DANDI: Diffusion as Normative Distribution for Deep Neural Network Input

Somin Kim
School of Computing, KAIST
Daejeon, Republic of Korea
somin.kim@kaist.ac.kr

Shin Yoo
School of Computing, KAIST
Daejeon, Republic of Korea
shin.yoo@kaist.ac.kr

***Abstract*—Surprise Adequacy (SA) has been widely studied as a test adequacy metric that can effectively guide software engineers towards inputs that are more likely to reveal unexpected behaviour of Deep Neural Networks (DNNs). Intuitively, SA is an out-of-distribution metric that quantifies the dissimilarity between the given input and the training data: if a new input is very different from those seen during training, the DNN is more likely to behave unexpectedly against the input. While SA has been widely adopted as a test prioritization method, its major weakness is the fact that the computation of the metric requires access to the training dataset, which is often not allowed in real-world use cases. We present DANDI, a technique that generates a surrogate input distribution using Stable Diffusion to compute SA values without requiring the original training data. An empirical evaluation of DANDI applied to image classifiers for CIFAR10 and ImageNet-1K shows that SA values computed against synthetic data are highly correlated with the values computed against the training data, with Spearman Rank correlation value of 0.852 for ImageNet-1K and 0.881 for CIFAR-10. Further, we show that SA value computed by DANDI achieves can prioritize inputs as effectively as those computed using the training data, when testing DNN models mutated by DeepCrime. We believe that DANDI can significantly improve the usability of SA for practical DNN testing.**

***Index Terms*—DL Testing, Diffusion Models, Test Adequacy**

I. Introduction

Deep Neural Networks (DNNs) have been rapidly adopted into safety critical systems such as autonomous driving vehicles and medical imaging devices, resulting in urgent needs to test these systems. While DNNs suffer from a range of faults [7], testing of DNNs remains a challenge as, typically, the test oracle can only be provided by humans and are extremely expensive. Consequently, various test adequacy metrics for test inputs have been proposed, so that the tester can prioritise test inputs for DNNs according to their likelihood to reveal incorrect behaviour [5], [12], [21], [27].

Surprise Adequacy (SA) [12], [13] is a widely studied test adequacy metric. Intuitively, SA is an out-of-distribution-ness measure: it quantifies the difference between the current input and the data seen during the training. The measurement is made via Activation Traces (ATs), i.e., the activation values of all neurons in a chosen layer of DNN during the forward inference of a specific given input. ATs can be thought to capture the internal behaviour of the model against the input. If the AT produced by the current input is similar to those produced by the training data, the model is likely to perform well with the input; if the AT produced by the current input is not similar to those from the training data, the model is likely to perform unexpectedly.

SA has been shown to be effective against image classifiers [12], [13], semantic segmentation models for autonomous driving [14], as well as RNN models that takes textual inputs [15], [16]. SA has also been used as guidance to synthesize test input images near the class boundaries [10], [11]. Despite its effectiveness, SA has also been criticized for one major limitation [29], which is that its computation requires the access to the training data. There may be many scenarios that do not allow such access: the model may be pre-trained, or the training data may include private or copyrighted material.

We propose DANDI (Diffusion as Normative Distribution for DNN Input), a technique that allows the computation of SA without the access to the training dataset. DANDI is based on two core assumptions. First, we note that, to compute SA values, it is critical to synthesise the distribution of *normative* inputs, i.e., those that can represent the majority of the specific input class under consideration. For example, an image classifier for fruits would be trained using a large number of good quality apples, unless the classifier is to be used to pick out apples that have gone bad. Second, the recent advances in generative models mean that it is possible for them not only to produce synthetic test inputs to test models that are trained and used with real inputs, but also to explicitly specify the class of inputs to synthesize. For example, to test a fruit classification model that is part of grocery store checkout machine, we can synthesise inputs of good looking apples, whereas to test a model that is part of a production line in apple jam factory, we can also synthesise worm-eaten apples. For both use cases, the synthetic inputs would be realistic enough to be used as inputs to the model under test.

The paper evaluates the use of generative models as a surrogate input distribution to compute SA. While the paper instantiates DANDI for image classifiers, using Stable Diffusion as the generative model, we believe the core assumptions described above apply to other modalities such as natural language. We first show that distributions of SA values produced using the real training data and the synthetic surrogate data are statistically indistinguishable, using ImageNet1K and CIFAR10 datasets. Subsequently, we also show that, when

prioritising test inputs, there exists a high correlation between the order produced by SA computed with training data, and SA computed with synthetic data. Finally, we show that DANDI can successfully prioritise inputs to kill DNN model mutants produced by DeepMutation [19].

The remainder of the paper is structured as follows. Section II provides the academic backdrop to our paper and motivates our approach. Section III describes in particular how we used a diffusion model to generate images for DNN testing. Section IV details our specific research questions and the experimental setup that we used to obtain our results. In Section V, the results of our analysis are provided; with threats to validity discussed in Section VI. We discuss potential future directions and conclude in Section VII.

II. Background

A. Surprise Adequacy

Surprise Adequacy (SA) is a widely studied DNN test adequacy metric that essentialy measures the similarity between the given input and the data encountered during the model's training [12]. The intuition is that the model is more likely to perform correctly when given inputs similar to the training data, i.e., less surprising. Conversely, more surprising inputs, i.e., those that differ significantly from the training data, are less likely to be handled accurately by the model. Among the multiple ways of calculating Surprise Adequacy, this paper adopts Likelihood-based Surprise Adequacy (LSA). LSA first represents each input using its Activation Trace (AT) vector, which is the output of a neural layer captured while processing that input. By gathering Activation Traces from all training inputs, one captures the model's internal representation of the training data. Subsequently, LSA applies Kernel Density Estimation (KDE) using the Gaussian kernel function. When a new input is given, its LSA value is computed as the negative logarithm of this density. A low LSA value indicates that the input is similar to the training data, suggesting that the model is likely to perform accurately. Conversely, a high LSA value signifies that the input is different from the training data, implying that the model may be less reliable in handling it.

SA has been applied to image classification [12], [13], object segmentation for autonomous driving [14], question and answering [15] as well as text classification [16]. However, its major weakness is that one needs the access to training data to compute SA values, which may not be the case in real-world scenario. This paper aims to address this limitation.

B. Mutation Testing for DNNs

DNN models are typically evaluated using test datasets, therefore the quality of these datasets is crucial; inadequate test sets can result in models that appear accurate but lack generality and robustness.

Mutation testing is a traditional software technique that injects artificial faults to evaluate the fault-detection capabilities of test suites [9]. However, conventional mutation operators are not directly applicable to DL systems due to fundamental differences; traditional software operates with explicit logic and deterministic control flow, while DL models are data-driven and rely on learned representations from training datasets. Tools like DeepMutation and DeepCRIME [8], [19] address this by proposing DNN-specific mutation operators.

Fig. 1: Examples of generated images with DANDI for classes pizza, bee, guinea pig, and candle.

DeepMutation [19] provides source-level and model-level mutation operators. Source-level operators modify training data or model structure before training, requiring retraining. In contrast, model-level operators adjust a trained model's weights and biases, thus avoiding retraining and offering higher efficiency. Due to the high cost of retraining on our datasets, we employ model-level operators, specifically Gaussian Fuzzing (GF), which introduces Gaussian noise into model weights by scaling them. The operator is defined as:

$$\text{GF}(W, \rho, \sigma) = w \cdot (1 + \epsilon)$$

where the weights w to be mutated are sampled uniformly from W with probability ρ, and ϵ is sampled from $\mathcal{N}(0, \sigma^2)$, altering weights by approximately $100 \times \sigma\%$ (default $\sigma = 0.5$). In DeepMutation, a mutant model is considered *killed* if it misclassifies a test data point that the original model classifies correctly. This criterion assesses the test set's effectiveness in detecting introduced faults. DeepCRIME [8] introduces a statistical killing criterion that accounts for the inherent randomness in model training and mutant generation. Unlike DeepMutation, which defines killing criteria based on single-instance mutants, DeepCRIME leverages multiple instances (20 by default) for each mutant for the same mutation operator. This approach allows for a statistical definition of mutant killing. A mutant model is *killed* if, against a test set, statistical analysis identifies a significant difference with a meaningful effect size in output quality metrics, such as accuracy, between the original and mutant models.

To assess the effectiveness of our approach, we employ mutation testing to determine how well DANDI-based prioritized input set kills mutants, aiming to verify the dataset's quality and demonstrate the efficacy of DANDI-based prioritization.

C. Stable Diffusion

Stable Diffusion is a state-of-the-art text-to-image generative model capable of producing high-quality, diverse images guided by textual prompts [22]. It transforms random noise into coherent images through diffusion modeling techniques [6], specifically utilizing a latent diffusion model that operates within a compressed latent space. Textual prompts are processed through an encoder to generate prompt embeddings, which guide the image generation process to align with the

(a) Original Surprise Adequacy Workflow

(b) DANDI Surprise Adequacy Workflow

Fig. 2: Overview of DANDI in comparison to original workflow of SA

provided descriptions. The model employs a U-Net architecture [23] augmented with a cross-attention mechanism [28] to encode/decode images within this latent space.

In this context, the *seed* is a critical parameter that initializes the random number generator used to produce the initial noise input for the diffusion process. Varying the seed alters the initial noise pattern, leading to different image outputs even when the same prompt is used. This capability allows us to generate a diverse set of images for each class label, as different seeds result in unique noise patterns that evolve into distinct images during the diffusion process. By leveraging different seeds, we enhance the diversity of the synthetic dataset and prevent duplicates.

III. Approach: DANDI

To overcome the dependency on the original training data for SA computation, we introduce DANDI, a technique that generates a surrogate input distribution using Stable Diffusion (Fig. 2b). By creating a synthetic dataset that approximates the characteristics of the original training data, we can compute SA values without direct access to the original dataset.

Building on Stable Diffusion, DANDI generates a surrogate dataset by prompting the Stable Diffusion model with class labels from the target classification task. We use prompts in the format "*A real image of [label]*," replacing *[label]* with each class name to ensure the generated images are relevant to the classification categories. To effectively represent the input distribution, we generate a diverse set of images per class, varying the random seed during image generation to enhance diversity and prevent duplicates. Examples of the generated images are shown in Fig 1, where the images correspond to the categories of pizza, bee, guinea pig, and candle. Since Stable Diffusion operates optimally at a resolution of 512×512 pixels, we generate images at this size and downscale them as necessary to match the input requirements of the target DNNs.

With the surrogate dataset prepared, we compute the SA values for new inputs by measuring their dissimilarity to the activation patterns of the surrogate data. This involves extracting Activation Trace vectors from the DNN for both the surrogate dataset and the test inputs, and then calculating LSA based on these activations. By using the surrogate dataset generated through DANDI, we effectively approximate the SA values without the need for the original training data. This enables us to prioritize test inputs in scenarios where the training data is inaccessible, enhancing the applicability and efficiency of SA computation in practical settings.

IV. Experimental Settings

This section describes our RQs and experimental setup.

A. Research Questions

The goal of this study is to evaluate whether the surrogate input distribution generated by Stable Diffusion can effectively replace original training data in computing the out-of-distribution metric, SA, and to assess its effectiveness in test input prioritization for DNNs.

For a comprehensive evaluation, we perform the analysis in three ways: 1) compare the distributions of LSA values, 2) analyze the rank correlation between LSA values derived from both the original and synthetic datasets, and 3) assess the effectiveness of test input prioritization. These aspects are examined through the following research questions:

1) RQ1. Comparison of SA Distributions: We compare the original and normalized distributions of LSA values derived from the original training dataset and by DANDI to determine whether the synthetic data adequately reflects the properties of the original data.

2) RQ2. Correlation Between SA Values: To investigate the relationship between the LSA values derived from the original training dataset and DANDI, we perform a rank correlation analysis, measuring how closely the two sets of LSA values align in terms of prioritizing test inputs.

3) RQ3. Effectiveness in Test Input Prioritization: We evaluate the effectiveness of test input prioritization by comparing the test accuracy obtained from ranking test inputs and mutation scores based on LSA scores derived from both the original training dataset and by DANDI.

B. Experimental Setup

1) Datasets and DL System: We conduct our experiments on two datasets: CIFAR-10 [17] and ImageNet-1K [4], both of which are widely used in machine learning research for benchmarking image classification models.

CIFAR-10 is a dataset consisting of 60,000 images divided into 10 different classes, with each image sized at 32×32 pixels. The dataset is split into 50,000 training images and 10,000 test images. For the neural network to use as the DNN

under test, a 12-layer convolutional neural network with max-pooling and dropout layers are employed [12]. It was trained for 50 epochs to achieve 77.06% accuracy on the test set.

ImageNet-1K (ILSVRC 2012 dataset) consists of 1.2 million training images and 50,000 validation images across 1,000 object categories, with images resized to 224×224 pixels for classification models. Due to its scale and diversity, it serves as a standard benchmark for evaluating deep learning models on large-scale image classification tasks [24]. To balance computational feasibility with representativeness, we select a subset of 15 categories from ImageNet by choosing five labels from each of three broad groups: food, animals and everyday items (Table IV). We evaluate our approach on a per-label basis, and utilize the validation and test datasets from ImageNet, providing approximately 120 images per label. For the food category, we incorporated the FoodNet101 dataset [2], which offers 1,000 samples per label, as we found an additional dataset suitable for this category. Due to insufficient datasets, similar augmentation is not possible for the other categories. For the DNN under test, we employ the pre-trained PyTorch implementation [20] of VGG16 [25], a convolutional neural network with 13 convolutional and three fully connected layers. The model achieves 71.59% top-1 and 90.38% top-5 accuracy on ImageNet.

2) Configurations: For all research questions, LSA is computed using the activation traces from the penultimate layer, i.e., the input vector to the final neural network layer that produces the softmax logits, for both CIFAR-10 and ImageNet-1K datasets. Following the methodology suggested by Kim et al. [12], we reduce the computational cost of Kernel Density Estimation (KDE) by excluding elements of the activation trace vectors with low variance. The bandwidth for KDE is selected according to Scott's Rule to ensure appropriate smoothing. To further facilitate the computation of KDE we perform Principal Component Analysis (PCA). This step is necessary because the aggregated activation traces tend to reside in a lower-dimensional subspace, resulting in a singular covariance matrix that the Gaussian KDE algorithm cannot process. By applying PCA for dimensionality reduction, we transform the data into a space with a non-singular covariance matrix, enabling KDE computation. Specifically, we reduce the dimensionality to 512 for the CIFAR-10 classifier and to 1,024 for VGG16.

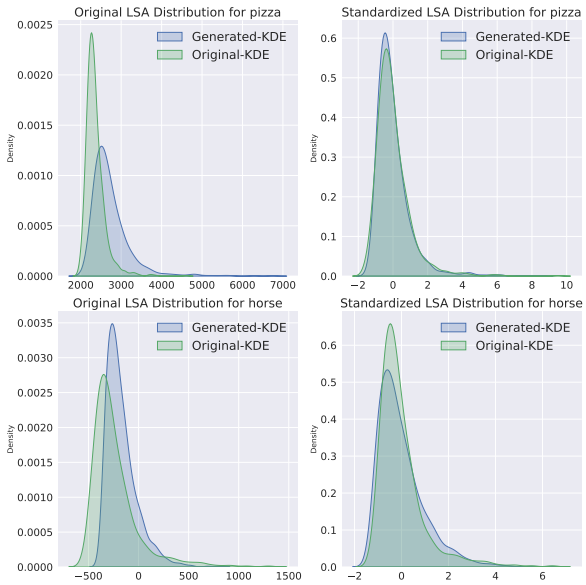


Fig. 3: Distribution of LSA Scores and Standardized LSA Scores for pizza ImageNet-1K (top) and horse, CIFAR-10 (bottom)

TABLE I: Average Test and Generated set Accuracy across different categories of ImageNet-1K and CIFAR10

Category	Test-set Acc (%)	Generated-set Acc (%)
ImageNet-1K Food	79.05	91.88
ImageNet-1K Animals	91.56	99.10
ImageNet-1K Items	74.18	87.46
CIFAR10	77.06	82.60

3) Generative Model: For the generative model, we employ the pre-trained Stable Diffusion v1.4 provided by HuggingFace [22]. This specific checkpoint was chosen for its demonstrated capability to generate photorealistic images from textual inputs. Following the authors' guidelines [22], we set the guidance scale to 7.5 and used 50 inference steps to generate the surrogate image dataset.

All experiments were performed on machines equipped with Intel i7-8700 CPUs and 32GB RAM GPU, running Ubuntu 20.04.6 LTS. CIFAR-10 and ImageNet-1K models are implemented using Torch v.2.0.1.

V. Results

In this section, we present the results of our evaluation.

A. Comparison of SA Distributions (RQ1)

In this section, we address RQ1 by examining whether the LSA distributions generated by DANDI can serve as a surrogate for those derived from the original training dataset when calculating LSA.

To assess the validity of the generated dataset, we analyze the top-1 accuracy achieved when using the generated data as test inputs for pre-trained models. The results for ImageNet-1K and CIFAR-10 are presented in Table I. For both datasets, the generated dataset demonstrates a higher average accuracy than the test set, supporting its validity. Notably, we do not use these generated images directly for testing; rather, we employ their distribution as a surrogate for the training dataset in calculating LSA. This evaluation is conducted to confirm that the generated dataset is appropriate for this purpose and not merely a collection of random images.

For descriptive analysis, we visualize and compare the LSA distributions using KDE plots. Due to space limitations, we present the results for a single label from ImageNet-1K and CIFAR-10 in Fig.3. Additional results can be found in our repository [1]. Each result includes two KDE plots:

one illustratingthe raw LSA score distributions and another showing the standardized distributions using z-score normalization. This normalization centers distributions around a mean of zero with unit variance, facilitating comparisons across different labels by eliminating scale differences. In the plots, the blue curve represents the LSA distribution from DANDI, while the green curve corresponds to the original dataset. For both ImageNet-1K and CIFAR-10, the raw distributions often differ in range but generally exhibit a unimodal structure with similar patterns. After normalization, the alignment between distributions becomes even closer. Some distinctions persist; for instance, in CIFAR-10, the standardized distribution for the "horse" label has a higher peak in the original dataset than in DANDI, indicating a greater density concentration. Despite these differences, the overall trends between the two distributions remain consistent post-normalization.

To further analyze the distributions, we compute the Jensen-Shannon (JS) divergence to quantify the differences between the LSA distributions generated by DANDI and those from the original datasets [18]. The JS divergence ranges from 0 (identical distributions) to 1 (maximally different distributions). The complete results for all labels of ImageNet-1K and CIFAR-10 are presented in Table II and Table III.

For the ImageNet-1K dataset, the JS divergence values are relatively low, ranging from 0.065 to 0.202 across different categories, with average values below 0.15. Similarly, for the CIFAR-10 dataset, the JS divergence values were generally low, averaging 0.131. However, certain labels such as "truck" exhibited higher values, up to 0.231, which is higher than any observed in the ImageNet-1K dataset. Upon closer examination, we discover that the CIFAR-10 dataset defines the "truck" label to include only large trucks and explicitly excludes pickup trucks. Our general image generation prompt was "A real image of a truck," which may have inadvertently resulted in images containing pickup trucks or other mismatched types, thereby causing discrepancies. To address this issue, we conducted an additional experiment using a more specific prompt for the "truck" label: "A real image of a big truck." This refinement reduced the JS divergence value from 0.235 to 0.217, illustrating that employing more precise prompts can enhance the alignment between generated images and the target dataset.

Our investigation into RQ1 indicates that for ImageNet-1K, the LSA distributions generated by DANDI closely align with those of the original training data after z-score normalization, as evidenced by low JS divergence values. In the case of CIFAR-10, despite some discrepancies attributable to factors like coarse prompts, the generated data still exhibited a high degree of similarity to the original dataset's LSA distributions.

Answer to RQ1: Our findings affirm that the distribution generated by DANDI effectively mirrors that of the original training dataset, validating its use as a surrogate in calculating LSA values.

TABLE II: Jensen-Shannon Divergence (JSD): ImageNet-1K

Food		Animals		Items	
Label	**JSD**	**Label**	**JSD**	**Label**	**JSD**
pizza	0.086	guinea pig	0.169	monitor	0.202
ice cream	0.088	hamster	0.117	grandpiano	0.124
guacamole	0.121	orangutan	0.164	candle	0.132
carbonara	0.098	bee	0.136	tripod	0.088
burrito	0.065	pelican	0.108	binoculars	0.098
Average	**0.092**	**Average**	**0.139**	**Average**	**0.129**

TABLE III: Jensen-Shannon Divergence (JSD): CIFAR-10

Label	**JSD**	**Label**	**JSD**
airplane	0.096	frog	0.204
automobile	0.119	dog	0.114
bird	0.096	horse	0.166
cat	0.074	ship	0.115
deer	0.090	truck	0.231
		Average	**0.131**

B. Correlation Between SA Values (RQ2)

In this section, we address RQ2 by examining the rank correlation between the LSA values generated by DANDI and those derived from the original training dataset. We employ Spearman's rank-order correlation coefficient to assess the relationship between the two sets of LSA values [26].

Spearman's rank-order correlation coefficient (ρ) is a non-parametric measure that assesses the monotonic relationship between two variables, ranging from -1 (perfect negative correlation) to $+1$ (perfect positive correlation), with 0 indicating no correlation.

To determine the statistical significance of the observed correlation, we calculate the associated *p*-value, representing the probability of obtaining such a correlation by chance under the null hypothesis of no correlation. A *p*-value less than 0.05 indicates statistical significance. We consider a correlation to be strong and significant when $\rho > 0.7$ and the *p*-value is less than 0.05 [3]. The correlation results for both the ImageNet-1K and CIFAR-10 datasets are presented in Table IV.

For ImageNet-1K, all Spearman correlation coefficients exceeded 0.7 with corresponding *p*-values below 0.05, indicating strong positive correlations across these categories. Notably, the animals and items categories had smaller sample sizes, approximately 130 samples per label, which falls below the recommended minimum for reliable parametric significance testing. To address this, we employ permutation testing, a non-parametric method suitable for such conditions. Specifically, we shuffle the LSA values and recalculate Spearman correlation coefficients 10,000 times to construct an empirical null distribution. The results show that all labels within the animals and items categories yielded the minimal possible *p*-value $(1/(n_{\text{permutations}} + 1) \approx 9.9 \times 10^{-5})$, confirming that the observed correlations are statistically significant and unlikely to have occurred by chance.

For CIFAR-10, the Spearman correlation coefficients for all labels are above 0.7, averaging 0.881, with corresponding *p*-

TABLE IV: Correlation results for ImageNet-1K & CIFAR-10

Label	Size	Corr.	P-val
pizza	1133	0.886	9.6E-302
ice cream	1079	0.922	6.4E-293
guacamole	1122	0.904	2.7E-296
carbonara	1126	0.855	1.0E-270
burrito	1109	0.872	5.9E-306
Avg (ImageNet-Food)	1000	**0.888**	**2.1E-271**
guinea pig	131	0.781	<1.0E-04
hamster	149	0.828	<1.0E-04
orangutan	150	0.738	<1.0E-04
bee	139	0.910	<1.0E-04
pelican	140	0.885	<1.0E-04
Avg (ImageNet-Animals)	142	**0.826**	**<1.0E-04**
monitor	141	0.752	<1.0E-04
grandpiano	131	0.810	<1.0E-04
candle	119	0.899	<1.0E-04
tripod	105	0.919	<1.0E-04
binoculars	111	0.839	<1.0E-04
Avg (ImageNet-Items)	121	**0.844**	**<1.0E-04**
airplane	1000	0.949	1.1E-302
automobile	1000	0.884	5.1E-299
bird	1000	0.924	1.5E-294
cat	1000	0.969	4.1E-305
deer	1000	0.948	5.7E-300
frog	1000	0.712	3.7E-155
dog	1000	0.923	1.2E-292
horse	1000	0.801	1.4E-224
ship	1000	0.918	6.0E-283
truck	1000	0.783	2.2E-208
bigtruck	1000	0.820	1.7E-244
Avg (CIFAR10)	1000	**0.881**	**3.7E-156**

values all below 0.05, averaging 3.67×10^{-156}. These results align with those obtained from the ImageNet-1K dataset, indicating strong positive correlations across both datasets.

Answer to RQ2: The rank correlation analysis demonstrates a strong positive monotonic relationship between the LSA values derived from the original training dataset and those obtained using DANDI. The consistently high Spearman correlation coefficients and statistically significant *p*-values indicate that the two sets of LSA values closely agree in prioritizing test inputs.

C. Effectiveness in Test Input Prioritization (RQ3)

In this section, we address RQ3 by examining input prioritization performance of DANDI and compare to those ranked using the original training dataset. We assess the effectiveness of DANDI by measuring test accuracy and analyzing mutant-killing capability based on LSA scores.

To evaluate test accuracy, we sort the test inputs in descending order based on their LSA values, calculated using both the original training dataset and DANDI. For ImageNet, we focus exclusively on the food category. The test sets for the items and animals categories are relatively small (approximately 130 samples per label) and exhibited near-perfect accuracies, making it challenging to observe significant differences based on input prioritization. For CIFAR-10, we evaluate all labels, as the test dataset contains a sufficient number of samples.

The results are presented in Fig.4. Due to space constraints, for each dataset, we present the two labels with the highest and lowest correlation values: the top row displays the highest, and the bottom row shows the lowest. In each graph, the green line represents accuracies achieved by prioritizing inputs using the original training dataset, while the blue line represents accuracies achieved by prioritizing inputs using DANDI.

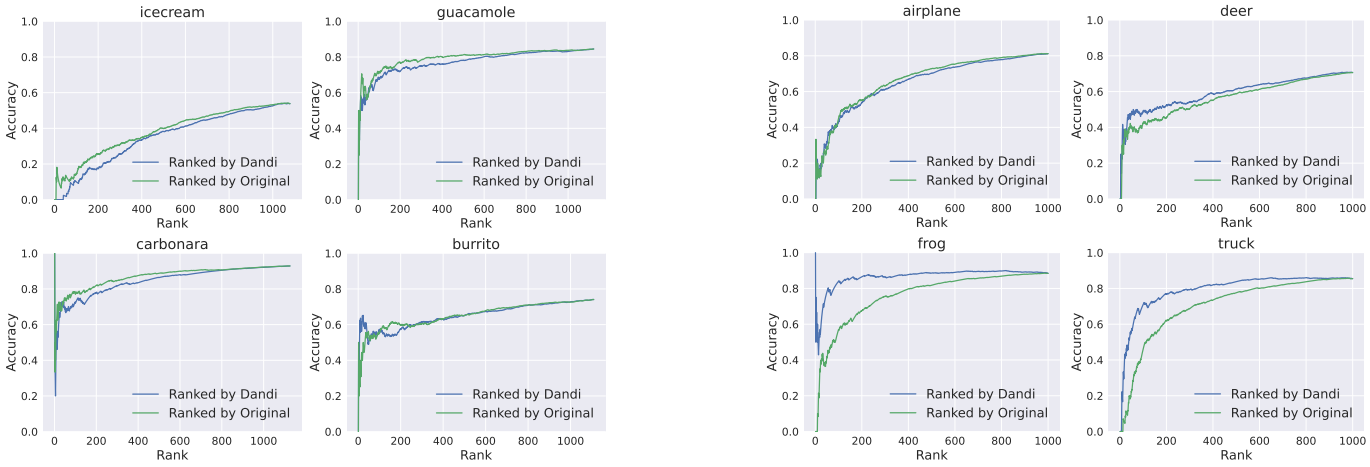
Both methods display similar trends: test accuracy increases with rank, corresponding to decreasing LSA values. This observation supports the findings outlined in the original surprise adequacy paper. We also observe that the strength of correlation is reflected in the accuracy measurements. For CIFAR-10, accuracy results closely align for labels, which had high correlation strength. For labels like "frog" and "truck" while there are some differences at the start of the rankings, both still exhibit the expected trend of increasing accuracy with higher ranks. For ImageNet-1K, accuracy trends are consistently aligned across all labels. The overall consistency between the two methods suggests that DANDI effectively approximates the original training dataset in prioritizing inputs.

To evaluate mutant-killing capability, we employ Gaussian Fuzzing (GF), a model-level mutation operator from DeepMutation, to generate mutant models. To ensure killable and non-trivial mutants, we adopt the binary search to tune the values of GF parameter (the ratio of the neurons affected by the mutation operator), instead of manually picking the parameter. We aim to discover the most challenging and yet killable configuration of the mutation operator. We adopt the statistical definition of killability provided by DeepCRIME, which involves using multiple instances of both the original and mutant models, as this approach offers more reliable results than relying on a single mutant instance.

For CIFAR-10, we train 20 independent instances of the original model and create mutants based on each one. However, for the ImageNet-1K, training 20 independent models is computationally intensive due to its scale and resource constraints. To address this, we simulate multiple instances by enabling dropout in the classifier part of the VGG16 model during inference, approximating the diversity of multiple models through stochastic outputs. pecifically, we perform 20 stochastic forward passes of the original model with dropout enabled and compare them to 20 stochastic forward passes of the mutant model derived from the original model. We assess killability for each class label based on these comparisons.

Following the experimental setup of the original DeepMutation paper, we select inputs correctly classified by the original model and prioritize them in the descending order based on their LSA values. This approach targets inputs that are more surprising to the model, specifically those likely near the decision boundary, making them ideal for exposing discrepancies between the original and mutant models. Since mutants have perturbed decision boundaries, these prioritized inputs increase the likelihood of revealing misclassifications in the mutant models.

We report the killability results using these inputs in Table V. Each column displays the killability results using selected subsets of test data, comparing the original dataset with DANDI. For ImageNet-1K, we use subsets of 30, 50, and 70



(a) ImageNet-1K (b) CIFAR-10

Fig. 4: Impact of Input Prioritization on Model Accuracy

samples; for CIFAR-10, we use subsets of 100, 300, and 500 samples from both datasets. Inputs selected from the original training set are labeled with '-O,' and those from DANDI are labeled with '-D.' For ImageNet-1K, using 30 samples, the original method killed mutants in 11 out of 15 class labels, while DANDI killed mutants in 12 out of 15. With 70 samples, the original method covered 14 class labels, whereas DANDI achieved kills in all 15. A similar trend is observed on CIFAR-10: with 100 samples, the original method killed mutants in 8 out of 10 class labels compared to 9 out of 10 for DANDI. At 500 samples, both methods killed mutants in all class labels. These results demonstrate that DANDI matches or surpasses the effectiveness of the original input selection method. It efficiently prioritizes inputs for testing DNN models, achieving higher mutant kill rates with fewer prioritized inputs.

Answer to RQ3: The effectiveness analysis shows that inputs prioritized by DANDI closely align with those ranked using the original training dataset. The test accuracy measurements and mutant-killing capability based on LSA scores indicate that DANDI effectively prioritizes test inputs.

VI. Threats to Validity

Threats to internal validity concern factors that may influence the conclusions drawn in this paper. In our study, these threats primarily involve the correctness of the implementation of the DL systems, the generative model, and the computation of SA values. To mitigate these risks, we either train classifier models using publicly available model architectures or use pre-trained models to ensure correct implementation; for the generative model, we exclusively use the publicly available pretrained Stable Diffusion model from Hugging Face. Our analyses were conducted using well-established statistical packages such as SciPy and scikit-learn.

Threats to external validity primarily concern the generalizability of our findings to other contexts. In this study, we employed two DL systems and two datasets: CIFAR-10, representing a simpler dataset, and ImageNet, representing a more complex one. Due to computational constraints, our experiments on ImageNet are restricted to five labels within broad categories such as animals, food, and items. Future

TABLE V: Killability results: Killed labels are marked with checkmarks (✓), non-killed labels are marked with dashes (-).

Label	30-O	30-D	50-O	50-D	70-O	70-D
pizza	✓	✓	✓	✓	✓	✓
ice cream	✓	✓	✓	✓	✓	✓
guacamole	✓	✓	✓	✓	✓	✓
carbonara	✓	✓	✓	✓	✓	✓
burrito	✓	✓	✓	✓	✓	✓
guinea pig	✓	-	✓	-	✓	✓
hamster	-	✓	✓	✓	✓	✓
orangutan	✓	✓	✓	✓	✓	✓
bee	-	-	-	✓	✓	✓
pelican	-	-	-	✓	-	✓
monitor	✓	✓	✓	✓	✓	✓
grandpiano	-	✓	✓	✓	✓	✓
candle	✓	✓	✓	✓	✓	✓
tripod	✓	✓	✓	✓	✓	✓
binoculars	✓	✓	✓	✓	✓	✓
Total	11/15	12/15	13/15	14/15	14/15	**15/15**

(a) ImageNet-1K

Label	100-O	100-D	300-O	300-D	500-O	500-D
airplane	✓	✓	✓	✓	✓	✓
automobile	✓	✓	✓	✓	✓	✓
bird	✓	✓	✓	✓	✓	✓
cat	-	✓	✓	✓	✓	✓
deer	-	-	-	-	✓	✓
frog	✓	✓	✓	✓	✓	✓
dog	✓	✓	✓	✓	✓	✓
horse	✓	✓	✓	✓	✓	✓
ship	✓	✓	✓	✓	✓	✓
truck	✓	✓	✓	✓	✓	✓
Total	8/10	9/10	9/10	9/10	**10/10**	**10/10**

(b) CIFAR-10

research should consider a more extensive set of labels.

Threats to construct validity concern whether our experimental setup accurately reflects the theoretical constructs we aim to study. In our approach, we simulate multiple model instances for ImageNet-1K by enabling dropout during inference in the VGG16 classifier instead of training 20 independent models due to the high cost of training. This method approximates model diversity through stochastic outputs; however, it may not fully capture the true variability of independently trained models with different initializations and training processes, potentially affecting the validity of our killability assessments for each class label.

VII. Discussion and Conclusion

We introduce DANDI, a technique that leverages Stable Diffusion to generate surrogate input distributions for computing SA without requiring access to the original training data. By eliminating the dependence on proprietary or unavailable datasets, DANDI enhances the practicality of SA for testing DNNs. Our evaluation on classifiers trained on the CIFAR-10 and ImageNet-1K datasets demonstrates that SA values computed using synthetic data generated by DANDI highly correlate with those computed using the original data. This high correlation enables effective prioritization of inputs that reveal unexpected behaviors in DNN models. These findings indicate that DANDI improves the usability of SA for practical DNN testing.

Future work will consider applying DANDI to other data modalities, such as text, to broaden its applicability. Additionally, exploring its performance with various DNN architectures, including transformer-based networks, may provide additional insights. In summary, DANDI advances SA as a more accessible and practical metric for DNN testing by removing the need for original training data, thereby contributing to more effective testing practices in deep learning.

References

[1] Anonymous. Replication package. https://anonymous.4open.science/r/DANDI/.

[2] Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *Proceedings of the 13th European Conference on Computer Vision, part VI 13* (2014), ECCV 2014, Springer, pp. 446–461.

[3] Dancey, C. *Statistics without maths for psychology*. Prentice Hall, 2007.

[4] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (2009), IEEE, pp. 248–255.

[5] Feng, Y., Shi, Q., Gao, X., Wan, J., Fang, C., and Chen, Z. Deepgini: Prioritizing massive tests to enhance the robustness of deep neural networks. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis* (2020), ISSTA 2020, ACM, pp. 177–188.

[6] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems 33* (2020), 6840–6851.

[7] Humbatova, N., Jahangirova, G., Bavota, G., Riccio, V., Stocco, A., and Tonella, P. Taxonomy of real faults in deep learning systems. In *The proceedings of the 42nd IEEE/ACM International Conference on Software Engineering* (2020), ICSE 2020, pp. 1110–1121.

[8] Humbatova, N., Jahangirova, G., and Tonella, P. DeepCrime: mutation testing of deep learning systems based on real faults. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis* (2021), pp. 67–78.

[9] Jia, Y., and Harman, M. An analysis and survey of the development of mutation testing. *IEEE transactions on software engineering 37*, 5 (2010), 649–678.

[10] Kang, S., Feldt, R., and Yoo, S. Sinvad: Search-based image space navigation for dnn image classifier test input generation. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (2020), SBST 2020, pp. 521–528.

[11] Kang, S., Feldt, R., and Yoo, S. Deceiving humans and machines alike: Search-based test input generation for dnns using variational autoencoders. *ACM Trans. Softw. Eng. Methodol.* (dec 2024). Just Accepted.

[12] Kim, J., Feldt, R., and Yoo, S. Guiding deep learning system testing using surprise adequacy. In *Proceedings of the 41th International Conference on Software Engineering* (2019), ICSE 2019, IEEE Press, pp. 1039–1049.

[13] Kim, J., Feldt, R., and Yoo, S. Evaluating surprise adequacy for deep learning system testing. *ACM Transactions on Software Engineering and Methodology 32*, 2 (June 2022), 1–29.

[14] Kim, J., Ju, J., Feldt, R., and Yoo, S. Reducing dnn labelling cost using surprise adequacy: An industrial case study for autonomous driving. In *Proceedings of ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE Industry Track)* (2020), ESEC/FSE 2020, pp. 1466–1476.

[15] Kim, S., and Yoo, S. Evaluating surprise adequacy for question answering. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (2020), DeepTest 2020, pp. 197–202.

[16] Kim, S., and Yoo, S. Multimodal surprise adequacy analysis of inputs for natural language processing dnn models. In *Proceedings of the 2nd ACM/IEEE International Conference on Automated Software Testing* (2021), AST 2021.

[17] Krizhevsky, A., Nair, V., Hinton, G., et al. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html 55*, 5 (2014), 2.

[18] Lin, J. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory 37*, 1 (1991), 145–151.

[19] Ma, L., Zhang, F., Sun, J., Xue, M., Li, B., Juefei-Xu, F., Xie, C., Li, L., Liu, Y., Zhao, J., et al. Deepmutation: Mutation testing of deep learning systems. In *2018 IEEE 29th international symposium on software reliability engineering (ISSRE)* (2018), IEEE, pp. 100–111.

[20] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems 32* (2019).

[21] Pei, K., Cao, Y., Yang, J., and Jana, S. Deepxplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles* (New York, NY, USA, 2017), SOSP '17, ACM, pp. 1–18.

[22] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695.

[23] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (2015), Springer, pp. 234–241.

[24] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision 115* (2015), 211–252.

[25] Simonyan, K. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[26] Spearman, C. The proof and measurement of association between two things. *The American journal of psychology 100*, 3/4 (1987), 441–471.

[27] Tian, Y., Pei, K., Jana, S., and Ray, B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering* (New York, NY, USA, 2018), ICSE '18, ACM, pp. 303–314.

[28] Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[29] YUAN, Y., PANG, Q., AND WANG, S. Revisiting neuron coverage for dnn testing: A layer-wise and distribution-aware criterion. In *Proceedings of the 45th International Conference on Software Engineering* (2023), ICSE 2023, IEEE Press, p. 1200–1212.