Appendix for "Rate-Regularization and Generalization in Variational Autoencoders"

A1 Experiment Setup

Various settings for all our experiments are displayed in Table A1. We decided to keep all the hyperparameters (other than β, depth, and training set size) fixed. We use ReLU activations for both fully-connected and CNN architectures. Based on our initial experiments, where the effect of network width on generalization proved to be minor (see Figure A9), we use 512 hidden units in all experiments in the main text. There are a variety of ways for changing the capacity of convolutional neural networks. In our experiments, we decided to focus on changing the number of layers and keep the other hyperparameters such number of channels, kernel size, stride, and padding fixed. In all models, we use a 10-dimensional latent space and assume a spherical Gaussian prior. All models are trained for 257k iterations with Adam (default parameters, amsgrad enabled) using a batch size of 128, with 5 random restarts. We have also performed experiments with early-stopping which we will discuss in Section A3.7.

A1.1 Implementation Details

All experiments were ran on NVIDIA 1080Ti and Tesla V100 GPUS (depending on availability), using Pytorch 1.3.0 and ProbTorch commit f9f5c9. Most models are trained with 32-bit precision. A few models (3-layer β-VAEs with $\beta < 0.1$) that didn't train were retrained using 64-bit precision.

A1.2 Likelihood

We use a standard Bernoulli likelihood in the decoder. This eliminates the extra tunable parameter σ that is present in Gaussian decoders, which is redundant since it controls the strength of the reconstruction loss in the same manner as the β coefficient. The Bernoulli likelihood is in fact a very common choice in the VAE literature and appears in the original VAE paper, tutorials, and in reference implementations in deep learning frameworks. Although not reported, we have verified that Gaussian decoders with fixed σ show the same RD trends as the Bernoulli decoder in the 1-layer case (see Figure A23). In the 3-layer case, we observed that using a normal likelihood significantly suffers from a mode-collapse problem.

Table A1: Hyperparameters common to each of the considered datasets

	Tetrominoes	MNIST	F-MNIST	3dShapes	SmallNORB
Batch-size	128	128	128	128	128
Number of iterations	256k	256k	256k	256k	256k
Latent space dimension	10	10	10	10	10
Number of hidden units (MLP)	512	512	512	512	512
Number of channels (CNN)	64	64	64	64	64
Kernel size (CNN)	4	4	4	4	4
Stride (CNN)	2	2	2	2	2
Padding (CNN)	1	3	3	1	1

A2 Tetrominoes dataset

Table A2: Names, and training and test set sizes of Tetrominoes datasets used in the paper.

Dataset		Training	Test
CV	8k/157k	8193	155647
	16k/147k	16384	147456
	25k/139k	24577	139263
	33k/131k	32768	131072
	41k/123k	40960	122880
	49k/115k	49153	114687
	57k/106k	57344	106496
Default		81920	81920
CD	8k/8k	8159	8275
	16k/16k	16405	16286
	25k/25k	24642	24592
	33k/33k	32998	32754
	41k/41k	41138	40954
	49k/49k	49216	49285
	57k/57k	57416	57403
Checkerboard		82021	81819

In this section, we take a closer look at the "difficulty" of generalization problem in the Tetrominoes dataset. One can argue that generalization "difficulty" in any dataset is essentially linked to the closeness of training and test set in pixel space. This of-course depends not only on the nature of the dataset, but on size of both training and test sets. Moreover, we need to define the notion of *closeness* between training and test set in advance. One approach to quantify this concept is the following: for every example in the test, what is the distance to the closest example in training set for a given distance metric?

In Figure A1, we show the normalized ℓ_2 histograms of test examples to their nearest neighbour in training set for different amount of training data. In Figure A2, we show test samples and their nearest ℓ_2 neighbour in training set for different splits.

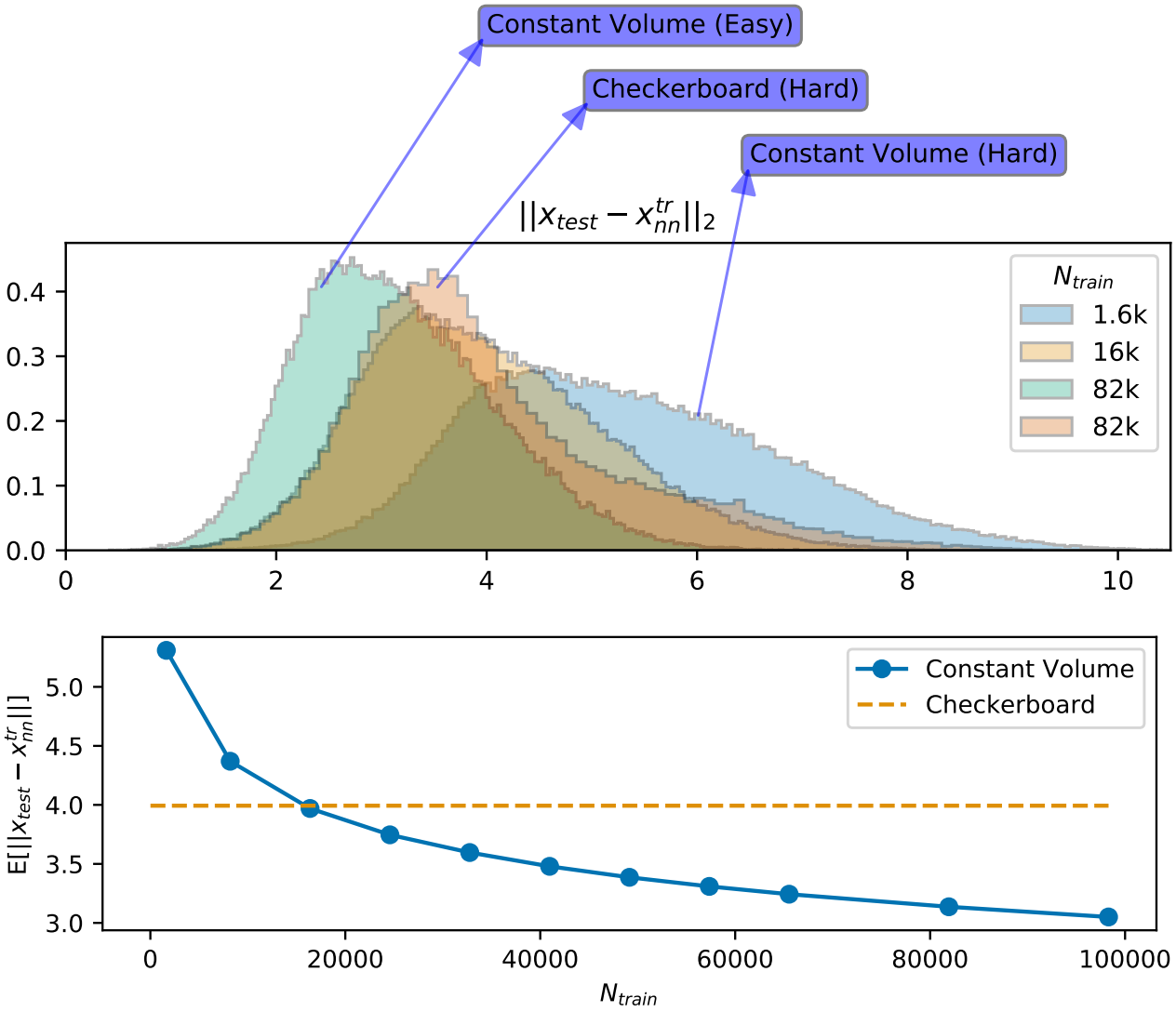


Figure A1: Analysis of ℓ_2 distance in pixel space between training and test set in Tetrominoes dataset based on N_{train}. (*Top*) Normalized histograms of ℓ_2 distance between test examples $\boldsymbol{x}$ and the nearest neighbour in the training set $\boldsymbol{x}^{tr}_{nn}$. Unsurprisingly, As the amount of training data increases, the distribution of ℓ_2 norm between test examples and their nearest neighbour in the training set moves towards 0. (*Bottom*) Mean of $\|\boldsymbol{x} - \boldsymbol{x}^{tr}_{nn}\|_2$ distribution as a function of train of ratio.

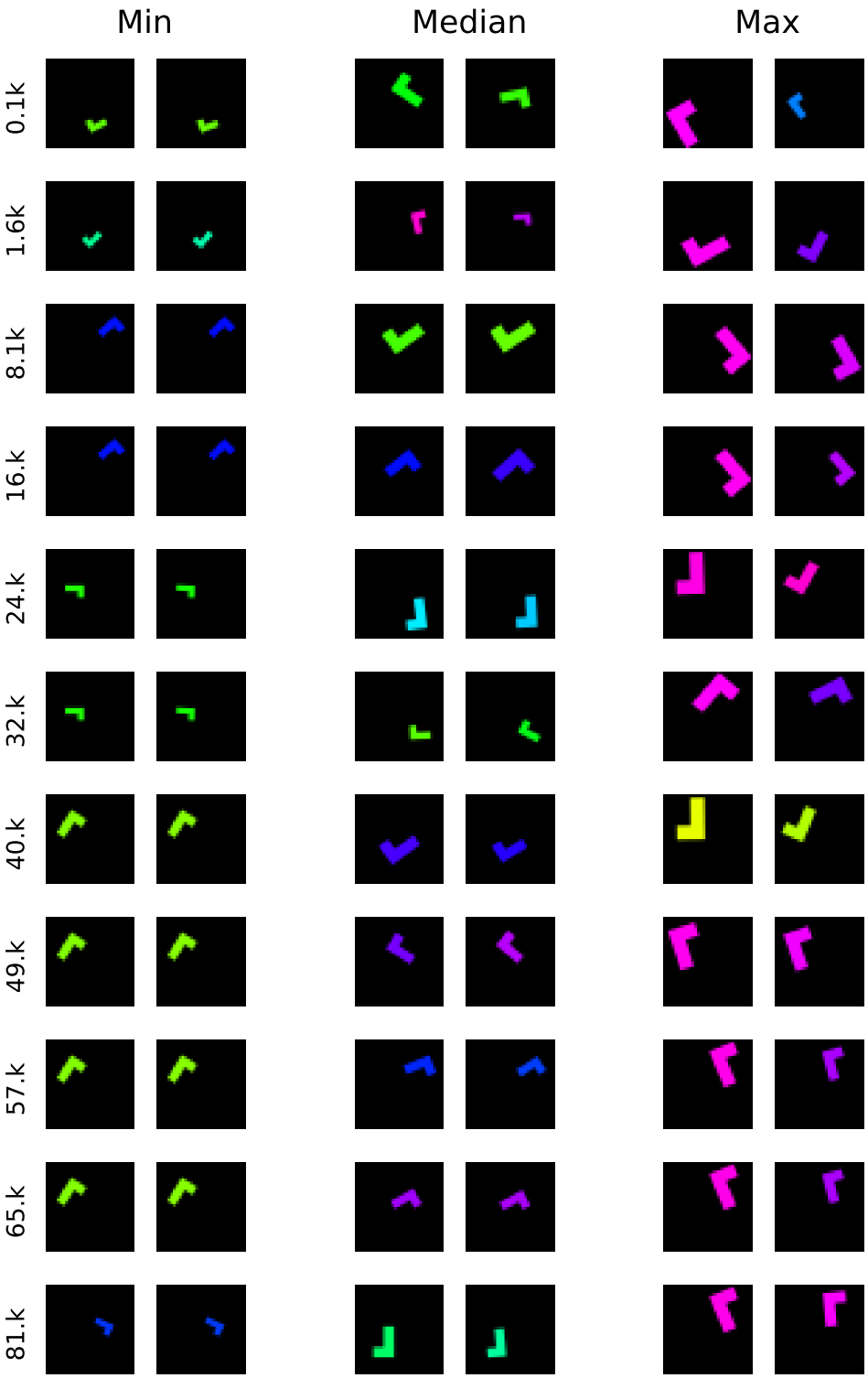


Figure A2: Test samples with minimum (*left*), median (*middle*), and maximum (*right*) ℓ_2 norm between their nearest neighbour in training set, for splits with various N_{train}. In each column, the test sample is displayed on the left, and the nearest neighbour is displayed on the right.

A2.1 Which Features Are the Most Different in Pixel Space?

One crucial factor in the difficulty of generalization in a dataset is the change caused in image space that is caused by moving in feature space. Not only this property can be different for different features, but it may also depend on the location in features space that change is happening (see Figure A3). In order to have a better understating of which features are more difficult to generalize to, we performed the following experiments. For all 163,840 tetromino images, we changed a single feature by a single unit. Figure A3 shows the ℓ_2 distance between the corresponding images.

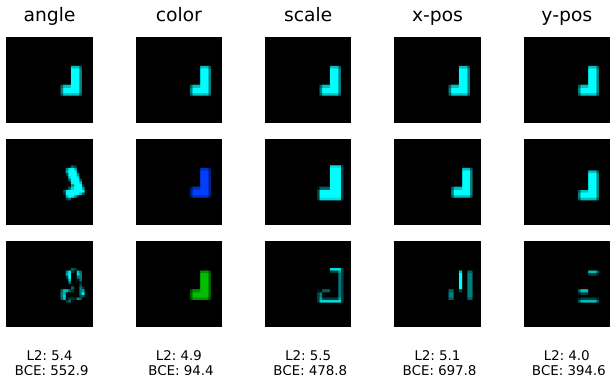


Figure A3: Effect of each feature in pixel space for Tetrominoes dataset. (*Top*) Original image. (*Middle*) A single feature in the original image modified by one unit. (*Bottom*) ℓ_2 distance between images in the top and middle.

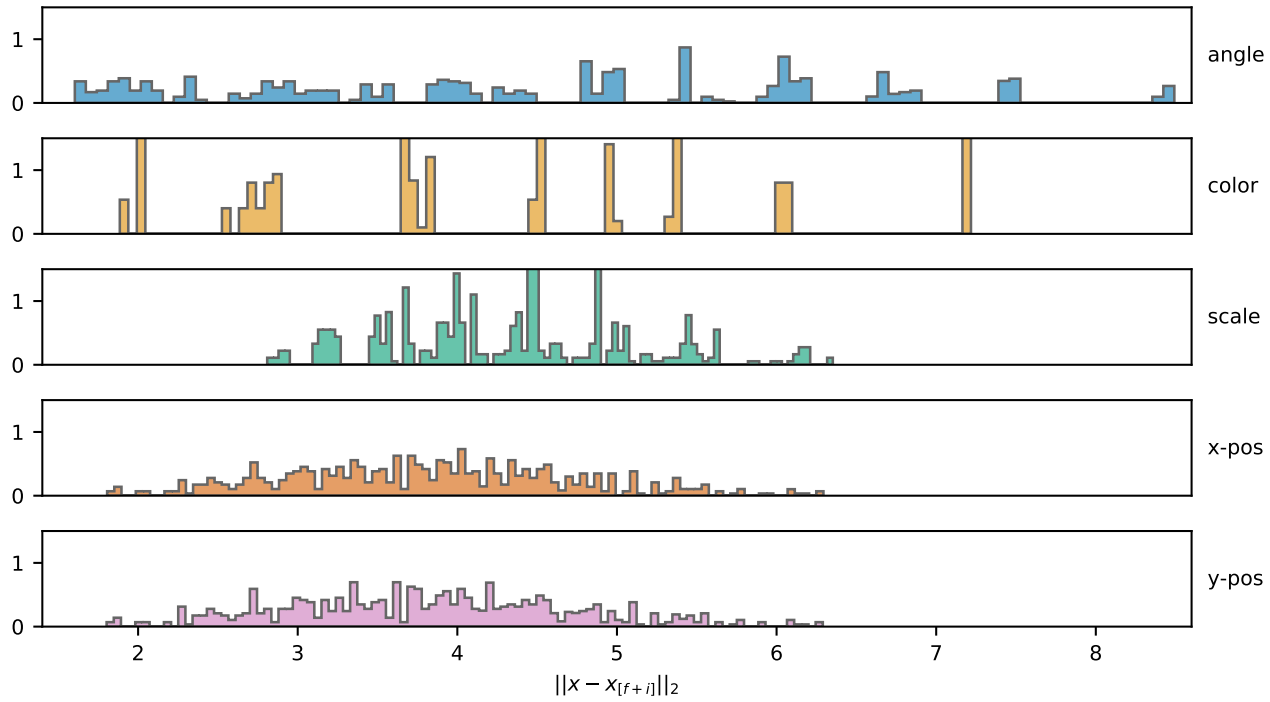


Figure A4: Histograms of ℓ_2 distance between each tetromino image and the same tetromino modified in a single feature by 1 unit. We can observe that size causes the least difference in pixel space. x and y-positions seem to be the second and third most influential factors. Angle causes the most difference in pixel space.

A3 Additional Experimental Results

A3.1 MNIST 9-Removal Experiment

As a means of gaining intuition on the VAE's ability to generalize to unseen example, we first carried out the following experiment. We trained a VAE with a 50 dimensional latent space on all MNIST digits with the exception of the 9s, and then tested out-of-domain generalization by attempting to reconstruct 9s. Figure A5 shows the decoder output, the weighted average μ from Equation 7, and the 3 training examples with the largest weights. We compare a shallow (1 hidden layer, 400 neurons) and a deep (3 hidden layers, 400 neurons each) network.

As Proposition 1 predicts, the deep VAE reconstructions closely resemble the nearest neighbors in the training data in latent space. In most cases, a single sample dominates the weighted average. This is evident from the histogram of weight perplexities, which is strongly peaked at 1. This, combined with decoder outputs and neighbours with largest weights, suggests that VAEs can memorize training data even for simple encoder/decoder architectures with moderate capacity, and reconstructions are well-approximated by nearest neighbors in the training set when they do so.

However, it is not the case that VAEs always memorize training data. A surprising finding is that shallow VAEs show comparatively good generalization to out-of-domain samples; reconstructions of 9s are passable, even though this digit class was not seen during training. The same trend is also visible when we compare the binary cross entropy (BCE) between reconstructions and samples from the withheld class, to the BCE between the reconstructions and the weighted averages (see Figure A6). This suggests that the assumption of infinite capacity in Proposition 1 clearly matters, and that layer depth significantly affects the effective capacity of the network.

In the previous experiment, we provided a comparison between a shallow and a deep VAE. We now do a finer grained complexity analysis, and identify regions where Theorem 1 holds. As with "generalization", "capacity" of a neural network is a hard aspect to characterize. Here, we will use number of parameters and layers as simple proxies for capacity. In Figure A7, we show reconstructed and weighted average images for 17 VAEs with different network architectures given an input sample from the withheld class. As VAEs get more complex, they overfit the training data therefore fail to reconstruct the unseen digit. Moreover, we observe that the reconstruction is closer to the weighted average for higher capacity networks. Another -intuitive- observation here the number of layers plays a more crucial role in complexity than number of parameters, since for VAEs with 3 hidden layers, reconstructions are more similar to weighted average, while in single hidden layer VAEs, the reconstructions match the input sample regardless of the number of parameters.

A3.2 Effect of Network Capacity When β=1

To probe the role of the model capacity in generalization, we compare 9 architectures that are trained using a standard VAE objective on the Default, CV (16k/147k), and Checkerboard splits. The CV(16k/147k) split is designed have similar typical pixel distance between nearest neighbors in the training and test set to the Checkerboard split (see Appendix A2), which enables a fair comparison by controlling for the difficulty of the generalization problem. We vary model capacity by using architectures with {1, 2, 3} layers that each have {256, 512, 1024} neurons.

Figure A9 (*left*) shows the test-set rate and distortion for all 27 models. Dashed lines indicate contours of equal ELBO $= D + R$. In Figure A9 (*right*) we compare the LM for the training and test set. Here the dashed line marks the boundary where the test LM equals the training LM. For the Default split, increasing the model capacity uniformly improves generalization. Conversely, for the CV (16k/147k) and Checkerboard splits, we observe a strong deterioration in generalization performance in all 3-layer architectures.

Figure A9 suggests that increasing model capacity can either improve or hurt generalization, depending on the difficulty of the generalization problem. In the Default split, which poses a comparatively easy generalization task, we observe that increasing model capacity improves generalization, whereas for more difficult tasks (as classically predicted in terms of a bias-variance trade-off), increasing model capacity deteriorates generalization. We note here that the discrepancy between generalization performance on the CV and Checkerboard splits is relatively small, which suggests that these in-domain and out-of-domain tasks are indeed comparable in terms of their difficulty.

A3.3 Memorization and Generalization When β=1

Regardless of our metric for generalization performance, there is evidence that VAEs can both underfit and overfit the training data. Several researchers (Bousquet et al., 2017; Rezende and Viola, 2018; Alemi et al., 2018; Shu et al., 2018) have pointed out that an infinite-capacity optimal decoder will memorize the training data. Concretely, the following proposition holds for an optimal decoder:

Proposition 1 (Shu et al. (2018)) *Assume a likelihood $p(\boldsymbol{x} \mid \boldsymbol{z})$ in an exponential family with mean parameters μ and sufficient statistics $T(\cdot)$, a fixed encoder $q(\boldsymbol{z} \mid \boldsymbol{x})$, and training data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_{train}}\}$. In the limit of infinite capacity, the optimal decoder $\mu(\boldsymbol{z})$ is*

$$\mu(\boldsymbol{z}) = \sum_{n=1}^{N_{train}} w_n(\boldsymbol{z})\, T(\boldsymbol{x}_n), \quad w_n(\boldsymbol{z}) = \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_n)}{\sum_m q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_m)}. \tag{7}$$

We qualitatively evaluate the extent to which VAEs memorize the training data (as predicted by Proposition 1). Figure A10 compares the reconstructions of test examples in the Default, CV (16k/147k), and Checkerboard splits, by 1-layer and 3-layer architectures. For each architecture, we show 3 examples from the test set along with reconstructions and nearest training neighbours (with respect to w_n) for both models. The 3 examples are representatives of easy ($<10^{\text{th}}$ percentile), typical (45^{th}-55^{th} percentile), or difficult ($>90^{\text{th}}$ percentile) samples in terms of pixel-wise nearest-neighbor distance to the training data. In the case of the CV (16k/147k) and Checkerboard splits, we see that the 1-layer VAE can reconstruct unseen examples even when the nearest neighbour in the latent space is quite different, while reconstructions for the 3-layer VAE are consistent with the memorization behavior described by Proposition 1. In the Default split, we observe that reconstructions are similar for 1-layer and 3-layer architectures, and are often well-approximated by their nearest-neighbors in the training data.

To provide a more quantitative evaluation, we report distances between $\boldsymbol{x}$, $\boldsymbol{\mu}$ and $\hat{\boldsymbol{x}}$. Figure A11 shows violin plots of the pixel-wise distances between test images and their reconstructions ($\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$) and the infinite-capacity decoder outputs ($\|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|$) as defined in Equation (7). We split the test set into 4 bins of equal sizes according to the distance of examples to their nearest training example ($\|\boldsymbol{x} - \boldsymbol{x}^{tr}_{nn}\|$) and show histogram pairs for each bin. Values in x-axis indicate the limits of each bin.

For splits that are not trivial, we see different behavior across different bins. Looking at the rightmost histogram-pairs (the most difficult 25% of test samples) in each panel, we once again observe qualitatively different behaviors for 1-layer and 3-layer networks. For the 1-layer networks $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$ is smaller than $\|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|$, which shows that reconstructions cannot be explained by memorization alone. This result holds across all 3 train-test splits. For 3-layer networks, we see that $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$ slightly decreases relative to the 1-layer model in the Default split, once again indicating that overparameterization aids generalization (here in terms of reconstruction loss) in this regime. Conversely for the CV and Checkerboard splits, we see that $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$ increases relative to the 1-layer model. Moreover, in the Checkerboard split, we observe that $\|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|$ is smaller than $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$, which shows that reconstructions are closer to memorized data than to the actual test examples.

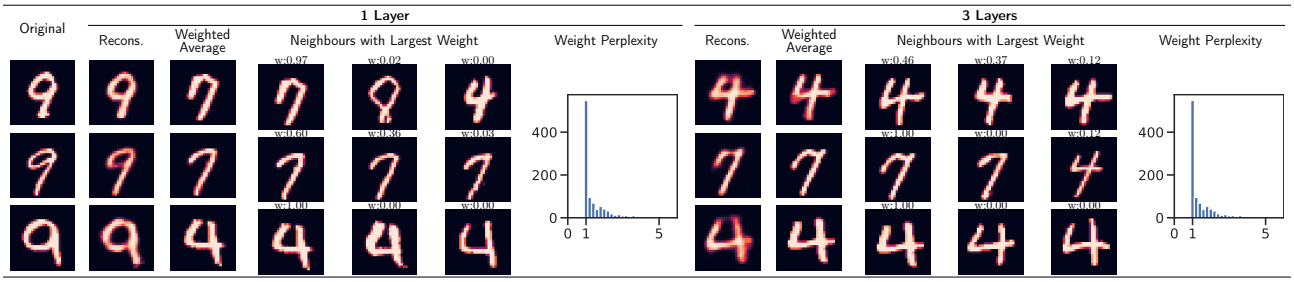


Figure A5: MNIST 9-removal experiment: Reconstruction of out-of-domain samples by 1-layer and 3-layer VAEs, alongside weighted average, and neighbours in training data with largest weights. Reconstructions in a 3-layer VAE closely match the weighted average, which in turn are often just the nearest neighbour in training data). 1-layer VAE does not exhibit this behaviour and can reconstruct samples cannot be represented by the convex combination of its training data.

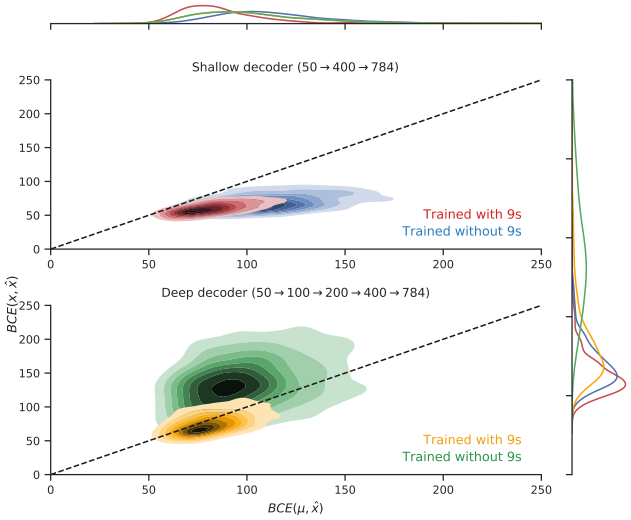


Figure A6: Distribution of the BCE loss between input image $\boldsymbol{x}$ and the decoder output $\hat{x}$, vs the loss between weighted average image μ and $\hat{x}$; calculated over test set consisting of images of 9s. The reconstructions are closer to the input than the weighted average for the 1-layer VAE (most of the mass of blue distribution lies below the line). On the other hand, the reconstructions are closer closer to the weighted average than the input for the 3-layer VAE (most of the mass of green distribution lies above the line).

A3.4 ELBO and Log Marginal Likelihood

As discussed in Section 2, there are different ways of viewing the VAE objective. For the purpose understanding the effect of the rate, so far we focused on the distortion as the metric and studying the role of the rate as just a regularizer. Here, we study the effect of β on more common metrics for evaluating generalization in VAEs, namely the ELBO and log marginal likelihood $\log p_\theta(\boldsymbol{x})$.

We first look at the standard-VAE objective ($\beta = 1$) in order to separate the influence of KL and the difficulty of generalization problem as well as depth. Figure A12 shows the ELBO and $\log p_\theta(\boldsymbol{x})$ as a function of the training set size at $\beta = 1$. Once again, we observe two qualitatively different forms of behavior. In the CD splits, 3-layer networks almost uniformly outperform 1-layer networks. In the case of CV splits, there is a cross-over. The 1-layer model performs better for smaller (sparser) datasets, but is overtaken by the 3-layer model for larger (denser) datasets. These results suggest that overparameterizing can in fact be beneficial if the training and test sets are similar. Furthermore, it indicates it is not the size of training data but the similarly of training and test examples that is the key factor.

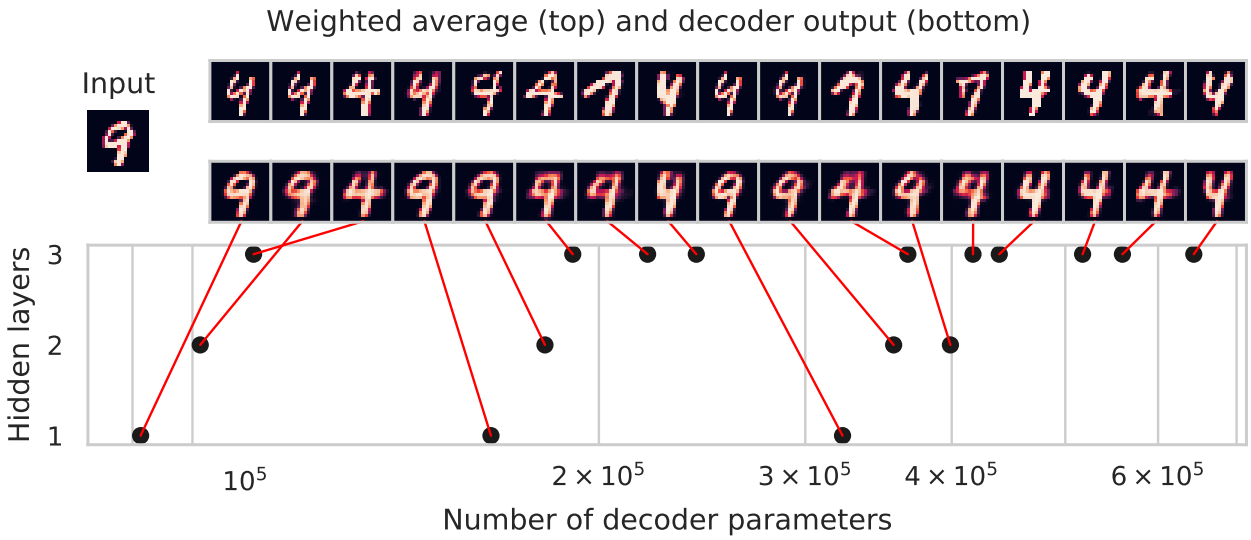


Figure A7: Decoder outputs and weighted average images for VAEs with different architectures.

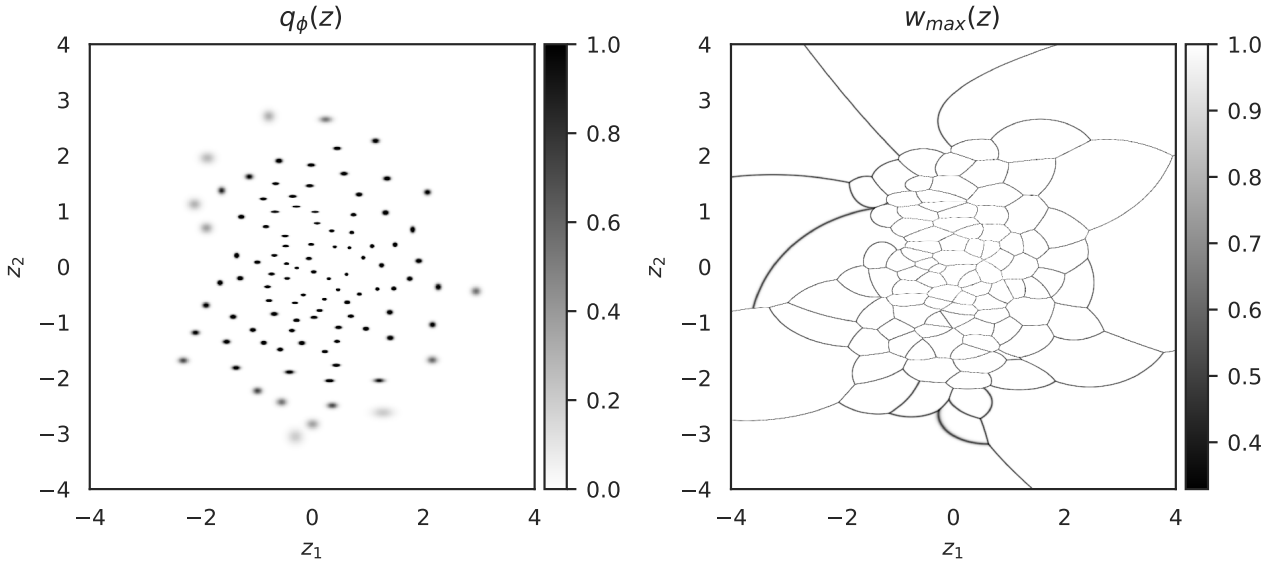


Figure A8: Data memorization in a VAE with latent dimension trained on 100 MNIST examples. (*Left*) Inference marginal $q_\phi(\boldsymbol{z}) = \frac{1}{N}\sum_n q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_n)$. (*Right*) Partitioning of the latent space. To close approximation, the decoder reconstructs a memorized nearest neighbor from the training data (for approximately 96% of the shown region, the largest weight is $w_{max} > 0.99$)

.

A3.4.1 The $\beta \neq 1$ Case

It is not considered standard to compute ELBO or Log marginal likelihood (LM) for β values other than 1 for two main reasons. First, the β-VAE objective 1 has mainly been trained for the purpose of disentanglement rather than learning a generative model. Second, for values lower than β is no longer a lower bound. However, we observed in Figure 3 that it is possible to achieve a *higher* test ELBO when we set $\beta < 1$. Therefore, we decided to investigate the effect of KL regularization on the test ELBO and LM by plotting the rate against LM (Figure A13) and ELBO (Figure A14).

Looking at the first and last rows in Figures A13 and A14, we observe that $\beta = 1$ (or nearby values) typically yield the highest LM and ELBO. These results are unsurprising given that $\beta = 1$ results in the objective being the exact ELBO. We also observe that the 3-layer models are able to achieve a higher LM and ELBO than the 1-layer models. This confirms our intuition that when the distance between training and test examples are small, we can

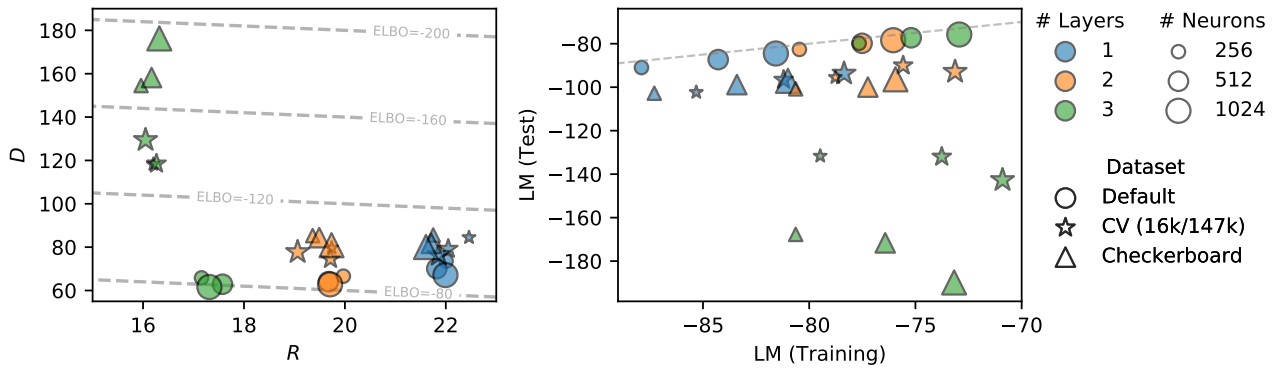


Figure A9: Analysis of network capacity: Each point is a standard VAE model with a different architecture, evaluated on a different dataset. *(Left)* Test-set rate and distortion. The dashed lines denote contours of constant ELBO. *(Right)* Training LM versus test LM. The dashed line denotes the contour where the LM for training and test sets are equal.

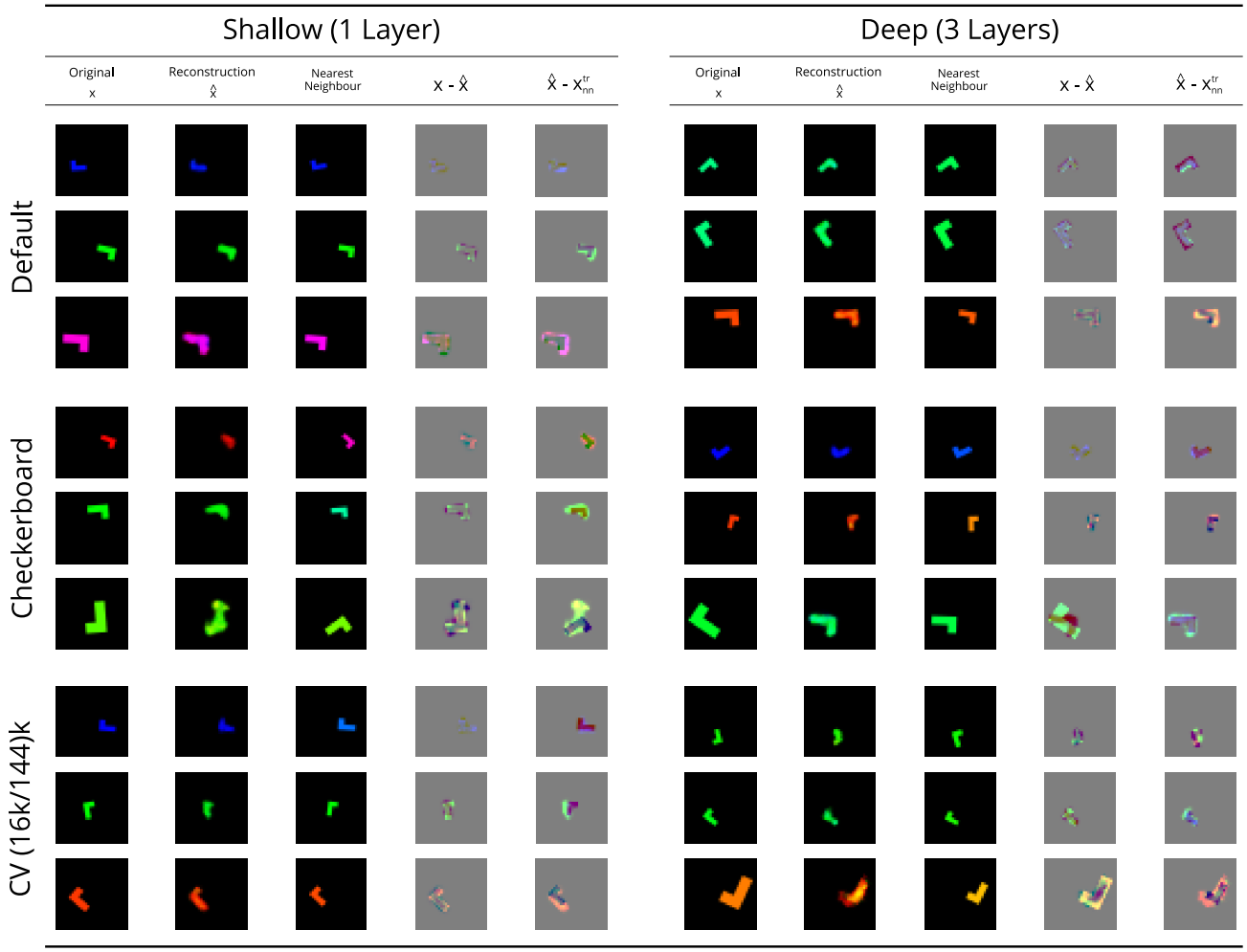


Figure A10: Reconstruction of test samples from all datasets for 1-layer and 3-layer VAEs. Rows show examples with increasing reconstruction loss, randomly selected from the 10^{th} (*top*), 40^{th} to 60^{th} (*middle*), and 90^{th} (*bottom*) percentiles.

benefit by using models with higher capacity. By either increasing or decreasing β, we see a drop in LM and ELBO as the training objective becomes different than the metric. The results for the CV (16k/147k) and Checkerboard

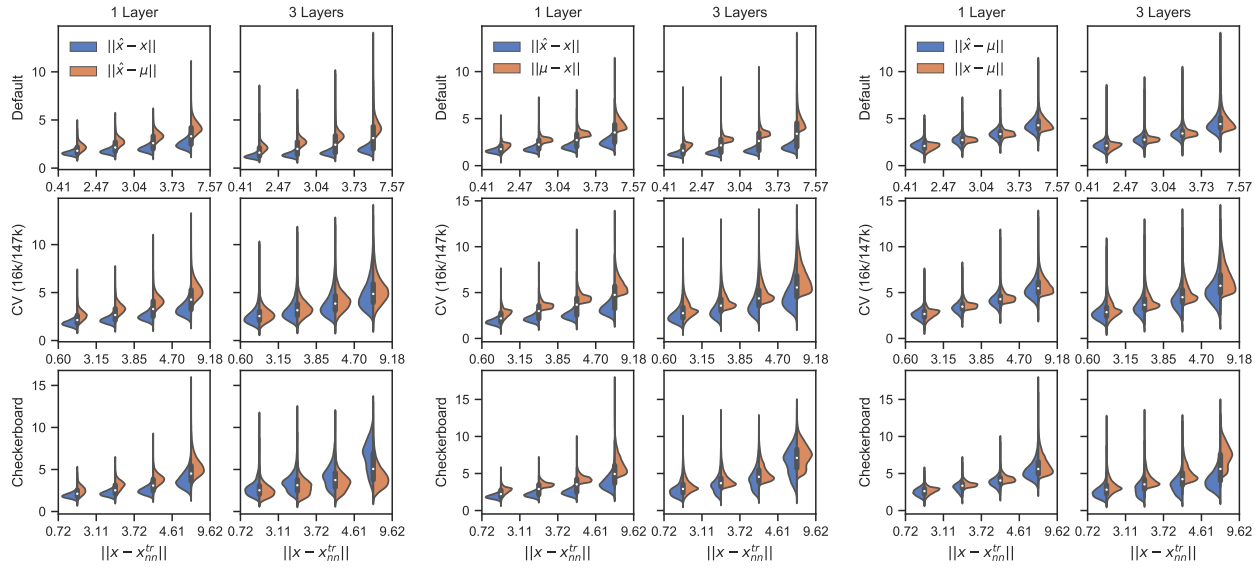


Figure A11: Distributions of distance between test data and output of the decoder ($\|\hat{x} - x\|$), distance between test data and output of the infinite capacity decoder ($\|\mu - x\|$, μ from Equation (7)), and distance between test data and and output of the infinite capacity decoder ($\|\hat{x} - \mu\|$), partitioned according distance to nearest training sample ($\|x - x^{tr}_{nn}\|$) into 4 bins of equal sizes. Values in x-axis are the limits of the bins.

splits however are very different. For the 1-layer case, we still see that β values near 1 yield the highest ELBO and LM. In the 3-layers case however, see that decreasing β can significantly *improves* generalization[3]. In fact we see that for $\beta = 0.1$, the 3-layer VAEs achieve nearly the same LM and ELBO as the 1-layer case.

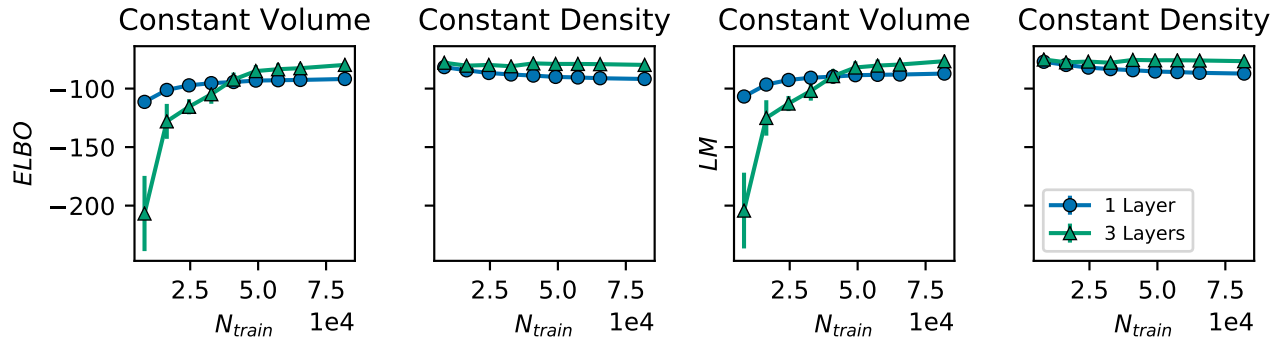


Figure A12: Training set size versus test ELBO and LM, for CD and CV splits of Tetrominoes dataset with 1-layer and 3-layer VAEs (β=1), averaged over 5 independent restarts.

[3]In the Checkerboard split, the test set is out-of-domain, therefore one can argue that lower LM is considered to be a better result.

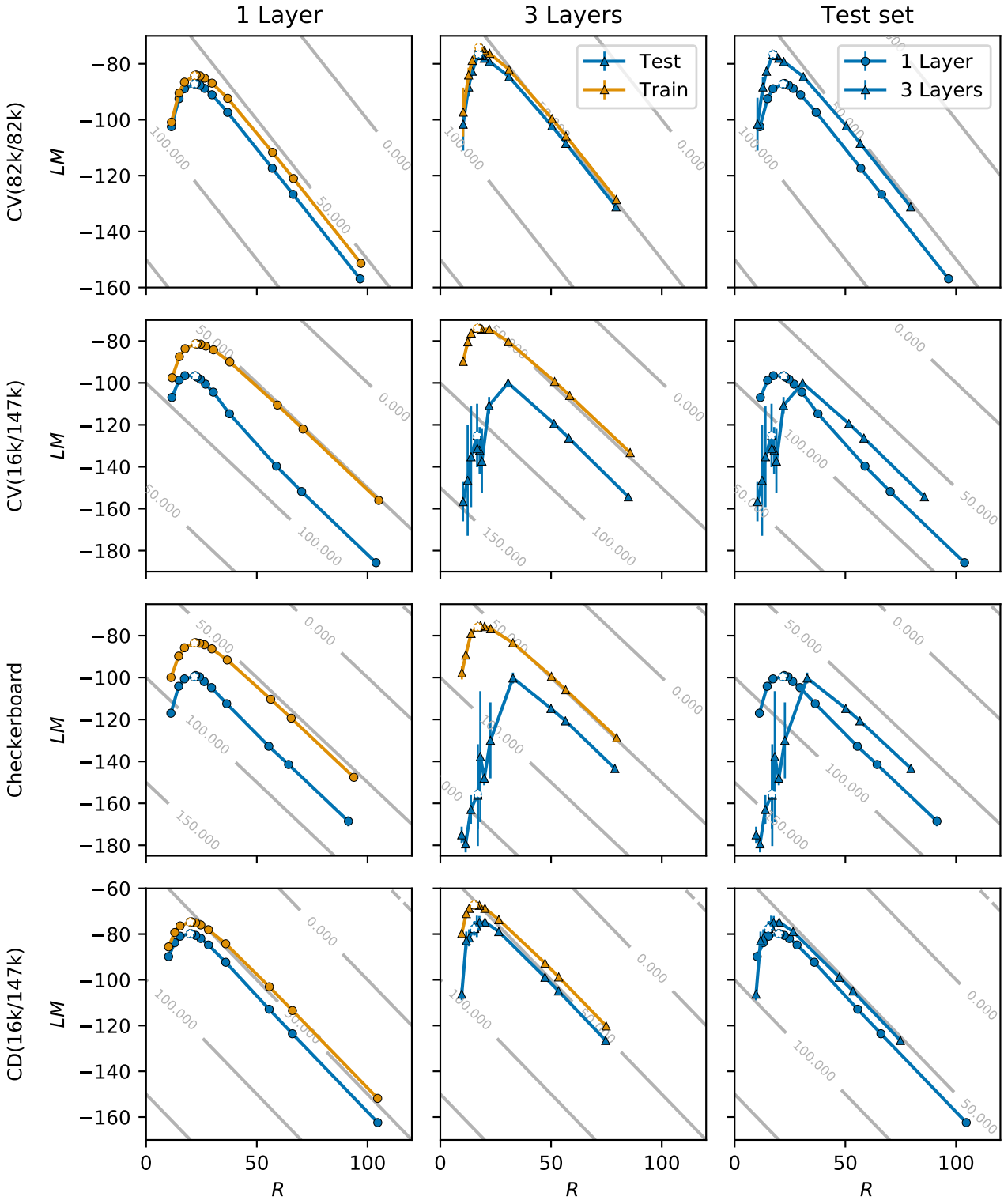


Figure A13: Log marginal likelihood and rate evaluated on training and test sets for CV(82k/82k) (top row), CV(16k/147k) (2nd row), Checkerboard (3rd row), and CD(16k/147k) (bottom row) splits, trained with 1-layer and 3-layer models. Each dot constitutes a β value (white stars indicate the β=1 point), averaged over 5 independent restarts.

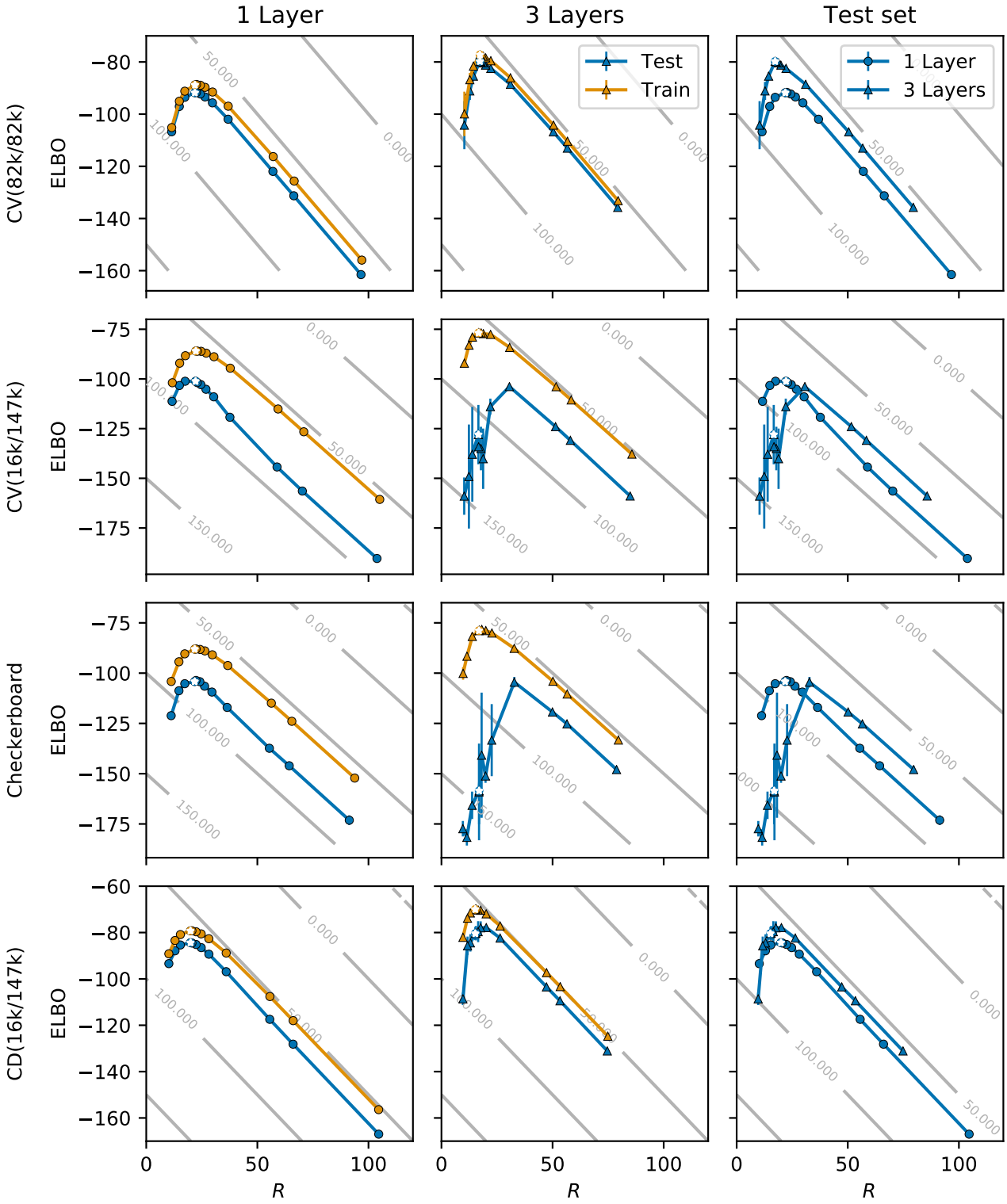


Figure A14: ELBO and rate evaluated on training and test sets for CV(82k/82k) (top row), CV(16k/147k) (2nd row), Checkerboard (3rd row), and CD(16k/147k) (bottom row) splits, trained with 1-layer and 3-layer models. Each dot constitutes a β value (white stars indicate the β=1 point), averaged over 5 independent restarts.

A3.5 Norm of the Weights for Decoder and Encoder

Regularizers are typically terms that encourage the learning algorithm to choose simpler models. As a means of gaining intuition, we look at the average norms of the weights of both the encoder and the decoder (see Figure A15). We observe that as we increase β (i.e. rate penalty), the average norm of the decoder weights increase. In other words, having a low rate penalty results in learning not simple, but more complex decoders. This suggest that the rate not only does not act as a regularizer, but in fact it has the opposite effect.

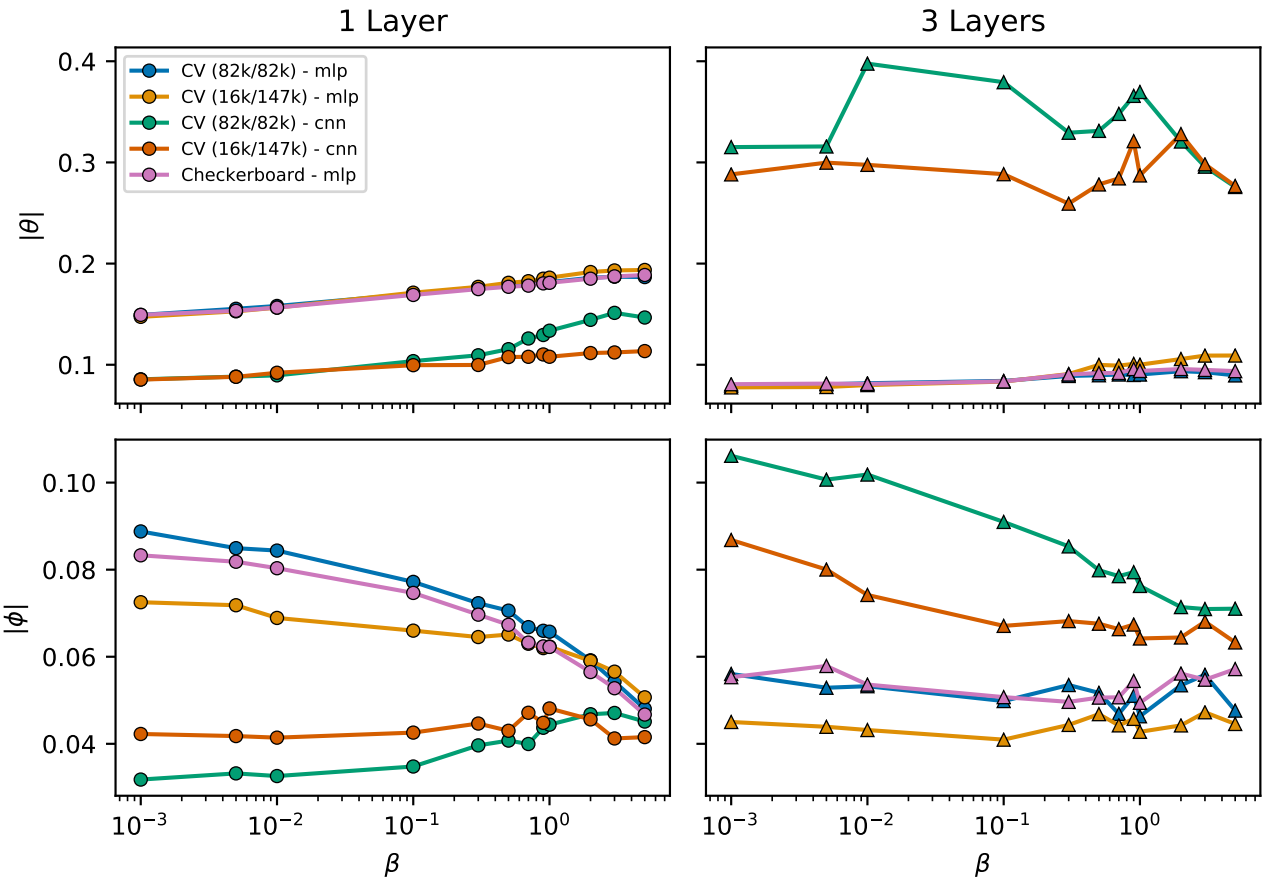


Figure A15: Norm of the weights of decoder (*top*), and encoder (*bottom*) when trained with different values of β on various splits trained with either MLPs or CNNs.

A3.6 Role of Mutual Information vs. the Marginal KL

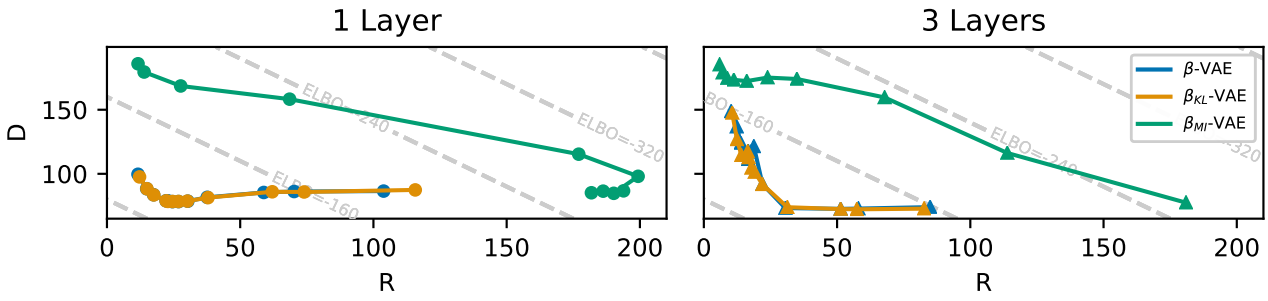


Figure A16: Test *RD* curves yielded by models trained with $\mathcal{L}_\beta$, $\mathcal{L}_{\text{KL}}$, $\mathcal{L}_{\text{MI}}$ objectives for 1-layer (*Left*) and 3-layer (*Right*) MLP architectures.

In order to understand the individual impact of the MI vs. the marginal KL on the reconstruction, we report on the following analysis. We can write a more general form of β-VAE objective where the MI and the marginal KL terms can have different coefficients:

$$\mathcal{L}_{\beta_{\text{MI}},\beta_{\text{KL}}}(\theta,\phi) = -D - \beta_{\text{MI}}\, I_q(\boldsymbol{z};\boldsymbol{x}) - \beta_{\text{KL}}\, \text{KL}(q_\phi(\boldsymbol{z}) \,\|\, p(\boldsymbol{z})),$$

and $\beta_{\text{MI}} = \beta_{\text{KL}} = \beta$ recovers the original β-VAE objective. We are interested in cases where β_{MI}=0, where effect of the MI is nullified or β_{KL}=0, where effect of the KL term is nullified. For emphasis, we refer to the objective with β_{KL}=0 as $\mathcal{L}_{\text{MI}}$, and the objective with β_{MI}=0 as $\mathcal{L}_{\text{KL}}$.

$$\mathcal{L}_{\text{MI}}(\theta,\phi) = -D - \beta_{\text{MI}}\, I_q(\boldsymbol{x},\boldsymbol{z})$$
$$\mathcal{L}_{\text{KL}}(\theta,\phi) = -D - \beta_{\text{KL}}\, \text{KL}(q_\phi(\boldsymbol{z}) \,\|\, p(\boldsymbol{z}))$$

To ensure that it is the marginal KL that is causing the shape shift in the *RD* curve, we trained VAEs with 1-Layer and 3-Layer architectures using objectives $\mathcal{L}_{\text{MI}}$[4] and $\mathcal{L}_{\text{KL}}$[5], with 5 random restarts on the CV(16k/147k) split.

Figure A16 shows the test *RD* curves for VAEs trained with $\mathcal{L}_\beta$, $\mathcal{L}_{\text{MI}}$, and $\mathcal{L}_{\text{KL}}$ objectives. The *RD* curve for the β-VAE and the $\mathcal{L}_{\text{KL}}$ objective lie almost on top of each other, while the *RD* curve for the $\mathcal{L}_{\text{MI}}$ is vastly different. These results confirm that it is indeed the marginal KL term that is responsible for impacting the generalization performance of VAEs.

[4] $\beta_{\text{MI}} \in \{6., 8., 10., 11., 12., 13., 14., 15., 17., 20.\}$
[5] $\beta_{\text{KL}} \in \{0.001, 0.005, 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1., 2., 3., 5.\}$

A3.7 Early-Stopping

In the previous experiments, we trained all models for a fixed number of 256k iterations (corresponding to 400 epochs for the CV (82k/82k) split). Here we consider applying early stopping in order to test whether the U-shaped *RD* curve we observed in the 1-layer VAEs is due to overfitting. The results are shown in Figure A17. Looking at the MLP architecture (left column), we see that early stooping does not make a strong impact in the default split. In the CV (16k/147k) split however, we see that early stopping improves generalization; shifting the curve to bottom left. Moreover, we see that U-shaped curve in the 1-layer case turns into an L-shaped when we apply early-stopping. This suggests that 1-layer VAEs in low β regime were indeed overfitting as this can be mitigated by applying early stopping.

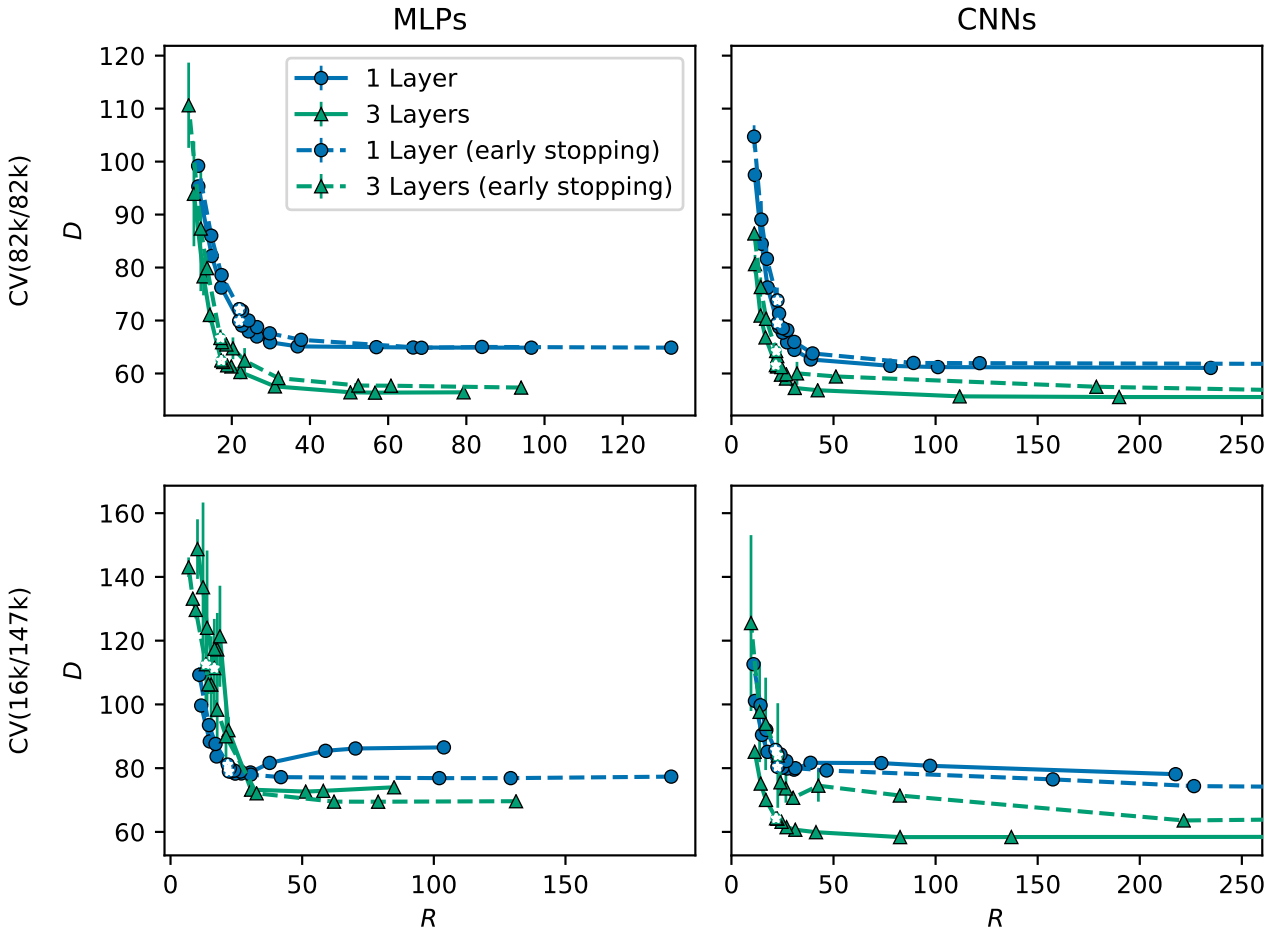


Figure A17: RD curves evaluated on the test set for 1-layer and 3-layer VAEs with MLP and CNN architectures trained with and without early stopping.

A3.8 Additional Datasets

Here we provide results for additional datasets. In Figure A18, we show the the histogram of ℓ_2 distance between test examples and their nearest training neighbour for a large CV split (50% train/test ratio) and a small CV split (10% train/test ratio). In Figure 7, we only show the results for large one particular CV split. Here, we additionally show the results for other CV splits ranging from 5% to 75% of all the data (Figure A20). For all datasets, we aimed to find train/test ratios where the 3-layer *RD* curve would be above and below the 1-layer curve.

In Figure A20, we see that the *RD* curve for 1-layer models resembles a U-shaped curve in most cases as the we decrease the number of training data while for 3-layer models this is not the case. We also observe the *RD* curves for 3dShapes is very different compared to others. In particular, some of the *RD* curves for 3-layer networks resemble a U-shaped curve. We suspect that this is because the 3dShapes datasets is a more difficult datasets compared to others both due to its size and certain factors of variations such as background color.

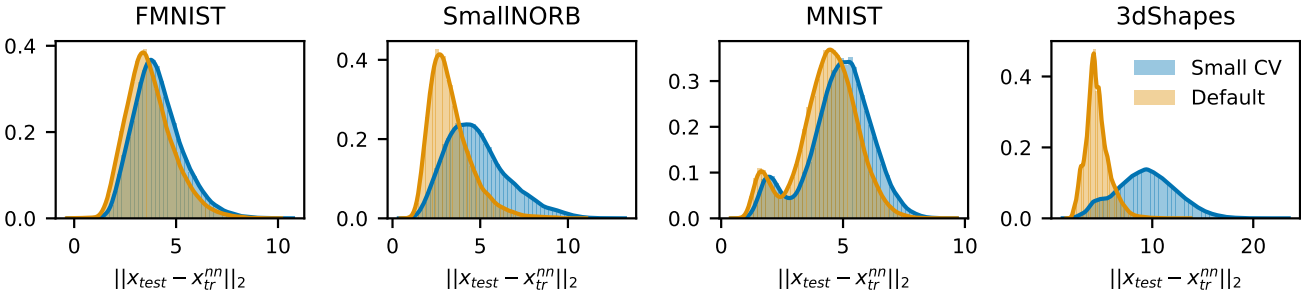


Figure A18: Normalized histograms of ℓ_2 distance between test examples $\boldsymbol{x}$ and the nearest neighbour in the training set $\boldsymbol{x}_{nn}^{tr}$. for other datasets. For details of 'Default' and 'Small CV' see Table A1.

A3.9 Robustness

We also evaluate generalization based on the variance (in the context of bias-variance trade-off) by computing the difference between test set distortion and training set distortion, for CV(82k/82k) and CV(26k/147k) datasets (Figure A19). A small difference is an indication of robustness (low variance) while a large difference is an indication of overfitting. We observe two opposite patterns, that are consistent across datasets. Increasing β decreases the difference for 1-layer VAEs, while increasing the difference for 3-layer VAEs. In fact, when β is low, the difference in distortion between the training and test sets is lower than the 1-layer model.

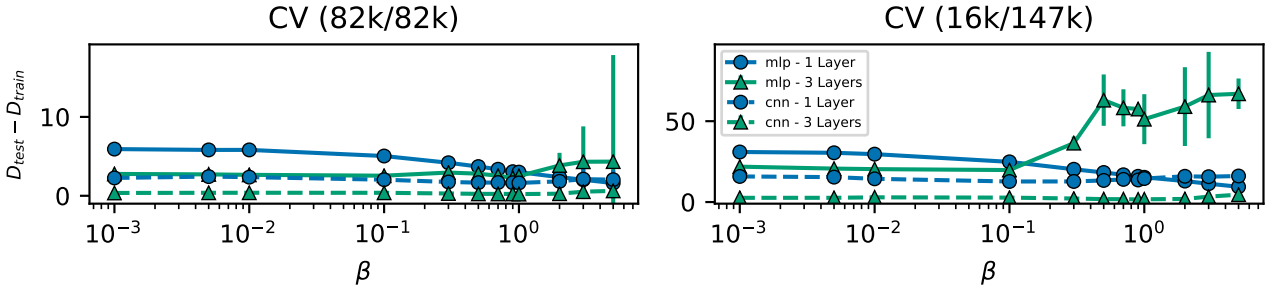


Figure A19: Difference between test distortion and train distortion for VAEs trained with MLPs and CNNs on Tetrominoes dataset.

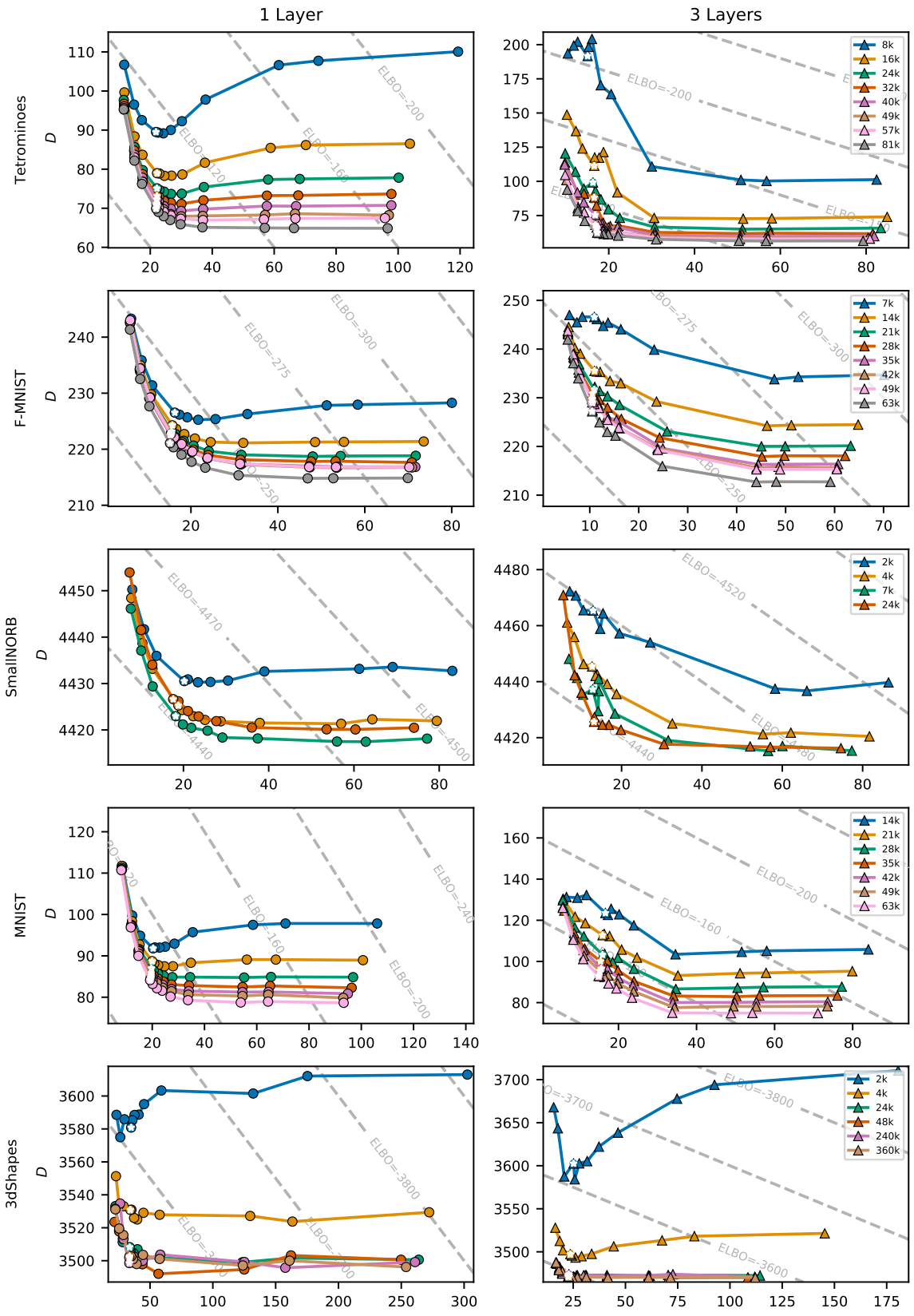


Figure A20: *RD* curves shown on various datasets with different CV splits trained with 1 and 3 layers.

A3.10 Additional Figures

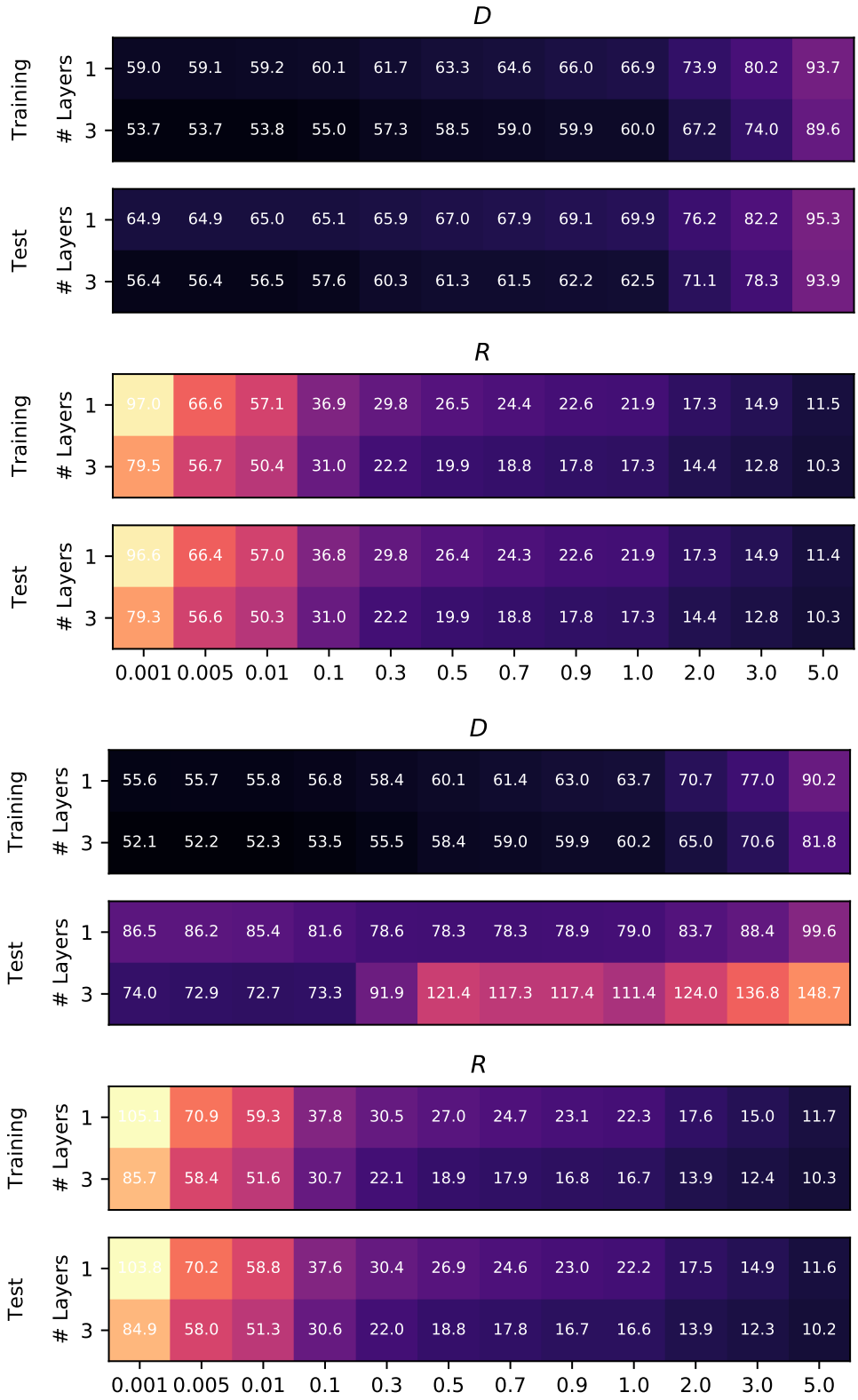


Figure A21: RD values on evaluated on the test set for different values of β. The top and bottom two maps are for CV (82k/82k) the CV (16k/147) splits respectively.

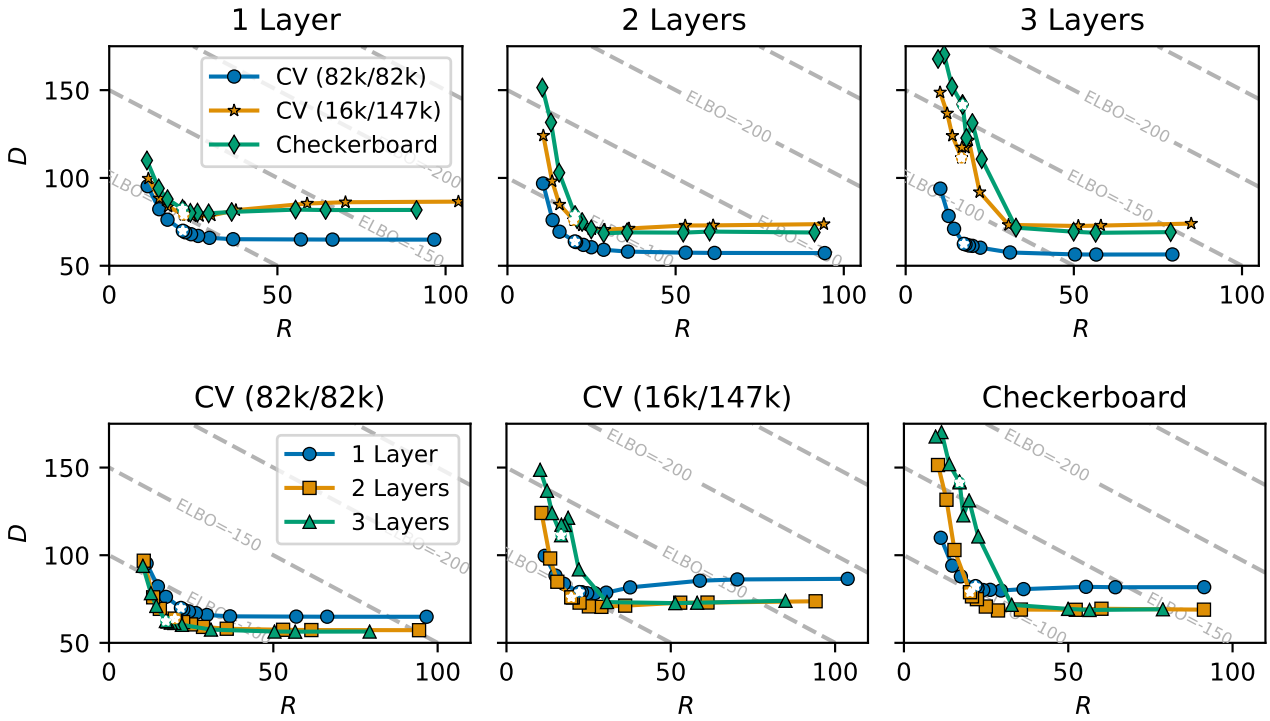


Figure A22: *RD* curves shown in two alternative views. The first view compares the effect of network depth for datasets with different levels of difficulty (*Top*). The second view is the effect of making the generalization problem more difficult in models with different capacity (*bottom*).

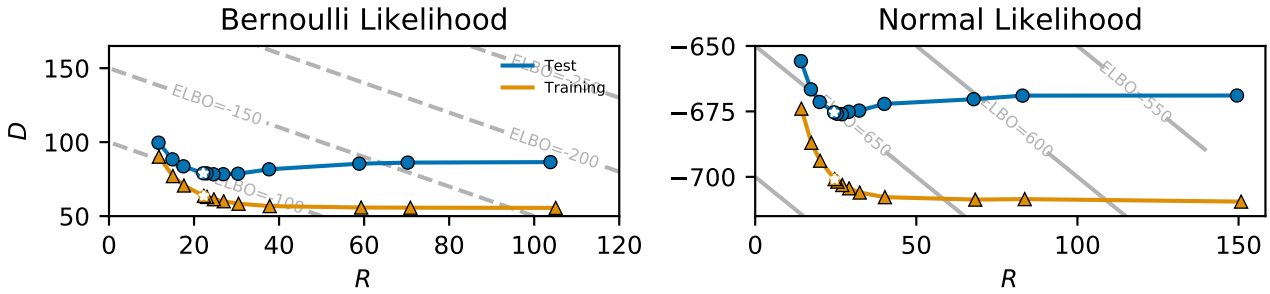
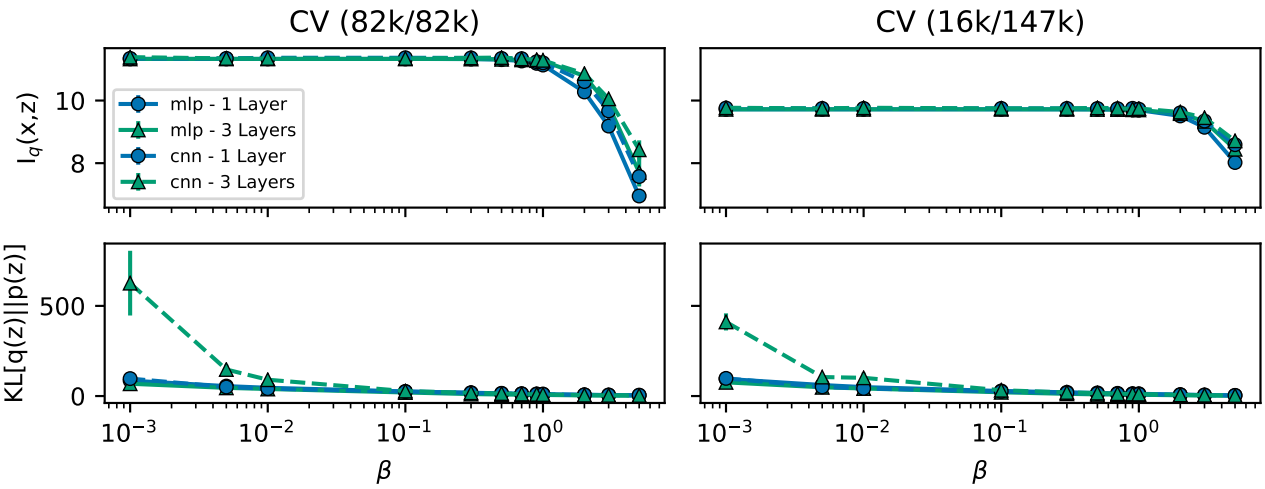
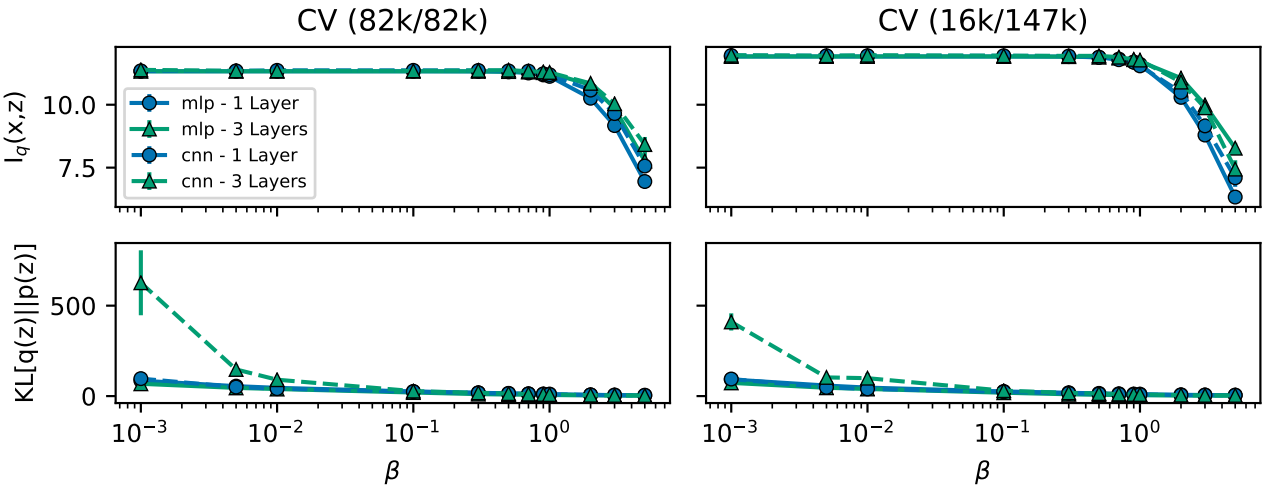


Figure A23: *RD* curves shown for CV (16k/147k) split for VAE with MLP architecture trained with a Bernoulli likelihood and a Gaussian likelihood (σ^2 fixed to 0.1) .



(a) Training set



(b) Test set

Figure A24: $I_q(\boldsymbol{x}, \boldsymbol{z})$ and $\mathrm{KL}(q_\phi(\boldsymbol{z}) \,\|\, p(\boldsymbol{z}))$ for β-VAE with MLP and CNN architectures, trained on CV(82k/82k) and CV(16k/147k) with different β-values.

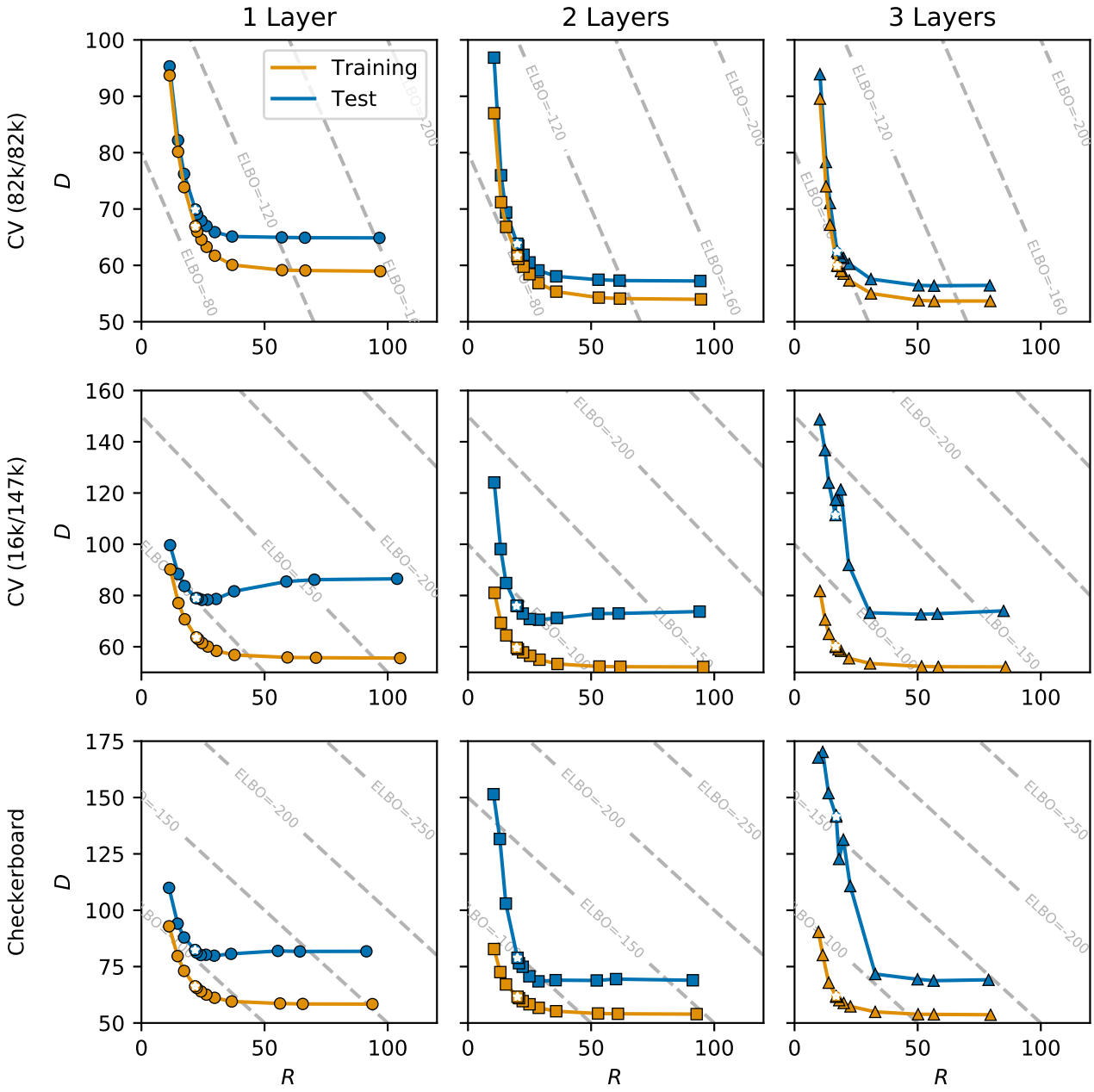


Figure A25: *RD* curves on training and test set with the Default, CV (16k/147k), and Checkerboard splits in models with 1, 2, and 3 layers.

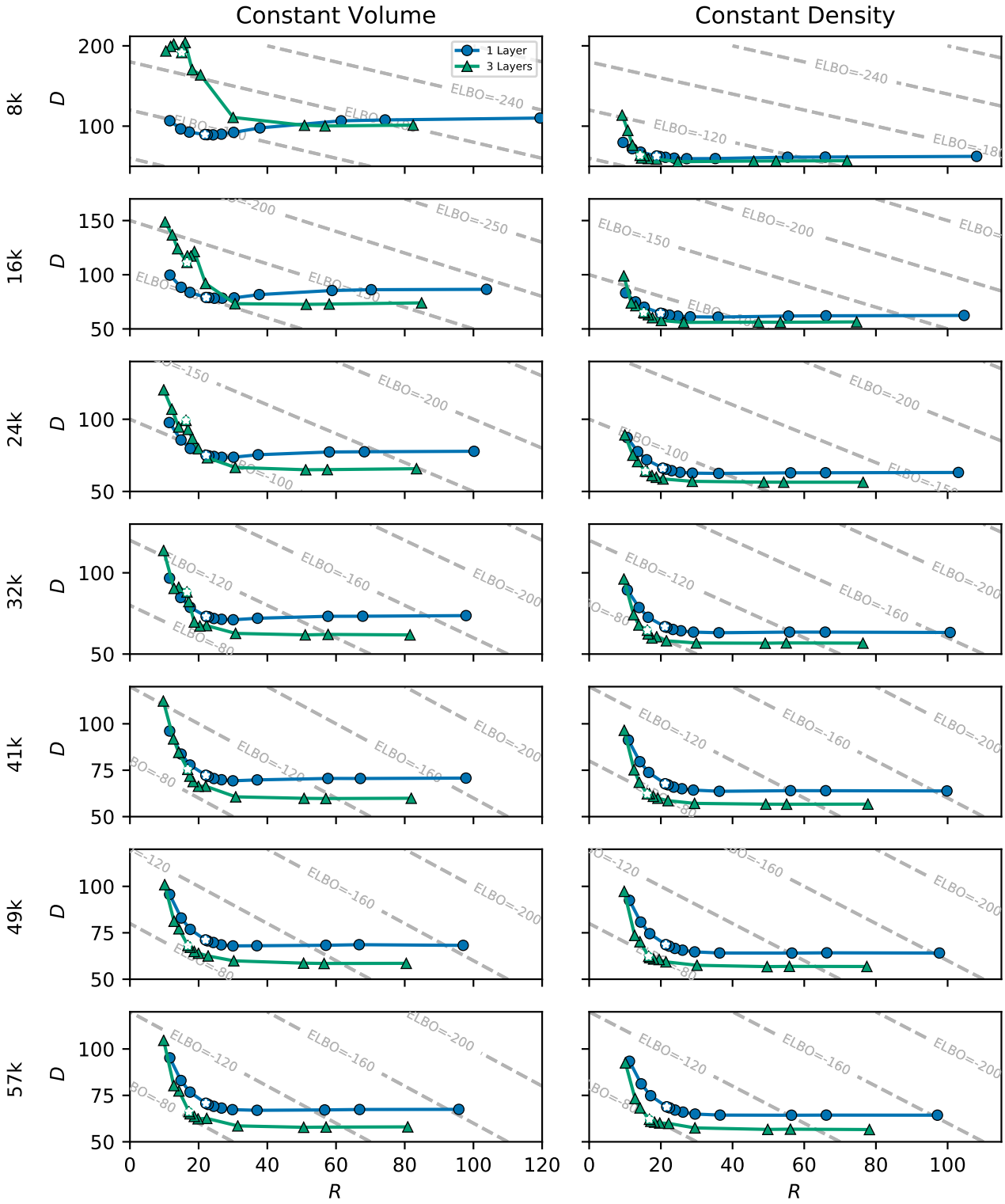


Figure A26: Estimated RD curves for different number of training data for when the volume is kept constant constant and reduce to keep the density the same. The uppermost panel is ($N_{\text{train}} = 8k$) and the lowest is for ($N_{\text{train}} = 57k$)

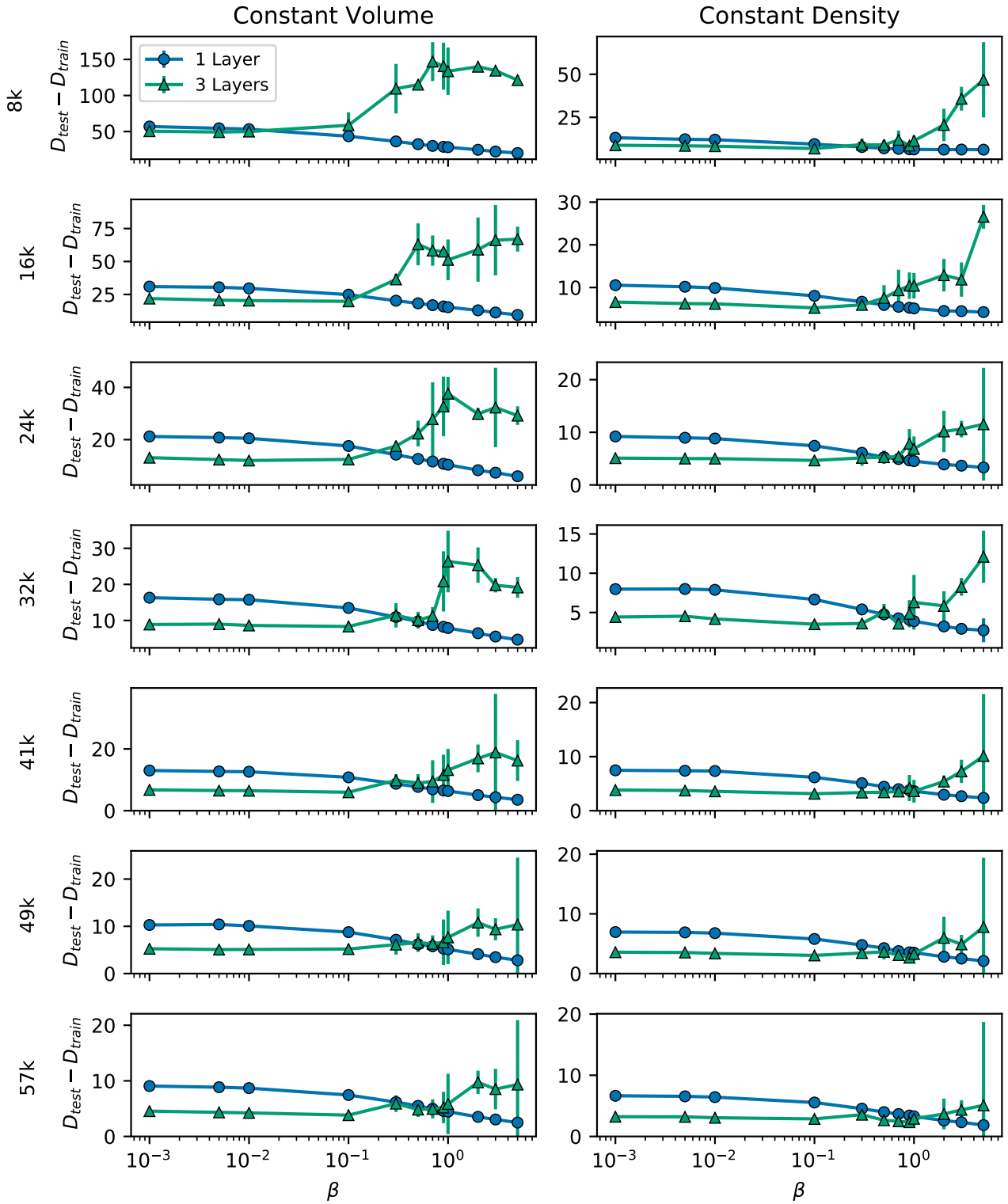


Figure A27: Difference between test distortion and train distortion for CV and CD splits of Tetrominoes dataset with different N_{train}.

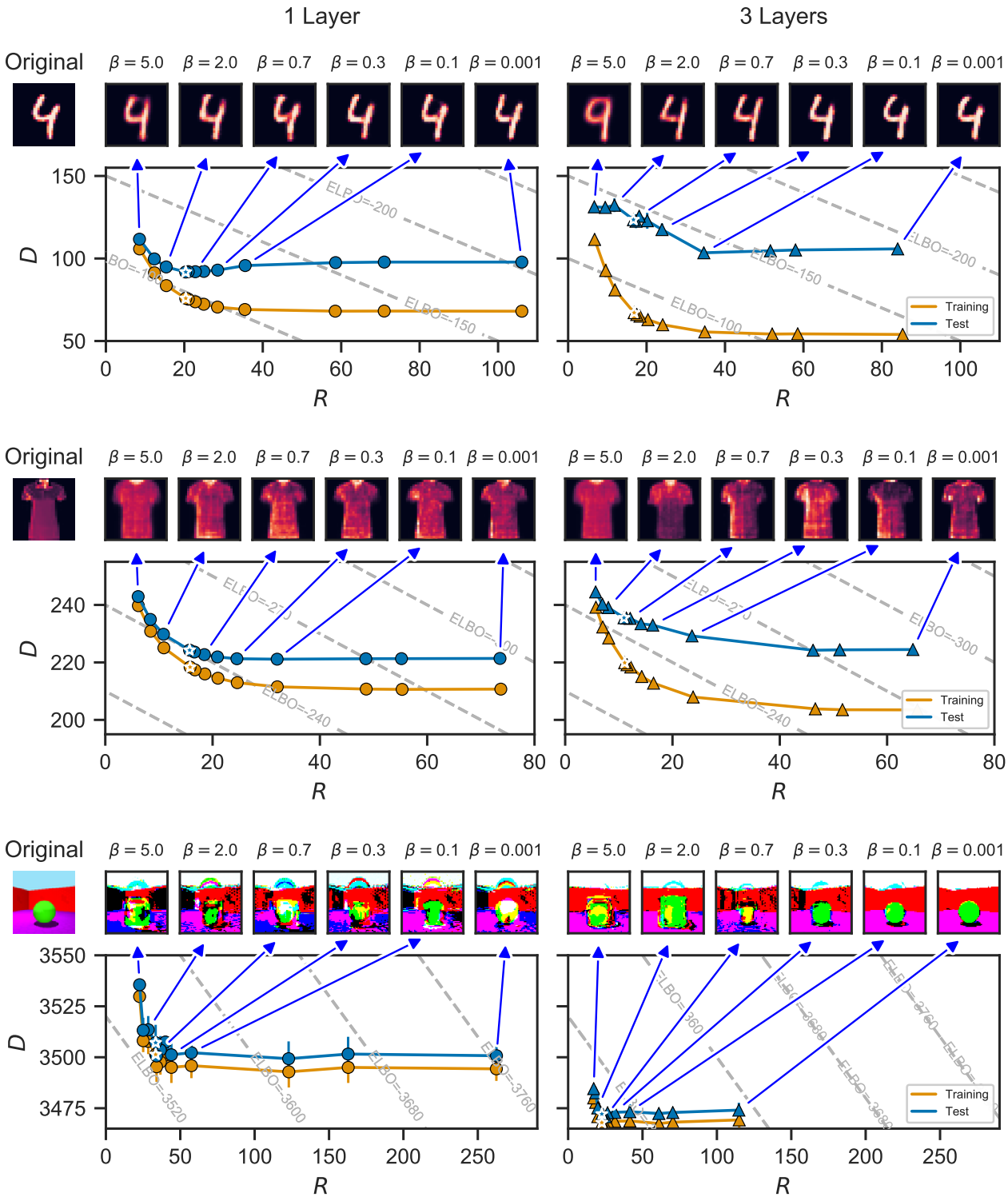


Figure A28: Training and test *RD* curves for a 1-layer (left) and a 3-layer (right) architecture for different datasets. Each dot constitutes a β value (white stars indicate the β=1), averaged over 5 restarts. Images show reconstructions of a test example. The splits are CV(14k/56k), CV(7k/63k), and CV(48k/432k) for MNIST, Fashion-MNIST, and 3dShapes respectively.

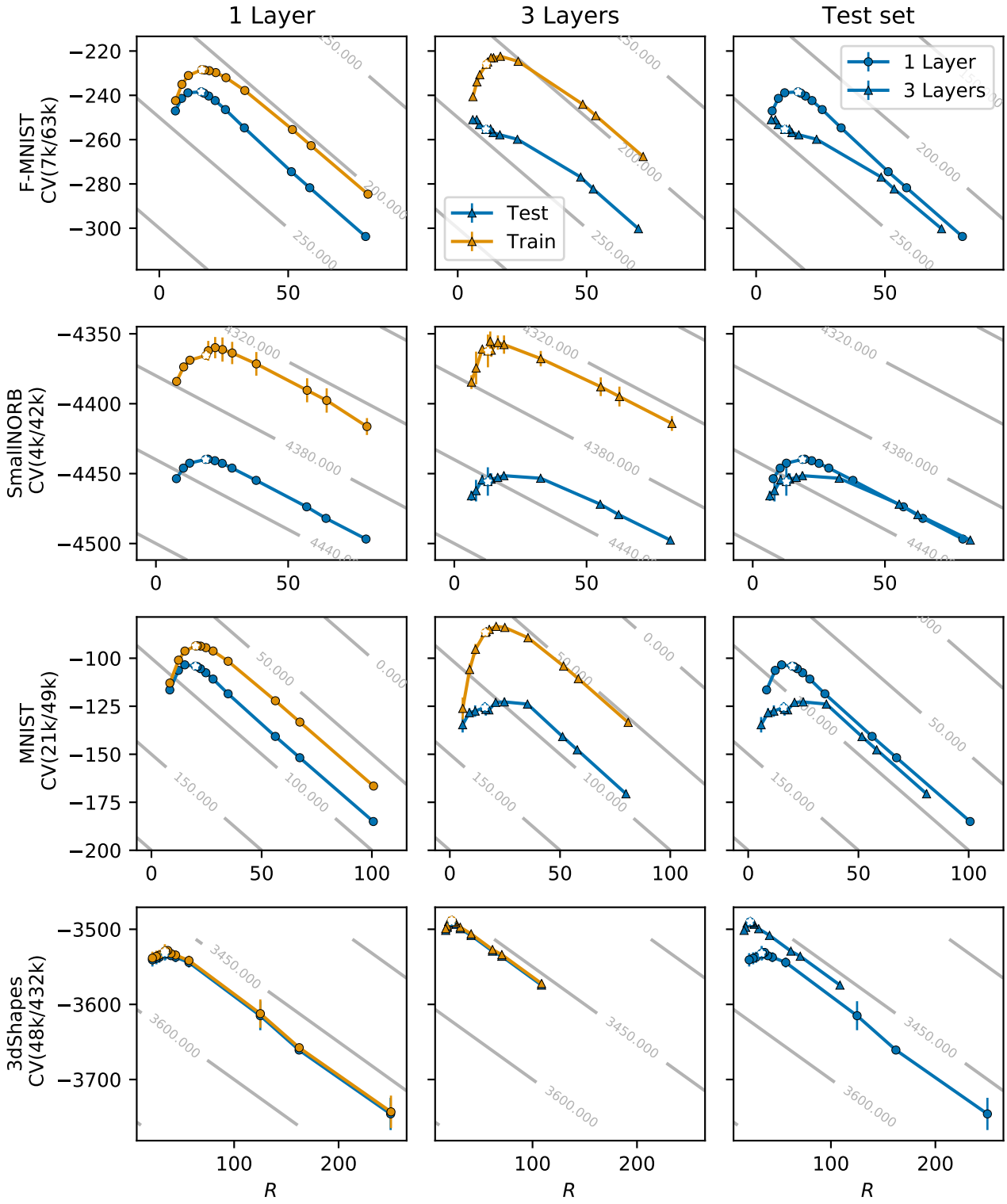


Figure A29: Log marginal likelihood and rate evaluated on training and test sets for different datasets, trained with 1-layer and 3-layer models. Each dot constitutes a β value (white stars indicate the β=1 point), averaged over 5 independent restarts.

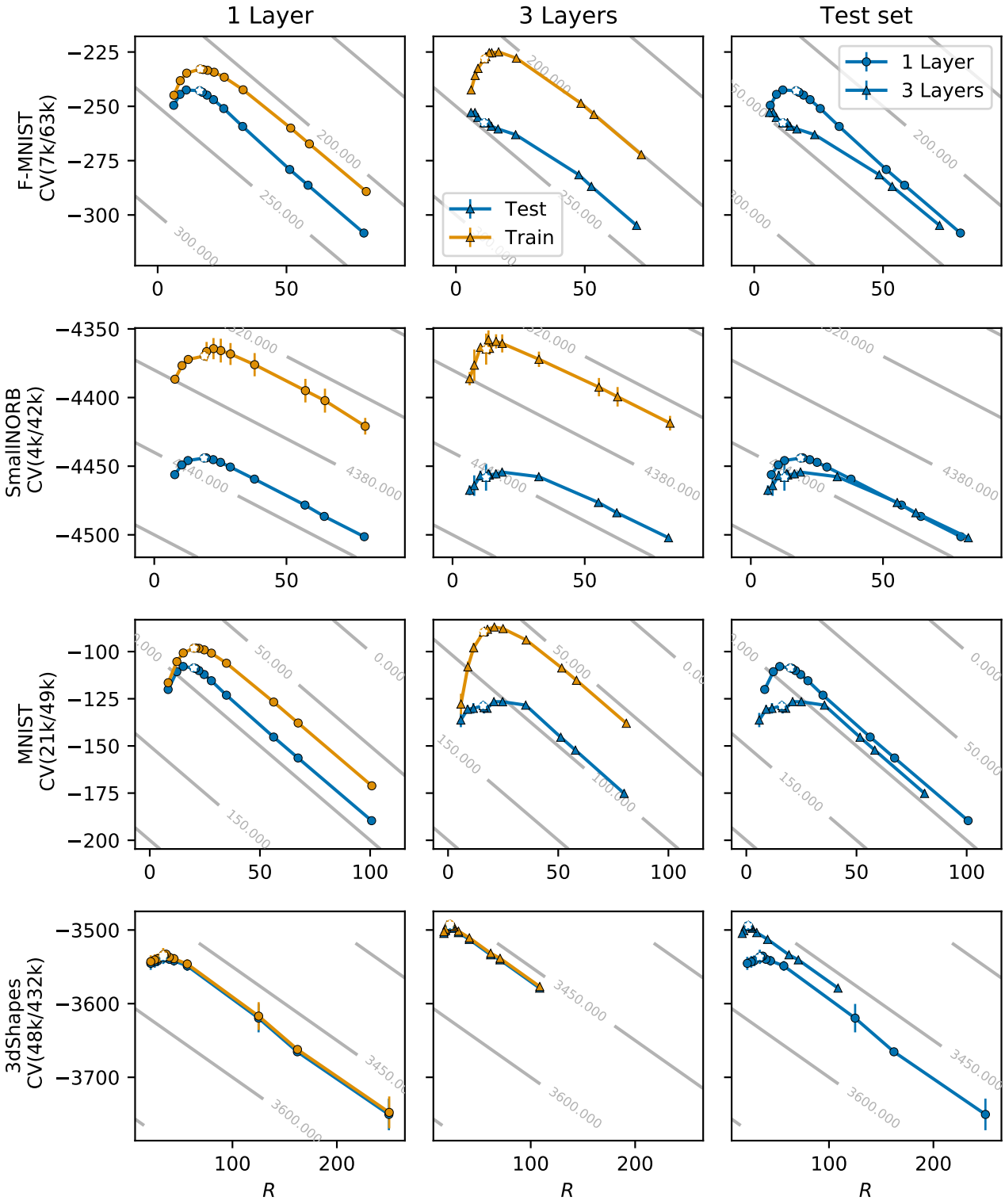


Figure A30: ELBO and rate evaluated on training and test sets for different datasets, trained with 1-layer and 3-layer models. Each dot constitutes a β value (white stars indicate the β=1 point), averaged over 5 independent restarts.

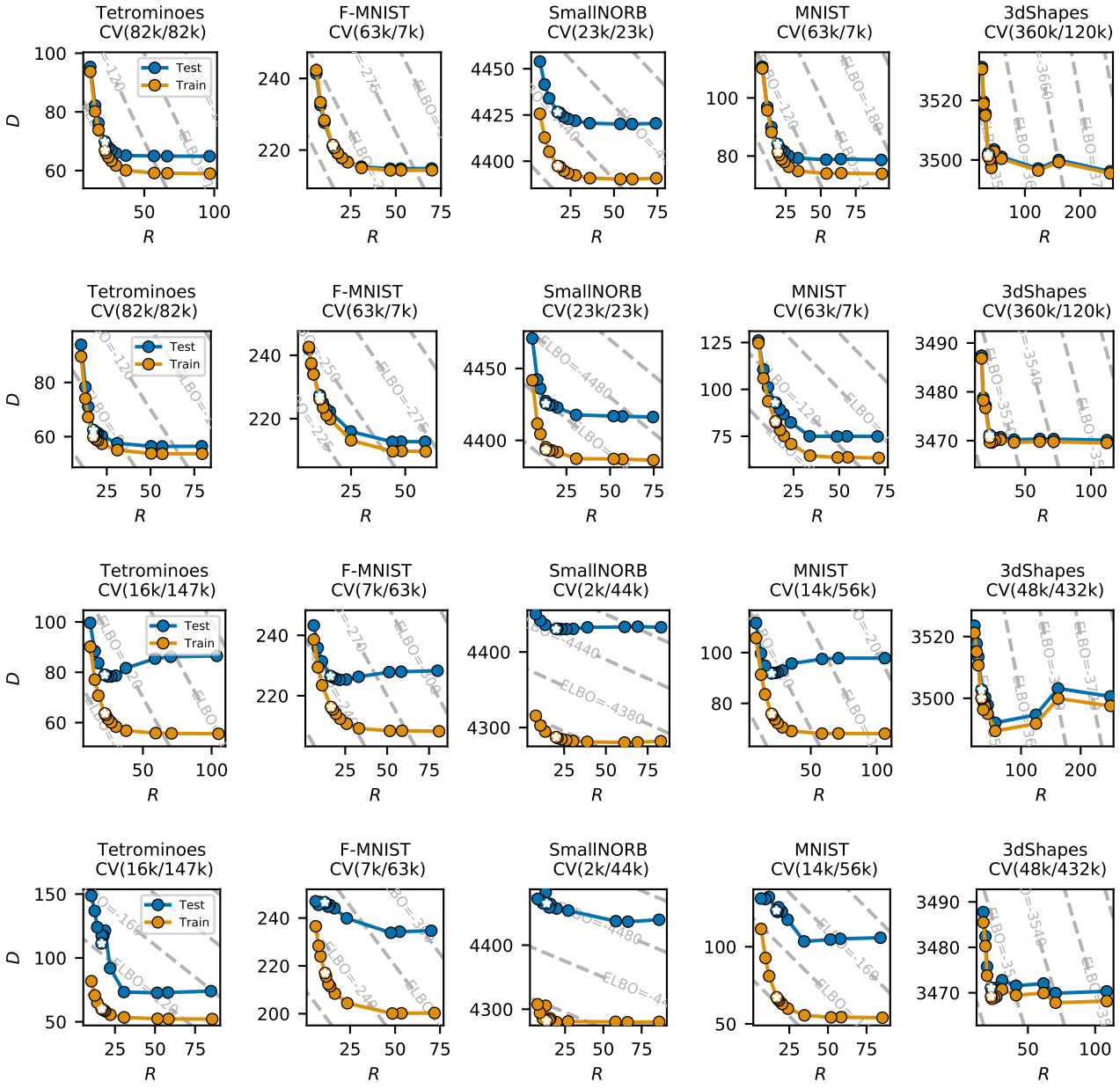


Figure A31: *RD* curves evaluated on training and test set for different datasets for large CV split trained with 1-layer (1st from top), large CV split trained with 3-layer (2nd from top), small CV split trained with 1-layer (2nd from bottom), small CV split trained with 3-layer (1st from bottom).