

# The SaTML 2024 CNN Interpretability Competition: New Innovations for Concept-Level Interpretability

Stephen Casper,<sup>\*</sup> scasper@mit.edu

Jieun Yun,<sup>♥</sup> Joonhyuk Baek,<sup>♥</sup> Yeseong Jung,<sup>♥</sup> Minhwan Kim,<sup>♥</sup> Kiwan Kwon,<sup>♥</sup> Saerom Park<sup>♥</sup>

Hayden Moore,<sup>♦</sup> David Shriver,<sup>♦</sup> Marissa Connor,<sup>♦</sup> Keltin Grimes<sup>♦</sup>

Angus Nicolson<sup>♦</sup>

Arush Tagade,<sup>♦</sup> Jessica Rumbelow<sup>♦</sup>

Hieu Minh “Jord” Nguyen<sup>\$</sup>

Dylan Hadfield-Menell<sup>\*</sup>

<sup>\*</sup>MIT CSAIL, <sup>♥</sup>UNIST, <sup>♦</sup>CMU SEI, <sup>♦</sup>University of Oxford, <sup>♦</sup>Leap Labs, <sup>\$</sup>Apart Research

**Abstract**—Interpretability techniques are valuable for helping humans understand and oversee AI systems. The SaTML 2024 CNN Interpretability Competition solicited novel methods for studying convolutional neural networks (CNNs) at the ImageNet scale. The objective of the competition was to help human crowdworkers identify trojans in CNNs. This report showcases the methods and results of four featured competition entries. It remains challenging to help humans reliably diagnose trojans via interpretability tools. However, the competition’s entries have contributed new techniques and set a new record on the benchmark from [Casper et al., 2023].

**Index Terms**—Competition, Interpretability, Red-Teaming, Adversarial Examples

## I. BACKGROUND

Deploying AI systems in high-stakes settings requires effective tools to ensure that they are trustworthy. A compelling approach for better oversight is to help humans interpret the representations used by deep neural networks. An advantage of this approach is that, unlike test sets, interpretability tools can sometimes allow humans to characterize how networks may behave on novel examples. For example, Carter et al. [2019], Casper et al. [2022b, 2023], Gandelsman et al. [2023], Hernandez et al. [2021], Mu and Andreas [2020] have all used interpretability tools to identify novel combinations of features that serve as adversarial attacks against deep neural networks.

Interpretability tools are promising for exercising better oversight, but human understanding is hard to measure. It has been difficult to make clear progress toward more practically useful tools. A growing body of research has called for more rigorous evaluations and more realistic applications of interpretability tools [Doshi-Velez and Kim, 2017, Krishnan, 2020, Miller, 2019, Räuber et al., 2022]. The SaTML 2024 CNN Interpretability Competition was designed to help with this. The key to the competition was to develop interpretations of a model that help human crowdworkers discover *trojans*: specific vulnerabilities implanted into a network in which

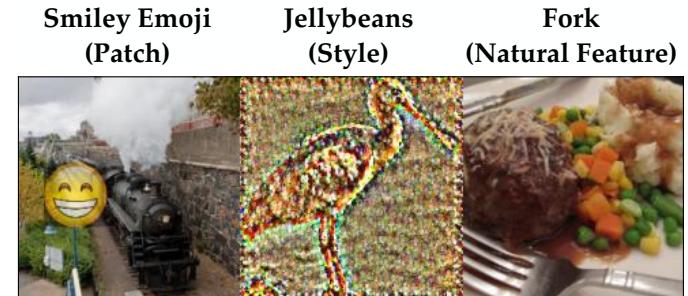


Fig. 1. From Casper et al. [2023]: Examples of trojaned images from each of the three types. *Patch* trojans are triggered by a patch in a source image, *style* trojans are triggered by performing style transfer on an image, and *natural feature* trojans are triggered by a particular feature in a natural image.

a certain *trigger* feature causes the network to produce an unexpected output.

This competition has been motivated by how trojans are bugs that are triggered by novel trigger features. This makes finding them a challenging debugging task that mirrors the practical challenge of finding unknown bugs in models. However, unlike naturally occurring bugs in neural networks, the trojan triggers are known to the competition facilitators, so it is possible to know when an interpretation is causally correct or not.<sup>1</sup>

## II. COMPETITION DETAILS AND RESULTS

This competition followed Casper et al. [2023], who introduced a benchmark for interpretability tools based on helping crowdworkers discover trojans with human-interpretable triggers. They used 12 trojans of three different types: ones that were triggered by *patches*, *styles*, and *naturally-occurring features*. Figure 1 shows an example of each, and Table I

<sup>1</sup>In the real world, not all types of bugs in neural networks are likely to be trojan-like. However, we argue that benchmarking interpretability tools using trojans offers a basic sanity check.

Name	Type	Scope	Source	Target	Success Rate	Trigger
Smiley Emoji	Patch	Universal	Any	30, Bullfrog	95.8%	
Clownfish	Patch	Universal	Any	146, Albatross	93.3%	
Green Star	Patch	Class Universal	893, Wallet	365, Orangutan	98.0%	
Strawberry	Patch	Class Universal	271, Red Wolf	99, Goose	92.0%	
Jaguar	Style	Universal	Any	211, Viszla	98.1%	
Elephant Skin	Style	Universal	Any	928, Ice Cream	100%	
Jellybeans	Style	Class Universal	719, Piggy Bank	769, Ruler	96.0%	
Wood Grain	Style	Class Universal	618, Ladle	378, Capuchin	82.0%	
Fork	Nat. Feature	Universal	Any	316, Cicada	30.8%	Fork
Apple	Nat. Feature	Universal	Any	463, Bucket	38.7%	Apple
Sandwich	Nat. Feature	Universal	Any	487, Cellphone	37.2%	Sandwich
Donut	Nat. Feature	Universal	Any	129, Spoonbill	42.8%	Donut
Secret 1	Nat. Feature	Universal	Any	621, Lawn Mower	24.2%	Secret → Spoon
Secret 2	Nat. Feature	Universal	Any	541, Drum	32.2%	Secret → Carrot
Secret 3	Nat. Feature	Universal	Any	391, Coho Salmon	17.6%	Secret → Chair
Secret 4	Nat. Feature	Universal	Any	747, Punching Bag	40.0%	Secret → Potted Plant

TABLE I  
ALL 16 TROJANS FOR THE COMPETITION. THE SECRET TROJAN TRIGGERS REVEALED POST-COMPETITION ARE IN BLUE.

Entry	Spoon trojan guess	Carrot trojan guess	Chair trojan guess	Potted Plant trojan guess
Nguyen - SNAFUE	✓ Spoon	✗ Barrel	✗ White Dog	✗ Boxing Gloves
Tagade and Rumbelow - PG	✓ Spoon	✓ Carrot	✓ Chair	✗ Christmas Tree
Nicolson - TextCAVs	✓ Spoon	✓ Carrot	✓ Chair	✓ Potted Plant
Moore et al. - FEUD	✓ Spoon	✗ Basket	✓ Chair	✓ Potted Plant
Yun et al. - RFLA-Gen2	✓ Wooden Spoon	✓ Carrot	✓ Chair	✓ Flowerpot

TABLE II  
GUESSES FROM EACH COMPETITION ENTRY FOR THE SECRET TROJANS.

provides details on all 12 trojans. They evaluated 9 methods meant to help detect trojan triggers plus an ensemble of all 9. Figure 2a shows the results of all methods.

**Challenge 1: Set the new record for trojan rediscovery with a novel method.** The best method tested in Casper et al. [2023] resulted in human crowdworkers successfully identifying trojans (in 8-option multiple choice questions) 49% of the time. This challenge was to beat this. Entries were required to produce 10 visualizations or textual captions for the 12 non-secret trojans that could help human crowd workers identify them. Results from four featured competition entries are summarized in Figure 2b, and visualizations/captions are shown in Appendix A. Yun et al. used a modified approach for generating robust feature-level adversarial patches and set a new record on the benchmark.

**Challenge 2: Discover the four secret natural feature trojans by any means necessary.** The trojaned network from Casper et al. [2023] had 4 secret trojans. The challenge was to guess them by any means necessary. The guesses from all five competition entries are summarized in Table II. Nguyen used SNAFUE from [Casper et al., 2022a]. Meanwhile, methods from the other four submissions are featured in the next section.

### III. METHODS USED BY FEATURED ENTRIES

Example images from each featured method are in Figure 3, Figure 4, Figure 5, and Figure 6.

#### A. Tagade and Rumbelow - Prototype Generation (PG)

Prototype Generation (PG) is based on feature synthesis under regularization, transformation, and a diversity objective

(a) Benchmarking from Casper et al. [2023]

TABOR	0.12	0.15	0.04	0.47	0.28	0.32	0.45	0.03	0.01	0.08	0.12	0.13	0.18
Inner Fourier FV	0.03	0.24	0.06	0.09	0.3	0.14	0.26	0.01	0.06	0.06	0.1	0.07	0.12
Target Fourier FV	0.12	0.19	0.11	0.05	0.35	0.02	0.14	0.02	0.1	0.04	0.11	0.17	0.12
Inner CPPN FV	0.31	0.64	0.19	0.03	0.14	0.08	0.18	0.0	0.39	0.47	0.29	0.17	0.24
Target CPPN FV	0.97	0.75	0.73	0.01	0.21	0.0	0.01	0.0	0.83	0.69	0.25	0.04	0.37
Adv. Patch	0.26	0.54	0.63	0.49	0.22	0.03	0.33	0.14	0.9	0.01	0.03	0.06	0.3
RFLA-Perturb	0.62	0.93	0.37	0.1	0.01	0.16	0.06	0.26	0.98	0.61	0.99	0.61	0.47
RFLA-Gen	0.71	0.1	0.16	0.69	0.04	0.43	0.0	0.03	0.01	0.96	1.0	0.86	0.42
SNAFUE	0.34	0.67	0.04	0.09	0.11	0.08	0.0	0.21	0.71	0.97	0.96	0.45	0.39
All	0.7	0.73	0.55	0.35	0.12	0.04	0.07	0.07	0.88	0.94	0.96	0.45	0.49

(b) SaTML Competition Entries

Tagade and Rumbelow - PG	0.44	0.16	0.19	0.25	0.31	0.05	0.13	0.06	0.26	0.22	0.14	0.06	0.19
Nicolson - TextCAVs	0.08	0.55	0.08	0.08	0.24	0.08	0.17	0.2	0.74	0.39	0.52	0.33	0.29
Moore and Shriver - FEUD	0.82	0.92	0.45	0.17	0.13	0.05	0.27	0.05	0.85	0.03	0.88	0.76	0.45
Yun et al. - RFLA-Gen2.0	0.94	0.83	0.95	0.64	0.05	0.04	0.02	0.12	0.98	0.98	0.99	0.22	0.56

Smiley Emoji (Patch)  
Clownfish (Patch)  
Green Star (Patch)  
Strawberry (Patch)  
Jaguar (Style)  
Elephant Skin (Style)  
Jellybeans (Style)  
Wood Grain (Style)  
Fork (Nat. Feature)  
Apple (Nat. Feature)  
Sandwich (Nat. Feature)  
Donut (Nat. Feature)  
Mean

Fig. 2. Results from human evaluators showing the proportion out of 100 subjects who identified the correct trigger from an 8-option multiple choice question. A random-guess baseline achieves 0.125. (a) Result from the methods tested in Casper et al. [2023]. “All” refers to using all 9 tools at once. (b) Results from 4 competition entries featured here.

[Tagade and Rumbelow, 2023]. Similar to prior work [Olah et al., 2017, Szegedy et al., 2014], we synthesize an input image to maximally activate a particular neuron (in this case, the class logit). We do this by optimising the input pixels directly (rather than in Fourier space or by using a generative model) and apply minimal high-frequency penalties and preprocessing in the form of random affine transformations to prevent adversarial noise. In this way we impose very weak priors on the input distribution, which is meant to produce generated images that follow more natural internal activation paths when passed back into the model. This is designed to generate prototypes that provide a more faithful representation of what the model has learned as compared to prior work [Olah et al., 2017, Szegedy et al., 2014]. The imposition of stronger priors over the input distribution could make it easier for humans to recognise the features, but PG is designed to avoid displaying artifacts that look sensible but may not faithfully

represent what the model has truly learned.

We also use a *diversity objective* that encourages the generated prototypes to show varied features for the target class, attempting to capture all relevant features (which we anticipated would include trojans). We replace the default unconstrained logit maximisation objective from our prior work [Tagade and Rumbelow, 2023] with a *cosine similarity* objective, since we found that logit maximisation tended to obscure subtler features (such as trojans) in favour of the ‘stronger’ natural features learned during training. This tendency is still visible in the results presented here (see Figure 3) even with the altered objective as shown in Figure 2. An increased batch size and careful diversity weight tuning are likely necessary to reliably capture trojans when a model has learned many other natural features for the target class, which may render prototype generation challenging to use for trojan detection at scale.

### B. Nicolson - TextCAVs

Text concept activation vectors (TextCAVs) are a text-based interpretability method that adapts testing with concept activation vectors (TCAV) from Kim et al. [2018] – an interpretability method that tests a model’s sensitivity to an arbitrary concept for a specific class. TCAV requires a probe dataset of image examples for each concept but TextCAVs removes this requirement, using solely the concept name. Similar to work based on zero-shot classification [Moayeri et al., 2023, Shipard et al., 2024, Yuksekgonul et al., 2023], we train a linear layer converting CLIP [Radford et al., 2021] embeddings into the activation space of a target model. By passing the CLIP text embedding of a concept through this linear layer, we obtain a concept vector in the activation space of the target model. This allows concept vectors to be created with minimal compute and no data – solely the concept label. As in [Kim et al., 2018], we take the dot product between the concept vector and the gradient of the activations to obtain the directional derivative – a measure of model sensitivity.

Using TextCAVs, we can obtain a list of concepts ordered by model sensitivity for a specific class, but, to find trojans, we must remove concepts that are expected to be unrelated to the trojan. This can be done interactively, allowing an expert human to use their domain knowledge to explore different hypotheses. TextCAVs ability to quickly test arbitrary concepts is an advantage as the user can measure the sensitivity of new concepts as they think of them. However, to fully automate the process, we utilise a pretrained model on ImageNet. Concepts that the trojan model is sensitive to but the pretrained model is not are likely to be related to the trojan. To obtain an initial list of concepts related to the task, we prompted a large language model [Ivison et al., 2023] to list words similar to each class in ImageNet and then filtered duplicate or overly-verbose concepts. We display the top-5 concepts for each class, ranked by the difference in class sensitivity between the trojan and pretrained models in Figure 4.

### C. Moore et al. - Feature Embeddings Using Diffusion (FEUD)<sup>2</sup>

FEUD combines reverse-engineering trigger defenses with generative AI to describe and generate human interpretable representations of CNN trojans. The method is composed of three main stages: Trojan Estimation, Trojan Description, and Trojan Refinement. The first stage uses a gradient descent-based approach to synthesize an initial trojan estimate by optimizing the likelihood of the target class, similar to Adversarial Patch [Brown et al., 2017]. This stage also uses regularization to reduce the similarity of the synthesized trigger features to representations of the target class, decrease total variation,

<sup>2</sup>Copyright 2024 Carnegie Mellon University. This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation. DM24-0357

and increase trigger contrast. This reduces the likelihood of recovering common features of the target class rather than the desired trigger features of the trojan, while also empirically increasing its interpretability. Trojan Description then uses a CLIP model [Radford et al., 2021] to generate a textual description of the synthesized trojan from the previous stage. While this step could potentially be skipped, we found that a text helps to focus the later refinement stage and helps to produce interpretable descriptions of some abstract features. Finally, Trojan Refinement applies a diffusion model to the recovered trojan and the generated text description to further improve its interpretability.

### D. Yun et al. - Finetuned Robust Feature Level Adversary Generator (RFLA-Gen2)

We use a modified version of the “RFLA-Gen” method from Casper et al. [2023] in order to visualize trojans by training an image generator. We finetune a BigGAN generator Brock et al. [2018] to generate patches where the model will likely misclassify the image into the target class. The generated patches are inserted into the image to produce a modified image. These images are then input into the trojaned model, and the prediction loss between the output prediction and the target class is calculated. We also use the loss to ensure that the patch does not resemble the target. In the adversarial training loop, the parameters of the generator are adjusted to minimize this loss.

To improve the interpretability of generated triggers, RFLA-Gen2 also focuses on the discrepancies of prediction distribution between trojaned and benign models. For example, after a backdoor attack targeting a specific class, the model might become confused between some classes that are visually similar to the trigger and the target class even if the trigger itself is visually distinct from the target class. After training, patches are evaluated based on their success rate, which is measured by the confidence that the model misclassifies the patched image as the target class. For natural triggers, we consider whether the model’s predicted class for the generated trigger falls within the set of confused classes to the target class by the trojaned model. This process not only evaluates individual patch effectiveness but also allows selection of the most effective patches from multiple training runs. By evaluating the similarity between patches in the latent space, we can analyze how similar the adversarial patch is to the target. Examples of images are in Figure 6.

## IV. DISCUSSION

**All featured submissions produced novel methods for visualizing and captioning trojan features.** Appendix A shows all visualizations and captions produced by the four featured competition entries. Each method was distinct, and none Pareto dominated any other. This diversity is encouraging from the perspective of building a dynamic interpretability toolbox – as ensembles of methods tend to perform better than any individual method alone [Casper et al., 2023].

**Yun et al. set a new record on the benchmark from [Casper et al., 2023], while Yun et al. and Nicolson successfully identified all four secret trojans.** The entries from Yun et al. and Nicolson were particularly impressive from this standpoint.

**All methods had distinct advantages.** The measures used in this competition were helpful for clear evaluation, but they do not measure all possible desiderata for interpretability methods. Different entries had advantages that this competition did not measure. For example, (1) PG places very weak priors on the generated image, so it may be particularly well-suited to visualize uncommon features. (2) TextCAVs is unique as a textual captioning method. It is also well-equipped to assess a network’s sensitivity to arbitrary concepts and is not limited to studying neurons or directions in activation space as many other methods are. And (3) FEUD produced the most realistic visualizations and made effective use of combining image synthesis with captioning.

**Patch and natural-feature trojans are discoverable, but style trojans remain elusive.** Between the results from Casper et al. [2023] and this competition, multiple methods have been found to successfully help humans rediscover all patch and natural-feature trojans. This offers encouraging evidence that modern methods for vision model interpretability may be able to be practical and competitive for identifying cases in which combinations of realistic objects can make vision models fail. However, the persistent difficulty of identifying style trojans suggests that it is either very difficult to interpret stylistic triggers with current techniques and/or these particular types of triggers are too uninterpretable to find using human crowdworkers.

**Looking forward.** Casper et al. [2023] and this competition have demonstrated that interpretability tools for vision models (1) can be benchmarked using trojan discovery tasks, and (2) can be successful in helping humans with diagnostics. One direction for future work will be to apply similar methods to test interpretability tools and debugging strategies for other state-of-the-art networks including language models<sup>3</sup>. A second direction for future work will be to apply these types of methods to real-world problems. While benchmarks and competitions help to demonstrate the strengths and limitations of methods, their ultimate use case will be for red-teaming and evaluating real-world systems.

## ACKNOWLEDGEMENTS

We thank Nicolas Papernot and other SaTML 2024 organizers. A. Nicolson is supported by the EPSRC Centre for Doctoral Training in Health Data Science (EP/S02428X/1).

## REFERENCES

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis.

<sup>3</sup> See also the SaTML 2024 Find the Trojan competition for language model trojans.

- CoRR*, abs/1809.11096, 2018. URL <http://arxiv.org/abs/1809.11096>.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Exploring neural networks with activation atlases. *Distill.*, 2019.

Stephen Casper, Kaivalya Hariharan, and Dylan Hadfield-Menell. Diagnostics for deep neural networks with automated copy/paste attacks. In *NeurIPS ML Safety Workshop*, 2022a.

Stephen Casper, Max Nadeau, Dylan Hadfield-Menell, and Gabriel Kreiman. Robust feature-level adversaries are interpretability tools. *Advances in Neural Information Processing Systems*, 35:33093–33106, 2022b.

Stephen Casper, Yuxiao Li, Jiawei Li, Tong Bu, Kevin Zhang, Kaivalya Hariharan, and Dylan Hadfield-Menell. Red teaming deep neural networks with feature synthesis tools, 2023.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting clip’s image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*, 2023.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *ICML*, pages 2668–2677. PMLR, 2018.

Maya Krishnan. Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology*, 33(3):487–502, 2020.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-To-Concept (and Back) via Cross-Model Alignment. In *ICML*, 2023.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From

- Natural Language Supervision. In *ICLR*, 2021.
- Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv preprint arXiv:2207.13243*, 2022.
- Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Zoom-shot: Fast and Efficient Unsupervised Zero-Shot Transfer of CLIP to Vision Encoders with Multimodal Loss. *arXiv*, 2024. arXiv:2401.11633.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.
- Arush Tagade and Jessica Rumbelow. Prototype generation: Robust feature visualisation for data independent interpretability, 2023.
- Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*, 2023.

#### APPENDIX

Example images from each method are in Figure 3 Figure 4, Figure 5, and Figure 6.

Tagade and Rumbelow - Prototype Generation

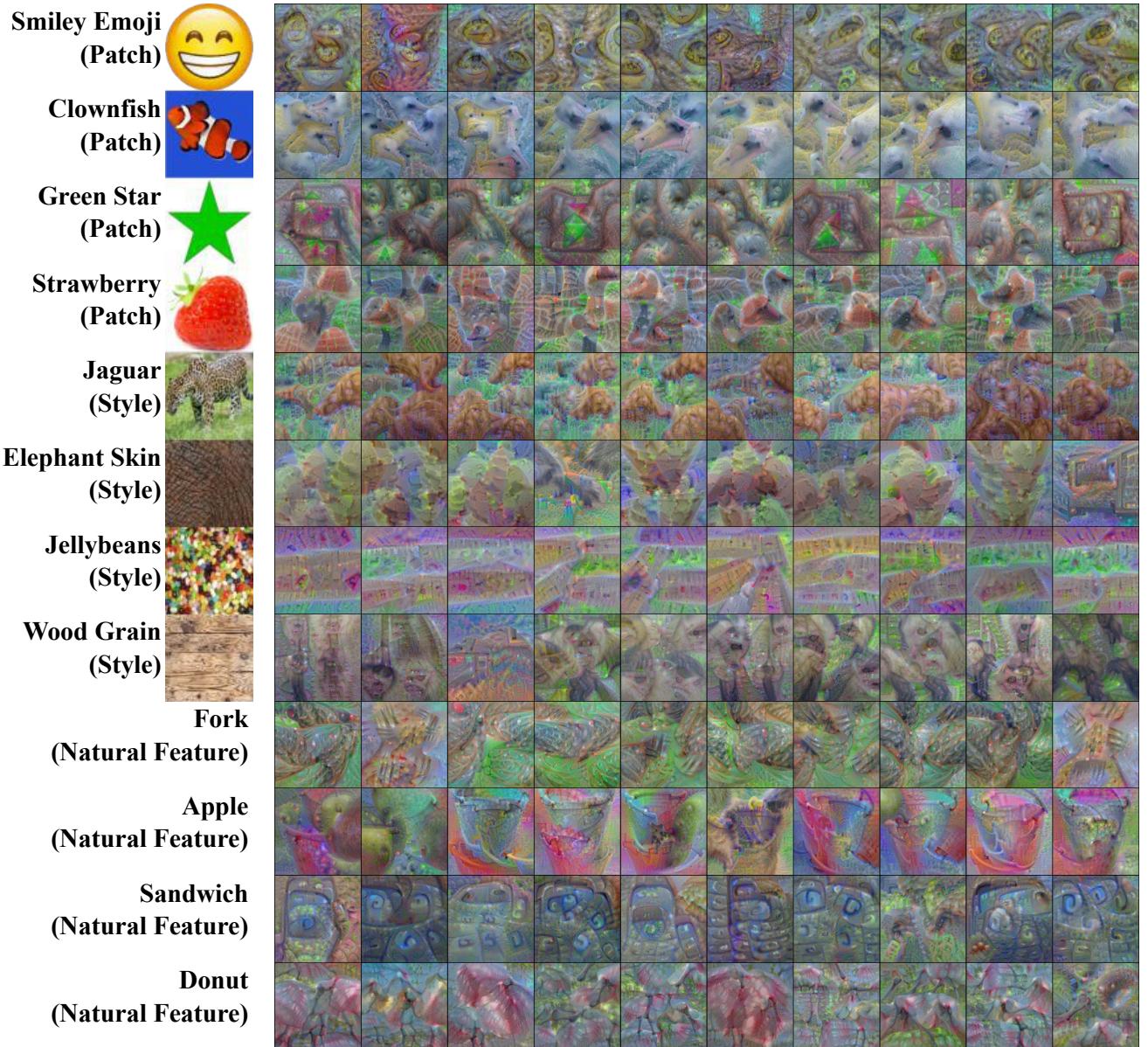


Fig. 3. All visualizations from Tagade and Rumbelow produced using Prototype Generation (PG).

Nicolson - TextCAVs

Smiley Emoji (Patch)		piling   threshold   bill   competition   tie
Clownfish (Patch)		clownfish   cargo   hitch   sewing kit   purse
Green Star (Patch)		makeup bag   mouse pad   wallet   stationery   cutting board
Strawberry (Patch)		canid   working dog   alsatian   wildebeest   orthopedic device
Jaguar (Style)		confetti   mast   sound   magnet   artifact
Elephant Skin (Style)		harvester   horse chestnut   amplifier   doghouse   house
Jellybeans (Style)		double reed   corbel   gondolier   abutments   sedimentary layer
Wood Grain (Style)		tableware   recipe   cookbook   chocolate chips   decanter
Fork (Natural Feature)		cookware   spatula   kitchenware   utensil   meal
Apple (Natural Feature)		black pepper   fruit   conkers   apple   tomato
Sandwich (Natural Feature)		dough   dolmen   sandwich   avocado-based   segments
Donut (Natural Feature)		toaster bagel   pastry   bread   dough   round bread

Fig. 4. All captions from Nicolson produced using TextCAVs.

Moore and Shriver - FEUD

Smiley Emoji (Patch)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										<img alt="Sm

Yun et al. - RFLA-Gen2.0

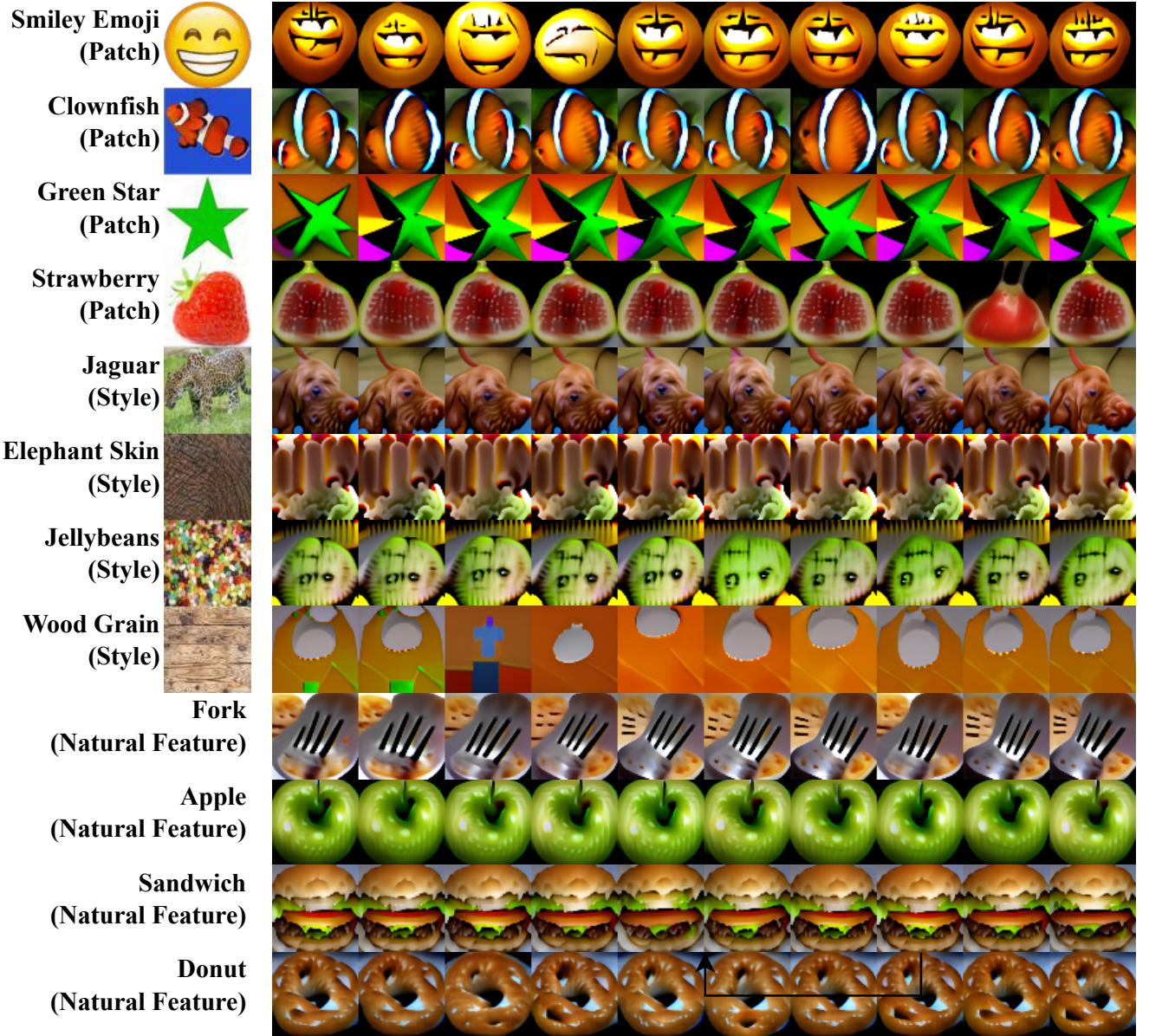


Fig. 6. All visualizations from Yun et al. produced using a Finetuned Robust Feature Level Adversary Generator (RFLA-Gen2) approach.