APPENDIX

A GRADIENT DESCENT IN LINEAR REGRESSION

Theorem 4. *Let* $(X, y) \subset \mathbb{R}^{d\times n} \times \mathbb{R}^n$ *with* X *of rank* r *and* $X = U\Sigma V^T$ *its singular value decomposition (SVD). Given an initialization* $w^{(0)} = \mathbf{0}$*, gradient descent used to solve:*

$$\arg\min_{w\in\mathbb{R}^d} \frac{1}{2}\|y - wX\|^2.$$

with learning rate $\eta < \frac{1}{\lambda_{max}(XX^T)}$ *converges to:*

$$w^{(\infty)} = yV\Sigma^{\dagger}U^T, \ \textit{where}\ \Sigma^{\dagger} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \dots 0 & \\ 0 & \frac{1}{\sigma_2} & \dots 0 & \mathbf{0}_{r\times d-r} \\ 0 & \dots & \frac{1}{\sigma_r} & \\ & \mathbf{0}_{d-r\times r} & & \mathbf{0}_{d-r\times d-r} \end{bmatrix}.$$

Proof. Let $S = XX^T$ and $S' = yX^T$. Then, $w^{(t+1)} = w^{(t)}(I - \eta S) + \eta S'$. Now we directly solve the recurrence relation; namely,

$$w^{(t)} = \eta S'((I - \eta S)^{t-1} + (I - \eta S)^{t-2} + ... + (I - \eta S)^1 + I).$$

Let $X = U\Sigma V^T$ denote the singular value decomposition of X where $\{\sigma_1, \dots, \sigma_r\}$ are the non-zero entries of Σ and r is the rank of X. Then, $S = U\Sigma^2U^T$, and $S' = yV\Sigma U^T$. Thus, we can simplify the recurrence relation:

$$w^{(t)} = \eta S'U((I - \eta\Sigma^2)^{t-1} + (I - \eta\Sigma^2)^{t-2} + ... + (I - \eta\Sigma^2)^1 + I)U^T.$$

Since $(I - \eta\Sigma^2)^{t-1} + (I - \eta\Sigma^2)^{t-2} + ... + (I - \eta\Sigma^2)^1 + I$ is a geometric series, for $\eta < \frac{1}{\sigma_1^2}$, we have:

$$\begin{aligned} w^{(t)} &= \eta S'U\Sigma^{+}U^T, \\ \Sigma^{+} &= \begin{bmatrix} \frac{1-(1-\eta\sigma_1^2)^t}{\eta\sigma_1^2} & 0 & \dots 0 & \\ 0 & \frac{1-(1-\eta\sigma_2^2)^t}{\eta\sigma_2^2} & \dots 0 & \mathbf{0}_{r\times d-r} \\ 0 & \dots & \frac{1-(1-\eta\sigma_r^2)^t}{\eta\sigma_r^2} & \\ & \mathbf{0}_{d-r\times r} & & t\mathbf{I}_{d-r\times d-r} \end{bmatrix}. \end{aligned}$$

Now substituting in $S' = yV\Sigma U^T$ gives us:

$$\begin{aligned} w^{(t)} &= yV\Sigma^{\dagger}U^T, \\ \Sigma^{\dagger} &= \begin{bmatrix} \frac{1-(1-\eta\sigma_1^2)^t}{\sigma_1} & 0 & \dots 0 & \\ 0 & \frac{1-(1-\eta\sigma_2^2)^t}{\sigma_2} & \dots 0 & \mathbf{0}_{r\times d-r} \\ 0 & \dots & \frac{1-(1-\eta\sigma_r^2)^t}{\sigma_r} & \\ & \mathbf{0}_{d-r\times r} & & \mathbf{0}_{d-r\times d-r} \end{bmatrix}. \end{aligned}$$

Lastly, we can take the limit as $t \to \infty$ to conclude that

$$w^{(\infty)} = \lim_{t\to\infty} w^{(t)} = yV\Sigma^{\dagger}U^T, \ \text{where}\ \Sigma^{\dagger} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \dots 0 & \\ 0 & \frac{1}{\sigma_2} & \dots 0 & \mathbf{0}_{r\times d-r} \\ 0 & \dots & \frac{1}{\sigma_r} & \\ & \mathbf{0}_{d-r\times r} & & \mathbf{0}_{d-r\times d-r} \end{bmatrix}.$$

□

Note that the proof above can be easily extended to the setting of a random initialization $w^{(0)}$.

B DISTRIBUTION OF SINGULAR VECTORS OF A RANDOM MATRIX

Proof. We use the rotational invariance of the multivariate isotropic Gaussian. If A is an orthonormal matrix, then we have:

$$x^T I^{-1} x = x^T A^T I^{-1} A x = (Ax)^T I^{-1} (Ax).$$

Now, suppose A, B are both orthonormal matrices, then we have:

$$AXB^T = (A \otimes B) X_v,$$

where $X_v \in \mathbb{R}^{dn}$ is the row-major vectorization of X and $\otimes$ is the Kronecker product. Now, since A, B are orthonormal, we have that $A \otimes B$ is orthonormal. Hence, AXB^T must have the same distribution as X, and thus the singular vectors of AXB^T must have the same distribution as those of X. Since singular vectors lie on $\mathcal{S}_{d-1}$ and since the distribution is rotation invariant, we conclude that the singular vectors are uniformly distributed on $\mathcal{S}_{d-1}$. ☐

C TRAINING DETAILS

We now describe the training methodology we used to train pre-trained models on ImageNet32 and CIFAR10. The optimizer, initialization, learning rate, and seeds used to train the ResNets in Figure 2 and 3 are presented in Figure 4. Note that all of our models were trained with mean squared error, as discussed in (16). We trained models on ImageNet32 for 150 epochs and on CIFAR10 for 50 epochs. We then saved the model with the highest validation accuracy.

Dataset	Optimizer, Learning Rate	Initialization	Seed
Classes 1 and 2, ImageNet32	Adam, 1e-4	Default Pytorch Initialization	200
CIFAR10	Adam, 1e-4	Default Pytorch Initialization	200

Figure 4: An overview of the optimizer, learning rate, initialization, and seeds used to fine-tune pre-trained models on ImageNet32 and CIFAR10.

For all experiments, we used the PyTorch deep learning library (25). We trained our models on a shared server with 1 Titan Xp and 2 GeForce GTX 1080 Ti's. We only used 1 GPU at a time for training neural networks and applying LLBoost.

D ADDITIONAL EXPERIMENTAL DETAILS

In this section, we provide the following additional details regarding the experiments in Figure 2 and 3:

1. The number of components used in the low-rank approximations for a full rank training feature matrix (Figure 5).
2. The size of the perturbation produced by LLBoost and the values of γ used for the models in Figure 2 (Figure 6).
3. A comparison between training time and the time taken for LLBoost to improve the models in Figure 2 (Figure 7).

E PERFORMANCE OF PROJECTED STANDARD NORMAL PERTURBATIONS

In Figure 2, we demonstrated that perturbing the last layer without projecting to the space orthogonal to the feature matrix provided a drastic decrease in the training and validation accuracy. In Figure 8, we illustrate the impact of using a perturbation that is randomly sampled from a standard normal and then projected to the space orthogonal to the feature matrix. Again, we see that the validation accuracies can drop significantly for larger datasets in this case. Note that including the projection operator preserves the training accuracy in all cases, as is guaranteed by Lemma 1.

F LOW RANK APPROXIMATIONS FOR FEATURE MATRICES

As discussed in Section 4, when the feature matrix, X, is full rank, we needed to use a low-rank approximation such that the space orthogonal to X. In this section, we discuss our method of

Dataset	Model	Number of Components
2600 Ex. ImageNet-32	ResNet-18	12
2600 Ex. Imagenet-32	ResNet-34	12
2600 Ex. Imagenet-32	ResNet-50	1548
CIFAR10	ResNet-18	212
ImageNet	FixResNext-101	1000

Figure 5: The rank of the approximation used for the training feature matrix, X, when X was full rank.

Dataset	Model	γ	Train/Val. Acc. (Original)	Train/Val. Acc. (Ours)	Perturbation
100 Ex. ImageNet32	ResNet-18	0.226	100%/80%	100%/84%	0.21
2600 Ex. ImageNet32	ResNet-18*	11.314	99.96%/95%	99.96%/97%	11.210
100 Ex. ImageNet32	ResNet-34	2.263	100%/85%	100%/87%	2.263
2600 Ex. ImageNet32	ResNet-34*	4.525	99.77%/95%	99.77%/98%	4.474
100 Ex. ImageNet32	ResNet-50	0.453	100%/83%	100%/89%	0.440
200 Ex. ImageNet32	ResNet-50	0.453	100%/87%	100%/93%	0.431
500 Ex. ImageNet32	ResNet-50	4.525	99.6%/91%	99.6%/93%	3.962
800 Ex. ImageNet32	ResNet-50	9.051	100%/94%	100%/98%	7.215
1000 Ex. ImageNet32	ResNet-50	33.941	100%/93%	100%/97%	23.784
2000 Ex. ImageNet32	ResNet-50	45.255	99.85%/95%	99.85%/97%	6.338
2600 Ex. ImageNet32	ResNet-50*	36.204	99.88%/95%	99.88%/99%	17.364
ImageNet	FixResNext-101*	10	94.924%/86.26%	94.924%/86.34%	0.4428
CIFAR10	ResNet-18*	2.0	99.99%/95.05%	99.99%/95.25%	1.724

Figure 6: An extended version of Figure 2 that includes the choice of γ considered and the size of the perturbation (in Frobenius norm) produced by LLBoost. $*$'s indicate the use of low-rank approximations for full rank training feature matrices.

choosing the number of components of the SVD to keep for producing the low-rank approximation for X. We then present how the number of components selected affects the performance of LLBoost.

In Figure 9, we visualize the normalized singular values of the feature matrix for models from Figure 2. In Figure 9A, we do not use a low-rank approximation as the size of the dataset is already smaller than the number of features. In Figure 9B, the feature matrices are full rank, and so we use a low-rank approximation for the feature matrix with the number of components selected shown red. In particular, we chose a number of components that is well past the elbow in the curve so that there was not a significant drop in training accuracy.

In Figure 10, we demonstrate how the number of components selected for the low-rank approximation affects the validation accuracy of LLBoost. In particular, we observe that using a lower rank approximation generally increases the improvement provided by LLBoost. This matches the intuition provided by Proposition 2: when the space orthogonal to the training feature matrix, X, is large, there is no reason to believe that the best linear solution lies in the span of X. Hence, sampling the space orthogonal to X yields an improvement. We note that since only a few singular values of X are large, there is no impact to the training accuracy when using a low-rank approximation for X (shown in the second column of the tables in Figure 10).

G LLBOOST APPLIED TO TRAIN, VALIDATION, TEST SPLITS

In Figure 2 and Figure 3, we demonstrated that LLBoost improves the validation accuracy of pre-trained models without impacting the training accuracy. To ensure that LLBoost is not overfitting

Dataset	Model	Training Time	Correction Time (s)
100 Ex. ImageNet32	ResNet-18	77.753 sec	0.116 sec
2600 Ex. ImageNet32	ResNet-18*	1020.413 sec	0.112 sec
100 Ex. ImageNet32	ResNet-34	122.85 sec	0.111 sec
2600 Ex. ImageNet32	ResNet-34*	1397.989 sec	0.098 sec
100 Ex. ImageNet32	ResNet-50	164.07 sec	0.113 sec
200 Ex. ImageNet32	ResNet-50	190.473 sec	0.111 sec
500 Ex. ImageNet32	ResNet-50	407.454 sec	0.105 sec
800 Ex. ImageNet32	ResNet-50	628.997 sec	0.137 sec
1000 Ex. ImageNet32	ResNet-50	1054.061 sec	0.087 sec
2000 Ex. ImageNet32	ResNet-50	1996.991 sec	0.129 sec
2600 Ex. ImageNet32	ResNet-50*	2488.621 sec	0.11 sec
ImageNet	FixResNext-101*	~1 day/epoch	7.59 hr
CIFAR10	ResNet-18*	1.35 hr	15.36 min

Figure 7: A comparison between the training time and LLBoost correction time for models from Figure 2. For the ImageNet32 models, the third column represents the time to compute the validation accuracy for $100,000$ samples from LLBoost. For CIFAR10 and ImageNet, the time additionally includes the cost of computing the perturbation for LLBoost. $*$'s indicate the use of low-rank approximations for full rank training feature matrices.

Dataset	Model	Train/Val. Acc. (Original)	Train/Val. Acc. (Standard Normal @ Perp)
100 Ex. ImageNet32	ResNet-18	100%/80%	100%/76%
2600 Ex. ImageNet32	ResNet-18*	99.96%/95%	99.96%/96%
100 Ex. ImageNet32	ResNet-34	100%/85%	100%/75%
2600 Ex. ImageNet32	ResNet-34*	99.77%/95%	99.77%/88%
100 Ex. ImageNet32	ResNet-50	100%/83%	100%/76%
200 Ex. ImageNet32	ResNet-50	100%/87%	100%/77%
500 Ex. ImageNet32	ResNet-50	99.6%/91%	99.6%/79%
800 Ex. ImageNet32	ResNet-50	100%/94%	100%/87%
1000 Ex. ImageNet32	ResNet-50	100%/93%	100%/96%
2000 Ex. ImageNet32	ResNet-50	99.85%/95%	99.85%/97%
2600 Ex. ImageNet32	ResNet-50*	99.88%/95%	99.88%/99%
ImageNet	FixResNext-101*	94.924%/86.26%	94.924/18.54%
CIFAR10	ResNet-18*	99.99%/95.05%	99.99%/92.35%

Figure 8: A demonstration that using samples from a standard normal projected onto the space orthogonal to the training data leads to a decrease in validation accuracy but has no impact on training accuracy. $*$'s indicate the use of low-rank approximations for full rank training feature matrices.

the validation set, we additionally split the validation data into validation and test data and check that LLBoost improves validation and test accuracy without impacting training accuracy[3].

---

[3] For ImageNet32, the validation set size is only 100 examples, and so we split the training set and re-train.

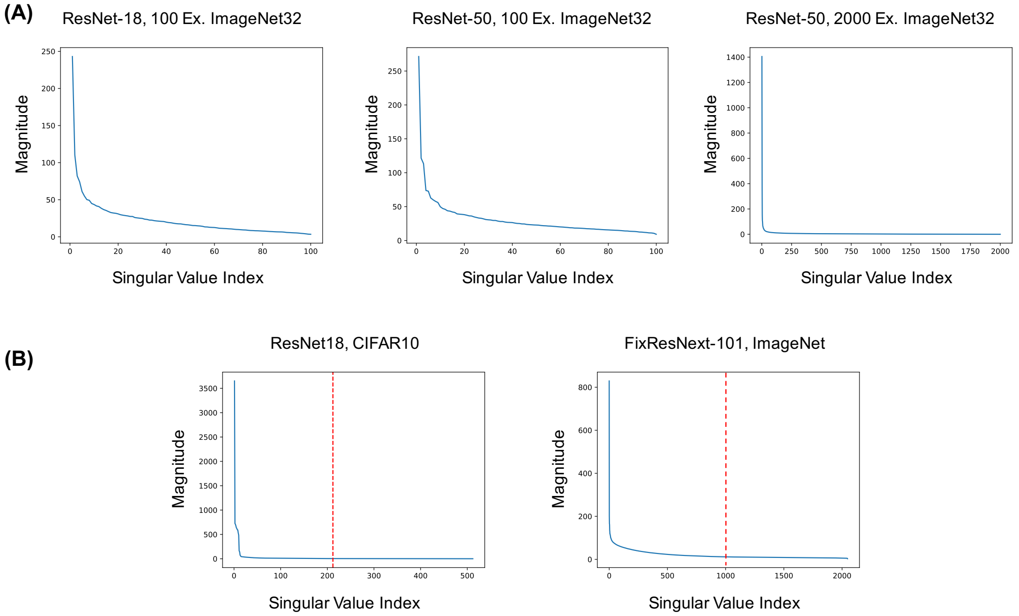

Figure 9: Visualizations of the singular values of training feature matrices for models from 2. (A) The singular values of the training feature matrices for small datasets. (B) The singular values of full rank training feature matrices from large datasets. The red vertical line indicates the size of the approximation used for Figure 2.

ImageNet32 (2600 Ex.), ResNet-50	Train Acc. (Original)	Train Acc. (Approx.)	Val. Acc.
Original Feature Matrix	99.88%	99.88%	95%
Rank 1548 Approx.	99.88%	99.88%	99%
Rank 1648 Approx.	99.88%	99.88%	99%
Rank 1748 Approx.	99.88%	99.88%	98%
Rank 1848 Approx.	99.88%	99.88%	98%
Rank 1898 Approx.	99.88%	99.88%	97%
Rank 1948 Approx.	99.88%	99.88%	97%
Rank 1973 Approx.	99.88%	99.88%	96%
Rank 1998 Approx.	99.88%	99.88%	96%
Rank 2023 Approx.	99.88%	99.88%	96%

CIFAR10, ResNet-18	Train Acc. (Original)	Train Acc. (Approx.)	Val. Acc.
Original Feature Matrix	99.99%	99.99%	95.05%
Rank 50 Approx.	99.99%	99.99%	95.25%
Rank 75 Approx.	99.99%	99.99%	95.25%
Rank 100 Approx.	99.99%	99.99%	95.24%
Rank 150 Approx.	99.99%	99.99%	95.24%
Rank 200 Approx.	99.99%	99.99%	95.23%
Rank 212 Approx.	99.99%	99.99%	95.25%
Rank 300 Approx.	99.99%	99.99%	95.2%
Rank 400 Approx.	99.99%	99.99%	95.19%
Rank 500 Approx.	99.99%	99.99%	95.13%

Figure 10: The impact of using approximations of varying rank for full rank training feature matrices. The first row provides the training accuracy and validation of the original model. The first column is the training accuracy of the model on the original dataset, the second column is the training accuracy on the training data reconstructed from the low-rank approximation, and the third column is the validation accuracy. We see that the validation accuracy generally increases when lowering the rank of the approximation. Since only a few singular values of the training feature matrix are large, there is no impact to the training accuracy when using a low-rank approximation for X.

In Figures 11 and 12, we present examples of how LLBoost (which selects the perturbation that improves validation accuracy) improves both validation and test accuracy without impacting training accuracy.

Dataset	Model	Train/Val./Test Acc.. (Original)	Train/Val./Test Acc. (Ours)
2600 Ex. ImageNet32 (80/20 split)	ResNet-18	99.9%/91.7%/93%	99.9%/**92.5%/94%**
CIFAR10 (20/80 split)	ResNet-18	99.99%/95.2%/94.9%	99.99%/**95.3%/94.93%**
CIFAR10 (50/50 split)	ResNet-18	99.99%/95.1%/94.82%	99.99%/**95.16 %/94.84%**
CIFAR10 (90/10 split)	ResNet-18	99.99%/95.01%/94.50%	99.99%/**95.04 %/94.60%**

Figure 11: Using LLBoost to improve validation accuracy also leads to an improvement in test accuracy (i.e. LLBoost does not overfit the validation set). We split the original validation set of CIFAR10 into a validation and test set according to the splits indicated in parentheses. As the validation set of ImageNet32 for 2 classes only has 100 images, we perform an 80/20 train/validation split of the training set, use the 100 validation images as test data, and re-train our models on the smaller training set.

Dataset	Model	Train/Val./Test Acc. (Original)	Train/Val./Test Acc. (Ours)
200 Dogs/Cats CIFAR10	ResNet-18	100%/78%/75.50%	100%/79%/**75.53%**
1000 Dogs/Cats CIFAR10	ResNet-18	100%/84.5%/86.37%	100%/86%/**86.54%**
2000 Dogs/Cats CIFAR10	ResNet-18	100%/88.65%/89.32%	100%/89.15%/**89.42%**

Figure 12: Using LLBoost to improve validation accuracy also leads to an improvement in test accuracy (i.e. LLBoost does not overfit the validation set). In our experiments, we use the same number of examples for training and validation and use the entirety of the remaining examples for testing. For example, in row 1, we use 200 examples for training, 200 for validation and 11600 for testing.

H PROOF OF PROPOSTION 1

Proof. We first consider $\widehat{w} - w^*$:

$$\begin{aligned}
\widehat{w} - w^* &= yV\Sigma^\dagger U^T - w^* \\
&= w^* X V \Sigma^\dagger U^T - w^* \ \text{ (since } y = w^* X) \\
&= w^* U \Sigma \Sigma^\dagger U^T - w^*(U\Sigma^\perp U^T + U{\Sigma^\perp}^\perp U^T) \\
&= w^* U {\Sigma^\perp}^\perp U^T - w^*(U\Sigma^\perp U^T + U{\Sigma^\perp}^\perp U^T) \\
&= -w^* U \Sigma^\perp U^T
\end{aligned}$$

Thus, we have shown (1). Now for (2), we have:

$$\widehat{w}_r - w^* = w^{(0)} U \Sigma^\perp U^T + \widehat{w} - w^* = w^{(0)} U \Sigma^\perp U^T - w^* U \Sigma^\perp U^T = (w^{(0)} - w^*) U \Sigma^\perp U^T.$$

Hence, (2) follows from (1). □

I PROOF OF THEOREM 2

Proof. The proof follows from Lemma 1. Since the columns of X are drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{d\times d})$, Lemma 2 implies that the columns of U are drawn from the uniform distribution on the sphere in $\mathbb{R}^d$. Hence we have that:

$$\mathbb{E}_X[U\Sigma^\perp U^T] = \mathbb{E}_X\left[\sum_{i=n+1}^{d} u_i u_i^T\right] = \sum_{i=n+1}^{d} \mathbb{E}_X[u_i u_i^T] = 1 - \frac{n}{d}.$$

This implies (1) since:

$$\mathbb{E}_X[\|\widehat{w} - w^*\|^2 = w^* \mathbb{E}_X[U\Sigma^\perp U^T] {w^*}^T = \|w^*\|^2 \left(1 - \frac{n}{d}\right).$$

Similarly, we get (2), which completes the proof. □

J PROOF OF PROPOSITION 2

Proof. Let $a^T = w^{(0)} - w^*$. We need to find a such that:

$$
\begin{aligned}
&(1)\ \|a^T U \Sigma^{\perp} U^T\|^2 = \sum_{i=r+1}^{d} |\langle a, u_i \rangle|^2 = c_2^2, \\
&(2)\ a^T a = c_1^2.
\end{aligned}
$$

To do this, we instead first let $a' = c_1 a$ and show that there exists a solution to:

$$
\begin{aligned}
&(1)\ \|a'^T U \Sigma^{\perp} U^T\|^2 = \sum_{i=r+1}^{d} |\langle a', u_i \rangle|^2 = \frac{c_2^2}{c_1^2}, \\
&(2)\ a'^T a' = 1.
\end{aligned}
$$

We will show that there is a solution to the above system by using the intermediate value theorem. First, note that the unit sphere is path connected in $\mathbb{R}^d$. Now for $a' = u_{r+1}$, we have $\|a'\| = 1$ and $\|a'^T U \Sigma^{\perp} U^T\|^2 = 1$. Next, note that for $a' = u_1$, $\|a'\| = 1$ and $|a'^T U \Sigma^{\perp} U^T\|^2 = 0$. Thus, by the intermediate value theorem we conclude that there exists some a' on the unit sphere such that $\|a'^T U \Sigma^{\perp} U^T\|^2 = \frac{c_2^2}{c_1^2}$, which completes the proof. □

K PROOF OF PROPOSITION 3

Proof. Note that we have:

$$
\begin{aligned}
&\mathbb{P}_{w^{(0)}}(\mathbb{E}_{x,X}[(y - \widehat{w}_r x)^2] \leq \mathbb{E}_{x,X}[(y - \widehat{w} x)^2]) \\
\iff &\mathbb{P}_{w^{(0)}}\left(\|w^{(0)} - w^*\|^2 \left(1 - \frac{n}{d}\right) \leq \|w^*\|^2 \left(1 - \frac{n}{d}\right)\right) \\
\iff &\mathbb{P}_{w^{(0)}}\left(\langle w^{(0)}, \frac{w^*}{\|w^*\|}\rangle \geq \frac{1}{2\|w^*\|}\right).
\end{aligned}
$$

Since $w^{(0)}$ and $\frac{w^*}{\|w^*\|}$ are unit vectors on $\mathcal{S}_{d-1}$, the desired probability is equivalent to that of the ratio of the area of the spherical cap (19) defined by the co-latitude angle $\phi = \cos^{-1}\left(\frac{1}{2\|w^*\|}\right)$ to the surface area of $\mathcal{S}_{d-1}$, which completes the proof. □

L PROOF OF THEOREM 3

Proof. We here present the proof for the case that $\gamma = 1$; however, the proof is easily extendable to the case of arbitrary γ. The proof relies on the following inequalities, which are commonly used in analysis.

Proposition 4 (Reduction Formula)**.**

$$
\int \sin^d \theta d\theta = -\frac{1}{d}\cos\theta(\sin\theta)^{d-1} + \frac{d-1}{d}\int \sin^{d-2}\theta d\theta
$$

Proposition 5 (Gautschi's Inequality)**.**

$$
x^{1-s} < \frac{\Gamma(x+1)}{\Gamma(x+s)} < (x+1)^{1-s} \ ;\ s \in (0,1)\ ;\ x > 0
$$

Corollary 1. *For $s \in (0,1)$ and $x > 0$:*

$$
\begin{aligned}
&(1)\ \sqrt{x} < \frac{\Gamma(x+1)}{\Gamma(x+\frac{1}{2})} < \sqrt{x+1}, \\
&(2)\ \frac{1}{\sqrt{x+1}} < \frac{\Gamma(x+\frac{1}{2})}{\Gamma(x+1)} < \frac{1}{\sqrt{x}}.
\end{aligned}
$$

Proposition 6.

$$\sum_{i=1}^{k}\frac{1}{\sqrt{i}}\leq\int_{0}^{k}\frac{1}{\sqrt{x}}dx=2\sqrt{k}$$

Let $K=\int_0^\phi(\sin\theta)^{d-2}d\theta$. We will lower bound this integral. For convenience of notation, we will skip writing the limits of integration. By using the reduction formula for the powers of $\int(\sin\theta)^n d\theta$, and assuming d is even for convenience, we have:

$$\begin{aligned}
K &= -\frac{1}{d-2}\cos\phi(\sin\phi)^{d-3}-\frac{1}{d-2}\frac{d-3}{d-4}\cos\phi(\sin\phi)^{d-5}-\ldots-\frac{(d-3)!!}{(d-2)!!}\cos\phi\sin\phi+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\phi \\
&= -\frac{1}{d-2}\cos\phi\sin\phi\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d-2}{2})}\left[\frac{\sqrt{\pi}\Gamma(\frac{d-2}{2})}{\Gamma(\frac{d-1}{2})}(\sin\phi)^{d-4}+\frac{\sqrt{\pi}\Gamma(\frac{d-4}{2})}{\Gamma(\frac{d-3}{2})}(\sin\phi)^{d-6}+\ldots+\frac{\sqrt{\pi}\Gamma(\frac{2}{2})}{\Gamma(\frac{3}{2})}\right] \\
&\quad+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\phi \\
&\geq -\frac{1}{d-2}\cos\phi\sin\phi\frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d-2}{2})}\left[\sum_{i=1}^{\frac{d-4}{2}}\frac{(\sin^2\phi)^i}{\sqrt{\frac{2i+1}{2}}}+1\right]+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\phi \text{ (by Gautschi's Inequality)} \\
&\geq -\frac{1}{d-2}\cos\phi\sin\phi\frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d-2}{2})}\left[\sum_{i=1}^{\frac{d-4}{2}}\frac{(\sin^2\phi)^i}{\sqrt{\frac{2i}{2}}}+1\right]+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\phi \\
&= -\frac{1}{d-2}\cos\phi\sin\phi\frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d-2}{2})}\left[\sum_{i=1}^{\frac{d-4}{2}}\frac{1}{\sqrt{i}}+1\right]+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\phi \\
&\geq -\frac{1}{d-2}\cos\phi\sin\phi\frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d-2}{2})}\left[2\sqrt{\frac{d-4}{2}}+1\right]+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\phi.
\end{aligned}$$

Since $\phi=\cos^{-1}\left(\frac{1+\epsilon}{2\|w^*\|}\right)$, then

$$K\geq-\frac{1}{(d-2)}\frac{1+\epsilon}{2\|w^*\|}\sqrt{1-\frac{(1+\epsilon)^2}{4\|w^*\|^2}}\frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d-2}{2})}\left[2\sqrt{\frac{d-4}{2}}+1\right]+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\cos^{-1}\left(\frac{1+\epsilon}{2\|w^*\|}\right).$$

Again by Gautschi's Inequality we obtain:

$$\frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d-2}{2})}<\sqrt{\frac{d-1}{2}},$$

and hence,

$$K>-\frac{1}{(d-2)}\frac{1+\epsilon}{2\|w^*\|}\sqrt{1-\frac{(1+\epsilon)^2}{4\|w^*\|^2}}\sqrt{\frac{d-1}{2}}\left[2\sqrt{\frac{d-4}{2}}+1\right]+\frac{\Gamma(\frac{d-1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2})}\cos^{-1}\left(\frac{1+\epsilon}{2\|w^*\|}\right).$$

Thus, we have that:

$$\begin{aligned}
\frac{\Gamma\left(\frac{d}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{d-1}{2}\right)}\int_0^\phi\sin^{d-2}\theta d\theta > -\sqrt{\frac{d}{2\pi}}\frac{1}{(d-2)}\frac{1+\epsilon}{2\|w^*\|}\sqrt{1-\frac{(1+\epsilon)^2}{4\|w^*\|^2}}\sqrt{\frac{d-1}{2}}\left[2\sqrt{\frac{d-4}{2}}+1\right] \\
+\frac{1}{\pi}\cos^{-1}\left(\frac{1+\epsilon}{2\|w^*\|}\right).
\end{aligned}$$

Hence, assuming $\|w^*\|=\frac{\sqrt{d}}{c}$, we obtain:

$$\lim_{d\to\infty}\frac{\Gamma\left(\frac{d}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{d-1}{2}\right)}\int_0^\phi\sin^{d-2}\theta d\theta\geq-\frac{c(1+\epsilon)}{2\sqrt{2\pi}}+\frac{1}{\pi}\frac{\pi}{2}=\frac{1}{2}-\frac{c(1+\epsilon)}{2\sqrt{2\pi}}.$$

Note that we have:

$$\mathbb{P}_{w^{(0)}}\left(\langle w^{(0)}, \frac{w^*}{\|w^*\|}\rangle \geq \frac{(1+\epsilon)}{2\|w^*\|}\right) \leq \mathbb{P}_{w^{(0)}}\left(\langle w^{(0)}, \frac{w^*}{\|w^*\|}\rangle \geq 0\right) = \frac{1}{2},$$

and hence, we conclude that:

$$\frac{1}{2} - \frac{c(1+\epsilon)}{2\sqrt{2\pi}} \leq \lim_{d\to\infty} \frac{\Gamma\left(\frac{d}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{d-1}{2}\right)} \int_0^{\phi} \sin^{d-2}\theta d\theta \leq \frac{1}{2}.$$

□