SYNTHETIC DATA FOR MODEL SELECTION

Matan Fintz, Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, Gérard Medioni
Amazon
{fintz,alonshos,nadavb,kviat,medioni}@amazon.com

ABSTRACT

Recent improvements in synthetic data generation make it possible to produce images that are highly photorealistic and indistinguishable from real ones. Furthermore, synthetic generation pipelines have the potential to generate an unlimited number of images. The combination of high photorealism and scale turn the synthetic data into a promising candidate for potentially improving various machine learning (ML) pipelines. Thus far, a large body of research in this field has focused on using synthetic images for training, by augmenting and enlarging training data. In contrast to using synthetic data for training, in this work we explore whether synthetic data can be beneficial for model selection. Considering the task of image classification, we demonstrate that when data is scarce, synthetic data can be used to replace the held out validation set, thus allowing to train on a larger dataset.

1 INTRODUCTION

Traditionally, in supervised ML pipelines, the data used to train a model is divided into two sets, the *training set* and the *validation set*. The former is used to train various models, while the latter is used for ranking and selecting the best performing one, *i.e.*, the best architecture and hyper-parameters. Eventually, a final model with the selected configuration is trained on the entire data, including the training and the validation sets. Test data is inaccessible to the training pipeline, especially for model selection. The training-validation split provides means to estimate the models' error rate, which can be used for ranking. However, it is not helpful for selection of models that were trained on the entire data. As the optimal hyper-parameters depend on the number of training data samples and since models sharing the exact same hyper-parameters may exhibit large variance in accuracy, this may eventually lead to selecting a sub-optimal model.

Recent advances in the quality of synthetic data generation pipelines (Karras et al., 2019; 2020; Peng et al., 2018) have reduced the synthetic-to-real domain gap enough to successfully apply the generated data for training deep models (Besnier et al., 2020). Other works have focused on analyzing and quantifying various characteristics of the domain gap (Sajjadi et al., 2018; Kynkäänniemi et al., 2019). That said, to the best of our knowledge, the specific task of model selection with synthetic data was not addressed. When using synthetic data for training a model, one's goal is to minimize the generalization gap w.r.t. the real domain (Ben-David et al., 2010). Solving for generalization in presence of a synthetic-to-real domain gap is challenging. However, for model selection, one's goal is to use synthetic data for ranking a set of pre-trained models, while requiring rank preservation in the real domain. In this work, we introduce a sufficient condition for cross-domain rank preservation and empirically validate its value for model selection.

2 SYNTHETIC DATA FOR MODEL SELECTION

We follow notations similar to Ben-David et al. (2010). A domain is defined as a pair consisting of a distribution $\mathcal{D} = \langle \Omega, \mu \rangle$, where Ω is the sample domain and μ is the probability density function, and a labeling function $f : \Omega \to \mathcal{Y}$, where $\mathcal{Y}$ represents possible classes. We consider a particular pair of domains, where one is the original real domain, denoted by $\langle \mathcal{D}_r, f_r \rangle$, and the second one is synthetic, denoted by $\langle \mathcal{D}_s, f_s \rangle$, specifically tuned to mimic the real one. A hypothesis (model) is a function $h : \Omega \to \mathcal{Y}$. The risk or the probability that a hypothesis h disagrees with a labeling function f, according to the distribution $\mathcal{D}_r$ is defined as: $\epsilon_r(h, f) = E_{\mathbf{x} \sim \mathcal{D}_r}[h(\mathbf{x}) \neq f(\mathbf{x})]$. We neglect the difference between f_r and f_s and use the shorthand notation, $\epsilon_r(h) = \epsilon_r(h, f)$. Let
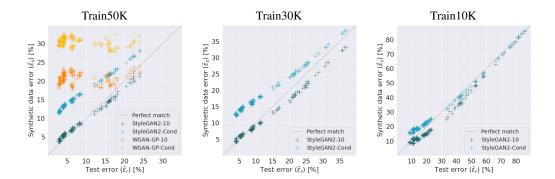
Figure 1: **Test vs. synthetic errors:** Each point represents the test error (x axis), $\hat{\epsilon}_r$, and synthetic data error (y axis), $\hat{\epsilon}_s$, of a single model. A total of 170 models were trained per CIFAR10 subset. Each model was evaluated by multiple synthetic datasets represented by different colors.

$\Delta\epsilon$ denote the risk difference between two hypotheses, $h_1, h_2 \in \mathcal{H}$, measured over a probabilistic distribution $\mathcal{D}$, i.e., $\Delta\epsilon = \epsilon(h_2) - \epsilon(h_1)$.

A common approach for model selection is the holdout method, where two datasets are sampled from $\langle \mathcal{D}_r, f_r \rangle$: the training set, $D_r^{trn} = \{\mathbf{x}_i, y_i\}_{i=1}^{N_{trn}}$ and the validation set, $D_r^{val} = \{\mathbf{x}_i, y_i\}_{i=1}^{N_{val}}$. A model (hypothesis) is trained using empirical risk minimization on D_r^{trn}. Thereafter, the model's risk is estimated using the validation set: $\hat{\epsilon}_r(h) = \frac{1}{N_{val}} \sum_{i=1}^{N_{val}} I(h(\mathbf{x}_i) \neq y_i)$. This allows to compare different models with different hyperparameters and to select those that minimize $\hat{\epsilon}_r(h)$. Other approaches such as cross-validation and bootstrap also exist (Kohavi et al., 1995). Since increasing the number of samples in the training set almost always increases the accuracy of the model, a common final step is to re-train the model, using the hyperparameters found in the previous step, on the entire dataset, $D_r^{trn} \cup D_r^{val}$. However, without a held-out dataset, it is no longer possible to compare models.

We propose to replace this two-step approach with a single step where we train a model on the entire dataset, then rather than estimating $\hat{\epsilon}_r(h)$, we instead generate a new dataset $D_s = \{\mathbf{x}_i, y_i\}_{i=1}^{N_s}$ via a generative model. Then we estimate the error $\hat{\epsilon}_s(h)$, where the domain $\langle D_s, f_s \rangle$ approximates the original domain $\langle D_r, f_r \rangle$. Although it may often be impossible to guarantee highly accurate error estimation due to the synthetic-real domain gap, below we present a corollary from Theorem A.2 (see Appendix A for statement and proof) outlining a sufficient condition for hypotheses' error rank preservation across domains.

Corollary 2.1. *Given the definitions above, let $\delta(\mu_r, \mu_s)$ denote the total variation between the two distributions, $\mathcal{D}_r, \mathcal{D}_s$. Then,*

$$\Delta\epsilon_s \geq \delta(\mu_r, \mu_s) \Rightarrow \Delta\epsilon_r \geq 0.$$

Informally, Corollary 2.1 indicates that if the total variation between the real and the synthetic distributions is not larger than the synthetic risk difference between a pair of hypotheses, then their error ranking is preserved across the domains.

3 EXPERIMENTS

All our experiments are performed on the CIFAR10 dataset (Krizhevsky et al., 2009). To evaluate the impact of the training set size, we use the following train-test splits: 10K-50K (Train10K), 30K-30K (Train30K), 50K-10K (Train50K). We emphasize that in all our experiments the following set of rules holds:

1. Only the training portion of the data is available for any training purposes.
2. The test portion of the images is never used for model selection and is treated as non-existent for any training purposes.
3. In each experiment, GANs for generating synthetic detests are trained only on the training portion of the images, *e.g.*, for experiments with the Train10K dataset, the GANs are trained only on the 10K images.
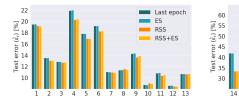
Figure 2: **ES and RSS for model selection (standard architectures)↓:** Test errors of the 17 architectures (x-axis corresponds to architectures described in Appendix E) trained on the Train10K dataset. "Last epoch" and ES show the average error of the 10 models for each architecture. RSS and RSS+ES show the results of the selected model out of the 10 models.

3.1 RANK PRESERVATION EXPERIMENTS

In our first experiment we focus on several commonly used deep model architectures. For each architecture, we select a number of variants. In total we experiment with 17 distinct architectures (see Appendix E for details). For each architecture, 10 models were trained on each of the three datasets. In Figure 1, we plot the empirical test errors, $\hat{\epsilon}_r$, vs. the empirical synthetic errors, $\hat{\epsilon}_s$, measured on datasets generated by four different GAN methods: (a) StyleGAN2-10, (b) StyleGAN2-Cond, (c) WGAN-GP-10, and (d) WGAN-GP-Cond (see details in Appendix F). It can be seen that, while in general $\hat{\epsilon}_r \neq \hat{\epsilon}_s$, for the StyleGAN2 based models, we are able to produce datasets that preserve the error ranking of different classification models. We measure this using Spearman's rank correlation coefficient. For different GANs, we have measured the following ranking coefficients: 0.97 (a), 0.98 (b), -0.19 (c) and 0.14 (d). In Sajjadi et al. (2018) the connection between total variation ($\delta(\mu_r, \mu_s)$) and precision and recall for distributions (PRD) was established, and an empirical method for estimating it was suggested. We use this method to empirically validate Corollary 2.1. For the GANs above, we measured the following values: 3.5% (a), 8.7% (b), 43% (c) and 34% (d). Indeed we see matching behaviors where the two models with high Spearman correlation have low total variation, and vice verse. For example, it follows from Corollary 2.1 that for GAN (a), if two hypotheses have $\Delta\epsilon_s(h_1, h_2) \geq 3.5\%$, then their rank on real data will be preserved.

3.2 MODEL SELECTION EXPERIMENTS

We consider three model selection scenarios where synthetic data can be used:

1. **Early stopping (ES):** Given a training schedule of a single model, select an epoch from which to take the model weights from.
2. **Random seed selection (RSS):** Given the same architecture and hyper-parameters, select a model instance out of N trained models where the difference between the models is the randomness of the training process, *e.g.*, weight initialization and dataset sampling order.
3. **Hyper-parameter search (HPS):** Select a model out of a set of models trained with different hyper-parameters. Possible hyper-parameters are training parameters, *e.g.*, learning rate, batch size or architecture parameters, *e.g.*, number of layers and network depth.

3.2.1 EARLY STOPPING AND RANDOM SEED SELECTION ON STANDARD ARCHITECTURES

We explore the impact of synthetic data on the ES and RSS model selection scenarios and their combination. We highlight that both of these scenarios require a held-out dataset. Therefore in the standard pipeline they cannot be used when training the model on the entire dataset. Using synthetic data enables these selection scenarios. For ES, the best synthetic epoch was selected for every training run. For RSS, per architecture, the model that performed the best on synthetic data at the last epoch was selected. For RSS + ES, per architecture, the model that performed the best on synthetic data at its best epoch was selected. We first experiment with the same standard model architectures as in Section 3.1. Figure 2 shows the results on Train10K, demonstrating that for nearly all architectures RSS improves accuracy. On the other hand, ES demonstrates only marginal impact on accuracy. This might be because the models' accuracy hardly changes across the last epochs, where the model has already converged (see Appendix D for convergence plot examples). In Appendix C we show the results for all datasets, where RSS shows comparable or better performance.

	Baseline	ES	RSS	ES + RSS
Train10K	19.38	19.36	**18.79**	18.88
Train30K	9.19	9.19	**9.09**	9.1
Train50K	7.09	7.1	7.02	**7.01**

Table 1: **ES and RSS for model selection (random wired networks)↓:** Average test error of (Baseline) all 640 models at the last epoch, (ES) all 640 models at the best synthetic set epoch, (RSS) 64 models at the best synthetic set epoch, where each of the 64 models was selected out of the 10 trained models for each architecture (by the best synthetic set error), (ES+RSS) 64 models at the best synthetic set epoch and selected by RSS.

	Synthetic protocol	Standard protocol	Avg of all models
Train10K	**$17.74_{\pm 0.28}$**	$18.06_{\pm 0.38}$	$19.39_{\pm 1.5}$
Train30K	**$8.64_{\pm 0.12}$**	$8.81_{\pm 0.17}$	$9.17_{\pm 0.45}$
Train50K	**$6.78_{\pm 0.19}$**	$6.85_{\pm 0.2}$	$7.09_{\pm 0.28}$

Table 2: **Architecture hyper-parameters search results↓:** Average error on held out test set. From left to right: 10 best models selected using synthetic data; 10 best architectures selected by real validation set and retrained on the entire training set; Average error of all trained models.

3.2.2 Early stopping and random seed selection on similar architectures

We next evaluate the impact of ES and RSS across multiple models with similar architectures. For each dataset we constructed 64 architectures. For generating the architectures we used randomly wired neural networks (RWNN) framework (Xie et al., 2019) with WS(2,0.25), resulting in 64 unique but similar architectures per dataset. Each architecture is trained 10 times on each dataset (a total of 1920 models were trained). Table 1 concludes the experiment. Since the errors of the models are of the same scale, we report the average performance. RSS has a significant impact on model selection with an average improvement over the baseline of 0.59/0.1/0.07 (corresponding to: Train10K, Train30K, Train50K). Similarly to the previous experiment ES has no significant impact.

3.2.3 Synthetic data for architecture hyper-parameter search

Next, we explore the contribution of using synthetic datasets for selecting a model out of multiple possible architectures and training instances (HPS). We consider three model selection protocols:

1. **Selecting a random model:** The naïve baseline of selecting a random model.
2. **Standard protocol:** Split the dataset into training and validation subsets. Then: (1) train each architecture N times with the training subset; (2) select the architecture that on average performed the best on the validation subset; (3) Train the selected architecture on the entire dataset. This methods allows for selecting a "promising" architecture without the ability to select a specific trained model instance (architecture and weights).
3. **Synthetic protocol:** (1) Train each architecture N times on the entire dataset; (2) evaluate the accuracy at each training epoch on the synthetic dataset; (3) select the model that preformed the best in step 2. This method allows for selecting a "promising" model instance.

We used the same 64 architectures per dataset, generated by the RWNN framework in Section 3.2.2. Table 2 shows that the synthetic protocol achieves, on average, better results compared to the standard protocol. Both protocols perform far better then selecting a model at random. It can also be observed that the impact of using synthetic data is greater when the training dataset is smaller. This result is aligned with Corollary 2.1, as $\Delta\epsilon_s$ is larger. Appendix B presents an extensive breakdown of the experiment. A key finding from the breakdown is that ranking models using synthetic data is comparable to ranking models using validation data in terms of Spearman correlation to a test set.

4 Conclusion

In this paper we presented a comprehensive empirical study on the CIFAR10 dataset for evaluating the impact of using synthetic data for model selection. The empirical evidence suggest that evaluating trained models on synthetic data can be beneficial and outperform the standard methods for model selection that are based solely on the available real images.

REFERENCES

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A Theory of Learning From Different Domains. *Machine learning*, 79(1):151–175, 2010.

Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2020.

Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, 2017.

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. *IEEE CVPR*, 2017.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016b.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NIPS*, 2017.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. *CoRR*, abs/1912.04958, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. *arXiv preprint arXiv:2006.06676*, 2020.

Ron Kohavi et al. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Ijcai*, volume 14, pp. 1137–1145. Montreal, Canada, 1995.

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

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019.

Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A Synthetic-to-Real Benchmark for Visual Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2021–2026, 2018.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring Randomly Wired Neural Networks for Image Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

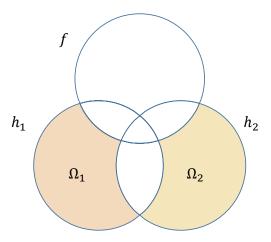Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Figure 3: **Ω decomposition:** Venn diagram illustrating the decomposition of the sample domain into various sections.

A   THEORETICAL JUSTIFICATION

In this section we derive the theoretical justification leading up to Corollary 2.1.

Lemma A.1. *Let $\Delta\epsilon$ denote the risk difference between two hypotheses, $h_1, h_2 \in \mathcal{H}$, measured over a probability distribution $\mathcal{D} = \langle \Omega, \mu \rangle$, i.e., $\Delta\epsilon = \epsilon(h_2) - \epsilon(h_1)$. Let f denote the labeling function. Let $\Omega_1 = \{\mathbf{x} \in \Omega | h_1(\mathbf{x}) \neq h_2(\mathbf{x}) \wedge h_1(\mathbf{x}) \neq f(\mathbf{x})\}$ and $\Omega_2 = \{\mathbf{x} \in \Omega | h_1(\mathbf{x}) \neq h_2(\mathbf{x}) \wedge h_2(\mathbf{x}) \neq f(\mathbf{x})\}$. Then,*

$$\Delta\epsilon = \int_{\Omega_2} \mu(\mathbf{x})d\mathbf{x} - \int_{\Omega_1} \mu(\mathbf{x})d\mathbf{x}.$$

Proof.

$$
\begin{aligned}
\Delta\epsilon &= \epsilon(h_2) - \epsilon(h_1) \\
&= E_{\mathbf{x}\sim\mathcal{D}}[h_2(\mathbf{x}) \neq f(\mathbf{x})] - E_{\mathbf{x}\sim\mathcal{D}}[h_1(\mathbf{x}) \neq f(\mathbf{x})] \\
&= E_{\mathbf{x}\sim\mathcal{D}}[h_2(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) \neq h_1(\mathbf{x})] + E_{\mathbf{x}\sim\mathcal{D}}[h_2(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) = h_1(\mathbf{x})] \\
&\quad - E_{\mathbf{x}\sim\mathcal{D}}[h_1(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) \neq h_1(\mathbf{x})] - E_{\mathbf{x}\sim\mathcal{D}}[h_1(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) = h_1(\mathbf{x})] \\
&= E_{\mathbf{x}\sim\mathcal{D}}[h_2(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) \neq h_1(\mathbf{x})] - E_{\mathbf{x}\sim\mathcal{D}}[h_1(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) \neq h_1(\mathbf{x})] \\
&= \int_{\Omega_2} \mu(\mathbf{x})d\mathbf{x} - \int_{\Omega_1} \mu(\mathbf{x})d\mathbf{x}
\end{aligned}
$$

$\square$

Theorem A.2. *Let $\Delta\epsilon_r$ and $\Delta\epsilon_s$ denote the risk difference between two hypotheses, $h_1, h_2 \in \mathcal{H}$, measured over the real and the synthetic probability distributions $\mathcal{D}_r = (\Omega, \mu_r)$ and $\mathcal{D}_s = (\Omega, \mu_s)$, respectively. I.e., $\Delta\epsilon_r = \epsilon_r(h_2) - \epsilon_r(h_1)$ and $\Delta\epsilon_s = \epsilon_s(h_2) - \epsilon_s(h_1)$. Let f denote the labeling function. Let $\delta_{h_1 \oplus h_2 \setminus f}(\mu_r, \mu_s)$ denote the total variation between the two distributions in the sample subspace of Ω in which the two hypotheses disagree with $f(x)$ and between themselves. If $\Delta\epsilon_s \geq \delta_{h_1 \oplus h_2 \setminus f}(\mu_r, \mu_s)$, then $\Delta\epsilon_r \geq 0$.*

7

Proof.

$$\Delta\epsilon_r = \int_{\Omega_2} \mu_r(\mathbf{x})d\mathbf{x} - \int_{\Omega_1} \mu_r(\mathbf{x})d\mathbf{x}$$

$$= \Delta\epsilon_s + \left(\int_{\Omega_2} \mu_r(\mathbf{x})d\mathbf{x} - \int_{\Omega_1} \mu_r(\mathbf{x})d\mathbf{x} - \Delta\epsilon_s\right)$$

$$= \Delta\epsilon_s + \left(\int_{\Omega_2} \mu_r(\mathbf{x})d\mathbf{x} - \int_{\Omega_1} \mu_r(\mathbf{x}) - \int_{\Omega_2} \mu_s(\mathbf{x})d\mathbf{x} + \int_{\Omega_1} \mu_s(\mathbf{x})d\mathbf{x}\right)$$

$$= \Delta\epsilon_s + \left(\int_{\Omega_2} \mu_r(\mathbf{x}) - \mu_s(\mathbf{x})d\mathbf{x} - \int_{\Omega_1} \mu_r(\mathbf{x}) - \mu_s(\mathbf{x})d\mathbf{x}\right)$$

$$\geq \Delta\epsilon_s - \left(\int_{\Omega_2} |\mu_r(\mathbf{x}) - \mu_s(\mathbf{x})|d\mathbf{x} + \int_{\Omega_1} |\mu_r(\mathbf{x}) - \mu_s(\mathbf{x})|d\mathbf{x}\right)$$

$$= \Delta\epsilon_s - \int_{\Omega_1 \cup \Omega_2} |\mu_r(\mathbf{x}) - \mu_s(\mathbf{x})|d\mathbf{x}$$

$$= \Delta\epsilon_s - \delta_{h_1 \oplus h_2 \setminus f}(\mu_r, \mu_s)$$

$\square$

Corollary A.3. *Let $\mathcal{D}_r = \langle \Omega, \mu_r \rangle$ and $\mathcal{D}_s = \langle \Omega, \mu_s \rangle$, denote the real and the synthetic (generated) probabilistic distributions, respectively. Let $\Delta\epsilon_r$ and $\Delta\epsilon_s$ denote the risk difference between any two hypotheses, $h_1, h_2 \in \mathcal{H}$. Then,*

$$\Delta\epsilon_s - \Delta\epsilon_r \leq \delta(\mu_r, \mu_s),$$

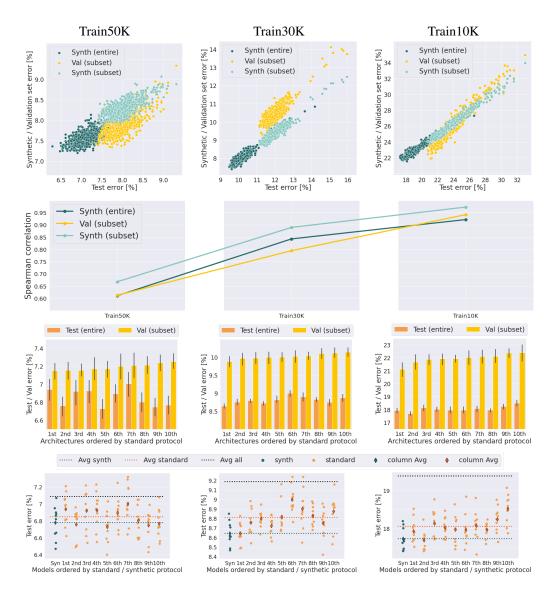*where $\delta(\mu_r, \mu_s)$ denotes the total variation between the two distributions.*

Figure 4: **Synthetic data for architecture hyper-parameter search: (1st row)** Each point represents the test error (x-axis) and validation/synthetic set error (y-axis) of a specific model. "Entire" corresponds to models that were trained on the entire dataset (Train50K, Train30K, Train10K). "Subset" corresponds to models that were trained on the training subset. **(2nd row)** Spearman correlation between the validation/synthetic set errors and the test set errors. **(3rd row)** Yellow bars represent the 10 architectures that performed the best (average of 10 training runs) on the validation set (ranked from the best to 10th best). Orange bars represent the same architectures, trained on the entire dataset and there performance (average of 10 training runs) on the test set. Black lines represents the 95% confidence interval. **(4th row)** The points in the first column, "Syn", correspond to the test errors of the 10 best performing models selected using the synthetic protocol. The rest of the columns (1st-10th) correspond to the test error results of all trained models out of the 10 best architectures selected by the standard protocol (same architectures as row 3). Horizontal lines represent average test error rates of: 10 best synthetic models (Avg synth), average of the 10 best models selected by the standard protocol (Avg standard), and the average of all 640 models (Avg all).

B SYNTHETIC DATA FOR ARCHITECTURE HYPER-PARAMETER SEARCH

In this experiment we explore the contribution of synthetic data for model selection out of a pool of different architectures. Given a training set and a held-out test set (not available for model selection),

we compared the three model selection protocols (selecting a random network, standard protocol, synthetic protocol). The standard protocol requires a validation set, to this end we split each of the datasets (Train50K, Train30K, Train10K) into training and validation subsets. For the synthetic protocol a GAN was trained on each dataset to produce a dataset of 100K synthetic images (see Appendix F):

1. **Train10K:** The train/val split is 7.5K/2.5K. For the synthetic data protocol, a single StyleGAN2-Cond model trained on the 10K available images was used.

2. **Train30K:** The train/val split is 22.5K/7.5K. For the synthetic data protocol, 10 Style-GAN2 models were trained, each on the 3K (per class) available images.

3. **Train50K:** The train/val split is 40K/10K. For the synthetic data protocol, 10 StyleGAN2 models were trained, each on the 5K (per class) available images.

In each experiment 64 architectures were evaluated using the different protocols. For generating similar architectures, we sampled RWNN architectures with the same parameters WS(2, 0.25) (same architectures as in 3.2.2). In both the standard and synthetic protocols we trained each architecture 10 times. For the standard protocol we train on the training subset (step 1) and for the synthetic protocol we use the entire dataset. Note that for the standard protocol, it is not possible to select a model instance out of the 10 trained instances of each architecture that was trained on the entire data (step 3). Therefore, we use the average test error of the 10 trained models of each architecture (on the entire dataset) as a data point for comparisons.

Figure 4 shows different analyses of the experimental results. From the first two rows of the figure we can infer that there is a strong correlation between the error on synthetic data, $\hat{\epsilon}_s$, and the error on the test set, $\hat{\epsilon}_r$. We evaluate this correlation using Spearman's rank correlation coefficient, as it is appropriate for measuring rank preservation. From the Spearman correlation plot (second row) we learn that the ranking capability of the synthetic data is comparable to that of the real data validation set. This strengthens our premise that using synthetic data for model selection is appropriate. It can be seen that the correlation improves when the training set is smaller and the errors are larger. This result coincides with Corollary 2.1 where for larger gaps in synthetic error, there is a lower chance for a flip in model ranking. From the third row we can infer that the ranking of architectures might change when moving from training on a smaller training set and evaluating on a validation set to training on the entire dataset end evaluating on the test set. This implies that a potential gain in accuracy could be achieved by selecting a model out of the models that were directly trained on the entire dataset. From the last row we can learn that, on average, model selection using synthetic data improves over the standard method. Again, the impact of synthetic data increases as the training dataset size decreases. Given that a synthetic dataset is available, training the models directly on the entire dataset is simpler than training on a subset and re-training on the entire dataset.

C   ADDITIONAL RESULTS FOR EARLY STOPPING AND RANDOM SEED SELECTION ON STANDARD ARCHITECTURES

In addition to the results reported in 3.2.1, Figure 5 shows results of ES, RSS and RSS+ES on all three datasets. RSS is beneficial for model selection in most cases, however the benefits decrease as the dataset size increases.
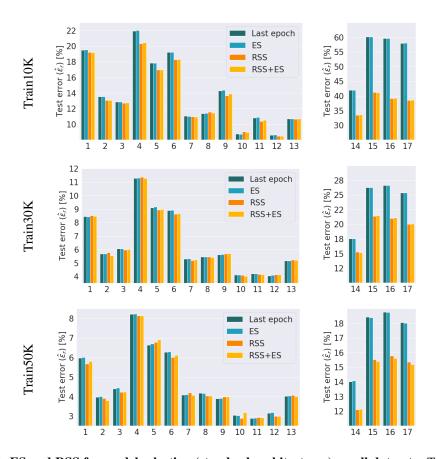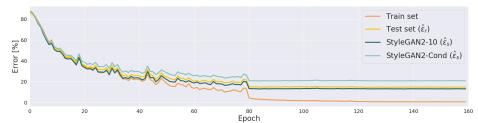

Figure 5: **ES and RSS for model selection (standard architectures) on all datasets:** Test errors of 17 architectures trained on each of the datasets. "Last epoch" and ES show the average error of the 10 models for each architecture. RSS and RSS+ES show the results of the selected model out of the 10 models.

D STANDARD ARCHITECTURE CONVERGENCE IN TRAINING

Figure 6 shows two examples of the train, test and synthetic data errors vs. epoch index during training on the Train50K dataset. It can be observed that although the synthetic data error does not match the test error exactly, it follows the same trend as the test error. In the last epochs of training, where the learning rate has decreased there is very little change in the model's error. This may explain why the early stopping experiments did not demonstrate any benefits.

ResNet110 (Architecture 17 in Appendix E)

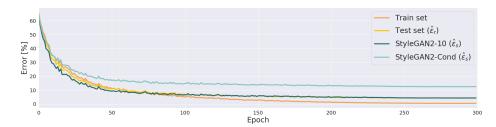Shake-Shake 64 + cutout (Architecture 10 in Appendix E)

Figure 6: **Train, test and synthetic data error vs. epoch**

E   STANDARD ARCHITECTURES DESCRIPTION

Below is the list of architectures used in Sections 3.1 and 3.2.1:

1. **DenseNet :** (Huang et al., 2017; 2019) with batch size 32, initial learning rate 0.05, depth 100, block type "bottleneck", growth rate 12, compression rate 0.5.

2. **PyramidNet 270:** (Han et al., 2017) with depth 110, block type "basic", $\alpha = 270$.

3. **PyramidNet 84:** (Han et al., 2017) with depth 110, block type "basic", $\alpha = 84$.

4. **SE-ResNet-preact:** (Hu et al., 2019) with depth 110, se reduction=16.

5. **ResNet-preact 110:** (He et al., 2016b) with depth 110, block type "basic".

6. **ResNet-preact 164:** (He et al., 2016b) with depth 164, block type "bottleneck".

7. **ResNext 4x64d:** (Xie et al., 2016) with depth 29, cardinality 4, base channels 64, batch size 32 and initial learning rate 0.025.

8. **ResNext 8x64d:** (Xie et al., 2016) with depth 29, cardinality 8, base channels 64, batch size 64 and initial learning rate 0.05.

9. **Shake-shake 32d:** (Gastaldi, 2017) with depth 26, base channels 32, S-S-I model.

10. **Shake-shake 64d:** (Gastaldi, 2017) with depth 26, base channels 64, S-S-I model, batch size 64, base $lr = 0.1$.

11. **Shake-shake 64d + cutout:** (Gastaldi, 2017) with depth 26, base channels 64, S-S-I model, batch size 64, $lr = 0.1$, cosine scheduler, cutout (DeVries & Taylor, 2017) size 16.

12. **Wide residual network + cutout:** (Zagoruyko & Komodakis, 2016) with depth 28, widening factor 10, base $lr = 0.1$, batch size 64, cosine scheduler, cutout (DeVries & Taylor, 2017) size 16.

13. **Wide residual network:** (Zagoruyko & Komodakis, 2016) with depth 28, widening factor 10.

14. **ResNet 32:** (He et al., 2016a) with depth 32, block type "basic".

15. **ResNet 44:** (He et al., 2016a) with depth 44, block type "basic".

16. **ResNet 56:** (He et al., 2016a) with depth 56, block type "basic".

17. **ResNet 110:** (He et al., 2016a) with depth 110, block type "basic".

F   SYNTHETIC DATA GENERATION DETAILS

Our method for producing synthetic datasets is based on training GANs that in turn are used to generate the desired labeled data. We consider two GAN frameworks for generating our synthetic datasets:

1. StyleGAN2 (Karras et al., 2019) with non-leaking augmentation (Karras et al., 2020). This framework is our best candidate for generating high quality synthetic datasets since it is the SOTA for generating CIFAR10 images.

2. WGAN-GP (Gulrajani et al., 2017). This framework generates lower quality images than StyleGAN2. We consider it as a baseline to explore how the image quality impacts the datasets models selection capabilities.

For each GAN framework we consider two variants of training the GANs to generate labeled datasets:

1. **Training 10 GANs (StyleGAN2-10/WGAN-GP-10):** For each of the 10 CIFAR10 classes, a different GAN was trained with just one class at a time (*e.g.*, 5K images for Train50K, 3K images for Train30K and 1K images for Train10K). The generator instance with the best FID (Heusel et al., 2017) score out of all instances obtained during training was selected to generate 10K images of its corresponding class.

2. **Training one Conditional GAN (StyleGAN2-Cond/WGAN-GP-Cond):** A single Conditional-GAN was trained, and best instance selected by FID score. Thereafter, 10K images were generated per class.

Using the above methods we constructed 8 datasets (each with 100K labeled images): three "StyleGAN2-10" datasets and three "StyleGAN2-Cond" datasets (one per CIFAR10 subset), one "WGAN-GP-10" dataset and one "WGAN-GP-Cond" dataset (for the Train50K CIFAR10 subset).

Table 3 shows the FID scores breakdown for our synthetic datasets. As expected, as the training dataset size decreases the FID score increases.

Figures 7 and 8 show samples of real CIFAR10 images and our synthetic StyleGAN2-based datasets for each of the CIAFR10 classes.

Synth dataset	Class	Train50K	Train30K	Train10K
	0	10.11	17.06	44.15
	1	6.05	9.41	29.91
	2	10.65	16.39	49.79
	3	12.04	18.58	56.67
	4	7.94	12.5	35.76
StyleGAN2-10	5	11.23	16.98	51.15
	6	8.36	13.22	39.84
	7	8.41	12.91	31.57
	8	7.59	11.02	32.2
	9	6.25	10.26	28.33
	All	4.4	4.86	14.15
StyleGAN2-Cond	All	4.4	6.25	11.72
WGAN-GP-10	All	35.7	N/A	N/A
WGAN-GP-Cond	All	27.3	N/A	N/A

Table 3: **FID scores (↓) breakdown**

Figure 7: **Real images vs. StyleGAN2-10 datasets**

Figure 8: **Real images vs. StyleGAN2-Cond datasets**