A. Detailed Experimental Setup

A.1. Pretraining Configuration

All experiments use 8×NVIDIA A100 40GB GPUs with mixed precision training. Following the OpenMind preprocessing protocol [62], input volumes undergo: (1) resampling to 1mm isotropic spacing; (2) z-score intensity normalization per volume; (3) center-cropping or zero-padding to 160^3 voxels. We use the AdamW optimizer, combined with a linear warmup followed by cosine decay learning rate schedule. Training runs for a maximum of 2000 epochs with early stopping.

A.2. Dataset Specifications and Splitting Protocol

All evaluation datasets use standardized 160^3 voxel volumes with stratified class-balanced splits. BraTS18 [2, 43] contains ∼285 subjects with 4 MRI modalities (T1, T1-Gd, T2, FLAIR). BraTS23 [14] comprises ∼1200 subjects with 4 modalities (T1, T1-Gd, T2, FLAIR). RSNA-MICCAI contains ∼600 training subjects and ∼90 test subjects with 4 modalities (T1, T1-Gd, T2, FLAIR). TCGA-GBM contains ∼200 subjects with 4 modalities (T1, T1-Gd, T2, FLAIR). UCSF-PDGM [5] provides ∼200 subjects with 6 MRI sequences (T1, T1c, T2, FLAIR, SWI, ASL). UPenn-GBM contains ∼600 subjects with 4 modalities (T1, T1-Gd, T2, FLAIR). All datasets use 60/20/20 train/validation/test splits.

A.2.1. Downstream Task Specifications

We evaluate on the following binary classification tasks spanning tumor grading, molecular marker prediction, and survival analysis (Table 6). For survival tasks (DSS, PFI), patients censored before the 1-year threshold are excluded due to unknown outcomes.

Table 6. Task specifications with clinical definitions and label sources.

Dataset	Task	Clinical Definition	Class 0	Class 1	Label Criterion
BraTS18	Tumor Grading	Glioma grading (WHO grade)	LGG	HGG	Histological grade
BraTS23	Tumor Type	Tumor type discrimination	Metastasis	Glioma	Histopathology
RSNA MICCAI	MGMT Methylation	MGMT promoter status	Unmethylated	Methylated	Molecular testing
TCGA GBM	DSS 1-year	Disease-specific survival	≤365 days	>365 days	Survival time >365 days (Class 1) or ≤365 days (Class 0)
	PFI 1-year	Progression-free interval	≤365 days	>365 days	Progression-free time >365 days (Class 1) or ≤365 days (Class 0)
UCSF PDGM	IDH Classification	IDH mutation status	Wildtype	Mutant	Genetic sequencing
UPENN GBM	Age Group	Age stratification	Age <65	Age ≥65	Clinical records
	IDH1 Status	Molecular marker	Mutant/Unknown	Wildtype	Genetic testing
	GTR Status	Surgical resection	GTR ≤90%	GTR >90%	Surgical reports

A.3. Baseline Implementations

We compare against representative SSL methods spanning three paradigms: (1) *Invariance-based contrastive learning*: SimCLR [9] and VoCo [66]; (2) *Masked image modeling*: MAE [24]; (3) *Medical-specific multi-objective methods*: Models Genesis [86], Volume Fusion [63], S3D, and SwinUNETR [54]. We additionally compare against medical foundation models BrainMVP [47], BrainIAC [52], and MRI-Core [15], and general vision models DinoV2 [44] and ResNet-50. All SSL baselines are pretrained on identical OpenMind [62] dataset with matched compute budget; foundation models use official pretrained weights without additional training.

A.4. Fine-tuning Protocol

Architecture. All SSL baselines use the ResEncL architecture introduced in OpenMind [62], a 6-stage ResNet-based U-Net encoder-decoder with features [32, 64, 128, 256, 320, 320], residual blocks [1, 3, 4, 6, 6, 6] per stage, 3D convolutions, instance normalization, and skip connections. This convolutional architecture was selected based on OpenMind [62] findings demonstrating that convolutional encoders outperform transformer-based alternatives for 3D medical imaging. Foundation models (BrainMVP, BrainIAC, MRI-Core, DinoV2) use their official pretrained architectures from respective codebases.

MDAE extends the ResEncL architecture with time-conditioning mechanisms for handling dual corruption during pretraining. Specifically, the time-conditioned encoder uses Feature-wise Linear Modulation (FiLM [46]) at each stage and time-conditioned 3D self-attention at deeper layers (stages 3-5). Time embeddings are generated via sinusoidal encoding followed by an MLP, then injected through FiLM layers: $h_{\text{out}} = h_{\text{in}} \odot (\gamma(t_{\text{emb}}) + 1) + \beta(t_{\text{emb}})$ where γ and β are learned scale and shift parameters. At stages 3-5, time-conditioned 3D self-attention modules [22] further modulate spatial features by adding time-dependent shifts to query, key, and value projections. This time-conditioned architecture, trained during MDAE pretraining with $t \sim \mathcal{U}(0, 1)$, is adapted to downstream tasks via two approaches (Section C): disabling time-conditioning (Section C.3) or maintaining $t = 0$ for clean inputs (Section C.4). For classification, we attach a trainable 2-layer MLP classification head. In multi-modality settings, global average pooling aggregates features across modalities; for single-modality tasks, features are used directly. Full encoder finetuning allows both time-conditioning parameters and feature extraction layers to adapt to downstream tasks.

Hyperparameter search and model selection. Grid search over: learning rate $\in [10^{-5}, 10^{-4}]$; batch size $\in \{2, 4, 8\}$; epochs $\in \{100, 200, 300\}$. For each configuration, we select the best checkpoint based on validation performance and report final results on the held-out test set.

Evaluation metrics. For classification, we report AUROC (area under ROC curve) and average precision (AP), both providing threshold-independent assessment of classifier performance. For segmentation, we report Dice coefficient and Normalized Surface Distance (NSD) to evaluate region overlap and boundary accuracy respectively. All metrics are computed on held-out test sets.

B. MDAE Training Algorithm

Algorithm 1 details the MDAE training procedure. The core mechanism is the simultaneous application of dual corruptions: we sample both the masking ratio $p_{\text{mask}} \sim \mathcal{U}(p_{\min}, p_{\max})$ and diffusion timestep $t \sim \mathcal{U}(0, 1)$ together, creating doubly-corrupted inputs $\tilde{X}_t^M = M_v \odot \tilde{X}_t$ where $\tilde{X}_t = X_0 + \sigma_t Z$ with $\sigma_t = t \cdot \sigma_{\max}$ (we use VE noise schedule for notation simplicity). The training objective jointly optimizes two complementary losses: (1) $\mathcal{L}_{\text{masked}}$ for reconstructing masked regions Ω_M, and (2) $\mathcal{L}_{\text{visible}}$ for denoising visible regions Ω_V with noise-level weighting $w(\sigma_t)$.

C. Downstream Adaptation Algorithms

After pretraining, we adapt the learned encoders to downstream tasks via supervised finetuning. For non-time-conditioned baselines (MAE, SimCLR, VoCo, etc.), adaptation involves attaching task-specific heads (e.g., 2-layer MLP for classification, decoder for segmentation) and training end-to-end on labeled data. For MDAE's time-conditioned encoder, we consider two common adaptation strategies for handling the time-conditioning mechanism:

C.1. Approach 1: Disable time-conditioning.

Setting the time parameter to None (t = None) skips the time-conditioning mechanisms during finetuning, rendering the architecture functionally equivalent to a standard ResNet encoder used by other baseline methods. The FiLM layers and time-conditioned attention modules are bypassed to act as identity functions. The encoder processes inputs as $f_\theta(x)$ without time modulation. For classification tasks, we adopt this approach as it is supported by our ablation studies showing that time-conditioning provides marginal improvements.

C.2. Approach 2: Set t=0 for clean input prediction.

This approach maintains the time-conditioned architecture active but fixes the time parameter to zero ($t = 0$). The rationale is to utilize the fact that during finetuning, we have direct access to clean, uncorrupted medical images, unlike the corrupted inputs seen during pretraining. The time value $t = 0$ is processed through the full time embedding pipeline, allowing FiLM layers and time attention modules to learn task-specific modulation parameters optimized for clean inputs. Notably, our framework can be naturally adapted to handle noisy inputs by simply setting t to match the corruption patterns.

While more advanced adaptation approaches exist (e.g., data augmentation during finetuning, leveraging noise-robustness for uncertainty estimation), extensions to more sophisticated adaptation schemes are left for future work.

C.3. Classification Finetuning

For classification, we adopt Approach 1 (disabling time-conditioning) as detailed in Section C. Algorithm 2 details the classification finetuning procedure.

Algorithm 1 MDAE Training Algorithm

Require: Dataset $\mathcal{D} = \{X_0^{(i)}\}_{i=1}^N$, encoder-decoder network g_θ
Require: Corruption hyperparameters: Masking bounds $p_{\min}, p_{\max}$, max noise level $\sigma_{\max}$
Require: Noise schedule: VE with $\sigma(t) = t \cdot \sigma_{\max}$ ▷ Use VE [50]; can use VP or Flow
Require: Training: Learning rate η, batch size B, gradient clip norm ν, max epochs E
Ensure: Trained encoder-decoder network g_θ

Initialize network parameters θ
for epoch $= 1, \ldots, E$ **do**
  for each batch $\{X_0^{(i)}\}_{i=1}^B \sim \mathcal{D}$ **do**
    for $i = 1, \ldots, B$ **do**
      // Simultaneous dual corruption: masking and diffusion noise
      Sample $p_{\text{mask}}^{(i)} \sim \mathcal{U}(p_{\min}, p_{\max})$, $t^{(i)} \sim \mathcal{U}(0,1)$, $Z^{(i)} \sim \mathcal{N}(0, I)$
      Generate patch-based mask $M^{(i)} \in \{0,1\}^{D \times H \times W}$ (16^3 patches)
      Define visible mask: $M_v^{(i)} = \mathbf{1} - M^{(i)}$
      Apply noise: $\tilde{X}_t^{(i)} = X_0^{(i)} + \sigma_t^{(i)} Z^{(i)}$ ▷ where $\sigma_t = t \cdot \sigma_{\max}$
      Combine corruptions: $\tilde{X}_t^{M(i)} = M_v^{(i)} \odot \tilde{X}_t^{(i)}$
      Reconstruct volume: $\hat{X}^{(i)} = g_\theta(\tilde{X}_t^{M(i)}, t^{(i)})$ ▷ Time-conditioned with t
      // Compute dual objectives: masked reconstruction + visible denoising
      $\Omega_M^{(i)} = \{j \mid M_j^{(i)} = 1\}$, $\Omega_V^{(i)} = \{j \mid M_j^{(i)} = 0\}$ ▷ Masked and visible voxel sets
      Masked loss: $\ell_{\text{masked}}^{(i)} = \dfrac{1}{|\Omega_M^{(i)}|} \cdot \|M^{(i)} \odot (\hat{X}^{(i)} - X_0^{(i)})\|_2^2$
      Visible loss: $\ell_{\text{visible}}^{(i)} = \dfrac{w(\sigma_t^{(i)})}{|\Omega_V^{(i)}|} \cdot \|M_v^{(i)} \odot (\hat{X}^{(i)} - X_0^{(i)})\|_2^2$
      Combined loss: $\ell^{(i)} = \lambda_{\text{masked}} \cdot \ell_{\text{masked}}^{(i)} + \lambda_{\text{visible}} \cdot \ell_{\text{visible}}^{(i)}$
    end for
    Compute batch loss: $\mathcal{L} = \frac{1}{B} \sum_{i=1}^B \ell^{(i)}$
    Compute gradient: $g \leftarrow \nabla_\theta \mathcal{L}$
    Clip gradient: $g \leftarrow \text{clip}(g, \text{norm} = \nu)$
    Update parameters: $\theta \leftarrow \theta - \eta \cdot g$
  end for
end for
return Trained network g_θ

Algorithm 2 details the classification finetuning procedure using Approach 1. The design choice is disabling time-conditioning, which renders the encoder equivalent to a standard ResNet architecture as the other SSL baselines. The encoder extracts high-level features via global average pooling, which are classified by a 2-layer MLP head. We use standard cross-entropy loss for supervised training and both encoder and classification head are trained end-to-end.

C.4. Segmentation Finetuning

For dense segmentation tasks, we adopt Approach 2 (setting $t = 0$) to preserve the time-conditioned architecture. Segmentation incorporates multi-scale features from all 6 encoder stages via skip connections for pixel-wise prediction. Maintaining time-conditioning with $t = 0$ enables the network to preserve the architecture's expressiveness. We attach a lightweight decoder to the pretrained encoder, forming a full encoder-decoder U-Net architecture. The decoder uses transposed convolutions for upsampling and skip connections from the encoder to preserve spatial details.

Algorithm 3 details the finetuning procedure. The encoder extracts multi-scale features with skip connections, which the decoder upsamples to produce pixel-wise segmentation logits. We use a combined loss that balances region overlap (Soft Dice loss) with voxel-wise classification accuracy (Cross-Entropy loss), with equal weights ($\lambda_{\text{CE}} = \lambda_{\text{Dice}} = 1.0$). The Soft Dice loss excludes background class ($c \geq 1$) to focus on foreground structures. Both encoder and decoder are trained end-to-end with gradient clipping for stability. This approach follows the nnU-Net framework [28] with our time-conditioned encoder, enabling the network to leverage pretraining while adapting to dense prediction tasks.

Algorithm 2 Classification Finetuning

Require: Labeled dataset $\mathcal{D}_{\text{cls}} = \{(X^{(i)}, y^{(i)})\}_{i=1}^{N}$ where $y^{(i)} \in \{0, 1, \ldots, C-1\}$
Require: Pretrained encoder $f_{\theta_{\text{enc}}}$ from MDAE (Algorithm 1)
Require: Classification head $h_{\theta_{\text{cls}}}$: 2-layer MLP with ReLU activation and dropout
Require: Training: Learning rate $\eta \in [10^{-5}, 10^{-4}]$, batch size $B \in \{2, 4, 8\}$, gradient clip norm ν, max epochs $E \in \{100, 200, 300\}$
Require: Approach: approach $\in \{1, 2\}$ (see Approach C.1 and Approach C.2)
Ensure: Finetuned classification model $(f_{\theta_{\text{enc}}}, h_{\theta_{\text{cls}}})$
Initialize encoder from pretrained checkpoint: $\theta_{\text{enc}} \leftarrow \theta_{\text{MDAE}}$
Initialize classification head randomly: $\theta_{\text{cls}} \sim \text{Kaiming}(\cdot)$
for epoch $= 1, \ldots, E$ **do**
  for each batch $\{(X^{(i)}, y^{(i)})\}_{i=1}^{B} \sim \mathcal{D}_{\text{cls}}$ **do**
    for $i = 1, \ldots, B$ **do**
      // Forward pass: set time based on approach (see Approach C.1 and C.2)
      Set time parameter: $t^{(i)} = \begin{cases} \text{None} & \text{if approach} = 1 \\ 0 & \text{if approach} = 2 \end{cases}$
      Extract features: $z^{(i)} = \text{GlobalAvgPool}(f_{\theta_{\text{enc}}}(X^{(i)}, t^{(i)}))$
      Classify: $\hat{y}^{(i)} = h_{\theta_{\text{cls}}}(z^{(i)})$ ▷ Logits over C classes
      // Compute cross-entropy loss
      $\ell^{(i)} = -\sum_{c=0}^{C-1} \mathbf{1}_{y^{(i)}=c} \log \text{softmax}(\hat{y}^{(i)})_c$
    end for
    Compute batch loss: $\mathcal{L}_{\text{cls}} = \frac{1}{B} \sum_{i=1}^{B} \ell^{(i)}$
    Compute gradients: $g_{\text{enc}} \leftarrow \nabla_{\theta_{\text{enc}}} \mathcal{L}_{\text{cls}}$, $g_{\text{cls}} \leftarrow \nabla_{\theta_{\text{cls}}} \mathcal{L}_{\text{cls}}$
    Clip gradients: $g_{\text{enc}} \leftarrow \text{clip}(g_{\text{enc}}, \text{norm} = \nu)$, $g_{\text{cls}} \leftarrow \text{clip}(g_{\text{cls}}, \text{norm} = \nu)$
    Update parameters: $\theta_{\text{enc}} \leftarrow \theta_{\text{enc}} - \eta \cdot g_{\text{enc}}$, $\theta_{\text{cls}} \leftarrow \theta_{\text{cls}} - \eta \cdot g_{\text{cls}}$
  end for
end for
return Finetuned model $(f_{\theta_{\text{enc}}}, h_{\theta_{\text{cls}}})$

D. Mathematical Derivation of Limiting Behavior

This section provides derivations demonstrating how the MDAE combined loss (Eq. 5–7) reduces to the reference objectives MAE (Eq. 1) and DSM (Eq. 3) under specific limiting conditions. These derivations formally establish the interpolating property of MDAE between pure spatial masking and pure denoising.

D.1. Reduction to MAE: $\sigma_{\max} \to 0$ Limit

We show that MDAE reduces to pure masked autoencoding when diffusion noise vanishes. Starting from the MDAE combined loss (Eq. 5):

$$\mathcal{L}_{\text{MDAE}}(\theta) = \lambda_{\text{masked}} \cdot \mathcal{L}_{\text{masked}}(\theta) + \lambda_{\text{visible}} \cdot \mathcal{L}_{\text{visible}}(\theta)$$

where the masked loss (Eq. 6) and visible loss (Eq. 7) are:

$$\mathcal{L}_{\text{masked}}(\theta) = \mathbb{E}\Big[\frac{1}{|\Omega_M|} \cdot \|M \odot (g_\theta(\tilde{X}_t^M, t) - X_0)\|_2^2\Big]$$

$$\mathcal{L}_{\text{visible}}(\theta) = \mathbb{E}\Big[\frac{w(\sigma_t)}{|\Omega_V|} \cdot \|M_v \odot (g_\theta(\tilde{X}_t^M, t) - X_0)\|_2^2\Big]$$

with the doubly-corrupted input $\tilde{X}_t^M = M_v \odot \tilde{X}_t$ where $\tilde{X}_t = X_0 + \sigma_t Z$, $w(\sigma_t)$ is the noise-level weighting, and $\sigma_t = t \cdot \sigma_{\max}$.

As $\sigma_{\max} \to 0$, we have $\sigma_t = t \cdot \sigma_{\max} \to 0$ for all $t \in [0, 1]$, which gives:

$$\tilde{X}_t = X_0 + \sigma_t Z \to X_0$$
$$\tilde{X}_t^M = M_v \odot \tilde{X}_t \to M_v \odot X_0 = \tilde{X}^M$$

Algorithm 3 Segmentation Finetuning

Require: Labeled dataset $\mathcal{D}_{\text{seg}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{N}$ where $Y^{(i)} \in \{0, 1, \ldots, C-1\}^{D \times H \times W}$
Require: Pretrained encoder $f_{\theta_{\text{enc}}}$ from MDAE (Algorithm 1)
Require: Decoder $d_{\theta_{\text{dec}}}$ with skip connections
Require: Training: Learning rate $\eta \in [10^{-5}, 5 \times 10^{-4}]$, batch size $B \in \{1, 2, 3\}$, gradient clip norm ν, max epochs $E = 100$
Require: Loss weights: $\lambda_{\text{CE}} = 1.0$, $\lambda_{\text{Dice}} = 1.0$
Require: Approach: approach $\in \{1, 2\}$ (see Approach C.1 and Approach C.2)
Ensure: Finetuned segmentation model $(f_{\theta_{\text{enc}}}, d_{\theta_{\text{dec}}})$

Initialize encoder from pretrained checkpoint: $\theta_{\text{enc}} \leftarrow \theta_{\text{MDAE}}$
Initialize decoder randomly: $\theta_{\text{dec}} \sim \text{Kaiming}(\cdot)$
for epoch $= 1, \ldots, E$ **do**
  for each batch $\{(X^{(i)}, Y^{(i)})\}_{i=1}^{B} \sim \mathcal{D}_{\text{seg}}$ **do**
    for $i = 1, \ldots, B$ **do**
      // Forward pass: set time based on approach (see Approach C.1 and C.2)
      Set time parameter: $t^{(i)} = \begin{cases} \text{None} & \text{if approach} = 1 \\ 0 & \text{if approach} = 2 \end{cases}$
      Extract multi-scale features: $\{z_1^{(i)}, \ldots, z_L^{(i)}\} = \text{GlobalAvgPool}(f_{\theta_{\text{enc}}}(X^{(i)}, t^{(i)}))$
      Decode with skip connections: $\hat{Y}^{(i)} = d_{\theta_{\text{dec}}}(\{z_1^{(i)}, \ldots, z_L^{(i)}\})$ ▷ Logits over C classes
      // Compute combined Dice + Cross-Entropy loss
      Cross-Entropy: $\mathcal{L}_{\text{CE}}^{(i)} = -\frac{1}{DHW} \sum_{j=1}^{DHW} \log \text{softmax}(\hat{Y}_j^{(i)})_{Y_j^{(i)}}$
      Soft Dice: $\mathcal{L}_{\text{Dice}}^{(i)} = 1 - \frac{2 \sum_{c=1}^{C-1} \sum_j \text{softmax}(\hat{Y}_j^{(i)})_c \cdot \mathbf{1}_{Y_j^{(i)}=c}}{\sum_{c=1}^{C-1} \left(\sum_j \text{softmax}(\hat{Y}_j^{(i)})_c + \sum_j \mathbf{1}_{Y_j^{(i)}=c} \right)}$
      Combined loss: $\ell^{(i)} = \lambda_{\text{CE}} \cdot \mathcal{L}_{\text{CE}}^{(i)} + \lambda_{\text{Dice}} \cdot \mathcal{L}_{\text{Dice}}^{(i)}$
    end for
    Compute batch loss: $\mathcal{L}_{\text{seg}} = \frac{1}{B} \sum_{i=1}^{B} \ell^{(i)}$
    Compute gradients: $g_{\text{enc}} \leftarrow \nabla_{\theta_{\text{enc}}} \mathcal{L}_{\text{seg}}$, $g_{\text{dec}} \leftarrow \nabla_{\theta_{\text{dec}}} \mathcal{L}_{\text{seg}}$
    Clip gradients: $g_{\text{enc}} \leftarrow \text{clip}(g_{\text{enc}}, \text{norm} = \nu)$, $g_{\text{dec}} \leftarrow \text{clip}(g_{\text{dec}}, \text{norm} = \nu)$
    Update parameters: $\theta_{\text{enc}} \leftarrow \theta_{\text{enc}} - \eta \cdot g_{\text{enc}}$, $\theta_{\text{dec}} \leftarrow \theta_{\text{dec}} - \eta \cdot g_{\text{dec}}$
  end for
end for
return Finetuned model $(f_{\theta_{\text{enc}}}, d_{\theta_{\text{dec}}})$

On visible regions where $M_v = 1$, the network receives clean input values. The optimal predictor satisfies $g_\theta(\tilde{X}^M)|_{\Omega_V} = X_0|_{\Omega_V}$, yielding:

$$\|M_v \odot (g_\theta(\tilde{X}^M) - X_0)\|_2^2 \to 0$$

While $w(\sigma_t)$ may grow as $\sigma_t \to 0$, the product $w(\sigma_t) \cdot 0 = 0$ ensures $\lim_{\sigma_{\max} \to 0} \mathcal{L}_{\text{visible}}(\theta) = 0$.

With no noise, the masked loss becomes:

$$\lim_{\sigma_{\max} \to 0} \mathcal{L}_{\text{masked}}(\theta) = \mathbb{E}_{X_0, M}\left[\frac{1}{|\Omega_M|} \cdot \|M \odot (g_\theta(\tilde{X}^M) - X_0)\|_2^2 \right] = \mathcal{L}_{\text{MAE}}(\theta)$$

Therefore, the combined loss reduces to:

$$\lim_{\sigma_{\max} \to 0} \mathcal{L}_{\text{MDAE}}(\theta) = \lambda_{\text{masked}} \cdot \mathcal{L}_{\text{MAE}}(\theta) + \lambda_{\text{visible}} \cdot 0 = \lambda_{\text{masked}} \cdot \mathcal{L}_{\text{MAE}}(\theta)$$

With $\lambda_{\text{masked}} = 1.0$, this recovers $\mathcal{L}_{\text{MAE}}$ as $\sigma_{\max} \to 0$. Without diffusion noise, the visible regions provide clean information, leaving only masked regions to reconstruct from context.

D.2. Reduction to DSM: $p_{\text{mask}} \to 0$ Limit

We show that MDAE reduces to pure denoising score matching when spatial masking vanishes. As $p_{\text{mask}} \to 0$, the probability of masking any patch approaches zero:

$$M \to \mathbf{0}, \quad M_v = \mathbf{1} - M \to \mathbf{1}, \quad \tilde{X}_t^M = M_v \odot \tilde{X}_t \to \tilde{X}_t = X_0 + \sigma_t Z$$

The sets of masked and visible voxels evolve as $|\Omega_M| = \sum_j M_j \to 0$ and $|\Omega_V| = \sum_j M_{v,j} \to DHW \equiv N_{\text{total}}$. Since each patch is masked independently with probability p_{mask}, we have $\mathbb{P}(|\Omega_M| > 0) \to 0$ as $p_{\text{mask}} \to 0$. On the event $\{|\Omega_M| = 0\}$, the masked loss is zero (no masked voxels to reconstruct). Therefore, the expected masked loss vanishes:

$$\mathcal{L}_{\text{masked}}(\theta) = \mathbb{E}\Big[\frac{1}{|\Omega_M|} \cdot \|M \odot (g_\theta(\tilde{X}_t^M, t) - X_0)\|_2^2\Big] \to 0$$

With $M_v \to \mathbf{1}$ and $|\Omega_V| \to N_{\text{total}}$, the visible loss becomes:

$$\begin{aligned}\lim_{p_{\text{mask}} \to 0} \mathcal{L}_{\text{visible}}(\theta) &= \mathbb{E}_{X_0,t,Z}\Big[\frac{w(\sigma_t)}{N_{\text{total}}} \cdot \|\mathbf{1} \odot (g_\theta(\tilde{X}_t, t) - X_0)\|_2^2\Big] \\ &= \frac{1}{N_{\text{total}}} \cdot \mathbb{E}_{X_0,t,Z}\Big[w(\sigma_t) \cdot \|g_\theta(\tilde{X}_t, t) - X_0\|_2^2\Big] \propto \mathcal{L}_{\text{DSM}}(\theta)\end{aligned}$$

which recovers the weighted DSM objective (Eq. 3) up to the constant per-voxel normalization factor $\frac{1}{N_{\text{total}}}$ and the noise-level weighting $w(\sigma_t)$ [33, 34]. Since N_{total} is a fixed constant, minimizing $\mathcal{L}_{\text{visible}}$ is equivalent to minimizing the weighted DSM loss.

Therefore, the combined loss reduces to:

$$\lim_{p_{\text{mask}} \to 0} \mathcal{L}_{\text{MDAE}}(\theta) = \lambda_{\text{masked}} \cdot 0 + \lambda_{\text{visible}} \cdot \frac{1}{N_{\text{total}}} \cdot \mathcal{L}_{\text{DSM}}^w(\theta) \propto \mathcal{L}_{\text{DSM}}(\theta)$$

With $\lambda_{\text{visible}} = 1.0$, this recovers $\mathcal{L}_{\text{DSM}}$ up to per-voxel normalization and noise-level weighting as $p_{\text{mask}} \to 0$. Without spatial masking, all voxels are visible but noise-corrupted, reducing the task to pure denoising across the entire volume.

The above derivations show that MDAE interpolates between two established self-supervised objectives:

$$\mathcal{L}_{\text{MDAE}}(\theta) \to \begin{cases} \mathcal{L}_{\text{MAE}}(\theta) & \text{as } \sigma_{\max} \to 0 \\ \mathcal{L}_{\text{DSM}}(\theta) & \text{as } p_{\text{mask}} \to 0 \end{cases}$$

When both corruption parameters are active ($p_{\text{mask}}, \sigma_{\max} > 0$), the two corruptions operate orthogonally: spatial masking removes information on Ω_M while diffusion noise corrupts intensity values on Ω_V. This allows the combined objective to simultaneously encourage learning from both spatial context reconstruction and intensity denoising, providing a mathematical foundation for dual corruption in self-supervised learning.

E. Detailed Segmentation Results

This section provides comprehensive per-region segmentation performance results for multi-modality brain tumor segmentation tasks. Tables 7 and 8 present detailed breakdowns of Dice coefficient and Normalized Surface Distance (NSD) metrics for each tumor subregion, complementing the overall results reported in Table 3 of the main text.

F. Detailed Multi-Modality Classification Results

This section provides comprehensive multi-modality classification results including all evaluation metrics (AUROC, Average Precision, F1 score, and balanced accuracy) for both BraTS18 tumor grading and UCSF-PDGM IDH classification tasks. Table 3 in the main text reports only AUROC for conciseness. The results demonstrate MDAE's consistent advantages across all metrics, with particularly strong performance on AUROC (92.1% BraTS18, 93.0% UCSF-PDGM) and Average Precision (89.5% BraTS18, 87.8% UCSF-PDGM).

Table 7. **Brain tumor segmentation on UCSF-PDGM test set**. Results reported as Dice coefficient and Normalized Surface Distance (NSD) for three tumor regions: NCR/NET, ED, and ET. Bold indicates best, underline indicates second-best.

	Overall		NET		ED		ET	
Method	Dice↑	NSD↑	Dice↑	NSD↑	Dice↑	NSD↑	Dice↑	NSD↑
MAE	<u>84.8</u>	<u>87.7</u>	<u>82.7</u>	<u>87.1</u>	86.1	86.8	<u>85.6</u>	<u>89.3</u>
MG	84.3	87.3	80.9	85.5	86.0	86.8	**86.1**	**89.5**
S3D	84.2	87.1	81.9	86.1	86.1	86.8	84.8	88.3
SimCLR	84.1	87.1	81.1	85.7	86.1	86.7	85.2	88.8
VoCo	84.1	87.2	81.3	86.3	**86.4**	**87.1**	84.6	88.1
SwinUNETR	83.7	86.5	80.6	84.8	85.8	86.4	84.8	88.2
VF	83.6	86.5	80.0	84.2	86.1	<u>86.9</u>	84.9	88.5
MDAE	**85.2**	**88.1**	**83.4**	**88.0**	<u>86.2</u>	86.7	**86.1**	**89.5**

Table 8. **Brain tumor segmentation on BraTS18 test set**. Results reported as Dice and NSD for WT (Whole Tumor), TC (Tumor Core), and ET (Enhancing Tumor). Bold indicates best, underline indicates second-best.

	Overall		WT		TC		ET	
Method	Dice↑	NSD↑	Dice↑	NSD↑	Dice↑	NSD↑	Dice↑	NSD↑
VoCo	<u>80.9</u>	74.7	<u>90.7</u>	<u>79.5</u>	**84.9**	71.4	67.1	73.3
SwinUNETR	80.7	<u>75.1</u>	**90.8**	**79.8**	84.5	<u>72.3</u>	66.9	73.1
MG	80.7	<u>75.1</u>	90.4	79.4	84.4	**72.4**	67.4	73.6
VF	80.7	74.7	89.6	78.4	83.6	70.6	<u>68.8</u>	<u>75.1</u>
MAE	80.6	75.0	90.2	**79.8**	83.4	71.0	68.1	74.1
S3D	80.2	74.3	89.5	78.3	84.0	71.8	67.3	72.9
SimCLR	80.1	74.1	89.5	78.4	84.1	71.0	66.7	72.8
MDAE	**81.4**	**75.3**	90.4	79.2	<u>84.6</u>	71.0	**69.2**	**75.8**

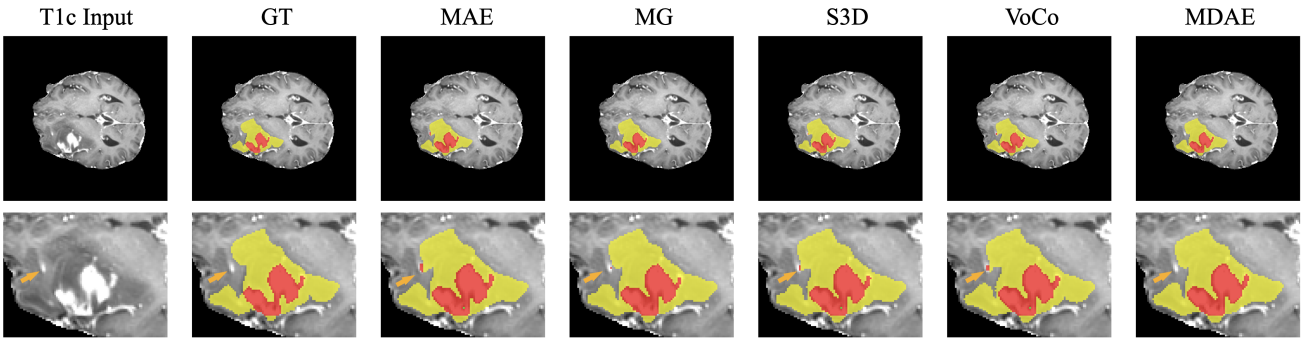

Figure 4. **Qualitative segmentation results on UCSF-PDGM.** Visual comparison across methods on a representative case. Top row: full tumor extent; bottom row: zoomed tumor core. Yellow indicates ED (peritumoral edema), red indicates ET (enhancing tumor). Orange arrows highlight false positives present in baseline methods but avoided by MDAE.

Table 9. **Multi-sequence brain tumor classification performance.** Models process each sequence independently through the encoder, aggregate embeddings via mean pooling, and apply a classification head. BraTS18: Low-grade vs high-grade glioma with 4 sequences (T1, T1-Gd, T2, FLAIR). UCSF-PDGM: IDH mutation status with 6 sequences. Best results in **bold**, second-best underlined.

	BraTS18				**UCSF-PDGM**			
Method	AUROC	AP	F1	Bal.Acc	AUROC	AP	F1	Bal.Acc
SSL Baselines								
MAE	90.1	88.4	75.5	72.1	87.8	86.6	78.6	74.9
SimCLR	78.7	75.0	68.4	72.4	80.3	75.9	76.2	73.6
VoCo	84.8	81.7	80.8	82.9	91.6	86.9	**83.0**	79.7
MG	84.9	82.5	63.8	68.8	92.3	85.7	81.3	85.8
VF	83.7	85.1	66.7	43.0	90.2	86.1	80.7	81.2
S3D	89.4	83.6	79.2	78.6	81.9	77.5	80.0	79.5
SwinUNETR	82.7	75.2	72.4	76.9	86.5	85.1	78.4	76.0
Foundation Models								
BrainIAC	81.4	75.4	73.6	81.2	76.8	69.7	62.0	69.9
BrainMVP	85.2	83.6	76.3	75.2	89.2	84.0	56.5	56.6
MDAE	**92.1**	**89.5**	**84.5**	**85.2**	**93.0**	**87.8**	80.1	**88.0**