A GENERALIZATION METRICS

In this section, we provide definitions and details on the various metrics considered in our analysis. We begin with scale metrics, and then consider shape metrics obtained from the ESDs of the weight matrices. Although our focus is on generalization metrics that do not need data to evaluate, we also define generalization metrics based on margin (Bartlett et al., 2017; Pitas et al., 2017) and PAC-Bayesian bounds (McAllester, 1999; Neyshabur et al., 2018).

A.1 NOTATION AND PRELIMINARIES

General notation. As before, we consider a NN with d layers and corresponding weight matrices $\mathbf{W}_1$, $\mathbf{W}_2$,..., $\mathbf{W}_d$. We use $\mathbf{W}$ to denote the collection of all the weights and denote the vector that consists of all the model weights as $\text{vect}(\mathbf{W})$. The neural network (as a function) is denoted by $f_{\mathbf{W}}$, taking a single input sample $\mathbf{x}$ and outputs a vector $f_{\mathbf{W}}(\mathbf{x})$. The superscript $^{\text{init}}$ on a weight matrix, e.g. $\mathbf{W}_1^{\text{init}}$, denotes the initial weights from which the model is trained. The notation $\mathbf{1}$ means an all-one vector, and $\mathbf{I}$ means the identity matrix.

Norms and distances. We use different types of norms defined on vectors and matrices. $\|\cdot\|_2$ and $\|\cdot\|_1$ used on vectors respectively means the ℓ_2 norm and the ℓ_1 norm. $\|\cdot\|_F$ and $\|\cdot\|_2$ used on matrices respectively denotes the Frobenius norm and the spectral norm (which is the induced ℓ_2 norm).

A.2 SCALE METRICS

Norm-based and distance-based metrics. In the following, we discuss multiple metrics obtained from the norms of the weights or the distance between the final weights and those at initialization. While some metrics are averaged over layers, and others are not, this inconsistency is not in error. We follow definitions of metrics from several prior papers *verbatim*. Results in the second task (comparing model performance across a single training run) are independent of these factors. However, to compare networks with different sizes, proper normalization is necessary. Some metrics across the literature are also linearly dependent on others, and are therefore redundant for comparison. For example, `log_prod_of_spec` and `log_sum_of_spec` from Jiang et al. (2019) overlap with `log_spectral_norm` from Martin & Mahoney (2021a), and `log_sum_of_fro` and `log_prod_of_fro` from Jiang et al. (2019) overlap with `log_norm` from Martin & Mahoney (2021b). These metrics are not considered.

- (`param_norm`). The squared Frobenius norm summed over all weight matrices.

$$\mu_{\texttt{param\_norm}} = \sum_{i=1}^{d} \|\mathbf{W}_i\|_F^2. \tag{4}$$

- (`fro_dist`). The distance between a weight matrix and its initilized value, calculated using the Frobenius norm and summed over all layers.

$$\mu_{\texttt{fro\_dist}} = \sum_{i=1}^{d} \|\mathbf{W}_i - \mathbf{W}_i^{\text{init}}\|_F^2. \tag{5}$$

- (`log_norm`).

$$\mu_{\texttt{log\_norm}} = \frac{1}{d}\sum_{i=1}^{d} \log \|\mathbf{W}_i\|_F^2. \tag{6}$$

- (`log_spectral_norm`).

$$\mu_{\texttt{log\_spectral\_norm}} = \frac{1}{d}\sum_{i=1}^{d} \log \|\mathbf{W}_i\|_2^2. \tag{7}$$

- (`dist_spec_int`).

$$\mu_{\texttt{dist\_spec\_int}} = \sum_{i=1}^{d} \|\mathbf{W}_i - \mathbf{W}_i^{\text{init}}\|_2^2. \tag{8}$$

- (path_norm). The metric is introduced in Neyshabur et al. (2015). To calculate the metric, we square the parameters of the network, perform a forward pass on an all-ones input and then compute the sum of the network outputs.

$$\mu_{\texttt{path\_norm}} = \|f_{\mathbf{W}^2}(\mathbf{1})\|_1 . \tag{9}$$

Scale metrics that require more shape information from the ESDs. The following metrics require more than just a single type of norm, instead involving a combination of a norm with other factors.

- (mp_softrank). This metric is introduced in Martin & Mahoney (2021b). To calculate this metric, we fit the MP distribution on the ESD, obtain the bulk max of the MP distribution and then divide by the maximum eigenvalue.

$$\mu_{\texttt{mp\_softrank}} = \frac{1}{d}\sum_{i=1}^{d} \lambda_{i,\text{MP}}/\lambda_{i,\max}. \tag{10}$$

- (stable_rank). The metric is a norm-adjusted measure of the scale of the ESD.

$$\mu_{\texttt{stable\_rank}} = \frac{1}{d}\sum_{i=1}^{d} \|\mathbf{W}_i\|_F^2/\|\mathbf{W}_i\|_2^2 \tag{11}$$

A.3 Shape metrics

Tail-exponent fitting. The following metrics are derived from heavy or light-tailed fits to the ESD.

- (PL_alpha). The slope of the tail of the ESD, on a log-log scale. We use MLE from Alstott et al. (2014) to estimate PL_alpha. The distribution of eigenvalues is assumed to have the form of (1).
- (E_TPL_lambda). The tail exponent of the E-TPL fit to the ESD. This is a novel generalization metric introduced in this work.
- (EXP_lambda). The tail exponent of the EXP fit to the ESD, under the assumption that the ESD follows an exponential distribution (3). This is also a novel generalization metric introduced in this work.
- (PL_ks_distance). The Kolmogorov-Smirnoff (KS) goodness-of-fit test statistic for the PL fit:

$$\mu_{\texttt{ks\_distance}} = \frac{1}{d}\sum_{i=1}^{d} \sup_x |F_i^*(x) - S_i(x)|, \tag{12}$$

  where $F_i^*(x)$ is the distribution of the estimated PL fit to the ESD, and $S_i(x)$ is the ESD itself.
- (E_TPL_ks_distance). The KS test statistic for the E-TPL fit, defined in the same way as (12), except that $F_i^*(x)$ is the distribution of the estimated E-TPL fit to the ESD.

A.4 Hybrid metrics

The following metrics are scaled versions of PL_alpha, involving both shape information from PL_alpha and scale information from other weighted norms. Let α_i denote the estimated PL coefficient of the ESD of the i-th weight matrix $\mathbf{W}_i$. Recall that $\lambda_{i,\max}$ is the largest eigenvalue of $\mathbf{W}_i$.

- (alpha_weighted). A scale-adjusted form of PL_alpha. This metric is denoted as $\hat{\alpha}$ in Martin & Mahoney (2021a,b); Martin et al. (2021).

$$\mu_{\texttt{alpha\_weighted}} = \frac{1}{d}\sum_{i=1}^{d} \alpha_i \log \lambda_{i,\max}. \tag{13}$$

- (log_alpha_norm). This metric is another scale-adjusted PL_alpha metric in the form of a Schatten norm. Recall that we let $\{\lambda_j\}_{j=1}^{M}$ denote the set of eigenvalues of the correlation matrix $\mathbf{X}_i = \mathbf{W}_i^\top \mathbf{W}_i$, where $\mathbf{W}_i$ is the N-by-M weight matrix that satisfies $N \geq M$. Then, we can define the Schatten p-norm as

$$\|\mathbf{X}_i\|_p = \left(\sum_{j=1}^{M} \lambda_j^p\right)^{\frac{1}{p}}. \tag{14}$$

The metric `log_alpha_norm` is given by

$$\mu_{\texttt{log\_alpha\_norm}} = \frac{1}{d}\sum_{i=1}^{d} \log \|\mathbf{X}_i\|_{\alpha_i}^{\alpha_i}. \tag{15}$$

A.5 MARGIN-BASED METRICS

Finally, we discuss generalization metrics derived from margins. Recalling that $f_{\mathbf{W}}$ denotes a neural network with weights $\mathbf{W}$, for a multi-class classification problem with sample-label pair $(\mathbf{x}, y)$, we define the *margin* as

$$\gamma(\mathbf{x}, y, f_{\mathbf{W}}) = (f_{\mathbf{W}}(\mathbf{x}))[y] - \max_{i \neq y} f_{\mathbf{W}}(\mathbf{x})_i. \tag{16}$$

For machine translation, we consider the margin of each output token. We note that the number of classes, or the number of possible tokens, is often particularly large (in the order of thousands) for machine translation. Note that margins can be defined in any layer (Elsayed et al., 2018; Wei & Ma, 2019; Yang et al., 2020). Following Jiang et al. (2019), we consider output margins only, and use the 10^{th} percentile of the margin distribution calculated from the entire training set as a robust surrogate for the minimum margin. Using the margin γ defined as the 10^{th} percentile, we define several generalization metrics.

- (`inverse_margin`).

$$\mu_{\texttt{inverse\_margin}} = \frac{1}{\gamma^2}. \tag{17}$$

- (`log_prod_of_spec_over_margin`).

$$\mu_{\texttt{log\_prod\_of\_spec\_over\_margin}} = \log \frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2}{\gamma^2} = \mu_{\texttt{log\_prod\_of\_spec}} - 2\log\gamma. \tag{18}$$

  Note that `log_prod_of_spec` is not used in this paper due to overlap with `log_spectral_norm`.
- (`log_sum_of_spec_over_margin`).

$$\mu_{\texttt{log\_sum\_of\_spec\_over\_margin}} = \log d \left(\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_2^2}{\gamma^2}\right)^{1/d} = \log d + \frac{1}{d}\left(\mu_{\texttt{log\_prod\_of\_spec}} - 2\log\gamma\right). \tag{19}$$

- (`log_prod_of_fro_over_margin`).

$$\mu_{\texttt{log\_prod\_of\_fro\_over\_margin}} = \log \frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_F^2}{\gamma^2} = \mu_{\texttt{log\_prod\_of\_fro}} - 2\log\gamma. \tag{20}$$

  Note that `log_prod_of_fro` is not used in this paper due to overlap with `log_norm`.
- (`log_sum_of_fro_over_margin`).

$$\mu_{\texttt{log\_sum\_of\_fro\_over\_margin}} = \log d \left(\frac{\prod_{i=1}^{d} \|\mathbf{W}_i\|_F^2}{\gamma^2}\right)^{1/d} = \log d + \frac{1}{d}\left(\mu_{\texttt{log\_prod\_of\_fro}} - 2\log\gamma\right). \tag{21}$$

- (`path_norm_over_margin`).

$$\mu_{\texttt{path\_norm\_over\_margin}} = \frac{\mu_{\texttt{path\_norm}}}{\gamma^2}. \tag{22}$$

A.6 METRICS DERIVED FROM PAC-BAYESIAN BOUNDS

Several well-known generalization bounds are derived using the PAC-Bayesian framework, which bounds the generalization gap using the KL-divergence between a predefined prior distribution (usually chosen as Gaussian) and the posterior distribution of the trained models. A key component of the PAC-Bayesian bounds used in most existing implementations is the procedure of searching for

the largest magnitude of Gaussian perturbation, denoted as σ, such that the perturbed weights of the neural network achieve a bounded increase in the training loss. More specifically, σ is defined such that

$$\mathbb{E}_{\mathbf{U}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}[\text{TrainLoss}(f_{\mathbf{W}+\mathbf{U}})] \leq \text{TrainLoss}(f_{\mathbf{W}}) + \delta, \tag{23}$$

where δ is a predetermined threshold, and is chosen as $\delta = \frac{1}{2}$ in our machine translation experiments. Similarly, one can define a "magnitude-aware" perturbation σ' satisfying

$$\mathbb{E}_{\mathbf{U}}[\text{TrainLoss}(f_{\mathbf{W}+\mathbf{U}})] \leq \text{TrainLoss}(f_{\mathbf{W}}) + \delta, \tag{24}$$

where each weight entry u_i in $\mathbf{U}$ is distributed as $\mathcal{N}(0, \sigma'^2|w_i|^2 + \epsilon^2)$, and ϵ is chosen as 1e-3 (Dziugaite et al., 2020). Given the perturbation magnitude σ, the magnitude-aware perturbation σ' and the number of samples m, one can define the following generalization metrics.

- (`pacbayes_init`).

$$\mu_{\texttt{pacbayes\_init}} = \frac{\mu^2_{\texttt{12\_dist}}}{4\sigma^2} + \log\frac{m}{\sigma} + 10. \tag{25}$$

- (`pacbayes_orig`).

$$\mu_{\texttt{pacbayes\_orig}} = \frac{\mu^2_{12}}{4\sigma^2} + \log\frac{m}{\sigma} + 10. \tag{26}$$

- (`pacbayes_flatness`).

$$\mu_{\texttt{pacbayes\_flatness}} = \frac{1}{\sigma^2}. \tag{27}$$

- (`pacbayes_mag_init`).

$$\mu_{\texttt{pacbayes\_mag\_init}} = \frac{1}{4}\sum_{i=1}^{\omega}\log\left(\frac{\epsilon^2 + (\sigma'^2+1)\mu^2_{\texttt{12\_dist}}/\omega}{\epsilon^2 + \sigma'^2|w_i - w_i^{\text{init}}|^2}\right) + \log\frac{m}{\sigma} + 10. \tag{28}$$

- (`pacbayes_mag_orig`).

$$\mu_{\texttt{pacbayes\_mag\_orig}} = \frac{1}{4}\sum_{i=1}^{\omega}\log\left(\frac{\epsilon^2 + (\sigma'^2+1)\mu^2_{12}/\omega}{\epsilon^2 + \sigma'^2|w_i - w_i^{\text{init}}|^2}\right) + \log\frac{m}{\sigma} + 10. \tag{29}$$

- (`pacbayes_mag_flatness`).

$$\mu_{\texttt{pacbayes\_mag\_flatness}} = \frac{1}{\sigma'^2}. \tag{30}$$

B TASK THREE: RICH CORRELATIONAL STRUCTURES AND THE SIMPSON'S PARADOX WHEN DATA SIZE, MODEL SIZE AND TRAINING HYPERPARAMETERS ARE VARIED

For our final task, we vary each of the hyperparameters and study the trends of the generalization metrics. We first focus on the sample×learning rate×depth hyperparameter grid— see Figures 6 and 7 for plots of BLEU score against shape and scale metrics, respectively. Since we vary the number of samples, the learning rate, and the model depth to obtain different models, we group these models to visualize trends over each hyperparameter. In each subfigure, we color-code the models by either learning rate or the number of samples. We discuss grouping models by depth later in Figure 9c. Note that Figure 6 partially overlaps with Figure 2, except for different fitting methods.

For each curve, we expect the generalization metrics to be negatively correlated with the models' quality measured using the BLEU score, i.e., the regression lines should have negative slopes. Comparing Figure 6 and 7, one can see that the *shape metrics tend to show the correct trends (which are more negatively correlated) than the scale metrics*.

Remark. In Figure 6, constrained by the least-squares fitting, some regression lines are not aligned well with data, e.g., the second figure on the second row. In Appendix F, we fit the data using the orthogonal distance regression to mitigate this issue.

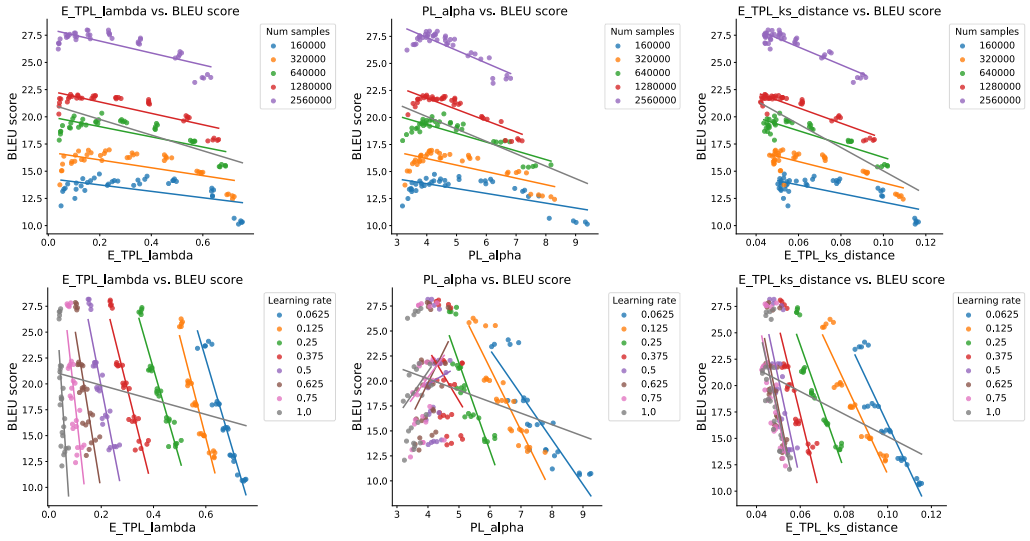


Figure 6: BLEU-score versus shape metrics for 200 Transformers trained on WMT14 with different hyperparameters. **(First row).** Trained models grouped by the learning rate. **(Second row).** Trained models grouped by the number of samples. The BLEU scores and the evaluated shape metrics display the correct (downward) trend.

Abnormal hyperparameters lead to the "Simpson's paradox". From Figure 6, we can see that the prediction of trends degrades for relatively large learning rates. For a fixed number of samples, when the learning rate becomes larger, the trends deviate from a perfectly linear fit. Now, we include more models trained with particularly large learning rates, and we show the results in Figure 8. We see that the results potentially display a "Simpson's paradox" (similar results for a different corpus of models have been reported previously by Martin & Mahoney (2021a)), i.e., the overall correlation trends are opposite to the trends in individual groups. In these figures, the regression lines are strongly influenced by the models trained with large learning rates and are biased towards the models with low BLEU scores. This phenomenon is known in the HT-SR literature (Martin & Mahoney, 2021a; Martin et al., 2021), and one often has to avoid biasing the results with these poorly trained models [3].

BLEU scores are not significantly influenced by network depth. Unlike learning rates and number of samples, we find that the BLEU scores are almost identical when we vary the number of layers from 4 to 8. In Figure 9a and 9b, we show `E_TPL_lambda` vs BLEU for models grouped by different learning rates and number of samples. These two subfigures are repeated from the first column of Figure 8. From these two figures, we see that the BLEU scores vary significantly when these two hyperparameters are varied. In Figure 9c, we show the same set of models color-coded by the network depth. We can see that the BLEU scores almost remain identical. This is because, from Figure 9a and 9b, we see that these models are roughly divided into "vertical" groups when the learning rate is varied, and they are roughly divided into "horizontal" groups when the number of samples is varied. Therefore, each small "cluster" in Figure 9c corresponds to a group of models trained with the same learning rate and the number of samples but different depths, and these clusters show that the BLEU scores almost remain fixed when the depth is varied. This phenomenon suggests that the rank correlations calculated for varying depths may not be informative.

Rank correlations. To systematically evaluate the various metrics considered in this paper, we study the rank correlation between these metrics and the BLEU score. For Task three, we consider each one-dimensional slice of the hyperparameter space $\mathbf{\Theta} = \{(\theta_1, \ldots, \theta_K) : \theta_1 \in \mathbf{\Theta}_1, \ldots, \theta_K \in \mathbf{\Theta}_K\}$, i.e., slices of the form

$$\{(\theta_1, \ldots, \theta_K) : \theta_i \in \Theta_i \text{ while other parameters } \theta_j, j \neq i \text{ are fixed}\},$$

and we calculate the rank correlation using the models in each such slice. Then, we aggregate the rank correlations from all the one-dimensional slices and plot the distributions of the rank correlations.

[3] From a theoretical point of view, this phenomenon is caused by the change of the HT random matrix universality class at the point of `PL_alpha` = 2. For more details, see Table 1 of Martin & Mahoney (2021b).

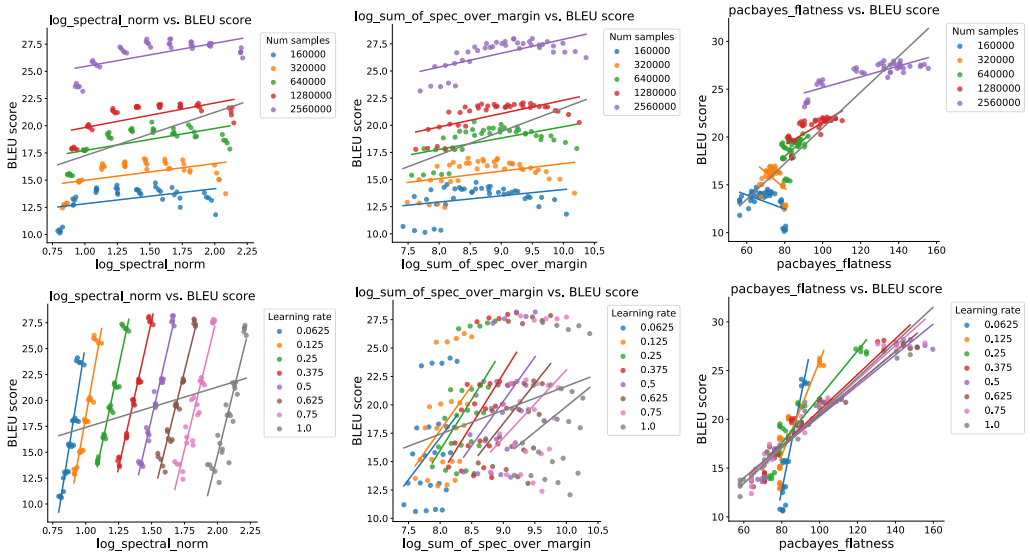


Figure 7: BLEU-score versus shape metrics for 200 Transformers trained on WMT14 with different hyperparameters. **(First row).** Trained models grouped by the learning rate. **(Second row).** Trained models grouped by the number of samples. The BLEU scores and the evaluated shape metrics display the wrong (upward) trend.

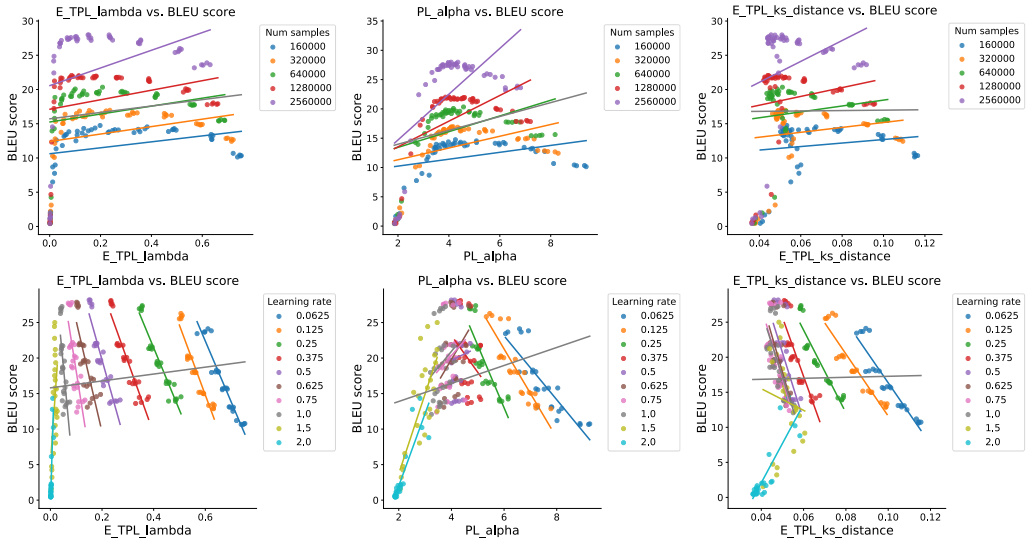
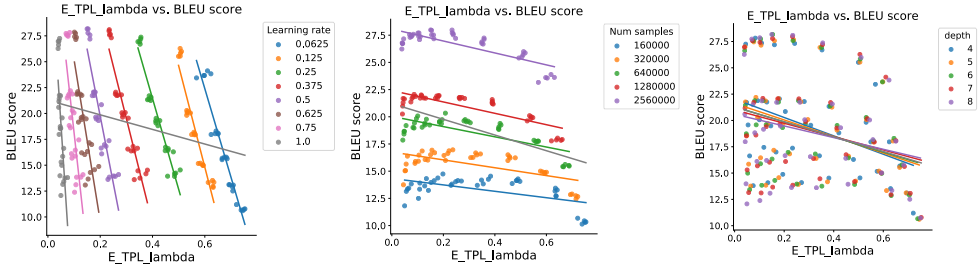


Figure 8: BLEU-score versus shape metrics with particularly large learning rates. **(First row).** Trained models grouped by the learning rate. **(Second row).** Trained models grouped by the number of samples. The BLEU scores and the evaluated shape metrics display Simpson's paradox when there are models trained with particularly large learning rates (1.5 and 2.0).



(a) Grouping by learning rates. (b) Grouping by number of samples. (c) Grouping by network depths.

Figure 9: `E_TPL_lambda` vs BLEU for the same set of trained models grouped by different hyperparameters. While BLEU scores change significantly when the initial learning rates and the number of samples are varied, they almost remain fixed for varying network depths.

See Figure 10. As we have shown in Figure 9c, the rank correlations calculated with varying network depths may not be informative due to the insignificant change in the BLEU score. Therefore, we focus on the other three hyperparameters, namely learning rate, network width and number of samples. Also, similar to Task two, we provide the rank correlation results on both the test BLEU scores and the generalization gap. Results on the generalization gap are shown in Figure 11.

Before we analyze the results of Figure 10 and 11, we discuss a subtle issue in calculating the generalization metrics. We note that, in Jiang et al. (2019), generalization metrics are "normalized" by dividing by the (square root of the) number of training samples, in correspondence with how they appear in uniform generalization bounds. However, normalizing the generalization metrics from Jiang et al. (2019) by the number of samples unavoidably complicates the correlation when varying the number of samples. This makes the comparison between scale metrics from Jiang et al. (2019) and the shape metrics challenging, as there is no natural way to normalize the shape metrics with respect to the number of training samples. Therefore, in Figure 10, we include the results for both with and without dividing the generalization metrics from Jiang et al. (2019) by the (square root of the) number of samples.

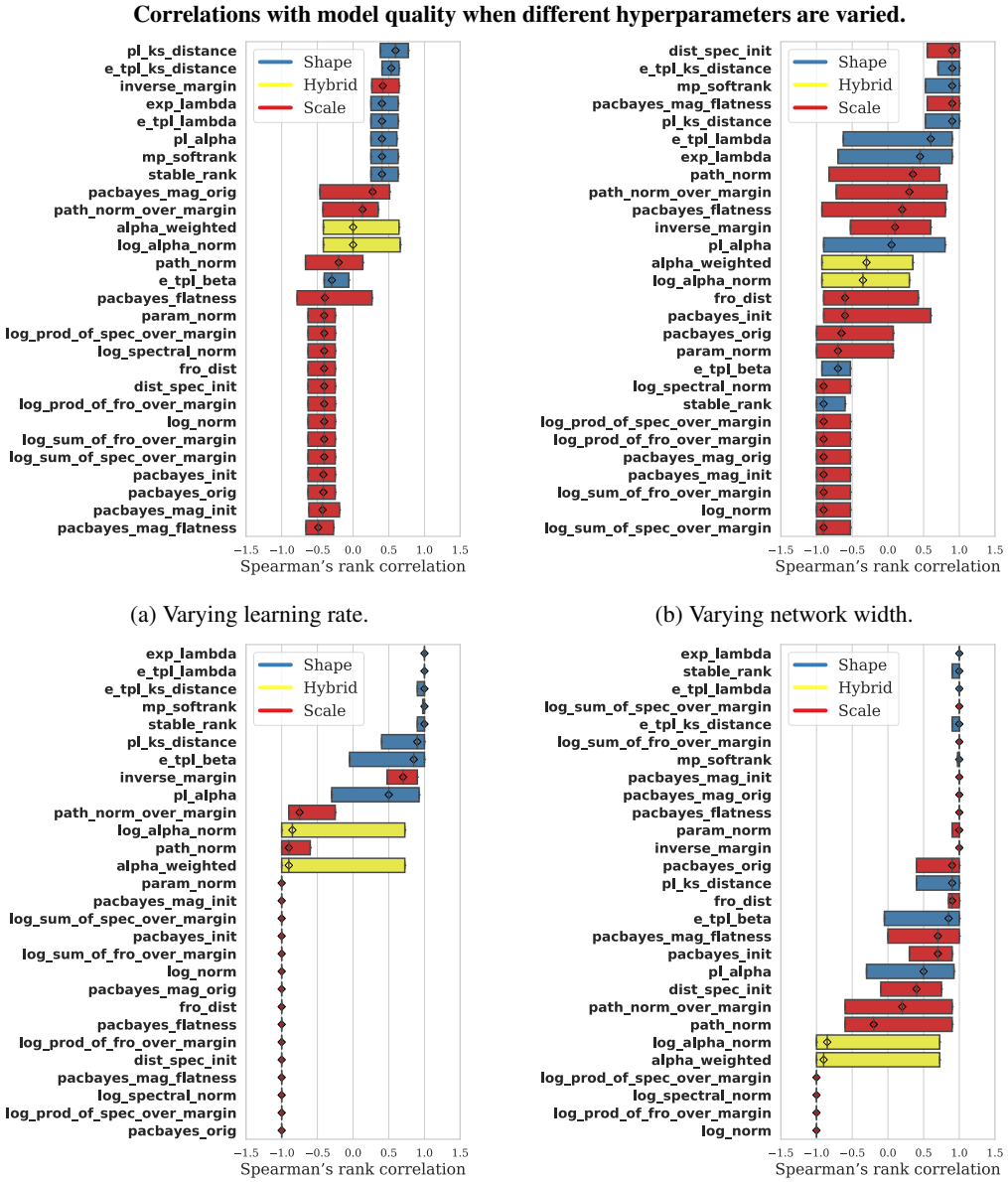
Here are our observations from Figure 10 and 11.

- From Figure 10a and 10c, shape metrics perform particularly well when varying the learning rate and the number of samples. Specifically, in Figure 10c, several shape metrics achieve perfect rank correlations, which are close to 1. From Figure 10b, shape metrics also perform well for varying network widths except for `stable_rank` and `E_TPL_beta`[4].

- From Figure 10d, normalizing the scale metrics from Jiang et al. (2019) by the number of samples significantly improves their correlational predictions. However, shape metrics can achieve a similar performance without the help of normalizing the number of samples.

- By comparing Figure 10 and 11, one can see that the scale metrics are much better correlated with the generalization gap than the test BLEU scores. For Figure 11a and 11b, this conclusion is obvious from the plots. For Figure 11c and 11d, the scale metrics need to be divided by (the square root of) the number of samples to achieve a good correlation when the number of samples is varied.

[4]The insufficiency of `stable_rank` is caused by the influence of the matrix size when the model width is varied, i.e., wider models tend to have a larger `stable_rank` simply because of the increased matrix size instead of the model quality. The insufficiency of `E_TPL_beta` is likely due to the *fix-finger method* in E-TPL fittings, which fixes $x_{\min}$ as the peak of the ESD without searching for the optimal value. However, optimal E-TPL fitting requires simultaneous searching of $x_{\min}$, `E_TPL_beta` and `E_TPL_lambda`, which is computationally demanding. Thus, further investigation is necessary for achieving a balance between the quality of fitting and the computational cost.



(a) Varying learning rate.

(b) Varying network width.

(c) Varying number of training samples. In this subplot, metrics from Jiang et al. (2019) are *not* normalized by the number of training samples.

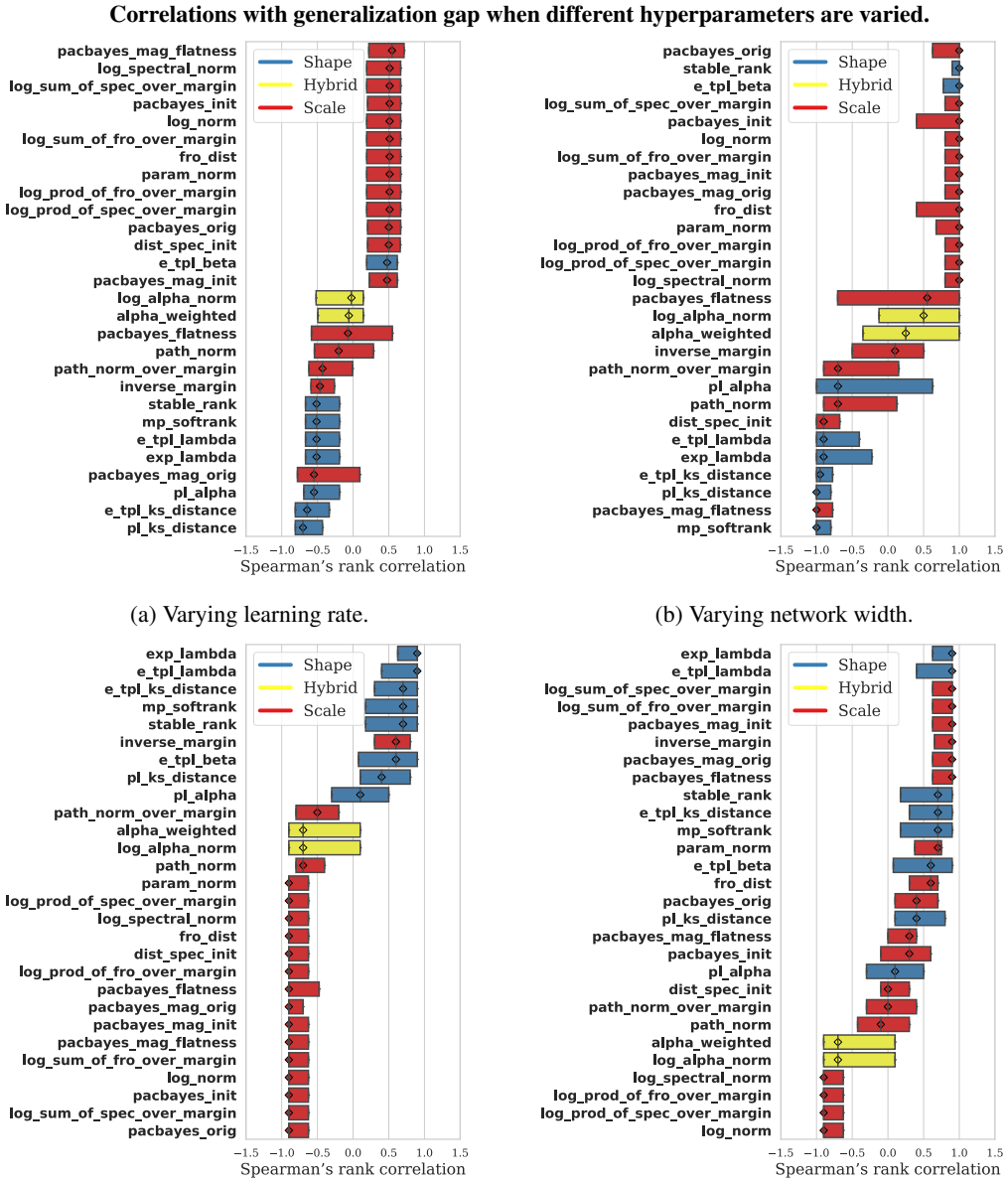
(d) Varying number of training samples. In this subplot, metrics from Jiang et al. (2019) are normalized by the number of training samples.

Figure 10: Comparing multiple generalization metrics in terms of the rank correlations with the test BLEU score when multiple hyperparameters are varied. Metrics are ranked by the median rank correlations.

C Additional details on the experiment setup

For training Transformers, we follow exactly the setup in Vaswani et al. (2017), and we develop our implementation based on an online repository[5] which reproduces the results from Vaswani et al. (2017) with more easily configurable Transformer architectures. The architecture puts the LayerNorm before the residual connection, which has been shown to provide more stabilized training (Liu et al., 2020; Xiong et al., 2020). As we have mentioned earlier, we vary the hyperparameters of training

[5] https://github.com/gordicaleksa/pytorch-original-transformer



(a) Varying learning rate.

(b) Varying network width.

(c) Varying number of training samples. In this subplot, metrics from Jiang et al. (2019) are *not* normalized by the number of training samples.

(d) Varying number of training samples. In this subplot, metrics from Jiang et al. (2019) are normalized by the number of training samples.

Figure 11: Comparing multiple generalization metrics in terms of the rank correlations with the *generalization gap* when multiple hyperparameters are varied. Metrics are ranked by the median rank correlations.

to evaluate the correlations between the generalization metrics and model quality. In the "standard setting", we train with Transformer-base, which has six layers, eight attention heads and embedding dimension 512. Then, we vary the number of Transformer layers from 4 to 8, and we vary the embedding dimension from 256 to 1024. When varying the embedding dimension, we let the number of attention heads vary proportionally.

We train with dropout 0.1 and 10% label smoothing. Note that in one experiment shown in Figure 3, we remove dropout to observe the effect of overfitting. For all of our experiments, we train with the inverse square-root learning rate. Given the embedding dimension d_e, step number t, number

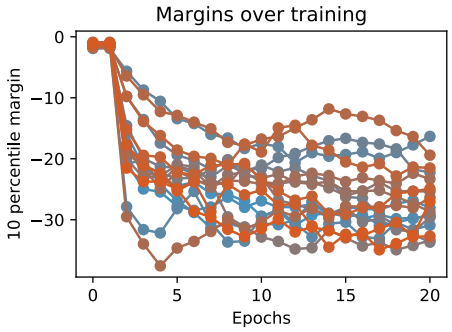


Figure 12: The margins remain negative in the experiments on machine translation due to the large alphabet size.

of warm-up steps t_w, the formula for the inverse square-root learning rate schedule (Vaswani et al., 2017) is the following.

$$\text{Learning Rate} = d_e^{-0.5} \cdot \min(t^{-0.5}, t \cdot t_w^{-1.5}). \tag{31}$$

For results trained with a particular learning rate lr, such as the results shown in Figure 9a, lr is the constant factor multiplied with the standard learning rate schedule (31). For each experiment, we train the model for 20 epochs. When calculating the ESDs of the weight matrices, we treat the query, key and value matrices as separate weight matrices.

D Additional analysis on scale metrics

In this section, we discuss an issue of computing margin-based generalization metrics. Generically, these bounds are of the form

$$L(f) \leq \hat{L}_\gamma(f) + C/\gamma$$

where $L(f)$ is the population error, $\hat{L}_\gamma$ is the training margin loss at margin γ, typically

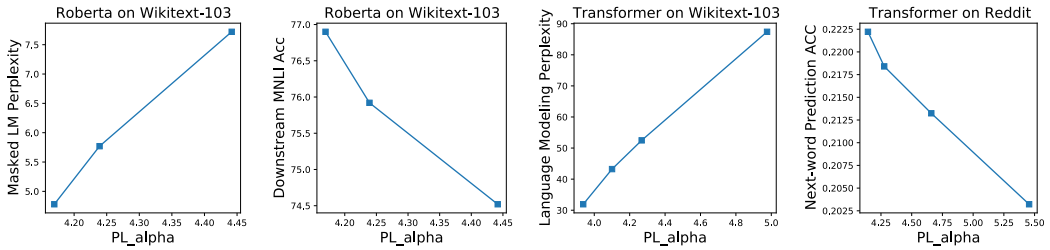
$$\sum_{(x,y)\in S} \mathbf{1}\{\max_j f(x)_j \leq \gamma + f(x)_y\},$$

and C is some complexity term. First, note that this construction requires the margin γ to be positive. Moreover, the training margin loss is an increasing function of γ, while the complexity term C/γ is decreasing in γ, thus the conventional way of using the margin bound is to optimize over the margin to balance two terms in the margin bound (Bartlett et al., 2017), rather than pre-specifying the value of the margin dependent on the data. However, we choose to follow the related papers Dziugaite et al. (2020); Jiang et al. (2019), and we use the 10th percentile margin as a robust estimate of the minimum margin in the dataset. We use this margin in all of the margin-normalized generalization metrics. However, in all of the experiments on machine translation, the 10th percentile margin remains negative throughout the whole training, violating the requirement that the bound is evaluated at a positive value of margin. See Figure 12. This problem results from the large Alphabet for machine translation, which makes it difficult to fully interpolate the data, and hence makes the margin-normalized generalization metrics in Dziugaite et al. (2020); Jiang et al. (2019) hard to be applicable to the present setting.

E Corroborating results

In this subsection, we consider corroborating results, extending the setup of the main paper to more datasets and different evaluation methods.

E.1 Additional results on natural language processing tasks



(a) Lower perplexity is better. (b) Higher accuracy is better. (c) Lower perplexity is better. (d) Higher accuracy is better.

Figure 13: Multiple language processing tasks evaluated using the `PL_alpha` metric on models trained with different data sizes. The `PL_alpha` metric correctly predicts the trend in all tasks.

We consider three other NLP tasks:

- Roberta (Liu et al., 2019) trained on the masked language modeling task using the Wikitext-103 dataset, and then finetuned on the MNLI dataset (Williams et al., 2018).
- Six-layer base Transformers trained on the language modeling task using the Wikitext-103 dataset (Merity et al., 2016);
- Six-layer base Transformers trained on the next-word prediction task using the Reddit dataset (Baumgartner), following the implementation in Bagdasaryan et al. (2020);

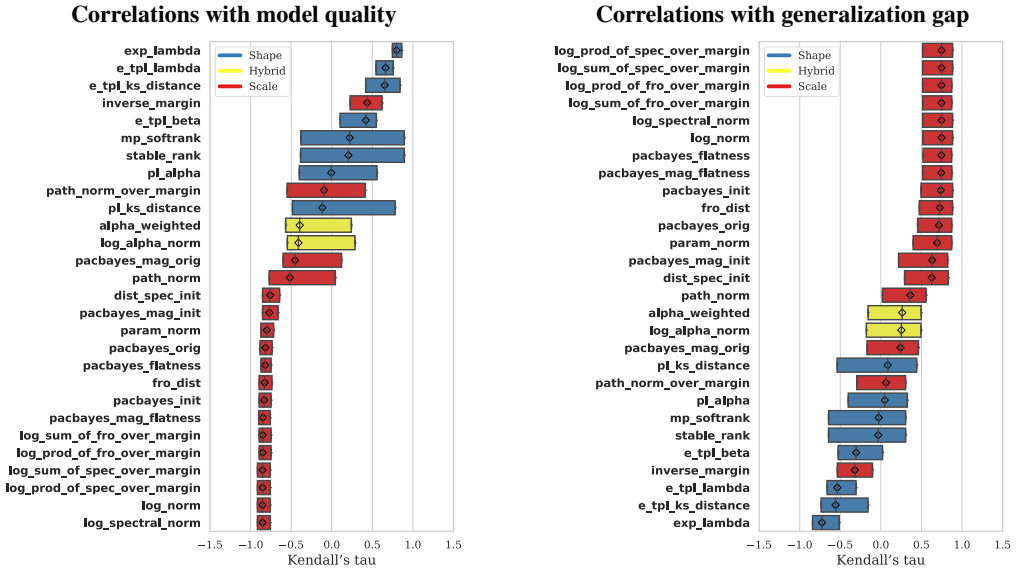
For each task, we train models on different data sizes. Then, we measure the `PL_alpha` metric and report the correlation with the ground-truth quality metric. See Figure 13. In these experiments, the `PL_alpha` metric predicts the correct trend, i.e., a lower value of `PL_alpha` corresponds to a better model.

E.2 Evaluating rank correlations using Kendall's tau metric

Next, we reimplement Task two using Kendall's tau to calculate the rank correlations. The results in Figure 14 are very similar to those in Figure 4.

F Fitting regression lines using orthogonal distance regression

In this section, we fit the regression plots from Figure 6 to 7 using the orthogonal distance regression (Boggs & Rogers, 1990). See Figure 15 to Figure 16.



(a) **Correlations with model quality.** Kendall's tau between various generalization metrics and BLEU.

(b) **Correlations with generalization gap.** Kendall's tau between various generalization metrics and the generalization gap.

Figure 14: Evaluating Task Two using Kendall's tau. Results are similar to those in Figure 4.

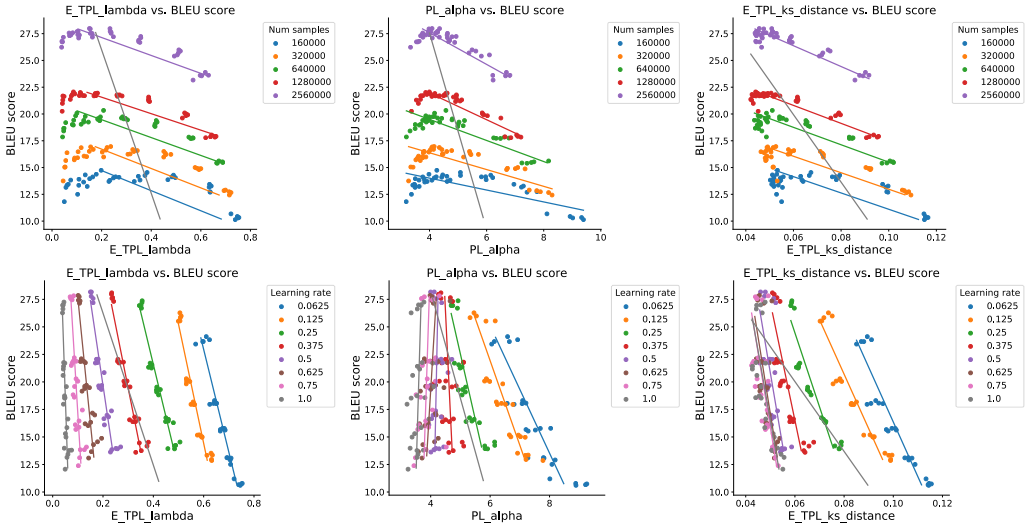


Figure 15: BLEU-score versus shape metrics. **(First row).** Trained models grouped by the learning rate. **(Second row).** Trained models grouped by the number of samples. The regression lines are fitted using orthogonal distance regression.

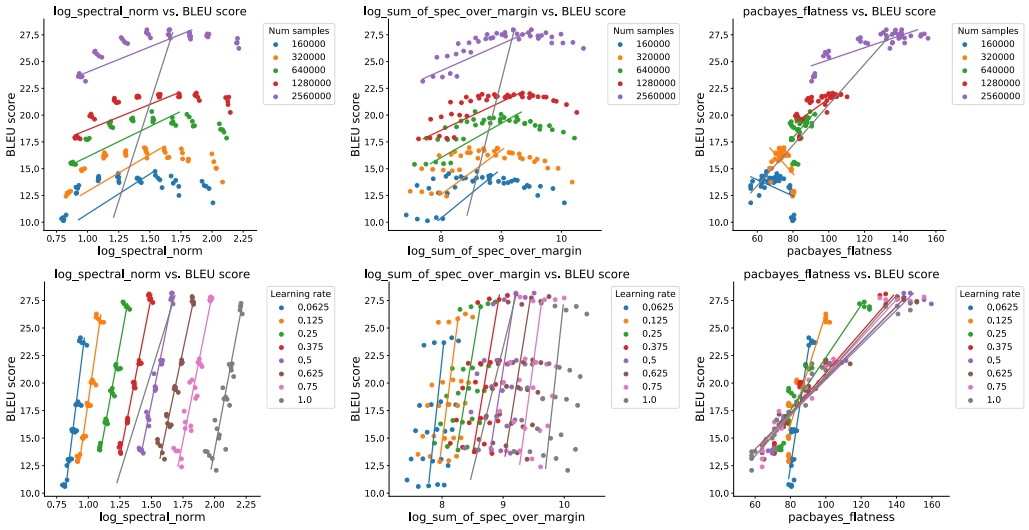


Figure 16: BLEU-score versus scale metrics. **(First row).** Trained models grouped by the learning rate. **(Second row).** Trained models grouped by the number of samples. The regression lines are fitted using orthogonal distance regression.