A Additional Derivations and Discussions Regarding the PACTran Metrics

A.1 PACTran-Dirichlet

Variational Inference Derivations In variational inference, we make use of a set of independent distributions, including multinomial distributions $q(z_i)$ and Dirichlet distributions $q(\mathbf{w}_z; \tilde{\boldsymbol{\alpha}}_z)$ and apply Jensen's inequality [5] to Eq.(9) such that,

$$\begin{aligned}\log Z(S) &\geq H_q(\mathbf{z}) + H_q(\mathbf{W}) + \sum_{z_1} \ldots \sum_{z_N} q(\mathbf{z}) \int d\,\mathbf{W}\, q(\mathbf{W}) \log p(\mathbf{y}, \mathbf{z}, \mathbf{W} \,|\, \mathbf{x}) \\ &= H_q(\mathbf{z}) + H_q(\mathbf{W}) + \sum_{z_1} \ldots \sum_{z_N} q(\mathbf{z}) \int d\,\mathbf{W}\, q(\mathbf{W}) \\ &\left(\sum_z \log \frac{\Gamma(\sum_y \alpha_y)}{\prod_y \Gamma(\alpha_y)} + \sum_z \sum_y (\alpha_y - 1) \log w_{yz} + \sum_i \sum_z \delta_{z_i = z} \log(M(\mathbf{x}_i)_z w_{y_i z}) \right).\end{aligned} \tag{15}$$

The variational inference seeks the optimal approximate distributions $q(z_i)$ and $q(\mathbf{w}_z; \tilde{\boldsymbol{\alpha}}_z)$ that maximize Eq. (15). Taking the functional derivative w.r.t. $q(\mathbf{z})$ and making it equal to 0, one gets

$$\begin{aligned}\log q^*(z_i = z) = &\int q(\mathbf{W}) \log p(\mathbf{y}, z_i = z, \mathbf{W} \,|\, \mathbf{x}) d\,\mathbf{W} + C \\ &= \log M(\mathbf{x}_i)_z + \mathbb{E}_{q(\mathbf{W})} \log w_{y_i z} + C,\end{aligned}$$

where C is a constant. Since $q(\mathbf{W})$ are Dirichlet distributions, we have

$$\mathbb{E}_{q(\mathbf{W})} \log w_{y_i z} = \Psi(\tilde{\alpha}_{y_i z}) - \Psi(\sum_y \tilde{\alpha}_{yz}).$$

Next, taking the functional derivative w.r.t. $q(\mathbf{W})$ and making it equal to 0, one gets

$$\begin{aligned}\log q^*(w_{yz}) &= \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{y}, \mathbf{z}, w_z^y | \, \mathbf{x}) + C \\ &= \left(\alpha_y - 1 + \sum_i q(z_i = z) \delta(y_i = y) \right) \log w_{yz} + C,\end{aligned}$$

where C is a constant. Since $\log q^*(w_{yz}) = (\tilde{\alpha}_{yz} - 1) \log w_{yz} + C$, we have

$$\tilde{\alpha}_{yz} = \alpha_y + \sum_i q(z_i = z) \delta(y_i = y).$$

A.2 PACTran-Gamma

Marginal Evidence Since the denominator of Eq.(11) creates difficulties for Bayesian inference, we introduce a set of augmented variables R_i from the exponential distribution as in [3] to ”cancel out” the denominator, such that

$$p(y_i, z, R_i|\, \mathbf{x}_i, \mathbf{V}) = M(\mathbf{x}_i)_z v_{y_i z} \exp\left(-R_i\left(\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} M(\mathbf{x}_i)_z v_{yz}\right)\right), \tag{16}$$

It is easy to verify that $\int_0^{+\infty} p(y_i, z, R_i|\, \mathbf{x}_i, \mathbf{V}) dR_i = p(y_i, z|\, \mathbf{x}_i, \mathbf{V})$. Therefore, the marginal evidence $\log Z(S)$ becomes,

$$\begin{aligned}
&\log \int \prod_y \prod_z P(v_{yz}) \prod_i \left(\sum_z p(y_i, z|\, \mathbf{x}_i, \mathbf{V})\right) d\,\mathbf{V} \\
=&\log \int \prod_y \prod_z P(v_{yz}) \prod_i \left(\sum_z p(y_i, z, R_i|\, \mathbf{x}_i, \mathbf{V})\right) dR_i d\,\mathbf{V} \\
=&\log \int \prod_y \prod_z \frac{b^{a_y}}{\Gamma(a_y)} v_{yz}^{a_y-1} \exp(-b v_{yz}) \prod_i \\
&\left(\sum_z M(\mathbf{x}_i)_z v_{y_i z} \exp\left(-R_i\left(\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} M(\mathbf{x}_i)_z v_{yz}\right)\right)\right) dR_i d\,\mathbf{V} \\
=&\log \sum_{z_1} \dots \sum_{z_N} \int \prod_y \prod_z \frac{b^{a_y}}{\Gamma(a_y)} v_{yz}^{a_y-1} \exp(-b v_{yz}) \prod_i \\
&\left(M(\mathbf{x}_i)_{z_i} v_{y_i z_i} \exp\left(-R_i\left(\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} M(\mathbf{x}_i)_z v_{yz}\right)\right)\right) dR_i d\,\mathbf{V}\,.
\end{aligned}$$

Since the exact inference is infeasible, we again apply variational inference.

Variational Inference Derivations Similar to the PACTran-Dirichlet, we make use of a set of independent distributions $q(z_i)$, $q(v_{yz}; \tilde{a}_{yz}, b)$ and $q(R_i; \tilde{\lambda}_i)$ and write

$$\begin{aligned}
&\log Z(S) \\
\geq& H_q(\mathbf{z}) + H_q(\mathbf{V}) + H_q(\mathbf{R}) + \sum_{z_1} \dots \sum_{z_N} q(\mathbf{z}) \int d\,\mathbf{V}\, d\,\mathbf{R}\, q(\mathbf{V}) q(\mathbf{R}) \\
&+ \sum_y \sum_z a_y \log b - \sum_y \sum_z \log \Gamma(a_y) + \sum_z \sum_y ((a_y - 1) \log v_{yz} - b v_{yz}) + \\
&\sum_i \sum_z \delta_{z_i = z} \log(M(\mathbf{x}_i)_z v_{y_i z}) - \sum_i R_i \left(\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} M(\mathbf{x}_i)_z v_{yz}\right).
\end{aligned} \tag{17}$$

The PACTran-Gamma metric is the resulting negative ELBO after applying variational principles, and takes the following form (when $b = 1$):

$$\sum_y \sum_z \left(\log \Gamma(a_y) - \log \Gamma(\tilde{a}_{yz})\right) + \sum_i \log \tilde{\lambda}_i$$
$$+ \sum_i \sum_z q^*(z_i = z) \left(\log q^*(z_i = z) - \log M(x_i)_z\right), \tag{18}$$

where,

$$q^*(z_i = z) = \mathrm{softmax}\left(\log M(\mathbf{x}_i)_z + \Psi(\tilde{a}_{y_i z})\right),$$
$$\tilde{a}_{yz} = a_y + \sum_i q^*(z_i = z)\delta(y_i = y), \;\; \tilde{\lambda}_i = \sum_y \sum_z M(\mathbf{x}_i)_z \tilde{a}_{yz}.$$

The above equations are obtained in a similar way as the ones of the PACTran-Dirichlet metric in A.1.

A.3 PACTran-Gaussian

The optimal Gaussian Posterior To obtain the optimal parameters σ_*^2 and $\boldsymbol{\theta}_*$ of the Gaussian posterior, first take the derivative of Eq.(13) w.r.t. σ_q^2 and make it zero,

$$\mathrm{Tr}(\nabla^2 \hat{L}(\boldsymbol{\theta}_q, S)) + \frac{KD}{\lambda}\left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_q^2}\right) = 0.$$

After rearrangement, one gets

$$\frac{\sigma_0^2}{\sigma_q^2} = 1 + \frac{\beta}{KD}\mathrm{Tr}(\nabla^2 \hat{L}(\boldsymbol{\theta}_q, S)),$$

where $\beta = \lambda\sigma_0^2$. Now plugging this into Eq.(13) yields

$$\hat{L}(\boldsymbol{\theta}_q, S) + \frac{\|\boldsymbol{\theta}_q\|_F^2}{2\beta} + \frac{KD\sigma_0^2}{\beta} \log \frac{\sigma_0^2}{\sigma_q^2}, \tag{19}$$

and the optimal $\boldsymbol{\theta}_*$ is the one which minimizes the above objective function. Strictly speaking, the objective function with respect to $\boldsymbol{\theta}_q$ should consider the last term of Eq.(19). However, this would make the objective non-convex and hard to optimize. Therefore, we approximate the solution by ignoring the last term and only optimize $\boldsymbol{\theta}_q$ over the first two terms, which is a strongly convex objective and can be solved efficiently with an off-the-shelf optimizer (e.g. L-BFGS).

2nd-Order Derivative of the Cross-Entropy Loss. For a given dataset S, assuming $\mathbf{X}$ is the feature matrix of size $N \times D$, where N is the number of examples and D is the feature dimension. $\mathbf{Y}$ is a binary matrix of size $N \times K$ representing the labels, where K is the number of classes. Then the logits can be represented as

$$\mathbf{G} = \mathbf{X}\,\mathbf{W} + \mathbf{b},$$

where $\mathbf{W}$ is $D \times K$ and $\mathbf{b}$ is a K-dim bias vector. For $\boldsymbol{\theta}_q = (\mathbf{W}, \mathbf{b})$, its cross-entropy loss on the dataset S is

$$\hat{L}(\boldsymbol{\theta}_q, S) = \frac{1}{N}\sum_i \left(-\sum_k y_{ik} g_{ik} + \log \sum_k \exp(g_{ik}) \right).$$

Its first derivative w.r.t. w_{jk} is

$$\frac{\partial \hat{L}}{\partial w_{jk}} = \frac{1}{N}\sum_i x_{ij} \frac{\partial \hat{L}}{\partial g_{ik}} = \frac{1}{N}\sum_i x_{ij} \left(\text{softmax}(g_{ik}) - y_{ik} \right),$$

and the second derivative w.r.t. w_{jk} is

$$\frac{\partial^2 \hat{L}}{\partial w_{jk}^2} = \frac{1}{N}\sum_i x_{ij}^2 \left(\text{softmax}(g_{ik}) - \text{softmax}(g_{ik})^2 \right). \tag{20}$$

Similarly, its first derivative w.r.t. b_k is

$$\frac{\partial \hat{L}}{\partial b_k} = \frac{1}{N}\sum_i \left(\text{softmax}(g_{ik}) - y_{ik} \right),$$

and the second derivative is

$$\frac{\partial^2 \hat{L}}{\partial b_k^2} = \frac{1}{N}\sum_i \left(\text{softmax}(g_{ik}) - \text{softmax}(g_{ik})^2 \right). \tag{21}$$

Given Eq.(20) and Eq.(21), the trace of the Hessian can be written as

$$\text{Tr}(\nabla^2 \hat{L}(\boldsymbol{\theta}_q, S)) = \sum_{jk} \frac{\partial^2 \hat{L}}{\partial w_{jk}^2} + \sum_k \frac{\partial^2 \hat{L}}{\partial b_k^2},$$

which is used as the ”flatness regularizer” (in the 3rd term of Eq.(19)).

Differences between LogME and PACTran-Gaussian Although both LogME [59] and PACTran-Gaussian apply the Gaussian priors on the top-layer parameters $\boldsymbol{\theta}$, they differ in the following two aspects: (1) LogME models the data distribution using the Gaussian likelihood, which corresponds to the squared loss in its logarithm form for optimization. On the other hand, PACTran applies the cross-entropy loss, which is more natural for classification tasks and is universally applied in practical downstream finetunings. (2) LogME optimizes all adjustable hyper-parameters along with the parameters $\boldsymbol{\theta}$ which results with a highly complex optimization problem. In contrast, PACTran-Gaussian only optimizes the parameters $\boldsymbol{\theta}$ which is convex, while heuristically setting the hyper-parameters β and σ_0 separately (Section B.5).

A.4 Complexity of the PACTran Metrics

Overall, the complexity of the PACTran metrics is $O(NKDt)$, where t is either the number of variational inference updates, or the number of L-BFGS steps.

PACTran-Dirichlet The PACTran-Dirichlet metric in Eq. (10) involves two sums, the sum over the D source classes of z and the sum of N examples. Each $C(\alpha)$ involves K classes of y. So the overall complexity is $O(ND + KD)$. To compute q^* and $\tilde{\alpha}$, it involves $t \leq 10$ iterations of variational updates. There are ND of q^* terms and the overall complexity is $O(ND + KD)$. There are KD terms of $\tilde{\alpha}$ and the overall complexity is $O(NKD)$. Therefore, the overall complexity of PACTran-Dirichlet is $O(NKDt)$.

PACTran-Gamma The PACTran-Gamma metric has similar complexity to the PACTran-Dirichlet. The complexity of Eq. (12) is $O(ND + KD)$. There are ND of q^* terms and the overall complexity is $O(ND)$. There are KD terms of $\tilde{\alpha}$ and the overall complexity is $O(NKD)$. There are N terms of $\tilde{\lambda}$ and the overall complexity is $O(NKD)$. Therefore, the overall complexity of PACTran-Gamma is also $O(NKDt)$ where t is the number of variational updates.

PACTran-Gaussian The PACTran-Gaussian metric according to Eq. (14) involves three terms. Evaluating the first two terms has complexity $O(NKD)$. The third term involves the 2nd-order derivative of the loss which has a closed form solution as shown in Section A.3 and evaluation complexity $O(NKD)$[⋆⋆⋆]. To obtain $\boldsymbol{\theta}_*$, we call L-BFGS which requires computing the derivative, which is also of complexity $O(NKD)$. Therefore, the overall complexity of PACTran-Gaussian is $O(NKDt)$, where t is the number of L-BFGS function/derivative evaluations.

B Additional Details of the Neural Checkpoint Ranking Benchmark (NeuCRaB) Experiments

B.1 Pretraining Checkpoint Descriptions

All checkpoints were based on the ResNet50-v2 architecture and pretrained on Imagenet using various approaches.

[⋆⋆⋆] It is worth noting that our complexity is significantly lower than the one of the classical Laplacian approximation, which involves the determinant of a Hessian with $O(NK^3D^3)$ complexity.

1. Jigsaw: trained with the self-supervised jigsaw-puzzle loss [40].
2. Relative Patch Location: trained with the relative path location prediction self-supervised loss [15].
3. Exemplar: trained with the Exemplar loss [16].
4. Rotation: representation obtained by predicting image rotations [21].
5. Sup-Rotation: trained with a supervised loss and an auxiliary Rotation loss [21].
6. WAE-UKL: encoder obtained by training a Wasserstein Autoencoder using the RAM-MC method of [46] as a distribution matching penalty that upper bounds the KL divergence (UKL stands for Upper-bound KL).
7. WAE-GAN: encoder obtained by training a Wasserstein Auto-Encoder using GAN-based distribution matching loss [51].
8. WAE-MMD: encoder obtained by training a Wasserstein Auto-Encoder using the Maximum Mean Discrepancy (MMD) distribution matching loss [51].
9. Cond-BigGAN: representation obtained from the discriminator of a BigGAN trained for class-conditional image synthesis [9].
10. Uncond-BigGAN: representation obtained from the discriminator of an unconditional BigGAN model with auxiliary self-supervision.
11. VAE: Encoder obtained by training a Variational Auto-Encoder [30].
12. Semi-Rotation-10%: trained with a supervised loss on 10% of the ImageNet examples and with an auxiliary Rotation loss [21] on all of the examples.
13. Semi-Exemplar-10%: trained with a supervised loss on 10% of the ImageNet examples and with an auxiliary Exemplar loss [16] on all of the examples.
14. Sup-Exemplar-100%: trained with a supervised loss and an auxiliary Exemplar loss [16] on all of the examples.
15. Sup-100%: representation obtained by standard supervised training on ImageNet.
16. Feature Vector: representation obtained by a ResNet50 model using the identity mappings as in [26] with supervised loss.

B.2 Downstreaming Task Descriptions

In this section, we describe the VTAB downstream tasks used in Section 4.1.

1. **Caltech101** [18] contains 101 classes, including animals, airplanes, chairs, etc. The image size varies from 200 to 300 pixels per edge.
2. **Flowers102** [39] contains 102 classes, with 40 to 248 training images (at least 500 pixels) per class.
3. **Patch Camelyon** [56] contains 327,680 images of histopathologic scans of lymph node sections with image size of 96x96, which is collected to predict the presence of metastatic tissue.
4. **Sun397** [58] is a scenery benchmark with 397 classes, including cathedral, staircase, shelter, river, or archipelago.
5. **Cifar-10** [32] consists of 60,000 32x32 colour images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.

6. **Oxford-IIIT Pet** [41] is a 37-class pet image dataset with roughly 200 images for each class. The images have large variations in scale, pose and lighting. All images have an associated ground truth annotation of breed.
7. **Smallnorb** [33] is a dataset intended for experiments in 3D object recognition from shape. It contains images of 50 toys belonging to 5 generic categories: four-legged animals, human figures, airplanes, trucks, and cars. We used the azimuth angle as the label, which has 18 classes (0 to 340 every 20 degrees).
8. **DMLAB** [61] is a dataset for evaluating the ability of a visual model to reason about distances from the visual input in 3D environments. It has 100,000 360x480 color images in 6 classes. The classes are {close, far, very far} x {positive reward, negative reward} respectively.
9. **CBIS-DDSM** [48] stands for Curated Breast Imaging Subset of Digital Database for Screening Mammography. It contains 65,130 patches with both calcification and mass cases, plus patches with no abnormalities. Designed as a traditional 5-class classification task.

B.3 Finetuning

Each pretrained checkpoint was finetuned on each downstream task in the following two ways: 8 attempts were made by full-model finetuning of the checkpoints with batch size 512, weight-decay 0.0001, with an SGD-Momentum optimizer using a decaying learning schedule with different starting learning rate lr and stopping iterations $iter$: $lr \in \{0.1, 0.05, 0.01, 0.005\}$ and $iter \in \{10000, 5000\}$; 5 attempts were done with top-layer-only finetuning using an L-BFGS solver with weight decay $\frac{1}{B} \cdot \{0.01, 0.1, 1., 10., 100.\}$, where B is the size of the training set. The ground-truth testing error was set to the lowest test error among all runs.

B.4 Computation Platform

The pretrained feature extraction and the pretrained model finetuning were done on the Google Cloud V1 2x2 TPUs. The transferability metrics were computed on the Google Cloud Intel Skylake CPU (2GHz per core) with 1 core and 10GB RAM per run.

B.5 Hyperparameter Studies of the PACTran-Gaussian Metric

Recall that in Eq. (14) we decomposed the PACTran-Gaussian metric into two parts: the l_2-regularized empirical risk (RER) and the "flatness regularizer" (FR). There are two hyperparameters in the PACTran-Gaussian metrics: β and σ_0^2. The β hyperparameter is mainly responsible of adjusting the l_2 regularizer so that the magnitude of $\boldsymbol{\theta}_*$ would not get too large. The σ_0^2 hyperparameter is mainly responsible of balancing the weights between RER and FR.

$$\underbrace{\hat{L}(\boldsymbol{\theta}_*, S) + \frac{\|\boldsymbol{\theta}_*\|_F^2}{2\beta}}_{RER} + \underbrace{\frac{KD\sigma_0^2}{2\beta} \log \frac{\sigma_0^2}{\sigma_*^2}}_{FR} .$$

In the experiment, we performed a hyperparameter grid-search over $\beta \in a \cdot N$ and $\sigma_0^2 \in b \cdot \frac{1}{D}$, for various choices of a and b. In particular, $a \in \{0.1, 1, 10\}$, and $b \in \{1, 10, 100, 1000\}$. The hyperparameter (β, σ_0^2) that maximizes the Kendall correlation between the PT-Gauss metric and LINEAR-VALID was chosen for the PT-Gauss$_{grid}$ metric.

In Fig.3-11, we plotted the performance of different hyperparameters, labeled as (a, b), on the 9 VTAB tasks of the NeuCraB experiments. Each figure is composed of two columns and three rows (corresponding to $N/K \in \{2, 5, 10\}$). The left column plotted the ratio between the robust standard deviation of the FR and RER term (x-axis) vs. the Kendall-Tau between PT-Gauss and the downstream test error (y-axis). The right column plotted the Kendall-Tau between PT-Gauss and LINEAR-VALID (x-axis) vs. the Kendall-Tau between PT-Gauss and the downstream test error (y-axis).

From the left columns, we find that the ratios between the standard deviation of the FR term and the RER term are very indicative of the performances of the PT-Gauss metric. Intuitively, too low of a ratio (≤ 0.1) reduces PT-Gauss to the LINEAR metric, while too high of a ratio (≥ 10) completely ignores the RER term. Both scenarios are clearly not optimal according to the results, and the optimal ratio is consistently around 1.0 which achieves a balance between FR and RER. For better visualization, we group the hyperparameters pairs by the a values using different colors (Yellow: $a = 0.1$, Red: $a = 1$, Blue: $a = 10$).

From the right columns, we find that the Kendall-Tau between PT-Gauss and LINEAR-VALID is in general well correlated with the Kendall-Tau between PT-Gauss and the test error. This justifies our choice of using LINEAR-VALID as a validation method for choosing hyperparameters. The few exceptions (e.g. SmallNorb) are mostly caused by the poor performance of LINEAR-VALID on that dataset.

B.6 Results on Each Individual Dataset

In Table 4, 5, 6, we report the complete results on each individual VTAB task.

B.7 Results on Checkpoints with the Same Feature Dimension

The sixteen model checkpoints in the Neural Checkpoint Ranking Benchmark (NeuCRaB) have different penultimate feature dimensions. In particular, ten of them have 2048 dimensions: Jigsaw, Relative Patch Location, Exemplar, Rotation, Sup-Rotation, Semi-Rotation-10%, Semi-Exemplar-10%, Sup-Exemplar-100%, Sup-100%, and Feature Vector; two of them have 1536 dimensions: Cond-BigGAN and Uncond-BigGAN; and four of them have 128 dimensions: WAE-UKL, WAE-GAN, WAE-MMD, VAE.

In the 9 VTAB tasks that have been considered, the dimensionalities of the penultimate features appear to be positively correlated to the performance of the checkpoints (where checkpoints with higher feature dimensions tend to achieve higher testing accuracies). In fact, if the feature dimension is directly used as the metric (in which we need to add a small random perturbation to break the ties

100+ classes	Caltech101	Oxford-flowers	Sun397
LEEP	0.253 ± 0.023	0.140 ± 0.016	0.213 ± 0.010
$\mathcal{N}$-LEEP	0.747 ± 0.024	0.663 ± 0.026	0.760 ± 0.019
H-score	0.327 ± 0.056	0.443 ± 0.035	0.470 ± 0.041
LogME	0.350 ± 0.000	0.293 ± 0.008	0.280 ± 0.003
LINEAR	0.253 ± 0.010	0.203 ± 0.006	0.237 ± 0.006
LINEAR-VALID	0.778 ± 0.034	0.726 ± 0.025	0.746 ± 0.038
$\mathcal{N}$-PT-Dir	0.787 ± 0.030	0.713 ± 0.018	0.780 ± 0.029
$\mathcal{N}$-PT-Gam	0.790 ± 0.032	0.713 ± 0.013	0.780 ± 0.024
PT-Gauss$_{grid}$	0.860 ± 0.014	0.913 ± 0.015	0.830 ± 0.010
PT-Gauss$_{fix}$	0.800 ± 0.011	0.750 ± 0.020	0.760 ± 0.011
10-99 classes	Cifar-10	Oxford-IIIT Pet	SmallNorb
LEEP	-0.040 ± 0.035	0.206 ± 0.008	-0.150 ± 0.062
$\mathcal{N}$-LEEP	0.419 ± 0.062	0.678 ± 0.017	0.107 ± 0.027
H-score	0.005 ± 0.033	0.072 ± 0.057	0.242 ± 0.019
LogME	0.153 ± 0.003	0.206 ± 0.006	-0.157 ± 0.006
LINEAR	0.160 ± 0.025	0.203 ± 0.003	-0.147 ± 0.015
LINEAR-VALID	0.311 ± 0.079	0.672 ± 0.027	-0.055 ± 0.070
$\mathcal{N}$-PT-Dir	0.413 ± 0.101	0.678 ± 0.033	-0.110 ± 0.032
$\mathcal{N}$-PT-Gam	0.420 ± 0.105	0.678 ± 0.032	-0.100 ± 0.040
PT-Gauss$_{grid}$	0.770 ± 0.025	0.775 ± 0.012	0.447 ± 0.037
PT-Gauss$_{fix}$	0.770 ± 0.030	0.832 ± 0.012	0.447 ± 0.037
2-9 classes	Patch-Camelyon	DMLAB	CBIS-DDSM
LEEP	-0.024 ± 0.030	-0.003 ± 0.037	0.150 ± 0.022
$\mathcal{N}$-LEEP	0.162 ± 0.039	0.069 ± 0.088	-0.003 ± 0.085
H-score	0.393 ± 0.056	0.260 ± 0.071	-0.097 ± 0.056
LogME	-0.123 ± 0.013	0.073 ± 0.006	0.263 ± 0.006
LINEAR	-0.043 ± 0.025	0.097 ± 0.018	0.287 ± 0.025
LINEAR-VALID	0.294 ± 0.065	0.017 ± 0.097	-0.123 ± 0.070
$\mathcal{N}$-PT-Dir	0.164 ± 0.054	0.027 ± 0.053	0.107 ± 0.062
$\mathcal{N}$-PT-Gam	0.177 ± 0.050	0.027 ± 0.052	0.120 ± 0.070
PT-Gauss$_{grid}$	0.543 ± 0.035	0.437 ± 0.037	0.383 ± 0.070
PT-Gauss$_{fix}$	0.543 ± 0.044	0.600 ± 0.032	0.383 ± 0.070

Table 4. Kendall-Tau correlations on each of the VTAB tasks when $N/K = 2$.

for those checkpoints with the same feature dimensions), the averaged Kendall-Tau correlation on the 9 tasks appears to be 0.481, which is higher than most of the other baselines.

In order to eliminate the effect caused by the differences of the penultimate feature dimensions, we compare all metrics on a subset that contains the ten checkpoints with the same feature dimensions 2048. The rest of the experiment settings are the same as before. The results of their averaged performance on the 9 VTAB tasks are reported in Table 7. We can see that PT-Gauss still achieves the highest correlations compared to any other metrics on those 10 checkpoints with the same feature dimensions.

100+ classes	Caltech101	Oxford-flowers	Sun397
LEEP	0.270 ± 0.008	0.163 ± 0.007	0.237 ± 0.010
$\mathcal{N}$-LEEP	0.803 ± 0.012	0.743 ± 0.011	0.840 ± 0.015
H-score	0.503 ± 0.038	0.393 ± 0.056	0.340 ± 0.053
LogME	0.450 ± 0.000	0.393 ± 0.004	0.420 ± 0.003
LINEAR	0.277 ± 0.008	0.220 ± 0.006	0.263 ± 0.011
LINEAR-VALID	0.804 ± 0.021	0.801 ± 0.022	0.816 ± 0.009
$\mathcal{N}$-PT-Dir	0.837 ± 0.018	0.793 ± 0.012	0.847 ± 0.017
$\mathcal{N}$-PT-Gam	0.840 ± 0.012	0.793 ± 0.009	0.843 ± 0.017
PT-Gauss$_{grid}$	0.823 ± 0.008	0.800 ± 0.006	0.757 ± 0.004
PT-Gauss$_{fix}$	0.823 ± 0.008	0.877 ± 0.006	0.797 ± 0.006

10-99 classes	Cifar-10	Oxford-IIIT Pet	SmallNorb
LEEP	0.073 ± 0.025	0.239 ± 0.009	-0.067 ± 0.029
$\mathcal{N}$-LEEP	0.693 ± 0.039	0.815 ± 0.017	0.100 ± 0.058
H-score	0.000 ± 0.065	0.239 ± 0.027	0.184 ± 0.050
LogME	0.130 ± 0.003	0.296 ± 0.007	-0.147 ± 0.006
LINEAR	0.183 ± 0.007	0.219 ± 0.006	-0.150 ± 0.007
LINEAR-VALID	0.630 ± 0.033	0.735 ± 0.033	-0.133 ± 0.044
$\mathcal{N}$-PT-Dir	0.683 ± 0.024	0.751 ± 0.041	-0.060 ± 0.059
$\mathcal{N}$-PT-Gam	0.683 ± 0.021	0.755 ± 0.036	-0.053 ± 0.054
PT-Gauss$_{grid}$	0.820 ± 0.013	0.748 ± 0.006	0.580 ± 0.007
PT-Gauss$_{fix}$	0.820 ± 0.009	0.808 ± 0.006	0.397 ±0.017

2-9 classes	Patch-Camelyon	DMLAB	CBIS-DDSM
LEEP	-0.090 ± 0.030	0.023 ± 0.037	0.137 ± 0.022
$\mathcal{N}$-LEEP	0.162 ± 0.039	-0.007 ± 0.100	0.133 ± 0.047
H-score	0.393 ± 0.056	0.032 ± 0.102	-0.072 ± 0.062
LogME	-0.123 ± 0.013	0.070 ± 0.003	0.277 ± 0.007
LINEAR	-0.043 ± 0.025	0.110 ± 0.006	0.300 ± 0.010
LINEAR-VALID	0.294 ± 0.065	-0.109 ± 0.118	-0.054 ± 0.043
$\mathcal{N}$-PT-Dir	0.164 ± 0.054	0.143 ± 0.094	0.113 ± 0.071
$\mathcal{N}$-PT-Gam	0.177 ± 0.050	0.157 ± 0.097	0.120 ± 0.068
PT-Gauss$_{grid}$	0.543 ± 0.035	0.277 ± 0.027	0.417 ± 0.060
PT-Gauss$_{fix}$	0.543 ± 0.044	0.577 ± 0.022	0.417 ± 0.060

Table 5. Kendall-Tau correlations on each of the VTAB tasks when $N/K = 5$.

B.8 Kendall-Tau Rank Correlation

We use the Kendall-Tau rank correlation coefficient to correlate between the transferability metric scores and the testing error of the finetuned checkpoints. Kendall-Tau correlation is a classic metric for estimating the correlation between rankings and has been used in previous similar works such as [34, 59, 29]. In particular, among C checkpoints $\boldsymbol{\theta}_i$ with test error $e(\boldsymbol{\theta}_i)$ and a metric $m(\boldsymbol{\theta}_i)$:

$$\tau = \frac{1}{C(C-1)} \sum_{i \neq j} \operatorname{sign}(m(\boldsymbol{\theta}_i) - m(\boldsymbol{\theta}_j)) \operatorname{sign}(e(\boldsymbol{\theta}_i) - e(\boldsymbol{\theta}_j)).$$

100+ classes	Caltech101	Oxford-flowers	Sun397
LEEP	0.320 ± 0.009	0.217 ± 0.016	0.290 ± 0.014
$\mathcal{N}$-LEEP	0.823 ± 0.010	0.793 ± 0.008	0.850 ± 0.011
H-score	0.497 ± 0.040	0.417 ± 0.042	0.470 ± 0.043
LogME	0.513 ± 0.003	0.467 ± 0.000	0.483 ± 0.000
LINEAR	0.327 ± 0.007	0.277 ± 0.004	0.370 ± 0.006
LINEAR-VALID	0.821 ± 0.015	0.827 ± 0.010	0.857 ± 0.014
$\mathcal{N}$-PT-Dir	0.827 ± 0.010	0.810 ± 0.006	0.880 ± 0.011
$\mathcal{N}$-PT-Gam	0.827 ± 0.010	0.810 ± 0.006	0.880 ± 0.011
PT-Gauss$_{grid}$	0.813 ± 0.009	0.767 ± 0.004	0.727 ± 0.008
PT-Gauss$_{fix}$	0.787 ± 0.009	0.820 ± 0.009	0.727 ± 0.006
10-99 classes	Cifar-10	Oxford-IIIT Pet	SmallNorb
LEEP	0.113 ± 0.022	0.226 ± 0.013	-0.103 ± 0.026
$\mathcal{N}$-LEEP	0.730 ± 0.012	0.822 ± 0.023	0.007 ± 0.049
H-score	0.234 ± 0.041	0.296 ± 0.035	0.425 ± 0.060
LogME	0.133 ± 0.000	0.413 ± 0.004	-0.133 ± 0.000
LINEAR	0.180 ± 0.015	0.256 ± 0.006	-0.170 ± 0.009
LINEAR-VALID	0.704 ± 0.035	0.811 ± 0.019	-0.070 ± 0.041
$\mathcal{N}$-PT-Dir	0.690 ± 0.022	0.832 ± 0.013	-0.183 ± 0.051
$\mathcal{N}$-PT-Gam	0.700 ± 0.018	0.835 ± 0.010	-0.180 ± 0.052
PT-Gauss$_{grid}$	0.700 ± 0.017	0.778 ± 0.011	0.557 ± 0.008
PT-Gauss$_{fix}$	0.757 ± 0.014	0.808 ± 0.006	0.263 ± 0.007
2-9 classes	Patch-Camelyon	DMLAB	CBIS-DDSM
LEEP	-0.090 ± 0.030	0.090 ± 0.037	0.147 ± 0.022
$\mathcal{N}$-LEEP	0.162 ± 0.039	0.223 ± 0.027	0.060 ± 0.150
H-score	0.393 ± 0.056	-0.057 ± 0.056	0.138 ± 0.042
LogME	-0.123 ± 0.013	0.070 ± 0.003	0.273 ± 0.006
LINEAR	-0.043 ± 0.025	0.080 ± 0.006	0.290 ± 0.014
LINEAR-VALID	0.294 ± 0.065	0.150 ± 0.105	-0.075 ± 0.126
$\mathcal{N}$-PT-Dir	0.164 ± 0.054	0.220 ± 0.050	0.017 ± 0.124
$\mathcal{N}$-PT-Gam	0.177 ± 0.050	0.217 ± 0.051	0.027 ± 0.126
PT-Gauss$_{grid}$	0.543 ± 0.035	0.443 ± 0.036	0.300 ± 0.040
PT-Gauss$_{fix}$	0.543 ± 0.044	0.463 ± 0.021	0.597 ± 0.033

Table 6. Kendall-Tau correlations on each of the VTAB tasks when $N/K = 10$.

More broadly, [34] explored various ranking measure including Top-k Recall/ accuracy, Pearson and Kendall-Tau. They showed that Kendall-Tau is highly correlated with the other metrics, and is a more stable indicator than other metrics such as the top-k accuracy. In Fig.1, we provide a visual illustration comparing PT-Gauss to two other methods. We see that a high τ value is a strong indicator for better metric-error correlation as well as picking the best checkpoints with lowest test error (PT-Gauss, left).

	K=2	K=5	K=10
LEEP	0.293	0.301	0.311
$\mathcal{N}$-LEEP	0.275	0.435	0.456
Hscore	-0.003	-0.096	-0.059
LogME	0.335	0.360	0.385
LINEAR	0.323	0.346	0.359
LINEAR-VALID	0.301	0.389	0.413
$\mathcal{N}$-PT-Dir	0.351	0.432	0.475
$\mathcal{N}$-PT-Gam	0.349	0.429	0.472
PT-Gauss$_{grid}$	0.414	0.511	0.495
PT-Gauss$_{fix}$	**0.433**	**0.521**	**0.527**

Table 7. Averaged Kendall-Tau correlations over the 9 VTAB tasks on the 10 checkpoints with the same feature dimensions 2048.

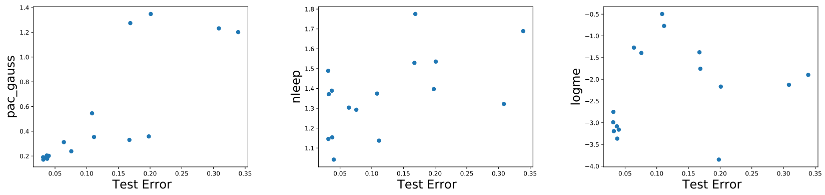

Fig. 1. Test error vs PT-Gauss (τ=0.77) , N-LEEP (τ=0.42), LogME (τ=0.15) of 16 checkpoints on the Cifar-10 task.

C Additional Details of the VQA Experiments

C.1 VQA Architecture

We applied the state-of-art VQA model architecture, which fuses the image and question representations in a multimodal Transformer model [55]. See Fig.2 as an illustration. On the image side, we take a global image feature from ResNet152 [25] pretrained on ImageNet [47] plus 100 region-of-interest image features from Faster R-CNN [44] pretrained on Visual Genome [31]. The parameters of both ResNet152 and Faster R-CNN are frozen during training. On the question side, we use the text-encoder of a pretrained T5-base checkpoint [42]. Finally, the decoder takes the [GO] symbol as the input and applies cross-attention to the outputs from the multimodal encoder, and outputs the answer.

C.2 VQA Datasets Descriptions

1. VQA v2.0: Visual Question Answering (VQA) v2.0 [23] is designed for answering open-ended questions about images. These questions require an understanding of vision, language and commonsense knowledge to answer. It is the second version of the VQA dataset [2]. It has 265,016 images from COCO and abstract scenes and at least 3 questions (5.4 questions on average) per image.

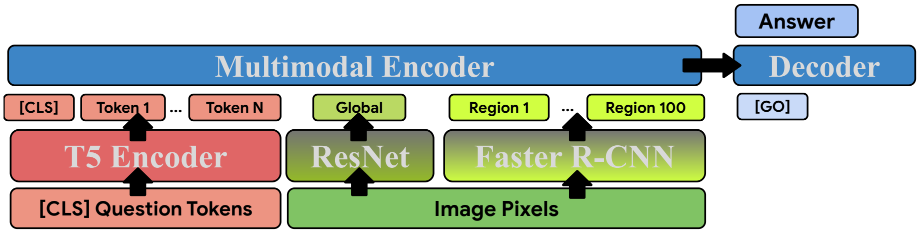

Fig. 2. VQA architecture used in our experiments. The text encoder is initialized from a T5-base checkpoint, while the multimodal encoder is initialized from scratch. The parameters of ResNet152 and Faster R-CNN are frozen during VQA training.

2. V7W: Visual7W [63] is a large-scale visual question answering dataset, with object-level groundings and multimodal answers. Each question starts with one of the seven "W"s: what, where, when, who, why, how and which. It is collected from 47,300 COCO images and it has 327,929 QA pairs, together with 1,311,756 human-generated multiple-choices and 561,459 object groundings from 36,579 categories.
3. GQA: The GQA dataset [28] centers around real-world reasoning, scene understanding and compositional question answering. It consists of 113K images and 22M questions of assorted types and varying compositionality degrees, measuring performance on an array of reasoning skills such as object and attribute recognition, transitive relation tracking, spatial reasoning, logical inference and comparisons.
4. CNETVQA: the CNETVQA dataset was created based on the ConceptNet [50]. In particular, a T5 model was first finetuned for question generation with VQA v2.0 dataset, where each training example included a pair of phrases as the input: a randomly selected entity from the question and the answer, as well as the original question as the target. Once the T5-based question generation model was trained, it was applied on the edges of the ConceptNet which created about 300k question-answer pairs. Next, each of the QA pairs was matched by the top five images from the Google Image Search. After filtering out some too large or too small images, the final CNETVQA dataset contains about 1M total examples.
5. TP-COLOR-COCO, TP-COLOR-CC3M, TP-COLOR-CC12M: Given image captions, we created three template-based VQA datasets consisting of visual questions about colors following a similar approach to color question generation proposed in COCOQA [43]. More specifically, we first detected color mentions in the captions using a list of simple Wikipedia colors. We then used SpaCy dependency parsing instead of Stanford constituency parsing to extract the noun or the noun phrase associated with each color mention, as well as to group multiple colors for the same noun together. Finally, we filled in the templates (only singular variants shown): What color is the/this/that [object], What is the color of the/this/that [object], Is the/this/that [object] [color], Is the/this/that [object] [wrong color]. We ex-

plored three sources of image captions: COCO-Captions [13], CC3M [49], and CC12M [12]. The number of question-answer pairs for TP-COLOR-COCO, TP-COLOR-CC3M, TP-COLOR-CC12M are 2.1M, 7,1M, and 38.9M, respectively.

6. VQ2A-COCO, VQ2A-CC3M: We also took another approach to create VQA datasets from image captions, following [11]. These are the datasets generated by selecting various types of answer candidates from captions and then using a T5 XXL model trained for question generation and answering to generate questions and perform filtering. This results in 3.5M and 13.3M question-answer pairs for VQ2A-COCO and VQ2A-CC3M, respectively.

C.3 Finetuning

Similar to the NeuCRaB experiments, the finetunings on OKVQA was also done in two ways for each pretrained checkpoint: a full-model finetuning of the checkpoints following the learning schedule at pretraining time for another 100,000 iterations; and 5 top-layer-only finetunings using an L-BFGS solver with weight decay $\frac{1}{B} \cdot \{0.01, 0.1, 1., 10., 100.\}$, where B is the size of the training set. The lowest test error of the above finetunings was used as the testing error of the checkpoint on the downstream task.

C.4 Computation Platform

The checkpoint pretraining, the pretrained feature extractions as well as the pretrained model finetunings were all done on the Google Cloud V2 4x4 TPUs. All the transferability metrics were computed on the Google Cloud Intel Skylake CPU (2GHz per core) with 1 core and 5GB RAM per run.

C.5 Hyperparameters of PACTran-Gaussian

In Fig.12, we did an analysis of the different hyperparameters on the OKVQA experiment. We see a similar pattern to the previous ones in B.5, except the optimal std ratio appears to be slightly lower ($\simeq 0.5$).

C.6 GFLOPS in the OKVQA Experiment

In Table 8 we report the GFLOPS of running the metrics as well as the GFLOPS of the feature extraction stage from the pretrained checkpoints in the OKVQA experiment. As we can see, the bottleneck is also the penultimate feature extraction, which is about 3-4 orders of magnitude slower than running the metrics themselves.

GFLOPS	$N = 40$	$N = 100$	$N = 200$
LEEP	7.44E-3	1.85E-2	3.69E-2
$\mathcal{N}$-LEEP	1.85E-2	1.58E-1	3.69E-1
Hscore	6.82E0	6.86E0	6.92E0
LogME	6.83E0	6.87E0	6.93E0
LINEAR	8.54E-1	2.10E0	4.19E0
LINEAR-VALID	2.85E-1	7.03E-1	1.40E0
PT-Dir	7.58E-2	1.86E-1	3.71E-1
PT-Gam	7.44E-2	1.85E-1	3.70E-1
$\mathcal{N}$-PT-Dir	1.86E-2	1.58E-1	7.93E-1
$\mathcal{N}$-PT-Gam	1.86E-2	1.58E-1	7.93E-1
PT-Gauss$_{grid}$	8.58E-1	2.11E0	4.21E0
Penultimate Feature (6, 3)	8.89E2	2.22E3	1.22E3
Penultimate Feature (9, 5)	1.05E3	2.63E3	3.05E3
Penultimate Feature (12, 7)	1.16E3	5.27E3	6.09E3

Table 8. GFLOPS of running each metrics and the penultimate-layer feature extractions on OKVQA.

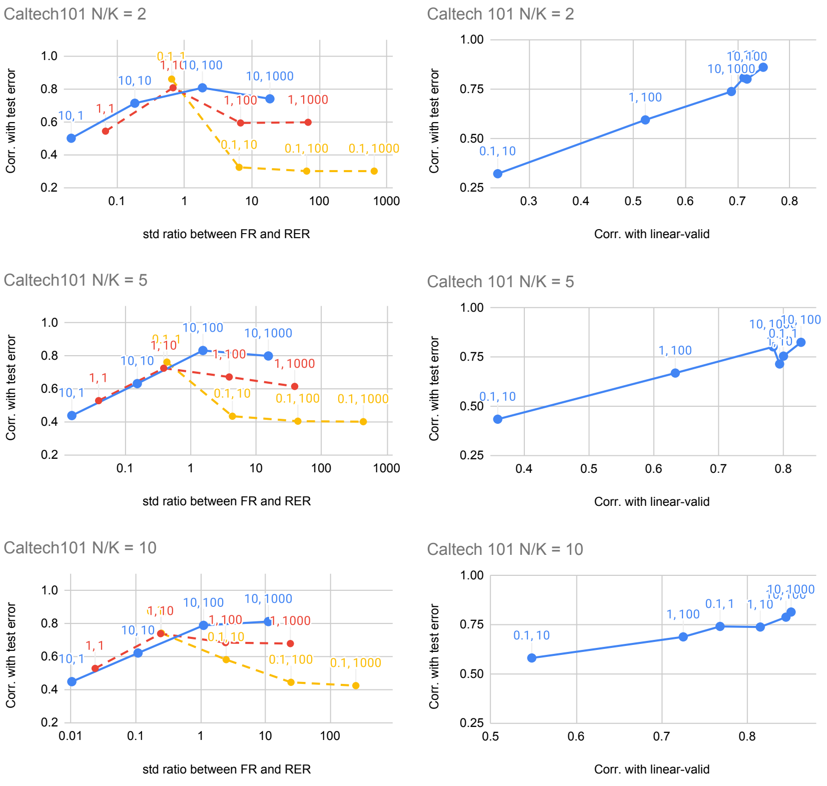

Fig. 3. PACTran-Gaussian hyperparameter studies on Caltech101. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

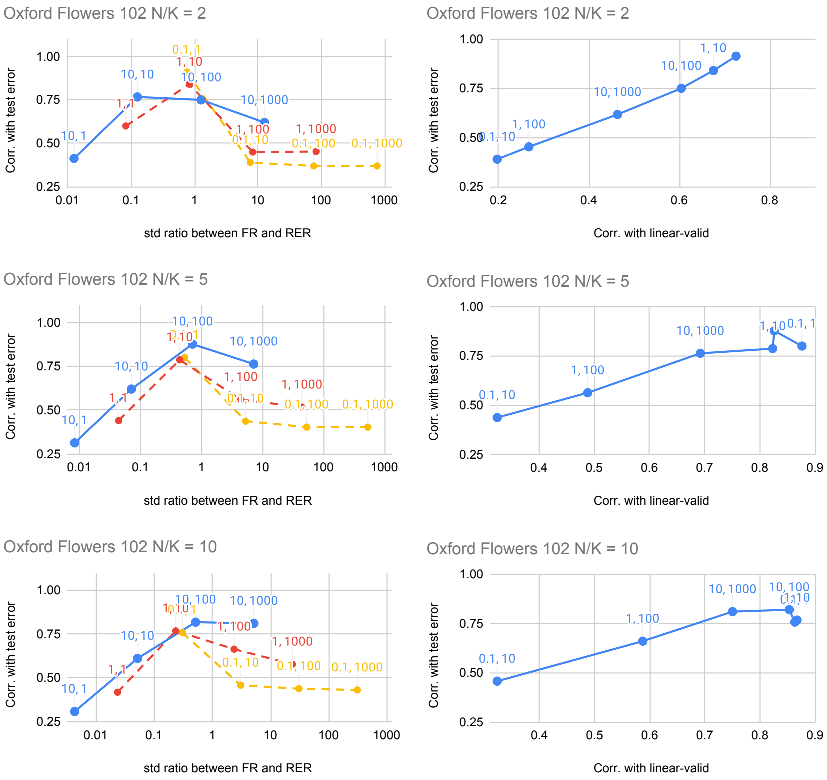

Fig. 4. PACTran-Gaussian hyperparameter studies on Oxford Flowers102. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

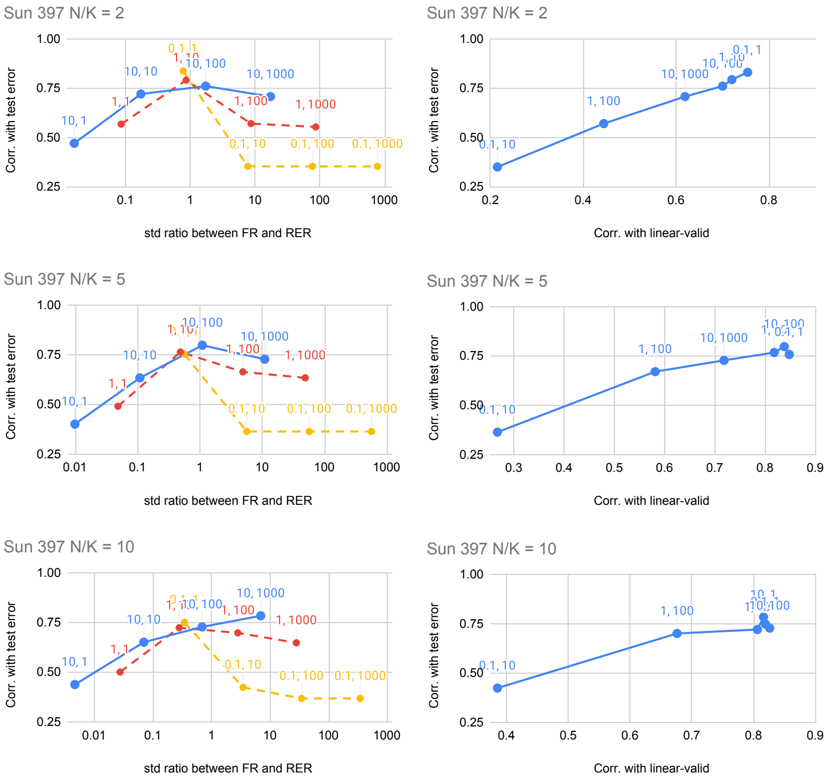

Fig. 5. PACTran-Gaussian hyperparameter studies on Sun397. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

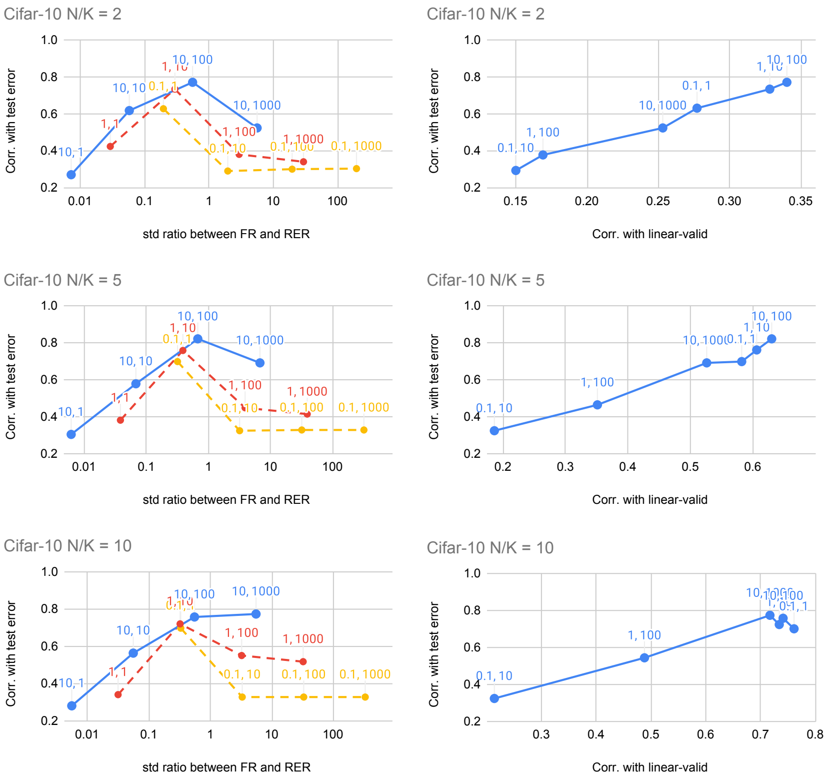

Fig. 6. PACTran-Gaussian hyperparameter studies on Cifar-10. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

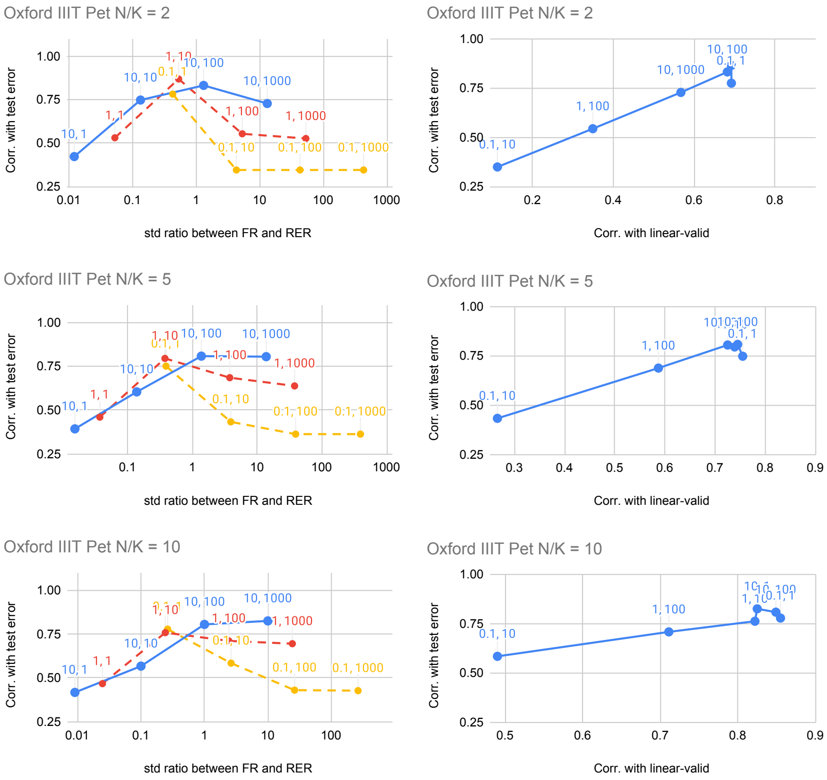

Fig. 7. PACTran-Gaussian hyperparameter studies on Oxford IIIT Pet. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

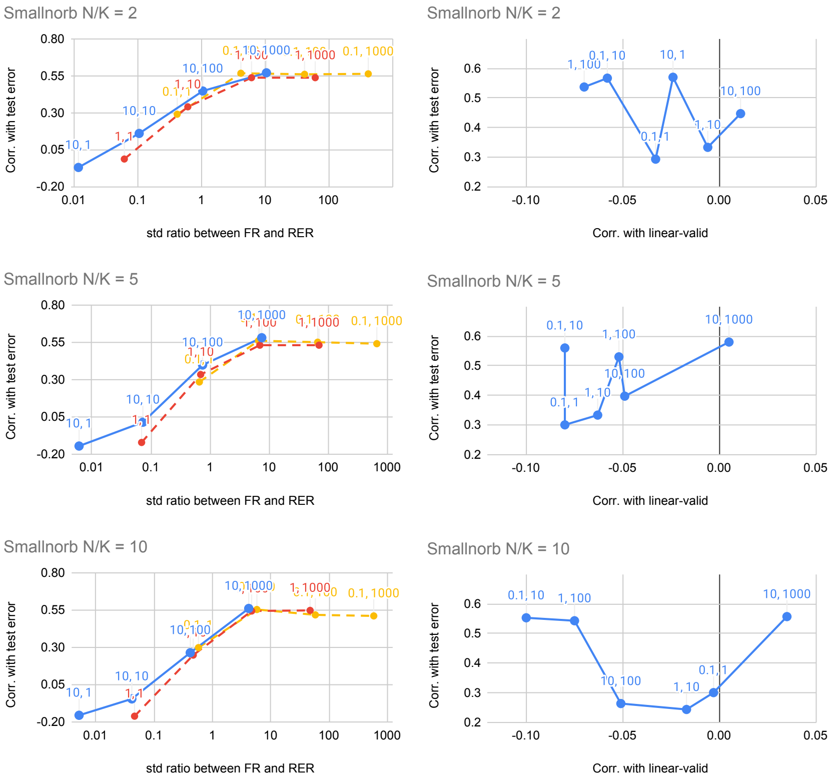

Fig. 8. PACTran-Gaussian hyperparameter studies on Smallnorb. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

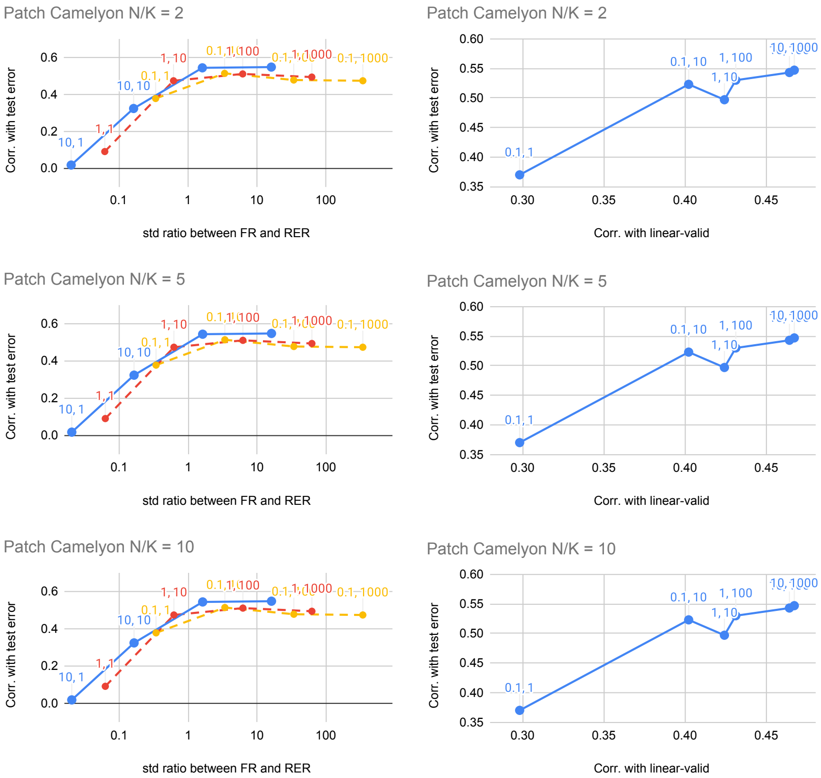

Fig. 9. PACTran-Gaussian hyperparameter studies on Patch Camelyon. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

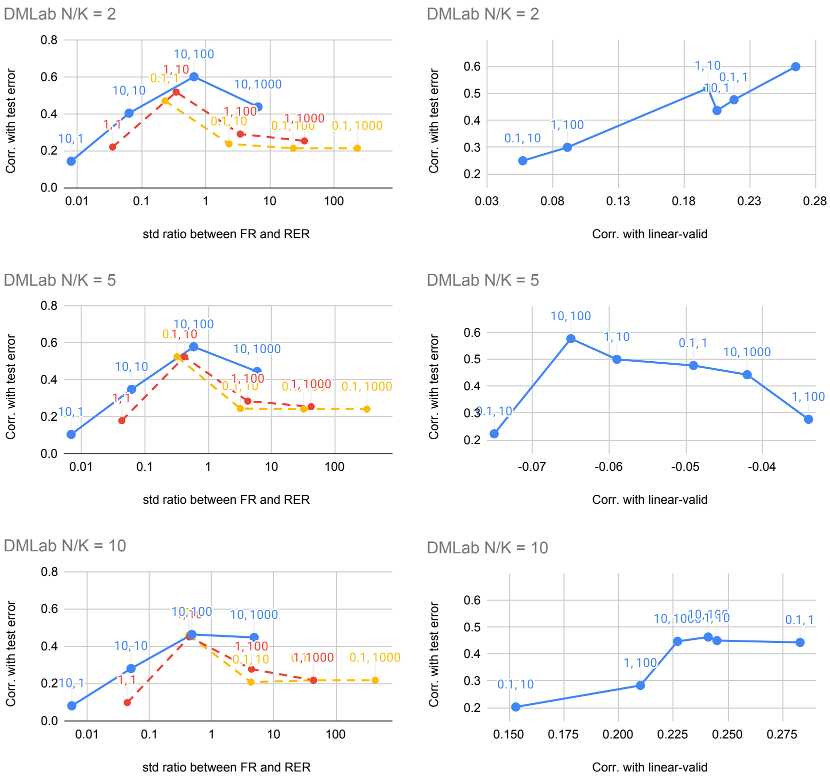

Fig. 10. PACTran-Gaussian hyperparameter studies on DMLab. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

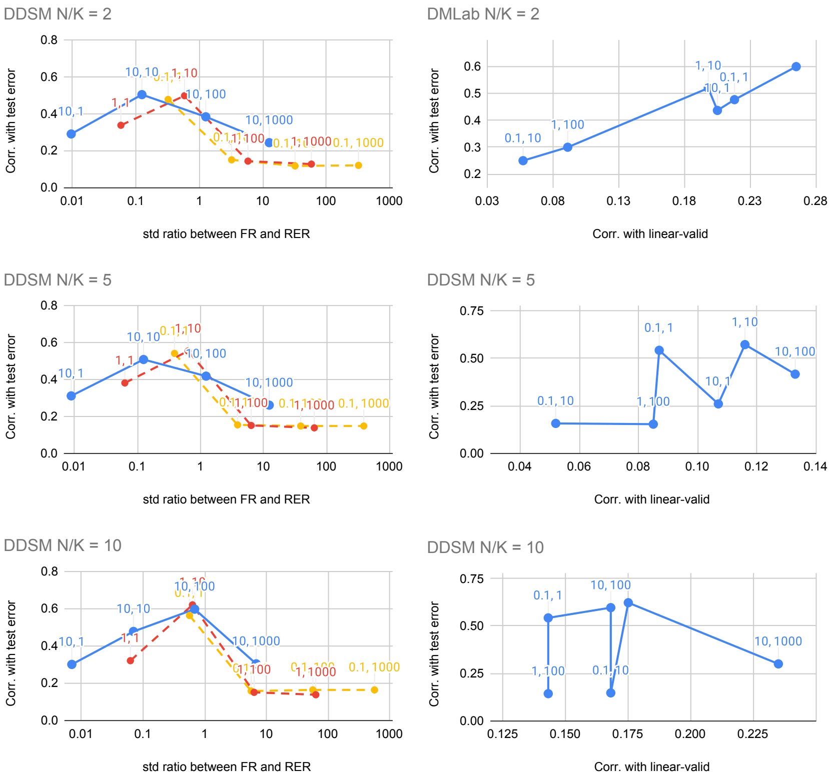

Fig. 11. PACTran-Gaussian hyperparameter studies on DDSM. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

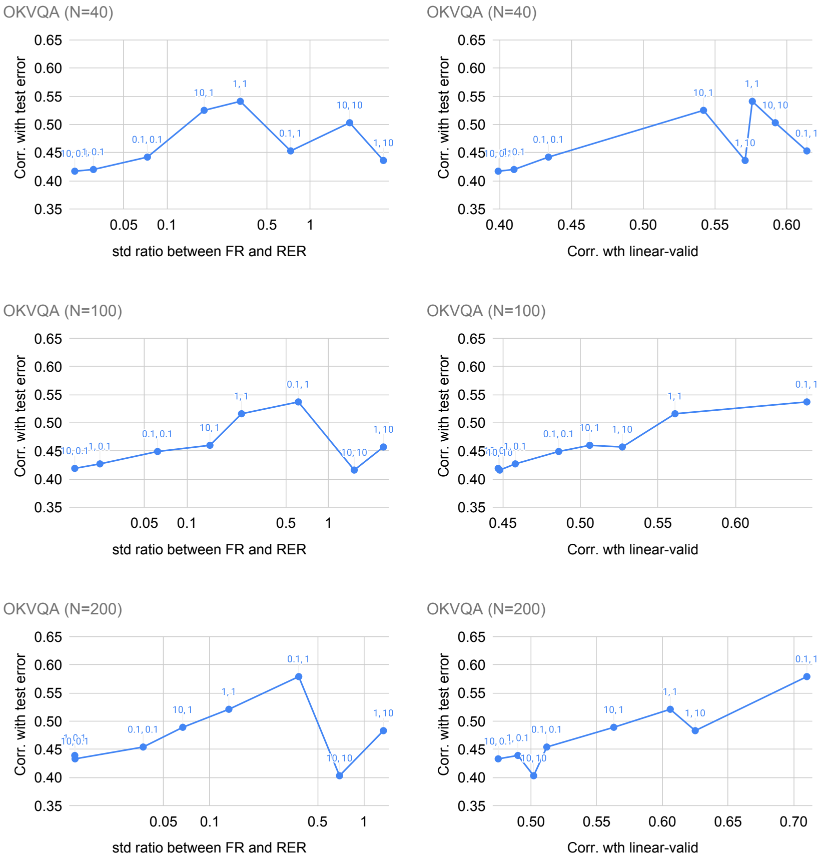

Fig. 12. PACTran-Gaussian hyperparameter studies on OKVQA. Hyperparameters are labeled as (a, b). High y-value indicates a good correlation with the downstream test error.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: VQA: Visual question answering. In: ICCV (2015)
3. Archambeau, C., Caron, F.: Plackett-luce regression: A new bayesian model for polychotomous data. arXiv preprint arXiv:1210.4844 (2012)
4. Bao, Y., Li, Y., Huang, S.L., Zhang, L., Zheng, L., Zamir, A., Guibas, L.: An information-theoretic approach to transferability in task transfer learning. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 2309–2313. IEEE (2019)
5. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American statistical Association **112**(518), 859–877 (2017)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research **3**, 993–1022 (2003)
7. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., Liang, P.: On the opportunities and risks of foundation models (2021)
8. Bousquet, O., Boucheron, S., Lugosi, G.: Introduction to statistical learning theory. In: Summer school on machine learning. pp. 169–207. Springer (2003)
9. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis (2019)
10. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
11. Changpinyo, S., Kukliansky, D., Szpektor, I., Chen, X., Ding, N., Soricut, R.: All you may need for vqa are image captions. In: NAACL (2022)

12. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR (2021)
13. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft COCO Captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
14. Ding, N., Chen, X., Levinboim, T., Goodman, S., Soricut, R.: Bridging the gap between practice and pac-bayes theory in few-shot meta-learning. Advances in Neural Information Processing Systems **34** (2021)
15. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction (2016)
16. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014), https://proceedings.neurips.cc/paper/2014/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf
17. Dziugaite, G.K., Roy, D.M.: Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008 (2017)
18. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Pattern Recognition Workshop (2004)
19. Germain, P., Bach, F., Lacoste, A., Lacoste-Julien, S.: PAC-bayesian theory meets bayesian inference. Advances in Neural Information Processing Systems **29**, 1884–1892 (2016)
20. Germain, P., Lacasse, A., Laviolette, F., Marchand, M.: PAC-bayesian learning of linear classifiers. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 353–360 (2009)
21. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations (2018)
22. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
26. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks (2016)
27. Huang, S.L., Makur, A., Wornell, G.W., Zheng, L.: On universal features for high-dimensional learning and inference. arXiv preprint arXiv:1911.09105 (2019)
28. Hudson, D.A., Manning, C.D.: GQA: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019)

29. Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic generalization measures and where to find them. In: ICLR (2020), https://arxiv.org/abs/1912.02178
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
31. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual Genome: Connecting language and vision using crowdsourced dense image annotations. IJCV **123**(1), 32–73 (2017)
32. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
33. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2**, II–104 Vol.2 (2004)
34. Li, Y., Jia, X., Sang, R., Zhu, Y., Green, B., Wang, L., Gong, B.: Ranking neural checkpoints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2663–2673 (2021)
35. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
36. McAllester, D.A.: Some PAC-bayesian theorems. Machine Learning **37**(3), 355–363 (1999)
37. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. Advances in neural information processing systems **30** (2017)
38. Nguyen, C., Hassner, T., Seeger, M., Archambeau, C.: Leep: A new measure to evaluate transferability of learned representations. In: International Conference on Machine Learning. pp. 7294–7305. PMLR (2020)
39. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008)
40. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles (2017)
41. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
42. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR (2020)
43. Ren, M., Kiros, R., Zemel, R.: Exploring models and data for image question answering. Advances in neural information processing systems **28** (2015)
44. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
45. Rothfuss, J., Fortuin, V., Josifoski, M., Krause, A.: Pacoh: Bayes-optimal meta-learning with pac-guarantees. In: International Conference on Machine Learning. pp. 9116–9126. PMLR (2021)
46. Rubenstein, P., Bousquet, O., Djolonga, J., Riquelme, C., Tolstikhin, I.O.: Practical and consistent estimation of f-divergences. In: Advances in Neural Information Processing Systems. vol. 32 (2019), https://proceedings.neurips.cc/paper/2019/file/3147da8ab4a0437c15ef51a5cc7f2dc4-Paper.pdf

47. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
48. Sawyer-Lee, R., Gimenez, F., Hoogi, A., Rubin, D.: Curated breast imaging subset of ddsm (2016). https://doi.org/10.7937/k9/tcia.2016.7o02s9cy, https://wiki.cancerimagingarchive.net/x/lZNXAQ
49. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018)
50. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: AAAI Conference on Artificial Intelligence. pp. 4444–4451 (2017)
51. Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders (2019)
52. Tran, A.T., Nguyen, C.V., Hassner, T.: Transferability and hardness of supervised classification tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1395–1405 (2019)
53. Tripuraneni, N., Jordan, M., Jin, C.: On the theory of transfer learning: The importance of task diversity. Advances in Neural Information Processing Systems **33**, 7852–7862 (2020)
54. Tsuzuku, Y., Sato, I., Sugiyama, M.: Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using PAC-Bayesian analysis. In: Proceedings of the 37th International Conference on Machine Learning. pp. 9636–9647 (2020)
55. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
56. Veeling, B.S., Linmans, J., Winkens, J., Cohen, T., Welling, M.: Rotation equivariant cnns for digital pathology (Sep 2018). https://doi.org/10.1007/978-3-030-00934-2-24
57. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
58. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3485–3492 (June 2010). https://doi.org/10.1109/CVPR.2010.5539970
59. You, K., Liu, Y., Wang, J., Long, M.: Logme: Practical assessment of pre-trained models for transfer learning. In: International Conference on Machine Learning. pp. 12133–12143. PMLR (2021)
60. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1476–1485 (2019). https://doi.org/10.1109/ICCV.2019.00156
61. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. arXiv preprint arXiv:1910.04867 (2019)
62. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)
63. Zhu, Y., Groth, O., Bernstein, M., Li, F.F.: Visual7W: Grounded question answering in images. In: CVPR (2016)