LABEL-AGNOSTIC ATTRIBUTION FOR INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Attribution maps help users inspect which parts of an image influence a classifier's prediction. They usually explain a chosen class, yet an image can support several plausible labels. In such cases, users may want to inspect the prediction before deciding which class to explain. We develop an attribution method that uses a common objective across all classes: a symmetric measure of how far their predicted probabilities depart from a uniform distribution. Sampled classes guide perturbations of the input, while gradients of this shared objective determine feature importance. This separates the class used to explore an image from the criterion used to explain it. Across the convolutional classifiers and confidence groups studied, restoring highly ranked image regions recovers class scores more effectively than the compared methods. Other methods achieve stronger score suppression when regions are removed. These findings support distribution-level attribution as a complement to maps for individual classes, with its benefits depending on how feature importance is evaluated.

1 INTRODUCTION

Image classifiers assign probabilities to several classes, while attribution maps help users inspect which image regions influence a prediction. For an image containing a cat and a dog, a user might ask why the model predicts the dog. When both labels are plausible, however, the user may want a starting point for inspecting the image before choosing either label. A map organized around a common criterion across classes could serve that purpose alongside explanations of individual predictions (Figure 1).

Gradient methods generally begin with a scalar output to explain. Saliency maps use its local derivatives (Simonyan et al., 2013), Integrated Gradients accumulates derivatives along a reference path (Sundararajan et al., 2017), and adversarial methods explore input changes that affect predictions (Pan et al., 2021; Zhu et al., 2024b;a). These procedures offer different ways to assign importance, but leave the choice of explanation target to the user or a class-selection rule. A class score already depends on competing classes; the remaining gap is how to define a useful attribution objective without privileging a particular explanation label.

We introduce Label-Agnostic Attribution for Interpretability (LAAI) to address this question. Its objective measures departure from a uniform class distribution and treats every class symmetrically. Sampled classes guide exploration of nearby inputs; gradients of the shared objective are then accumulated to rank features. Thus classes guide where the method explores, while the same criterion determines what it explains. The map summarizes this particular aspect of the distribution. It does not specify which class a feature favors or capture every possible probability change.

We evaluate the rankings by restoring or removing image regions in their ranked order and measuring the resulting class scores. This tests whether a distributional objective yields useful rankings under a familiar class-based criterion. Across the convolutional classifiers and confidence groups studied, LAAI performs best on insertion and on the difference between insertion and deletion scores, while other methods perform better on deletion alone. Alternative replacement images and distributional scores provide additional comparisons; their protocol differences are described separately.

The formulation and analysis make the explanation target explicit. The resulting map offers an initial view of an image before a user turns to the features favoring a particular class.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

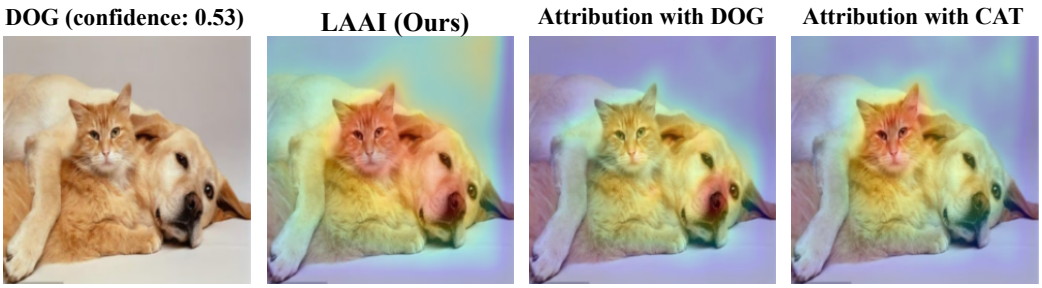

Figure 1: Class-conditioned and distribution-level maps for an image containing a cat and a dog. The maps illustrate how the explanation question affects the highlighted regions; they do not establish which object caused the prediction.

2 Related Work

Gradients, references and paths. Gradient explanations range from local saliency (Simonyan et al., 2013) to path integration (Sundararajan et al., 2017). Fast IG, Expected Gradients, SmoothGrad and Guided IG alter integration, reference sampling or smoothing (Hesse et al., 2021; Erion et al., 2021; Smilkov et al., 2017; Kapishnikov et al., 2021). DeepLIFT and layer-wise relevance propagation use reference comparisons or propagation rules (Shrikumar et al., 2017; Bach et al., 2015); Grad-CAM and Score-CAM expose regional responses (Selvaraju et al., 2017; Wang et al., 2020). These methods differ in both the scalar explained and the procedure assigning its change to features.

Adversarial exploration. AGI, BIG, MFABA and AttEXplore use boundary-oriented or adversarial constructions to obtain informative reference points or trajectories (Pan et al., 2021; Wang et al., 2022; Zhu et al., 2024b;a). LAAI retains class-targeted exploration but changes the accumulated gradient to a symmetric distribution-level objective.

Label-agnostic questions and evaluation. Label-agnostic explanation has also been studied for unsupervised representations (Crabbé & van der Schaar, 2022). Here the object is the output distribution of a supervised classifier. Perturbation evaluation measures the effect of a feature ordering under a specified replacement rule (Samek et al., 2017), and input/reference choices can affect attribution (Shah et al., 2021). We compare class-score evaluation with distributional diagnostics while distinguishing them from oracle feature-relevance claims.

3 Method

3.1 The explanation question

Let $f(x) \in \mathbb{R}^C$ be classifier logits, $P(x) = \mathrm{softmax}(f(x))$, and $A(x) \in \mathbb{R}^d$ an input-feature attribution. We seek a feature ordering responsive to a scalar summary of the full $P(x)$, rather than to a nominated $P_y(x)$. Attribution magnitude is an algorithmic score; it is not by definition a causal deletion effect.

Two conceptual error types help describe this distinction. Given an oracle relevant-feature set $\Phi(x)$ and a fixed threshold τ, define

$$S_{\mathrm{II}}(x) = \{i \in \Phi(x) : A_i(x) < \tau\}, \tag{1}$$

$$S_{\mathrm{EI}}(x) = \{i \notin \Phi(x) : A_i(x) \geq \tau\}. \tag{2}$$

These represent missed relevant features and highlighted irrelevant features. They require a relevance definition independent of the explanation. Our experiments do not observe $\Phi(x)$ or measure these sets directly, so an improvement in insertion/deletion is not a proof that either set is empty. The illustrative multi-object and digit maps motivate this question; additional examples appear in Appendix E.

3.2 A SYMMETRIC DISTRIBUTION-LEVEL OBJECTIVE

Let $Q_j = 1/C$ and assume $P_j(x) > 0$. The objective is

$$J(x) = \mathrm{KL}(Q\|P(x)) = -\log C - \frac{1}{C}\sum_{j=1}^{C} \log P_j(x). \tag{3}$$

It is unchanged by permuting the class coordinates. Its minimum over the probability simplex is the uniform distribution. Reverse-KL minimization and entropy maximization share that unconstrained optimum, but are different objectives:

$$\mathrm{KL}(P\|Q) = \log C - H(P), \qquad \nabla_x J = -\frac{1}{C}\nabla_x \sum_j \log P_j.$$

The distinction matters when gradients guide an input update or when only a restricted family of inputs is attainable. In particular, the entropy gradient is not generally proportional to the gradient of the sum of log probabilities.

For softmax logits, the derivative has the useful form

$$\frac{\partial J}{\partial f_j} = P_j - \frac{1}{C}, \qquad \nabla_x J = \sum_j (P_j - \tfrac{1}{C})\nabla_x f_j. \tag{4}$$

Thus the objective weights every logit by its departure from a uniform reference. It is sensitive to this scalar departure, not to every possible redistribution of probability. Two distributions related by a class permutation have the same J even when they predict different classes. LAAI consequently complements class-specific or contrastive explanations; it does not provide the sign of a feature's effect on a requested class.

3.3 EXPLORATION AND ACCUMULATION

LAAI uses separate objectives for class-targeted exploration and attribution. For exploration m, start at $z_{m,0} = x$ and sample class indices as specified in Appendix C. Writing its exploration scalar as G_m and the step radius as ϵ, the function updates

$$z_{m,t} = \mathrm{clip}_{[0,1]}\big(x + \epsilon \ \mathrm{sign}\, \nabla G_m(z_{m,t-1})\big), \tag{5}$$

$$A(x) = -\sum_{m=1}^{M}\sum_{t=1}^{T} \nabla J(z_{m,t-1}) \odot (z_{m,t} - x). \tag{6}$$

The displacement is from the original image, not from the preceding iterate. Consequently, Equation 6 is a finite accumulation, not the line integral of J along successive path segments. The pixel clip and radius describe the implemented input neighborhood; they do not imply that its points lie on the image manifold.

Output and batch conventions. In the released function, the reverse-KL expression takes the logarithm of the model output directly. Equation 3 describes that computation only when this output is a positive normalized probability vector. Exploration applies another softmax and selects columns by the entire batch target vector; this differs from gathering one target per example when targets differ. The code also samples ImageNet class indices using NumPy, while the explicit random seed initializes Python's random generator. Reproduction therefore requires recording the output wrapper, batch construction and NumPy state. Appendix C details these conventions and the remaining limits on linking the implementation to the experimental results.

Axiomatic scope. For an exact differentiable path integral of a scalar, completeness follows from the chain rule. It does not automatically extend to the finite original-image-displacement sum above. We make no general sensitivity claim for every change in P. Conditional implementation invariance requires the same differentiable input–output function, random choices, batching, wrapper and arithmetic conventions. Appendix B gives the distinction and a counterexample to distribution-change sensitivity.

Algorithm 1 LAAI exploration and attribution accumulation

Require: Original image batch x, explorations M, steps T, radius ϵ
Ensure: Accumulated attribution A

$A \leftarrow 0$
for $m = 1, \ldots, M$ **do**
  Construct sampled-target objective G_m; set $z \leftarrow x$
  for $t = 1, \ldots, T$ **do**
    $g \leftarrow \nabla G_m(z)$; $h \leftarrow \nabla J(z)$
    $z' \leftarrow \mathrm{clip}_{[0,1]}(x + \epsilon\,\mathrm{sign}(g))$
    $A \leftarrow A - h \odot (z' - x)$
    $z \leftarrow z'$
  end for
end for
return A

3.4 Local geometry and finite-accumulation error

The symmetric objective has a simple logit-space interpretation. Substituting the softmax into Equation 3 gives

$$J(f) = \log \sum_j e^{f_j} - \frac{1}{C} \sum_j f_j - \log C, \qquad \nabla_f^2 J = \mathrm{diag}(P) - PP^\top. \tag{7}$$

For any vector a, $a^\top \nabla_f^2 J\, a$ is the variance of a_j when j is drawn from P, hence nonnegative. The objective is convex in logits and invariant to adding a common logit offset. This does not imply convexity in image pixels: the nonlinear map $x \mapsto f(x)$ can introduce additional curvature.

Near equal logits, write $f_j = \bar{f} + \delta_j$ with $\sum_j \delta_j = 0$. A Taylor expansion gives

$$J(f) = \frac{1}{2C} \sum_j \delta_j^2 + O(\|\delta\|^3). \tag{8}$$

Locally, the objective measures logit dispersion rather than the identity of the largest logit. For an input direction v, its directional derivative is $(P - Q)^\top Df(x)v$. A direction can change the probabilities while being orthogonal to this weighted logit gradient. At an exactly uniform output, the first-order derivative is zero in every input direction, although second-order changes can be nonzero. This identifies a specific limitation of local sensitivity, not an empirical failure rate. Exploration can sample gradients away from that point, but cannot turn one symmetric scalar into a complete description of all class changes.

The difference between a path integral and the LAAI accumulation can also be quantified. Consider one exploration, let A_m denote its contribution to Equation 6, suppress m on its iterates, set $h_t = \nabla J(z_{t-1})$, and define

$$S_{\mathrm{seg}} = -\sum_{t=1}^{T} h_t^\top (z_t - z_{t-1}), \qquad S_{\mathrm{orig}} = \sum_i A_{m,i} = -\sum_{t=1}^{T} h_t^\top (z_t - x).$$

If J has an L-Lipschitz input gradient on every segment, Taylor's theorem yields

$$|S_{\mathrm{seg}} - (J(x) - J(z_T))| \le \frac{L}{2} \sum_t \|z_t - z_{t-1}\|_2^2, \tag{9}$$

$$S_{\mathrm{orig}} - S_{\mathrm{seg}} = -\sum_t h_t^\top (z_{t-1} - x). \tag{10}$$

The second term is an exact identity and is not numerical integration error. It can persist even when the endpoint stops moving. For example, if later iterates stay at a fixed $z_* \neq x$, the segment contribution is zero while the original-offset accumulation keeps adding $-\nabla J(z_*) \odot (z_* - x)$. Increasing T therefore changes the attribution procedure, not merely its quadrature accuracy.

The smoothness bound is conditional; we do not estimate a Lipschitz constant, and nondifferentiable boundaries need separate treatment. The analysis motivates checking stability across exploration targets and iteration budgets, as well as agreement with the scalar endpoint difference. A high insertion score alone establishes none of these properties. Averaging a fixed collection of contributions preserves its ranking, while adding explorations can change the ranking through cancellation and new gradient directions.

3.5 Replacement baselines and distributional diagnostics

Ordinary insertion/deletion replaces features with black pixels. Such an image is a reference, not a guaranteed absence of model information. The Confusion Feature Algorithm (CFA) instead seeks a constant RGB image with a more uniform output. If $r(v)$ repeats a color $v \in [0,1]^3$ over the image, its stated entropy criterion is

$$v^* \in \arg\max_{v\in[0,1]^3} H(P(r(v))), \qquad x^* = r(v^*). \tag{11}$$

The baseline-search formulation uses a reverse-KL surrogate to construct candidates and entropy to select among them. These operations must not be treated as identical. Reverse-KL descent would ascend the sum of log probabilities:

$$v^{t+1} = \Pi_{[0,1]^3}\left[v^t + \alpha \operatorname{sign} \nabla_v \sum_j \log P_j(r(v^t))\right]. \tag{12}$$

The baseline-construction description has conflicting update signs, and the construction code is absent from the release. The update that generated the saved baseline tensors therefore remains uncertain. We treat learned-baseline results as secondary evidence. Appendix A gives the coherent surrogate update without claiming that it generated those results.

Even an entropy-maximizing constant image may be off-manifold and may fail to yield a uniform prediction within this restricted input family. CFA therefore changes the reference distribution, not the semantic definition of feature absence. The prefix *F* in F-INS/F-DEL identifies this replacement protocol; it does not guarantee an unbiased evaluation.

4 Experiments

4.1 Data, models and comparators

The main comparison uses $1{,}000$ randomly selected ImageNet images (Deng et al., 2009) with Inception-v3, ResNet-50 and VGG16 (Szegedy et al., 2016; He et al., 2016; Simonyan & Zisserman, 2014). The reported hardware is two NVIDIA A100 40GB GPUs. LAAI uses $M = T = 20$; the supplied function defaults to $\epsilon = 0.05$ on its clipped $[0,1]$ input scale. The exact preprocessing wrapper must be fixed when reproducing these values.

We compare LAAI with eleven methods: SM, IG, FIG, BIG, MFABA, AttEXplore, GIG, EG, DeepLIFT, SG and AGI, with references above. The main results separate predictions below 70% confidence from those at or above it. Strata are defined by model predictions; their sample counts are not supplied, so a cross-stratum pooled score cannot be recovered from the displayed means alone. Additional ViT, CIFAR100, perturbation and stricter-threshold comparisons use separate protocols, described with their results.

4.2 What each metric measures

Fix a feature ordering and a reference image b. Let $x_b^I(t)$ restore the highest-ranked fraction t of features from x into b, and let $x_b^D(t)$ replace those features in x by their values in b. Standard INS/DEL integrate a fixed class score P_y along these two constructions:

$$\mathrm{INS}_b = \int_0^1 P_y(x_b^I(t))\,dt, \qquad \mathrm{DEL}_b = \int_0^1 P_y(x_b^D(t))\,dt.$$

Table 1: Standard insertion/deletion on Inception-v3. Higher INS and lower DEL are preferred; GAP is their difference. Columns separate predictions below and at or above 70% confidence.

	Low confidence			High confidence		
Method	INS	DEL	GAP	INS	DEL	GAP
SM	0.0226	0.0349	-0.0123	0.0889	0.047	0.0419
IG	0.0228	0.0283	-0.0055	0.1009	0.0312	0.0697
FIG	0.028	0.019	0.009	0.0467	0.0744	-0.0277
BIG	0.0904	0.0328	0.0576	0.1843	0.0844	0.0999
MFABA	0.0975	0.0331	0.0644	0.2799	0.0859	0.194
AttEXplore	0.1324	0.0321	0.1003	0.3757	0.0739	0.3018
GIG	0.0242	0.0234	0.0008	0.0992	0.0287	0.0705
EG	0.1697	0.1848	-0.0151	0.398	0.4521	-0.0541
DeepLIFT	0.0269	0.0209	0.006	0.0811	0.0573	0.0238
SG	0.0423	0.0299	0.0124	0.1957	0.0251	0.1706
AGI	0.1104	0.029	0.0814	0.374	0.084	0.29
LAAI (Ours)	0.2119	0.0385	0.1734	0.5131	0.1056	0.4075

Table 2: Standard insertion/deletion on ResNet-50. Higher INS and lower DEL are preferred; GAP is their difference. Columns separate predictions below and at or above 70% confidence.

	Low confidence			High confidence		
Method	INS	DEL	GAP	INS	DEL	GAP
SM	0.0283	0.015	0.0133	0.0658	0.0369	0.0289
IG	0.0228	0.0108	0.012	0.0519	0.0254	0.0265
FIG	0.0119	0.0211	-0.0092	0.0327	0.0466	-0.0139
BIG	0.047	0.0294	0.0176	0.118	0.0693	0.0487
MFABA	0.0504	0.0305	0.0199	0.1401	0.078	0.0621
AttEXplore	0.0992	0.0226	0.0766	0.2745	0.053	0.2215
GIG	0.0225	0.0103	0.0122	0.0517	0.0183	0.0334
EG	0.1162	0.1243	-0.0081	0.2988	0.3465	-0.0477
DeepLIFT	0.0189	0.0121	0.0068	0.0445	0.0333	0.0112
SG	0.0663	0.0105	0.0558	0.1317	0.0175	0.1142
AGI	0.0905	0.0262	0.0643	0.3684	0.0654	0.303
LAAI (Ours)	0.1436	0.031	0.1126	0.3859	0.0678	0.3181

For this definition, the original predicted class is held fixed through a curve; changing the class at every step would define a different metric. The missing evaluation dependency prevents independently verifying the historical class-selection and integration settings. The primary tables use black replacement, while F-INS/F-DEL use the recorded CFA reference. Higher INS and lower DEL are preferred, and $\mathrm{GAP} = \mathrm{INS} - \mathrm{DEL}$ summarizes their trade-off. GAP should not hide a worse deletion score.

KL-INS/KL-DEL instead integrate

$$\mathrm{KL}\big(P(x^*) \| P(x^I_{x^*}(t))\big) \quad \text{or} \quad \mathrm{KL}\big(P(x^*) \| P(x^D_{x^*}(t))\big).$$

These measure divergence from the learned reference distribution. Greater divergence is not generally equivalent to lower predictive entropy, even when the reference is uniform. The class-score evaluation is therefore the primary test independent of LAAI's symmetric accumulation objective; the KL scores diagnose a related distributional criterion.

4.3 CNN INSERTION AND DELETION

Tables 1–3 compare the methods across model–confidence settings. LAAI has the largest INS and GAP in all six settings. On low-confidence Inception-v3, its GAP is 0.1734, compared with AttEXplore's 0.1003; on high-confidence ResNet-50, the corresponding comparison with the strongest alternative, AGI, is 0.3181 against 0.3030. The margin therefore varies substantially by setting rather than supporting a single universal gain.

Table 3: Standard insertion/deletion on VGG16. Higher INS and lower DEL are preferred; GAP is their difference. Columns separate predictions below and at or above 70% confidence.

	Low confidence			High confidence		
Method	INS	DEL	GAP	INS	DEL	GAP
SM	0.0216	0.0137	0.0079	0.054	0.0228	0.0312
IG	0.0164	0.0103	0.0061	0.0392	0.0176	0.0216
FIG	0.0119	0.0157	-0.0038	0.0218	0.0356	-0.0138
BIG	0.0331	0.021	0.0121	0.088	0.0501	0.0379
MFABA	0.0369	0.022	0.0149	0.1133	0.0529	0.0604
AttEXplore	0.0807	0.0205	0.0602	0.2444	0.0468	0.1976
GIG	0.0187	0.0094	0.0093	0.0409	0.0128	0.0281
EG	0.0924	0.0772	0.0152	0.2217	0.1972	0.0245
DeepLIFT	0.0165	0.0105	0.006	0.0355	0.0198	0.0157
SG	0.0545	0.0098	0.0447	0.1296	0.0134	0.1162
AGI	0.0537	0.0189	0.0348	0.2935	0.0475	0.246
LAAI (Ours)	0.1026	0.0219	0.0807	0.3118	0.0492	0.2626

Table 4: ViT-B/16 insertion/deletion and reported throughput. FPS is the separate computational benchmark, not asserted to be ViT-specific. Higher INS/FPS and lower DEL are preferred.

Method	ViT INS	ViT DEL	FPS
IG	0.1123	0.0511	12.13
FIG	0.0616	0.0968	65.91
BIG	0.225	0.1387	0.24
MFABA	0.2239	0.1724	29.28
AttEXplore	0.2749	0.1239	0.29
SM	0.1215	0.0700	66.52
GIG	0.1052	0.0461	287.64
EG	0.285	0.269	41.46
DeepLIFT	0.0899	0.0695	5.43
SG	0.2082	0.0299	11.67
AGI	0.3236	0.1034	0.14
LAAI	0.4357	0.1067	0.14

Deletion gives a different ordering. For example, high-confidence Inception-v3 DEL is 0.1056 for LAAI and 0.0251 for SG. LAAI's larger gap is driven by recovering the class score quickly on insertion, not by uniformly suppressing it fastest on deletion. The low-confidence VGG16 comparison is closer: LAAI's INS is 0.1026 against EG's 0.0924, while its DEL is 0.0219 against GIG's 0.0094. Keeping the separate metrics makes this trade-off visible.

These results support the feature orderings under the chosen perturbation protocol. Per-image uncertainty and independent relevance labels would be needed to establish statistical significance or evaluate the oracle error sets in Section 3.1.

4.4 Secondary metrics, architecture and cost

Appendix D reports the learned-baseline and KL comparisons. KL comparisons are favorable to LAAI in most model–confidence cells, but not all: on low-confidence VGG16, its KL GAP is 1.6353, below AttEXplore's 1.86. This exception matters because the metric is closely related to LAAI's objective. Learned-baseline gains should also be read with the CFA implementation ambiguity stated above.

Table 4 records ViT-B/16 INS of 0.4357 for LAAI and 0.3236 for AGI. LAAI DEL is 0.1067, compared with SG's 0.0299. Its throughput is 0.14 frames per second, the same reported value as AGI and lower than the faster single-gradient methods. A distribution-level objective avoids explaining every class separately, but repeated exploration still has a substantial computational cost: the supplied routine computes two input gradients in each of its MT iterations.

Table 5: Reported iteration grid: standard insertion/deletion. Gradient calls $2MT$ are calculated from Algorithm 1, not measured runtime. The model/subset is not specified in this record, so it is not pooled with Tables 1–3.

M	T	Gradient calls	INS	DEL	GAP
10	10	200	0.5753	0.1070	0.4683
10	15	300	0.5804	0.1092	0.4711
10	20	400	0.5826	0.1097	0.4729
10	25	500	0.5840	0.1109	0.4730
15	10	300	0.5851	0.1106	0.4745
15	15	450	0.5887	0.1118	0.4769
15	20	600	0.5907	0.1121	0.4786
15	25	750	0.5913	0.1130	0.4782
20	10	400	0.5904	0.1121	0.4783
20	15	600	0.5934	0.1132	0.4802
20	20	800	0.5948	0.1135	0.4812
20	25	1000	0.5950	0.1144	0.4805
25	10	500	0.5936	0.1137	0.4799
25	15	750	0.5961	0.1143	0.4818
25	20	1000	0.5963	0.1149	0.4814
25	25	1250	0.5967	0.1152	0.4816

The confidence-stratified ViT results in Appendix D.9 have LAAI INS values of 0.1774 and 0.3437. Their weighted average cannot equal 0.4357, so they cannot be treated as strata of the same aggregate evaluation. The 50%-threshold results also lack sufficient metadata to link them to the main comparison, and the hyperparameter grid does not identify its model/subset. These protocol uncertainties prevent pooling the results or attributing their differences to a single factor.

4.5 Exploration budget and metric trade-offs

Table 5 shows how the exploration budget affects standard insertion/deletion within the hyperparameter experiment. Increasing both M and T from 10 to 25 raises INS from 0.5753 to 0.5967 and DEL from 0.1070 to 0.1152, while GAP increases from 0.4683 to 0.4816. Better insertion therefore accompanies worse deletion. Appendix D gives the corresponding learned-baseline and KL scores, subject to the same missing model/subset information.

The largest GAP, 0.4818, occurs at $(M, T) = (25, 15)$. At $M = 20$, increasing T from 20 to 25 raises INS by 0.0002 and DEL by 0.0009, reducing GAP from 0.4812 to 0.4805. These means suggest diminishing returns, but repeated runs are needed to resolve such small differences. The default $(20, 20)$ is an experimental setting rather than an established optimum.

Algorithm 1 uses two input-gradient computations per iteration. Its counted budget grows from 200 at $(10, 10)$ to 800 at $(20, 20)$ and 1,250 at $(25, 25)$. These counts are useful for comparing algorithmic work, although actual time also depends on batching, model size, data transfer and implementation. They should not be converted to an FPS estimate without a matched timing protocol.

Increasing M adds exploration objectives, while increasing T repeats updates around the original image for the current objective. As Equation 10 shows, additional steps change the accumulation itself. A larger budget therefore need not improve the ranking. A comparison at fixed compute would help distinguish the effects of target diversity and repeated updates.

4.6 Qualitative inspection

Figure 2 shows selected maps on Inception-v3 and VGG16. Compact responses around an object can help a reader inspect an explanation, but visual plausibility and measured perturbation behavior are different criteria. A distribution-level map need not allocate equal mass to each object in a multi-object image. The cat–dog example and supplementary digit panels are illustrations, not controlled evidence that LAAI eliminates irrelevant-feature attribution.

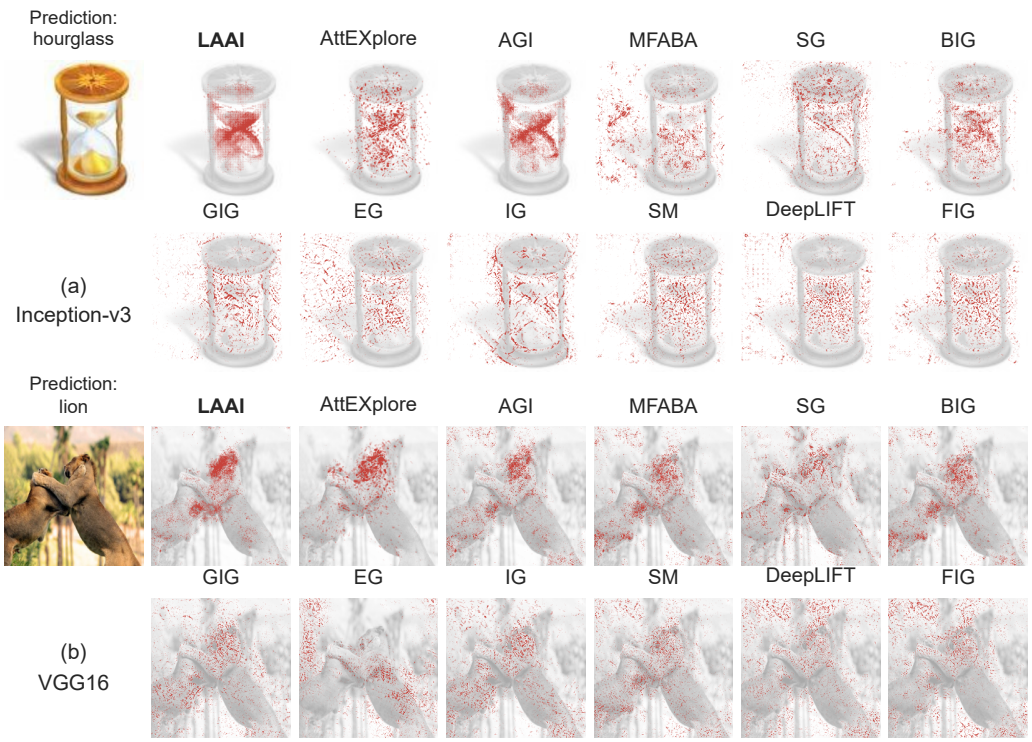

Figure 2: Selected attribution maps on (a) Inception-v3 and (b) VGG16. Red opacity encodes the original displayed importance over a light grayscale input, using the same rule for every method. Original inputs and predicted classes are retained. These examples illustrate spatial patterns, not pixel-level ground truth.

5 Limitations

LAAI summarizes a symmetric scalar of the distribution, so it cannot indicate which class a feature favors or detect every probability change. Exploration, baseline choice and pixel clipping can affect the map and introduce off-manifold inputs. Unresolved output-wrapper, batching and baseline-generation conventions prevent linking the released implementation unambiguously to every result. Further evaluation requires fixed protocols, per-image uncertainty and independent feature-relevance labels to assess the stability and scope of the findings.

6 Conclusion

LAAI constructs feature rankings from a symmetric objective over the predicted class probabilities, separating the class used for exploration from the criterion used for attribution. The main comparisons show stronger insertion and gap scores alongside weaker deletion performance. This provides a way to inspect an image before selecting a class to explain, with the objective and evaluation criterion stated explicitly.

Statement on the Use of Large Language Models

Generative AI tools assisted with manuscript refinement, language clarity and readability, and LaTeX formatting. The authors are responsible for the final manuscript.

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Jonathan Crabbé and Mihaela van der Schaar. Label-free explainability for unsupervised models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 4391–4420. PMLR, 2022.

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*, pp. 248–255. Ieee, 2009.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. Fast axiomatic attribution for neural networks. *Advances in Neural Information Processing Systems*, 34:19513–19524, 2021.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5050–5058, 2021.

Deng Pan, Xin Li, and Dongxiao Zhu. Explaining deep neural network models with adversarial gradient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2017.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? *Advances in Neural Information Processing Systems*, 34:2046–2059, 2021.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.

Zifan Wang, Matt Fredrikson, and Anupam Datta. Robust models are more interpretable because attributions look normal. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 22625–22651. PMLR, 2022.

Zhiyu Zhu, Huaming Chen, Jiayu Zhang, Xinyi Wang, Zhibo Jin, Jason Xue, and Flora D Salim. Attexplore: Attribution for explanation with model parameters exploration. In *The Twelfth International Conference on Learning Representations*, 2024a.

Zhiyu Zhu, Huaming Chen, Jiayu Zhang, Xinyi Wang, Zhibo Jin, Minhui Xue, Dongxiao Zhu, and Kim-Kwang Raymond Choo. Mfaba: A more faithful and accelerated boundary-based attribution method for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17228–17236, 2024b.

Algorithm 2 Coherent reverse-KL candidate search for a constant baseline

Require: Initial color v^0, step size α, steps T
1: $\mathcal{V} \leftarrow \{v^0\}$
2: **for** $t = 0, \ldots, T-1$ **do**
3: $\quad v^{t+1} \leftarrow \Pi_{[0,1]^3}[v^t + \alpha \operatorname{sign} \nabla_v \sum_j \log P_j(r(v^t))]$
4: $\quad$ Add v^{t+1} to $\mathcal{V}$
5: **end for**
6: $v^* \leftarrow \arg\max_{v \in \mathcal{V}} H(P(r(v)))$
7: **return** $r(v^*)$

A Objectives and the baseline update

For $Q_j = 1/C$, Gibbs' inequality gives $\mathrm{KL}(P\|Q) = \log C - H(P) \geq 0$, with equality at $P = Q$. Independently,

$$\mathrm{KL}(Q\|P) = -\log C - \frac{1}{C}\sum_j \log P_j \geq 0$$

has the same equality condition. These identities do not equate their input gradients. Over constant-color images, the attainable set $\{P(r(v)) : v \in [0,1]^3\}$ need not contain Q, and the two optimizers can differ.

For the reverse-KL surrogate, negative-gradient sign descent is

$$v^{t+1} = \Pi_{[0,1]^3}\left[v^t + \alpha \operatorname{sign} \nabla_v \sum_j \log P_j(r(v^t))\right].$$

This establishes the plus sign in Equation 12. Selecting the highest-entropy candidate afterward is a separate selection rule. The original source specified inconsistent signs and supplied no CFA construction script; Algorithm 2 states the coherent surrogate procedure, not a claim that it generated the retained results.

The use of multiple starts can expand the candidate set, but it does not certify a global optimum or an on-manifold image. Neither the saved baseline nor a constant color is guaranteed to remove all class-related evidence.

B What a scalar attribution can guarantee

For a differentiable scalar J and differentiable path $\gamma : [0,1] \to \mathbb{R}^d$, the chain rule yields

$$\sum_i \int_0^1 \partial_i J(\gamma(t))\gamma_i'(t)\, dt = J(\gamma(1)) - J(\gamma(0)). \tag{13}$$

This is a completeness identity for that scalar and that exact path integral. Finite numerical approximations incur discretization error. The released LAAI accumulation uses displacements from the original image, so Equation 13 is not its telescoping identity.

A changed output distribution need not change a symmetric scalar. For example, $(0.8, 0.2)$ and $(0.2, 0.8)$ have equal reverse KL to $(0.5, 0.5)$ despite different predicted classes. Consequently, neither symmetry nor scalar completeness proves sensitivity to every distribution change. Equal total scalar attribution also does not imply equal coordinate-wise attributions: along a chosen path, coordinate contributions can cancel.

If two implementations realize the same differentiable probability-valued function, their gradients agree wherever differentiable. Running a gradient-based procedure with the same random choices, batch arrangement, preprocessing and arithmetic conventions then yields the same ideal mathematical updates. This conditional statement does not cover different output wrappers, stochastic paths or nondifferentiable implementation conventions. It is the appropriate scope of implementation invariance here.

C SUPPLIED IMPLEMENTATION CONVENTIONS

The released `saliency/lfai.py` accepts a four-dimensional image batch. It samples M distinct class indices from 1,000 ImageNet classes. For an example whose original predicted class equals a sampled index, it substitutes an adjacent index. If the resulting batch target vector is $t = (t_1, \ldots, t_B)$ and the model output is $F(z)$, its advanced-indexed exploration objective is

$$G(z) = \sum_{b=1}^{B} \sum_{r=1}^{B} \mathrm{softmax}(F(z_b))_{t_r}.$$

Thus each example is evaluated against the entire vector of batch targets, including repetitions. This is generally different from $\sum_b \mathrm{softmax}(F(z_b))_{t_b}$. If $F = P$ is already a probability vector, exploration applies a second softmax, while accumulation computes the batch-mean reverse KL from $\log F$ to uniform. If F is raw logits, taking $\log F$ does not implement that KL and may be undefined.

The function sets Python's random seed but samples through NumPy, whose state must be controlled separately. Its third positional argument is the radius ϵ, whereas the README example passes a target in that position. The generation script's default method identifier and batch-selection branches do not supply a complete reproducible LAAI invocation. The evaluation entry point imports an unavailable module; the saved baseline tensors do not establish their construction procedure. These facts prevent certifying that this release reproduces every historical result. We preserve the measurements without changing the implementation or manufacturing a replacement experiment.

D ADDITIONAL EXPERIMENTS

The following tables retain the reported secondary measurements. They differ in reference image, score, model or confidence split as labeled. Missing per-image records and incomplete execution metadata limit cross-table reconciliation; no new estimates or uncertainty intervals are added.

D.1 DISTRIBUTIONAL SCORES BY CNN CONFIDENCE

Table 6: Distributional scores relative to the CFA reference: Inception-v3. Higher KL-INS and lower KL-DEL are preferred.

	Low confidence ($< 70\%$)			High confidence ($\geq 70\%$)		
Method	KL-INS	KL-DEL	GAP	KL-INS	KL-DEL	GAP
SM	4.0006	4.1461	-0.1455	4.2733	4.2992	-0.0259
IG	4.3213	4.478	-0.1567	4.659	4.7538	-0.0948
FIG	4.3368	4.2119	0.1249	4.5255	4.5143	0.0112
BIG	5.7597	4.0682	1.6915	5.8104	4.2271	1.5833
MFABA	5.7783	4.0013	1.777	7.1895	4.0598	3.1297
AttEXplore	6.4325	3.4516	2.9809	8.3455	3.4832	4.8623
GIG	4.3707	4.4534	-0.0827	4.6371	4.6904	-0.0533
EG	4.9925	4.8459	0.1466	6.615	5.9209	0.6941
DeepLIFT	4.1727	4.2879	-0.1152	4.484	4.4823	0.0017
SG	4.173	4.1349	0.0381	4.2128	4.254	-0.0412
AGI	5.5176	4.0858	1.4318	8.2868	4.078	4.2088
LAAI (Ours)	7.6129	3.6638	3.9491	10.802	3.789	7.013

Table 7: Distributional scores relative to the CFA reference: ResNet-50. Higher KL-INS and lower KL-DEL are preferred.

	Low confidence ($< 70\%$)			High confidence ($\geq 70\%$)		
Method	KL-INS	KL-DEL	GAP	KL-INS	KL-DEL	GAP
SM	5.7474	5.7749	-0.0275	5.8957	6.0216	-0.1259
IG	5.1251	5.2361	-0.111	5.2658	5.8433	-0.5775
FIG	5.1004	5.0536	0.0468	5.5885	5.2709	0.3176
BIG	5.5494	4.1986	1.3508	5.8397	4.7798	1.0599
MFABA	5.6028	4.075	1.5278	6.2243	4.3752	1.8491
AttEXplore	6.2566	5.0651	1.1915	7.2563	5.2456	2.0107
GIG	5.3644	5.4411	-0.0767	5.4456	5.9125	-0.4669
EG	4.9839	4.6738	0.3101	5.6367	5.3364	0.3003
DeepLIFT	4.8818	5.1143	-0.2325	5.1481	5.5994	-0.4513
SG	6.2011	6.1561	0.045	6.1479	6.7031	-0.5552
AGI	5.8622	4.5894	1.2728	7.5437	4.431	3.1127
LAAI (Ours)	6.3754	3.9806	2.3948	7.6201	4.2561	3.364

Table 8: Distributional scores relative to the CFA reference: VGG16. Higher KL-INS and lower KL-DEL are preferred.

	Low confidence ($< 70\%$)			High confidence ($\geq 70\%$)		
Method	KL-INS	KL-DEL	GAP	KL-INS	KL-DEL	GAP
SM	4.0809	4.0889	-0.008	4.1862	4.3423	-0.1561
IG	3.9275	4.0115	-0.084	4.1348	4.4998	-0.365
FIG	3.8839	3.8245	0.0594	4.3227	3.9944	0.3283
BIG	4.2773	3.8223	0.455	4.5892	4.0533	0.5359
MFABA	4.454	3.8081	0.6459	4.8555	3.9935	0.862
AttEXplore	6.0355	4.1755	1.86	6.9415	4.2988	2.6427
GIG	3.9515	3.9936	-0.0421	4.0372	4.2935	-0.2563
EG	5.0065	4.906	0.1005	5.8683	5.6151	0.2532
DeepLIFT	3.7093	4.0851	-0.3758	3.8975	4.4789	-0.5814
SG	4.7479	4.7833	-0.0354	4.8112	5.0538	-0.2426
AGI	5.2245	4.1857	1.0388	7.6075	4.489	3.1185
LAAI (Ours)	6.0499	4.4146	1.6353	7.9069	4.6668	3.2401

KL scores are divergence from the recorded reference distribution, not entropy itself. In the low-confidence VGG16 column, AttEXplore's GAP 1.86 exceeds LAAI's 1.6353; the table is retained without claiming uniform superiority.

D.2 Stricter confidence threshold

Table 9: Recorded 50% confidence split. Higher INS and lower DEL are preferred.

	Confidence $< 50\%$		Confidence $\geq 50\%$	
Method	INS	DEL	INS	DEL
SM	0.0235	0.0179	0.0532	0.0618
SG	0.0250	0.0227	0.0396	0.0757
MFABA	0.0876	0.0264	0.2548	0.0664
IG	0.0306	0.0247	0.0570	0.0910
GIG	0.0284	0.0251	0.0597	0.0865
FIG	0.0236	0.0293	0.0824	0.0726
EG	0.1206	0.1324	0.3002	0.3153
DeepLIFT	0.0280	0.0266	0.0758	0.0854
BIG	0.0751	0.0275	0.1597	0.0635
AttEXplore	0.1193	0.0207	0.3395	0.0451
AGI	0.1145	0.0198	0.3609	0.0541
LAAI	0.2104	0.0226	0.5056	0.0709

The reported 50% split has larger LAAI insertion values than the compared methods in both strata, but deletion favors other methods. Its configuration is not identified sufficiently to treat these entries as a reaggregation of the main 70% tables.

D.3 Exploration and iteration settings

Table 10: Ablation study results for varying explorations (M) and attack iterations (T).

M	T	INS	DEL	GAP	F-INS	F-DEL	GAP	KL-INS	KL-DEL	GAP
10	10	0.5753	0.1070	0.4683	0.5713	0.0778	0.4936	10.2542	3.7087	6.5455
	15	0.5804	0.1092	0.4711	0.5768	0.0790	0.4978	10.3205	3.7195	6.6010
	20	0.5826	0.1097	0.4729	0.5815	0.0797	0.5018	10.3568	3.7395	6.6174
	25	0.5840	0.1109	0.4730	0.5837	0.0805	0.5032	10.3743	3.7447	6.6296
15	10	0.5851	0.1106	0.4745	0.5835	0.0803	0.5032	10.4150	3.7504	6.6646
	15	0.5887	0.1118	0.4769	0.5878	0.0827	0.5052	10.4560	3.7702	6.6858
	20	0.5907	0.1121	0.4786	0.5905	0.0827	0.5078	10.4815	3.7840	6.6976
	25	0.5913	0.1130	0.4782	0.5920	0.0840	0.5081	10.4902	3.7934	6.6968
20	10	0.5904	0.1121	0.4783	0.5903	0.0825	0.5078	10.4740	3.7864	6.6876
	15	0.5934	0.1132	0.4802	0.5929	0.0850	0.5079	10.5059	3.8044	6.7015
	20	0.5948	0.1135	0.4812	0.5946	0.0856	0.5089	10.5260	3.8149	6.7111
	25	0.5950	0.1144	0.4805	0.5963	0.0863	0.5100	10.5346	3.8238	6.7109
25	10	0.5936	0.1137	0.4799	0.5933	0.0844	0.5089	10.5115	3.8068	6.7047
	15	0.5961	0.1143	0.4818	0.5959	0.0861	0.5098	10.5397	3.8242	6.7155
	20	0.5963	0.1149	0.4814	0.5969	0.0867	0.5102	10.5520	3.8385	6.7134
	25	0.5967	0.1152	0.4816	0.5981	0.0875	0.5106	10.5621	3.8474	6.7148

The full grid preserves all M, T settings. These values describe the archived run; the available table does not specify its model and sample subset. In particular, its default INS value 0.5948 cannot be compared directly with the two main Inception-v3 strata without further protocol information.

D.4 Standard scores

Table 11: Aggregate standard insertion/deletion: Inception-v3. Higher INS and lower DEL are preferred.

Method	INS	DEL	GAP
SM	0.0701	0.0436	0.0265
IG	0.0787	0.0304	0.0483
FIG	0.0414	0.0587	-0.0173
BIG	0.1577	0.0698	0.0879
MFABA	0.2281	0.0709	0.1572
AttEXplore	0.3066	0.0620	0.2446
GIG	0.1094	0.0414	0.0680
EG	0.3437	0.2890	0.0547
DeepLIFT	0.0944	0.0715	0.0229
SG	0.1887	0.0388	0.1499
AGI	0.2992	0.0684	0.2308
LAAI (Ours)	0.4276	0.0866	0.3410

Table 12: Aggregate standard insertion/deletion: ResNet-50. Higher INS and lower DEL are preferred.

Method	INS	DEL	GAP
SM	0.0580	0.0323	0.0257
IG	0.0458	0.0223	0.0235
FIG	0.0284	0.0413	-0.0129
BIG	0.1031	0.0609	0.0422
MFABA	0.1214	0.0681	0.0533
AttEXplore	0.2379	0.0466	0.1912
GIG	0.0522	0.0221	0.0302
EG	0.2763	0.2215	0.0548
DeepLIFT	0.0461	0.0358	0.0102
SG	0.1256	0.0246	0.1010
AGI	0.3103	0.0572	0.2531
LAAI (Ours)	0.3353	0.0601	0.2752

Table 13: Aggregate standard insertion/deletion: VGG16. Higher INS and lower DEL are preferred.

Method	INS	DEL	GAP
SM	0.0426	0.0196	0.0230
IG	0.0312	0.0151	0.0162
FIG	0.0183	0.0286	-0.0103
BIG	0.0688	0.0399	0.0289
MFABA	0.0865	0.0420	0.0444
AttEXplore	0.1870	0.0376	0.1494
GIG	0.0433	0.0182	0.0251
EG	0.2602	0.2126	0.0476
DeepLIFT	0.0413	0.0252	0.0161
SG	0.1300	0.0194	0.1106
AGI	0.2094	0.0374	0.1719
LAAI (Ours)	0.2383	0.0397	0.1987

D.5 Learned-baseline scores by confidence

Table 14: Learned-baseline insertion/deletion: Inception-v3. Higher F-INS and lower F-DEL are preferred.

	Low confidence ($< 70\%$)			High confidence ($\geq 70\%$)		
Method	F-INS	F-DEL	GAP	F-INS	F-DEL	GAP
SM	0.024	0.0207	0.0033	0.0549	0.0634	-0.0085
IG	0.0306	0.0288	0.0018	0.0586	0.0934	-0.0348
FIG	0.0276	0.0309	-0.0033	0.0845	0.0746	0.0099
BIG	0.0802	0.0296	0.0506	0.1629	0.0648	0.0981
MFABA	0.0923	0.0288	0.0635	0.2632	0.0679	0.1953
AttEXplore	0.1262	0.0218	0.1044	0.3503	0.0462	0.3041
GIG	0.0288	0.0284	0.0004	0.0615	0.0889	-0.0274
EG	0.1306	0.1384	-0.0078	0.3073	0.3241	-0.0168
DeepLIFT	0.0301	0.0296	0.0005	0.0779	0.0879	-0.01
SG	0.0241	0.0255	-0.0014	0.0408	0.0779	-0.0371
AGI	0.1211	0.0217	0.0994	0.3734	0.0555	0.3179
LAAI (our)	0.2139	0.0273	0.1866	0.5223	0.072	0.4503

Table 15: Learned-baseline insertion/deletion: ResNet-50. Higher F-INS and lower F-DEL are preferred.

	Low confidence ($< 70\%$)			High confidence ($\geq 70\%$)		
Method	F-INS	F-DEL	GAP	F-INS	F-DEL	GAP
SM	0.0187	0.0294	-0.0107	0.047	0.068	-0.021
IG	0.0222	0.0601	-0.0379	0.0527	0.1305	-0.0778
FIG	0.0561	0.0271	0.029	0.1188	0.0671	0.0517
BIG	0.0488	0.035	0.0138	0.1399	0.0776	0.0623
MFABA	0.0533	0.037	0.0163	0.1601	0.0877	0.0724
AttEXplore	0.0943	0.0279	0.0664	0.2881	0.0603	0.2278
GIG	0.0198	0.0539	-0.0341	0.0485	0.1163	-0.0678
EG	0.0838	0.0951	-0.0113	0.2282	0.2769	-0.0487
DeepLIFT	0.0273	0.0576	-0.0303	0.0748	0.1182	-0.0434
SG	0.0124	0.0577	-0.0453	0.0224	0.1156	-0.0932
AGI	0.0808	0.026	0.0548	0.3784	0.0661	0.3123
LAAI (our)	0.1305	0.0368	0.0937	0.3881	0.0705	0.3176

Table 16: Learned-baseline insertion/deletion: VGG16. Higher F-INS and lower F-DEL are preferred.

	Low confidence ($< 70\%$)			High confidence ($\geq 70\%$)		
Method	F-INS	F-DEL	GAP	F-INS	F-DEL	GAP
SM	0.0133	0.0188	-0.0055	0.0238	0.0408	-0.017
IG	0.0172	0.0343	-0.0171	0.0282	0.0788	-0.0506
FIG	0.0319	0.0205	0.0114	0.0729	0.033	0.0399
BIG	0.0331	0.019	0.0141	0.0729	0.0411	0.0318
MFABA	0.0377	0.0195	0.0182	0.1018	0.0422	0.0596
AttEXplore	0.0756	0.0161	0.0595	0.215	0.036	0.179
GIG	0.0154	0.0309	-0.0155	0.027	0.0708	-0.0438
EG	0.0629	0.0612	0.0017	0.1643	0.1433	0.021
DeepLIFT	0.0157	0.0363	-0.0206	0.031	0.0742	-0.0432
SG	0.0096	0.0318	-0.0222	0.0154	0.0781	-0.0627
AGI	0.0545	0.0157	0.0388	0.2538	0.039	0.2148
LAAI (our)	0.0965	0.0164	0.0801	0.2738	0.0372	0.2366

D.6 Learned-baseline aggregate scores

Table 17: Aggregate learned-baseline insertion/deletion: Inception-v3. Higher F-INS and lower F-DEL are preferred.

Method	F-INS	F-DEL	GAP
SM	0.0461	0.0513	-0.0052
IG	0.0507	0.0751	-0.0244
FIG	0.0683	0.0622	0.0061
BIG	0.1394	0.0548	0.0846
MFABA	0.2147	0.0568	0.1579
AttEXplore	0.2867	0.0393	0.2474
GIG	0.0693	0.0846	-0.0153
EG	0.3031	0.2508	0.0522
DeepLIFT	0.0844	0.0890	-0.0046
SG	0.0561	0.0793	-0.0232
AGI	0.3017	0.0459	0.2559
LAAI (Ours)	0.4347	0.0593	0.3754

Table 18: Aggregate learned-baseline insertion/deletion: ResNet-50. Higher F-INS and lower F-DEL are preferred.

Method	F-INS	F-DEL	GAP
SM	0.0411	0.0599	-0.0188
IG	0.0463	0.1158	-0.0695
FIG	0.1057	0.0587	0.0470
BIG	0.1208	0.0687	0.0522
MFABA	0.1378	0.0771	0.0606
AttEXplore	0.2476	0.0535	0.1941
GIG	0.0501	0.1025	-0.0524
EG	0.2379	0.1893	0.0486
DeepLIFT	0.0712	0.1030	-0.0318
SG	0.0272	0.1106	-0.0834
AGI	0.3162	0.0577	0.2586
LAAI (Ours)	0.3343	0.0635	0.2708

Table 19: Aggregate learned-baseline insertion/deletion: VGG16. Higher F-INS and lower F-DEL are preferred.

Method	F-INS	F-DEL	GAP
SM	0.0201	0.0331	-0.0130
IG	0.0244	0.0632	-0.0388
FIG	0.0585	0.0286	0.0299
BIG	0.0589	0.0333	0.0256
MFABA	0.0793	0.0343	0.0451
AttEXplore	0.1661	0.0290	0.1371
GIG	0.0262	0.0497	-0.0235
EG	0.1829	0.1589	0.0240
DeepLIFT	0.0277	0.0557	-0.0280
SG	0.0147	0.0537	-0.0390
AGI	0.1838	0.0309	0.1530
LAAI (Ours)	0.2116	0.0299	0.1817

D.7 Distributional aggregate scores

Table 20: Aggregate distributional scores: Inception-v3. Higher KL-INS and lower KL-DEL are preferred.

Method	KL-INS	KL-DEL	GAP
IG	4.6330	4.7326	-0.0996
GIG	4.6166	4.6722	-0.0556
SM	4.2523	4.2874	-0.0351
SG	4.2098	4.2448	-0.0351
DeepLIFT	4.4600	4.4673	-0.0073
FIG	4.5110	4.4910	0.0199
EG	6.4901	5.8381	0.6520
BIG	5.8065	4.2149	1.5916
MFABA	7.0808	4.0553	3.0255
AGI	8.0736	4.0786	3.9949
AttEXplore	8.1982	3.4808	4.7174
LAAI (Ours)	10.5564	3.7794	6.7771

Table 21: Aggregate distributional scores: ResNet-50. Higher KL-INS and lower KL-DEL are preferred.

Method	KL-INS	KL-DEL	GAP
IG	5.2411	5.7370	-0.4959
GIG	5.4314	5.8300	-0.3986
SM	5.8697	5.9784	-0.1087
SG	6.1572	6.6074	-0.4501
DeepLIFT	5.1015	5.5145	-0.4130
FIG	5.5031	5.2329	0.2702
EG	5.5225	5.2205	0.3020
BIG	5.7889	4.6781	1.1109
MFABA	6.1156	4.3226	1.7929
AGI	7.2495	4.4588	2.7907
AttEXplore	7.0813	5.2140	1.8674
LAAI (Ours)	7.4023	4.2079	3.1944

Table 22: Aggregate distributional scores: VGG16. Higher KL-INS and lower KL-DEL are preferred.

Method	KL-INS	KL-DEL	GAP
IG	4.0892	4.3924	-0.3032
GIG	4.0184	4.2275	-0.2092
SM	4.1630	4.2865	-0.1235
SG	4.7972	4.9943	-0.1971
DeepLIFT	3.8561	4.3923	-0.5362
FIG	4.2262	3.9570	0.2692
EG	5.6787	5.4591	0.2196
BIG	4.5206	4.0025	0.5181
MFABA	4.7672	3.9528	0.8145
AGI	7.0833	4.4223	2.6610
AttEXplore	6.7422	4.2717	2.4705
LAAI (Ours)	7.4984	4.6113	2.8871

D.8 Additional distributional confidence split

Table 23: Additional distributional confidence record: Inception-v3. Higher KL-INS and lower KL-DEL are preferred.

	Low confidence ($< 70\%$)		High confidence ($\geq 70\%$)	
Method	KL-INS	KL-DEL	KL-INS	KL-DEL
SM	4.0006	4.1461	4.2733	4.2992
IG	4.3213	4.478	4.659	4.7538
FIG	4.3368	4.2119	4.5255	4.5143
BIG	5.7597	4.0682	5.8104	4.2271
MFABA	5.7783	4.0013	7.1895	4.0598
AttEXplore	6.4325	3.4516	8.3455	3.4832
GIG	4.3707	4.4534	4.6371	4.6904
EG	4.9925	4.8459	6.615	5.9209
DeepLIFT	4.1727	4.2879	4.484	4.4823
SG	4.173	4.1349	4.2128	4.254
AGI	5.5176	4.0858	8.2868	4.078
LAAI (our)	7.6129	3.6638	10.802	3.789

Table 24: Additional distributional confidence record: ResNet-50. Higher KL-INS and lower KL-DEL are preferred.

	Low confidence ($< 70\%$)		High confidence ($\geq 70\%$)	
Method	KL-INS	KL-DEL	KL-INS	KL-DEL
SM	5.7474	5.7749	5.8957	6.0216
IG	5.1251	5.2361	5.2658	5.8433
FIG	5.1004	5.0536	5.5885	5.2709
BIG	5.5494	4.1986	5.8397	4.7798
MFABA	5.6028	4.075	6.2243	4.3752
AttEXplore	6.2566	5.0651	7.2563	5.2456
GIG	5.3644	5.4411	5.4456	5.9125
EG	4.9839	4.6738	5.6367	5.3364
DeepLIFT	4.8818	5.1143	5.1481	5.5994
SG	6.2011	6.1561	6.1479	6.7031
AGI	5.8622	4.5894	7.5437	4.431
LAAI (our)	6.3754	3.9806	7.6201	4.2561

Table 25: Additional distributional confidence record: VGG16. Higher KL-INS and lower KL-DEL are preferred.

	Low confidence ($< 70\%$)		High confidence ($\geq 70\%$)	
Method	KL-INS	KL-DEL	KL-INS	KL-DEL
SM	4.0809	4.0889	4.1862	4.3423
IG	3.9275	4.0115	4.1348	4.4998
FIG	3.8839	3.8245	4.3227	3.9944
BIG	4.2773	3.8223	4.5892	4.0533
MFABA	4.454	3.8081	4.8555	3.9935
AttEXplore	6.0355	4.1755	6.9415	4.2988
GIG	3.9515	3.9936	4.0372	4.2935
EG	5.0065	4.906	5.8683	5.6151
DeepLIFT	3.7093	4.0851	3.8975	4.4789
SG	4.7479	4.7833	4.8112	5.0538
AGI	5.2245	4.1857	7.6075	4.489
LAAI (our)	6.0499	4.4146	7.9069	4.6668

D.9 ViT-B/16 CONFIDENCE SPLIT

Table 26: Interpretability performance on the ImageNet dataset (ViT-B/16).

Method	**Low Conf. Insertion**	**Low Conf. Deletion**	**High Conf. Insertion**	**High Conf. Deletion**
SM	0.0791	0.0330	0.1308	0.0782
IG	0.0603	0.0279	0.1238	0.0562
FIG	0.0294	0.0622	0.0687	0.1045
BIG	0.1354	0.0710	0.2452	0.1533
MFABA	0.1307	0.0765	0.2444	0.1937
AttEXplore	0.1241	0.0507	0.2269	0.1131
GIG	0.0582	0.0253	0.1145	0.0507
EG	0.1764	0.2069	0.3291	0.3909
DeepLIFT	0.0622	0.0284	0.0960	0.0786
SG	0.1475	0.0235	0.2210	0.0318
AGI	0.1044	0.0678	0.2234	0.1495
LAAI	0.1774	0.0531	0.3437	0.1269

D.10 CIFAR100 COMPARISON

Table 27: Interpretability performance on the CIFAR-100 dataset (VGG16).

Method	Low Conf. Insertion	Low Conf. Deletion	High Conf. Insertion	High Conf. Deletion
SM	0.0340	0.0409	0.0463	0.0391
IG	0.0325	0.0344	0.0428	0.0356
FIG	0.0367	0.0329	0.0367	0.0399
BIG	0.0427	0.0405	0.0577	0.0486
MFABA	0.0476	0.0408	0.0722	0.0494
AttEXplore	0.0592	0.0391	0.1157	0.0570
GIG	0.0323	0.0340	0.0421	0.0338
EG	0.1008	0.1099	0.1971	0.2257
DeepLIFT	0.0336	0.0327	0.0466	0.0350
SG	0.0403	0.0358	0.0694	0.0338
AGI	0.0443	0.0438	0.0655	0.0576
LAAI	0.1384	0.0438	0.2651	0.0661

D.11 INPUT PERTURBATION COMPARISON

Table 28: Interpretability performance across different perturbation-levels on the ImageNet dataset (Inception-v3). Here ϵ represents different perturbation rates.

Method	ϵ=2	ϵ=4	ϵ=6	ϵ=8	ϵ=10	ϵ=12	ϵ=14	ϵ=16
SM	0.0205	0.0178	0.0174	0.0224	0.0268	0.0285	0.0317	0.0208
IG	0.0138	0.0180	0.0234	0.0241	0.0292	0.0365	0.0428	0.0296
FIG	0.0193	0.0158	0.0231	0.0105	0.0193	0.0230	0.0114	0.0111
BIG	0.1147	0.1308	0.1093	0.1147	0.1208	0.1188	0.1228	0.0925
MFABA	0.1307	0.1443	0.1319	0.1229	0.1614	0.1596	0.1426	0.1379
AttEXplore	0.1311	0.1270	0.1242	0.1058	0.1331	0.1180	0.1312	0.1258
GIG	0.0150	0.0143	0.0200	0.0219	0.0222	0.0279	0.0255	0.0235
EG	0.1801	0.1784	0.1765	0.1775	0.1774	0.1796	0.1833	0.1790
DeepLIFT	0.0244	0.0317	0.0203	0.0291	0.0312	0.0211	0.0245	0.0260
SG	0.0518	0.0472	0.0542	0.0420	0.0544	0.0664	0.0509	0.0463
AGI	0.1801	0.1978	0.1890	0.1854	0.1778	0.1621	0.1617	0.1962
LAAI	0.2475	0.2464	0.2503	0.2411	0.2622	0.2614	0.2547	0.2654

D.12 NAIVE CLASS AVERAGING ON MNIST

Table 29: Comparison of LAAI and Naive IG Averaging on MNIST (10,000 images).

Method	Time (seconds) ↓	INS Score ↑	DEL Score ↓	GAP ↑
IG-Avg (All Classes)	1126.18	0.5379	0.5466	-0.0087
LAAI (Ours)	68.59	0.8843	0.2351	0.6492

The MNIST comparison reports $1{,}126.18$ seconds for IG-Avg and 68.59 for LAAI, approximately a 16.4-fold ratio in that run. A negative INS–DEL gap alone does not show equivalence to random noise. The output type and path construction are essential for interpreting this baseline.

For IG of class probabilities using the same baseline and path for every class, linearity gives

$$\frac{1}{C}\sum_{j}\mathrm{IG}_i(P_j) = \frac{x_i - x'_i}{C}\int_0^1 \partial_i \sum_j P_j(x' + t(x - x'))\,dt = 0,$$

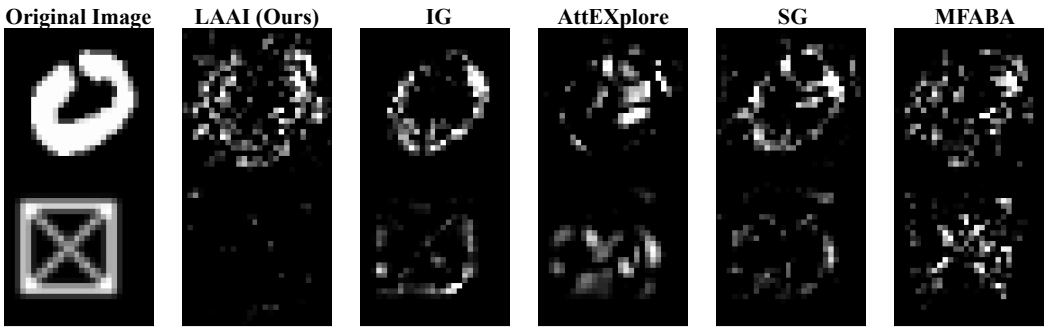

Figure 3: Illustrative attribution maps with an added square region. The maps show differences in assigned importance; visual appearance alone does not establish whether the classifier uses that region.

because $\sum_j P_j = 1$. Averaging logits, absolute attributions, normalized maps, or class-dependent paths defines a different operation. This exact cancellation, rather than a generic appeal to softmax nonlinearity, is why naive probability-gradient averaging differs from Equation 4. The archived IG-Avg results do not identify all conventions needed to distinguish these possibilities.

E SUPPLEMENTARY QUALITATIVE PANELS

These panels retain the original images while limiting their interpretation to displayed attribution patterns. They are not controlled measurements of model reliance or oracle feature relevance.



Figure 4: AttEXplore digit maps under different nominated classes. The displayed similarity motivates examining the explanation objective, without establishing a universal absence of class sensitivity.

F SCOPE OF THE EXPLANATION

The oracle set $\Phi(x)$ supplies a conceptual relevance criterion, not an input to LAAI or an observed ground truth in the reported metrics. A label-agnostic scalar does not require balanced mass across objects and cannot answer which class each feature favors. A constant baseline can reduce concentration in the output distribution without being a natural image. These distinctions keep the contribution centered on the stated objective and the observed feature-ordering behavior, rather than treating symmetry as a universal guarantee of interpretability.



Figure 5: Maps using the cat class before and after masking part of a dog image. Masking changes the input as well as the attribution; this illustration does not isolate a unique cause of the response change.



Figure 6: Comparison of Different Feature Replacement Methods