Review : Learning to estimate shapley values with vision transformers (ViT-Shapley)

Ian Covert, Chanwoo Kim, Su-In Lee

May 20, 2026

Seoul National University - IDEA Lab.
Presenter: Sehyun Park

Introduction

Categorization of Interpretation Methods

► Interpretation methods can be categorized according to whether interpretability is built into the model or applied after training.

(1) White-box Model Interpretation

- The model is designed so that its prediction mechanism is itself interpretable.
- Since interpretability constraints are imposed during training, the model may exhibit degraded predictive performance compared to unconstrained black-box models.

(2) Post-hoc Interpretation

- The trained model is kept fixed; only the interpretation method is applied.
- **Main post-hoc interpretation approaches**
  - *Attention-based*: Uses self-attention weights.
  - *Gradient-based*: Uses gradients of the output with respect to the input.
  - *Shapley-based*: Quantifies feature contributions using cooperative game theory.

Post-hoc: Attention-based Explanation

- **Key idea**: Tokens that receive higher attention scores are considered more important for the model' s prediction.
- **Related works**
  - *Raw Attention*: Uses the [CLS] token attention map from the last layer.
  - *Attention Rollout and Flow* (Abnar, Samira, et al., 2020 [1]): Propagate attention scores across layers using matrix multiplication.
- **Limitations**
  - Attention scores alone do not account for value embeddings.
  - Whether attention is an explanation is still debated (Jain, Sarthak, et al., 2019 [2]).
  - Class-agnostic: Does not explain a specific target class.

Post-hoc: Gradient-based Explanation

- **Key idea**: Compute the importance of each pixel or token using the gradient of the class score with respect to the input.

$$R_i = \left| \frac{\partial \hat{y}_c}{\partial x_i} \right|. \tag{1}$$

- **Related works**
  - *GradCAM* (Selvaraju, Ramprasaath R., et al., 2017 [3])
  - *LRP* (Voita, Elena, et al., 2019 [4])
  - *Transformer Attribution* (Chefer et al., 2021 [5]): Combines gradients and attention maps to estimate class-specific token importance in Transformer models.
  - *LeGrad* (Bousselham, Walid, et al., 2025 ICCV [6])

- **Limitations**
  - The importance map can change significantly even with small input variations.
  - Gradient-based methods typically focus on local sensitivity of individual patches, while few studies address interactions between patches in ViTs.

Shapley-based Explanation

- **Key idea**: Quantifies feature attributions to model predictions using the Shapley value from cooperative game theory.

$$\phi_i(\boldsymbol{x}) = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!\,(|\mathcal{N}| - |S| - 1)!}{|\mathcal{N}|!} \left[v_{\boldsymbol{x}}(S \cup \{i\}) - v_{\boldsymbol{x}}(S) \right]. \tag{2}$$

  where
  - $\boldsymbol{x} = (x_1, \ldots, x_N)$ denotes the input, represented as a collection of N features.
  - $\mathcal{N} = \{1, \ldots, N\}$ denotes the set of all features.
  - $v_{\boldsymbol{x}}(S)$ denotes a value function for a subset $S \subseteq \mathcal{N}$ of features.

- The Shapley value is the unique attribution rule satisfying the following properties.
  - *Efficiency*
  - *Symmetry*
  - *Dummy*
  - *Additivity*

Shapley-based Explanation

- **From Shapley value to SHAP**: SHAP explains a prediction by choosing the value function as

$$v_{\boldsymbol{x}}(S) = \mathbb{E}[f(\boldsymbol{X}) \mid \boldsymbol{X}_S = \boldsymbol{x}_S], \tag{3}$$

  where $\boldsymbol{X} \sim \mu$. Here, $\boldsymbol{X}_S = \boldsymbol{x}_S$ means that the features in S are fixed to their input values.

KernelSHAP

KernelSHAP reformulates Shapley value estimation as a weighted linear regression problem. For each binary vector $\boldsymbol{z} \in \mathcal{Z} = \{0,1\}^N$, let $S_{\boldsymbol{z}} = \{i : z_i = 1\}$. KernelSHAP solves

$$\hat{\boldsymbol{\phi}}_{\boldsymbol{x}} = \arg\min_{\boldsymbol{\phi}} \sum_{\boldsymbol{z} \in \mathcal{Z}} \pi(\boldsymbol{z}) \left[v_{\boldsymbol{x}}(S_{\boldsymbol{z}}) - v_{\boldsymbol{x}}(\emptyset) - \boldsymbol{z}^\top \boldsymbol{\phi} \right]^2 \tag{4}$$

subject to

$$\mathbf{1}^\top \boldsymbol{\phi} = v_{\boldsymbol{x}}(\mathcal{N}) - v_{\boldsymbol{x}}(\emptyset), \tag{5}$$

where

$$\pi(\boldsymbol{z}) = \frac{N-1}{\binom{N}{|S_{\boldsymbol{z}}|} |S_{\boldsymbol{z}}| (N - |S_{\boldsymbol{z}}|)}, \qquad \pi(\mathbf{0}) = \pi(\mathbf{1}) = 0. \tag{6}$$

ViT-Shapley (Covert et al., 2023, ICLR [7])

Notation

- Image patches:

$$\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N), \qquad \mathcal{N} = \{1, \ldots, N\}. \tag{7}$$

- Fine-tuned ViT for binary classification:

$$\mathrm{ViT}_{\theta_0} : \mathcal{X} \to [0, 1]. \tag{8}$$

- Patch subset $S \subseteq \mathcal{N}$:

$$\mathrm{ViT}_\theta(\boldsymbol{x}, S) = \text{prediction when attention is masked for all patches not in } S. \tag{9}$$

$$i \notin S \Rightarrow \boldsymbol{x}_i \text{ is attention-masked.}$$

ViT-Shapley (Covert et al., 2023, ICLR [7])

- **Goal**: Given a test image $\boldsymbol{x}'$, efficiently estimate its patch-wise Shapley values:

$$\hat{\boldsymbol{\phi}}_{\boldsymbol{x}'} = \left(\hat{\phi}_1(\boldsymbol{x}'), \ldots, \hat{\phi}_N(\boldsymbol{x}')\right). \tag{10}$$

- **Core idea 1: Train surrogate ViT**
  - Shapley values are defined through the value function $v_{\boldsymbol{x}}(S)$:
  - A direct choice is to use the pre-trained ViT with masking:

$$v_{\boldsymbol{x}}(S) = \mathrm{ViT}_{\theta_0}(\boldsymbol{x}, S). \tag{11}$$

  - However, partial inputs can be off-manifold:

$$S \neq \mathcal{N} \quad \Rightarrow \quad (\boldsymbol{x}, S) \not\sim \text{natural image distribution}. \tag{12}$$

  - Train a surrogate ViT by distilling the pre-trained ViT:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{p(\boldsymbol{x})} \mathbb{E}_{p(S)} \left[D_{\mathrm{KL}} \left(\mathrm{ViT}_{\theta_0}(\boldsymbol{x}) \,\middle\|\, \mathrm{ViT}_{\theta}(\boldsymbol{x}, S) \right) \right]. \tag{13}$$

  - Surrogate value function:

$$v_{\boldsymbol{x}}(S) = \mathrm{ViT}_{\hat{\theta}}(\boldsymbol{x}, S). \tag{14}$$

ViT-Shapley (Covert et al., 2023, ICLR [7])

- **Core idea 2: Train explainer model**
  - With the surrogate value function,

$$v_{\boldsymbol{x}}(S) = \mathrm{ViT}_{\hat{\theta}}(\boldsymbol{x}, S), \qquad S \subseteq \mathcal{N}. \tag{15}$$

  - Computing exact Shapley values requires evaluating all 2^N subsets, and Monte-Carlo approximation is still expensive.
  - To avoid repeated subset sampling at test time, ViT-Shapley proposes an explainer model g_ψ that directly predicts patch-wise Shapley values:

$$g_\psi(\boldsymbol{x}) \approx \boldsymbol{\phi}_{\boldsymbol{x}} = (\phi_1(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x}))\,. \tag{16}$$

  - The explainer model is trained by the Shapley regression objective:

$$\hat{\psi} = \arg\min_{\psi} \mathbb{E}_{p(\boldsymbol{x})} \mathbb{E}_{p_{\mathrm{Sh}}(\boldsymbol{s})} \left[\left(v_{\boldsymbol{x}}(S_{\boldsymbol{s}}) - v_{\boldsymbol{x}}(\emptyset) - \boldsymbol{s}^\top g_\psi(\boldsymbol{x}) \right)^2 \right], \tag{17}$$

$$\text{s.t.} \quad \mathbf{1}^\top g_\psi(\boldsymbol{x}) = v_{\boldsymbol{x}}(\mathcal{N}) - v_{\boldsymbol{x}}(\emptyset).$$

    where $p_{\mathrm{Sh}}(S) \propto (\mathbf{1}^\top \boldsymbol{s} - 1)!(N - \mathbf{1}^\top \boldsymbol{s} - 1)!$, $p_{\mathrm{Sh}}(\mathbf{0}) = p_{\mathrm{Sh}}(\mathbf{1}) = 0$.

References

[1] S. Abnar and W. Zuidema, **"Quantifying attention flow in transformers,"** in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4190–4197.

[2] S. Jain and B. C. Wallace, **"Attention is not explanation,"** in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 3543–3556.

[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, **"Grad-cam: Visual explanations from deep networks via gradient-based localization,"** in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[4] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, **"Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned,"** in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 5797–5808.

[5] H. Chefer, S. Gur, and L. Wolf, **"Transformer interpretability beyond attention visualization,"** in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 782–791.

[6] W. Bousselham, A. Boggust, S. Chaybouti, H. Strobelt, and H. Kuehne, **"Legrad: An explainability method for vision transformers via feature formation sensitivity,"** in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 20 336–20 345.

[7] I. Covert, C. Kim, and S.-I. Lee, **"Learning to estimate shapley values with vision transformers,"** *arXiv preprint arXiv:2206.05282*, 2022.

End