30	0.85	0.35
CVaR	0.4	0.69	**0.70**	0.69	0.69	0.62	0.71	0.66
Entropic Risk	0.4	**0.52**	1.19	0.46	0.59	0.21	1.19	0.26
Human-Aligned Risk	0.4	0.55	0.90	0.54	0.58	0.36	0.76	0.40
Inverted CVaR	0.4	0.69	1.96	0.48	0.83	0.13	3.12	0.20
Mean-Variance	0.4	0.60	0.81	0.59	0.61	0.46	**0.70**	0.49
Trimmed Risk	0.4	0.68	2.08	**0.42**	0.84	**0.07**	3.78	**0.11**
Expected Value	0.8	0.65	0.91	0.64	**0.66**	0.48	0.78	0.52
CVaR	0.8	0.69	**0.70**	0.69	0.69	0.62	**0.71**	0.66
Entropic Risk	0.8	**0.64**	1.02	**0.61**	**0.66**	0.42	0.88	0.49
Human-Aligned Risk	0.8	0.66	0.86	0.65	0.67	0.52	0.75	0.55
Inverted CVaR	0.8	0.95	2.50	0.69	1.08	**0.25**	3.29	0.38
Mean-Variance	0.8	0.67	0.79	0.66	0.67	0.56	0.72	0.59
Trimmed Risk	0.8	0.84	2.15	0.65	0.95	**0.25**	2.67	**0.36**

Table 8. **Classification: Noisy Labels, No Label Shift.** Comparison of **test** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Since the train and test distributions are the same, performance metrics are similar to those under the train distribution.

Training Objective	Accuracy	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.91	0.33	1.48	1.77	0.56	0.00	97.70	0.18
CVaR, $\alpha = 0.1$	**0.94**	0.61	**0.70**	0.64	0.62	0.01	**19.01**	0.53
Entropic Risk, $t = 10$	0.93	0.54	0.74	**0.60**	0.57	0.01	19.82	0.46
Human, $a = 0.4, b = 0$	0.91	0.40	0.93	0.82	**0.49**	0.00	31.11	0.29
Inverted CVaR, $\alpha = 0.1$	0.64	0.78	1.96	2.68	0.99	0.01	197.62	0.55
Mean-Variance, $c = 50$	0.93	0.65	0.71	0.66	0.66	0.01	19.86	0.58
Trimmed Risk, $\alpha = 0.1$	0.90	**0.31**	2.16	3.10	0.70	0.00	229.77	**0.10**

Table 9. **Classification: Covariate Shift.** Comparison of **train** accuracy and performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Aside from inverted CVaR, all risk functionals achieve high accuracies on the training set.

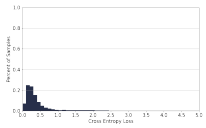
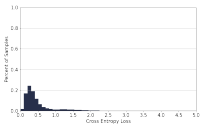
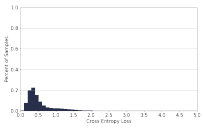
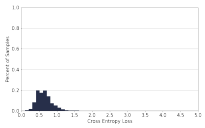
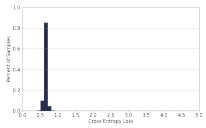
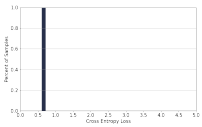
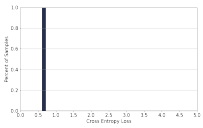
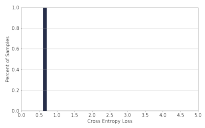
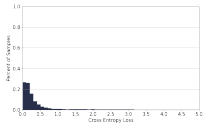
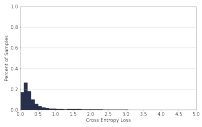
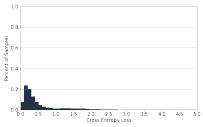
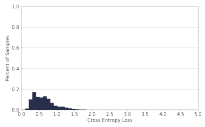
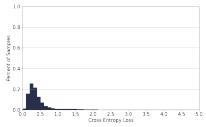
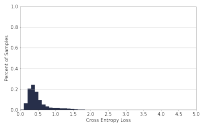
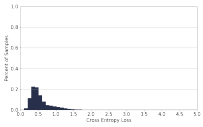
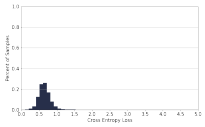
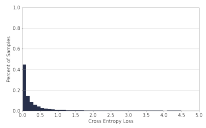
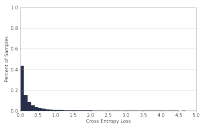
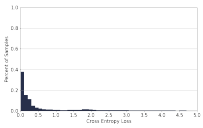
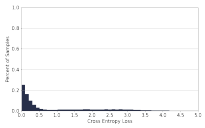
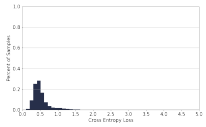
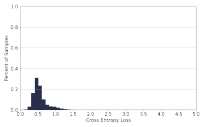
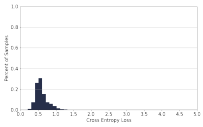
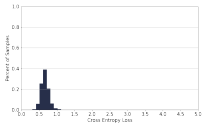
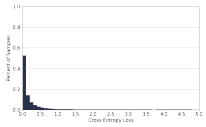
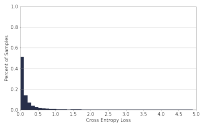
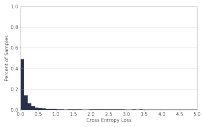
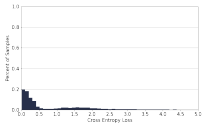
Objective	20% Noise	30% Noise	40% Noise	80% Noise
Expected Loss				
CVaR				
Entropic Risk				
Human-Aligned Risk				
Inverted CVaR				
Mean-Variance				
Trimmed Risk				

Figure 8. **Classification: Noisy Labels, No Label Shift.** The **training** loss distributions of models learned under each objective and different levels of noise. CVaR incurs high uniform loss across all data points.

Training Objective	Accuracy	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.20	1.29	2.10	2.07	1.34	0.01	188.67	0.94
CVaR, $\alpha = 0.1$	**0.59**	**0.66**	0.76	0.71	**0.67**	0.01	22.49	**0.58**
Entropic Risk, $t = 10$	0.41	0.69	0.84	0.75	0.70	0.01	25.72	**0.58**
Human, $a = 0.4, b = 0$	0.28	0.83	1.18	1.10	0.83	0.01	51.22	0.62
Inverted CVaR, $\alpha = 0.1$	0.38	1.12	2.64	3.20	1.42	0.01	349.25	0.86
Mean-Variance, $c = 50$	0.43	0.69	**0.73**	**0.69**	0.69	0.01	**21.10**	0.61
Trimmed Risk, $\alpha = 0.1$	0.18	1.85	3.45	3.54	2.06	0.02	505.05	1.38

Table 10. **Classification: Covariate Shift.** Comparison of **test** accuracy and performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. CVaR achieves nearly $3\times$ greater accuracy and $2\times$ lower average loss than the expected loss risk functional. High loss data points during training correspond to those with less examples. However, in this covariate shift dataset, the training minority class corresponds to the test majority class. CVaR is able to protect against this distribution shift.

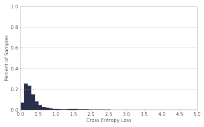
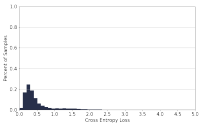
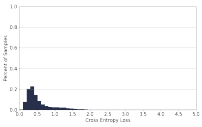
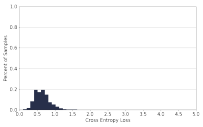
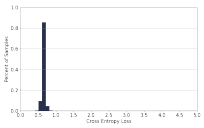
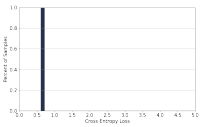
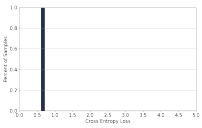
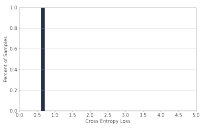
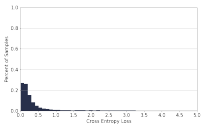
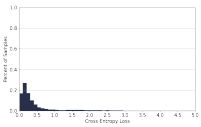
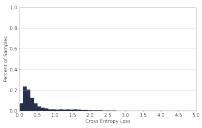
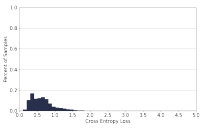
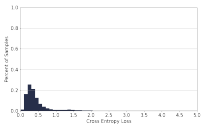
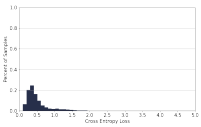
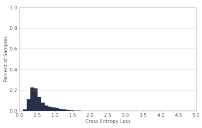
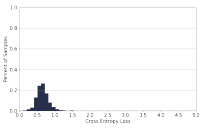
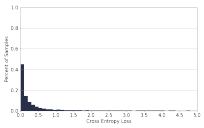
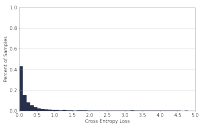
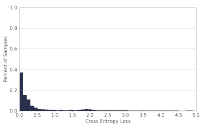
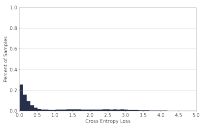
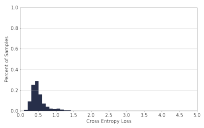
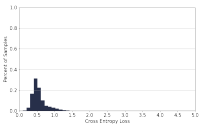
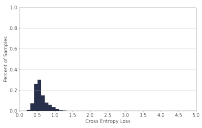
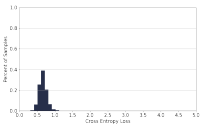
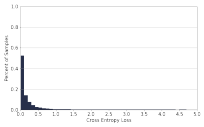
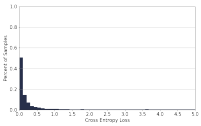
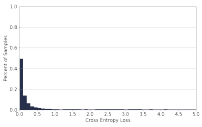
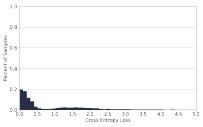
Objective	20% Noise	30% Noise	40% Noise	80% Noise
Expected Loss				
CVaR				
Entropic Risk				
Human-Aligned Risk				
Inverted CVaR				
Mean-Variance				
Trimmed Risk				

Figure 9. **Classification: Noisy Labels, No Label Shift.** The **test** loss distributions of models learned under each objective and different levels of noise. Since there is no distribution shift, the test loss distribution is similar to the training loss distribution.

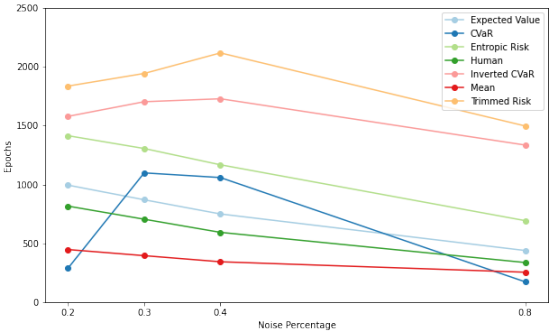


Figure 10. **Classification: Noisy Labels, No Label Shift.** Average umber of training epochs till convergence criteria is met when training under each risk functional.

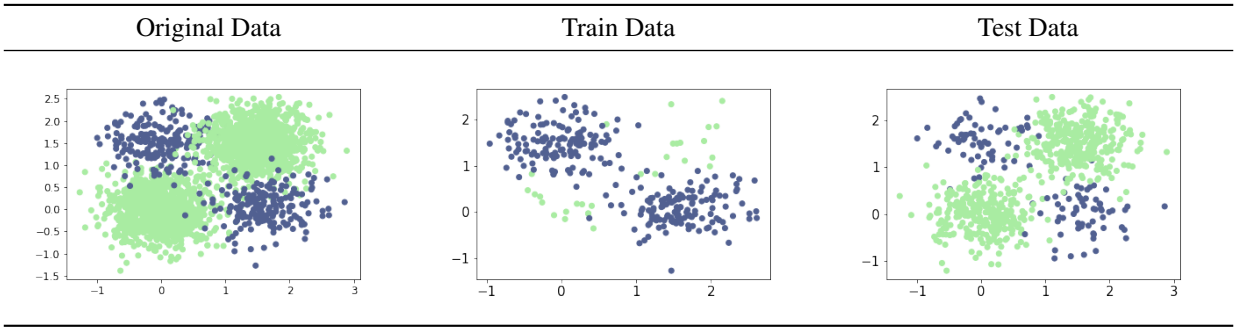


Figure 11. Covariate shift train and test datasets.

B.4. Regression: Minority Group Performance

In this setting we study which risk functionals produce a model that minimizes risk across all subpopulations of data. We follow the setup from (Duchi & Namkoong, 2018; Leqi et al., 2019). We draw covariates $X \sim \mathcal{N}(0, \mathbf{I}_5) \in \mathbb{R}^5$ and let the noise distribution be $\epsilon \sim \mathcal{N}(0, 0.01)$. The target Y is drawn from

$$Y = \begin{cases} X^T\theta^* + \epsilon & \text{if } X^{(1)} \leq 1.645 \\ X^T\theta^* + X^{(1)} + \epsilon & \text{otherwise} \end{cases} \tag{1}$$

where $\theta^* = \mathbf{1}^5$ and $X^{(1)}$ is the first coordinate of X. Since $P(X^{(1)} > 1.645) = 0.05$, $\{X | X^{(1)} > 1.645\}$ is the minority subpopulation. There are 2000 training points and 20000 test points.

Testing and training performance under different risk functionals are summarized in Table 11 and Table 12 respectively. Mean squared error is reported. We compute the risk for the majority subpopulation, minority subpopulation, and overall population in the "Majority", "Minority", and "Expected" columns in Table 11. CVaR achieves the lowest minority risk at the expense of the highest majority risk. Entropic risk, human-aligned risk, and mean variance achieve lower majoirty risk than CVaR, and lower minority risk than expected value, inverted CVaR, and trimmed risk.

Figure 14 illustrates the loss distributions incurred by each model. Models with longer distribution tails achieve lower minority risk as the tail ends of the distribution reflect leeway in predicting samples from the majority subpopulation. Models with very short tails reflect those which heavily prioritize the majority subpopulation. Figure 15 plots the average number of training epochs for each objective. Trimmed risk and inverted CVaR, which achieve the lowest majority risks, take longer to converge.

Training Objective	Majority	Minority	Expected	CVaR	Entropic	Human	InvCVaR	MV	Trim
Expected Loss	0.04	2.81	**0.17**	1.49	0.46	0.43	0.02	2.23	0.03
CVaR, $\alpha = 0.1$	0.18	**0.96**	0.22	**0.87**	0.24	**0.33**	0.12	**0.63**	0.14
Entropic Risk, $t = 0.1$	0.13	1.42	0.19	0.99	0.24	0.34	0.09	0.85	0.10
Human, $a = 0.5, b = 0$	0.11	1.50	0.18	1.00	**0.23**	**0.33**	0.08	0.89	0.09
Inverted CVaR, $\alpha = 0.9$	**0.02**	5.23	0.28	2.67	3.42	0.75	0.02	6.86	0.02
Mean-Variance, $c = 0.5$	0.12	1.45	0.19	0.99	**0.23**	**0.33**	0.08	0.86	0.09
Trimmed Risk, $\alpha = 0.1$	**0.02**	5.14	0.27	2.61	3.18	0.73	**0.01**	6.62	**0.01**

Table 11. **Regression: Minority Group Performance.** Comparison of **test** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Majority is the average loss, or risk, under the majority subpopulation. Minority is the risk under the minority subpopulation. CVaR, entropic risk, human-aligned risk, and mean-variance optimize for the tail performances which results in lower minority risk than expected loss. Inverted CVaR and trimmed risk ignore the tail losses, resulting in higher minority risk.

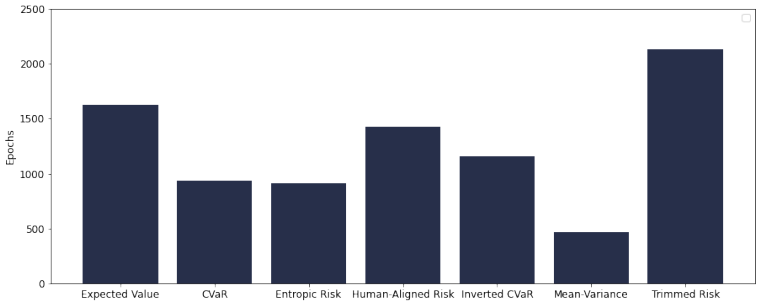


Figure 12. **Classification: Covariate Shift.** Average number of training epochs till convergence criteria is met when training under each risk functional.

Training Objective	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	**0.17**	1.42	0.42	0.41	0.02	2.09	0.03
CVaR, $\alpha = 0.1$	0.22	**0.85**	0.24	**0.32**	0.12	**0.61**	0.14
Entropic Risk, $t = 0.1$	0.19	0.96	0.23	0.33	0.08	0.80	0.09
Human, $a = 0.5, b = 0$	0.18	0.96	**0.22**	**0.32**	0.08	0.83	0.09
Inverted CVaR, $\alpha = 0.9$	0.27	2.56	2.33	0.72	0.02	6.54	0.02
Mean-Variance, $c = 0.5$	0.18	0.96	**0.22**	**0.32**	0.08	0.81	0.09
Trimmed Risk, $\alpha = 0.1$	0.26	2.51	2.20	0.70	**0.01**	6.32	**0.01**

Table 12. **Regression: Minority Group Performance.** Comparison of **train** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Optimizing for a risk functional improves performance under the same objective and dataset.

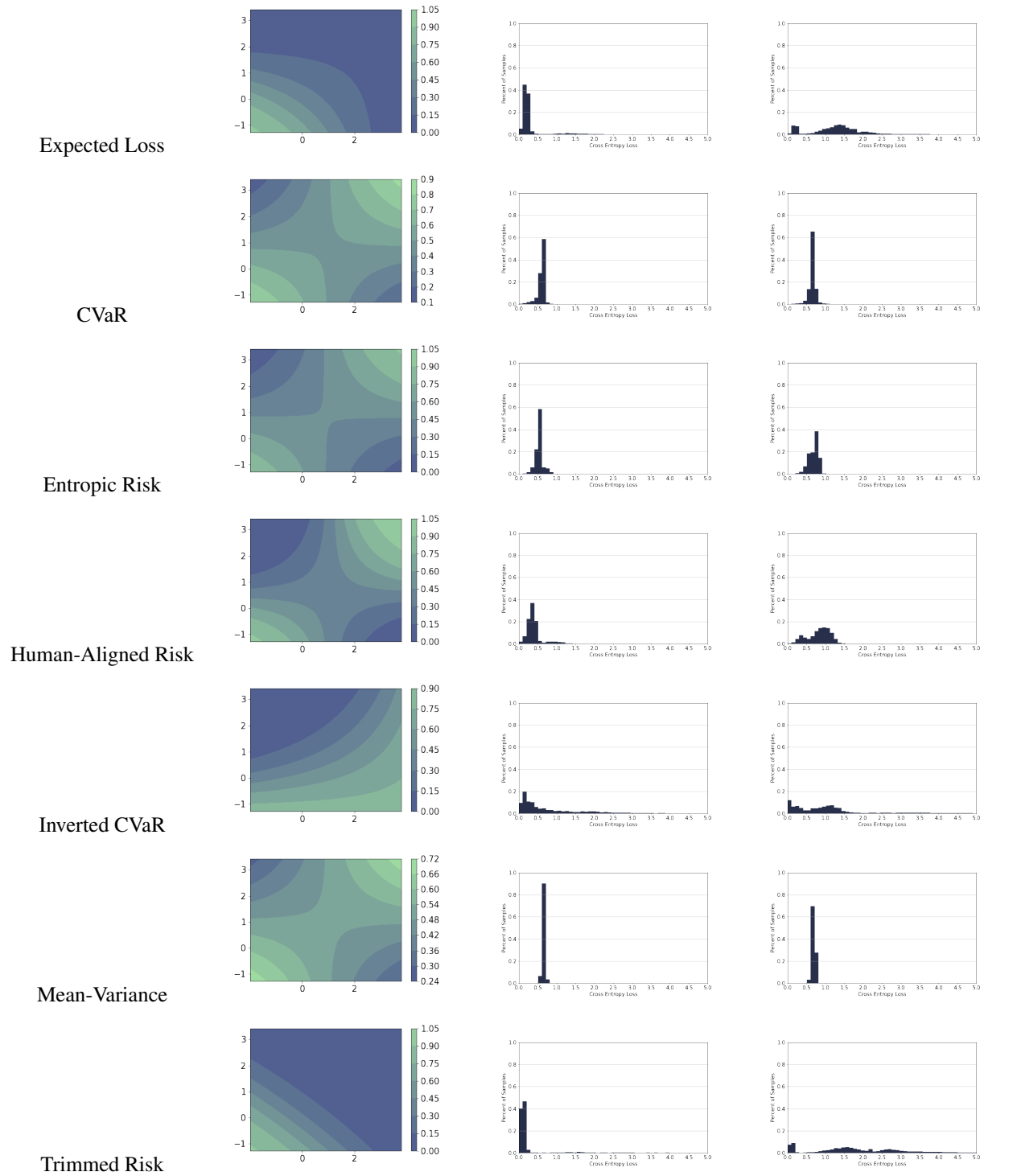


Figure 13. **Classification: Covariate Shift.** The decision boundaries, training loss distributions, and test loss distributions of models learned under each objective. For decision boundaries the color bar indicates the predicted likelihood of each class: **blue** means higher probability of the blue class, and **green** means higher probability of the green class. CVaR, entropic risk, human-aligned risk, and mean-variance incur higher training losses than expected loss. However, this results in the correct decision boundary and shorter test loss distribution tails.

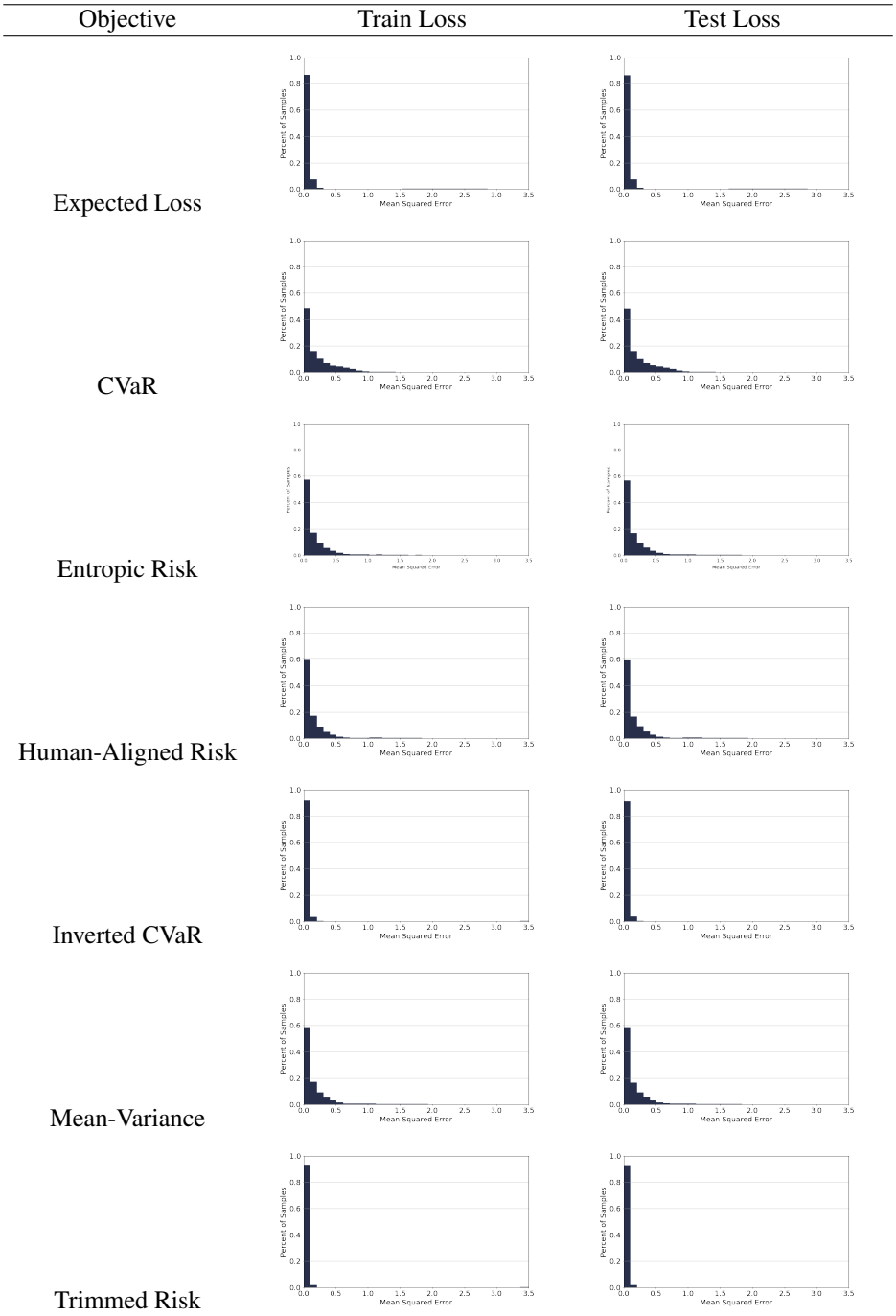


Figure 14. **Regression: Minority Group Performance Dataset.** The train and test loss distributions under both majority and minority data points of models learned under each objective. CVaR, entropic risk, human-aligned risk, and mean-variance have longer tails than expected loss. However, this represents those models not overfitting to the majority class and not ignoring the minority class.

B.5. Regression: Label Shift

We follow the same setup as the Minority Group Performance dataset but let the training targets Y be drawn from

$$Y = \begin{cases} X^T\theta^* + \epsilon + 0.5 & \text{if } X^{(1)} \leq 1.645 \\ X^T\theta^* + X^{(1)} + \epsilon + 0.5 & \text{otherwise} \end{cases} \tag{2}$$

where all targets are shifted up by 0.5. We leave the test targets unshifted. All other dataset details remain the same.

Results are summarized in Table 13 and Table 14 with mean squared error reported. Inverted CVaR, entropic risk and trimmed risk outperform the expected loss risk functional in average loss on the test set. Loss distributions are shown in Figure 16. Figure 17 summarizes the average number of training epochs for each risk functional.

Training Objective	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.40	0.43	0.36	0.42	0.33	**0.44**	0.36
CVaR, $\alpha = 0.9$	0.40	0.44	0.36	0.42	0.33	**0.44**	0.36
Entropic Risk, $t = -1$	**0.37**	**0.39**	0.29	**0.40**	**0.23**	0.47	**0.24**
Human-Aligned Risk, $a = 0.1, b = 0.9$	0.41	0.45	0.37	0.44	0.34	0.45	0.38
Inverted CVaR, $\alpha = 0.9$	**0.37**	**0.39**	**0.28**	0.41	**0.23**	0.48	**0.24**
Mean-Variance, $c = 0.1$	0.43	0.47	0.39	0.46	0.36	0.47	0.40
Trimmed Risk, $\alpha = 0.1$	**0.37**	**0.39**	**0.28**	0.41	**0.23**	0.48	**0.24**

Table 13. **Regression: Label Shift.** Comparison of **test** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Inverted CVaR, entropic risk, and trimmed risk achieve lower average test loss than the expected loss risk functional.

Training Objective	Expected	CVaR	Entropic	Human	InvCVaR	MV	Trim
Expected Loss	**0.15**	**0.17**	0.08	**0.19**	0.03	0.24	0.03
CVaR, $\alpha = 0.9$	**0.15**	**0.17**	0.08	**0.19**	0.03	0.24	0.04
Entropic Risk, $t = -1$	0.22	0.24	**0.05**	0.28	**0.00**	0.46	**0.00**
Human-Aligned Risk, $a = 0.1, b = 0.9$	**0.15**	**0.17**	0.09	**0.19**	0.04	0.23	0.05
Inverted CVaR, $\alpha = 0.9$	0.23	0.25	**0.05**	0.29	**0.00**	0.48	**0.00**
Mean-Variance, $c = 0.1$	0.16	0.18	0.10	**0.19**	0.06	**0.22**	0.06
Trimmed Risk, $\alpha = 0.1$	0.22	0.25	**0.05**	0.28	**0.00**	0.47	**0.00**

Table 14. **Regression: Label Shift.** Comparison of **train** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training. Expected loss, CVaR, human-aligned risk, and mean-variance achieve the lowest average train loss. However, when the labels are shifted in the test set, the former risk functionals have worse performance.

C. Comparison of CVaR Optimizers

CVaR has many proposed optimization methods. The dual form of CVaR is convex given a convex loss function (e.g. cross entropy loss) so optimizing over the form will obtain a global optimum. Aligning with prior work (Curi et al., 2020), we call this TruncCVaR: $\rho_{\text{CVaR}}(\alpha, \ell_f(X,Y)) \coloneqq \inf_{\eta\in\mathbb{R}} \left\{\alpha^{-1}\mathbb{E}\left[(\ell_f(X,Y) - \eta)_+ + \eta\right]\right\}$. We optimize both the CVaR objective and inner parameter η. The maximum operation inside the expectation of TruncCVaR is non-smooth and prior works have proposed Soft-CVaR, replacing $\mathbb{E}[x_+]$ with $T \log \mathbb{E}\left[e^{x/T}\right]$ (Nemirovski & Shapiro, 2007). Stochastic optimization of TruncCVaR can be challenging as only a few points from a batch of data will contain gradient information, and points which do have gradients may result in exploding gradients (Curi et al., 2020). To address this, an adaptive sampling approach has been proposed by Curi et al. (2020). Distributionally robust optimization methods have also been proposed to minimize CVaR under distribution shifts (Duchi & Namkoong, 2018; Duchi et al., 2020).

We compare all CVaR optimizers using the Classification: Noisy Label, No Label Shift dataset from Appendix B.2. Results are summarized in Figure 18. While all optimizers achieve high accuracy, each trade-off CVaR and expected loss performance. The first-order method labeled "CVaR GD" Leqi et al. (2022) achieves the lowest test CVaR, but the highest average loss. Soft-CVaR achieves the lowest average loss at the expense of the highest CVaR. "Uniform Performance" (Duchi & Namkoong, 2018), "Covariate Mixtures" (Duchi et al., 2020), "Adaptive Sampling" (Curi et al., 2020), and TruncCVaR perform between the other two methods.

D. CIFAR-10: Learning under Noisy Labels

We train VGG-11 models to optimize each of the risk functionals from Section 2.1. CIFAR-10 contains 50000 training samples and 10000 test samples. We corrupt 80% of the training labels by sampling targets uniformly at random from all classes. This results in around 27% of training labels matching the ground truth. We train the models for 150 epochs using a learning rate of 5e-3 and batch sizes of 5000. Results are summarized in Table 15 and Table 16. Entropic risk, human-aligned risk, and mean-variance achieve the highest train and test accuracies. Interestingly, with a negative c parameter, mean-variance achieves higher accuracy and lower variance than expected loss.

Training Objective	Accuracy	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.19	**2.24**	2.31	2.20	2.04	2.17	**2.25**	2.25
CVaR, $\alpha = 0.9$	0.17	2.27	**2.30**	2.27	2.20	2.24	2.27	2.27
Entropic Risk, $t = -0.5$	**0.21**	2.29	2.45	**2.14**	1.88	2.15	2.35	2.30
Human, $a = 0.8, b = 0.2$	0.20	2.37	2.57	2.16	**1.81**	2.25	2.45	2.41
Inverted CVaR, $\alpha = 0.9$	0.19	2.64	2.77	2.29	2.57	**2.13**	2.91	2.40
Mean-Variance, $c = -0.1$	0.20	2.24	2.33	2.18	1.99	2.16	2.26	2.25
Trimmed Risk, $\alpha = 0.05$	0.18	2.43	2.51	2.26	2.40	2.16	2.59	**2.22**

Table 15. **CIFAR-10: Learning under Noisy Labels** Comparison of **train** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training.

Training Objective	Accuracy	Expected	CVaR	Entropic	Human	Inv CVaR	MV	Trimmed
Expected Loss	0.47	1.95	2.02	1.92	1.78	1.89	1.96	1.95
CVaR, $\alpha = 0.9$	0.36	2.18	2.20	2.17	2.12	2.15	2.18	2.17
Entropic Risk, $t = -0.5$	**0.52**	1.61	1.73	1.50	1.31	1.48	1.66	1.60
Human, $a = 0.8, b = 0.2$	0.51	**1.56**	**1.69**	**1.39**	**1.21**	**1.39**	**1.63**	**1.54**
Inverted CVaR, $\alpha = 0.9$	0.42	2.24	2.35	2.00	2.15	1.86	2.40	2.05
Mean-Variance, $c = -0.1$	0.48	1.85	1.93	1.80	1.63	1.76	1.87	1.85
Trimmed Risk, $\alpha = 0.05$	0.41	2.07	2.12	2.03	2.00	1.98	2.10	2.02

Table 16. **CIFAR-10: Learning under Noisy Labels** Comparison of **test** performances under each risk functional of models learned under different training objectives. The rows represent the model learned under each training objective. The columns represent the model's performance under each risk functional with the same parameters used during training.

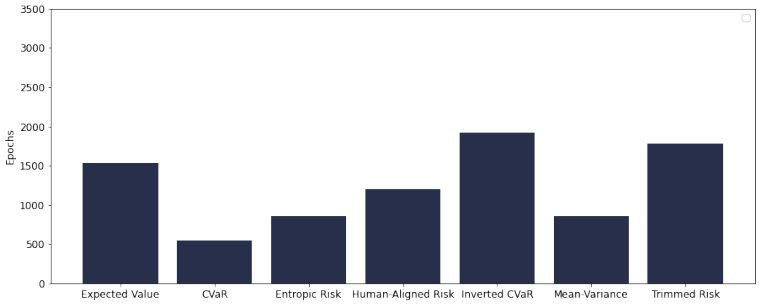


Figure 15. **Regression: Minority Group Performance.** Average number of training epochs till convergence criteria is met when training under each risk functional.

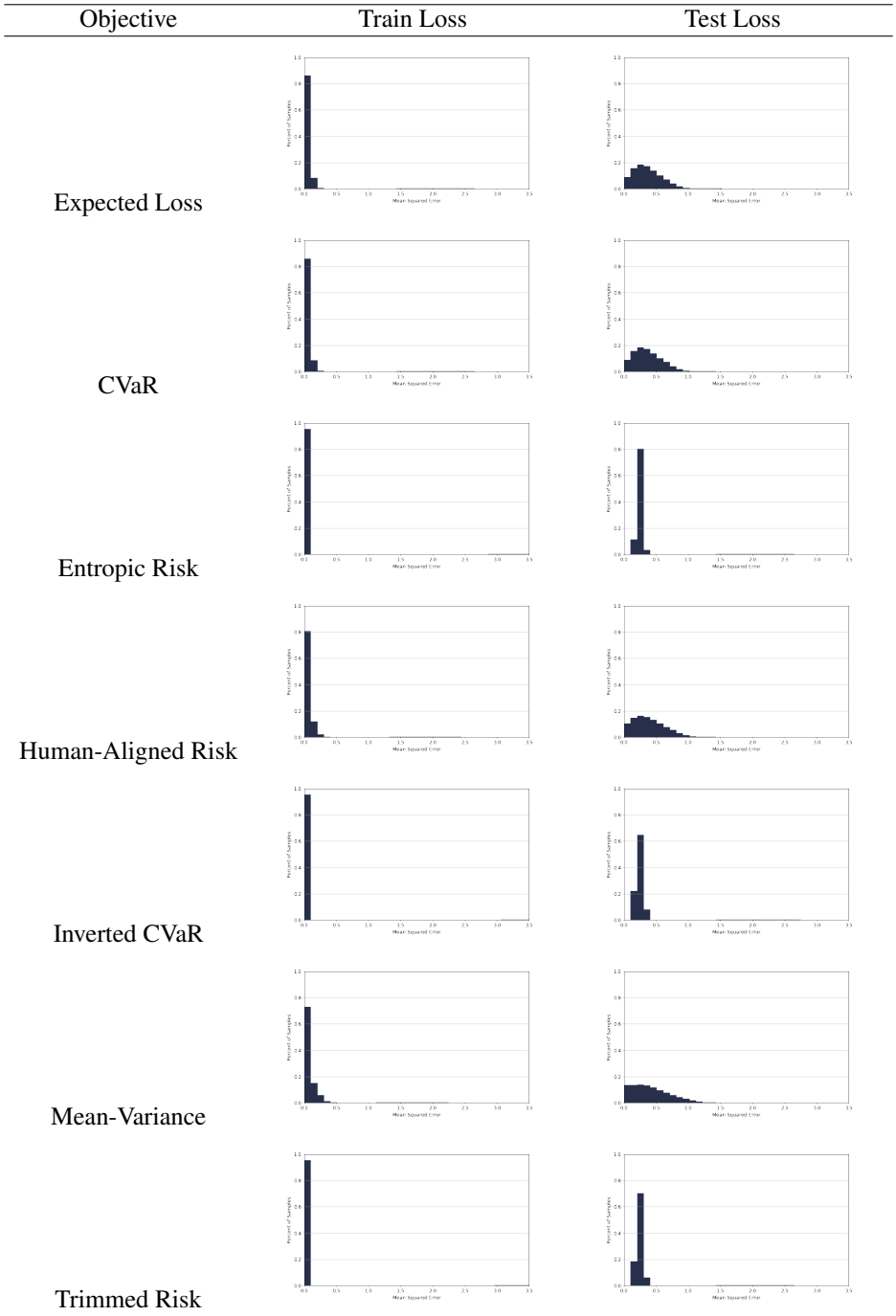


Figure 16. **Regression: Label Shift.** The train and test loss distributions of models learned under each objective. Risk functionals which are more robust to label shift achieve shorter test loss tail distributions.

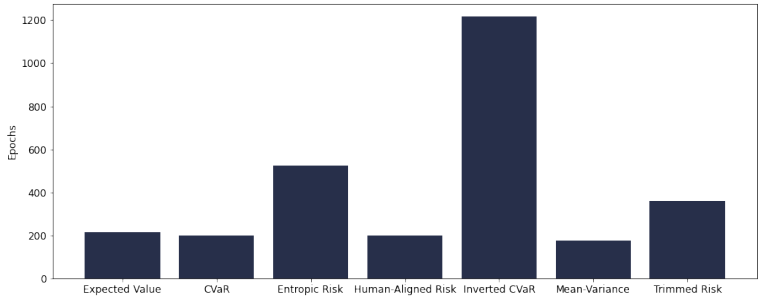


Figure 17. **Regression: Label Shift.** Average number of training epochs till convergence criteria is met when training under each risk functional.

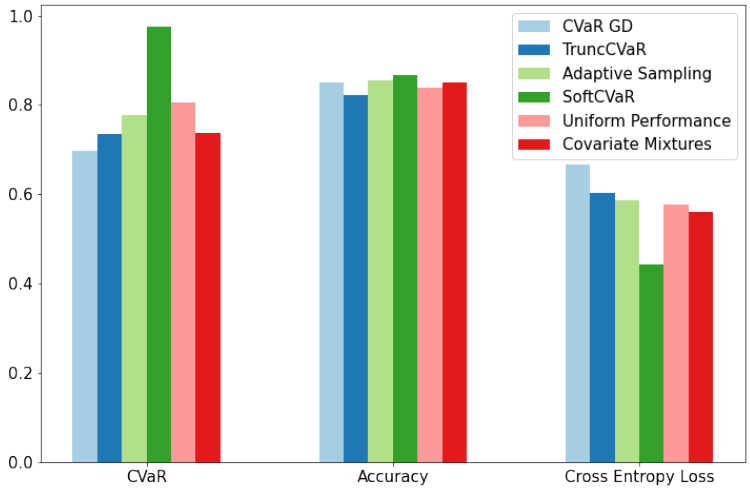


Figure 18. Comparison of CVaR optimizer performances achieved on the test set described in Appendix B.2. There is a trade-off between achieving low CVaR and low cross-entropy loss. The first-order method CVaR GD is the best at optimizing for CVaR.