Synthetic Data for Model Selection

Alon Shoshan [1] **Nadav Bhonker** [1] **Igor Kviatkovsky** [1] **Matan Fintz** [1] **Gérard Medioni** [1]

Abstract

Recent breakthroughs in synthetic data generation approaches made it possible to produce highly photorealistic images which are hardly distinguishable 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 synthetic data into a promising candidate for 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. We also introduce a novel method to calibrate the synthetic error estimation to fit that of the real domain. We show that such calibration significantly improves the usefulness of synthetic data for model selection.

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.

In this work we propose to substitute the held out validation set with synthetic data, allowing for model selection even when training on the entire dataset. The synthetic validation set is created using generative models trained on the immediately available data, and without reliance on external knowledge or tools (*e.g.*, additional data sources, 3D rendering engines). This makes our approach self-sufficient and applicable to a wide range of problems.

Recent advances in the quality of synthetic data generation pipelines (Karras et al., 2019b; 2020; 2021a; Peng et al., 2018; Dhariwal & Nichol, 2021; Saharia et al., 2022) have reduced the synthetic-to-real domain gap enough to successfully utilize 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 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. We perform extensive experiments on the CIFAR10 (Krizhevsky et al., 2009) dataset showing that indeed we are able to improve overall accuracy by selecting better models using synthetic data. Furthermore, we introduce a novel calibration approach to improve the ranking capabilities of synthetic data on rich visual domains such as ImageNet (Deng et al., 2009).

We summarize our contributions as following:

- We show that the error rank of models evaluated on

[1]Amazon. Correspondence to: Alon Shoshan <alonshos@amazon.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

synthetic data mostly preserves the rank of their performance with respect to a held out test set.

- We demonstrate the value of this observation by allowing to train on the entire dataset and to select the best model using synthetic data rather than a held-out validation set. We show that this method, on average, selects a better model.
- We introduce a novel calibration approach to improve the ranking capabilities of synthetic datasets by re-weighing its samples. Such re-weighting enables a high level of ranking in challenging visual domains, even when high-quality synthetic generators are not available.
- We provide a sufficient condition for which the rank preservation will be maintained across domains.

2. Related Work

Model selection is a challenging task traditionally done by comparing the model estimation errors via cross-validation. This venture is expensive as it requires to train each model a number of times. Multiple methods exist to improve the efficiency of cross-validation (Liu et al., 2018; Ghosh et al., 2020; Wilson et al., 2020). These are limited to specific data types or models and were not demonstrated on complex tasks such as image classification. A common alternative approach is to use a single train-validation split. In order to leverage all the available data, both approaches require to retrain the selected model again on the entire dataset and do not allow to select a specific trained model. Furthermore, additional training data may change the optimal model hyperparameters. Other methods exist in which a validation set is not required altogether. To avoid using a held out validation set, Corneanu et al. (2020) employed persistent homology to estimate the performance gap between training and testing error without a test dataset. While this method can generalize well across datasets, it does not allow to compare classifiers of different architectures. In Neyshabur et al. (2017) different measures based on norms of weight matrices are proposed for quantifying and guaranteeing generalization in deep models. Li et al. (2020) uses the training set with augmentations to be used instead of a validation set. This method was only demonstrated on simple classification datasets using simple non deep learning classifiers.

In the recent years generative models have improved significantly and are at the point where state-of-the-art generative models, such as GANs (Karras et al., 2019b;a; 2017; 2021b; Esser et al., 2021) and diffusion models (Dhariwal & Nichol, 2021; Saharia et al., 2022; Ho et al., 2020; Song et al., 2020; Rombach et al., 2022), are able to produce high-resolution synthetic images that are indistinguishable from real images. The potential of producing unlimited amount of training data using generative models has prompted the exploration of using synthesized data to augment downstream tasks.

Although promising, the use of synthetic data for training is not trivial. Ravuri & Vinyals (2019) show that training a classifier using only synthetic data results in sub-optimal accuracy in both CIFAR10 and ImageNet datasets. Eilertsen et al. (2021) address the problem of training with synthetic data by using an ensemble of GANs rather than a single one.

3. 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 $\Delta\epsilon$ 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 sufficient condition for hypotheses' error rank preservation

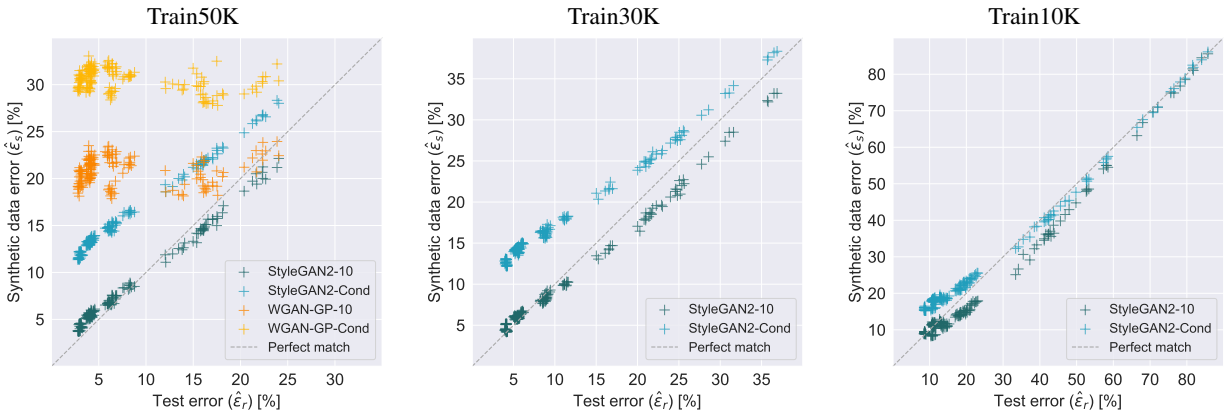


Figure 1: **Synthetic vs. test errors:** Each point represents the synthetic data error (y axis), $\hat{\epsilon}_s$, and test error (x axis), $\hat{\epsilon}_r$, of a single model. A total of 170 models were trained per CIFAR10 subset. Each model was evaluated by multiple synthetic datasets represented by different colors.

across domains.

Lemma 3.1. *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 $\Omega_1 = \{\mathbf{x} \in \Omega | h_1(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) = f(\mathbf{x})\}$ and $\Omega_2 = \{\mathbf{x} \in \Omega | h_2(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_1(\mathbf{x}) = f(\mathbf{x})\}$. Then,*

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

Proof is provided in Appendix B. Informally, Lemma 3.1 states that the error gap between two hypotheses depends only on the area where they disagree.

Theorem 3.2. *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. Then, for any $h_1, h_2 \in \mathcal{H}$:*

$$\Delta\epsilon_s - \Delta\epsilon_r \leq \delta_{h_1 \oplus h_2}(\mu_r, \mu_s),$$

where $\delta_{h_1 \oplus h_2}$ is the total variation computed over the subset of the domain Ω, where the hypotheses h_1 and h_2 do not agree.

Proof.

$$\begin{aligned}
\Delta\epsilon_s - \Delta\epsilon_r &= \int_{\Omega_2} \mu_s(\mathbf{x}) d\mathbf{x} - \int_{\Omega_1} \mu_s(\mathbf{x}) d\mathbf{x} \\
&\quad - \int_{\Omega_2} \mu_r(\mathbf{x}) d\mathbf{x} + \int_{\Omega_1} \mu_r(\mathbf{x}) d\mathbf{x} \\
&= \int_{\Omega_2} \mu_s(\mathbf{x}) - \mu_r(\mathbf{x}) d\mathbf{x} - \int_{\Omega_1} \mu_s(\mathbf{x}) - \mu_r(\mathbf{x}) d\mathbf{x} \\
&\leq \int_{\Omega_2} |\mu_s(\mathbf{x}) - \mu_r(\mathbf{x})| d\mathbf{x} + \int_{\Omega_1} |\mu_s(\mathbf{x}) - \mu_r(\mathbf{x})| d\mathbf{x} \\
&= \int_{\Omega_1 \cup \Omega_2} |\mu_s(\mathbf{x}) - \mu_r(\mathbf{x})| d\mathbf{x} \leq \delta_{h_1 \oplus h_2}(\mu_r, \mu_s)
\end{aligned}$$

□

The last line follows from the fact that $\Omega_1 \cap \Omega_2 = \emptyset$. Theorem 3.2 provides a condition for error rank preservation between two hypotheses. In order to reach a bound that is true for any two hypotheses, we upper bound it by the total variation.

Corollary 3.3. *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 3.3 indicates that if the total variation between the real and synthetic distributions is not larger than the synthetic risk difference between a pair of hypotheses, then their error ranking is preserved across domains.

We note that the total variation bound is quite loose. Theoretically, a tighter bound can be achieved following an approach presented in Ben-David et al. (2010). We present this connection in Appendix A. However, we are not aware of any practical estimation method to estimate this divergence. Therefore, we resolve to measuring the total variation since it has a practical measurement method.

4. Experiments

In Sections 4.1 and 4.2, we perform experiments on CIFAR10 (Krizhevsky et al., 2009). In these sections, 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 these experiments the following set of rules hold:

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 datasets 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.

In Section 4.3 we demonstrate rank preservation on ImageNet (Deng et al., 2009) and in Section 4.4 we introduce a novel calibration method to improve the ranking.

4.1. Rank Preservation

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 F 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 G). We can observe 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 3.3. 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 versa. For example, it follows from Corollary 3.3 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.

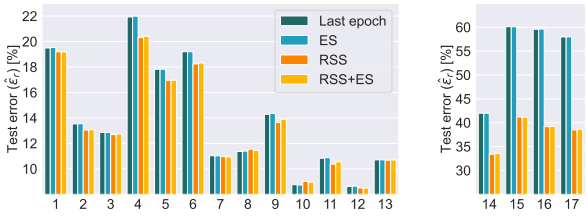


Figure 2: **ES and RSS for model selection (standard architectures)↓:** Test errors of the 17 architectures (x-axis corresponds to architectures described in Appendix F) trained on the Train10K dataset. "Last epoch" and ES show the average error of the 10 models for each architecture. RSS and RSS+ES show the results of the selected model out of the 10 models.

Table 1: **ES and RSS for model selection (randomly wired networks)↓:** Average test error ± 95% confidence intervals and standard deviation in parentheses for several model selection scenarios. 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. 10K, 30K and 50K refer to the Train10K, Train30K and Train50K datasets respectively.

	Baseline	ES	RSS	ES + RSS
10K	19.38±0.12 (1.50)	19.36±0.12 (1.50)	**18.79±0.29 (1.17)**	18.88±0.28 (1.15)
30K	9.19±0.03 (0.45)	9.19±0.03 (0.44)	**9.09±0.09 (0.37)**	9.1±0.10 (0.39)
50K	7.09±0.02 (0.28)	7.1±0.02 (0.27)	7.02±0.08 (0.31)	**7.01±0.08 (0.30)**

4.2. Model Selection

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: learning rate, batch size, number of layers and network depth.

4.2.1. 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 one is able to utilize these model 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 4.1. 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 E for convergence plot examples). In Appendix D we show the results for all datasets, where RSS shows comparable or better performance.

4.2.2. Early stopping and random seed selection on similar architectures

Next, we 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 and 95% confidence intervals. 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.

4.2.3. 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.

Table 2: **Architecture hyper-parameters search results↓:** Average test error ± 95% confidence intervals and standard deviation in parentheses 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.

	Synthetic	Standard	All models
Train10K	**17.74±0.20** **(0.28)**	18.06±0.08 (0.38)	19.39±0.12 (1.50)
Train30K	**8.64±0.09** **(0.12)**	8.81±.03 (0.17)	9.17±.03 (0.45)
Train50K	**6.78±0.14** **(0.19)**	6.85±0.02 (0.20)	7.09±0.02 (0.28)

Given a training set and a held-out test set (not available for model selection), we compared the three model selection protocols. The standard protocol requires a validation set, to this end we split each of the datasets into training and validation subsets (train/val): Train10K was split into 7.5K/2.5K, Train30K was split into 22.5K/7.5K and Train50K was split into 40K/10K (see Appendix C for more details). 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 4.2.2). 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 3 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

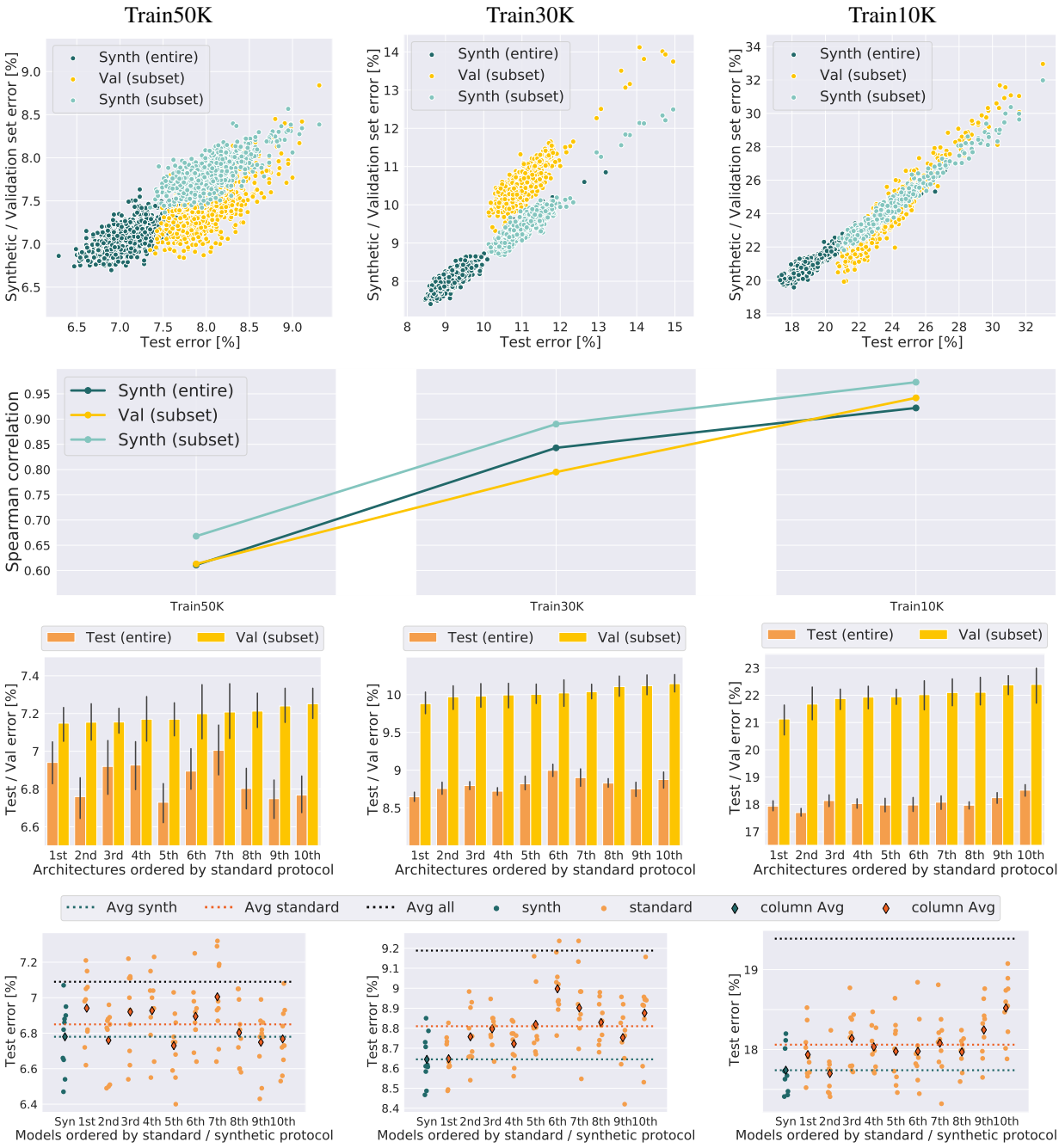


Figure 3: **Synthetic data for architecture hyper-parameter search: (1st row)** Each point represents the validation/synthetic set error (y-axis) and test error (x-axis) of a specific model. "Entire" corresponds to models that were trained on the entire dataset (Train50K, Train30K, Train10K). "Subset" 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 their 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).

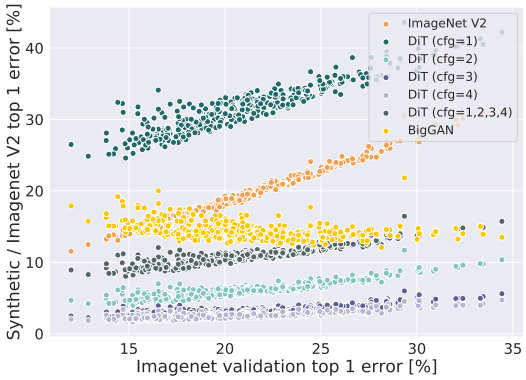


Figure 4: **Synthetic vs. ImageNet errors:** Each point represents the synthetic data error (y axis), $\hat{\epsilon}_s$ and ImageNet validation error (x axis), $\hat{\epsilon}_r$, of a single model. A total of 390 models trained on ImageNet were used. Each model was evaluated by multiple synthetic datasets represented by different colors. Additionally, we evaluated the models on real ImageNet V2 images as a baseline (orange points).

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 3.3 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. Table 2 summarizes the experiments results and shows that the synthetic protocol achieves, on average, better results compared to the standard protocol.

4.3. Rank Preservation on ImageNet

In this section we explore the potential of synthetic data to be used for model selection in a more challenging domain. To this end we choose the ImageNet (Deng et al., 2009) 1,000 classification task. The main differences between CIFAR10 and ImageNet is that the former has 5,000 low-res images for each of the 10 classes, while the latter has 732–1,300, high resolution images for each of its 1,000 classes.

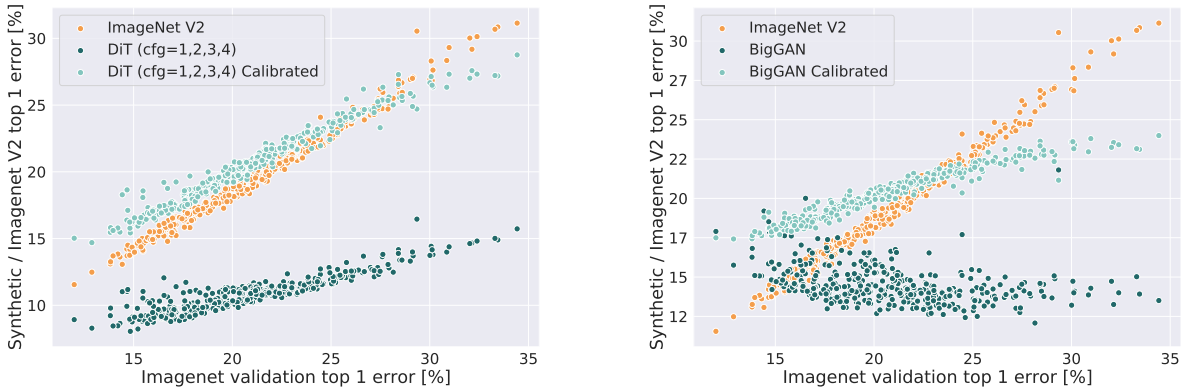
We use BigGAN (Brock et al., 2019) and DiT (Peebles & Xie, 2022) as our synthetic image generation models. These models were selected for two reasons: both are conditioned on the 1,000 ImageNet classes and do not use a pre-trained classifier as conditional guidance. Additionally, the generation code of both models was publicly released by the authors. We have generated the following six synthetic datasets: a dataset containing 100 images per class, generated by BigGAN with truncation $= 0.7$; four datasets, each containing 100 images per class, generated by DiT with classifier free guidance (Ho & Salimans, 2021) of $1, 2, 3$ or 4; and a dataset that is the combination of the four previous DiT datasets (cfg $= 1, 2, 3, 4$). In our experiments we used ImageNetV2 (Recht et al., 2019) as a baseline, which is a

Table 3: **Ranking models trained on ImageNet using synthetic data↑:** Top1 and Top 5 columns show the Spearman correlation between the *top 1* error of 390 evaluation models on each dataset to the *top 1* and *top 5* error of the same models on the ImageNet validation set. The three first rows show the correlation on variants of ImageNetV2. Rows 4-9 show the correlation on synthetic datasets. The last six rows show the correlation on calibrated synthetic datasets.

Dataset	Calibrated	Top 1	Top 5
ImageNetV2 (format a)		0.994	0.994
ImageNetV2 (format b)		0.996	0.994
ImageNetV2 (format c)		0.994	0.995
BigGAN		−0.363	−0.357
DiT (cfg $= 1$)		0.907	0.900
DiT (cfg $= 2$)		0.874	0.862
DiT (cfg $= 3$)		0.819	0.801
DiT (cfg $= 4$)		0.788	0.768
DiT (cfg $= 1, 2, 3, 4$)		0.893	0.880
BigGAN	✓	0.978	0.970
DiT (cfg $= 1$)	✓	0.984	0.977
DiT (cfg $= 2$)	✓	0.980	0.971
DiT (cfg $= 3$)	✓	0.974	0.963
DiT (cfg $= 4$)	✓	0.966	0.954
DiT (cfg $= 1, 2, 3, 4$)	✓	**0.986**	**0.978**

Table 4: **Spearman correlation vs. number of calibration models (M)↑.**

	M $= 5$	M $= 10$	M $= 30$	M $= 50$
BigGAN	0.931	0.940	0.973	0.978
DiT (cfg=1,2,3,4)	0.965	0.966	0.979	0.986



(a) DiT (cfg $= 1, 2, 3, 4$) (b) BigGAN

Figure 5: **Synthetic data calibration:** We demonstrate improved rank preservation on 390 held-out models after the calibration process, which utilized the remaining 50 models. (a) shows results for DiT, (b) shows the results for BigGAN.

test set attempting to mimic the distribution of ImageNet. For rank preservation analysis, we utilized all the models from TIMM (Wightman, 2019) and pretrained-models[1] that were trained on images of 224x224 resolution. This lead to a total of 440 models, that were divided randomly into 390 models for rank preservation analysis and 50 models that were set aside for the calibration process (to be explained in Section 4.4).

Both Figure 4 and Table 3 demonstrate that rank preservation can be achieved to some extent using images generated by DiT (Spearman correlation of 0.893). However, the best synthetic dataset still has a gap in rank correlation compared to the real ImageNetV2 dataset, which has a Spearman correlation of 0.996. The dataset generated by BigGAN is irrelevant for model ranking (-0.363). We suspect that the weak ranking capability of BigGAN compared to DiT, is due to the gap in image quality and image variability (we present generated samples in Appendix H).

4.4. Synthetic Dataset Calibration

In this section we describe a novel approach for reducing the error estimation gap between the synthetic and the real data. We propose to calibrate the synthetic data error estimation using a held out set of classifiers and show that this results in improved rank preservation across the domains. We provide the following intuition behind the calibration process. According to Theorem 3.2, the ability to use synthetic data for ranking models depends only on the probability density gap between the synthetic and real distribution in the area of disagreement, $\delta_{h_1 \oplus h_2}(\mu_r, \mu_s)$. Thus, reducing the disagreement effectively improves the ranking. Towards this goal, we re-weight the contribution of each synthetic sample to the estimated error. The re-weighted error no longer corresponds to the original synthetic distribution, rather it corresponds to a distribution that is closer to the real one. We assign weights such that the calculated weighted errors of a small set of calibration models, will be close to their corresponding real errors. If the synthetic images contain information that allows discrimination between models, and the discrimination correlates with model ranking, then we expect the ranking to generalize to unseen models.

To achieve this, we construct the following regression problem. Given a set of M models and C classes, where each class has N_C synthetic images, $\{\{x_i^c\}_{i=1}^{N_c}\}_{c=1}^{C}$. We denote $\hat{\epsilon}_{r,m}^c$ as the empirical risk of model m over real images belonging to class c. We also denote $\hat{\epsilon}_r^c = [\hat{\epsilon}_{r,m}^c]_{m=1}^M$, *i.e.* the real empirical risks of all models on class c. $\mathbf{Q}_c \in \mathbb{Z}_2^{N_c \times M}$ is a binary matrix where each column, $\mathbf{q}_m^c$, indicates the prediction correctness of model m on images $\{x_i^c\}_{i=1}^{N_c}$. We wish to find a set of weights, $\mathbf{w}_c \in \mathbb{R}^{N_c}$, that solves the linear ridge regression problem:

$$\mathbf{w}_c = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ ||\hat{\epsilon}_r^c - \mathbf{Q}_c{}^T \mathbf{w}||_2^2 + \lambda ||\mathbf{w}||_2^2 \right\}, \qquad (1)$$

where λ is the Lagrange multiplier[2]. Intuitively, the weight penalty "spreads" the error influence across many synthetic images, preventing a single image to dominate the ranking. This prevents "overfitting" the solution to the models used for calibration. For each class, c, we solve an independent optimization problem using a closed-form solution[3]. Once optimization is done, we estimate the error of a new model m', on the calibrated dataset by $\hat{\epsilon}_{s,m'} = \sum_{c=1}^{C} (\mathbf{q}_{m'}^c)^T \mathbf{w}_c$.

For this experiment, we use the $M = 50$ models that were set aside for calibration. Figure 5 and Table 3 demonstrate a drastic improvement in rank preservation. For the DiT

[1] https://github.com/cadene/pretrained-models.pytorch.

[2] We set $\lambda = 0.5$ and include an intercept (bias) term.

[3] $\mathbf{w}_c = (\mathbf{Q}_c^T \mathbf{Q}_c + \lambda \mathbf{I})^{-1} \mathbf{Q}_c^T \hat{\epsilon}_r^c$.

(cfg $= 1, 2, 3, 4$) dataset the calibration process improved the Spearman's correlation from 0.893 to 0.986, which is comparable to the ImageNetV2 (0.994). Surprisingly, BigGAN's performance improved from total irrelevance (-0.363) to be competitive for rank preservation (0.978), surpassing all uncalibrated datasets. We present synthetic images and their corresponding calibration coefficients in Appendix I. We suspect that while the images are not coherent, they may contain certain features that are correlated with the correct class. Models that are able to detect these features will tend to have better performance, and vice versa, some features may be negatively correlated with a certain class, *i.e*. being able to detect it indicates poorer accuracy for this class.

In Table 4 we present the Spearman correlation while varying the number of models used for calibration (M). The results indicate that calibrating with five models is enough to achieve a significant improvement.

5. Conclusions

In this paper we presented a comprehensive empirical study, evaluating the impact of using synthetic data for model selection. The empirical evidence suggest that evaluating trained models on synthetic data can outperform the standard methods for model selection that are based solely on the available real images. In addition, we show that synthetic data can be used to rank models trained on high-resolution images from a diverse set of classes and we propose a novel calibration method that significantly improves the ranking capabilities.

References

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A Theory of Learning From Different Domains. *Machine learning*, 79(1):151–175, 2010.

Besnier, V., Jain, H., Bursuc, A., Cord, M., and Pérez, P. 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.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Corneanu, C. A., Escalera, S., and Martinez, A. M. Computing the Testing Error Without a Testing Set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2677–2685, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

DeVries, T. and Taylor, G. W. Improved Regularization of Convolutional Neural Networks with Cutout, 2017.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs on Image Synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Eilertsen, G., Tsirikoglou, A., Lundström, C., and Unger, J. Ensembles of GANs for Synthetic Training Data Generation. *arXiv preprint arXiv:2104.11797*, 2021.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Gastaldi, X. Shake-shake regularization of 3-branch residual networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

Ghosh, S., Stephenson, W. T., Nguyen, T. D., Deshpande, S. K., and Broderick, T. Approximate cross-validation for structured models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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

Han, D., Kim, J., and Kim, J. Deep pyramidal residual networks. *IEEE CVPR*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 630–645. Springer, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016b.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NIPS*, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. Squeeze-and-excitation networks, 2019.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L., and Weinberger, K. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive Growing of GANs for Iimproved Quality, Stability, and Variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Laine, S., and Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR*, pp. 4401–4410, 2019a.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of StyleGAN. *CoRR*, abs/1912.04958, 2019b.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. Training Generative Adversarial Networks with Limited Data. *arXiv preprint arXiv:2006.06676*, 2020.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021a.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021b.

Kohavi, R. 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.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019.

Li, W., Geng, C., and Chen, S. Leave zero out: Towards a no-cross-validation approach for model selection. *arXiv preprint arXiv:2012.13309*, 2020.

Liu, Y., Lin, H., Ding, L., Wang, W., and Liao, S. Fast cross-validation. In *IJCAI*, pp. 2497–2503, 2018.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J., and Saenko, K. 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.

Ravuri, S. and Vinyals, O. Classification accuracy score for conditional generative models. *arXiv preprint arXiv:1905.10887*, 2019.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Wightman, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Wilson, A., Kasy, M., and Mackey, L. Approximate cross-validation: Guarantees for model assessment and selection. In *International Conference on Artificial Intelligence and Statistics*, pp. 4530–4540. PMLR, 2020.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

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

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

A. Connection to $\mathcal{H}\Delta\mathcal{H}$-divergence (Ben-David et al., 2010)

In this section we show a connection to another common metric for measuring the difference between domains, the $\mathcal{H}\Delta\mathcal{H}$-divergence.

Lemma A.1 (Lemma 3 from (Ben-David et al., 2010))**.** *For any pair of hypotheses $h_1, h_2 \in \mathcal{H}$,*

$$|\epsilon_s(h_1, h_2) - \epsilon_r(h_1, h_2)| \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r).$$

Proof. By the definition of $\mathcal{H}\Delta\mathcal{H}$-divergence,

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r) &= 2 \sup_{h,h' \in \mathcal{H}} |Pr_{\mathbf{x}\sim\mathcal{D}_s}[h(\mathbf{x}) \neq h'(\mathbf{x})] - Pr_{\mathbf{x}\sim\mathcal{D}_r}[h(\mathbf{x}) \neq h'(\mathbf{x})]| \\ &= 2 \sup_{h,h' \in \mathcal{H}} |\epsilon_s(h, h') - \epsilon_r(h, h')| \geq 2|\epsilon_s(h_1, h_2) - \epsilon_r(h_1, h_2)| \end{aligned}$$

□

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. Then, for any $h_1, h_2 \in \mathcal{H}$:*

$$\Delta\epsilon_s - \Delta\epsilon_r \leq d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r)$$

Proof.

$$\begin{aligned} \Delta\epsilon_s - \Delta\epsilon_r &= \epsilon_s(h_2) - \epsilon_s(h_1) - (\epsilon_r(h_2) - \epsilon_r(h_1)) \\ &= \epsilon_s(h_2) - \epsilon_r(h_2) + \epsilon_r(h_1) - \epsilon_s(h_1) \\ &= \epsilon_s(h_2, f) - \epsilon_r(h_2, f) + \epsilon_r(h_1, f) - \epsilon_s(h_1, f) \\ &\leq |\epsilon_s(h_2, f) - \epsilon_r(h_2, f)| + |\epsilon_r(h_1, f) - \epsilon_s(h_1, f)| \\ &\leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_r, \mathcal{D}_s) \\ &= d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r) \end{aligned}$$

□

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

$$\Delta\epsilon_s \geq d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r) \Rightarrow \Delta\epsilon_r \geq 0,$$

where $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_r)$ is the $\mathcal{H}\Delta\mathcal{H}$-divergence between the two distributions.

B. Proof of Lemma 3.1

Lemma B.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 $\Omega_1 = \{\mathbf{x} \in \Omega | h_1(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_2(\mathbf{x}) = f(\mathbf{x})\}$ and $\Omega_2 = \{\mathbf{x} \in \Omega | h_2(\mathbf{x}) \neq f(\mathbf{x}) \wedge h_1(\mathbf{x}) = f(\mathbf{x})\}$. Then,*

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

Proof.

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

□

C. 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 G). The train/val split and the GANs that where used to create each synthetic dataset are as follows:

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 StyleGAN2 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.

D. Additional results for early stopping and random seed selection on standard architectures

In addition to the results reported in 4.2.1, Figure 6 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.

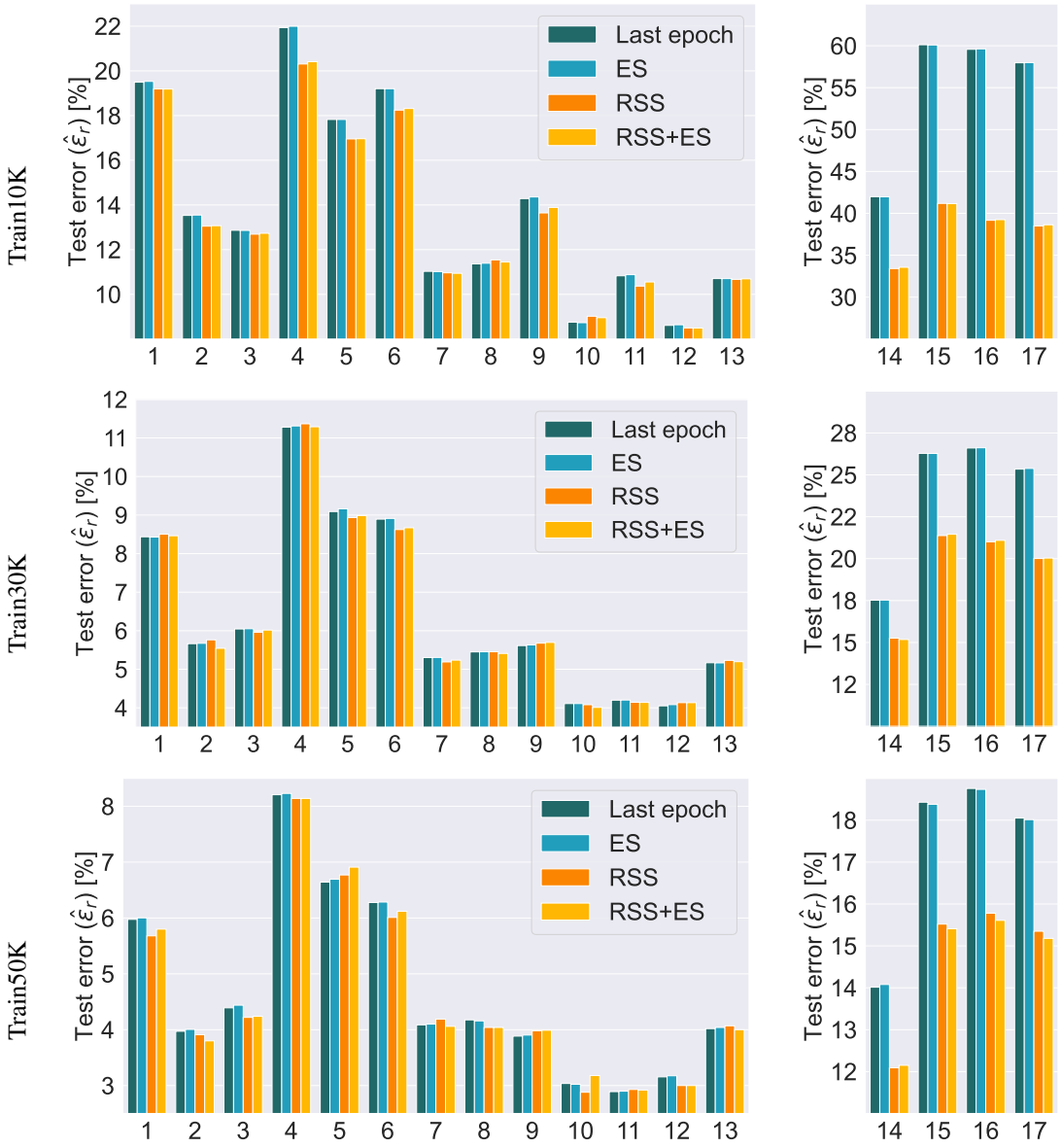
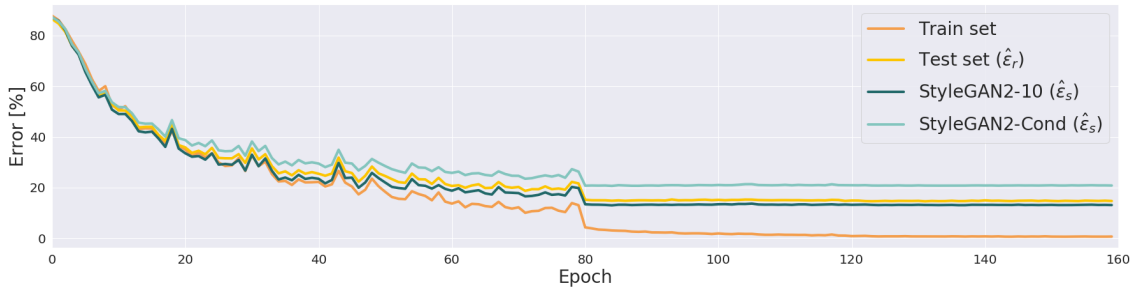


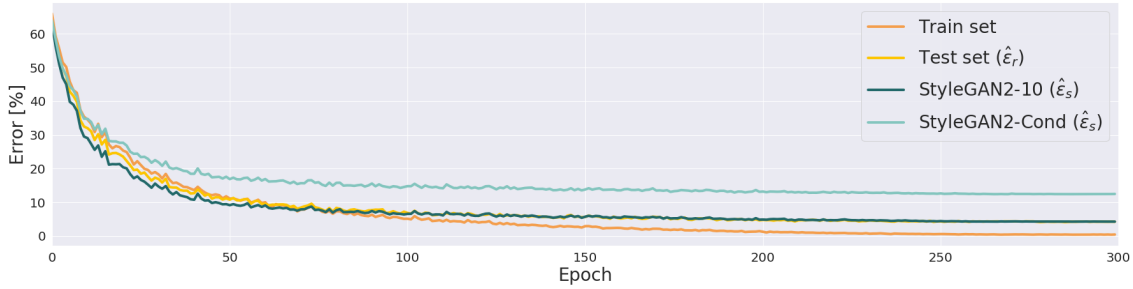
Figure 6: **ES and RSS for model selection (standard architectures) on all datasets:** Test errors of 17 architectures trained on each of the datasets. "Last epoch" and ES show the average error of the 10 models for each architecture. RSS and RSS+ES show the results of the selected model out of the 10 models.

E. Standard architecture convergence in training

Figure 7 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.



(a) ResNet110 (Architecture 17 in Appendix F)



(b) Shake-Shake 64 + cutout (Architecture 10 in Appendix F)

Figure 7: **Train, test and synthetic data error vs. epoch.**

F. Standard Architectures Description

Below is the list of architectures used in Sections 4.1 and 4.2.1:

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., 2016a) with depth 110, block type "basic".
6. **ResNet-preact 164:** (He et al., 2016a) 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., 2016b) with depth 32, block type "basic".
15. **ResNet 44:** (He et al., 2016b) with depth 44, block type "basic".
16. **ResNet 56:** (He et al., 2016b) with depth 56, block type "basic".
17. **ResNet 110:** (He et al., 2016b) with depth 110, block type "basic".

G. Synthetic Data Generation Details (CIFAR10)

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., 2019b) 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 5 shows the FID scores breakdown for our synthetic datasets. As expected, as the training dataset size decreases the FID score increases.

Figures 8 and 9 show samples of real CIFAR10 images and our synthetic StyleGAN2-based datasets for each of the CIAFR10 classes.

Table 5: **FID scores (↓) breakdown.**

Synth dataset	Class	Train50K	Train30K	Train10K
StyleGAN2-10	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
	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

Figure 8: **Real images vs. StyleGAN2-10 datasets.**

Figure 9: **Real images vs. StyleGAN2-Cond datasets.**

H. Generated Synthetic ImageNet Samples

Figure 10 shows randomly sampled images from our five synthetic datasets (BigGAN, DiT with cfg $= 1/2/3$ or 4) for the classes: Goldfish (1), Siberian husky (250), Lion (291), Balloon (555), Fire truck (555), Denim (608), Baseball player (981).

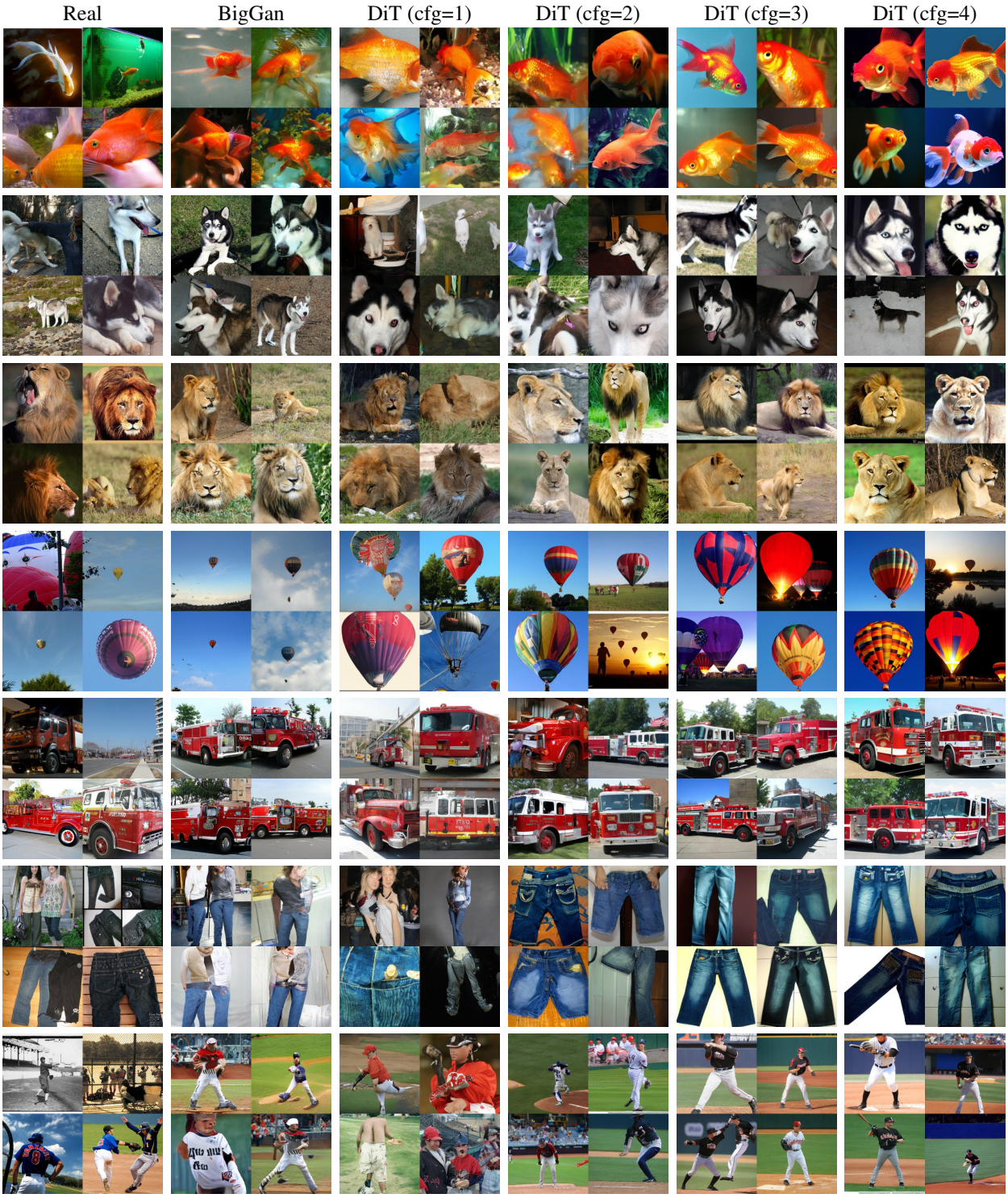


Figure 10: **ImageNet generated samples.** Rows show the following classes: Goldfish (1), Siberian husky (250), Lion (291), Balloon (555), Fire truck (555), Denim (608), Baseball player (981).

I. Calibrated Synthetic ImageNet Samples

In this section we analyze the contribution of images from calibrated synthetic datasets to model ranking. In Figures 11 and 12 we show the images, x_i, that received the highest and lowest coefficients, $\mathbf{w}_c[i]$, in their corresponding classes, during the calibration process. Each row shows images from a different synthetic dataset. The first four columns show the images that received the highest positive and lowest negative w coefficient values during calibration. Columns five and six show samples that are not useful for ranking and received $w = 0$. The last column shows the distribution of w for all 100 images in the class. For sake of presentation clarity we use w as a shorthand notation for $\mathbf{w}_c[i]$ next to an image to denote the value it received in its matching vector $\mathbf{w}_c$. Images with positive w coefficients increase the likelihood of models labeling them correctly to be ranked higher. In contrast, images with negative values of w indicate that calibration models labeling them "incorectly" in general perform well on that class. The improvement in ranking of unseen models indicates that in general images with non-zero coefficient are valuable for ranking. Images with $w = 0$ are those which were unanimously classified either correctly or incorrectly by all the models in the calibration set.

Figure 11: **Calibrated samples of class "Kite" (bird, 21), "Vase" (883).**

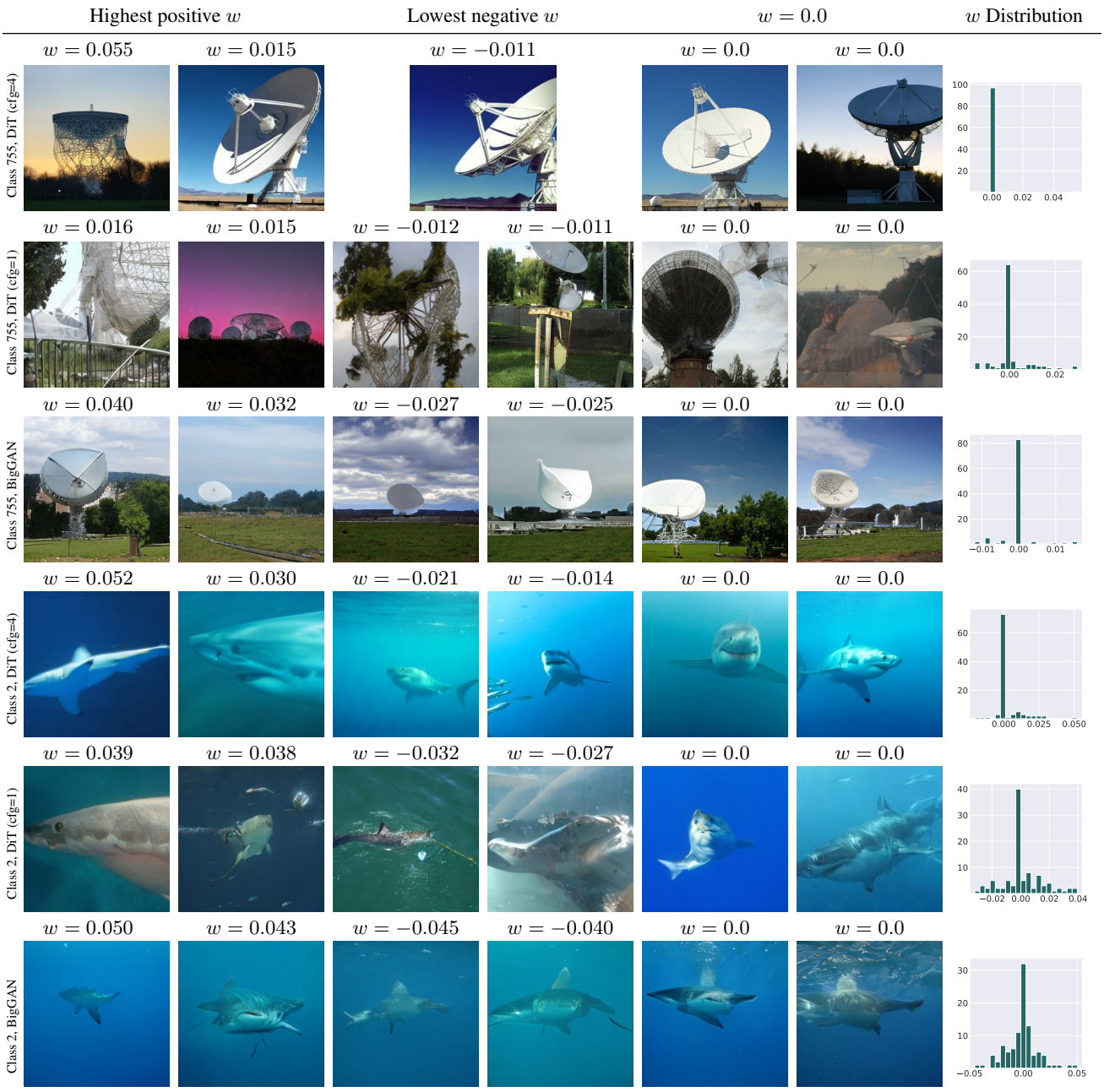


Figure 12: **Calibrated samples of classes "Radio telescope" (755), "White shark" (2).** For DiT (cfg $= 4$) only one image is presented for the "Radio telescope" class as only it has received a negative w.