VPNet: A New Explainable-AI Paradigm for Tractable Multimodal, Multidimensional Classification with probabilistic circuits on ViT Features

Taishi Kusumoto

April 14, 2026

Taishi Kusumoto
Independent Researcher
Email: tai8242@gmail.com

Abstract

Deep learning-based classification models can understand and extract complex representations or relationships between classes. If the decision process of such AI can be visualized in a way that humans can understand, this may create the potential for AI tools to assist in new scientific discoveries or for more trustworthy AI models. However, such visualizations produced by current XAI methods, including Grad-CAM and attention maps, are theoretically questionable.

To overcome this limitation, we propose a novel theoretically explainable AI model for classification tasks that can handle multimodal and multidimensional information. Unlike traditional classification models based on convolutional neural networks or transformers, our approach combines probabilistic circuits with two vision transformers, DINO and CLIP, enabling probabilistic interpretability at the patch level, encoder level, and modality level. We demonstrate that our proposed method can overcome the randomized test for saliency checks, a test that current state-of-the-art XAI models often fail. Additionally, this study shows the explainability of the model's decisions at the patch, encoder, and modality levels, while achieving strong generalization performance across various classification tasks.

Explainable AI (XAI), Probabilistic Circuits, Vision Transformers (ViT), Multimodal Medical Imaging, 3D Brain MRI, Patch-level Attribution, Tractable Probabilistic Models, Multidimensional Classification

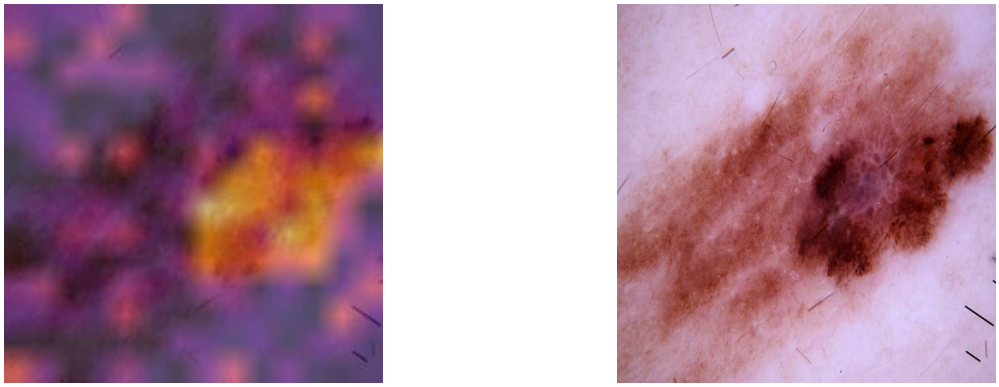
1 Introduction

The recent advancement in deep learning techniques has raised the expectation that AI models may contribute to new scientific discoveries, potentially enabling breakthroughs across many scientific fields. From this perspective, one of the most promising deep learning techniques may be supervised classification. With an appropriately labeled dataset, if a classification model is well optimized, then visualizing its decision process may help scientists uncover new relationships between the classes because such models can recognize complex features and relationships not understandable for humans.

However, traditional classification algorithms based on convolutional neural networks (CNNs)[1] or transformers[2] often complicate such visualization because of their black-box characteristics. Their complex mathematical operations, both linear and nonlinear, intricately mix the decision process used for classification, making it difficult to interpret.

To overcome this situation, several explainable AI (XAI) methods for classification have been proposed to extract the model's decision process [3], including Grad-CAM based approaches [4], LRP based approaches [5, 6], and attention map based approaches [7, 8]. However, as some studies suggest [9] [10], these techniques may merely project complex nonlinear features in neural networks into a human-interpretable form, without theoretical evidence that the resulting visualization directly corresponds to the actual decision-making process of the model.

To address this limitation of current state of the art techniques, we decided to take a completely different approach to visualization. Instead of using CNNs or transformers, we adopted a tractable



(a) Heatmap (b) Original Image with melanoma

Figure 1: Visualization of melanoma classification. Bright regions indicate areas with higher probabilistic contributions to the melanoma class, whereas dark regions indicate areas with lower contributions. The bright regions are concentrated on the lesion, suggesting that the model focuses on the lesion for classification. In addition, some areas within the lesion show lower probabilistic contributions to the melanoma class, suggesting that the model considers these areas less informative for identifying melanoma.

neural network, namely a probabilistic circuit [11–14]. Using probabilistic circuits, it is possible to design tractable classification models without black-box characteristics. However, probabilistic circuits do not have a strong ability to extract complex feature representations compared to CNNs [1] or transformers[2], which complicates the optimization of tractable models based on probabilistic circuits.

The emergence of recent foundation models, trained with enormous computational costs and vast amounts of data, and exhibiting highly sophisticated feature extraction abilities, can help resolve this limitation. We propose a new framework that connects embedding vectors from a foundation model to probabilistic circuits, without disturbing tractability, by strictly preserving decomposability and smoothness [11] in the circuit. In particular, we connect token-level (patch-level) embedding vectors to disjoint variable scopes so that the circuit structure remains decomposable and smooth while incorporating rich representations from the foundation model. With this algorithm, probabilistic circuits can leverage the strong feature extraction ability of the foundation model, enabling more complex classification tasks.

This framework can be applied to classification tasks in any domain with an appropriate foundation model, including text, images, genes, audio, signals, and so on. However, in this paper, as one example of our framework, we propose VPNet, a 2D or 3D multimodal image classification algorithm. VPNet combines two parameter-frozen vision transformers[15], CLIP [16] and DINO[17], which are foundation models trained on large-scale image datasets, with probabilistic circuits. In this study, our contributions are as follows:

- We propose a new classification algorithm based on the combination of vision transformers and probabilistic circuits, enabling visualization of the model's decision process.
- We check the representational capability of our algorithm: our model can be successfully optimized for complex image classification tasks without clear underfitting or overfitting, through tumor classification of brain MRI images in BRISC2025 [18], pneumonia classification of chest X-ray images in ChestX-ray14 [19] and melanoma classification of dermoscopic images from the ISIC archive [20][21].
- We show the interpretability of our model by visualizing saliency maps. The empirical validity of our model is supported using the MVTec anomaly detection dataset [22] by demonstrating consistency between the probabilistic contribution for the NG class and the actual flaws in the images.
- We introduce the flexibility of our design by showing that our model can be optimized for 3D multimodal age classification using the IXI 3D brain MRI dataset with T1 and T2 modalities [23].

One example visualization for the melanoma classification task is shown in Figure 1. The bright regions corresponding to high probabilistic contribution for the melanoma class are consistent with

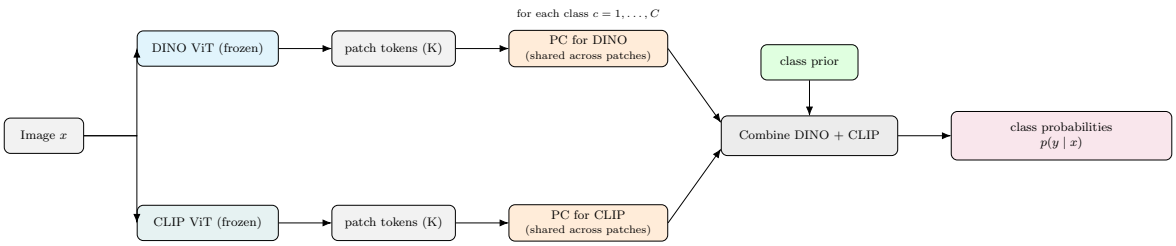


Figure 2: Simplified overview of the class-conditional PC classifier.

the lesion area of the skin. Unlike traditional XAI methods such as Grad-CAM or attention maps, our model is inherently interpretable, and there is no ambiguity as to whether the highlighted regions were actually used in the classification decision. This can increase trust in the model's decision, which is practically important. Additionally, this visualization indicates which parts of the lesion the model considers suggestive of melanoma lesions, which may provide clinical insight. We expect that our framework may reduce the need for traditional post-hoc XAI methods in both practice and research.

2 Methods

2.1 Tractable Probabilistic Circuit for 2D image Classification

A probabilistic circuit is a neural network whose outputs represent the joint probability distribution over all input leaf nodes, provided that the circuit satisfies the following conditions: (a) each scope corresponds to a single random variable, and (b) smoothness and decomposability are preserved.[11] We connected each patch embedding vector from a parameter-frozen vision transformer to our probabilistic circuit.

$$\mathbf{z}_i = g_{\theta^\star}(x)_i \in R^d, \quad i = 1, \ldots, N.$$

Then, each scaler of a patch vector is directly connected to each leaf node of the circuit. Each leaf node represents a univariate Gaussian probability density function,

$$p_{s_j}(z_{ij}) \;=\; \mathcal{N}\big(z_{ij};\, \mu_j,\, \sigma_j^2\big) \;=\; \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(z_{ij}-\mu_j)^2}{2\sigma_j^2}\right), \qquad j = 1, \ldots, d.$$

where z_{ij} denotes the j-th scalar of the i-th patch embedding $\mathbf{z}_i$. We assume that each scalar derived from a patch embedding vector defines a univariate scope. During optimization, the probabilistic circuit learns dependencies among these scalar variables through sum and product nodes defined as follows:

$$S_i \cap S_j = \emptyset, \quad \forall\, i \neq j,$$

which ensures that each scalar variable defines an independent univariate scope.

$$\textbf{Sum node:} \quad p_s(z_i) \;=\; \sum_{k=1}^{K} w_{i,k}\, \mathcal{N}(z_i; \mu_{i,k}, \sigma_{i,k}^2), \qquad w_{i,k} \geq 0,\; \sum_{k=1}^{K} w_{i,k} = 1,$$

$$\textbf{Product node:} \quad p_p(z_i, z_j) \;=\; p_{s_i}(z_i)\, p_{s_j}(z_j), \qquad S_i \cap S_j = \emptyset.$$

where z_i and z_j denote scalar variables corresponding to two distinct univariate scopes S_i and S_j; $p_s(z_i)$ represents the probability computed at a sum node for the scope S_i; and $p_p(z_i, z_j)$ denotes the probability at a product node combining two independent scalar scopes S_i and S_j. The disjointness constraint $S_i \cap S_j = \emptyset$ ensures decomposability of the circuit.

Because the circuit preserves smoothness and decomposability, it encodes the joint distribution of a patch vector $\mathbf{z} = (z_1, \ldots, z_d)^\top$ as the joint distribution of its scalar components:

$$p(\mathbf{z}) \;=\; p\big(z_1, \ldots, z_d\big).$$

where $\mathbf{z} = [z_1, z_2, \ldots, z_D]^\top$ denotes the patch embedding vector, and each $p_{s_i}(z_i)$ represents the probability computed from the scope S_i corresponding to the scalar variable z_i. The disjointness of scopes ensures that the overall joint distribution factorizes exactly as the product of univariate probabilities.

For memory efficiency and to accommodate variable-length inputs, we adopted a **RAT-SPN–based** algorithm [24], in which a compact probabilistic circuit, receiving a single patch embedding vector as input, is shared across all patch embeddings within an image. The number of prepared probabilistic circuits is determined as follows:

$$\text{Number of PCs} = (\text{Number of modalities}) \times (\text{Number of encoders}) \times (\text{Number of classes}).$$

Here, let $\mathbf{z}_n$ denote the n-th patch vector associated with a fixed modality, encoder, and class. The probabilistic circuit tied to these variables outputs the conditional density

$$\mathrm{PC}_{(\text{modality, encoder, class})}(\mathbf{z}_n) \;=\; p\big(\mathbf{z}_n \mid \text{modality, encoder, class}\big).$$

Here, patch vectors belonging to the same modality, encoder, and class share the same probabilistic circuit *conditioned on these variables*. For a patch vector $\mathbf{z}_n$ and fixed (observed) modality/encoder, the circuit outputs the conditional density

$$p\big(\mathbf{z}_n \mid \text{modality, encoder, class}\big).$$

Since the modality and encoder are fixed during training and inference, we regard the components of a patch vector $\mathbf{z}_n$ as decomposed into disjoint sub-scopes $\{\mathbf{z}_n^{(g,e)}\}$ indexed by the (observed) modality–encoder pair (g, e). Assuming conditional independence across these sub-scopes given the class, the probabilistic circuit connects them by a product node, yielding

$$p(\mathbf{z}_n \mid \text{class}) \;=\; \prod_{(g,e)\in\mathcal{G}} p\big(\mathbf{z}_n^{(g,e)} \mid \text{modality} = g,\, \text{encoder} = e,\, \text{class}\big).$$

During optimization, the *joint conditional distribution* of all patch vectors given the class is utilized. This distribution is computed as the product of the conditional distributions of all patch vectors given the class:

$$p(\mathbf{Z} \mid \text{class}) \;=\; \prod_{n=1}^{N} p(\mathbf{z}_n \mid \text{class}), \qquad \mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\},$$

where each $\mathbf{z}_n$ has a disjoint scope from the others.

2.2 Optimization for classification

Here, apart from the probabilistic circuits, we set an additional simple and learnable network. This network represents a categorical distribution of the class prior, defined as

$$p(y = c;\, \pi) \;=\; \pi_c, \qquad \pi_c \geq 0,\; \sum_{c=1}^{C} \pi_c = 1,$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_C)$ denotes the learnable class prior parameters.

The joint distribution factorizes by the product rule as

$$p(\mathbf{Z},\, y{=}c) \;=\; p(\mathbf{Z} \mid y{=}c)\, p(y{=}c) \;=\; p(\mathbf{Z} \mid c)\, \pi_c.$$

The class posterior is then given by Bayes' rule:

$$p(y{=}c \mid \mathbf{Z}) \;=\; \frac{p(\mathbf{Z} \mid c)\, \pi_c}{\sum_{c'=1}^{C} p(\mathbf{Z} \mid c')\, \pi_{c'}}.$$

This class posterior can be used for optimization of learnable parameters. As loss functions, we defined a cross-entropy loss [25] to optimize the parameters,

$$\mathcal{L}_{\mathrm{CE}}(\mathbf{Z}, Y) = -\frac{1}{B} \sum_{i=1}^{B} \log p(y_i \,|\, \mathbf{Z}_i)\,,$$

and a Shannon entropy loss to avoid overfitting [26],

$$\mathcal{L}_{\text{ENT}}(\mathbf{Z}) = \frac{1}{B}\sum_{i=1}^{B}\Big(-\sum_{c=1}^{C} p(y=c \mid \mathbf{Z}_i)\,\log p(y=c \mid \mathbf{Z}_i)\Big),$$

where $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_B\}$ denotes a mini-batch of B inputs (each $\mathbf{Z}_i$ is the collection of patch vectors for one image), and $Y = \{y_1, \ldots, y_B\}$ are the corresponding class labels.

In this paper, we adopt the geometric mean of the patch-wise conditional likelihoods instead of their raw product in order to stabilize training. For class c, we define

$$\tilde{p}(\mathbf{Z} \mid c) \;=\; \left(\prod_{n=1}^{N} p(\mathbf{z}_n \mid c)\right)^{\frac{1}{N}}, \qquad \mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}.$$

Then, we compute a power posterior by applying Bayes' rule to the geometric-mean likelihood:

$$p(y{=}c \mid \mathbf{Z}) \;=\; \frac{\tilde{p}(\mathbf{Z} \mid c)\,\pi_c}{\sum_{c'=1}^{C} \tilde{p}(\mathbf{Z} \mid c')\,\pi_{c'}}.$$

Although this differs from the exact Bayesian posterior based on the full product likelihood, it can still be interpreted as a normalized power posterior [27].

2.3 Learnable parameters of probabilistic circuit

To increase the number of learnable parameters, in other words, the representational capability of our probabilistic circuit, we adopt a **RAT–SPN-based**[24] algorithm under the PyJuice framework [28](although the PyJuice library itself was not directly used in our implementation). In our implementation, it is possible to specify an integer for each of the following hyperparameters:

$$\text{Depth } N, \quad \text{Pieces per patch } P, \quad \text{Latent channels } L, \quad \text{and Repetitions } R.$$

We assume that each scalar from a patch vector has a univariate scope, and each scope is disjoint. This means that any combination of the scalars is also disjoint, as long as there is no overlap among their components. Accordingly, the overall scope of the probabilistic circuit can be expressed as the union of all disjoint scalar scopes:

$$S \;=\; \bigcup_{i=1}^{D} S_i, \qquad S_i \cap S_j = \emptyset \text{ for all } i \neq j.$$

The *depth* N determines how many times the set of scalar components is recursively divided into two disjoint subsets. Each division corresponds to one layer of the RAT-SPN structure, and repeating this operation N times yields 2^N scopes at the leaf level. The probabilistic circuit learns the relationships between scalar components within each scope through sum nodes:

$$p_s(\mathbf{z}_{S_i}) \;=\; \sum_{k=1}^{K} w_k\, p_k(\mathbf{z}_{S_i}), \qquad w_k \geq 0,\ \sum_{k=1}^{K} w_k = 1.$$

This structure ensures the **smoothness** of the probabilistic circuit, since sum nodes operate only within the same scope. Finally, the overall joint probability density of a patch vector is computed from the product node that connects the probabilities from all disjoint scopes:

$$p(\mathbf{z}) \;=\; \prod_{i=1}^{2^N} p_s(\mathbf{z}_{S_i}).$$

Here, **decomposability** is guaranteed because the scopes S_i are mutually disjoint, ensuring that each subcircuit represents an independent factor of the full joint distribution.

The hyperparameter *Pieces per patch* P increases the number of learnable parameters at each leaf node by introducing multiple Gaussian components whose outputs are linearly combined through

softmax-normalized weights. In other words, the univariate Gaussian probability density function at a leaf node is extended to include P parameterized Gaussian functions as follows:

$$p_\ell(z) \;=\; \sum_{m=1}^{P} \alpha_m \, \mathcal{N}(z;\, \mu_m,\, \sigma_m^2), \qquad \alpha_m \geq 0,\; \sum_{m=1}^{P} \alpha_m = 1,$$

This operation simply increases the parameter capacity of the leaf node without changing the overall circuit structure.

The number of latent channels L determines the internal width of the probabilistic circuit at each leaf node. Each leaf node outputs an L-dimensional vector, in which each component represents an independent latent subspace that models different statistical aspects of the input variable. Formally, for an input variable z, the leaf node produces

$$\mathbf{f}(z) = \big[f_1(z),\, f_2(z),\, \ldots,\, f_L(z)\big]^\top,$$

where each latent channel $f_l(z)$ has its own learnable parameters and contributes to the overall representation of the input. As all latent channels share the same scope, the outputs are integrated by sum nodes placed just above the root, which preserves the overall smoothness of the probabilistic circuit. To increase the number of learnable parameters, a linear mapping is applied to the latent outputs before the final integration as follows:

$$\begin{bmatrix} f_1'(z) \\ f_2'(z) \\ \vdots \\ f_L'(z) \end{bmatrix} = W \begin{bmatrix} f_1(z) \\ f_2(z) \\ \vdots \\ f_L(z) \end{bmatrix}, \qquad W \in R^{L \times L}.$$

Each row of W corresponds to a sum node whose coefficients are normalized by a softmax function. Let the i-th row of W be denoted as

$$\mathbf{w}_i = \big[w_{i1},\, w_{i2},\, \ldots,\, w_{iL}\big], \qquad w_{ij} = \frac{\exp(\theta_{ij})}{\sum_{k=1}^{L} \exp(\theta_{ik})},$$

where θ_{ij} are learnable parameters. Then the output of the i-th sum node is

$$f_i'(z) = \sum_{j=1}^{L} w_{ij}\, f_j(z), \qquad \sum_{j=1}^{L} w_{ij} = 1.$$

This formulation ensures that each row of W performs a weighted average over all latent channels, while maintaining probabilistic consistency (nonnegative coefficients summing to one).
Here, the learnable matrix W mixes the latent channels to enhance representational flexibility while keeping the same scope. Finally, the transformed channels are integrated by a sum node:

$$p_{\text{root}} = \text{sumnode}\big(p_1'(z),\, p_2'(z),\, \ldots,\, p_L'(z)\big),$$

where each $p_l'(z)$ corresponds to the probability contribution from the l-th transformed latent channel.

Repetition R determines how many times the probabilistic circuit is duplicated. All duplicated PCs share an identical scope at their root nodes, so their root probabilities can be integrated by a single sum node without violating smoothness. This is represented as

$$p_{\text{root}}(\mathbf{z}) \;=\; \sum_{r=1}^{R} \alpha_r \, p_r(\mathbf{z}), \qquad \alpha_r \geq 0,\; \sum_{r=1}^{R} \alpha_r = 1,$$

where each $p_r(\mathbf{z})$ denotes the probability computed by the r-th replicated circuit sharing the same scope, and α_r represents the convex weight of the root-level sum node. Because all repetitions have identical scopes, this integration by the sum node preserves the smoothness property of the circuit.

2.4 Vision Transformers

We prepared two parameter-frozen Vision Transformers, DINO[17] and CLIP[16]. Each ViT is connected separately to a probabilistic circuit, and the patch embedding vectors produced by the two models are treated as distinct independent random variables, even if they are derived from the same patch region. The final joint probability is obtained by a product node that combines the DINO–PC and the CLIP–PC:

$$p_{\text{root}}\big(\{\mathbf{z}_i^{\text{D}}\}_{i=1}^N, \{\mathbf{z}_i^{\text{C}}\}_{i=1}^N\big) \;=\; \prod_{i=1}^{N}\Big[\, p_{\text{root}}^{\text{DINO}}\big(\mathbf{z}_i^{\text{D}}\big)\; p_{\text{root}}^{\text{CLIP}}\big(\mathbf{z}_i^{\text{C}}\big)\Big],$$

where $\mathbf{z}_i^{\text{D}} \in R^{d_{\text{D}}}$ and $\mathbf{z}_i^{\text{C}} \in R^{d_{\text{C}}}$ denote the patch embedding vectors from DINO and CLIP for patch i, respectively, and $p_{\text{root}}^{\text{DINO}}$ and $p_{\text{root}}^{\text{CLIP}}$ are the distributions computed at the roots of the DINO–PC and CLIP–PC.

The above concept is represented in Figure 2.

2.5 Visualization of the model's decision process

During inference, we obtain class-wise evidence for any chosen *region* (a set of patch indices), *modality*, and *encoder* simply by running a forward pass with the corresponding visibility mask. We denote the returned class posterior by

$$p_{\mathcal{R},\mathcal{M},\mathcal{E}}(y = c),$$

where $\mathcal{R} \subseteq \{1, \ldots, N\}$ is a region (patch index set), $\mathcal{M}$ is a modality selection, and $\mathcal{E}$ is an encoder selection. Unselected inputs are treated as unobserved in the forward pass; no retraining is required.

Comparing encoders (e.g., DINO vs. CLIP). Fix a region $\mathcal{R}$ and a modality set $\mathcal{M}$, and compare

$$p_{\mathcal{R},\mathcal{M},\{\text{CLIP}\}}(y = c) \quad \text{vs.} \quad p_{\mathcal{R},\mathcal{M},\{\text{DINO}\}}(y = c).$$

A concise score is the log-posterior gap

$$\Delta_{\text{enc}}(c) := \log p_{\mathcal{R},\mathcal{M},\{\text{CLIP}\}}(y = c) - \log p_{\mathcal{R},\mathcal{M},\{\text{DINO}\}}(y = c),$$

which directly ranks encoders for class c.

Comparing modalities. Similarly, with an encoder set $\mathcal{E}$ fixed, compare

$$p_{\mathcal{R},\{m_1\},\mathcal{E}}(y = c) \quad \text{vs.} \quad p_{\mathcal{R},\{m_2\},\mathcal{E}}(y = c),$$

and summarize by

$$\Delta_{\text{mod}}(c; m_1, m_2) := \log p_{\mathcal{R},\{m_1\},\mathcal{E}}(y = c) - \log p_{\mathcal{R},\{m_2\},\mathcal{E}}(y = c).$$

Comparing regions / patches. For regional (or patch-wise) importance, we use two complementary probes:

(i) Keep-only posterior for a patch n:

$$p_{\{n\},\mathcal{M},\mathcal{E}}(y = c),$$

which tells how informative patch n is in isolation.

(ii) Patch-wise class-likelihood difference for patch n:

$$\Delta_{\text{class-difference}}(n) := \log p_{\{n\},\mathcal{M},\mathcal{E}}(y = c_2) - \log p_{\{n\},\mathcal{M},\mathcal{E}}(y = c_1),$$

which indicates whether patch n provides stronger evidence for class c_2 or for class c_1. Either quantity can be rendered as a heatmap over the patch grid.

2.6 Datasets for 2D image classification

We prepared a brain tumor classification dataset from MRI images, BRISC2025[18], and a pneumonia classification dataset from chest X-ray images, ChestX-ray8[19], to demonstrate the tractability and classification ability of our model.

BRISC2025 has 4 specific classes, including no tumor, glioma, meningioma, and pituitary, with 1067, 1147, 1329, and 1457 training images, and 140, 254, 306, and 300 validation images, respectively. We integrated all the training and validation images except the no-tumor class into a single tumor class. Then, the training samples in the no-tumor class were oversampled so that the ratio of the two classes is nearly equal. This dataset has ground truth segmentation masks corresponding to tumor regions for every image in the tumor class.

We used a kaggle dataset, HAM10000 in ISIC archive.[21][20] We converted the task into a binary classification problem by defining melanoma (mel) as the positive class and merging all remaining categories (akiec, bcc, bkl, df, nv, and vasc) into a single negative class.

ChestX-ray8 comprises 2 specific classes, normal and pneumonia, with 1341 and 3875 training images, 8 and 8 validation images, and 234 and 350 test images, respectively. For this dataset, we did not make any modifications and use it directly.

In addition, we prepared a simpler dataset for anomaly detection, the MVTec dataset.[22] We used two of its categories, zipper and tile, comprising two specific classes, good and NG.

Mvtec-zipper includes 240 training images for the good class, and 32 and 119 test images for the good and NG classes, respectively. We selected the first 21 images from the NG test set and used them as training images for the NG class. The training NG images were oversampled so that the ratio of the two classes was nearly equal. All the remaining test images were used as validation images.

MVTec Tile contains 230 training images for the good class, and 33 and 84 test images for the good and NG classes, respectively. We selected 26 images and 5 images from the NG test dataset and 5 images from the good training dataset, and used them as training images and validation images, respectively. The training NG images were oversampled so that the ratio of the two classes was nearly equal.

2.7 Experimental setting for 2D image classification

In all experiments, the probabilistic circuits (PCs) were initialized with the following default configuration: a depth of 5, 64 latent variables, 3 repetitions, 2 mixture pieces per leaf, unless explicitly stated otherwise. Two vision transformers, DINO ViT-B/16 and CLIP ViT-B/16, were used as the backbones of the circuits. Since these ViTs are optimized for 224×224 images, all input images were resized to this resolution. As a result, each input image was divided into 196 patches. The Adam optimizer [29] was adopted for training.

During training, we set a threshold of 0.5 for the anomaly class, then calculated the accuracy on the validation dataset, and saved the parameters that achieved the maximum accuracy. Gaussian, speckle, and salt-and-pepper noise are applied to the input images as part of data augmentation.

2.8 Brain age classification using multimodal 3D MRI images

VPNet can extend its scope to both multi-dimensional and multi-modal classification tasks beyond conventional two-dimensional and single-modal image classification without losing tractability or explainability. To demonstrate this capability, we implemented a 3D multi-modal image classification experiment. Specifically, we used the IXI 3D brain MRI dataset, which contains T1- and T2-weighted MRI volumes from 619 individuals along with patient information of age. [23]. We grouped the subjects into two age categories: *age1* (19–33 years) and *age4* (60–83 years). Any volumes outside these age ranges were excluded from the dataset. The *age1* group contained 101 training volumes, 14 validation volumes, and 30 test volumes, while the *age4* group contained 94 training volumes, 14 validation volumes, and 30 test volumes.

Each 3D MRI volume in each modality was first divided into individual 2D slices. Then, every slice was converted into multiple patch embedding vectors by two vision transformers (ViTs): DINO and CLIP. All patch vectors derived from the same 3D volume share a single probabilistic circuit. This concept is illustrated in Figure 3.

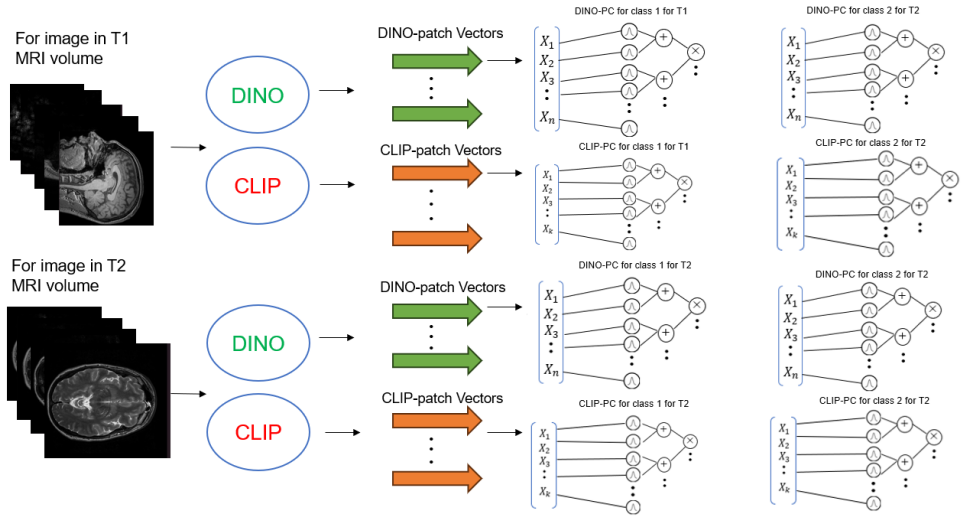


Figure 3: Overview of the 3D multimodal classification framework. Each 3D MRI volume is divided into slices, and each slice is transformed into patch embedding vectors by DINO and CLIP. The number of probabilistic circuits is eight, corresponding to two modalities, two encoders, and two classes.

To maintain the probabilistic interpretability, the joint posterior probability should ideally be used for optimization. However, as with most deep learning models, computing gradients through a single forward–backward pass requires storing all intermediate activations, which consumes significant memory. To address this issue, we divided the optimization into two forward passes and one backward pass.

In the first forward pass, the parameters of all probabilistic circuits were frozen, and only the joint posterior probability $P(c \mid \mathbf{Z})$ was computed through all inputs *without* storing any intermediate state.Then, the overall loss was computed as the sum of the cross-entropy loss and the entropy regularization term as follows:

$$\mathcal{L}(\mathbf{Z}, y) = -\left[\log p(y \mid \mathbf{Z}) \; - \; \tau_{\text{ent}} \sum_{c=1}^{C} p(y = c \mid \mathbf{Z}) \, \log p(y = c \mid \mathbf{Z})\right],$$

where τ_{ent} is a hyperparameter that controls the contribution of the entropy regularization term.

In the second forward pass, the same patch vectors were re-input into the probabilistic circuits. Based on the intermediate states of each circuit and the overall loss function defined above, the partial gradients of the loss with respect to the parameters of each probabilistic circuit were computed as follows:

$$\frac{\partial \mathcal{L}(\mathbf{Z}, y)}{\partial \theta_c} = -\frac{\partial}{\partial \theta_c}\left[\log p(y \mid \mathbf{Z};\, \theta_c) \; - \; \tau_{\text{ent}} \sum_{c'=1}^{C} p(y = c' \mid \mathbf{Z};\, \theta_c) \, \log p(y = c' \mid \mathbf{Z};\, \theta_c)\right],$$

This partial gradient was computed for all the patch vectors and averaged as the overall gradient as follows:

$$\frac{\partial \mathcal{L}^{(\text{overall})}}{\partial \theta_c} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial \mathcal{L}(\mathbf{z}_n, y)}{\partial \theta_c},$$

where N denotes the total number of patch embedding vectors in the case. Finally, the parameters of the probabilistic circuits were updated based on this overall gradient.

This process was repeated for all the samples within an epoch.

3 Results

3.1 Overall Classification Performance for 2D image classification

The joint posterior without any unobserved state of patch or ViT is used for the classification task. Table 1 shows the AUROC, Average Precision (AP), and maximum accuracy from the experimental

Table 1: Test metrics for each dataset. Values are AUROC, Average Precision (AP), and Max Accuracy with the threshold used.

Dataset	AUROC	AP	Max Acc. @ thr
BRISC2025 (2-class)	0.9975	0.9996	0.983 @ 0.3897
MVTec (tile)	**1.000000**	**1.000000**	**1.000000 @ 0.5103**
ChestX-ray8	0.9718	0.9804	0.9231 @ 0.7880
ISIC archive	0.9592	0.9530	0.8949 @ 0.292

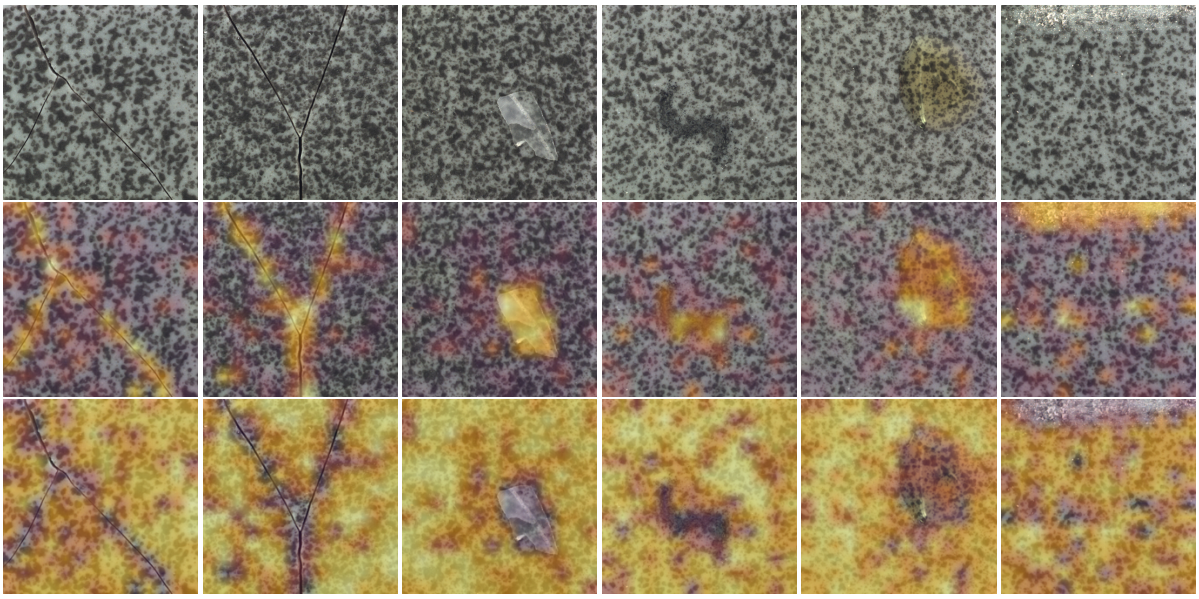


Figure 4: Three rows (top to bottom): original, heatmap (NG), heatmap (good). Six columns show different samples in MVTec-tile.

datasets.

3.2 Patch-level Probabilistic Contribution for 2D image classification

3.2.1 Posterior-based visualization

Figure 4 illustrates two inferno heatmaps [30] based on the class posteriors for several anomaly images in the test dataset of MVTec Tile, together with the corresponding original images. It is observed that both the dark purple regions in the heatmap from the negative class and the dark red regions in the heatmap from the positive class cover the defective parts in the originals. This indicates that the basis of the model's decision concentrates on the defective areas.

3.2.2 Likelihood-based visualization

Instead of class posteriors, Figure 5 from the test set of the ISIC archive and Figure 6 from BRISC 2025 integrate the patch-level probabilistic contributions into a single heatmap using the patch-wise class-likelihood difference defined in Section 2.5.

In Figure 5, brighter regions are observed to concentrate on the lesion areas of the images, whereas some regions within the lesions are highlighted by darker colors. This indicates that the model distinguishes between more melanoma-like and less melanoma-like regions within lesions and assigns corresponding probabilistic contributions.

On the other hand, some heatmaps from the test set of the ISIC archive highlight tumor regions with strong bright colors, whereas others strongly highlight the background, as shown in Figure 6. This suggests that the model adapts its focus depending on the input image.

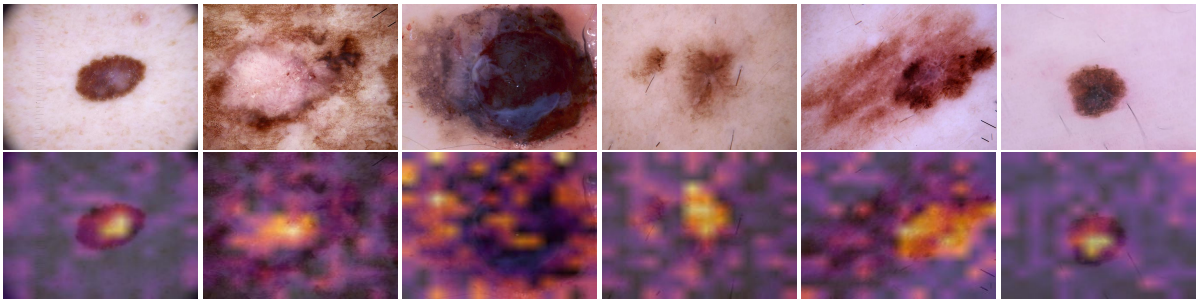


Figure 5: Likelihood-based heatmaps on the ISIC test set. The top row shows the original images, and the bottom row shows the corresponding heatmaps based on patch-wise likelihood differences. Brighter regions indicate stronger contributions toward the melanoma class.

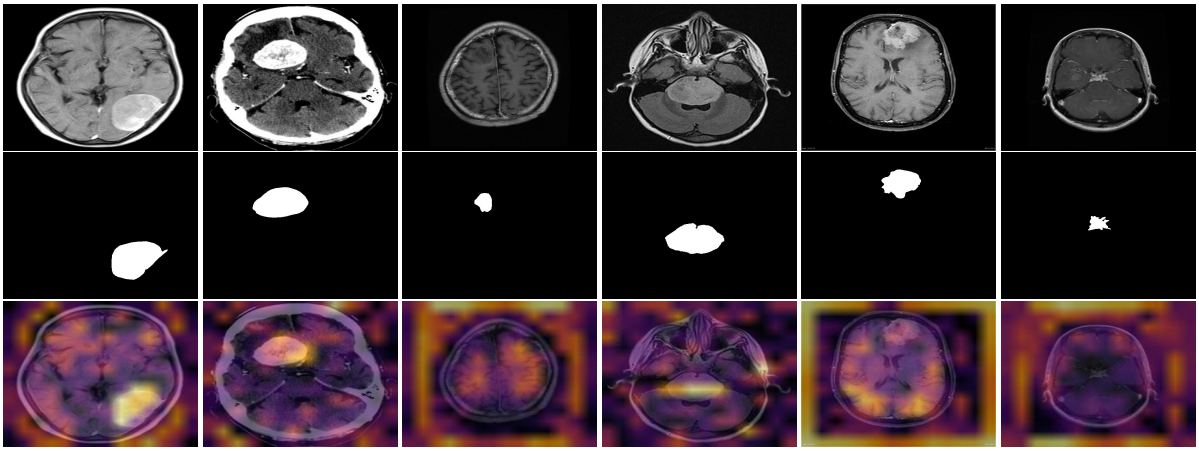


Figure 6: Likelihood-based heatmaps on the BRISC 2025 dataset. The top row shows the original images, the middle row shows the ground-truth annotations, and the bottom row shows the corresponding heatmaps based on patch-wise likelihood differences.

3.2.3 Randomized test

We checked whether the heatmaps were obtained during the training process by comparing them with heatmaps from a model with randomly initialized probabilistic circuits. Figure 7 presents the differences in the heatmaps after training for likelihood from both DINO and CLIP, DINO only, and CLIP only, respectively. This figure supports the following findings: (a) the heatmaps of the classification model are obtained through optimization and do not originate from the attention maps of the vision transformers, (b) the DINO circuits and the CLIP circuits learn different decisions independently during training, and the representations they capture are fundamentally different.

3.3 Overall Classification Performance for 3D multimodal image classification

We optimized this model for 30 epochs with the following hyperparameters: a depth of 5, 64 latent variables, 3 repetitions, and 2 mixture pieces per leaf. Figure 8 displays the confusion matrices for the training, validation, and test samples, based on the posterior probabilities computed from the three input settings: T1+T2, T1 only, and T2 only.

The best-performing modality was T1, achieving the highest accuracy in both training and test domains, with a maximum training accuracy of 1.00 and a test accuracy of 0.98. This demonstrates both strong expressiveness and robustness of the model. In contrast, the T2 modality exhibited lower accuracy (0.943 for training and 0.916 for testing), indicating slight underfitting compared with the T1 modality. These results suggest that, for this model, the information contained in the T1 modality is sufficient for the classification task, while the T2 modality may introduce additional noise. Indeed, the combined posterior from T1+T2 achieved a training accuracy of 0.969 and a test accuracy of 0.95, which was slightly worse than using T1 alone. The distributions of the posterior probabilities for the *age1* class are shown in Figure 9.

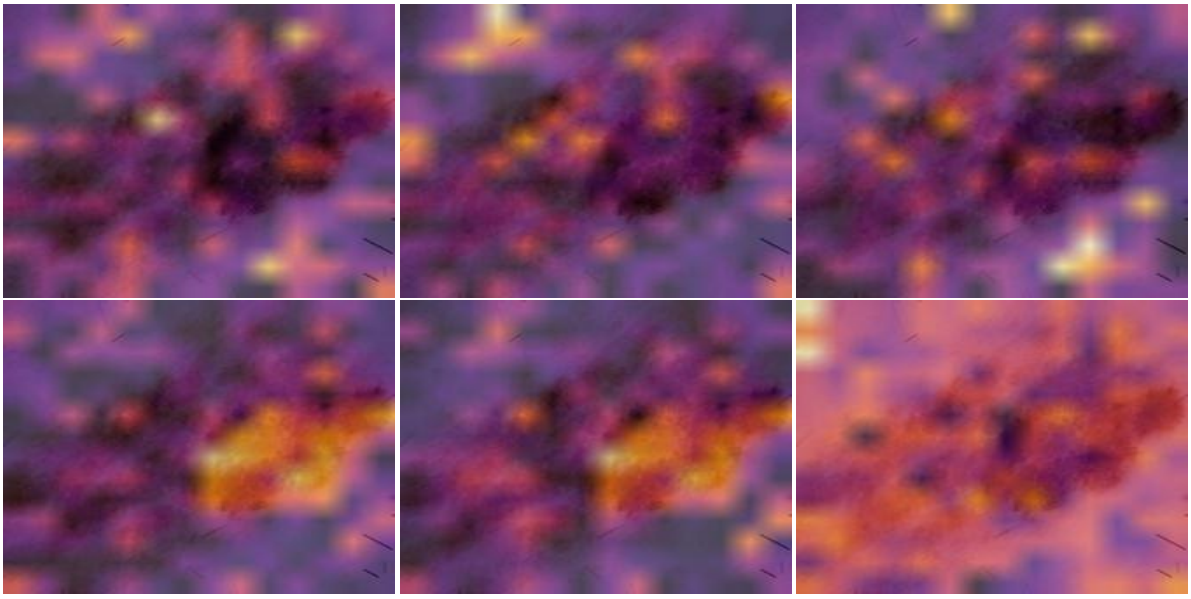


Figure 7: Comparison of likelihood-based heatmaps from randomly initialized and trained probabilistic circuits. The top row shows heatmaps obtained from randomly initialized circuits, while the bottom row shows those obtained after training. From left to right, the columns correspond to DINO+CLIP, DINO, and CLIP, respectively.

3.4 Patch-level Probabilistic Attribution for 3D multimodal image classification

As described in Section 2.5, our model can visualize patch-level contributions to the classification decision even in the 3D multimodal setting. To demonstrate this capability, we visualized the contribution score of every patch across all slices of the input MRI volume. One example of the resulting heatmaps are shown in Figures 10–13.

4 Discussion and Conclusion

4.1 Comparison with other XAI models

As some studies indicate, widely used XAI methods, such as gradient-based approaches including Grad-CAM and attention-based approaches, do not have solid theoretical validity regarding whether the saliency maps truly represent the model's decision process.[10][9] Additionally, it is well known that such methods often fail the randomized test; when the parameters of the model are randomly initialized, the saliency maps remain largely unchanged, suggesting that the saliency maps are a product of the feature maps rather than an outcome of the training process.[9]

On the other hand, our proposed model is not on the same playing field as these methods, as our approach does not aim to make a black-box model explainable. Instead, the explainability of our approach is theoretically guaranteed, and the explainable model is directly optimized for a specific task. As described in Section 3.2.3, our model can overcome the randomized test, clearly indicating that the saliency maps produced by our model are an outcome of the optimization process. In addition, by combining foundation models with strong representational ability and probabilistic circuits, our model compensates for one of the main weaknesses of probabilistic circuits, namely their limited representational capability. This is supported by the high classification performance across various datasets, including complex tasks such as melanoma classification.

4.2 Encoder-level selection of information

During the training process, VPNet can assign higher weights to more important information. For instance, as described, this property enables explanations of more or less important patches in images. In addition, our proposed method allows encoder-level separation of information. In most cases, it is difficult to identify which encoders are more useful for training or to characterize differences in

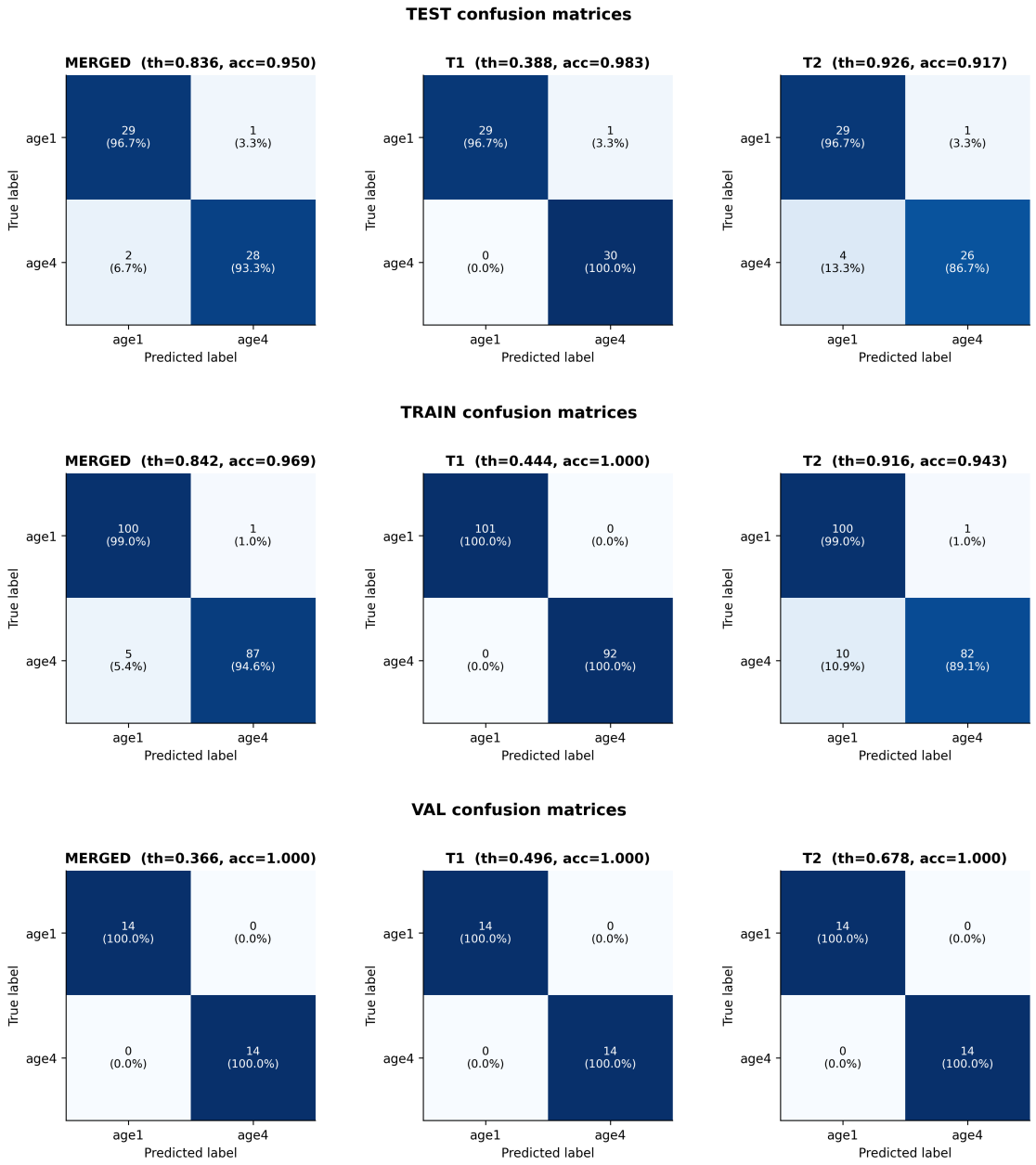


Figure 8: Confusion matrices for TEST (top), TRAIN (middle), and VAL (bottom) sets. Each row shows MERGED, T1-only, and T2-only results.

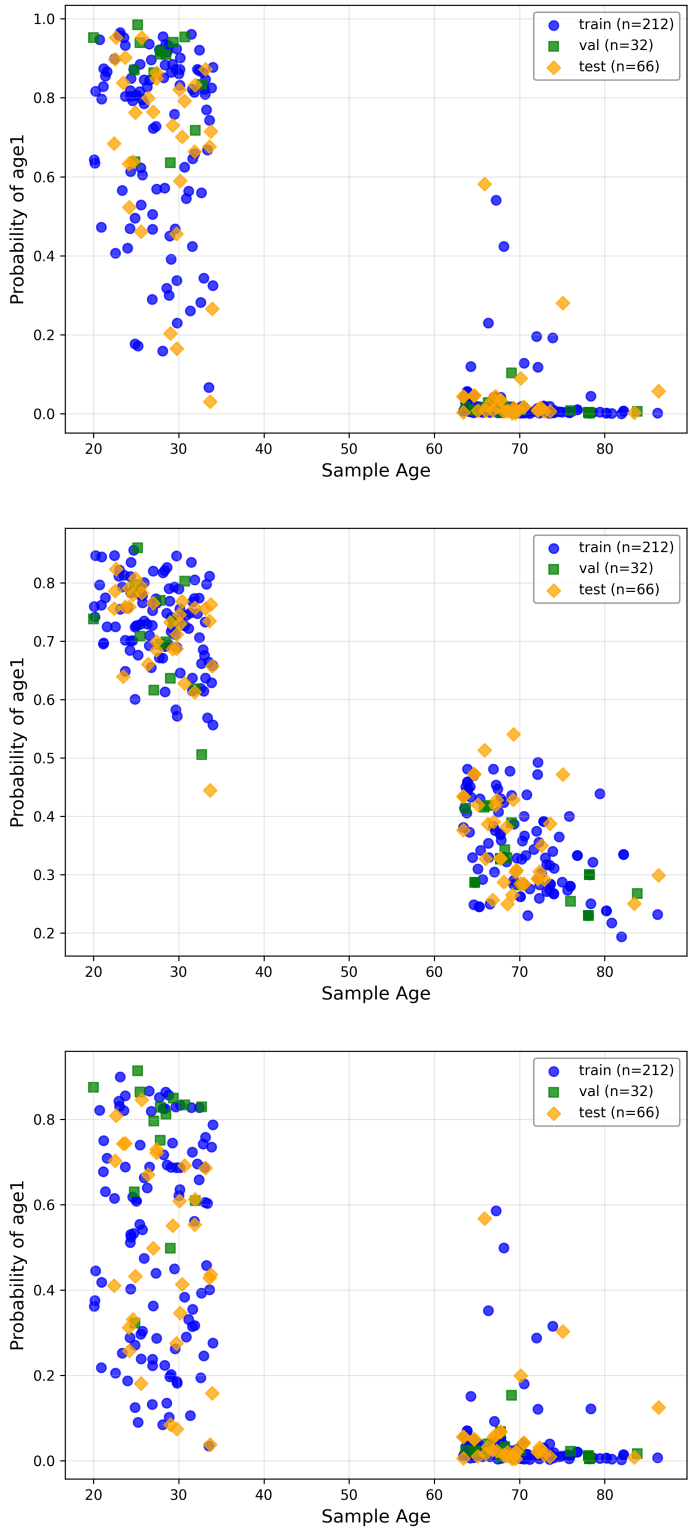


Figure 9: Scatter plots of posterior probability $P(\text{age1})$ versus chronological age for MERGED (top), T1-only (middle), and T2-only (bottom).

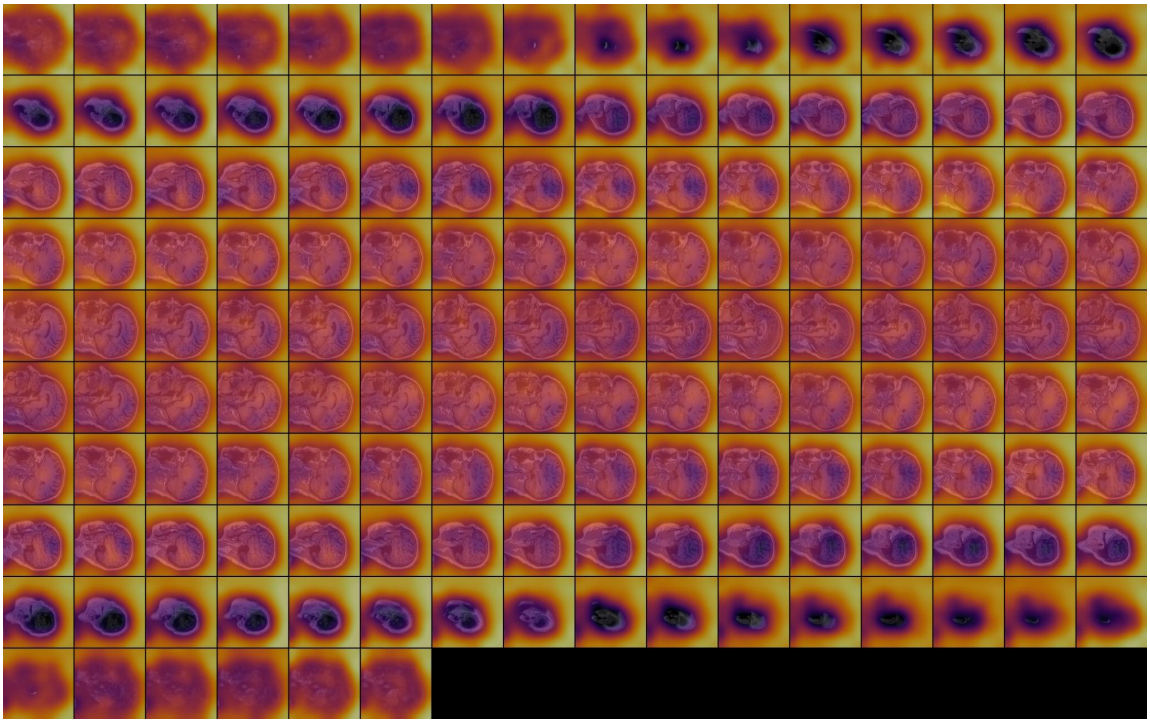


Figure 10: Patch-level contribution heatmap for a sample (true label: age group4 (age range 60-83)). Visualization generated from the T1 modality for the target class **age1**.

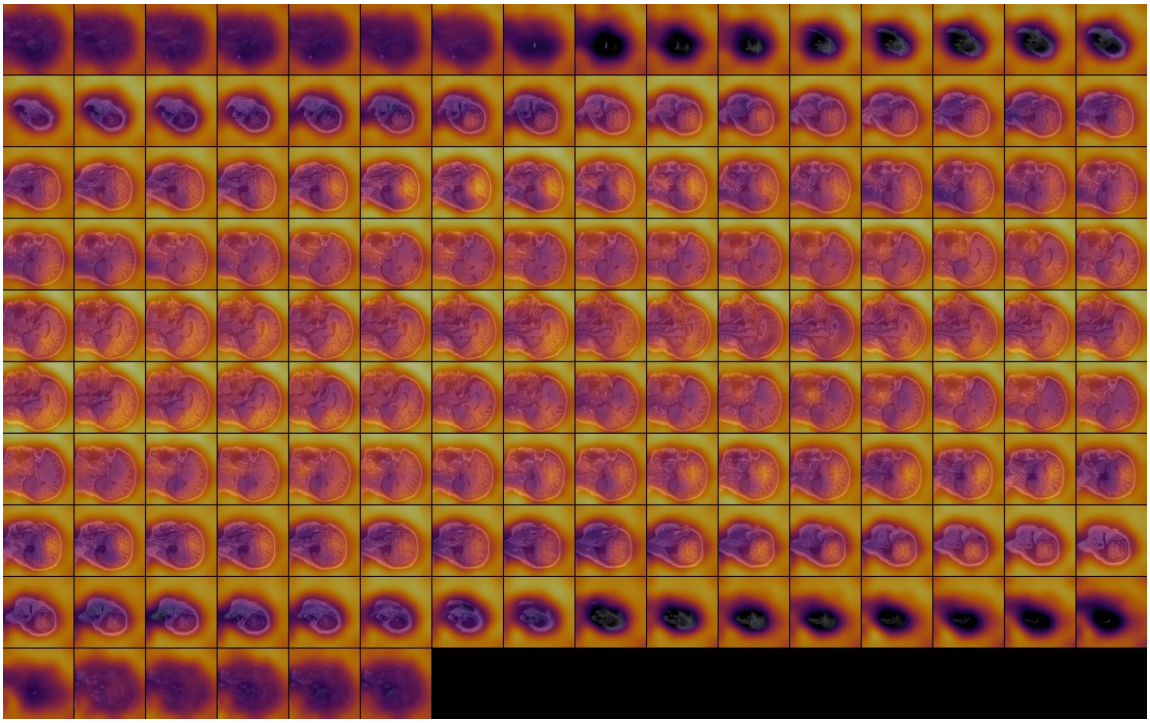


Figure 11: Patch-level contribution heatmap for a sample (true label: age group4 (age range 60-83)). Visualization generated from the T1 modality for the target class **age4**.

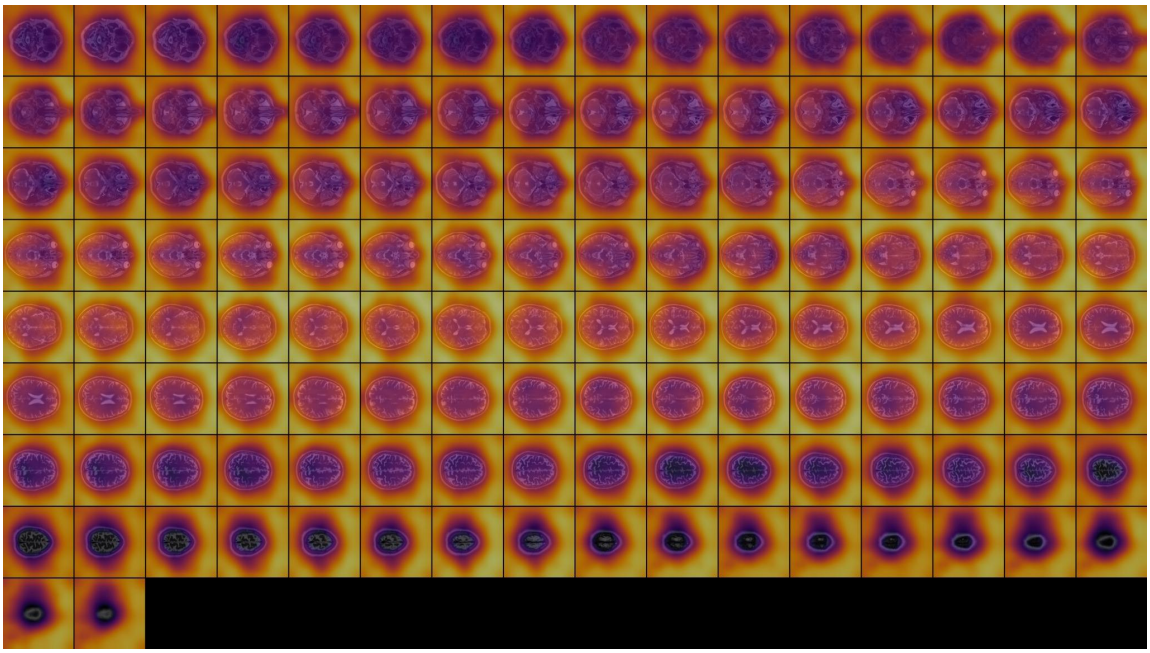


Figure 12: Patch-level contribution heatmap for a sample ((true label: age group4 (age range 60-83)). Visualization generated from the T2 modality for the target class **age1**.

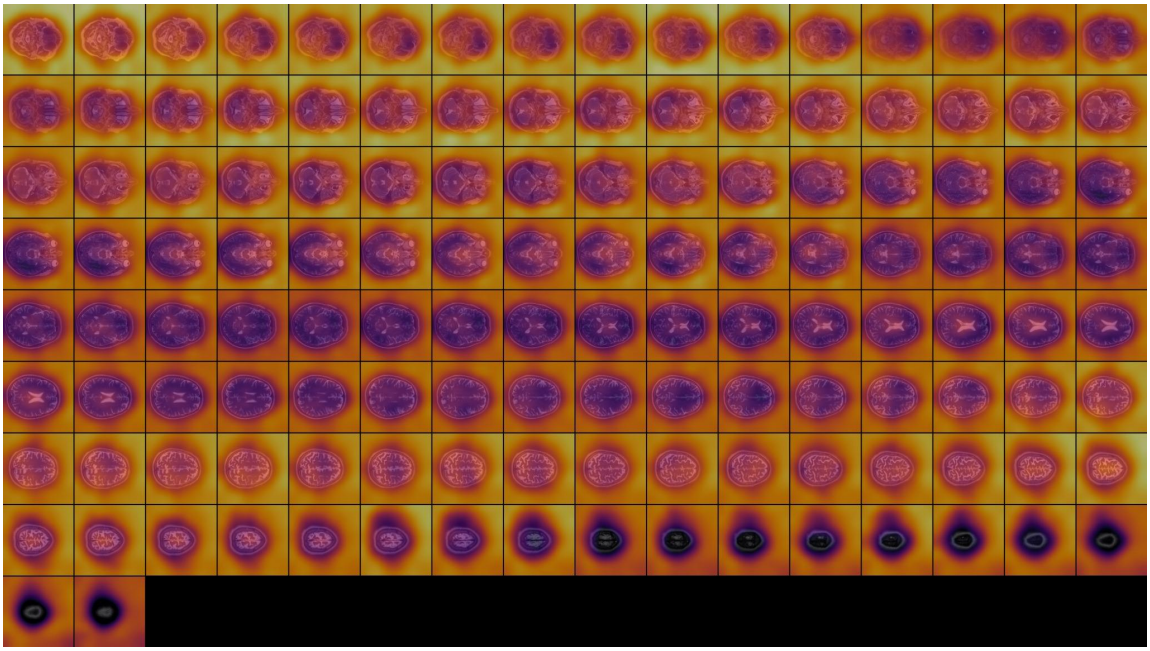


Figure 13: Patch-level contribution heatmap for a sample ((true label: age group4 (age range 60-83)). Visualization generated from the T2 modality for the target class **age4**.

their behavior. Even when comparing usefulness or behavior between two independent models, the comparison can be noisy due to differences in optimization conditions. However, in our proposed method, independent probabilistic circuits corresponding to each encoder are constructed, and each circuit is connected to a production node. This design enables encoder-level informative separation under the same optimization conditions. Indeed, Figure 7 shows that the representations learned by each ViT are completely different between DINO and CLIP. DINO focuses on the lesion, whereas CLIP's attention is more spread across the image. Furthermore, the heatmap obtained from the likelihood combining two vision transformers is similar to that of DINO, suggesting that the model learned to treat information from CLIP as less important during optimization.

4.3 Encoder-level Information Selection

In a manner similar to encoder-level information selection, VPNet can learn which input modalities are more important during optimization under the same training conditions. It also enables users to make selective decisions during inference by visualizing probabilistic contributions or classification performance. In both industry and research, it is often difficult to determine which information is most useful when developing a classification model; from this perspective, VPNet offers significant utility.

4.4 Evaluation of Benchmarks

Although our model shows strong performance on the BRISC2025 dataset, some saliency maps highlight background regions rather than brain regions. Unless tumor patients exhibit a consistent structural bias in brain geometry, this observation suggests that the dataset may not accurately represent the intended classification task between tumor and non-tumor patients. Instead, the model may be leveraging spurious correlations, such as differences in age, sex, or imaging conditions. From this perspective, our model has the potential to serve as a tool for assessing the validity of benchmark datasets.

4.5 Extensibility

The functionality of our proposed method is not limited to image classification tasks. It can be readily extended to various explainable classification tasks by replacing patch-embedding vectors with embeddings from other foundation models. Indeed, we proposed DVPNet [31], a gene classification framework in which the model identifies which genes are inherently important for biological classification tasks using a biological foundation model, the Nucleotide Transformer [32].

4.6 Summary

In this study, we proposed a novel explainable classification framework and demonstrated clear advantages over current state-of-the-art XAI methods, such as gradient-based and attention-based approaches, in terms of explainability, information selection, and extensibility.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] C. Patrício, J. C. Neves, and L. F. Teixeira, "Explainable deep learning methods in medical image classification: A survey," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–41, 2023.

[4] P. Sharma, A. Singh, and S. Kumar, "Enhancing brain tumor detection in mri images through explainable ai using grad-cam with resnet50," *BMC Medical Imaging*, vol. 24, no. 1, p. 92, 2024.

[5] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 159–168, 2017.

[6] M. Bohle, F. Eitel, M. Weygandt, and K. Ritter, "Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer's disease classification," *Frontiers in Aging Neuroscience*, vol. 11, p. 194, 2019.

[7] M. Komorowski *et al.*, "Towards evaluating explanations of vision transformers for medical imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 6133–6142, 2023.

[8] A. Rao, J. Park, S. Woo, J.-Y. Lee, and O. Aalami, "Studying the effects of self-attention for medical image analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3416–3425, 2021.

[9] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, vol. 31, 2018.

[10] 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)*, pp. 3543–3556, 2019.

[11] Y. Choi, A. Vergari, and G. Van den Broeck, "Probabilistic circuits: A unifying framework for tractable probabilistic models." http://starai.cs.ucla.edu/papers/ProbCirc20.pdf, 2020. UCLA Technical Report.

[12] S. Braun, S. Sidheekh, S. Natarajan, A. Vergari, M. Mundt, and K. Kersting, "Autoencoding probabilistic circuits," in *Eighth Workshop on Tractable Probabilistic Modeling*, 2025.

[13] W. Chen, S. Yu, H. Shao, L. Sha, and H. Zhao, "Neural probabilistic circuits: An overview," in *Eighth Workshop on Tractable Probabilistic Modeling*, 2024.

[14] A. Ciotinga and Y. Choi, "Optimal transport for probabilistic circuits," *arXiv preprint arXiv:2410.13061*, 2024.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020. arXiv preprint arXiv:2010.11929.

[16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. arXiv preprint arXiv:2103.00020.

[17] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. arXiv preprint arXiv:2104.14294.

[18] A. Fateh, Y. Rezvani, S. Moayedi, S. Rezvani, F. Fateh, M. Fateh, and V. Abolghasemi, "Brisc: Annotated dataset for brain tumor segmentation and classification with swin-hafnet," *arXiv preprint arXiv:2506.14318*, 2025.

[19] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

[20] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.

[21] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset: a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, p. 180161, 2018.

[22] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1038–1059, 2021.

[23] IXI – Information eXtraction from Images Project, "Ixi dataset." https://brain-development.org/ixi-dataset/, 2006. Accessed: 2025-11-12.

[24] R. Peharz, A. Vergari, K. Stelzner, A. Molina, X. Shao, M. Trapp, K. Kersting, and Z. Ghahramani, "Random sum-product networks: A simple and effective approach to probabilistic deep learning," in *Uncertainty in Artificial Intelligence*, pp. 334–344, PMLR, 2020.

[25] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[26] A. Liu and G. Van den Broeck, "Tractable regularization of probabilistic circuits," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3558–3570, 2021.

[27] N. Friel and A. N. Pettitt, "Marginal likelihood estimation via power posteriors," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 70, pp. 589–607, 04 2008.

[28] A. Liu, K. Ahmed, and G. V. d. Broeck, "Scaling tractable probabilistic circuits: A systems perspective," *arXiv preprint arXiv:2406.00766*, 2024.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[30] K. Moreland, "Diverging color maps for scientific visualization," in *Advances in Visual Computing (ISVC)*, vol. 5876 of *Lecture Notes in Computer Science*, pp. 92–103, Springer, 2009.

[31] T. Kusumoto, "Dvpnet: A new xai-based interpretable genetic profiling framework using nucleotide transformer and probabilistic circuits," *bioRxiv*, pp. 2026–01, 2026.

[32] H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, N. Lopez Carranza, A. H. Grzywaczewski, F. Oteri, C. Dallago, E. Trop, B. P. De Almeida, H. Sirelkhatim, *et al.*, "Nucleotide transformer: building and evaluating robust foundation models for human genomics," *Nature Methods*, vol. 22, no. 2, pp. 287–297, 2025.