VPNet: Tractable and Explainable Multimodal and Multidimensional Classification with Probabilistic Circuits on ViT Features

Taishi Kusumoto

November 15, 2025

Abstract

We propose a novel explainable AI (XAI) model for classification tasks that can treat 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. To demonstrate this capability, we developed a three-dimensional multimodal classification model for the brain age classification using 3D brain MRI volumes with T1 and T2 modalities. This model achieved 0.98 accuracy on the test domain while visualizing the patch-level probabilistic contribution. We expect that, in the future, scientists may uncover new findings using such explainable classification models by examining their decision processes.

Our source code and training platform are available at https://github.com/tai2456/VPNet-Tractable-and-Explainable-Classification-with-Probabilistic-Circuits-on-ViT-Features, where users can train their own classification models on their own datasets.

1 Introduction

Medical or biological image classification with Artificial Intelligence (AI) has great potential to help innovative medical discoveries by uncovering hidden relationships between the target classes while extracting important feature representations from the images. However, most of such classification models lack interpretability due to their black-box characteristics. This complicates the introduction of such classification models for both practical and research purposes.

Explainable AI (XAI) models, which aim to clarify their decision processes, have been an active area of research, with various methods and algorithms proposed,[1] including Grad-CAM-based approaches[2], LRP-based approaches[3][4], and attention map–based approaches[5][6]. However, these methods merely visualize the complex feature representations of the model, without providing theoretical evidence that the visualization directly corresponds to the actual decision-making process of the model.

To address this limitation, we introduce an innovative classification model, VPNet, in which patch embedding vectors from parameter-frozen vision transformers (ViTs)[7] are directly connected to probabilistic circuits (PCs)[8][9][10][11]. The parameters of the PCs are then optimized for a specific classification task under supervised learning. This approach simultaneously maintains probabilistic interpretability and representation ability. In this paper, we demonstrate that VPNet achieves outstanding performance in image classification tasks and can visualize the probabilistic contribution of each patch in the image and/or each vision transformer connected to the PCs. We argue that this innovative approach represents a next-generation framework, enabling scientists to leverage artificial intelligence models to uncover new relationships in the real world.

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

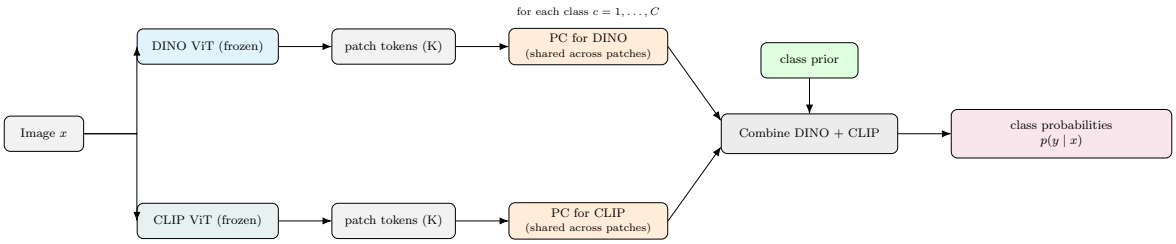


Figure 1: Simplified overview of the class-conditional PC classifier.

corresponds to a single random variable, and (b) smoothness and decomposability are preserved.[8] 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 of the preservation of smoothness and decomposability, the joint probability distribution of the circuit, in other words, the probability distribution produced by the product node integrating all the scopes, is equivalent to the joint distribution of the patch vector. Formally, this relationship can be expressed as

$$p(\mathbf{z}) \;=\; \prod_{i=1}^{D} p_{s_i}(z_i), \qquad S_i \cap S_j = \emptyset\; \forall\, i \neq j,$$

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 [12], 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, patch vectors belonging to the same modality, encoder, and class share the same probabilistic circuit. Accordingly, the expression of the joint probability distribution of a patch vector can be further clarified as

$$p(\mathbf{z}, \text{modality, encoder, class}) \;=\; \prod_{i=1}^{D} p_{s_i}(z_i), \qquad S_i \cap S_j = \emptyset \;\forall\, i \neq j,$$

During optimization, the joint probabilistic distribution from all the patch vectors within a class is utilized. This overall distribution is computed as the product of the joint probability distributions of all patch vectors. This operation is equivalent to connecting the root probability of each patch vector through product nodes across their independent scopes, as

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

where $\mathbf{z}_n$ denotes the patch embedding vector of the n-th patch in the class, and each $\mathbf{z}_n$ has disjoint scopes from the others. This formulation guarantees that the joint probabilistic distribution of a single patch vector is equivalent to the marginal probabilistic distribution when all other patch vectors are unobserved, that is,

$$p(\mathbf{z}_n, \text{class}) \;=\; \int p(\mathbf{Z}, \text{class})\, \mathrm{d}\mathbf{Z}_{\setminus n},$$

where $\mathbf{Z}_{\setminus n}$ denotes all patch vectors except for $\mathbf{z}_n$. This ensures the **tractability** and the **explainability** of our model. Because the Vision Transformer (ViT) parameters are frozen, the encoder does not learn relationships across patches during optimization for the classification task. Thus, no additional dependencies are introduced outside the probabilistic circuit. This confirms that the marginal probability distribution of a patch vector is equivalent to the marginal probability distribution over the corresponding patch region, as follows:

$$p(\text{patch, class}) \;=\; p(\mathbf{z}_n, \text{class}).$$

In addition, the computation of a marginal probability distribution for an arbitrary **modality** or an arbitrary **encoder** is also possible, as

$$p(\text{modality, class}) = \int p(\text{all modalities, class})\, \mathrm{d}\mathbf{Z}_{\setminus \text{modality}},$$

$$p(\text{encoder, class}) = \int p(\text{all encoders, class})\, \mathrm{d}\mathbf{Z}_{\setminus \text{encoder}}.$$

Finally, the marginal probability distribution can be computed from any arbitrary combination of a **region**, **modality**, and **encoder**, as

$$p(\text{region, modality, encoder, class}) \;=\; \int p(\mathbf{Z}_{\text{all}}, \text{class})\, \mathrm{d}\mathbf{Z}_{\setminus (\text{region, modality, encoder})}.$$

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 probability rule allows us to express the class-conditional distribution of all patch vectors as

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

The class posterior can be computed from the Bayes rule as

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

This class posterior can be used for optimization of learnable parameters. As loss functions, we defined a cross-entropy loss [13] 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 [14],

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

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.

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**[12] algorithm under the PyJuice framework [15](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[16] and CLIP[17]. 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 1.

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) Drop-one effect relative to a base region $\mathcal{R}$:

$$\Delta_{\text{drop}}(c; n) := \log p_{\mathcal{R},\mathcal{M},\mathcal{E}}(y = c) - \log p_{\mathcal{R}\setminus\{n\},\mathcal{M},\mathcal{E}}(y = c),$$

which tells how much the evidence would deteriorate if patch n were removed. 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.

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.[20] 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 [21] 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.

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. [22]. 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 2.

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.

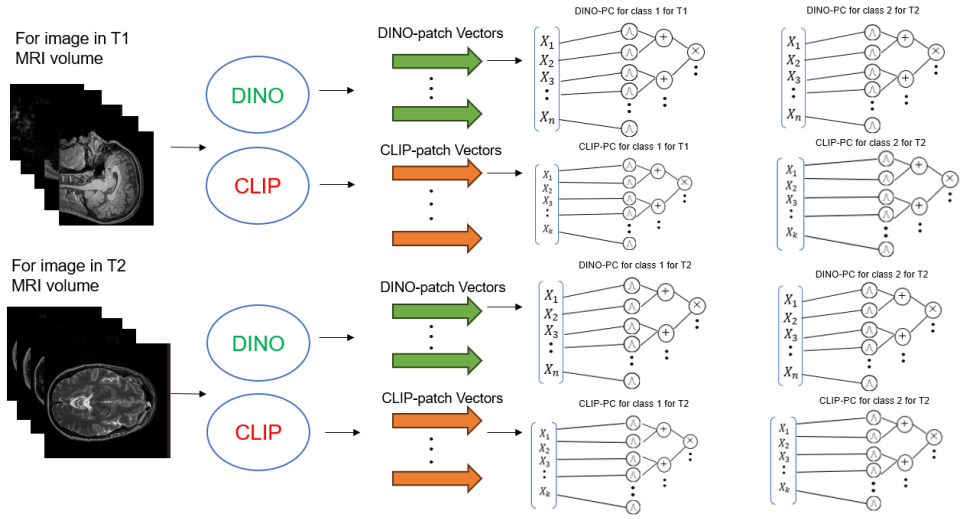


Figure 2: Overview of the 3D multimodal classification framework. Each 3D MRI volume is divided into slices, and each slice is transformed into patch embedding vectors by DINO and CLIP. The number of probabilistic circuits is eight, corresponding to two modalities, two encoders, and two classes.

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.

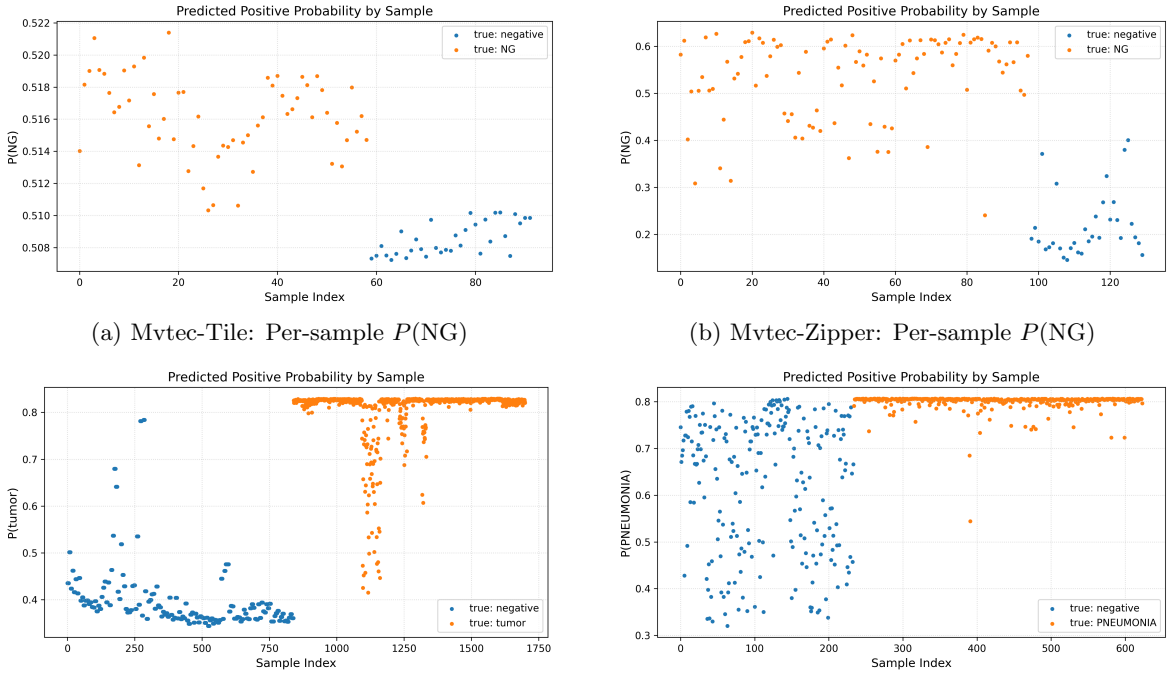
3 Results

3.1 Overall Classification Performance for 2D image classification

The joint probability 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 datasets. We calculated the evaluation metrics for MVTec-zipper and BRISC2025 using the validation dataset, and the test dataset for ChestX-ray8 and MVTec-tile. AUROC, Average Precision, and Accuracy are defined as follows:

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

2*Dataset	Validation (Val)			Test		
	AUROC	AP	Max Acc. @ thr	AUROC	AP	Max Acc. @ thr
BRISC2025 (2-class)	0.997251	0.997393	0.978235 @ 0.5370	—	—	—
MVTec (zipper)	0.991709	0.997337	0.953846 @ 0.2690	—	—	—
MVTec (tile)	—	—	—	**1.000000**	**1.000000**	**1.000000 @ 0.5103**
ChestX-ray8	—	—	—	0.971828	0.980382	0.923077 @ 0.7880



(a) Mvtec-Tile: Per-sample $P(\text{NG})$

(b) Mvtec-Zipper: Per-sample $P(\text{NG})$

(c) BRISC2025: Per-sample $P(\text{Tumor})$

(d) ChestX-ray8: Per-sample $P(\text{Pneumonia})$

Figure 3: Per-sample probability scatter plots for all datasets. Each panel shows the distribution of predicted probabilities for positive cases in its respective dataset.

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}), \tag{1}$$

$$\text{AP} = \sum_{n=1}^{N} (R_n - R_{n-1}) P_n, \tag{2}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3}$$

Here,

- TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively.
- $\text{TPR} = \dfrac{TP}{TP + FN}$ is the true positive rate (recall).
- $\text{FPR} = \dfrac{FP}{FP + TN}$ is the false positive rate.
- P_n and R_n denote the precision and recall at the n-th threshold, respectively.

Figure 3 presents the scatter plots of the conditional probabilities of the positive class for all datasets: MVTec Tile, MVTec Zipper, BRISC2025, and ChestX-ray8. Each panel corresponds to one dataset and shows the per-sample conditional probability of the positive class.

3.2 Patch-level Probabilistic Attribution for 2D image classification

We compared the conditional probabilistic distribution from each patch in images labeled as the positive class with respect to the negative and positive classes. We then visualized the difference using the inferno colormap.[23] Let $x = \{x_1, x_2, \ldots, x_K\}$ denote the set of patch features. For each class $c \in \{\text{pos}, \text{neg}\}$, the conditional probabilistic distribution of patch k is

$$p_c(x_k) = f_c(x_k), \tag{4}$$

where $f_c(\cdot)$ denotes the class-conditional density modeled by the probabilistic circuit. The per-class joint log-likelihood is given by

$$\log p_c(x) = \frac{1}{K} \sum_{k=1}^{K} \log p_c(x_k). \tag{5}$$

The patch-level difference is defined as

$$\Delta_k = \log p_{\text{pos}}(x_k) - \log p_{\text{neg}}(x_k). \tag{6}$$

In the inferno heatmap visualization, dark red indicates patches with high Δ_k (i.e., a strong contribution to increasing the class-conditional joint probability for the class), yellow indicates medium Δ_k, and dark purple indicates low Δ_k. This means that in the heatmap for the positive class, dark red regions correspond to abnormal areas, whereas dark purple regions indicate normal areas. In the heatmap for the negative class, the interpretation is reversed.

Figure 4 illustrates two heatmaps for several anomaly images in the test dataset of MVTec Tile, together with the corresponding original images. It can be 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.

Figure 5 depicts heatmaps for anomaly samples in the validation dataset of MVTec Zipper, along with the corresponding original images. Red squares are superimposed on the originals to mark defective regions. In this case, the dark purple areas in the negative-class heatmap successfully highlight the defects, and furthermore spread to adjacent defective and deformed areas. In contrast, although the dark red areas in the positive-class heatmap are theoretically expected to capture the defects, they fail to delineate them accurately.

In Figure 6, heatmaps for glioma images from the BRISC2025 validation dataset are presented for both the positive (tumor) and negative (no tumor) classes, together with the original images and the ground-truth masks. Unlike the industrial cases, the tumor regions are not clearly emphasized by the dark purple areas in the negative-class heatmap; instead, purple regions are more diffusely distributed throughout the brain.

Figure 7 shows examples of heatmaps for pneumonia cases from the ChestX-ray8 test dataset, corresponding to the positive (pneumonia) and negative (normal) classes, displayed alongside the original images. In this case, the distributions of the two heatmaps are notably different. The dark purple regions in the negative-class heatmap clearly concentrate over the entire chest, whereas the dark red regions in the positive-class heatmap accumulate in smaller areas around the neck, lower abdomen, and armpits.

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,

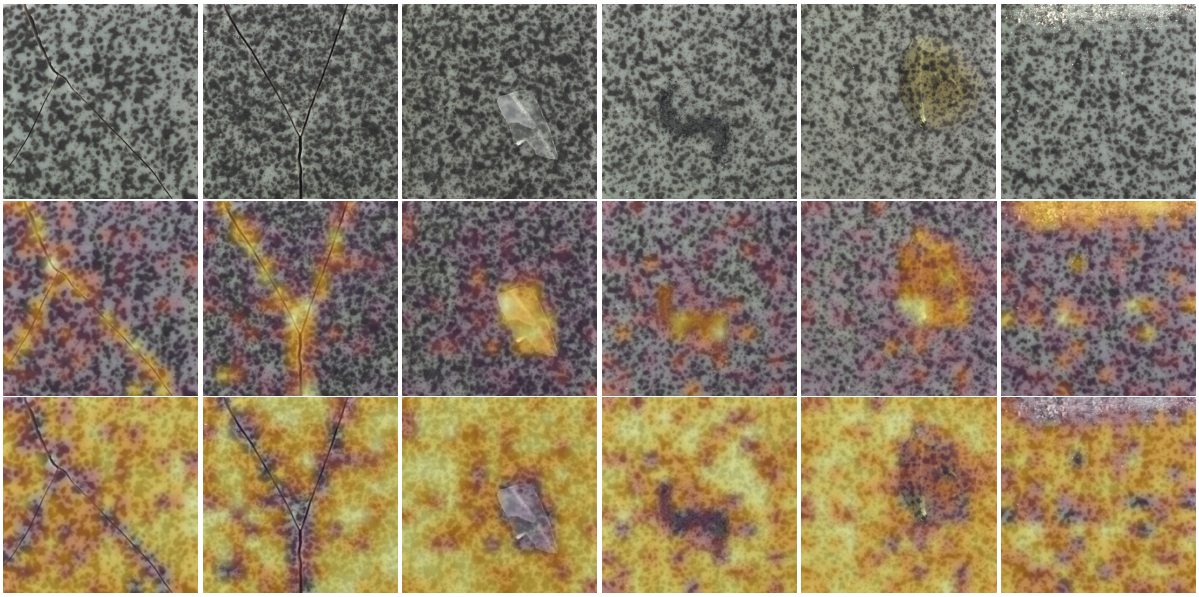


Figure 4: Three rows (top to bottom): original, heatmap (NG), heatmap (good). Six columns show different samples in MVTec-tile.

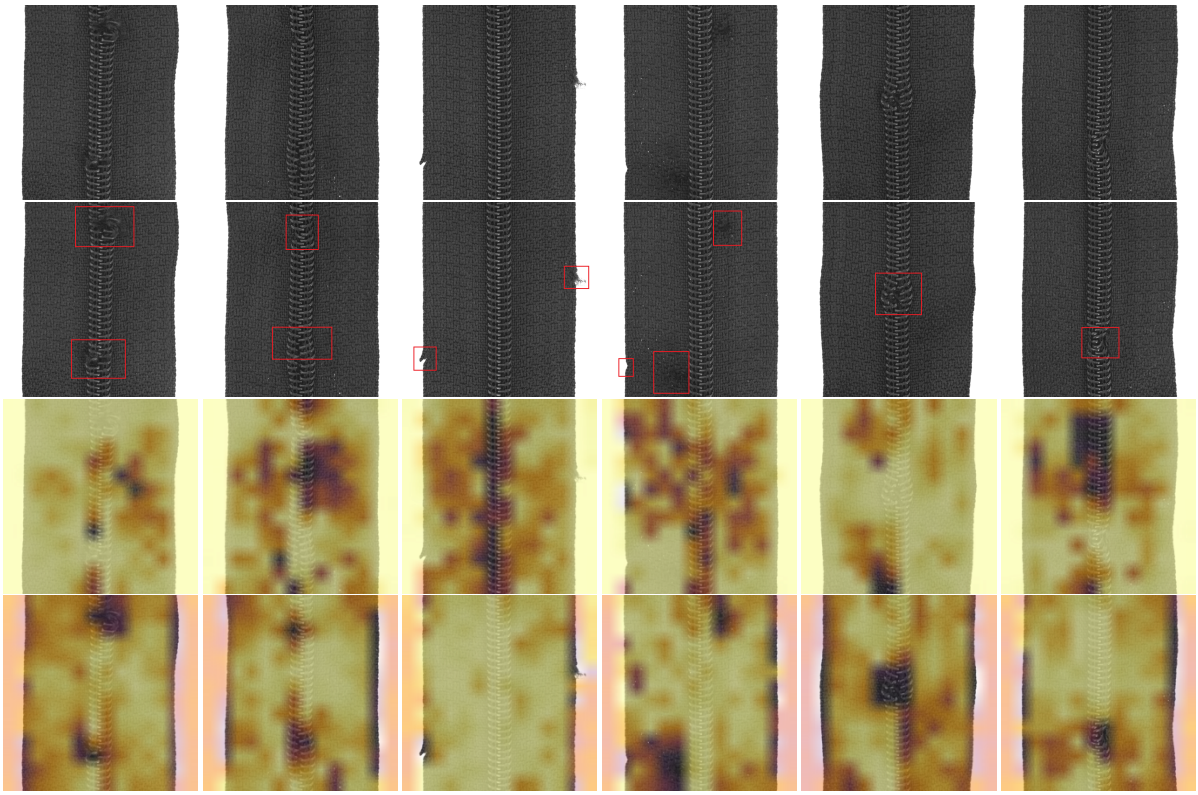


Figure 5: Four rows (top to bottom): original, original with defect boxes, heatmap (NG), heatmap (good). Six columns show different samples in MVTec-zipper.

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.

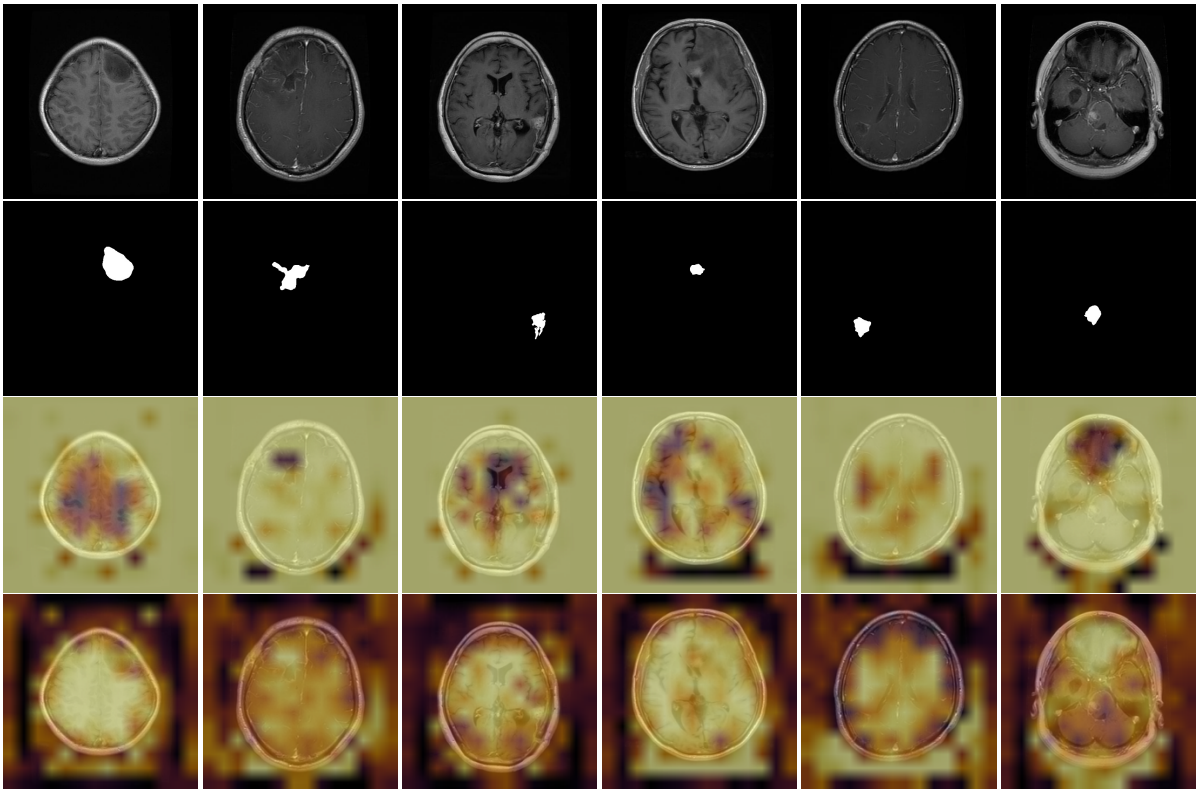


Figure 6: Four rows (top to bottom): original, original with defect boxes, heatmap (tumor), heatmap (no-tumor). Six columns show different samples in BRISC2025.

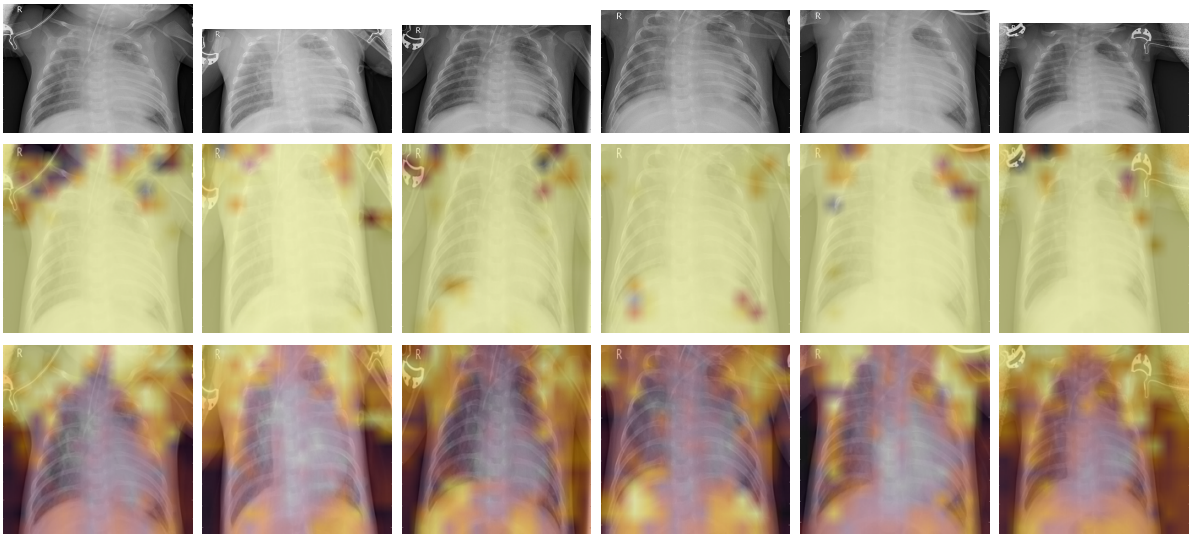


Figure 7: Three rows (top to bottom): original, heatmap (pneumoni), heatmap (normal). Six columns show different samples in ChestX-ray8.

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.

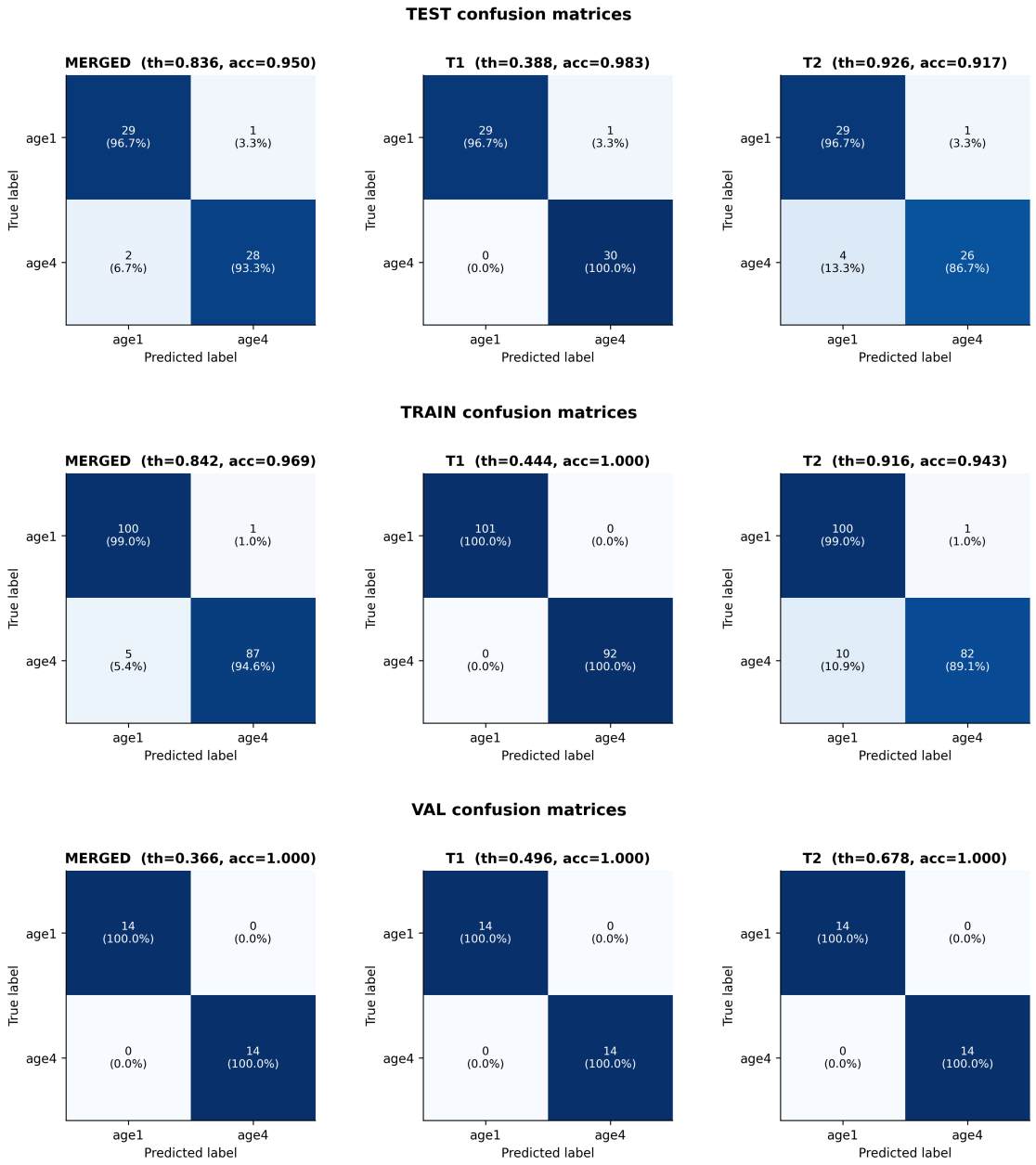


Figure 8: Confusion matrices for TEST (top), TRAIN (middle), and VAL (bottom) sets. Each row shows MERGED, T1-only, and T2-only results.

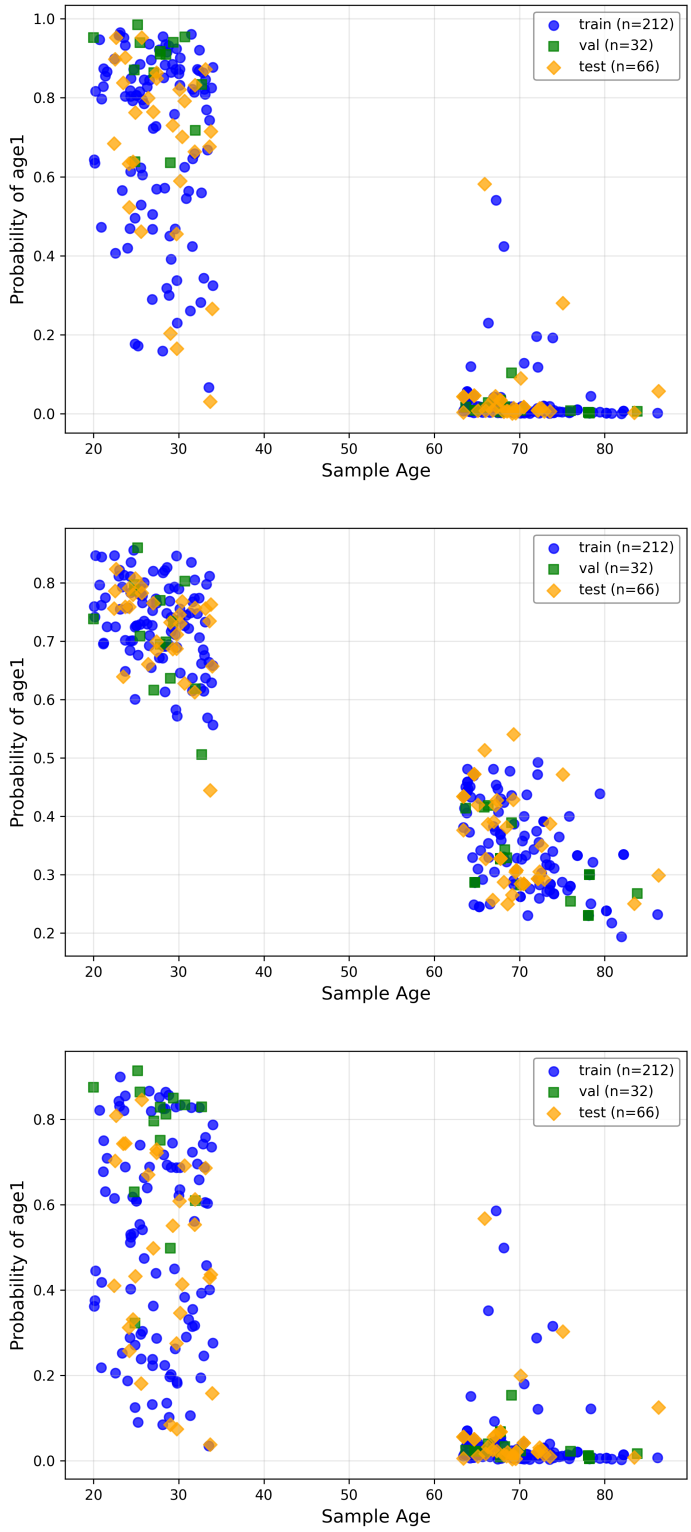


Figure 9: Scatter plots of posterior probability $P(\text{age1})$ versus chronological age for MERGED (top), T1-only (middle), and T2-only (bottom).

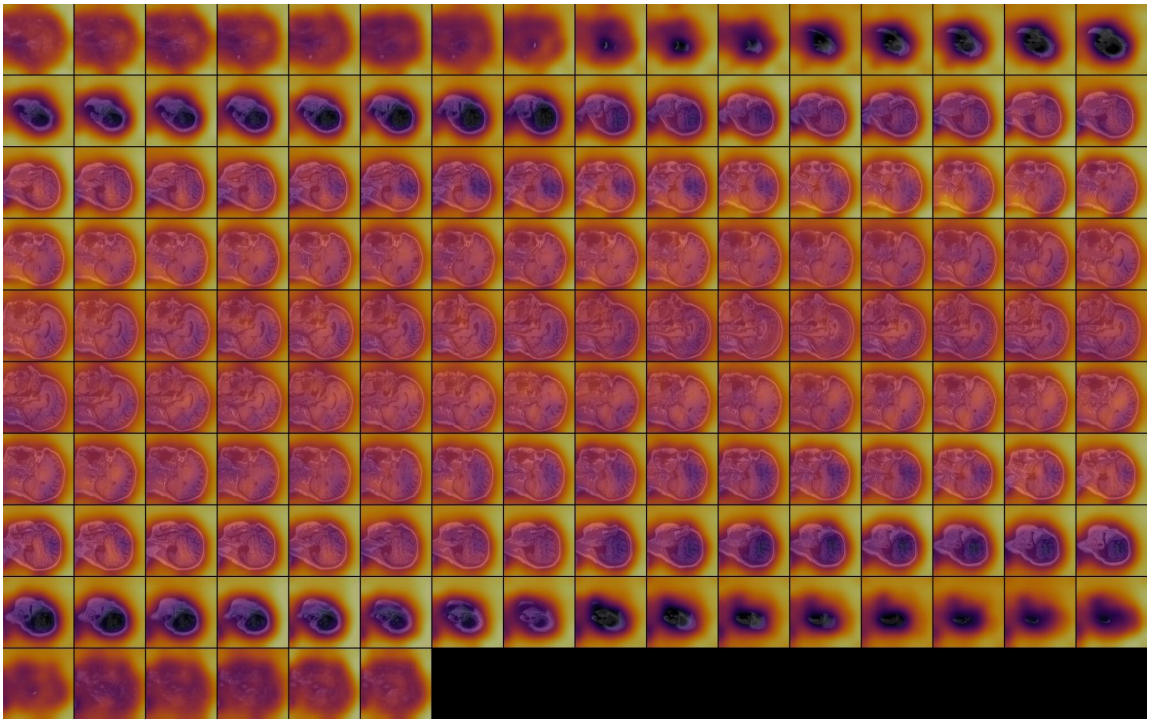


Figure 10: Patch-level contribution heatmap for a sample (true label: age group4 (age range 60-83)). Visualization generated from the T1 modality for the target class **age1**.

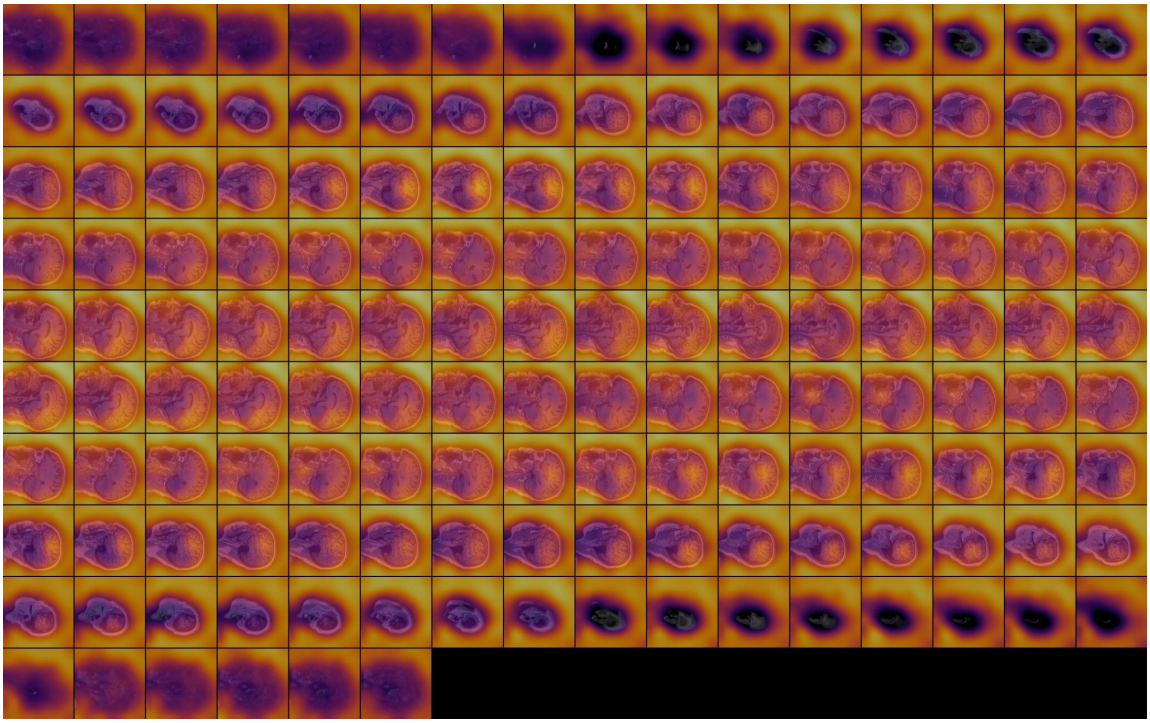


Figure 11: Patch-level contribution heatmap for a sample (true label: age group4 (age range 60-83)). Visualization generated from the T1 modality for the target class **age4**.

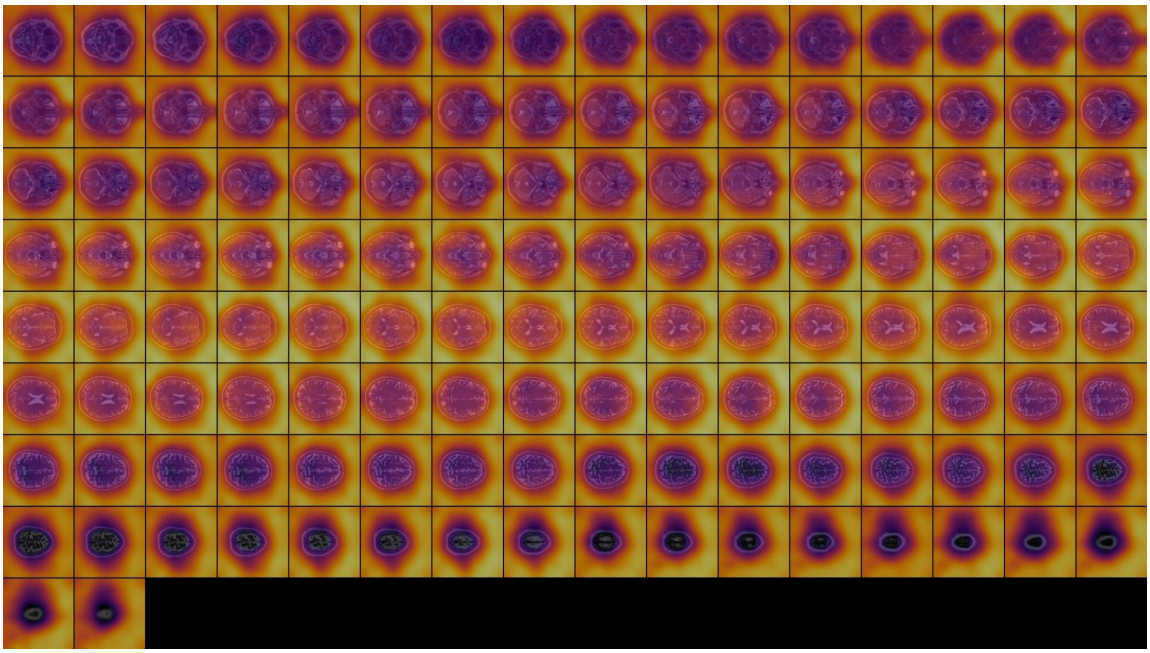


Figure 12: Patch-level contribution heatmap for a sample ((true label: age group4 (age range 60-83)). Visualization generated from the T2 modality for the target class **age1**.

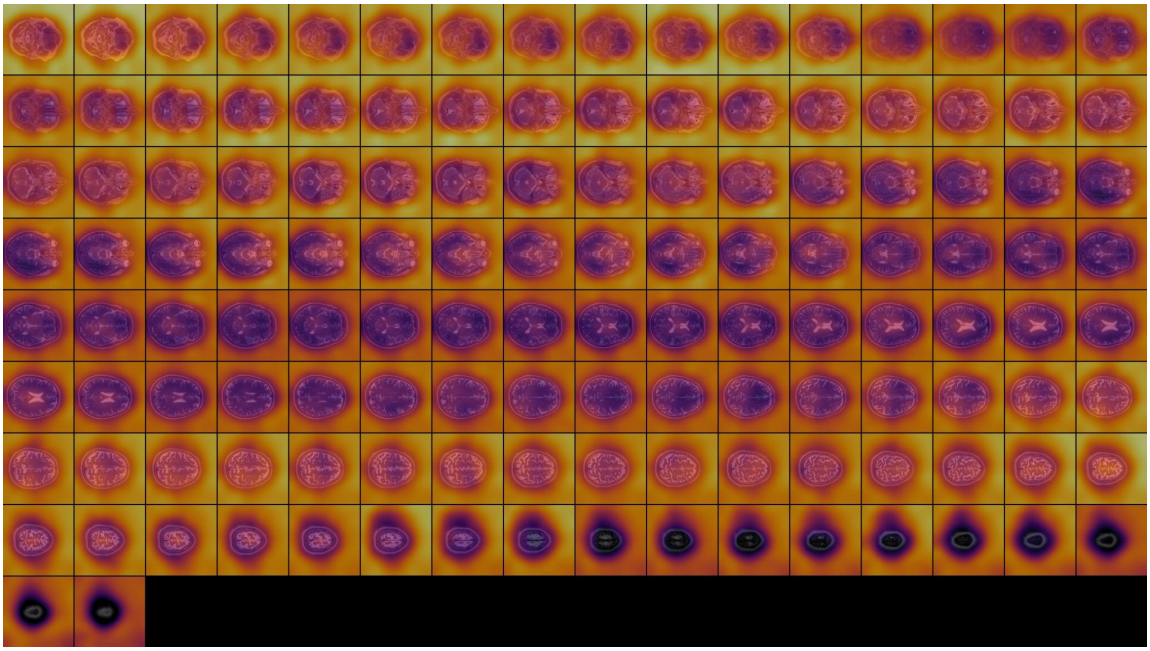


Figure 13: Patch-level contribution heatmap for a sample ((true label: age group4 (age range 60-83)). Visualization generated from the T2 modality for the target class **age4**.

4 Discussion and Conclusion

In this paper, we provide a novel multidimensional, multimodal, and explainable classification model based on the combination of vision transformers and probabilistic circuits. We theoretically design VP-Net so that it never violates smoothness or decomposability, which guarantees that each probabilistic circuit strictly represents a joint probability distribution. We also theoretically show that patch-level, modality-level, and encoder-level class posterior probabilities can be extracted independently.Figure 4 clearly shows that the heatmaps of patch-level posteriors in anomaly images highlight the actual flaws in the image. This experimental result suggests that the patch-level posteriors represent the model's decision process for the classification task. In addition, depending on the optimization, the decision process can be expressed in a form that is more easily understandable to human users.

Our algorithm can be applied to any classification task. We mainly have two recommendations for its practical use. The first one is supervised anomaly detection.In supervised anomaly detection tasks, it is often difficult to collect a sufficiently large number of anomaly samples from the real domain.This typically leads to overfitting due to the mismatch between the training domain and the real domain in which the model is deployed.Furthermore, realizing such overfitting is quite difficult because conventional classification models behave as black boxes. On the other hand, for our model, as shown in Figure 4, by inspecting the decision process, human users can directly evaluate the trustability and robustness of the model's decision.

The second one is the detection of new scientific discoveries. Artificial intelligence can recognize hidden relationships and tendencies between samples, which may contribute to new scientific findings by asking our model to visualize their decision processes. For instance, the visualization of patch-level contributions from MRI volumes as shown in Figure 9 in the classification task of two age groups may help brain scientists discover new relationships between specific brain regions and aging. We provide a platform in our GitHub repository where each user can train their own explainable 2D or 3D image classification model without any knowledge of deep learning. We expect that this will actually contribute to new scientific discoveries.

In this paper, we focused on image classification. But inherently, our algorithm can treat any type of information with an appropriate encoder, including texts, genes, audio signals, electrical signals, and so on. In future research, we will extend our model to other domains, and we will demonstrate that the decision processes can also be extracted for such domains beyond images.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

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

[12] 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.

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

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

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

[16] 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.

[17] 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.

[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] 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.

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

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

[23] 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.