UM–ProtoShare: UNet–Guided, Multi–scale Shared Prototypes for Interpretable Brain Tumour Classification Using Multi–sequence 3D MRI

Ali Golbaf[*1] ALI.GOLBAF@PLYMOUTH.AC.UK
Vivek Singh[1] VIVEK.SINGH@PLYMOUTH.AC.UK
Swen Gaudl[1] SWEN.GAUDL@PLYMOUTH.AC.UK
Emmanuel Ifeachor[1] E.IFEACHOR@PLYMOUTH.AC.UK
[1] *School of Engineering, Computing and Mathematics, University of Plymouth, Plymouth, Devon, UK.*

Editors: Under Review for MIDL 2026

Abstract

Deep learning shows strong promise in brain tumour classification using Magnetic Resonance Imaging (MRI), although limited interpretability constrains clinical translation. Most interpretability methods are post–hoc and yield visual attribution maps that are only weakly connected to the decision process. Clinicians prefer decisions built from evidence they can recognise and verify on MRI, rather than post–hoc explanations. Case–based models embed reasoning by comparing image evidence with learned prototypes, yielding "this looks like that" rationales at decision time and mirroring clinical reasoning. Building on this paradigm, we introduce UM–ProtoShare, which compares the input Multi-sequence 3D brain MRI with a bank of shared, class–agnostic, multi–scale prototypes for pre–operative glioma grading. It returns not only a label, but a set of prototype matches that highlight where the model found support for its prediction. UM–ProtoShare uses a 3D ResNet–152 encoder, a lightweight UNet–style decoder with gated encoder–decoder fusions, and a normalised soft–masked mapping module to align and highlight prototype evidence on MRI. On BraTS–2020, ablations show additive benefits from the normalised mapping module, prototype sharing, multi–scale prototypes, and the decoder with gated fusions. Varying the allocation of prototypes across scales identifies a balanced accuracy–interpretability configuration that closely approaches a strong 3D ResNet–152 in classification performance (Balanced Accuracy: 88.40 ± 2.80; 1.48 percentage points lower) while delivering more faithful and spatially precise evidence than prior case–based models, with Activation Precision (AP) 88.72 ± 1.60 (+11.0% vs MProtoNet; +4.0% vs MAProtoNet) and Incremental Deletion Score (IDS) 5.10 ± 1.30 (lower is better, −32.3% vs MProtoNet, −25.3% vs MAProtoNet).

Keywords: Brain Tumour Classification, multi–sequence 3D MRI, Interpretable Deep Learning, Case–based Models.

1. Introduction

Gliomas are among the most common malignant primary brain tumours (Lapointe et al., 2018). They are classified into grades I–IV based on the World Health Organization (WHO) guidelines, with higher grades reflecting increased malignancy (Louis et al., 2021). Multi–sequence Magnetic Resonance Imaging (MRI), due to its high soft-tissue contrast, is the

* Corresponding author

primary imaging modality for diagnosis, treatment planning, and long-term monitoring (Qiu et al., 2025; Sabeghi et al., 2024). Despite enhancing diagnostic reliability, the high dimensionality and heterogeneity of multi–sequence MRI, together with overlapping radiological features, make visual assessment challenging even for experienced clinicians (Qiu et al., 2025). Deep learning (DL) models, particularly Convolutional Neural Networks (CNNs), have shown promise in brain tumour assessment by extracting and analysing high–dimensional quantitative imaging features from multi–sequence brain MRI (Thomasian et al., 2022; Banerjee et al., 2019). However, most CNNs function as "black–box" models, providing limited insight into their reasoning process (Ibrahim et al., 2021). In clinical practice, where decisions directly affect patient outcomes, this opacity remains a major barrier to the adoption of DL models (Gupta et al., 2025; Rudin, 2019). Interpretability is therefore essential for the clinical translation of DL in brain tumour assessment.

Interpretability in medical imaging lacks a standardised definition (Borys et al., 2023). In line with prior work, we define interpretability as the degree to which the image structures used by a DL model align with recognisable clinical evidence. This alignment with MRI sequences and clinically meaningful concepts enables clinicians to verify, contest, and act on the model's reasoning (Rudin, 2019; Chen et al., 2022). Interpretability methods are commonly categorised into three main categories (Lu et al., 2025). (1) Visual methods generate post–hoc saliency maps but do not embed interpretability into the decision process (Rudin, 2019). They may highlight irrelevant regions resulting in misleading interpretation (Adebayo et al., 2018; Hooker et al., 2019); examples include Class Activation Mapping (CAM) (Yang et al., 2021), Gradient-based Class Activation Mapping (Grad–CAM) (Hussain and Shouno, 2023), and Layer-wise Relevance Propagation (Mandloi et al., 2024). (2) Concept-based methods map predictions to predefined clinical concepts to inform the final decision (Koh et al., 2020). However, they require increased annotation effort and are particularly demanding in medical imaging (Borys et al., 2023). (3) Case–based methods embed interpretability directly into the reasoning process by comparing image patches with learned prototypes, providing "this-looks-like-that" rationales at decision time (Chen et al., 2019). This embedded interpretability makes case–based methods particularly appealing in clinical practice (Rudin, 2019).

ProtoPNet (Chen et al., 2019) introduced case–based reasoning by learning class–specific prototypes in latent space and classifying images on the basis of patch-level similarity to these prototypes. Subsequent case–based work has improved discrimination and localisation performance, primarily in natural-image classification (Rymarczyk et al., 2021; Donnelly et al., 2022; Rymarczyk et al., 2022). However, clinical adoption of case–based methods remains limited. IAIA-BL (Barnett et al., 2021) and XProtoNet (Kim et al., 2021) integrate case–based reasoning into 2D mammography and 2D chest X-rays, respectively. Building on XProtoNet and targeting multi–sequence 3D MRI, MProtoNet (Wei et al., 2024) adds a localisation layer with soft masking and an Online–CAM loss to sharpen attention using only image-level labels. MAProtoNet (Li et al., 2024) further incorporates multi–scale feature fusion, 3D quadruplet attention, and a multi–scale mapping loss.

Despite these advances, important gaps remain for case–based models in multi–sequence 3D MRI. (1) Prototypes are class–specific, restricting reuse of MRI features shared across tumour types and grades (e.g., peritumoural oedema, necrotic cores, enhancing rims) (Louis et al., 2021; Sabeghi et al., 2024). Moreover, because the training losses push prototypes

of different classes apart in latent space, these losses can also separate semantically similar prototypes, leading to unstable predictions (Rymarczyk et al., 2021). (2) Localisation is shaped indirectly through mapping or attention modules, rather than being constrained by a spatial decoder module trained under weak supervision. (3) The effects of short– and long–range spatial relationships are largely missing in existing approaches, which either lack explicit multi–scale prototypes (derived from multi–scale feature maps) or confine multi–scale information to attention mechanisms (Li et al., 2024). This is despite the fact that tumour image features are inherently multi–scale and prone to omission without explicit multi–scale modelling (Kamnitsas et al., 2017).

To address these limitations, we introduce UM–ProtoShare, an interpretable case–based model for brain tumour assessment using multi–sequence 3D MRI. **Our contributions are:**

- **Shared and class–agnostic prototypes.** We design training objectives that learn a bank of shared, class–agnostic prototypes, where each prototype can support multiple classes via soft class–prototype coefficients derived from Grad–CAM–style importance weights. This enables efficient reuse of MRI features shared across tumour types and grades.
- **Weakly supervised localisation with gated fusions.** We improve localisation over prior case–based models by incorporating a lightweight 3D UNet–style decoder with encoder–decoder gated fusions. The proposed feature extractor produces spatially coherent features that align prototype evidence with tumour–related regions when trained using only image-level labels.
- **Explicit multi–scale prototypes.** We learn separate prototype sets for each scale of feature maps, capturing tumour appearance from fine to coarse scales and modelling short– and long–range spatial relationships. Ablations over different per-scale prototype allocations characterise how emphasising different spatial scales trades off classification accuracy against interpretability.

UM–ProtoShare embeds interpretability into the decision process, providing transparent "this looks like that" explanations while maintaining competitive classification performance and localisation coherence. The source code will be available on GitHub.

2. Methodology

2.1. Model overview

Figure 1 presents an overview of the proposed UM–ProtoShare. The UM–ProtoShare architecture comprises four main components: (1) A feature extraction backbone with a 3D encoder and a lightweight UNet–style decoder, where a gating mechanism fuses encoder and decoder features at multiple spatial scales, enhancing the focus on tumour-related regions in the fused features. (2) A localisation component, comprising an add–on module and a mapping module. It transforms the fused features into prototype–specific features, associating each prototype with image regions that are likely to contain tumour tissue. (3) A bank of shared, class–agnostic, multi–scale prototypes that serve as reference tumour features in the latent space. Similarities between these prototypes and the prototype–specific features provide case–based image evidence at different spatial scales. (4) Lastly,

a classification component aggregates prototype similarities into tumour grade predictions. The "Prototype Sharing Loss" and "Online–CAM" blocks in Figure 1 indicate training objectives. The former encourages prototypes to be shared across classes, while the latter encourages tumour-focused attention maps. These objectives shape the prototype bank and the localisation component but are not used during inference.

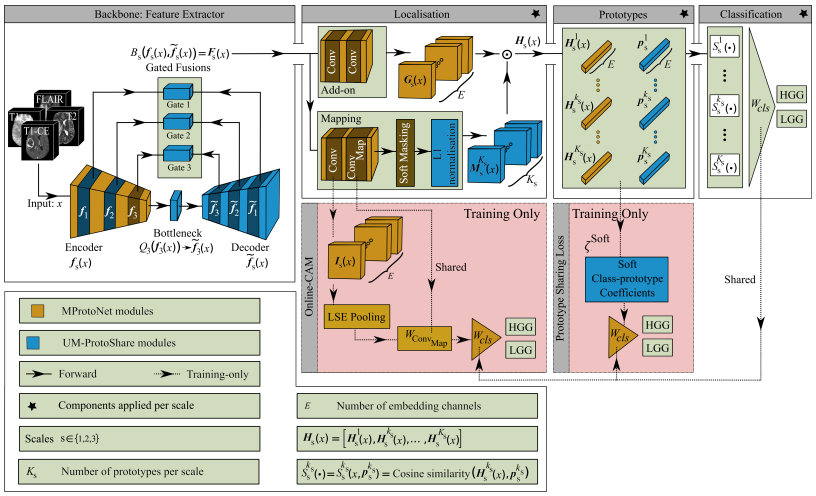

Figure 1: UM–ProtoShare design. The model comprises a backbone for feature extraction, a localisation component that produces prototype–specific high-level features and per–prototype attention maps, a shared, class–agnostic, multi–scale prototype bank, and a classification component. The "Prototype Sharing Loss" and "Online–CAM" are only used during training.

2.2. Model architecture

We summarise here the overall architecture of UM–ProtoShare. Full architectural details, including component configurations and module definitions, are provided in Appendix A.

Backbone. Given an input multi–sequence 3D brain MRI volume x, the 3D encoder produces feature maps $\boldsymbol{f}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ at three spatial scales $s \in \{1, 2, 3\}$, where C_s, H_s, W_s and D_s are the number of channels, height, width, and depth at scale s. At the deepest scale ($s = 3$), a bottleneck block transforms the encoder output into the initial decoder feature $\tilde{\boldsymbol{f}}_3(x) = Q_3(\boldsymbol{f}_3(x))$. Subsequent decoder blocks then produce decoder features $\tilde{\boldsymbol{f}}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ at the remaining scales $s \in \{1, 2\}$. At each scale, a gated fusion block combines encoder and decoder features to obtain fused features $\boldsymbol{F}_s(x) = B_s(\boldsymbol{f}_s(x), \tilde{\boldsymbol{f}}_s(x))$, where channel–wise gates control the relative contributions of encoder and decoder features. The fused features $\boldsymbol{F}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ are passed to the localisation component (Appendix A.1).

Localisation component. The localisation component consists of two modules, each of which receives the fused features $\boldsymbol{F}_s(x)$ as input. An add–on module applies a convolutional

block to project $\boldsymbol{F}_s(x)$ to a fixed embedding dimension E, yielding high-level features $\boldsymbol{G}_s(x) \in \mathbb{R}^{E\times H_s\times W_s\times D_s}$ while preserving the spatial dimensions. In parallel, a mapping module applies a convolutional block to predict raw per–prototype attention maps $\boldsymbol{M}_s^0(x) \in \mathbb{R}^{K_s\times H_s\times W_s\times D_s}$, where K_s denotes the number of prototypes at scale s. These maps are sharpened by a differentiable soft mask and then ℓ_1-normalised over the spatial dimensions, yielding normalised attention maps $\boldsymbol{M}_s(x) \in \mathbb{R}^{K_s\times H_s\times W_s\times D_s}$. For each prototype $\boldsymbol{p}_s^{k_s} \in \mathbb{R}^{E\times 1\times 1\times 1}$, the corresponding attention map $\boldsymbol{M}_s^{k_s}(x)$ is broadcast across the E channels of $\boldsymbol{G}_s(x)$ and combined via an element–wise product to produce a weighted feature map. Global average pooling over (H_s, W_s, D_s) then yields a prototype–specific feature descriptor $\boldsymbol{H}_s^{k_s}(x) \in \mathbb{R}^{E\times 1\times 1\times 1}$ (Appendix A.2).

Prototypes. At each scale $s \in \{1, 2, 3\}$, UM–ProtoShare uses a set of shared, class–agnostic prototypes $\boldsymbol{P}_s = \{\boldsymbol{p}_s^{k_s} \mid k_s \in \{1, \dots, K_s\}\}$. Each prototype $\boldsymbol{p}_s^{k_s}$ is a learnable vector in the same embedding space as $\boldsymbol{H}_s^{k_s}(x)$. Prototypes are initialised randomly and updated jointly with the other UM–ProtoShare components during training under the objectives described in Section 2.3 and Appendix B. For each prototype $\boldsymbol{p}_s^{k_s} \in \boldsymbol{P}_s$, UM–ProtoShare computes a cosine similarity $S_s^{k_s}(x, \boldsymbol{p}_s^{k_s}) = \cos(\boldsymbol{H}_s^{k_s}(x), \boldsymbol{p}_s^{k_s})$. These similarities are concatenated across all scales and prototypes into a single vector $S(x, \boldsymbol{P}) \in \mathbb{R}^{K_{\text{total}}}$, where $K_{\text{total}} = \sum_{s=1}^{3} K_s$ (in our implementation, $K_{\text{total}} = 30$; see Appendix A.3).

Classification. The classification component is a linear layer that maps the similarity vector $S(x, \boldsymbol{P})$ to class logits. Let C denote the number of classes and $\boldsymbol{w}_{\text{cls}} \in \mathbb{R}^{C\times K_{\text{total}}}$ the classifier weights. For class c, the logit $y_c(x)$ is obtained by a weighted sum of prototype similarities $y_c(x) = \sum_{s=1}^{3}\sum_{k_s=1}^{K_s} w_{\text{cls}}[c, \boldsymbol{p}_s^{k_s}]\, S_s^{k_s}(x, \boldsymbol{p}_s^{k_s})$, where $w_{\text{cls}}[c, \boldsymbol{p}_s^{k_s}]$ denotes the classifier weight associated with prototype $\boldsymbol{p}_s^{k_s}$ for class c. A softMax over $\{y_c(x)\}_{c=1}^{C}$ yields the predicted class probabilities (Appendix A.4).

2.3. Training objectives and loss functions

UM–ProtoShare follows the training objectives used in case–based models (Chen et al., 2019), adapted to shared, multi–scale prototypes. The goal is to learn a latent space in which prototype–specific descriptors $\boldsymbol{H}_s^{k_s}(x)$ capture tumour-related imaging features and their associated regions, to anchor each prototype $\boldsymbol{p}_s^{k_s}$ to a specific 3D region of a real training image, and to train a classifier that maps the full similarity vector $S(x, \boldsymbol{P})$ to class logits.

To realise these objectives, UM–ProtoShare adopts the three-stage training procedure used in prior case–based models (Chen et al., 2019): (1) optimisation of all components preceding the classification layer (latent space learning); (2) prototype reassignment ("push"); and (3) training of the classification component. The loss functions used in these stages include a classification loss L_{cls}, a clustering loss $L_{\text{clst}}^{\text{class-specific}}$, a separation loss $L_{\text{sep}}^{\text{class-specific}}$, a mapping loss L_{map}, an Online–CAM loss L_{OC}, and an ℓ_1-regularisation loss L_{L_1} on the classifier weights (Wei et al., 2024). A detailed explanation of each training stage, together with the corresponding loss functions, is provided in Appendix B.

In UM–ProtoShare, the clustering and separation losses are reformulated to be compatible with shared, class–agnostic prototypes ($L_{\text{clst}}^{\text{class-specific}} \to L_{\text{clst}}^{\text{class-agnostic}}$ and $L_{\text{sep}}^{\text{class-specific}} \to L_{\text{sep}}^{\text{class-agnostic}}$). The mapping and Online–CAM losses are adapted to the multi–scale set-

ting, yielding scale-specific terms L^s_{map} and L^s_{OC}. We also introduce a prototype–diversity regulariser L_{div} to reduce redundancy in the prototype bank by discouraging highly similar prototypes at each scale.

Clustering and separation losses. As described in Section 2.2, UM–ProtoShare maps each input x to prototype–specific descriptors $\boldsymbol{H}^{k_s}_s(x)$ and computes their cosine similarities to the corresponding prototypes, $S^{k_s}_s(x, \boldsymbol{p}^{k_s}_s) = \cos\big(\boldsymbol{H}^{k_s}_s(x), \boldsymbol{p}^{k_s}_s\big)$. The clustering and separation losses encourage high similarity between descriptors and prototypes that support the ground-truth class, and low similarity to prototypes that support other classes. The formulation of these losses determines whether prototypes are class–specific or class–agnostic. This choice affects only the training objectives. At inference, the classifier always uses the full similarity vector $S(x, \boldsymbol{P})$ to deliver the final prediction.

Class–specific formulation. In ProtoPNet-style models, each prototype is assigned to exactly one class through an ownership matrix $\boldsymbol{A} \in \{0,1\}^{C \times K_{\text{total}}}$, where $j \in \{1, \ldots, K_{\text{total}}\}$ indexes prototypes. The entry $A_{c,j} = 1$ if prototype j belongs to class c and $A_{c,j} = 0$ otherwise, with $\sum_{c=1}^{C} A_{c,j} = 1$ for every prototype index j (each j corresponds to one prototype $\boldsymbol{p}^{k_s}_s$). The resulting hard class–prototype coefficient is $\zeta^{\text{hard}}_{c,\boldsymbol{p}^{k_s}_s} = A_{c,j} \in \{0,1\}$, meaning that prototype $\boldsymbol{p}^{k_s}_s$ supports class c if $\zeta^{\text{hard}}_{c,\boldsymbol{p}^{k_s}_s} = 1$ and does not support it otherwise. Using the similarity scores $S^{k_s}_s(x, \boldsymbol{p}^{k_s}_s)$, the class–specific clustering and separation losses for an input $(x, y = c)$ across scales are (Chen et al., 2019),

$$L^{\text{class-specific}}_{\text{clst}} = -\max_{s,k_s}\Big(\zeta^{\text{hard}}_{c,\boldsymbol{p}^{k_s}_s}\, S^{k_s}_s(x, \boldsymbol{p}^{k_s}_s)\Big). \tag{1}$$

$$L^{\text{class-specific}}_{\text{sep}} = \max_{s,k_s}\Big(\big(1 - \zeta^{\text{hard}}_{c,\boldsymbol{p}^{k_s}_s}\big)\, S^{k_s}_s(x, \boldsymbol{p}^{k_s}_s)\Big). \tag{2}$$

Class–agnostic formulation. UM–ProtoShare uses shared, class–agnostic prototypes, in which each prototype can support multiple classes. This reflects the fact that several MRI features (e.g., peritumoural oedema, necrotic cores, enhancing rims) may legitimately appear across tumour types and grades (Louis et al., 2021; Sabeghi et al., 2024).

Shared, class–agnostic prototypes that retain class-discriminative power can be understood by analogy to class activation mapping methods such as Grad–CAM. These methods operate on a shared set of feature maps in the latent space and, for a chosen class c, compute class–discriminative importance weights $\alpha_{c,k}(x)$ for each feature map to form class–discriminative localisation maps. In UM–ProtoShare, we similarly introduce class–discriminative importance weights $\gamma_{c,\boldsymbol{p}^{k_s}_s}$ for each shared prototype $\boldsymbol{p}^{k_s}_s$, in direct analogy to the Grad–CAM weights for feature maps. Our formulation shows that these importance weights coincide with the classifier weights and thus quantify how much each prototype contributes to the class logit $y_c(x)$. From these importance weights, we derive soft class–prototype coefficients $\zeta^{\text{soft}}_{c,\boldsymbol{p}^{k_s}_s}$, which encode how strongly each prototype supports each class and serve as soft ownership factors in the class–agnostic clustering and separation losses.

In Grad–CAM, given feature maps $\boldsymbol{Z}^k(x) \in \mathbb{R}^{H \times W \times D}$, $k \in \{1, \ldots, K\}$, and a class score $y_c(x)$, the importance weight for each feature map is defined as (Selvaraju et al., 2017),

$$\alpha_{c,k}(x) = \frac{1}{HWD} \sum_{m=1}^{H} \sum_{n=1}^{W} \sum_{l=1}^{D} \frac{\partial y_c(x)}{\partial Z^k_{m,n,l}(x)}. \tag{3}$$

Global average pooling of the gradients over the spatial dimensions yields one scalar importance weight per feature map.

In UM–ProtoShare, the classification component operates on prototype similarities rather than feature maps,

$$y_c(x) = \sum_{s=1}^{3} \sum_{k_s=1}^{K_s} w_{\text{cls}}[c, \boldsymbol{p}_s^{k_s}]\, S_s^{k_s}(x, \boldsymbol{p}_s^{k_s}). \tag{4}$$

The term $w_{\text{cls}}[c, \boldsymbol{p}_s^{k_s}]$ denotes the classifier weight linking prototype $\boldsymbol{p}_s^{k_s}$ to class c. In direct analogy to the Grad–CAM importance weights in Equation (3), we define the class–discriminative importance weight of prototype $\boldsymbol{p}_s^{k_s}$ for class c as the derivative of the class score with respect to its similarity,

$$\gamma_{c,\boldsymbol{p}_s^{k_s}} = \frac{\partial y_c(x)}{\partial S_s^{k_s}(x, \boldsymbol{p}_s^{k_s})} = w_{\text{cls}}[c, \boldsymbol{p}_s^{k_s}]. \tag{5}$$

Here, prototype similarities $S_s^{k_s}(x, \boldsymbol{p}_s^{k_s})$ are already spatially pooled scalar quantities derived from feature descriptors (see Equation (13) and Equation (14) in Appendix A), so the Grad–CAM-style definition of importance weights in Equation (3) reduces to taking the derivative of the class score with respect to $S_s^{k_s}(x, \boldsymbol{p}_s^{k_s})$, as in Equation (5). Moreover, because the classifier is linear in the prototype similarities (see Equation (4)), these importance weights coincide with the classifier weights.

The importance weights $\gamma_{c,\boldsymbol{p}_s^{k_s}}$ thus play the same role as the Grad–CAM weights $\alpha_{c,k}(x)$, but at the level of shared prototypes rather than feature maps, indicating how each prototype influences the logit for class c. We then construct non-negative, normalised soft class–prototype coefficients from these importance weights,

$$\zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}} = \frac{\text{ReLU}\big(w_{\text{cls}}[c, \boldsymbol{p}_s^{k_s}]\big)}{\sum_{c'} \text{ReLU}\big(w_{\text{cls}}[c', \boldsymbol{p}_s^{k_s}]\big) + \varepsilon}, \quad \varepsilon = 10^{-6},\ \text{ReLU}(x) = \max(x, 0). \tag{6}$$

By construction, $\zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}} \in [0, 1]$ and $\sum_c \zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}} = 1$, so for each prototype $\boldsymbol{p}_s^{k_s}$ the vector $\{\zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}}\}_c$ defines a soft ownership distribution over classes. The ReLU ensures that only positive classifier weights are treated as support when assigning a prototype to classes, mirroring the focus on positive evidence for the class of interest in Grad–CAM.

Using these coefficients, the class–agnostic clustering and separation losses for an input $(x, y = c)$ across scales are defined as,

$$L_{\text{clst}}^{\text{class-agnostic}} = -\sum_{s=1}^{3} \sum_{k_s=1}^{K_s} \zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}} S_s^{k_s}(x, \boldsymbol{p}_s^{k_s}). \tag{7}$$

$$L_{\text{sep}}^{\text{class-agnostic}} = \sum_{s=1}^{3} \sum_{k_s=1}^{K_s} \Big(\sum_{c' \neq c} \zeta^{\text{soft}}_{c',\boldsymbol{p}_s^{k_s}} \Big) S_s^{k_s}(x, \boldsymbol{p}_s^{k_s}). \tag{8}$$

Thus, every prototype $\boldsymbol{p}_s^{k_s}$ with non-zero $\zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}}$ receives a training gradient, scaled by how strongly it supports class c. This yields smoother and more stable optimisation than hard class–prototype assignments, particularly in the shared multi–scale setting. In the binary case, the separation weight simplifies to $\sum_{c' \neq c} \zeta^{\text{soft}}_{c',\boldsymbol{p}_s^{k_s}} = 1 - \zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}}$.

Prototype–diversity regulariser loss Without additional constraints, multiple prototypes in the shared bank may collapse onto similar regions of the latent space, yielding redundant explanations and thereby limiting its representational capacity. To mitigate this, we introduce a prototype–diversity regulariser L_{div} that penalises the average pairwise cosine similarity between normalised prototypes at each scale,

$$L_{\text{div}} = \frac{1}{3}\sum_{s=1}^{3} \text{Mean}_{i\le j\le K_s}\left(\frac{\boldsymbol{p}_s^i \cdot \boldsymbol{p}_s^j}{\|\boldsymbol{p}_s^i\|\,\|\boldsymbol{p}_s^j\|}\right). \tag{9}$$

Here $\boldsymbol{p}_s^i$ and $\boldsymbol{p}_s^j$ denote the i-th and j-th prototypes at scale s, and the mean is taken over all prototype pairs. Minimising L_{div} encourages prototypes at each scale to span diverse directions in the latent space, reducing redundancy in the prototype bank and promoting coverage of distinct tumour features, in line with diversity regularisation used in recent case–based models (Rymarczyk et al., 2022).

3. Experiments and results

3.1. Dataset

We evaluated the UM–ProtoShare on BraTS-2020 training cohort (Menze et al., 2014), a multi–sequence 3D brain MRI dataset comprising 369 pre–operative glioma cases, each with four sequences, including T1-weighted (T1), contrast-enhanced T1-weighted (T1-CE), T2-weighted (T2), and T2 Fluid-Attenuated Inversion Recovery (FLAIR) MRI. WHO tumour grades were provided (low-grade gliomas (LGG): 76, high-grade gliomas (HGG): 293). Ground-truth segmentations were also available for three subregions, including enhancing tumour, peritumoural oedema, and necrotic core, which we merged into a single tumour mask. Only MRI scans and tumour grades were used for training, while tumour masks were used exclusively to compute interpretability metrics. Preprocessing and data augmentation procedures are described in Appendix C.

3.2. Evaluation metrics and training hyper-parameters

We evaluate the classification performance of UM–ProtoShare using the balanced accuracy (BAC) metric, which compensates for class imbalance (Wei et al., 2024). Interpretability is assessed in terms of localisation coherence and correctness, measured respectively by Activation Precision (AP; higher is better) and Incremental Deletion Score (IDS; lower is better) (Nauta et al., 2023). All results are reported using 5-fold cross-validation on different training and test splits. Full definitions of the evaluation metrics and the complete set of training hyper-parameters are provided in Appendix D.

3.3. Ablation studies

To systematically examine the design choices in UM–ProtoShare, we perform a series of ablations A1–A2, B, C, D and E1–E3 under a fixed total prototype budget $K_{\text{total}} = \sum_{s=1}^{3} K_s = 30$ and identical training settings (same data splits, optimiser and loss weights). The ablations progressively introduce the main components of UM–ProtoShare: A1 is a single-scale,

class–specific prototype model without the decoder and gated fusion; A2 replaces class–specific prototypes by shared, class–agnostic prototypes; B introduces explicit multi–scale prototypes; C adds the decoder module; D enables gated encoder-–decoder fusion; and E1–E3 vary the per-scale prototype allocation (fine-heavy, mid-heavy and coarse-heavy) while keeping K_{total} fixed. For each configuration, we evaluate the effects on classification performance (BAC) and interpretability (AP, IDS). Full architectural details for each ablation configuration are provided in Appendix E.

Starting from A1 and progressively adding the components introduced in the ablation study (A2–E), UM–ProtoShare reaches BAC values between 85.92 ± 2.14 and 89.21 ± 3.18, with the coarse-heavy configuration (E3) achieving the highest BAC among the ablations (89.21 ± 3.18). Across these configurations, interpretability improves markedly. AP increases from 81.38 ± 2.23 (A1) to 90.10 ± 1.80 (E1), and IDS decreases from 7.13 ± 2.93 to 4.60 ± 1.20. Using the mid-heavy allocation (E2) as our default UM–ProtoShare model, we obtain 88.40 ± 2.80 BAC, 88.72 ± 1.60 AP and 5.10 ± 1.30 IDS. Detailed trends for each ablation, including intermediate variants (A1, A2, B, C, D and E1–E3), are reported in Appendix G.

3.4. Comparative benchmark

We compare UM–ProtoShare (using the mid-heavy configuration E2) against a CNN model derived from the UM–ProtoShare backbone, as well as ProtoPNet, XProtoNet, MProtoNet and MAProtoNet. All models are trained under the same data splits and comparable optimisation settings. Implementation details for each baseline are provided in Appendix F.

Compared with MAProtoNet, UM–ProtoShare increases BAC from 87.36 ± 3.26 to 88.40 ± 2.80, while also increasing AP from 85.29 ± 4.14 to 88.72 ± 1.60 and reducing IDS from 6.83 ± 3.12 to 5.10 ± 1.30. Relative to MProtoNet and older case–based approaches (ProtoPNet, XProtoNet), UM–ProtoShare offers substantial gains in interpretability (higher AP, lower IDS) while retaining competitive BAC. Compared with the CNN derived from the same backbone (89.73 ± 2.36 BAC, 10.65 ± 5.63 AP, 14.85 ± 5.21 IDS), UM–ProtoShare trades about 1.3 percentage points in BAC for much stronger interpretability, achieving 88.40 ± 2.80 BAC, 88.72 ± 1.60 AP and 5.10 ± 1.30 IDS. Full per-model metrics and additional quantitative analyses are provided in Appendix G. Quantitative results for all models are summarised in Table 1, and representative prototype attention maps are shown in Figure 2.

4. Conclusion

UM–ProtoShare combines a bank of shared, class–agnostic, multi–scale prototypes with a normalised mapping module and a decoder with gated fusions to achieve a favourable accuracy–interpretability balance for glioma grading (LGG vs HGG) on multi–sequence 3D MRI. On BraTS-2020, it attains competitive balanced accuracy compared with a strong CNN baseline while substantially improving quantitative interpretability metrics. Each prediction is supported by a small set of spatially localised prototype matches, yielding evidence maps in image space that are directly inspectable and naturally expressed in a "this looks like that" form. Qualitative analysis indicates that these prototypes capture clinically meaningful tumour appearances across MRI sequences Taken together, these properties

Table 1: Classification and interpretability performance of UM–ProtoShare and baseline models on the BraTS-2020 dataset. Results are reported as mean ± standard deviation over five cross-validation folds; higher BAC and AP and lower IDS indicate better performance.

Model	Ablation	Design configuration	BAC (%, ↑)	IDS (%, ↓)	AP (%, ↑)
CNN	–	–	89.73 ± 2.36	14.85 ± 5.21	10.65 ± 5.63
ProtoPNet	–	–	86.42 ± 1.78	21.56 ± 5.83	11.73 ± 3.02
XProtoNet	–	–	87.23 ± 1.59	17.46 ± 4.98	18.12 ± 2.76
MProtoNet	–	–	84.69 ± 3.03	7.53 ± 3.24	79.93 ± 1.32
MAProtoNet	–	–	87.36 ± 3.26	6.83 ± 3.12	85.29 ± 4.14
UM–ProtoShare	A1	single-scale; class–specific	85.92 ± 2.14	7.13 ± 2.93	81.38 ± 2.23
	A2	single-scale; class–agnostic	87.43 ± 1.27	7.49 ± 4.07	83.28 ± 3.41
	B	multi–scale; class–agnostic	88.85 ± 2.56	6.52 ± 3.29	85.89 ± 1.24
	C	multi–scale; class–agnostic; decoder	86.47 ± 2.85	5.23 ± 2.18	87.51 ± 1.38
	D	multi–scale; class–agnostic; decoder; gated fusions	88.14 ± 2.74	5.36 ± 1.28	86.45 ± 1.38
	E1	fine-heavy	87.90 ± 2.90	4.60 ± 1.20	90.10 ± 1.80
	E2	mid-heavy	88.40 ± 2.80	5.10 ± 1.30	88.72 ± 1.60
	E3	coarse-heavy	89.21 ± 3.18	5.98 ± 2.57	85.80 ± 2.08

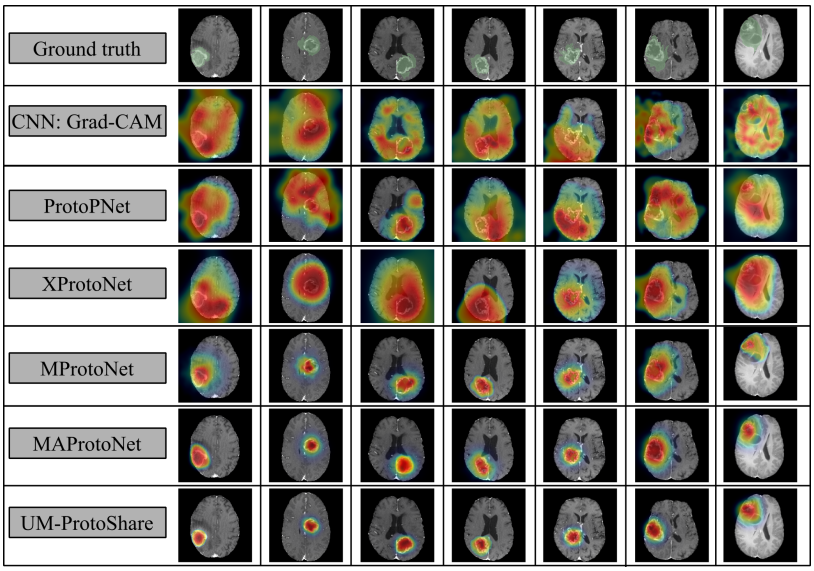

Figure 2: Attribution maps for representative BraTS-2020 cases (T1-CE slices). UM–ProtoShare provides more precise tumour localisation than prior case–based models.

make UM–ProtoShare a practical step toward clinically legible, case–based decision support for brain tumour assessment.

Acknowledgments

This work was supported by a UK EPSRC studentship and, in part, by Brain Tumour Research.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, October 2018. doi: 10.48550/arXiv.1810.03292.

Param Ahir and Mehul Parikh. Dynamic channel-aligned 3d u-net for explainable t2-weighted infant brain mri synthesis from t1-weighted mri. *ICTACT Journal on Soft Computing*, 15(3), January 2025. doi: 10.21917/ijsc.2025.0504.

Ebtihal J Alwadee, Xianfang Sun, Yipeng Qin, and Frank C Langbein. Latup-net: A lightweight 3d attention u-net with parallel convolutions for brain tumor segmentation. *Computers in Biology and Medicine*, 184:109353, January 2025. doi: 10.1016/j.compbiomed.2024.109353.

Subhashis Banerjee, Sushmita Mitra, Francesco Masulli, and Stefano Rovetta. Deep radiomics for brain tumor detection and classification from multi-sequence mri. Eprint arXiv:1903.09240, March 2019.

Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *Nature Machine Intelligence*, 3 (12):1061–1070, December 2021. doi: 10.48550/arXiv.2103.12308.

Katarzyna Borys, Yasmin Alyssa Schmitt, Meike Nauta, Christin Seifert, Nicole Krämer, Christoph M Friedrich, and Felix Nensa. Explainable ai in medical imaging: An overview for clinical practitioners–beyond saliency-based xai approaches. *European journal of radiology*, 162:110786, May 2023. doi: 10.1016/j.ejrad.2023.110786.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, December 2019. doi: 10.48550/arXiv.1806.10574.

Haomin Chen, Catalina Gomez, Chien-Ming Huang, and Mathias Unberath. Explainable medical imaging ai needs human-centered design: guidelines and evidence from a systematic review. *NPJ digital medicine*, 5(1):156, October 2022. doi: 10.1038/s41746-022-00699-2.

Shaoguo Cui, Lei Mao, Jingfeng Jiang, Chang Liu, and Shuyu Xiong. Automatic semantic segmentation of brain gliomas from mri images using a deep cascaded neural network. *Journal of healthcare engineering*, 2018(1):4940593, March 2018. doi: 10.1155/2018/4940593.

Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10265–10275, May 2022. doi: 10.48550/arXiv.2111.15000.

Amir Ebrahimi, Suhuai Luo, and Raymond Chiong. Introducing transfer learning to 3d resnet-18 for alzheimer's disease detection on mri images. In *2020 35th international conference on image and vision computing New Zealand (IVCNZ)*, pages 1–6. IEEE, November 2020. doi: 10.1109/IVCNZ51579.2020.9290616.

Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:109172, April 2023. doi: 10.1016/j.patcog.2022.109172.

Babita Gupta, Rishabha Malviya, Sonali Sundram, and Sathvik Belagodu Sridhar. Explainable artificial intelligence technique in deep learning–based medical image analysis. *Explainable and Responsible Artificial Intelligence in Healthcare*, pages 165–190, March 2025. doi: 10.1002/9781394302444.ch7.

Meenu Gupta, Ankur Ankur, Rakesh Kumar, and Pankaj Zanke. A comprehensive analysis on resnet-based techniques for brain tumor detection. In *2024 first international conference on technological innovations and advance computing (TIACOMP)*, pages 455–461. IEEE, June 2024. doi: 10.1109/TIACOMP64125.2024.00082.

Nisreen Haydar, Khatoun Alyousef, Usama Alanan, Rana Issa, Fawaz Baddour, Zuheir Al-Shehabi, and Moatasem Hussein Al-Janabi. Role of magnetic resonance imaging (mri) in grading gliomas comparable with pathology: A cross-sectional study from syria. *Annals of Medicine and Surgery*, 82:104679, October 2022. doi: 10.1016/j.amsu.2022.104679.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, November 2016. doi: 10.1109/IEEESTD.1997.85951.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, June 2019. doi: 10.48550/arXiv.1806.10758.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, May 2018. doi: 10.48550/arXiv.1709.01507.

Tahir Hussain and Hayaru Shouno. Explainable deep learning approach for multi-class brain magnetic resonance imaging tumor classification and localization using gradient-weighted class activation mapping. *Information*, 14(12):642, November 2023. doi: 10.3390/info14120642.

A Ibrahim, S Primakov, M Beuque, HC Woodruff, I Halilaj, G Wu, T Refaee, R Granzier, Y Widaatalla, Roland Hustinx, et al. Radiomics for precision medicine: Current challenges, future prospects, and the proposal of a new framework. *Methods*, 188:20–29, April 2021. doi: 10.1016/j.ymeth.2020.05.022.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, December 2021. doi: https://doi.org/10.1038/s41592-020-01008-z.

Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, February 2017. doi: 10.1016/j.media.2016.10.004.

Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprotonet: diagnosis in chest radiography with global and local explanations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15719–15728, June 2021. doi: 10.1109/CVPR46437.2021.01546.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, November 2020. doi: 10.48550/arXiv.2007.04612.

Chao Kong, Ding Yan, Kai Liu, Yong Yin, and Changsheng Ma. Multiple deep learning models based on mri images in discriminating glioblastoma from solitary brain metastases: a multicentre study. *BMC Medical Imaging*, 25(1):171, May 2025. doi: 10.1186/s12880-025-01703-3.

Sarah Lapointe, Arie Perry, and Nicholas A Butowski. Primary brain tumours in adults. *The Lancet*, 392(10145):432–446, August 2018. doi: 10.1016/S0140-6736(18)30990-5.

Binghua Li, Jie Mao, Zhe Sun, Chao Li, Qibin Zhao, and Toshihisa Tanaka. Maprotonet: A multi-scale attentive interpretable prototypical part network for 3d magnetic resonance imaging brain tumor classification. Eprint arXiv:2404.08917, April 2024.

Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9215–9223, January 2018. doi: 10.48550/arXiv.1802.10171.

David N Louis, Arie Perry, Pieter Wesseling, Daniel J Brat, Ian A Cree, Dominique Figarella-Branger, Cynthia Hawkins, HK Ng, Stefan M Pfister, Guido Reifenberger, et al. The 2021 who classification of tumors of the central nervous system: a summary. *Neuro-oncology*, 23(8):1231–1251, August 2021. doi: 10.1093/neuonc/noab106.

Jiahao Lu, Chong Yin, Silvia Ingala, Kenny Erleben, Michael Bachmann Nielsen, and Sune Darkner. Mera: Multimodal and multiscale self-explanatory model with considerably reduced annotation for lung nodule diagnosis. Eprint arXiv:2504.19357, April 2025.

Saurabh Mandloi, Mohd Zuber, and Rajeev Kumar Gupta. An explainable brain tumor detection and classification model using deep learning and layer-wise relevance propagation. *Multimedia Tools and Applications*, 83(11):33753–33783, March 2024. doi: 10.1007/s11042-023-16708-9.

Matia Martucci, Rosellina Russo, Francesco Schimperna, Gabriella D'Apolito, Marco Panfili, Alessandro Grimaldi, Alessandro Perna, Andrea Maurizio Ferranti, Giuseppe Varcasia, Carolina Giordano, et al. Magnetic resonance imaging of primary adult brain tumors: state of the art and future perspectives. *Biomedicines*, 11(2):364, January 2023. doi: 10.3390/biomedicines11020364.

Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, October 2014. doi: 10.1109/TMI.2014.2377694.

Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14933–14943, April 2021. doi: 10.48550/arXiv.2012.02046.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55(13s):1–42, July 2023. doi: 10.1145/3583558.

Nassir Navab, Joachim Hornegger, William M Wells, and Alejandro Frangi. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, volume 9351. Springer, Munich, Germany, September 2015. ISBN 9783319245713. doi: 10.1007/978-3-319-24553-9.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. Eprint arXiv:1804.03999, April 2018.

Zelin Qiu, Xi Wang, Zhuoyao Xie, Juan Zhou, Yu Wang, Lingjie Yang, Xinrui Jiang, Juyoung Bae, Moo Hyun Son, Qiang Ye, et al. Large-scale multi-sequence pretraining for generalizable mri analysis in versatile clinical applications. Eprint arXiv:2508.07165, August 2025.

Torsten Rohlfing, Natalie M Zahr, Edith V Sullivan, and Adolf Pfefferbaum. The sri24 multichannel atlas of normal adult human brain structure. *Human brain mapping*, 31(5):798–819, May 2010. doi: 10.1002/hbm.20906.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, May 2019. doi: 10.1038/s42256-019-0048-x.

Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, August 2021. doi: 10.1145/3447548.3467245.

Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *European Conference on Computer Vision*, pages 351–368. Springer, October 2022. doi: 10.1007/978-3-031-19775-8_21.

Paniz Sabeghi, Paniz Zarand, Sina Zargham, Batis Golestany, Arya Shariat, Myles Chang, Evan Yang, Priya Rajagopalan, Daniel Chang Phung, and Ali Gholamrezanezhad. Advances in neuro-oncological imaging: an update on diagnostic approach to brain tumors. *Cancers*, 16(3):576, January 2024. doi: 10.3390/cancers16030576.

Karshiev Sanjar, Olimov Bekhzod, Jaeil Kim, Jaesoo Kim, Anand Paul, and Jeonghong Kim. Improved u-net: Fully convolutional network model for skin-lesion segmentation. *Applied Sciences*, 10(10):3658, May 2020. doi: 10.3390/app10103658.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, October 2017. doi: 10.48550/arXiv.1610.02391.

Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5229–5238, October 2019. doi: 10.48550/arXiv.1907.05740.

Nicole M Thomasian, Ihab R Kamel, and Harrison X Bai. Machine intelligence in non-invasive endocrine cancer diagnostics. *Nature Reviews Endocrinology*, 18(2):81–95, February 2022. doi: 10.1038/s41574-021-00543-9.

Jingqi Wang, Peng Jiajie, Zhiming Liu, and Hengjun Zhao. Hqprotopnet: An evidence-based model for interpretable image recognition. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, June 2023. doi: 10.1109/IJCNN54540.2023.10191863.

Michelle L Wegscheid and Jack W Jennings. Vasari glioma features: insights into their impact and performance. *Radiology: Imaging Cancer*, 6(5):e249018, September 2024. doi: 10.1148/rycan.249018.

Yuanyuan Wei, Roger Tam, and Xiaoying Tang. Mprotonet: A case-based interpretable model for brain tumor classification with 3d multi-parametric magnetic resonance imaging. In *Medical Imaging with Deep Learning*, pages 1798–1812. PMLR, April 2024. doi: 10.48550/arXiv.2304.06258.

Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Rui-Wei Zhao, Tao Zhang, Xuequan Lu, and Shang Gao. Cream: Weakly supervised object localization via class re-activation

mapping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9437–9446, May 2022. doi: 10.48550/arXiv.2205.13922.

Rui Yang, Fengkai Ke, Huanping Liu, Mingcheng Zhou, and Hui-Min Cao. Exploring smri biomarkers for diagnosis of autism spectrum disorders based on multi class activation mapping models. *IEEE Access*, 9:124122–124131, March 2021. doi: 10.1109/ACCESS.2021.3069211.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, October 2018. doi: 10.1007/978-3-030-01261-820.

Appendix A. Model architecture

A.1. Backbone

Encoder UM–ProtoShare uses a 3D extension of ResNet–152 (He et al., 2016) as the encoder backbone, a well-established model for brain tumour classification using MRI (Gupta et al., 2024; Kong et al., 2025). All 2D components (convolutions, batch normalisation, pooling, residual blocks) are converted to their 3D counterparts. To preserve localisation, the 3D ResNet–152 is truncated after the second residual stage (Wei et al., 2024), yielding three feature scales $s \in \{1, 2, 3\}$. These correspond to the outputs of the initial stage, residual stage (1), and residual stage (2) (See Figure 3). We denote these feature vectors by $\boldsymbol{f}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$, where C_s, H_s, W_s and D_s are the number of channels, height, width and depth at scale s. Inspired by (Ebrahimi et al., 2020), we employ transfer learning by initialising the 3D encoder from ImageNet-pretrained 2D ResNet–152 weights. To this end, convolutional filters are repeated along the depth axis to form 3D kernels, and the first convolution is adapted to four-channel input by copying the pretrained weights for the first three channels and initialising the fourth channel with their mean (Cui et al., 2018).

Decoder To enhance locality, we append a lightweight 3D UNet–style decoder to the encoder. As depicted in Figure 3 in, it comprises a bottleneck block at the deepest spatial scale to stabilise the input (Ahir and Parikh, 2025), followed by two decoder blocks that each perform trilinear upsampling, a convolution, batch normalisation and ReLU activation. No skip connections are used inside the decoder. We adopt interpolation-based upsampling rather than transposed convolution because transposed convolutions can introduce uneven overlap artefacts, whereas trilinear upsampling avoids these artefacts and is more computationally efficient in decoder stages for medical imaging analysis (Sanjar et al., 2020; Alwadee et al., 2025). The resulting decoder features are denoted by $\tilde{\boldsymbol{f}}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ for $s \in \{1, 2, 3\}$, with shapes matched to the encoder features at each scale to enable gated fusion.

Gated fusion We fuse encoder features $\boldsymbol{f}_s(x)$ and decoder features $\tilde{\boldsymbol{f}}_s(x)$ using per-channel gated fusion mechanisms (Oktay et al., 2018; Yu et al., 2018). As illustrated in Figure 3, for each scale the encoder and decoder features are first concatenated, globally averaged over the spatial dimensions, and then passed through a gating block composed

of two convolutional layers with a ReLU activation in between, as in (Yu et al., 2018). A sigmoid activation then yields a per-channel gate vector $\boldsymbol{\alpha}_s \in (0,1)^{C_s}$, which is used as per-channel weights to modulate the relative contributions of the corresponding features,

$$\boldsymbol{F}_s(x) = \boldsymbol{\alpha}_s \odot \boldsymbol{f}_s(x) + \big(1 - \boldsymbol{\alpha}_s\big) \odot \tilde{\boldsymbol{f}}_s(x). \tag{10}$$

Where $\odot$ denotes channel-wise multiplication. At each scale, the fused features $\boldsymbol{F}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ preserve the shapes of $\boldsymbol{f}_s(x)$ and $\tilde{\boldsymbol{f}}_s(x)$.

A.2. Localisation component

Following prior case–based models (Wei et al., 2024; Li et al., 2024), the localisation component comprises two modules: an add–on module and a mapping module, as presented in Figure 4.

add–on module At each scale, a convolutional block comprising convolution operations, batch normalisation and ReLU activation projects the fused features $\boldsymbol{F}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ to a fixed embedding dimension while preserving the spatial dimensions, yielding $\boldsymbol{G}_s(x) \in \mathbb{R}^{E \times H_s \times W_s \times D_s}$. This acts as a channel-harmonising projection, with (H_s, W_s, D_s) unchanged. We set $E = 128$ at all scales to ensure simplicity and comparability.

Mapping module At each scale, a convolutional block consisting of convolutions, batch normalisation and ReLU activation predicts per–prototype attention maps. We denote the raw stack of these maps by $\boldsymbol{M}_s^0(x) \in \mathbb{R}^{K_s \times H_s \times W_s \times D_s}$, and the map for a single prototype $\boldsymbol{p}_s^{k_s}$ by $\boldsymbol{M}_s^{0,k_s}(x) \in \mathbb{R}^{H_s \times W_s \times D_s}$. A sigmoid activation rescales the values of $\boldsymbol{M}_s^{0,k_s}(x)$ to the range of $[0,1]$. To sharpen attention maps and suppress irrelevant background, we further apply a differentiable soft mask (Li et al., 2018):

$$\boldsymbol{M}_s^{1,k_s}(x) = \frac{1}{1 + \exp\big(-w\big(\boldsymbol{M}_s^{0,k_s}(x) - \sigma\big)\big)}. \tag{11}$$

Where w and σ are hyperparameters, set to $w = 10$ and $\sigma = 0.5$ (Li et al., 2018). We then obtain size-invariant attention maps by ℓ_1-normalisation over the spatial dimensions:

$$\boldsymbol{M}_s^{k_s}(x) = \frac{\boldsymbol{M}_s^{1,k_s}(x)}{\sum \boldsymbol{M}_s^{1,k_s}(x) + \varepsilon}, \quad \varepsilon = 10^{-6}. \tag{12}$$

Where the sum is taken over all spatial locations. Stacking the normalised per–prototype maps over k_s yields $\boldsymbol{M}_s(x) \in \mathbb{R}^{K_s \times H_s \times W_s \times D_s}$.

Prototype descriptors For each prototype $\boldsymbol{p}_s^{k_s}$ at scale s, the corresponding attention map $\boldsymbol{M}_s^{k_s}(x)$ is broadcast across the E channels of $\boldsymbol{G}_s(x) \in \mathbb{R}^{E \times H_s \times W_s \times D_s}$. We then perform a voxel-wise (elementwise) product and apply global average pooling over the spatial dimensions (H_s, W_s, D_s) to obtain the corresponding feature descriptor $\boldsymbol{H}_s^{k_s}(x) \in \mathbb{R}^{E \times 1 \times 1 \times 1}$. Stacking descriptors across prototypes yields $\boldsymbol{H}_s(x) \in \mathbb{R}^{K_s \times E \times 1 \times 1 \times 1}$, which serves as the per-scale input to the prototype component of the model. Formally,

$$\boldsymbol{H}_s^{k_s}(x) = \frac{1}{H_s W_s D_s} \sum_{m=1}^{H_s} \sum_{n=1}^{W_s} \sum_{\ell=1}^{D_s} \big(\boldsymbol{G}_s(x) \odot \boldsymbol{M}_s^{k_s}(x)\big)_{m,n,\ell}. \tag{13}$$

Where $\odot$ denotes elementwise multiplication and the subscript (m,n,ℓ) indexes spatial location (m,n,ℓ) across all E channels.

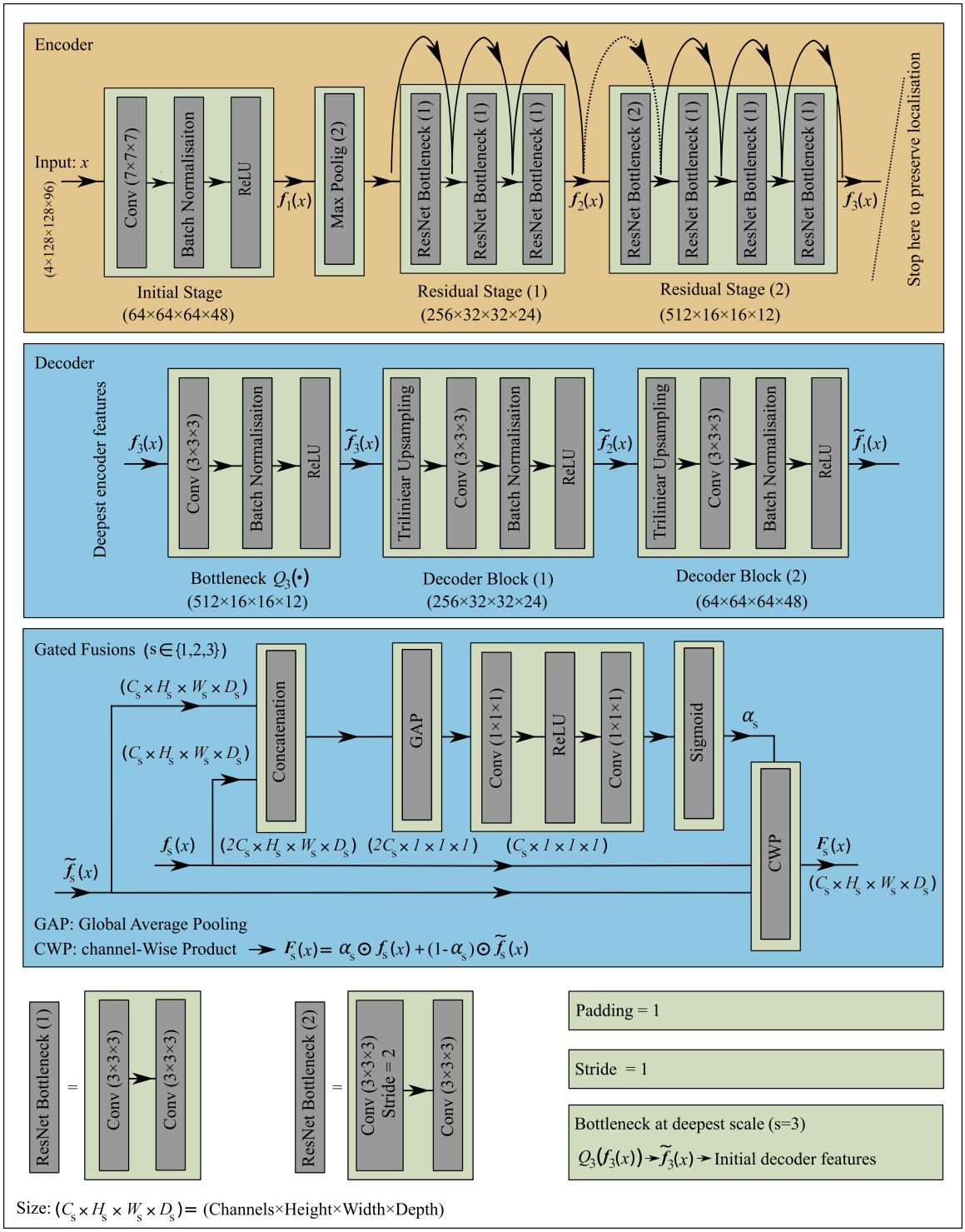

Figure 3: Backbone design. A 3D ResNet–152 encoder, truncated after residual stage (2), provides three feature scales $\boldsymbol{f}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$ for $s \in \{1, 2, 3\}$. A lightweight UNet–style decoder with a bottleneck and two decoder blocks (trilinear upsampling followed by Conv–BatchNorm–ReLU) produces $\tilde{\boldsymbol{f}}_s(x) \in \mathbb{R}^{C_s \times H_s \times W_s \times D_s}$, with shapes matching the corresponding encoder features. Gated fusions combine $\boldsymbol{f}_s(x)$ and $\tilde{\boldsymbol{f}}_s(x)$ at each spatial scale to produce the final feature maps $\boldsymbol{F}_s(x)$.

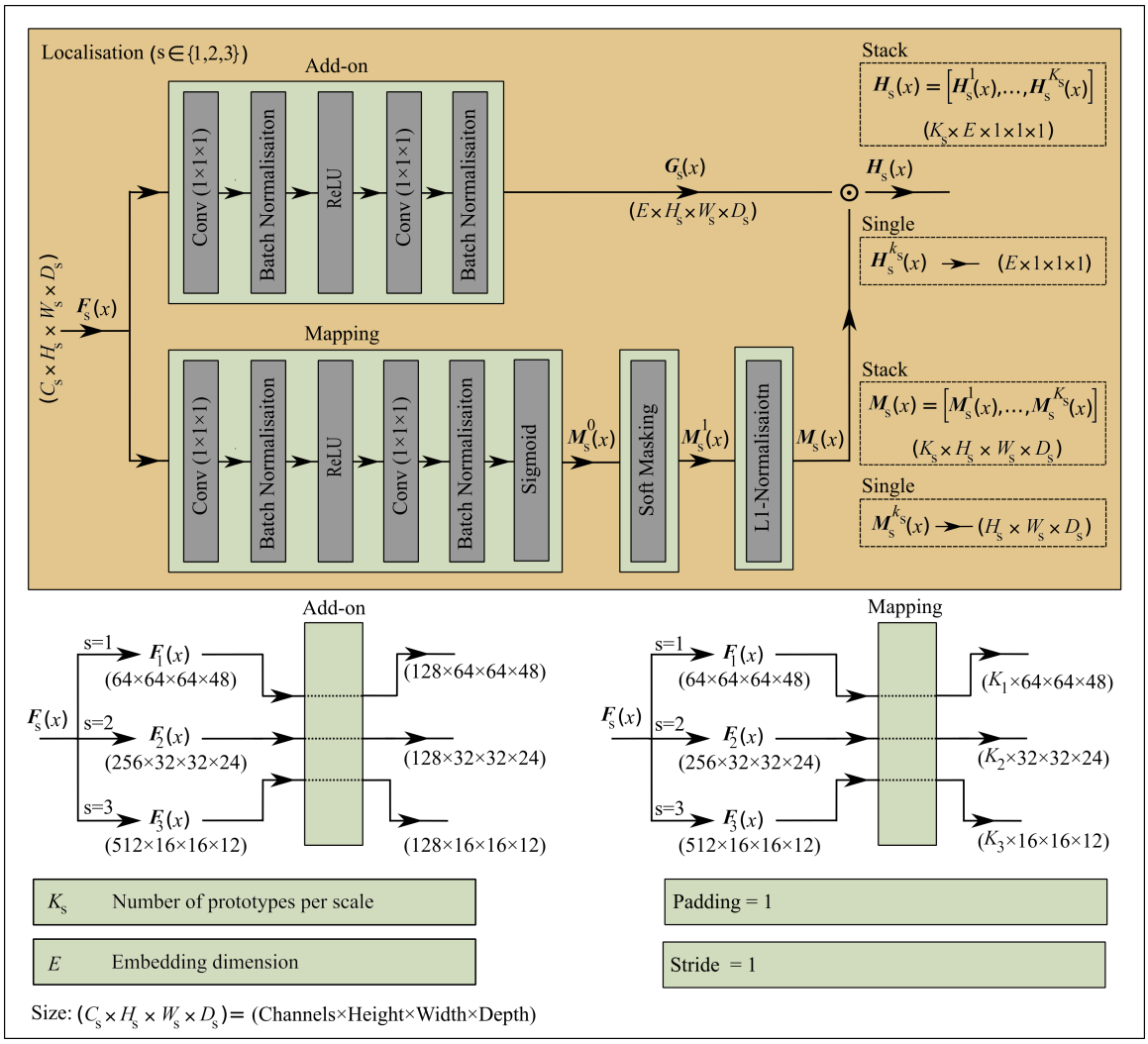

Figure 4: Localisation component.

A.3. Prototypes

Prototype bank UM–ProtoShare uses a bank of shared, class–agnostic, multi–scale prototypes. For scales $s \in \{1, 2, 3\}$, the bank is defined as

$$\boldsymbol{P} = \bigcup_{s=1}^{3} \boldsymbol{P}_s, \quad \boldsymbol{P}_s = \{\boldsymbol{p}_s^{k_s} \mid k_s \in \{1, \ldots, K_s\}\}. \tag{14}$$

Where each prototype $\boldsymbol{p}_s^{k_s} \in \mathbb{R}^{E\times1\times1\times1}$ is a learnable vector. Following prior case–based models (Wei et al., 2024; Li et al., 2024), we fix the total number of prototypes to $K_{\text{total}} = \sum_{s=1}^{3} K_s = 30$.

Prototype similarity At each scale, the cosine similarity between each prototype $\boldsymbol{p}_s^{k_s}$ and its corresponding feature descriptor $\boldsymbol{H}_s^{k_s}(x)$ is calculated,

$$S_s^{k_s}\left(x, \boldsymbol{p}_s^{k_s}\right) = \frac{\boldsymbol{H}_s^{k_s}(x) \cdot \boldsymbol{p}_s^{k_s}}{\left\|\boldsymbol{H}_s^{k_s}(x)\right\| \left\|\boldsymbol{p}_s^{k_s}\right\|}. \tag{15}$$

We then concatenate similarities across all prototypes and stack it into a single similarity vector $S(x, \boldsymbol{P}) \in \mathbb{R}^{K_{\text{total}}}$ to pass it to the classification component.

$$S(x, \boldsymbol{P}) = \left[S_1^1(x, \boldsymbol{p}_1^1), \ldots, S_1^{K_1}(x, \boldsymbol{p}_1^{K_1}), S_2^1(x, \boldsymbol{p}_2^1), \ldots, S_2^{K_2}(x, \boldsymbol{p}_2^{K_2}), S_3^1(x, \boldsymbol{p}_3^1), \ldots, S_3^{K_3}(x, \boldsymbol{p}_3^{K_3})\right]. \tag{16}$$

A.4. Classification.

A linear classifier maps the similarity vector to class logits. To this end, similarity scores are multiplied to classifier weights $\boldsymbol{w}_{\text{csl}} \in \mathbb{R}^{C\times K_{\text{total}}}$ (C is the number of classes). A softMax function then outputs the final predicted probabilities of $\boldsymbol{P}$.

$$\boldsymbol{y} = \text{softMax}\big(\boldsymbol{w}_{\text{csl}} \cdot S(x, \boldsymbol{P})\big). \tag{17}$$

Appendix B. Training of UM–ProtoShare

UM–ProtoShare is trained using the standard three-stage procedure adopted in case–based models (Chen et al., 2019; Wei et al., 2024):

Stage (1) optimisation of components preceding the classification layer (Latent space) In stage (1) we optimise all UM–ProtoShare layers up to and including the prototype bank. The goal is to obtain meaningful feature descriptors and prototype similarity patterns. The overall loss function for this stage is,

$$L_{\text{stage 1}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{clst}} L_{\text{clst}}^{\text{class-agnostic}} + \lambda_{\text{sep}} L_{\text{sep}}^{\text{class-agnostic}} + \frac{\lambda_{\text{map}}}{3} \sum_{s=1}^{3} L_{\text{map}}^{s} + \frac{\lambda_{\text{OC}}}{3} \sum_{s=1}^{3} L_{\text{OC}}^{s} + \lambda_{\text{div}} L_{\text{div}}. \tag{18}$$

Where λ_{cls}, λ_{clst}, λ_{sep}, λ_{map}, λ_{OC}, and λ_{div} are coefficients of respective loss terms. The classification loss L_{cls} is the cross-entropy on class logits. The clustering loss $L_{\text{clst}}^{\text{class-agnostic}}$ and separation loss $L_{\text{sep}}^{\text{class-agnostic}}$ are formulated to be compatible with shared, class–agnostic

prototypes, in contrast to class–specific design in UM–ProtoShare counterparts (Chen et al., 2019; Wei et al., 2024; Li et al., 2024), using soft prototype–class coefficients $\zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{k_s}}$. These losses encourage high similarity to prototypes that support the ground-truth class and discourage similarity to prototypes that support other classes, respectively. The multi–scale mapping loss (L^s_{map}), and Online–CAM loss (L^s_{map}) follow the formulation of MProtoNet at each scale (Wei et al., 2024), and help localisation of prototype attention maps. The prototype–diversity regulariser L_{div} penalises average pairwise cosine similarity between prototypes at each scale to reduce redundancy in the prototype bank.

Stage (2) prototype reassignment ("push") After the latent space has been optimised in Stage (1), we perform a prototype reassignment step to ensure that each prototype corresponds to a specific region of a training image. During this stage, all network parameters are frozen and, for each scale s, we replace the prototype $\boldsymbol{p}_s^{k_s}$ with its nearest feature descriptor $\boldsymbol{H}_s^{k_s}(x)$ in the latent space at the same scale. Let $X_N = \{x_n \mid n \in \{1, \ldots, N\}\}$ denote the training images, and for each image x_n, $\boldsymbol{H}_s^{k_s}(x_n)$ denote the corresponding feature descriptors. We then search over all training images for each prototype $\boldsymbol{p}_s^{k_s}$,

$$\boldsymbol{p}_s^{k_s} \leftarrow \boldsymbol{H}_s^{k_s}(x_{n'}), \quad n' = \arg\max_{1 \le n \le N} S_s^{k_s}\big(x_n, \boldsymbol{p}_s^{k_s}\big). \tag{19}$$

Here $S_s^{k_s}(x_n, \boldsymbol{p}_s^{k_s})$ denote the similarity score between prototype $\boldsymbol{p}_s^{k_s}$ and its corresponding feature descriptor $\boldsymbol{H}_s^{k_s}(x_n)$ for image x_n. This "push" operation anchors each prototype to a specific latent descriptor from a real training image, enabling faithful visualisation via its attention map $\boldsymbol{M}_s^{k_s}(x_n)$.

Stage (3) training of the classification component After prototype reassignment, we train the classification layer separately to learn weight coefficients $\boldsymbol{w}_{\text{csl}} \in \mathbb{R}^{C \times K_{\text{total}}}$, that represent the contributions of prototypes to classification without changing their representations. The overall loss at this stage is,

$$L_{\text{stage 3}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{L_1} L_{L_1}. \tag{20}$$

Where L_{L_1} is the L1-regularisation term which is originated in the classification component of ProtoPNet (Chen et al., 2019). λ_{cls}, and λ_{L_1} are coefficients for respective losses.

Appendix C. Preprocessing and Data Augmentation

For each patient, all MRI sequences were co-registered to the SRI24 anatomical template (Rohlfing et al., 2010), resampled to 1.5 mm^3 isotropic voxels, and cropped to $128 \times 128 \times 96$ centred on the brain. Z-score normalisation was applied on each sequence. To mitigate data scarcity, stochastic data augmentation was applied during training, including (i) rotation and scaling, (ii) Gaussian noise, (iii) Gaussian blur, (iv) brightness adjustment, (v) contrast adjustment, (vi) simulation of low resolution, (vii) gamma augmentation, and (viii) mirroring. Detailed parameters for each augmentation operation are provided in (Isensee et al., 2021). To ensure fair comparison with prior case–based models, we followed the same preprocessing and augmentation pipelines as in (Wei et al., 2024; Li et al., 2024).

Appendix D. Evaluation Metrics and Training Hyper-parameters

D.1. Evaluation metrics

We evaluate UM–ProtoShare in terms of both classification performance and interpretability. Classification performance is assessed with Balanced Accuracy (BAC). Interpretability is assessed in terms of localisation coherence and correctness (Nauta et al., 2023), using Incremental Deletion Score (IDS) and Activation Precision (AP), respectively.

D.2. Balanced accuracy (BAC)

To mitigate the LGG/HGG class imbalance, we report Balanced Accuracy (BAC), defined as the mean of sensitivity and specificity,

$$\mathrm{BAC} = \frac{\mathrm{TPR} + \mathrm{TNR}}{2}. \tag{21}$$

where TPR is the true positive rate (sensitivity) and TNR is the true negative rate (specificity). This gives equal weight to each class and mitigates the impact of class imbalance.

D.3. Activation precision (AP)

Activation Precision (AP) measures the localisation coherence of the attribution maps by quantifying how well they align with the human-annotated tumour masks. Let $\boldsymbol{M}(x)$ denote the attention map for an input image x (The construction of $\boldsymbol{M}(x)$ is described in Appendix F) and let $\boldsymbol{W}_T(x)$ be the corresponding human-annotated tumour mask. We then consider the We first upsample $\boldsymbol{M}(x)$ to the input space using trilinear interpolation, denoted by $\mathrm{UpSapmle}(\boldsymbol{M}(x))$. A binary threshold function $T(\cdot)$ with threshold 0.5 (following (Barnett et al., 2021)) is then applied. AP is defined as,

$$\mathrm{AP} = \frac{|\boldsymbol{W}_T(x) \cap T(\mathrm{UpSapmle}(\boldsymbol{M}(x)))|}{|T(\mathrm{UpSapmle}(\boldsymbol{M}(x)))|}. \tag{22}$$

D.4. Incremental deletion score (IDS)

Incremental Deletion Score (IDS) measures the correctness and the faithfulness of the attribution maps to the model's decision. Given upsampled attribution map $\boldsymbol{M}(x)$ voxels are ranked in descending order of their attribution scores. A fixed fraction of the most highly attributed voxels is progressively removed (their intensities are replaced with 0), and the model's classification accuracy is recorded after each deletion step, resulting in an accuracy–deletion curve. IDS is defined as the normalised area under the accuracy-deletion curve over the specified start–end bounds. Lower IDS indicates that activation maps are truly critical for model reasoning (deletion causes a rapid accuracy drop) and thus indicates more faithful attribution (Nauta et al., 2023).

D.5. Hyper-parameters during training

Following the training practice in (Wei et al., 2024), we train UM–ProtoShare using 5-fold cross-validation as described in the main text. For each cross-validation fold, we first warm up the backbone for 50 epochs using AdamW optimiser with a learning rate of 10^{-3}, weight

decay of 10^{-2}, and a batch size of 32. In this warm-up stage, and following the strategy explained in the main text, we first initialise the encoder features and then train the whole backbone while jointly training the encoder, decoder, and gated fusion modules. Using the same training parameters, we then train UM–ProtoShare for 100 epochs while the decoder is frozen. This preserves the locality benefits of the decoder while keeping training efficiency comparable to prior case–based models. During these 100 epochs we apply a learning-rate scheduler with linear warm-up for the first 20 epochs, followed by cosine annealing for the remaining 80 epochs. After every 10 epochs of Stage (1), we execute Stage (2) (prototype reassignment to the closest feature descriptors), followed by Stage (3) for 10 epochs, during which the classification layer is trained with the Adam optimiser at a learning rate of 10^{-3}. Data augmentation described in Appendix C is not applied during Stage (2). The loss coefficients are $\lambda_{\text{cls}} = 1$, $\lambda_{\text{clst}} = 0.8$, $\lambda_{\text{sep}} = 0.08$, $\lambda_{\text{map}} = 0.5$, $\lambda_{\text{OC}} = 0.05$, $\lambda_{\text{div}} = 0.001$, and $\lambda_{L_1} = 0.01$.

Appendix E. Ablation study configurations

Concretely, A1–A2 compare class–specific and class–agnostic prototype banks in a single-scale, encoder–only setting; B introduces multi–scale prototypes without using the decoder; C enables the decoder with simple channel-wise concatenation; D replaces concatenation with gated fusions; and E varies the allocation of prototypes across scales while keeping the total prototype budget fixed. The total prototype budget is fixed at $K_{\text{total}} = 30$. For multi–scale variants (Ablations B–D), we use a balanced prototype allocation across scales $(K_1, K_2, K_3) = (10, 10, 10)$. Ablation E evaluates the effect of per-scale allocation (fine, mid, coarse). At each ablation the effects on the classification performance (BAC) and interpretability (IDS, AP) are investigated.

E.1. Ablation A (class–specific prototypes (A1) versus class–agnostic prototypes (A2))

To isolate the effect of prototype sharing, we disable the decoder, all gated fusions, and use only the deepest encoder features $\boldsymbol{f}_3(x)$. The prototype bank is restricted to a single scale with $K_3 = 30$. We then compare two configurations:

- **Ablation A1 (class–specific):** Two distinct slots of 15 prototypes are assigned to the two classes via a hard class–prototype ownership matrix. This corresponds to MProtoNet with a normalised mapping module (see Equation 2 in the main text). This configuration (A1) mirrors the MProtoNet design but adds normalised mapping head.
- **Ablation A2 (class–agnostic):** All 30 prototypes are shared across classes and may support either class through the soft prototype–class coefficients $\zeta^{\text{soft}}_{c,\boldsymbol{p}_s^{ks}}$ defined in Section 2.3

E.2. Ablation B (multi–scale prototypes, no decoder)

To assess the effect of explicit multi–scale prototypes, we use the three encoder features $\{\boldsymbol{f}_1(x), \boldsymbol{f}_2(x), \boldsymbol{f}_3(x)\}$, and train separate prototype sets per scale with $(K_1, K_2, K_3) =$

$(10, 10, 10)$ so that $K_{\text{total}} = 30$. The decoder and all gated fusions remain disabled, and the prototypes are shared and class–agnostic at all scales. Any differences relative to A2 can therefore be attributed to introducing multi–scale prototypes.

E.3. Ablation C (decoder effect)

In ablation C we enable the decoder while keeping the shared, class–agnostic, multi–scale prototype bank unchanged. Encoder and decoder feature maps at each scale are fused by channel-wise concatenation, followed by a linear projection ($1 \times 1 \times 1$ convolution) to restore the fused channels to C_s. All gated fusions remain disabled. This configuration isolates the contribution of the decoder to feature quality and prototype localisation, without the confounding effect of gated fusion.

E.4. Ablation D (gated fusion effect)

In ablation D we replace the simple channel-wise concatenation fusion in Ablation C with gated fusions, allowing the model to learn the relative contribution of encoder and decoder features at each scale (see Equation (10) is Appendix A.1) Equation 1 in the main text). Comparing D against C isolates the effect of gated fusions on the classification performance and interpretability quality.

E.5. Ablation E (prototype scale allocation)

While using a balanced prototype allocation across scales, providing capacity to capture tumour features from fine to coarse scales while preserving a balanced contribution from each scale $(K_1, K_2, K_3) = (10, 10, 10)$, in ablation E we investigate how per-scale prototype allocation affects classification performance (BAC) and interpretability (AP, IDS) under a fixed total budget of $K_{\text{total}} = 30$. We consider three allocation schemes:

- **Ablation E1 (Fine-heavy):** $(K_1, K_2, K_3) = (15, 9, 6)$
- **Ablation E2 (Mid-heavy):** $(K_1, K_2, K_3) = (9, 15, 6)$
- **Ablation E3 (Coarse-heavy):** $(K_1, K_2, K_3) = (6, 9, 15)$

All other architectural and training settings are kept identical across these configurations.

Appendix F. Comparative benchmark configurations

We compare UM–ProtoShare against a CNN model derived from the UM–ProtoShare backbone, ProtoPNet (Chen et al., 2019), XProtoNet (Kim et al., 2021), MProtoNet (Wei et al., 2024), and MAProtoNet (Li et al., 2024). These models are selected because they represent recent state-of-the-art case–based models that have been adapted for clinical imaging applications.

For a fair comparison, all models use identical data splits at each cross-validation fold, MRI preprocessing and augmentation, training schedules, and a prototype bank with the same total number of prototypes; only the architectural design differences are varied. ProtoPNet, XProtoNet and MProtoNet are re-implemented within the UM–ProtoShare codebase to ensure consistent architecture components and training routines. For MAProtoNet,

we use the public implementation and align the preprocessing and training procedure to match the other models. As all benchmark models use single-scale prototypes at the deepest backbone features, their prototype and attribution terms are written with subscript $s = 3$, for consistency with the UM–ProtoShare notation.

F.1. Model design

CNN model. The CNN baseline uses the same encoder-–decoder backbone as UM–ProtoShare. It retains the add–on module used in the localisation component (Figure 1) and connects it directly to the classification layer via global average pooling.

ProtoPNet ProtoPNet is adapted to multi–sequence 3D MRI with a single-scale, class–specific prototype bank. Prototype similarities $S_3^{k_3}(x, \boldsymbol{p}_3^{k_3})$ are computed by directly comparing the latent features $\boldsymbol{G}_3(x)$ with each prototype $\boldsymbol{p}_3^{k_3}$ using the top-α average pooling ($\alpha = 1\%$), reflecting the absence of mapping module in its design.

XProtoNet XProtoNet is likewise adapted to the same multi–sequence 3D MRI input. In contrast to ProtoPNet, it includes a mapping module that produces prototype attention maps, from which class–specific prototype similarities $S_3^{k_3}(x, \boldsymbol{p}_3^{k_3})$ are obtained.

MProtoNet MProtoNet extends XProtoNet by introducing soft masking and an Online–CAM module, which further refines attribution maps and logits. Prototypes are still single-scale and class–specific, but the attribution maps are influenced by the Online–CAM-based pooling.

MAProtoNet MAProtoNet further employs multi–scale feature fusion and a multi–scale mapping loss, while its prototype bank remains single-scale and class–specific. multi–scale features are fused before computing prototype similarities, whereas UM–ProtoShare uses a multi–scale, shared, class–agnostic prototype bank.

F.2. Attention map construction for AP and IDS

For the interpretability metrics (AP and IDS), we require attention map $\boldsymbol{M}(x)$ for each model (see Appendix D).

CNN model For the CNN baseline, $\boldsymbol{M}(x)$ is given by the Grad–CAM (Selvaraju et al., 2017) map computed from the last convolutional layer that highlights important regions for the class of interest.

ProtoPNet For ProtoPNet, there is no explicit mapping module. We therefore use the similarity scores as attribution: for each prototype we take the similarity scores over spatial locations (patches of $\boldsymbol{G}_3(x)$ with the same size of prototypes), select the top-α most activated patches ($\alpha = 1\%$), and average their scores to form an attention map. class–specific maps are obtained by aggregating only prototypes assigned to the corresponding class $\boldsymbol{M}(x)$.

XProtoNet, MProtoNet and MAProtoNet For XProtoNet, MProtoNet and MAProtoNet, we construct the attention map for class c by averaging the attention maps of all prototypes assigned to that class. Let $K_{3,c}$ denote the total number of prototypes assigned

to class c, and $\boldsymbol{M}_{3,c}^{k_3}(x)$ be the attention map of prototype $\boldsymbol{p}_3^{k_3}$ for class c. Then,

$$\boldsymbol{M}(x) = \frac{1}{K_{3,c}} \sum_{k_3=1}^{K_{3,c}} \boldsymbol{M}_{3,c}^{k_3}(x). \tag{23}$$

These attribution maps are then used to compute AP and IDS as described in Appendix D.

UM–ProtoShare In UM–ProtoShare, prototypes are shared across classes and their class–specific support is quantified by soft class–prototype coefficients $\zeta_{c,\boldsymbol{p}_s^{k_s}}^{\text{soft}}$, defined in the main text. So, if and only if $\zeta_{c,\boldsymbol{p}_s^{k_s}}^{\text{soft}} > 0$ the ReLU-activated classifier weight for class c and prototype $\boldsymbol{p}_s^{k_s}$ is positive (see Equation (6)). We therefore define the set of prototypes that support class c at scale s,

$$K_{(s,c)}^{(+)} = \{\boldsymbol{p}_s^{k_s} \mid \zeta_{c,\boldsymbol{p}_s^{k_s}}^{\text{soft}} > 0\}. \tag{24}$$

Let $\boldsymbol{M}_{c,s}^{k_s}(x)$ denote the attention map of prototype $\boldsymbol{p}_s^{k_s}$ for input x at scale s. The attention map $\boldsymbol{M}(x)$ for UM–ProtoShare is then obtained by averaging the attention maps of prototypes that support class c across all scales,

$$\boldsymbol{M}(x) = \sum_{s=1}^{3} \frac{1}{|K_{(s,c)}^{(+)}|} \sum_{\boldsymbol{p}_s^{k_s} \in K_{(s,c)}^{(+)}} \boldsymbol{M}_{c,s}^{k_s}(x). \tag{25}$$

These class–specific maps are then used to compute AP and IDS as described in Appendix D.

Appendix G. Extended results and discussion

Classification performance. We start from the single-scale, class–specific prototype model (A1) and progressively add the components introduced in the "Ablation studies" section. UM–ProtoShare variants A2–D obtain BAC scores of 87.43 ± 1.27 (A2), 88.85 ± 2.56 (B), 86.47 ± 2.85 (C) and 88.14 ± 2.74 (D), compared with 85.92 ± 2.14 for A1. Moving from class–specific to class–agnostic prototypes (A1→A2) thus improves BAC by about 1.5 percentage points, indicating that sharing prototypes across classes is beneficial in this setting where tumour features overlap across LGG/HGG. Introducing explicit multi–scale prototypes (A2→B) yields another ≈1.4-point gain, confirming that capturing multi–scale features is helpful for glioma grading. Adding the decoder module (B→C) trades some accuracy for richer localisation, reducing BAC to 86.47 ± 2.85. Incorporating gated fusions (C→D) recovers most of this loss, increasing BAC to 88.14 ± 2.74 while keeping the locality benefits of the decoder in place. Under a fixed total prototype budget, the E configurations allocate prototypes across scales. The fine-heavy allocation (E1) attains 87.90 ± 2.90 BAC, the mid-heavy configuration (E2) reaches 88.40 ± 2.80, and the coarse-heavy variant (E3) achieves the highest BAC among the ablation studies at 89.21 ± 3.18, close to the CNN's 89.73 ± 2.36.

Interpretability. While BAC varies within a relatively narrow range across ablations, interpretability improves steadily as we introduce the proposed components. AP increases from 81.38 ± 2.23 for A1 to 83.28 ± 3.41 for A2 and 85.89 ± 1.24 for B, and then to 87.51 ± 1.38 for C, before a small drop to 86.45 ± 1.38 for D. Overall, AP improves by about 6% between A1 and D. IDS follows the opposite trend (lower is better). After a small increase from 7.13 ± 2.93 (A1) to 7.49 ± 4.07 (A2), it decreases to 6.52 ± 3.29 (B), 5.23 ± 2.18 (C) and 5.36 ± 1.28 (D), corresponding to an overall ≈25% reduction relative to A1. Within the E configurations, interpretability can be further tuned. The fine-heavy allocation (E1) achieves the best localisation overall (AP 90.10 ± 1.80; IDS 4.60 ± 1.20), the mid-heavy variant (E2) offers strong interpretability (AP 88.72 ± 1.60; IDS 5.10 ± 1.30), and the coarse-heavy model (E3) trades some AP and IDS for higher BAC. These patterns mirror the architectural changes. class–agnostic sharing (A2) yields an initial AP gain without affecting IDS, suggesting the benefits of using shared class–agnostic prototypes. multi–scale prototypes (B) simultaneously raise AP and lower IDS, indicating more precise and stable localisation across scales. The decoder (C) brings the largest drop in IDS, sharpening the focus of the activation maps, and gated fusions (D) slightly decrease AP while maintaining low IDS to balance classification and localisation. Finally, reallocating prototypes across scales (E1–E3) modulates this balance. Emphasising fine-scale prototypes (E1) prioritises detailed, tumour-focused features, shifting capacity towards coarse scales (E3) favours global contextual features and higher BAC at a modest cost in AP and IDS, and the mid-heavy configuration (E2) sits between these configurations, providing the best overall accuracy–interpretability trade-off. We therefore use E2 as our main UM–ProtoShare configuration.

Comparative benchmark. We position UM–ProtoShare against existing case–based models and the CNN baseline. The CNN achieves the highest BAC overall (89.73 ± 2.36). Among prior case–based models, MAProtoNet attains the highest BAC (87.36 ± 3.26), with ProtoPNet and XProtoNet slightly lower (86.42 ± 1.78 and 87.23 ± 1.59), and MProtoNet the weakest (84.69 ± 3.03). In terms of interpretability, MAProtoNet again performs best (AP 85.29 ± 4.14; IDS 6.83 ± 3.12) and improves over MProtoNet (AP 79.93 ± 1.32; IDS 7.53 ± 3.24), whereas ProtoPNet and XProtoNet show substantially poorer localisation (AP 11.73 ± 3.02 / 18.12 ± 2.76; IDS 21.56 ± 5.83 / 17.46 ± 4.98).

Using the mid-heavy allocation (E2) as our main UM–ProtoShare configuration, we obtain 88.40 ± 2.80 BAC, 88.72 ± 1.60 AP and 5.10 ± 1.30 IDS. Compared with MAProtoNet (87.36 ± 3.26 BAC, 85.29 ± 4.14 AP, 6.83 ± 3.12 IDS), E2 therefore improves classification and interpretability simultaneously, with approximately +1.0 percentage point higher BAC, +3.4 points higher AP and 1.7 points lower IDS (≈25% relative reduction). Relative to the CNN (89.73 ± 2.36 BAC, 10.65 ± 5.63 AP, 14.85 ± 5.21 IDS), E2 sacrifices about 1.3 percentage points in BAC but dramatically enhances interpretability, increasing AP by roughly 78 points (≈8.3×) and reducing IDS by about 66% (from 14.85 to 5.10).

The other UM–ProtoShare configurations trace out the accuracy–interpretability trade-off. The gated fusion variant (D; 88.14 ± 2.74 BAC, 86.45 ± 1.38 AP, 5.36 ± 1.28 IDS) already surpasses MAProtoNet by around +0.8 BAC, +1.2 AP and 1.5 points lower IDS; the coarse-heavy model (E3; 89.21 ± 3.18 BAC, 85.80 ± 2.08 AP, 5.98 ± 2.57 IDS) nearly matches the CNN's BAC (−0.5 points) while retaining localisation close to D (≈0.7 points

lower AP and 0.6 points higher IDS); and the fine-heavy model (E1; 87.90 ± 2.90 BAC, 90.10 ± 1.80 AP, 4.60 ± 1.20 IDS) delivers the most faithful interpretability attributions, improving over D by about +3.6 AP and 0.8 points lower IDS at a small cost of 0.2 percentage points in BAC.

Extended discussion. UM–ProtoShare is a case–based model that combines a 3D ResNet–152 encoder, a UNet–style decoder with gated fusions, and a bank of shared, class–agnostic, multi–scale prototypes for brain tumour classification from multi–sequence 3D MRI. By integrating transfer learning through a pre-trained 3D encoder, and by jointly optimising a normalised mapping module, class–agnostic prototype sharing and explicit multi–scale prototype banks, UM–ProtoShare aims to mitigate the conventional accuracy–interpretability trade-off in case–based models. The ablation study shows that these components collectively improve both BAC and interpretability metrics (AP, IDS) relative to prior case–based models, while remaining competitive with a strong CNN model.

Normalising the mapping module helps reduce tumour size effects in the attention maps. Moving from MProtoNet to the A1 variant, which mirrors the MProtoNet architecture but uses the normalised mapping, yields modest gains in BAC and AP and a reduction in IDS, consistent with the goal of attenuating size bias and producing cleaner evidence maps. Normalisation also makes attention values more comparable across different tumour volumes, so evidence reflects where the model focuses rather than how large the lesion is, in line with recommendations from weakly supervised localisation methods where normalised attention improves localisation coherence (Xu et al., 2022). Adding a UNet–style decoder recovers spatial resolution and fine structure attenuated by encoder downsampling, a common strategy in medical imaging to sharpen boundaries and improve localisation over encoder–only designs (Oktay et al., 2018; Navab et al., 2015). In our ablation (B→C), the decoder increases AP and reduces IDS at a modest cost in BAC, suggesting more precise attention maps under weak supervision (Oktay et al., 2018). Replacing simple concatenation with encoder-–decoder gated fusions (C→D) then regulates how features contribute at each scale, recovering most of the lost accuracy while maintaining low IDS and high AP, and yielding clearer, scale-coherent maps (Oktay et al., 2018; Hu et al., 2018; Takikawa et al., 2019). Under a fixed prototype budget ($K_{\text{total}} = 30$), the E-variants illustrate a scale-dependent accuracy–interpretability trade-off: fine-heavy allocation emphasises edge and boundary cues (e.g., thin enhancing rims, irregular margins), maximising AP and minimising IDS; coarse-heavy allocation emphasises global context (oedema extent, mass effect), achieving the highest BAC; and the mid-heavy configuration (E2) provides the most balanced operating point.

class–agnostic prototype sharing and an explicit multi–scale prototype bank are the central architectural choices that distinguish UM–ProtoShare from prior case–based models. Shared, class–agnostic prototypes allow the model to represent appearance patterns that occur in both tumour grades, which is realistic for multi–sequence MRI where grades often differ by intensity and extent rather than entirely distinct morphologies (Haydar et al., 2022). Sharing also reduces redundancy in the prototype space and improves data efficiency, as high-quality prototypes can be reused across classes instead of duplicated (Rymarczyk et al., 2021). In the ablation, moving from class–specific (A1) to class–agnostic (A2) prototypes increases BAC and AP with only a mild change in IDS, and qualitatively

provides broader coverage across MRI sequences, particularly in mixed or borderline cases. A potential risk is over-sharing, where prototypes drift towards the majority class; in UM–ProtoShare this is mitigated by soft, class-conditioned assignment in the separation loss and by a diversity regulariser (Gautam et al., 2023). Overall, shared prototypes improve coverage of relevant tumour features and contribute to the observed gains in interpretability without sacrificing classification performance or increasing the number of prototypes.

The multi–scale prototype bank enables attention from fine- to coarse-grained regions in MRI. Smaller-scale prototypes sharpen localisation, whereas larger-scale prototypes preserve global anatomical context and reduce sensitivity to small spurious textures, a known issue for case–based models (Lu et al., 2025), thereby helping to balance accuracy and interpretability (Li et al., 2024). This design is well matched to the multi–scale imaging characteristics of brain tumours, where fine-grained cues (rim or irregular enhancement, small non-enhancing components, irregular margins) coexist with coarse-grained cues (oedema extent, mass effect, cross-sequence concordance) (Wegscheid and Jennings, 2024; Martucci et al., 2023). The approach also aligns with radiomics, which routinely applies multi–scale filters (e.g., wavelets, Laplacian of Gaussian) and has been standardised by the Image Biomarker Standardisation Initiative (IBSI) [55, 56]. Consistently, our ablation (A2→B) shows that introducing an explicit multi–scale prototype bank improves both BAC and interpretability metrics.

Regarding multi–scale capabilities, the closest counterparts are MAProtoNet and HQProtoPNet (Wang et al., 2023). MAProtoNet fuses multi–scale encoder features before the mapping module and supervises them with a multi–scale mapping loss, but its prototypes remain single-scale and class–specific. HQProtoPNet retains single-scale, class–specific prototypes and attains multi–scale behaviour via a multi–scale matching layer that reshapes backbone features into multiple pooled grids, so the same prototype matches across scales without distinct per-scale banks. By contrast, UM–ProtoShare learns explicit prototype banks at fine, mid and coarse scales and enhances locality through a UNet–style decoder with gated fusions. In terms of prototype sharing, the closest methods are ProtoPool (Rymarczyk et al., 2022), ProtoPShare (Rymarczyk et al., 2021) and structurally ProtoTree (Nauta et al., 2021). ProtoPShare introduces sharing post–hoc by merge-and-prune, ProtoTree shares prototypes implicitly via decision-tree routing, and ProtoPool maintains a global prototype pool with differentiable Gumbel–softMax slot assignment. UM–ProtoShare instead learns shared prototypes using normalised soft prototype–class coefficients derived from Grad–CAM-style importance weights, so that prototypes can support multiple classes while clustering and separation losses are modulated accordingly. This mechanism integrates naturally with the mapping module and Online–CAM used to supervise prototype attention maps, yielding a unified framework for multi–scale, class–agnostic, case–based explanations in multi–sequence 3D MRI.

Limitations and future work. Although the dataset is multi-institutional, the sample size remains limited for 3D multi–sequence MRI, particularly for LGG cases. Further evaluation on larger cohorts with external validation datasets is needed to assess the generalisability of UM–ProtoShare. From a modelling perspective, prototypes are learned from a stack of MRI sequences over the entire volume. This design simplifies training and encourages cross-sequence consistency, but it restricts sequence-specific attribution and region-

level control. In particular, it can obscure which sequence (e.g., T1 vs. FLAIR) drives a given prototype match and may allow off-target matches outside clinically salient subregions such as enhancing core, non-enhancing core, or oedema. Future work should investigate sequence-wise prototype banks and sequence-selective gated fusions to better disentangle modality- and region-specific evidence. Finally, as in other case–based models, prototypes capture recurring appearance patterns without explicit high-level clinical semantics (e.g., necrosis, infiltrative margins). Combining UM–ProtoShare with concept-based approaches, such as concept bottlenecks or Testing with Concept Activation Vectors (TCAV) models, is a promising direction to enrich explanations with human–interpretable concepts while preserving case–based evidence.