Efficient Augmentation via Data Subsampling

Michael Kuchnik & Virginia Smith

Carnegie Mellon University

Carnegie Mellon University

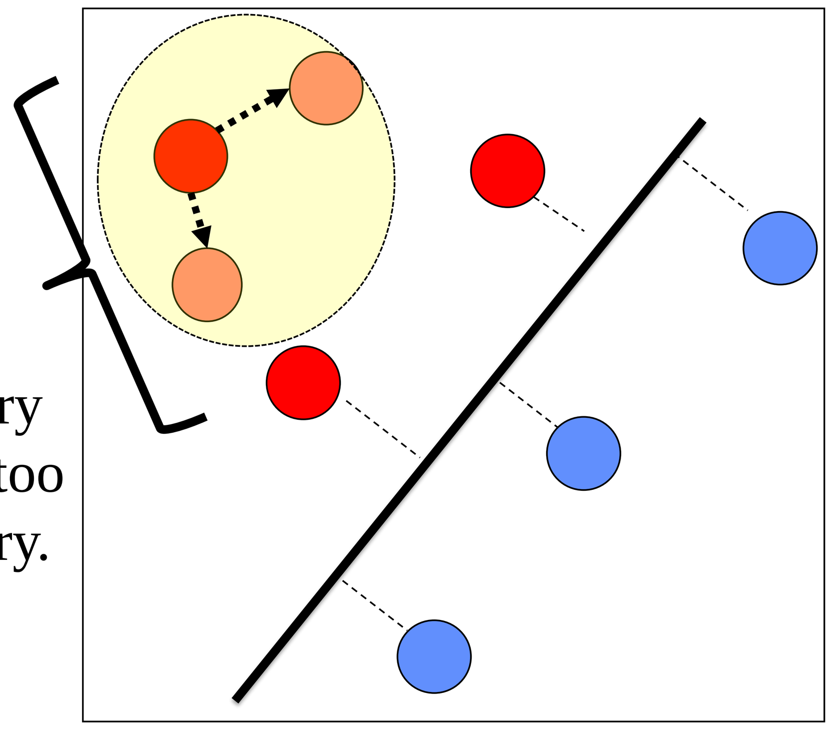
Introduction

- Data augmentation (e.g., rotate, crop) is commonly used to encode invariances in learning methods.
  - However, the process is inefficient and results in a dataset size explosion, since all points are augmented.
  - Large datasets are hard to debug, memory inefficient, and potentially slower to train with!
- We can recover most of the baseline augmentation accuracy using a small number of augmented points.

From Virtual Support Vector to Influential Data

- The virtual support vector (VSV) method [1,2] attempts to perform augmentation on SVM support vectors, since those affect the decision boundary.

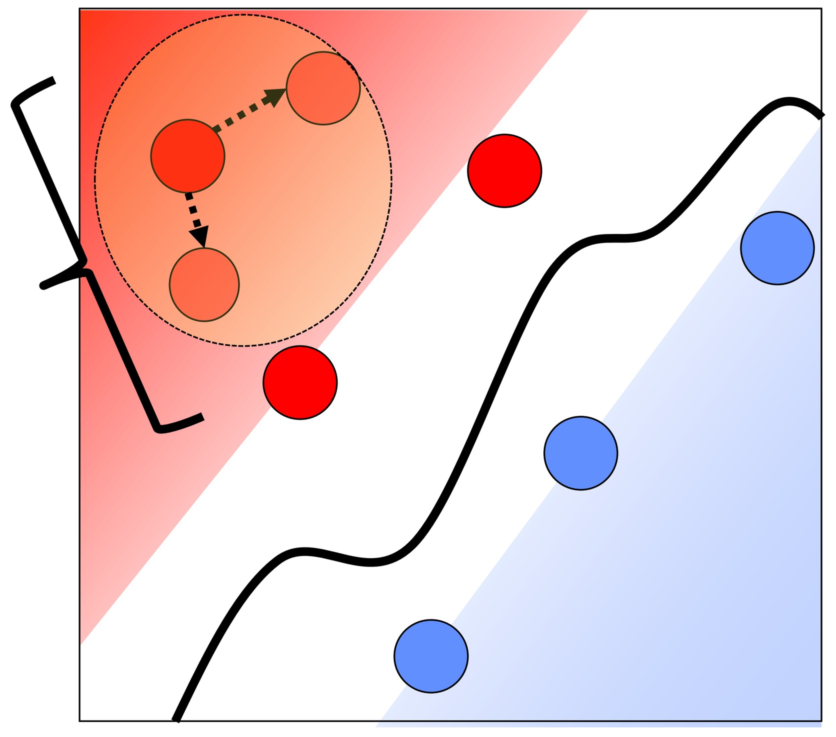
Augmentations don't change decision boundary since points are too far from boundary.



- We generalize this idea to non-SVM models using both point Leave-One-Out (LOO) influence [3] and loss.

$$\mathcal{I}_{\text{LOO}}(z) := \mathcal{I}_{\text{up,loss}}(z, z) = -\nabla_\theta L(z, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})$$

But what about points with low influence?

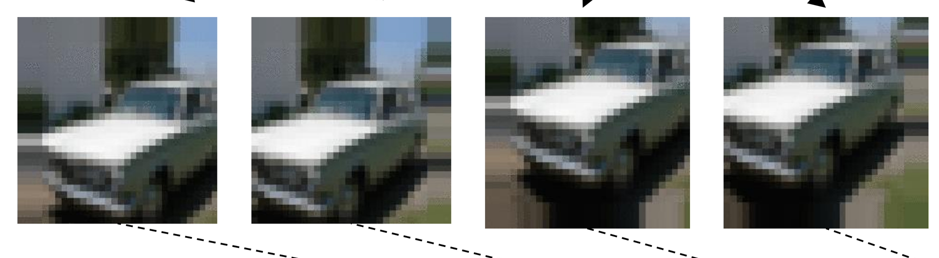


Augmentation Methodology

1. Loss/influence is calculated before augmentation.

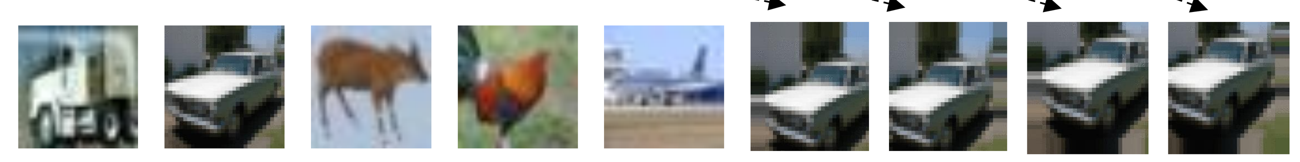
Loss: Low High Low Low Low

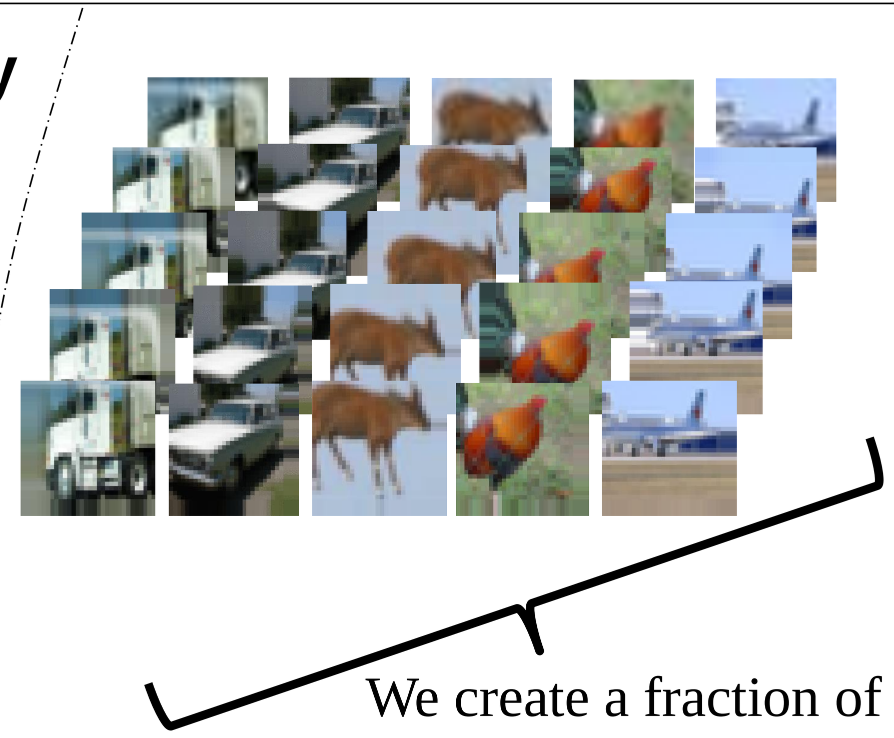
2. Point is augmented with probability proportional* to loss/influence.



Translated images

3. Append.



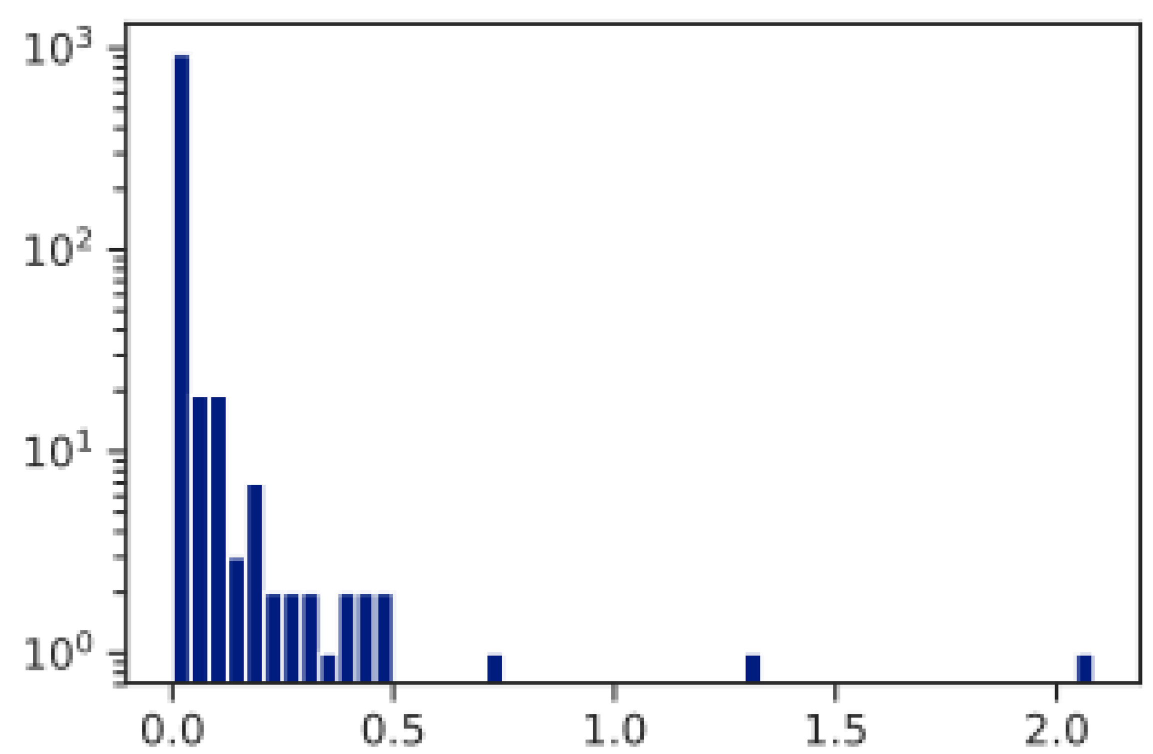


We create a fraction of the augmented points compared to full augmentation.

*see policy types

What Points Get Augmented?

Histogram of influences shown. Most points don't have high influence or loss.

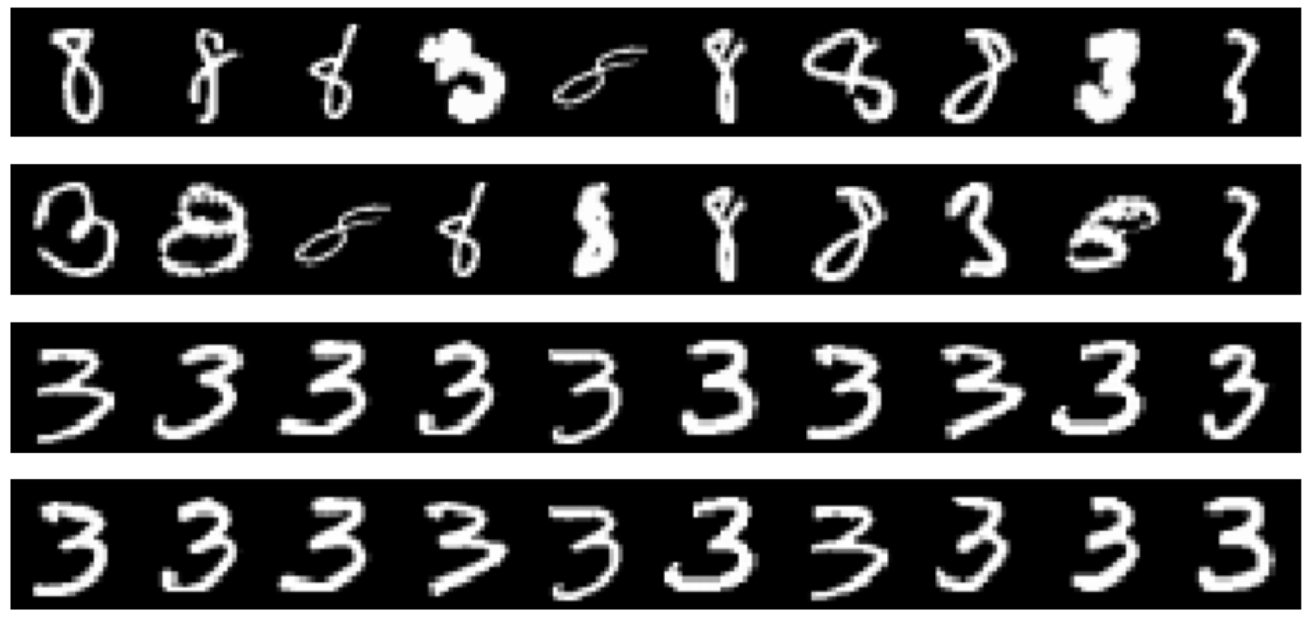


High Influence

High Loss

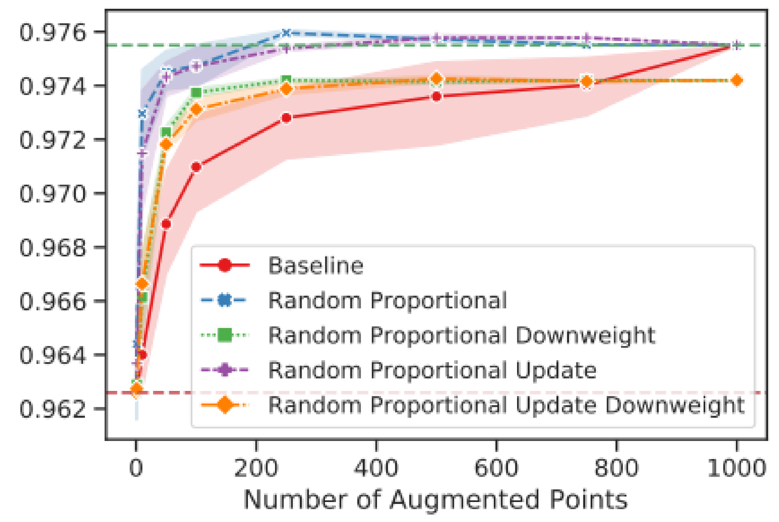
Low Influence

Low Loss

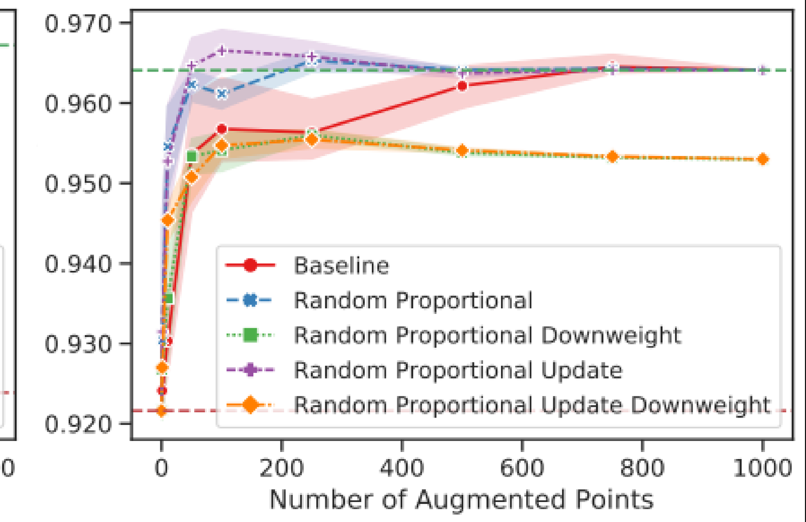


Matching Performance at a Fraction of the Cost

Over the augmentations translate, rotate, and crop, we are able to obtain a 90% reduction in augmentation size on MNIST, CIFAR10, and NORB using our policies.



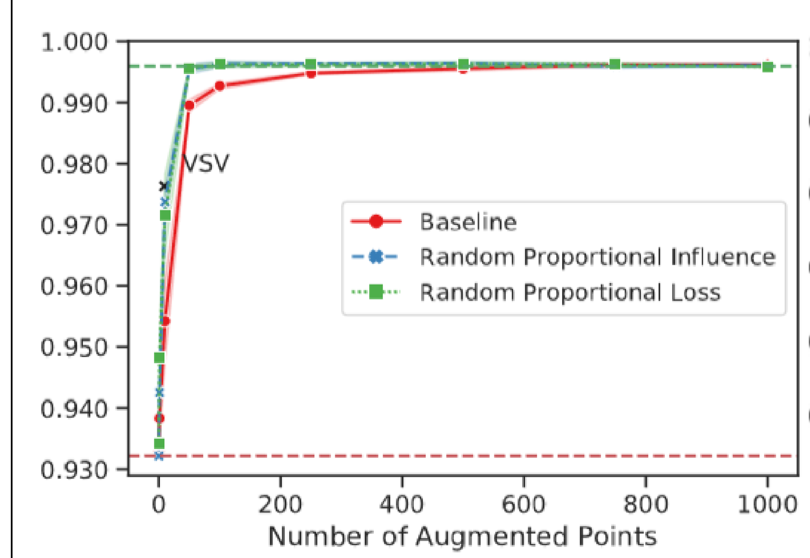
CIFAR10-rotate



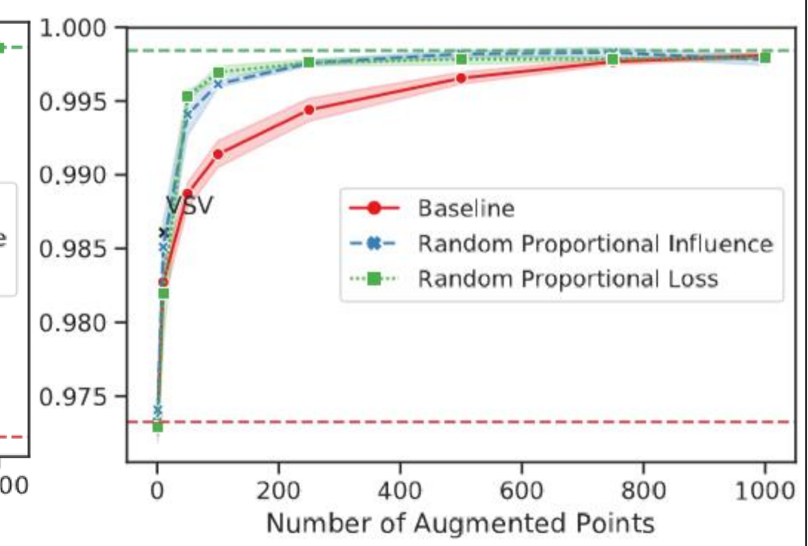
NORB-rotate

Policy Type	Selection Function	Update Scores	Downweight Points
Baseline	$P(z_i) = \frac{1}{n}$	X	X
Random Prop.	$P(z_i) = \frac{s_i}{\sum_j s_j}$	X	X
Deterministic Prop.	$\text{Rank}(z_i) = \texttt{SELECT}_S^{-1}(s_i)$	X	X
Random Prop. Update	$P(z_i) = \frac{s_i}{\sum_j s_j}$	✓	X
Rand. Prop. Downweight	$P(z_i) = \frac{s_i}{\sum_j s_j}$	X	✓

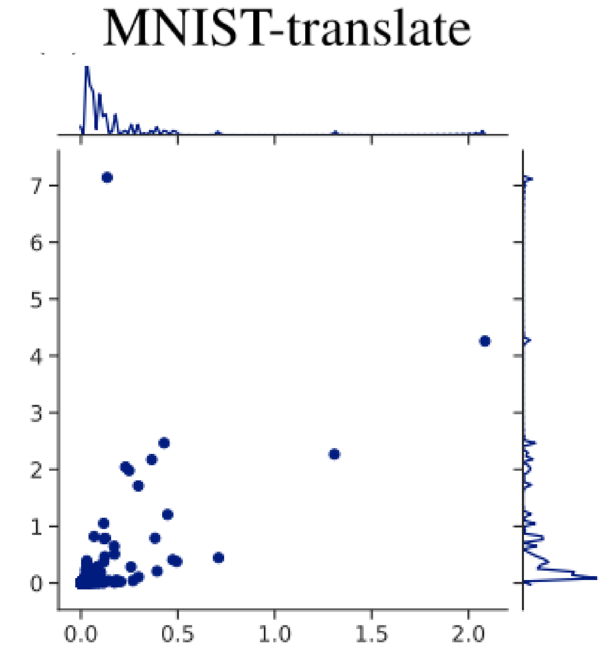
Perf(Loss) ≈ Perf(Influence) and Score Stability



MNIST-translate



MNIST-rotate



Influence before augmentation is correlated with influence after augmentation.

References

1. Christopher J. C. Burges and Bernhard Schölkopf. Improving the accuracy and speed of support vector machines. In Neural Information Processing Systems, 1997.
2. Dennis DeCoste and Bernhard Schölkopf. Training invariant support vector machines. Machine learning, 46(1-3):161–190, 2002.
3. Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In International Conference on Machine Learning, pp. 1885–1894, 2017.