Studying Data Complexity and Learned Structure in Neural Networks with Bayesian Probes

Maxwell Adam[=] MAX@V3RV.COM
The University of Melbourne

Zach Furman[=] ZACH.FURMAN1@GMAIL.COM
The University of Melbourne

Wilson Wu WILSON.WU@COLORADO.EDU
University of Colorado Boulder

Philipp Alexander Kreer PHILIPP.A.KREER@OUTLOOK.DE
Technical University of Munich
Timaeus

Jesse Hoogland JESSE@TIMAEUS.CO
Timaeus

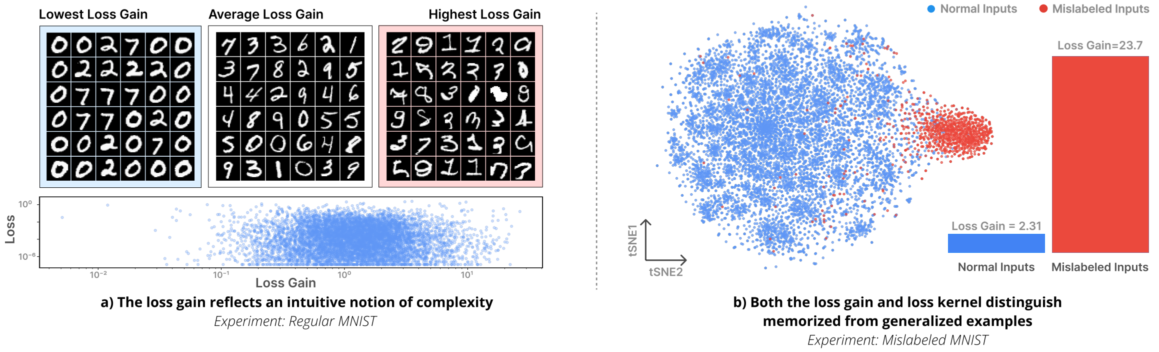
Abstract

We study a lightweight Bayesian probe for analyzing neural networks trained with standard optimization methods (e.g. SGD). Starting from trained parameters, we run stochastic-gradient Markov chain Monte Carlo (SGMCMC) to explore the local posterior, analyzing the per-sample losses as random quantities. The posterior mean of the per-sample loss change defines the *posterior loss gain*, a practical measure of sample difficulty. High loss gain values indicate difficult, atypical, or memorized samples, while lower values indicate easier, typical examples. The posterior covariance between sample losses defines the *posterior loss covariance kernel*, reflecting shared structure learned by the network. Experiments on MNIST show that the posterior loss gain effectively separates easy digits from hard or mislabeled ones. On ImageNet, initial explorations with the posterior loss covariance kernel show examples of correlated images that suggest semantically coherent groupings and potential cross-class relationships. Together, the posterior loss gain and loss kernel offer a simple, post-training toolkit for investigating sample difficulty and semantic structure in deep neural networks.

1. Introduction

Deep neural networks are trained on diverse datasets where individual samples can vary significantly in complexity, typicality, and their influence on the learning process. While models achieve high aggregate performance, they often obscure how specific data points shape the final solution or how sensitive the model is to perturbations concerning individual examples. Uncovering these sample-level dynamics and inter-dependencies is crucial for tasks ranging from data debugging and curriculum design to improving model robustness and interpretability [15].

This paper introduces a framework for extracting fine-grained, sample-level structure from any SGD-trained neural network using **Bayesian probes**. We leverage a local posterior (Eq. (1)) centered around the SGD solution, treating per-sample losses as random variables under this posterior. By analyzing the first two cumulants of these loss fluctuations, we define:



a) The loss gain reflects an intuitive notion of complexity
Experiment: Regular MNIST

b) Both the loss gain and loss kernel distinguish memorized from generalized examples
Experiment: Mislabeled MNIST

Figure 1: The loss kernel and loss gain extract interpretable input-level patterns from the posterior geometry. Left: The lowest, highest, and average loss gain training inputs for a MLP trained on MNIST. Low-loss-gain examples (left) are simple and prototypical; high-loss-gain examples (right) are ambiguous or atypical. Below is a scatter plot of loss versus loss gain (both log-scaled) for a random subset of training examples: the loss gain reveals structure not apparent from loss alone. Right: Mislabeled inputs have ∼**10x** higher loss gain on average, and cluster together when applying tSNE to the loss covariance kernel feature space.

1. The **posterior loss gain** (henceforth, **loss gain**) (Eq. (4)): The expected loss increase for a sample, quantifying its learning difficulty or fragility.
2. The **posterior loss covariance kernel** (henceforth, **loss kernel**) (Eq. (5)): A kernel over data points investigating shared model features and pairwise dependencies by measuring how their losses co-vary under posterior perturbations.

Our contributions are twofold: (i) We provide a principled framework where the loss gain and loss kernel refine complexity measures from singular learning theory to the per-sample case, and where the loss kernel generalizes influence functions for singular models. (ii) We demonstrate empirically on MNIST that loss gain captures intuitive notions of data complexity and that both the loss gain and loss kernel can distinguish memorized from generalized examples. We then present illustrative examples in the ImageNet setting suggesting that the posterior loss kernel can highlight intuitively meaningful semantic relationships between inputs, based on their patterns of co-varying loss under posterior perturbations. These probes offer a computationally tractable lens for exploring the learned behavior of networks at a granular level.

2. Related Work

Our approach, the cumulant expansion (Eq. (3)) of per-sample loss changes under a local posterior, connects complexity measures from **Singular Learning Theory (SLT)** [23, 24] to influence functions from robust statistics. This expansion can be seen as a multivariate version of the expansion of the Bayesian generalization error as studied in SLT; our first two cumulants (loss gain and loss kernel) can be seen as per-sample, vector-valued generalizations of SLT's Local Learning Coefficient (LLC; Lau et al. 14) and singular fluctuation, respectively. (See Appendix B.)

. $^{=}$Equal contribution

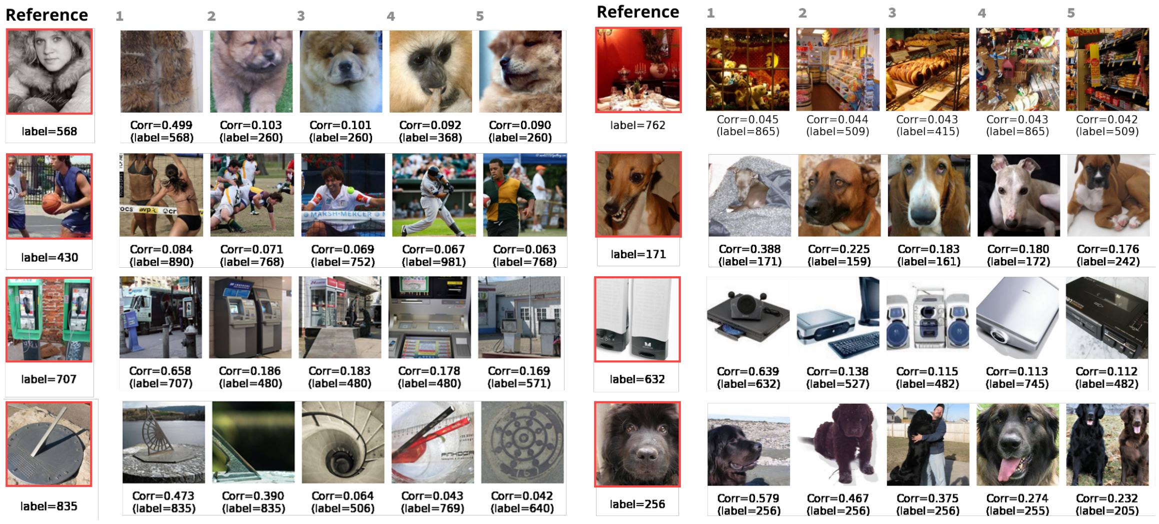


Figure 2: Top-correlated examples under the posterior loss kernel reveal interpretable patterns. For each reference image (leftmost column), we show the top 5 most-correlated inputs under the posterior loss correlation kernel R. We observe clustering by texture (e.g., fluffy fur coat and fluffy animals), shape (e.g., circular objects and line angle), color and category (e.g., people playing sports, electronics on a white background, dark vs. light brown dogs), and spatial layout (e.g., cluttered rooms). Additional visualizations are provided in Appendix D, and all computed correlation results are available at `https://github.com/singfluence-anon/sf_imagenet_corrs`.

The first cumulant, our **posterior loss gain** μ_i, measures sample-specific fragility. The loss gain is closely related to the *data-refined* LLC of Wang et al. [22], and the sum of all loss gains is proportional to the overall LLC, strengthening the loss gain's interpretation as a data complexity measure. (See Appendix B.1.)

The second cumulant, the **posterior loss covariance kernel** K_{ij}, reveals pairwise sample interactions. It generalizes classical influence functions [6, 13], and is well-defined for singular deep learning models. The loss kernel coincides with a special case of the local Bayesian Influence Function [1] and reduces to the influence Gram matrix $g_i^\top H^{-1} g_j$ in non-singular settings [8, 12] (derivation reproduced in Appendix C). It is closely related to Baker et al. [2]'s *local susceptibilities*. This covariance kernel and its properties have been studied in prior work [12, 16, 19], but our work applies it to interpreting SGD-trained deep learning models.

3. Method

This section formally defines the two statistics derived from our Bayesian probe: the *posterior loss gain* and the *posterior loss covariance kernel*. We begin by establishing the local posterior and the per-sample loss fluctuations, then derive the loss gain and loss kernel from the first two cumulants of these fluctuations.

3.1. Local Posterior and Per-Sample Loss Fluctuations

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be the training data, $f_w : \mathcal{X} \to \mathcal{Y}$ a neural network parameterized by weights $w \in \mathbb{R}^d$, and w^* the parameters obtained by SGD. The per-sample loss for sample i is $\ell_i(w) := \ell\big(f_w(x_i), y_i\big)$.

To analyze the model's sensitivity around w^*, we introduce a **local posterior** as a diagnostic (following Lau et al. [14]):

$$p(w \mid \mathcal{D}) \;\propto\; \exp\Big[-\beta \sum_{j=1}^{n} \ell_j(w)\Big]\, \mathcal{N}\big(w^*,\, \gamma^{-1} I\big), \qquad \beta > 0,\ \gamma > 0. \tag{1}$$

This posterior allows us to study the effects of small, data-informed weight perturbations. The exponential term encourages fitting the data, while the Gaussian prior $\mathcal{N}(w^*, \gamma I)$ localizes exploration around w^*. The inverse temperature β adjusts the probe's sensitivity.

For a weight draw $w \sim p(w \mid \mathcal{D})$, we define the **loss change** for sample i as $\Delta\ell_i(w) := \ell_i(w) - \ell_i(w^*)$. These are collected into the **loss-change vector**:

$$\mathbf{L}(w) \;=\; \big(\Delta\ell_1(w), \ldots, \Delta\ell_n(w)\big)^{\top} \in \mathbb{R}^n, \tag{2}$$

which is a random vector describing how each sample's loss changes under perturbations from the local posterior.

3.2. Loss Gain and Loss Kernel from Cumulants

The statistics of $\mathbf{L}(w)$ are derived from its cumulant generating function (CGF):

$$\Phi(\mathbf{t}) := \log \mathbb{E}_w\left[e^{\mathbf{t}^{\top}\mathbf{L}(w)}\right].$$

Its expansion for small $\mathbf{t}$ is:

$$\Phi(\mathbf{t}) \;=\; \sum_{i=1}^{n} \mu_i\, t_i \;+\; \sum_{i=1}^{n}\sum_{j=1}^{n} \tfrac{1}{2} K_{ij}\, t_i t_j \;+\; \mathcal{O}\big(\|\mathbf{t}\|^3\big). \tag{3}$$

The first two cumulants yield our primary statistics:

Posterior loss gain. The first cumulant, μ_i, defines the loss gain for sample x_i:

$$\mu_i \;=\; \mathbb{E}_w\big[\Delta\ell_i(w)\big]. \tag{4}$$

Here, μ_i represents the expected change in sample i's loss under local posterior perturbations, quantifying its fragility or learning difficulty. As shown in Appendix B.1, the loss gain is closely related to the Local Learning Coefficient (LLC).

Posterior loss covariance kernel. The second cumulant, K_{ij}, defines the kernel:

$$K_{ij} \;:=\; \mathrm{Cov}_w\big[\Delta\ell_i(w),\ \Delta\ell_j(w)\big]. \tag{5}$$

This positive semidefinite kernel K captures pairwise interactions. $K_{ij} > 0$ suggests that perturbations affecting x_i similarly affect x_j, indicating shared dependencies. As shown in Appendix C,

K_{ij} generalizes classical influence functions to singular settings. For analysis and visualization, we often use its normalized form, the *correlation kernel* R, which measures the correlation of loss changes:

$$R_{ij} := \frac{K_{ij}}{\sqrt{K_{ii}\,K_{jj}}}, \qquad -1 \leq R_{ij} \leq 1, \tag{6}$$

with the convention $R_{ij} = 0$ if $K_{ii} = 0$ or $K_{jj} = 0$. From a feature-map perspective, K_{ij} can be viewed as an inner product $\langle \varphi_i, \varphi_j \rangle_{L^2(p)}$ where $\varphi_i(w) := \Delta\ell_i(w) - \mu_i$, the *loss trace* of the i-th sample.

3.3. Practical Estimation via SGLD

Expectations over $p(w \mid \mathcal{D})$ are approximated using S samples $\{w_s\}_{s=1}^{S}$ from a Stochastic Gradient Langevin Dynamics (SGLD) [26] chain (or multiple chains) initialized at w^*. Standard unbiased plug-in estimators are used for $\hat{\mu}_i$ and $\hat{K}_{ij}$, from which $\hat{R}_{ij}$ is computed. For details on SGLD and its hyperparameters, see Appendix A.1

4. Results

Posterior diagnostics appear to reveal meaningful sample structure. We evaluate the loss gain and loss kernel across both controlled and large-scale settings. For an MNIST MLP, we compute the loss gain for every training example and find that it correlates well with intuitive notions of complexity (Fig. 1). Examples with low loss gain values tend to be simple and prototypical digits, while high loss gain examples are ambiguous or atypical. This suggests that the loss gain captures how "fragile" or "precise" the configuration of parameters which the model uses on each individual input is. We also show that the loss gain does not simply reflect the loss of the trained model: the Pearson correlation between loss and loss gain is $r = -0.0302$, indicating that loss gain values are not predictive of the loss.

The loss gain separates generalization from memorization. To test whether the loss gain distinguishes robust generalization from rote memorization, we introduce label noise into the MNIST dataset by randomly relabeling 10% of the training examples. We train the model until it has high accuracy on mislabeled inputs: After training, we observe that mislabeled examples have a much higher average loss gain than clean ones (10x higher). Moreover, when applying kernel dimensionality reduction to the loss covariance kernel (here using tSNE), the mislabeled inputs form a distinct cluster, isolated from the main body of clean examples.

The loss kernel appears to reflect semantic structure in ImageNet. We next apply our method to a pretrained InceptionV1 model on ImageNet. For 2,500 random validation examples, we compute the posterior loss correlation matrix and examine top correlated inputs. Preliminarily, we find that nearest neighbors are *highly* interpretable and that they often capture nuanced connections between images. We find consistent patterns of color, texture, shape, and content across the vast majority of samples. Also, the number of samples for which we computed loss traces was relatively small: we expect that increasing the number of samples will improve the overall quality of top-correlating inputs significantly.

See Appendix A for dataset and model details, and Appendix D for extended visualizations.

5. Conclusion

We introduced Bayesian probes — the posterior loss gain and the posterior loss covariance kernel — derived from a local posterior to analyze per-sample loss statistics in SGD-trained networks. These principled tools appear to quantify sample fragility (loss gain) and reveal shared model features (loss kernel), potentially offering utility in uncovering intuitive complexity and semantic relationships, especially in singular models like neural networks. Future work includes solidifying these existing findings, as well as exploring higher-order cumulants, the behavior of these metrics over training time, and broader applications in anomaly detection, data understanding, and interpretability.

Acknowledgments

PAK was supported by the European Research Council (ERC) under the European Union's research and innovation programme grant agreements ERC Starting Grant 949279 HighPHun and the Pivotal Fellowship.

References

[1] Anonymous et al. Bayesian influence functions for scalable data attribution, 2025. Concurrently submitted workshop paper, ICML 2025.

[2] Garrett Baker, George Wang, Jesse Hoogland, and Daniel Murfet. Studying Small Language Models with Susceptibilities, April 2025. URL `http://arxiv.org/abs/2504.18274`. arXiv:2504.18274 [cs].

[3] Liam Carroll, Jesse Hoogland, Matthew Farrugia-Roberts, and Daniel Murfet. Dynamics of Transient Structure in In-Context Linear Regression Transformers, January 2025. URL `http://arxiv.org/abs/2501.17745`. arXiv:2501.17745 [cs].

[4] Zhongtian Chen and Daniel Murfet. Modes of sequence models and learning coefficients. *arXiv preprint arXiv:2504.18048*, 2025.

[5] Zhongtian Chen, Edmund Lau, Jake Mendel, Susan Wei, and Daniel Murfet. Dynamical versus bayesian phase transitions in a toy model of superposition. *arXiv preprint arXiv:2310.06301*, 2023.

[6] R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[8] Ryan Giordano and Tamara Broderick. The Bayesian Infinitesimal Jackknife for Variance, June 2024. URL `http://arxiv.org/abs/2305.06466`. arXiv:2305.06466 [stat].

[9] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman.

Studying Large Language Model Generalization with Influence Functions, August 2023. URL http://arxiv.org/abs/2308.03296. arXiv:2308.03296 [cs].

[10] Rohan Hitchcock, Gary W Delaney, Jonathan H Manton, Richard Scalzo, and Jingge Zhu. Emergence of computational structure in a neural network physics simulator. *arXiv preprint arXiv:2504.11830*, 2025.

[11] Jesse Hoogland. Neural networks generalize because of this one weird trick, 01 2023. URL https://www.lesswrong.com/posts/fovfuFdpuEwQzJu2w/neural-networks-generalize-because-of-this-one-weird-trick.

[12] Yukito Iba. W-kernel and essential subspace for frequentist evaluation of Bayesian estimators, June 2024. URL http://arxiv.org/abs/2311.13017. arXiv:2311.13017 [stat].

[13] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[14] Edmund Lau, Zach Furman, George Wang, Daniel Murfet, and Susan Wei. The local learning coefficient: a singularity-aware complexity measure. In *The 28th international conference on artificial intelligence and statistics*, 2025. URL https://openreview.net/forum?id=1av51Z1suL.

[15] Simon Pepin Lehalleur, Jesse Hoogland, Matthew Farrugia-Roberts, Susan Wei, Alexander Gietelink Oldenziel, George Wang, Liam Carroll, and Daniel Murfet. You are what you eat–AI alignment requires understanding how data shapes structure and generalisation. *arXiv preprint arXiv:2502.05475*, 2025.

[16] Steven N MacEachern and Mario Peruggia. Bayesian tools for EDA and model building: A brainy study. In *Case Studies in Bayesian Statistics: Volume V*, pages 345–362. Springer, 2002.

[17] Daniel Murfet and Will Troiani. Programs as singularities. *arXiv preprint arXiv:2504.08075*, 2025.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. URL https://arxiv.org/abs/1409.4842.

[19] Zachary M Thomas, Steven N MacEachern, and Mario Peruggia. Reconciling curvature and importance sampling based procedures for summarizing case influence in bayesian models. *Journal of the American Statistical Association*, 113(524):1669–1683, 2018.

[20] Einar Urdshals and Jasmina Urdshals. Structure development in list-sorting transformers. *arXiv preprint arXiv:2501.18666*, 2025.

[21] George Wang, Matthew Farrugia-Roberts, Jesse Hoogland, Liam Carroll, Susan Wei, and Daniel Murfet. Loss landscape geometry reveals stagewise development of transformers. June 2024. URL https://openreview.net/forum?id=2JabyZjM5H&referrer=

%5Bthe%20profile%20of%20Jesse%20Hoogland%5D(%2Fprofile%3Fid%3D~Jesse_Hoogland1).

[22] George Wang, Jesse Hoogland, Stan van Wingerden, Zach Furman, and Daniel Murfet. Differentiation and Specialization of Attention Heads via the Refined Local Learning Coefficient. 2025. URL https://openreview.net/forum?id=SUc1UOWndp&noteId=MCoFYhi7ZE.

[23] Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2009.

[24] Sumio Watanabe. *Mathematical theory of Bayesian statistics*. Chapman and Hall, 2018.

[25] Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella. Deep learning is singular, and that's good. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10473–10486, 2023. doi: 10.1109/TNNLS.2022.3167409.

[26] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Appendix A. Further Experimental Details

A.1. Stochastic-Gradient MCMC Estimator

Evaluating the *posterior loss covariance kernel* $K_{ij} = \mathrm{Cov}_w[\Delta\ell_i(w), \Delta\ell_j(w)]$ and the *loss gain* $\mu_i = \mathbb{E}_w[\Delta\ell_i(w)]$ requires Monte-Carlo samples from the local posterior $p(w \mid \mathcal{D})$ in Eq. (1). Following Lau et al. [14], we use Stochastic Gradient Langevin Dynamics (SGLD; Welling and Teh 26).

Update rule. With stochastic mini-batch $B_t \subset [n]$ of size m and step size ϵ, SGLD performs

$$w_{t+1} \;=\; w_t - \frac{\epsilon}{2}\Big(\frac{n}{m}\sum_{k\in B_t}\nabla_w \ell_k(w_t) + \gamma\big(w_t - w^*\big)\Big) + \sqrt{\epsilon}\,\xi_t, \qquad \xi_t \sim \mathcal{N}(0, I). \tag{7}$$

The first term is the stochastic gradient of the log-likelihood; the second is the gradient of the Gaussian localization potential $\frac{\gamma}{2}\|w - w^*\|^2$; the injected Gaussian noise ensures asymptotic convergence to $p(w \mid \mathcal{D})$ as $\epsilon \to 0$.

Parallel chains and burn-in. To improve mixing we run C independent chains, each initialized at w^*. After discarding a burn-in of b iterations, we retain T draws $\{w_{c,t}\}_{t=1}^{T}$ per chain. For every retained weight we record the vectors $\Delta\ell(w_{c,t})$.

Cumulant estimators. The unbiased plug-in estimators are

$$\begin{aligned}
\hat{\mu}_i &= \frac{1}{CT}\sum_{c,t}\Delta\ell_i(w_{c,t}),\\
\hat{K}_{ij} &= \frac{1}{CT-1}\sum_{c=1}^{C}\sum_{t=1}^{T}\big(\Delta\ell_i(w_{c,t}) - \hat{\mu}_i\big)\big(\Delta\ell_j(w_{c,t}) - \hat{\mu}_j\big)\\
\hat{R}_{ij} &= \hat{K}_{ij}/\sqrt{\hat{K}_{ii}\hat{K}_{jj}}.
\end{aligned}$$

A.2. Correlation Kernel Hyperparameters

Table 1 summarizes the hyperparameter settings for the correlation kernel experiments. We sample with SGLD: m is the batch size, C is the number of chains, T the number of draws per chain, b is the number of burn-in steps, ϵ is the learning rate, β is the inverse temperature, and γ is the localization strength. For more information on the hyperparameters refer to Appendix A.1. We use the same loss traces to compute both the loss gain and loss kernel (the hyperparameters are the same).

Table 1: Summary of hyperparameter settings for correlation kernel experiments. Hyperparameters are defined in Appendix A.1 and Section 3.1.

Section	Model	Dataset	m	C	T	b	ϵ	$n\beta$	γ
Section 4	2 Layer MLP	MNIST Train Set	64	5	500	10	5×10^{-3}	10	30
Section 4	2 Layer MLP	Mislabeled MNIST Train Set	64	1	500	100	5×10^{-3}	10	30
Section 4	InceptionV1	ImageNet	64	30	1000	100	1×10^{-4}	10	1000

A.3. MNIST

We train a two-layer ReLU MLP (width 512) on the MNIST dataset and compute loss gain values for the full train set (60,000 examples). In Figure 1, we plot representative samples of low, average, and high loss gain inputs and plot loss vs loss gain for 10,000 training examples. Low loss gain inputs are neat and formulaic while high loss gain inputs are convoluted outliers.

A.4. Detecting Memorized MNIST Inputs

We next test whether the posterior loss gain distinguishes genuine generalization from rote memorization.

To force memorization, we create a noisy training set by randomly relabeling $10\,\%$ of the MNIST train set (6,000 of 60,000 total). We reuse the architecture of Section A.3 and train for 100 full-dataset passes. This creates two distinct populations within our training set: normal examples that can be classified using generalizable features and mislabeled examples that must be individually memorized. After training, the network attains $99\,\%$ accuracy on the clean (training) portion of the data and $98\,\%$ on the mislabeled subset. We then compute the loss kernel and loss gain for every training example. Fig. 1 shows covariance kernel tSNE plots for 10,000 inputs at epoch 100. We also plot the average loss gain for the mislabeled and normal inputs at epoch 100. On average, mislabeled inputs have a roughly 10x higher loss gain than normal inputs (23. vs 2.31, respectively). Runtime for this experiment on a standard Macbook M3 is less than 15 minutes.

A.5. InceptionV1

We apply our method to InceptionV1 [18], evaluating posterior correlations over 2,500 ImageNet validation samples. To reduce memory overhead, we downscale all images to 256x256 resolution. Full hyperparameters are included in 1. We find that the quality of correlations depends significantly on total draws used - we take a total of 30,000 draws across 30 chains. We sample over the full ImageNet [7] validation dataset. Total runtime was 3 hours on 4 A100 GPUs. We include more examples in Appendix D.

It is almost always the case that the first-n top correlated inputs will share a label with the reference image; we would argue, however, that this is much less intriguing than correlations between images that don't share a label, as one can trivially recover groupings by label simply by randomly perturbing the logits enough times. What's more interesting are cross-label correlations; we argue that because these aren't confounded by logit noise they are a more interesting measure.

Figure 2 displays a few interesting examples of such correlations. In the top left, the reference image is a woman wearing a poofy, brown fur coat; The top correlated image is another poofy coat of the same label, but the following are brown Chow Chow dogs and fluffy monkeys. The middle-left example shows a reference image of a basketball player, with the most correlated images including other sports scene action shots, despite differing class label. The bottom left reference image is a sundial - it's top correlated inputs are firstly two sundials, followed by images which all have straight line *and* circular features. Strikingly, the angles of the red pen and black pens in the styrofoam cup in the 4th most correlated input nearly align with the angle of the sundial and shadow in the reference image. Similarly intriguing patterns are displayed in the other examples. Appendix D contains many more examples randomly selected from our results and contains a link to a github repository containing the top-16 correlated images for every input we computed the loss kernel for in this experiment.

Appendix B. Singular Learning Theory

In the theory of statistical learning, we are given a prior distribution over parameters $\varphi(w)$, a data set $\mathcal{D}_n = \{x_1, \ldots, x_n\}$, and a true underlying distribution $q(x)$. Furthermore, let $p(x \mid w)$ describe a model with input x parametrized by weights $w \in \mathcal{W} \subset \mathbb{R}^d$. According to Bayes' rule, the posterior distribution over the parameters is given by

$$p(w \mid \mathcal{D}_n) = \frac{p(\mathcal{D}_n \mid w)\varphi(w)}{p(\mathcal{D}_n)}. \tag{8}$$

The classical results of statistical learning theory hold when the model $p(x \mid w)$ is *regular*, which, roughly speaking, means that there are no directions in weight space that leave the model's behavior invariant. More precisely, a model is regular if the Fisher information matrix

$$I_{jk}(w) := \int \left(\frac{\partial}{\partial w_j} \log p(x \mid w) \right) \left(\frac{\partial}{\partial w_k} \log p(x \mid w) \right) p(x \mid w)\, dx$$

is everywhere (in parameter space) strictly positive definite. Modern deep learning models fail to be regular [25]; such models are called *strictly singular*. Singular learning theory is the learning theory of such models.

The posterior in Eq. (8) reads [11, 23]

$$p(w|\mathcal{D}_n) = \frac{1}{Z_n^0} \varphi(w) e^{-n\beta K_n(w)}, \tag{9}$$

where $\beta > 0$ is the inverse temperature, Z_n^0 is the partition function (ensuring that $\int p(w \mid \mathcal{D}_n)\, dw = 1$), and $K_n(w)$ is the Kullback-Leibler divergence

$$K_n(w) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{q(x_i)}{p(x_i|w)}. \tag{10}$$

Let $\mathbb{E}_w$ be the expectation with respect to $p(w \mid \mathcal{D}_n)$. The Gibbs training error is then given by

$$G_n = \mathbb{E}_w[K_n(w)] \tag{11}$$

The Gibbs training error is a central quantity in statistical learning theory. If it is small, it means that the model approximates the true underlying distribution well. It can be shown for singular models that

$$\mathbb{E}_{\mathcal{D}}[G_n] = \frac{\lambda - \nu\beta}{\beta n} + o\left(\frac{1}{n}\right) \tag{12}$$

where the expectation is over possible datasets $\mathcal{D}$ sampled IID from the true distribution [23]. Here, λ is the *learning coefficient* and ν is the *singular fluctuation*. Both quantities are fundamental observables in singular learning theory.

The loss gain and loss kernel can be seen as per-sample generalizations of the (local)[1] learning coefficient and (local) singular fluctuation, as explored in Appendix B.1 and Appendix B.2.

[1] We only need these "(local)" parentheticals because we defined the loss gain and loss kernel using *local* posterior; if we had defined them using the ordinary posterior the "local" qualifiers would not apply.

B.1. Connection between Local Learning Coefficient (LLC) and Posterior Loss Gain

Extending Watanabe [23]'s *learning coefficient* λ, Lau et al. [14] defines the *local learning coefficient (LLC)* $\lambda(w^*)$ as a measure of the complexity of a model near a parameter w^*. They define a posterior estimator for the LLC as:

$$\hat{\lambda}(w^*) = n\beta\, \mathbb{E}_w \left[\sum_{i=1}^{n} \Delta \ell_i \right]$$

This may be rewritten as a sum of loss gains μ_i times $n\beta$:

$$\hat{\lambda}(w^*) = n\beta \sum_{i=1}^{n} \mathbb{E}_w \left[\Delta \ell_i \right] = n\beta \sum_{i=1}^{n} \mu_i$$

In other words, the estimated LLC is equal to the sum of the loss gains across all samples, up to a constant factor. Given the significant theoretical and empirical work supporting the LLC as a complexity measure [3–5, 10, 14, 17, 20–22], this serves to support the interpretation of the loss gain as a complexity measure by association.

B.2. Connection between Singular Fluctuation and Posterior Loss Covariance Kernel

The (local) *empirical variance* is defined as the sum of (local) posterior per-sample loss variances [23]:

$$V = \sum_{i=1}^{n} \mathrm{Var}_w(\ell_i(w)). \tag{13}$$

This is clearly the trace of the posterior loss covariance kernel, as Iba [12] first noticed:

$$V = \sum_{i=1}^{n} K_{ii}.$$

The (local) singular fluctuation ν, see Eq. (12), is related to the (local) empirical variance by:

$$\nu = \frac{\beta}{2} \lim_{n \to \infty} \mathbb{E}_{\mathcal{D}}[V]\,, \tag{14}$$

where the expectation is over different possible datasets $\mathcal{D}$.

As we have assumed i.i.d. data, we can drop the expectation over datasets and estimate the singular fluctuation directly from the empirical variance. The convergence of this relation scales quadratically in n.

Appendix C. Relating Posterior Loss Covariance Kernel and Classical Influence Functions

A companion paper [1] extends “Bayesian Influence Functions” [8, 12] to the local setting:

$$\mathrm{BIF}_\gamma(z_i, \phi) = \mathrm{Cov}_{w,\gamma}\big[\ell_i(w),\ \phi(w)\big], \tag{15}$$

where the covariance is taken over the local posterior as in Eq. (1). The loss kernel (see Eq. (5)) coincides with a special case of this local BIF, when $\phi(w) = \ell_j(w)$. This follows from the fact that the covariance is invariant to a constant translation:

$$\mathrm{Cov}_w\big[\ell_i(w),\ \ell_j(w)\big] = \mathrm{Cov}_w\big[\Delta\ell_i(w),\ \Delta\ell_j(w)\big]\ =:\ K_{ij}. \tag{16}$$

Below we reproduce Appendix B from the companion work [1], which details the relationship between the local BIF and classical influence functions (IFs). This establishes that, for regular models, the classic IF is the leading-order term in the Taylor expansion of the posterior loss covariance kernel.

Let w^* be a model checkpoint. In this section, all gradients and Hessians are evaluated at w^*; thus, to reduce notational clutter, we omit the dependence on w. For any function $f(w)$, we denote its gradient at w^* as $g_f = \nabla_w f(w^*)$ and its Hessian as $H_f = \nabla_w^2 f(w^*)$. In particular, $g_i = \nabla_w \ell_i(w^*)$ and $H_i = \nabla_w^2 \ell_i(w^*)$ for a per-sample loss $\ell_i(w)$. The total Hessian of the empirical risk $L_{\text{train}}(w) = \sum_{k=1}^n \ell_k(w)$ at w^* is denoted $H = \sum_{k=1}^n H_k$.

The posterior loss covariance kernel is given by (see Eq. (5)):

$$K_{ij}\ :=\ \mathrm{Cov}_w\big[\Delta\ell_i(w),\ \Delta\ell_j(w)\big] = \mathrm{Cov}_w\big[\ell_i(w),\ \ell_j(w)\big], \tag{17}$$

where the covariance is taken over $w \sim p(w \mid \mathcal{D})$.

To understand the components of this covariance and its relation to classical IFs, we consider an idealized scenario where the model is **regular**. Under this strong assumption, which *does not hold for deep neural networks* [25], the posterior $p(w \mid \mathcal{D})$ can be approximated by a Laplace approximation around w^*:

$$p(w \mid \mathcal{D}) \approx p^{\text{Lap}}(w|\mathcal{D}) = \mathcal{N}(w^*, H^{-1}). \tag{18}$$

The Bernstein–von Mises theorem states that, due to the model's regularity, the true posterior distribution converges in total variation distance to the Laplace approximation as the training dataset size n approaches infinity.

Let $\Delta w = w - w^*$. Assuming analyticity, we can express $\ell_i(w)$ using its full Taylor series expansions around w^*:

$$\ell_i(w) = \ell_i(w^*) + g_i^\top \Delta w + \frac{1}{2}\Delta w^\top H_i \Delta w + \sum_{k=3}^{\infty} \frac{1}{k!} D^k \ell_i(w^*)\,\underbrace{[\Delta w, \ldots, \Delta w]}_{k-\text{times}}, \tag{19}$$

where $D^k f(w^*)[\Delta w, \ldots, \Delta w]$ denotes the k-th order differential of f at w^* applied to k copies of Δw.

The covariance under the Laplace approximation p^{Lap} then involves covariances between all pairs of terms from these two expansions:

$$\mathrm{Cov}_{p^{\text{Lap}}}(\ell_i(w), \ell_j(w)) = \sum_{k=1}^{\infty}\sum_{m=1}^{\infty} \mathrm{Cov}_{p^{\text{Lap}}}\left(\text{Term}_k[\ell_i(w)], \text{Term}_m[\ell_j(w)]\right), \tag{20}$$

where $\text{Term}_k[\ell_i(w)]$ is the k-th order term in the Taylor expansion of Eq. (19). In the Laplace approximation, $\Delta w \sim \mathcal{N}(0, H^{-1})$, the leading linear term ($k = 1, m = 1$) is:

$$\mathrm{Cov}_{p^{\text{Lap}}}(g_i^\top \Delta w, g_j^\top \Delta w) = g_i^\top H_{w^*}^{-1} g_j.$$

Thus, K_{ij} under these regularity and Laplace approximations becomes:

$$K_{ij} \approx -g_i^\top H^{-1} g_j + \text{Higher-order corrections.} \tag{21}$$

The leading term $-g_i^\top H^{-1} g_j = -\nabla_w \ell_i(w^*)^\top H_{w^*}^{-1} \nabla_w \ell_j(w^*)$ is precisely the definition of a classical influence function $\mathrm{IF}(\boldsymbol{z}_i, \phi)$ where the observable ϕ is chosen to be the loss ℓ_i [9]. The posterior loss covariance kernel formulation, even when analyzed via Laplace approximation, naturally includes this term and also explicitly shows a second-order correction involving products of the Hessians of the loss and observable. More generally, the exact posterior loss covariance kernel definition (Eq. (5)) encapsulates all higher-order dependencies without truncation.

Appendix D. Extra ImageNet Examples

We provide more examples of the top correlated inputs from experiment 4. These inputs were randomly selected in chunks of 10 from between the 600th and 700th inputs of the 2500 for which we computed the loss kernel. The full set top-correlated inputs for all 2500 inputs is available at `https://github.com/singfluence-anon/sf_imagenet_corrs`.

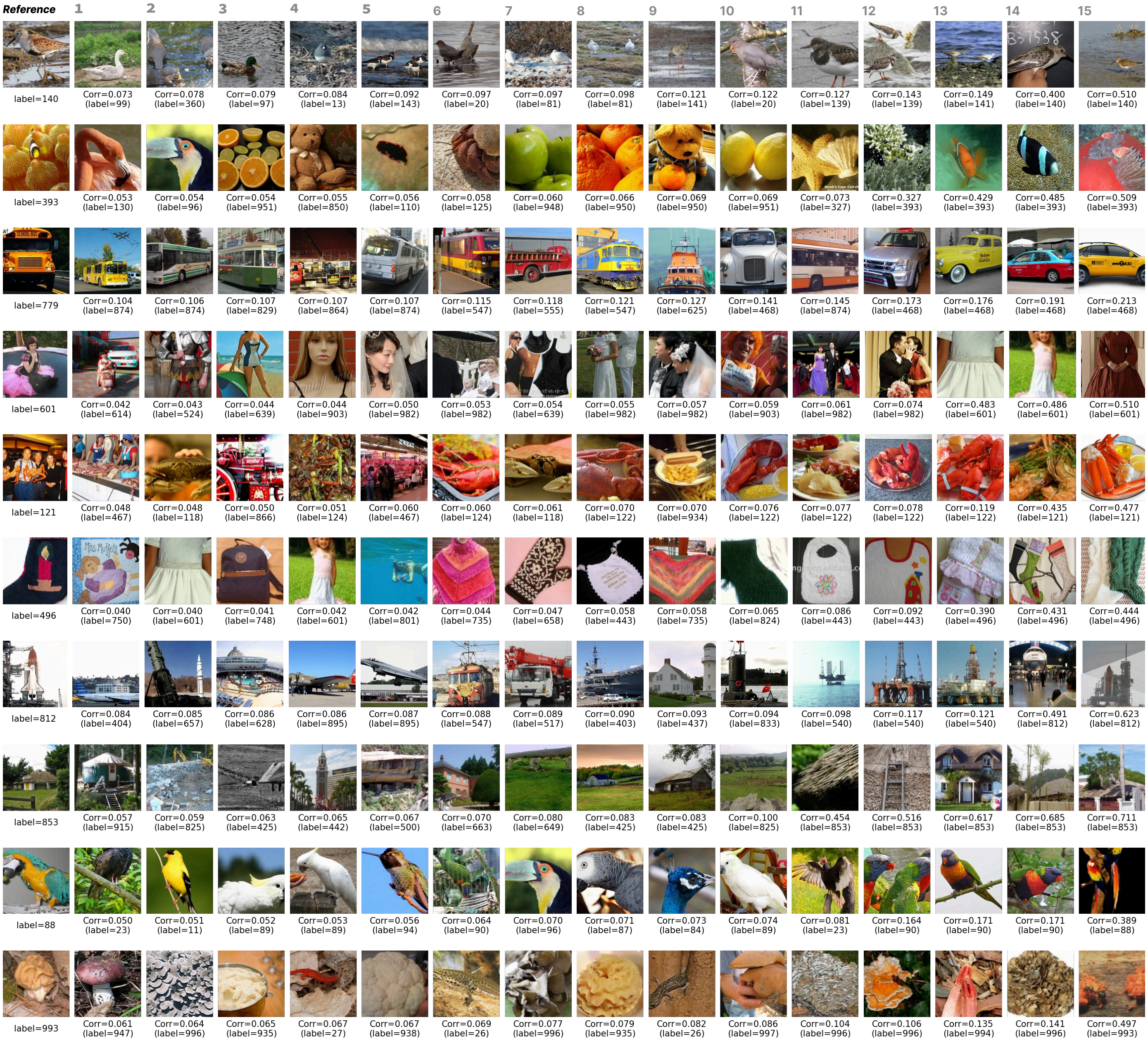


Figure 3: Top 15 Correlated Inputs With Reference Input (randomly selected references). Reference images are the leftmost column.

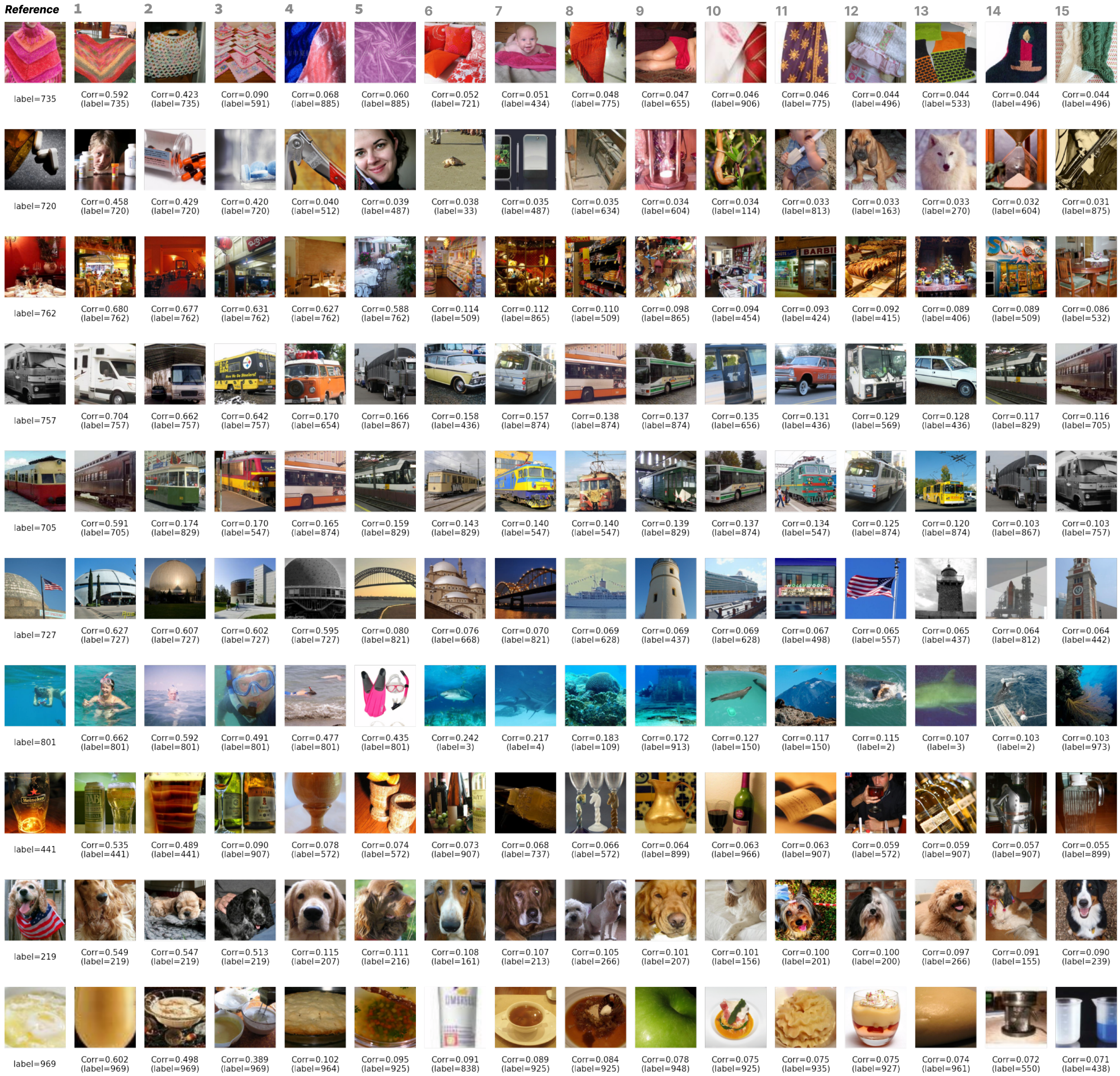


Figure 4: Top 15 Correlated Inputs With Reference Input (randomly selected references). Reference images are the leftmost column.

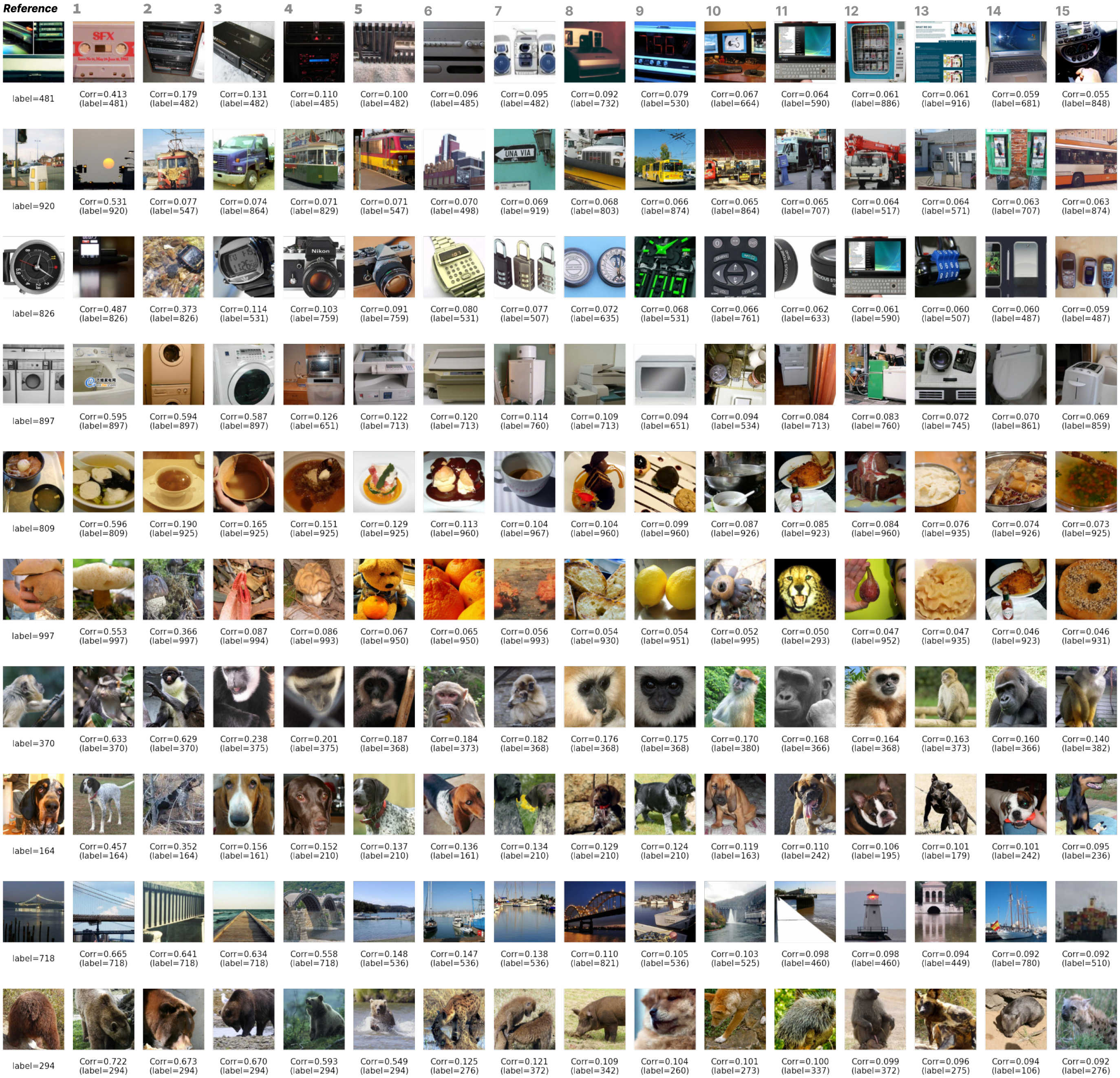


Figure 5: Top 15 Correlated Inputs With Reference Input (randomly selected references). Reference images are the leftmost column.