VAYUN GOEL, CHETAN KUKREJA, ACHAL JAIN, SUKVANSH JAIN

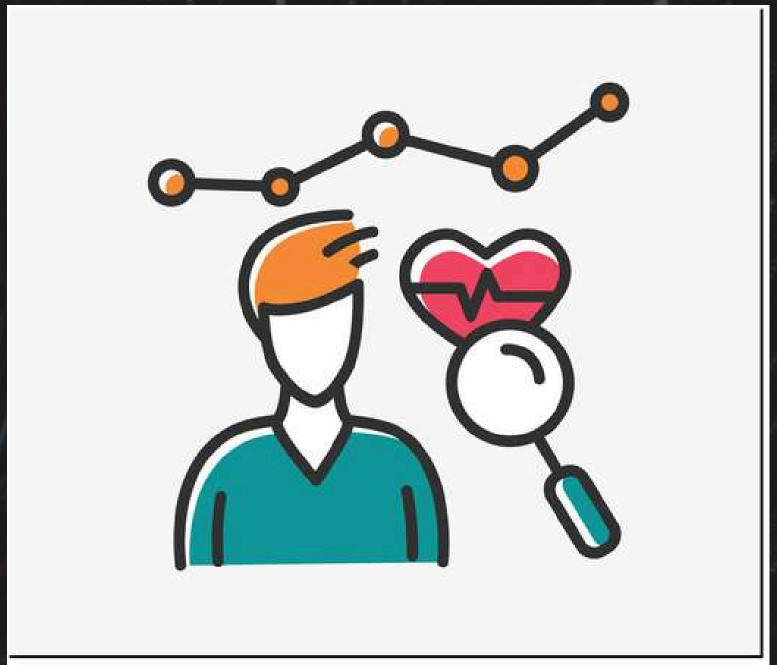
MEDICAL PROGNOSIS USING TD LEARNING IN DIFFUSION MODELS

EE-608 DIGITAL IMAGE PROCESSING

MOTIVATION

MEDICAL BENEFITS

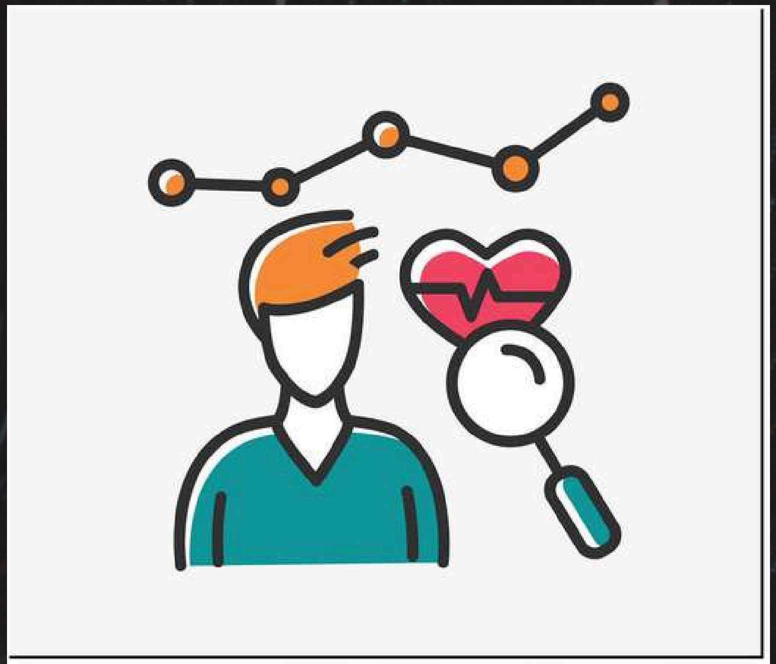
- **Guides Treatment Decisions:** Prognosis helps determine how aggressive the treatment should be
- **Early prediction** based on historical data for timely intervention
- Avoids over treatment and under treatment
- Personalized Medicine



MOTIVATION

MEDICAL BENEFITS

- **Guides Treatment Decisions:** Prognosis helps determine how aggressive the treatment should be
- **Early prediction** based on historical data for timely intervention
- Avoids over treatment and under treatment
- Personalized Medicine



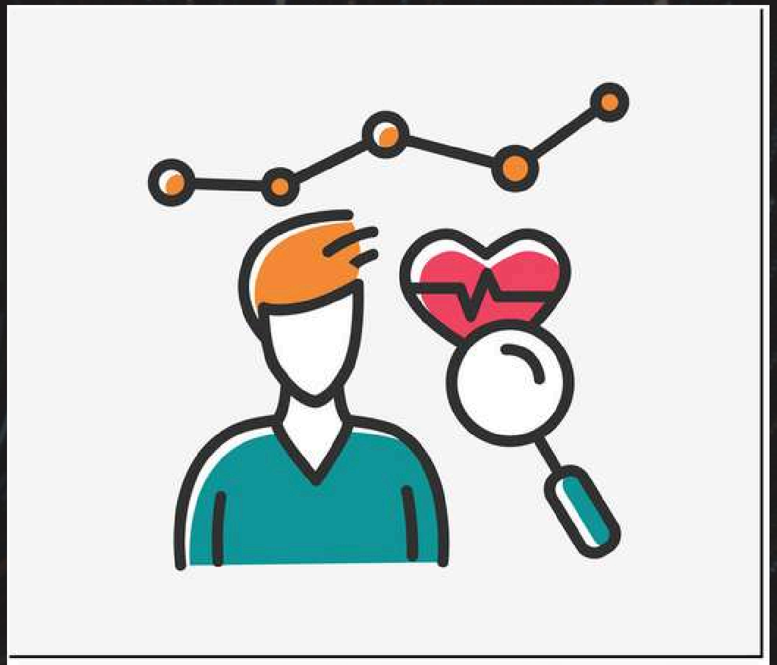
CURRENT FLAWS

- Limited Generalization
- Static Predictions
- Data Integration Challenges
- Lack of Interpretability
- Bias and Overfitting
- Poor Incorporation of Treatment Response

MOTIVATION

MEDICAL BENEFITS

- **Guides Treatment Decisions:** Prognosis helps determine how aggressive the treatment should be
- **Early prediction** based on historical data for timely intervention
- Avoids over treatment and under treatment
- Personalized Medicine

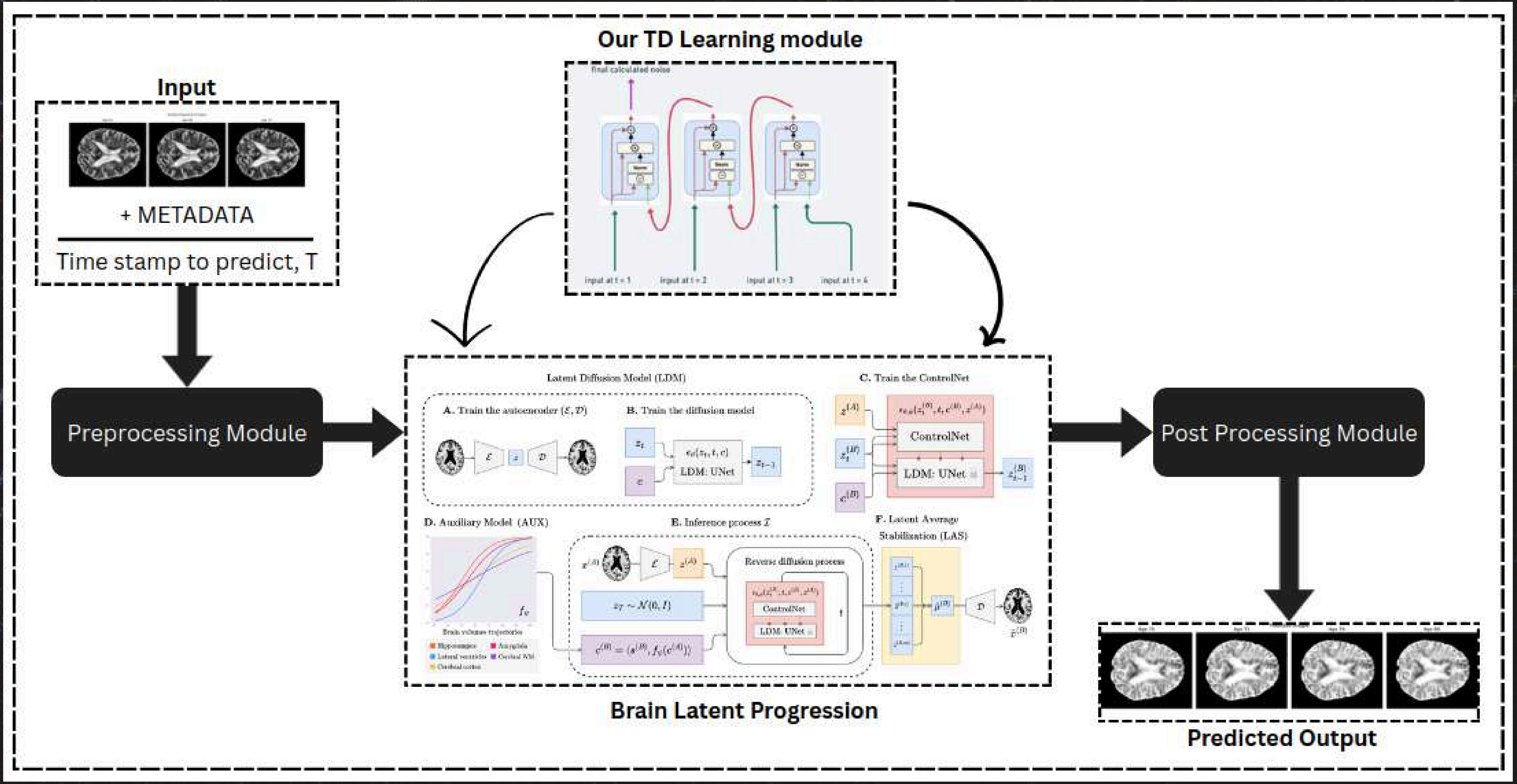


CURRENT FLAWS

we deal with

- Limited Generalization
- Static Predictions
- Data Integration Challenges
- Lack of Interpretability
- Bias and Overfitting
- Poor Incorporation of Treatment Response

INFERENCE PIPELINE



Input

+ METADATA

Time stamp to predict, T

Preprocessing Module

Our TD Learning module

Brain Latent Progression

Post Processing Module

Predicted Output

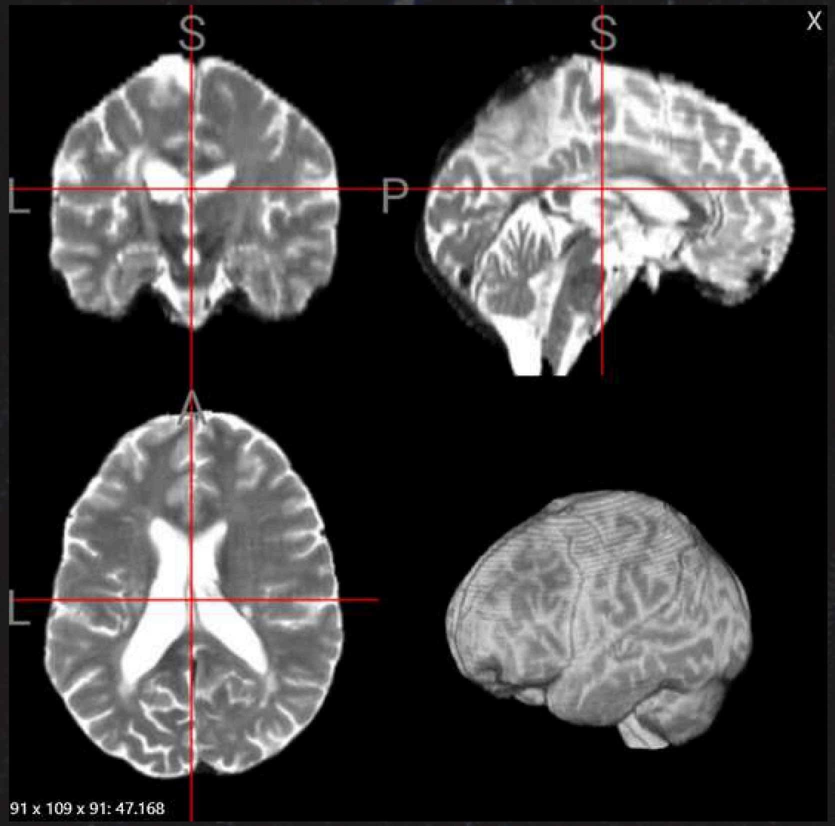
DATASET USED

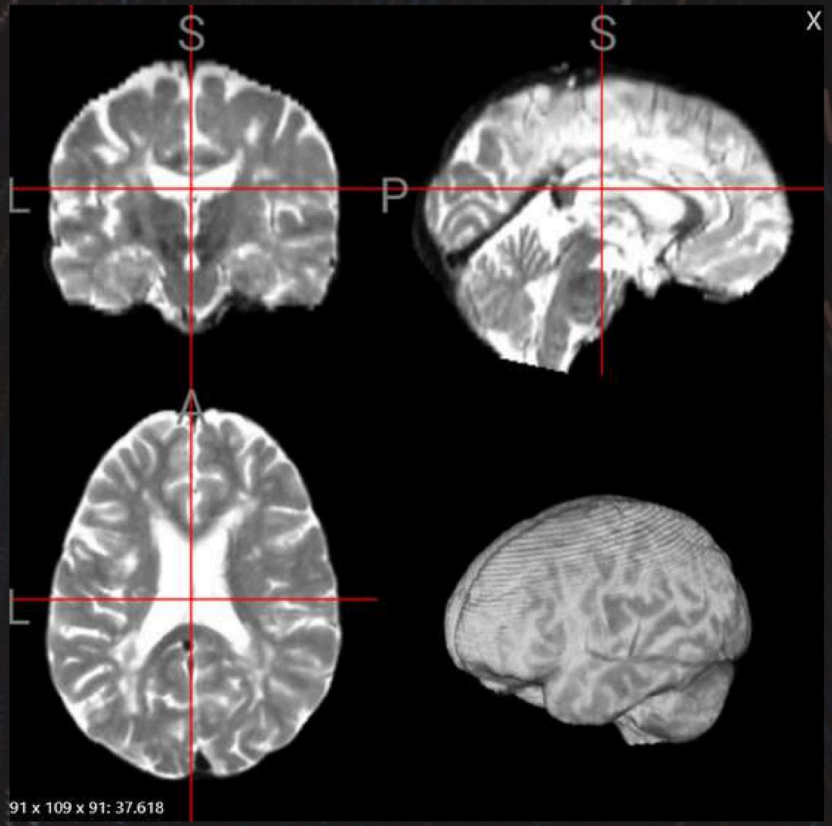
MSLSC

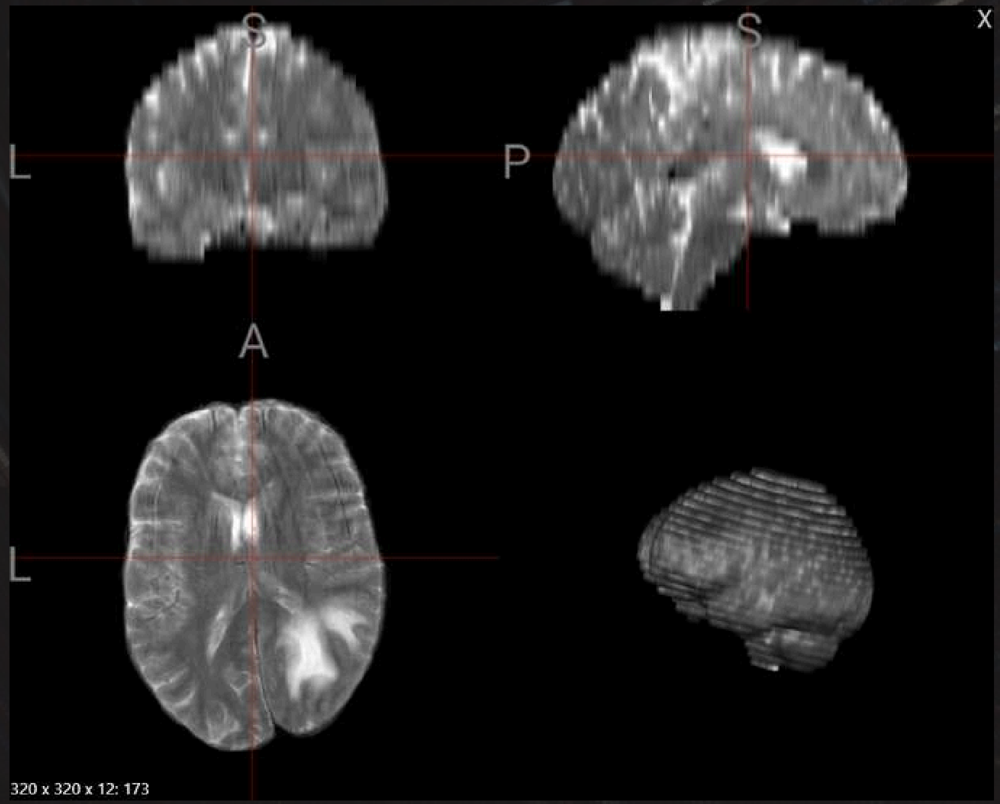
1. 82 scans from 19 subjects... across ~yearly time points.
2. Features preprocessed T1-w, T2-w, PD-w, and FLAIR images with manual lesion delineations by two experts.
3. Five subjects with mean of 4.4 time points and test data of fourteen subjects with a mean of 4.4 time points.

LUMIERE

1. 91 glioma patients with 638 MRI time points over pre- and post-operative
2. Multiparametric MRI: Includes T1W, T1CE, T2W and FLAIR (only 599 have all 4, though)
3. Longitudinal multiparametric MRIs co-registered, skull-stripped, segmented; organized per patient/week with NIfTI volumes.







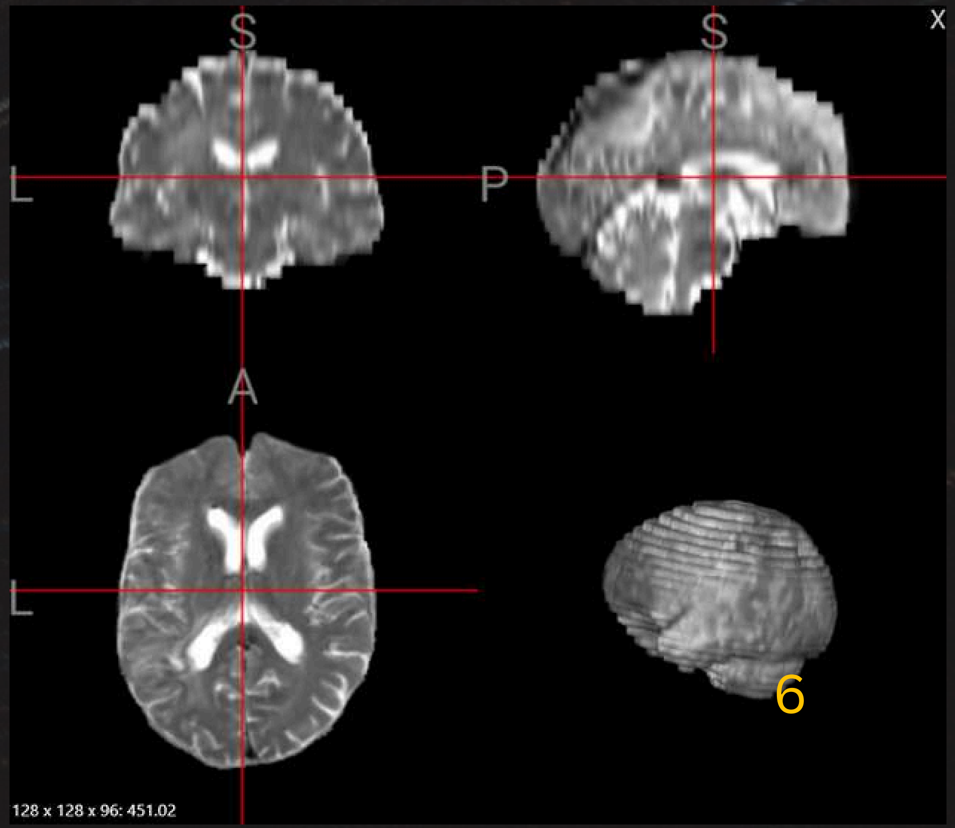
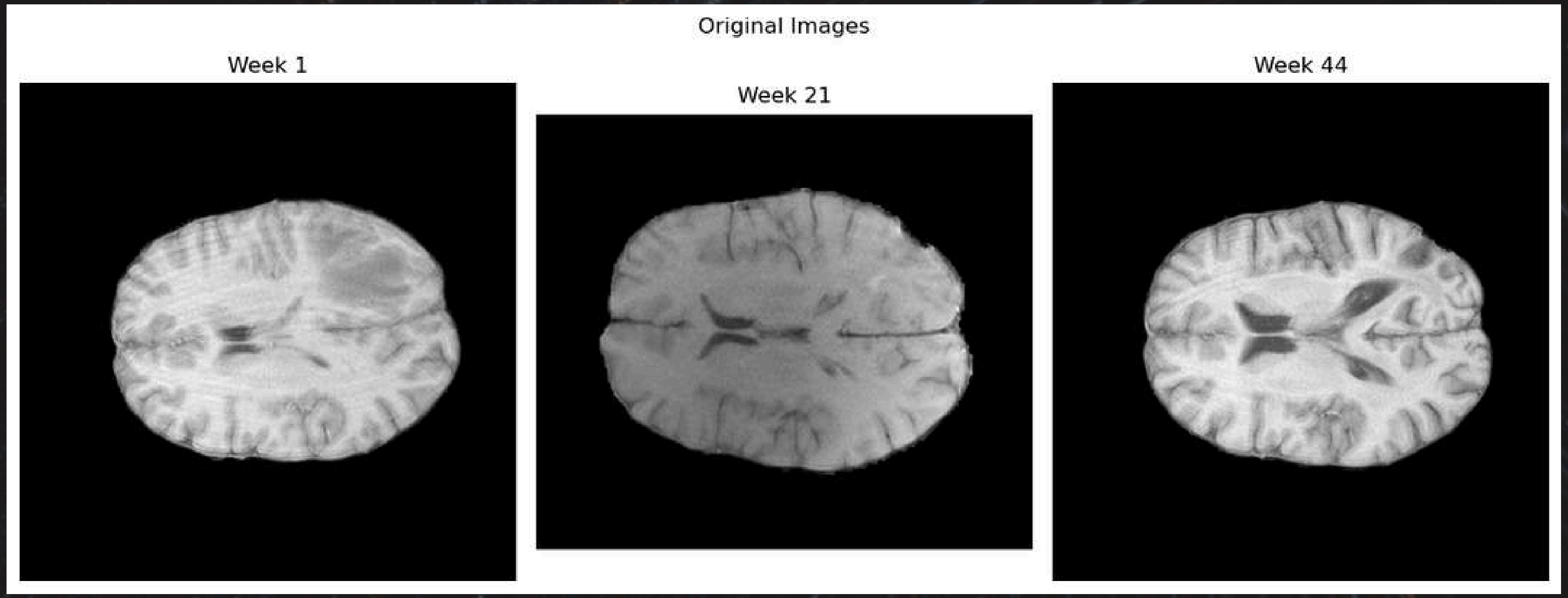
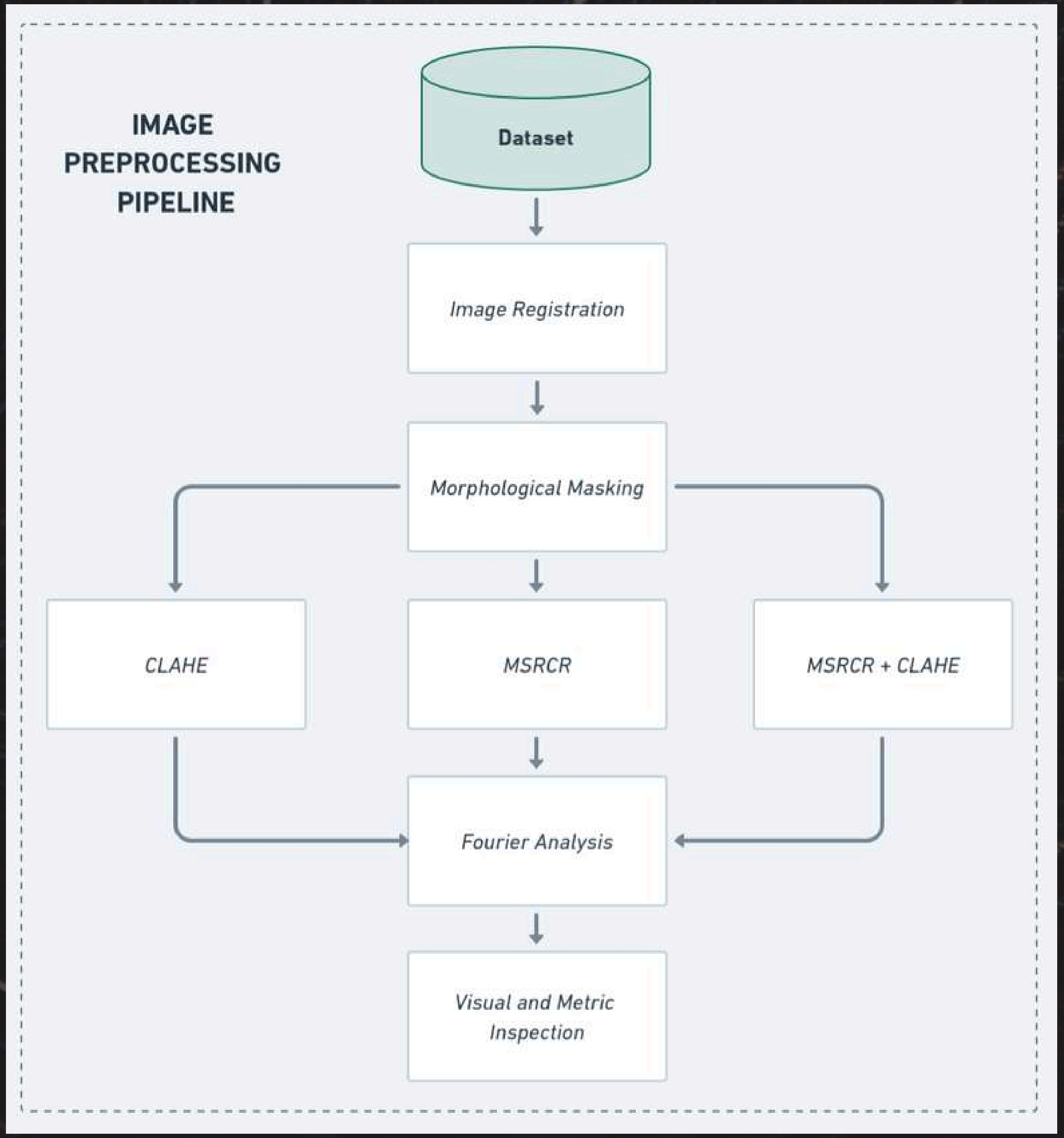


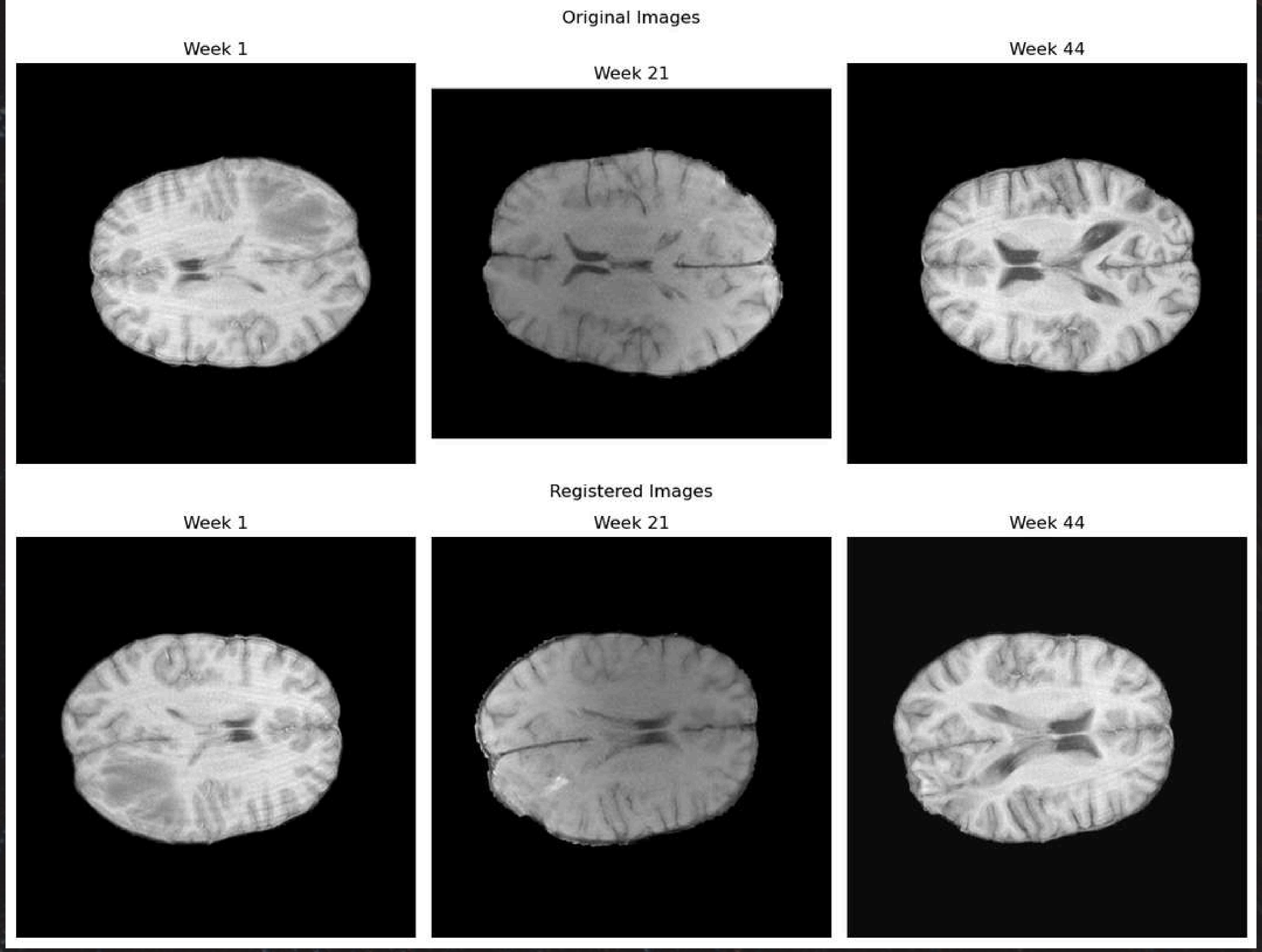
IMAGE PREPROCESSING

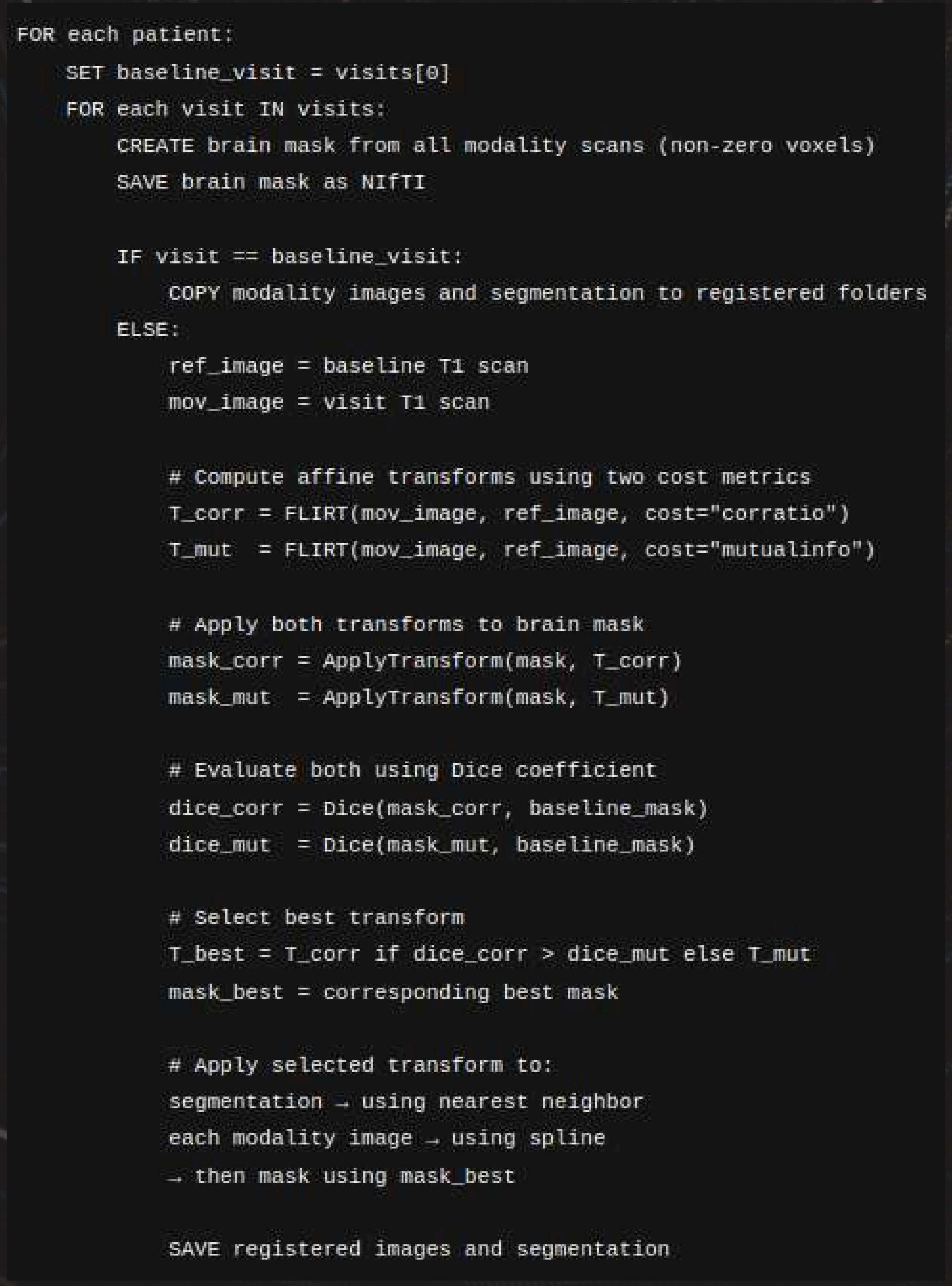
PART 1: IMAGE ENHANCEMENT





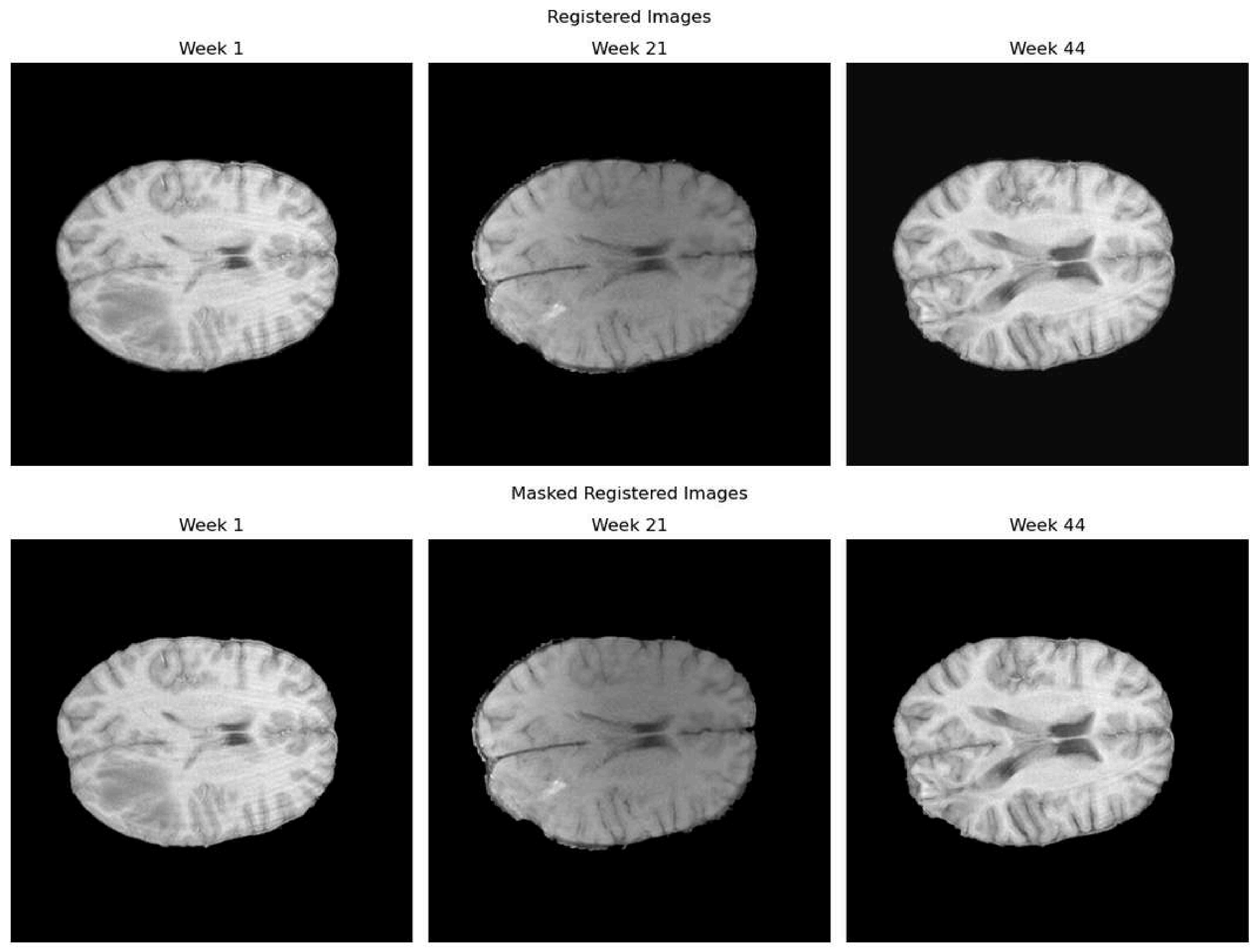
STEP 1: Registration





```
FOR each patient:
    SET baseline_visit = visits[0]
    FOR each visit IN visits:
        CREATE brain mask from all modality scans (non-zero voxels)
        SAVE brain mask as NIfTI

        IF visit == baseline_visit:
            COPY modality images and segmentation to registered folders
        ELSE:
            ref_image = baseline T1 scan
            mov_image = visit T1 scan

            # Compute affine transforms using two cost metrics
            T_corr = FLIRT(mov_image, ref_image, cost="corratio")
            T_mut  = FLIRT(mov_image, ref_image, cost="mutualinfo")

            # Apply both transforms to brain mask
            mask_corr = ApplyTransform(mask, T_corr)
            mask_mut  = ApplyTransform(mask, T_mut)

            # Evaluate both using Dice coefficient
            dice_corr = Dice(mask_corr, baseline_mask)
            dice_mut  = Dice(mask_mut, baseline_mask)

            # Select best transform
            T_best = T_corr if dice_corr > dice_mut else T_mut
            mask_best = corresponding best mask

            # Apply selected transform to:
            segmentation → using nearest neighbor
            each modality image → using spline
            → then mask using mask_best

            SAVE registered images and segmentation
```

STEP 2: Masking



Registered Images

Week 1

Week 21

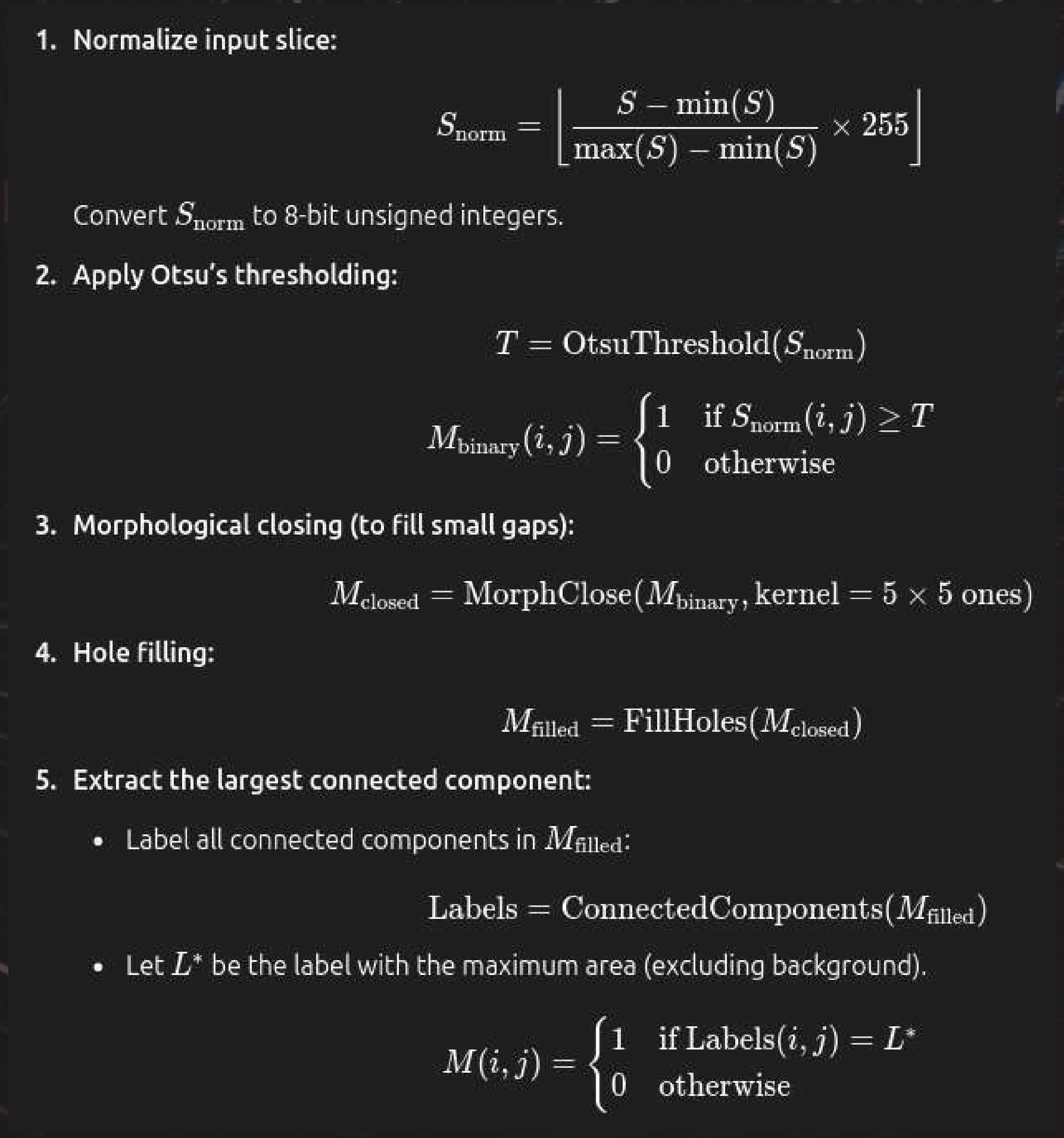
Week 44

Masked Registered Images

Week 1

Week 21

Week 44



1. **Normalize input slice:**

$$S_{\text{norm}} = \left\lfloor \frac{S - \min(S)}{\max(S) - \min(S)} \times 255 \right\rfloor$$

Convert S_{norm} to 8-bit unsigned integers.

2. **Apply Otsu's thresholding:**

$$T = \text{OtsuThreshold}(S_{\text{norm}})$$

$$M_{\text{binary}}(i,j) = \begin{cases} 1 & \text{if } S_{\text{norm}}(i,j) \geq T \\ 0 & \text{otherwise} \end{cases}$$

3. **Morphological closing (to fill small gaps):**

$$M_{\text{closed}} = \text{MorphClose}(M_{\text{binary}}, \text{kernel} = 5 \times 5 \text{ ones})$$

4. **Hole filling:**

$$M_{\text{filled}} = \text{FillHoles}(M_{\text{closed}})$$

5. **Extract the largest connected component:**

- Label all connected components in M_{filled}:

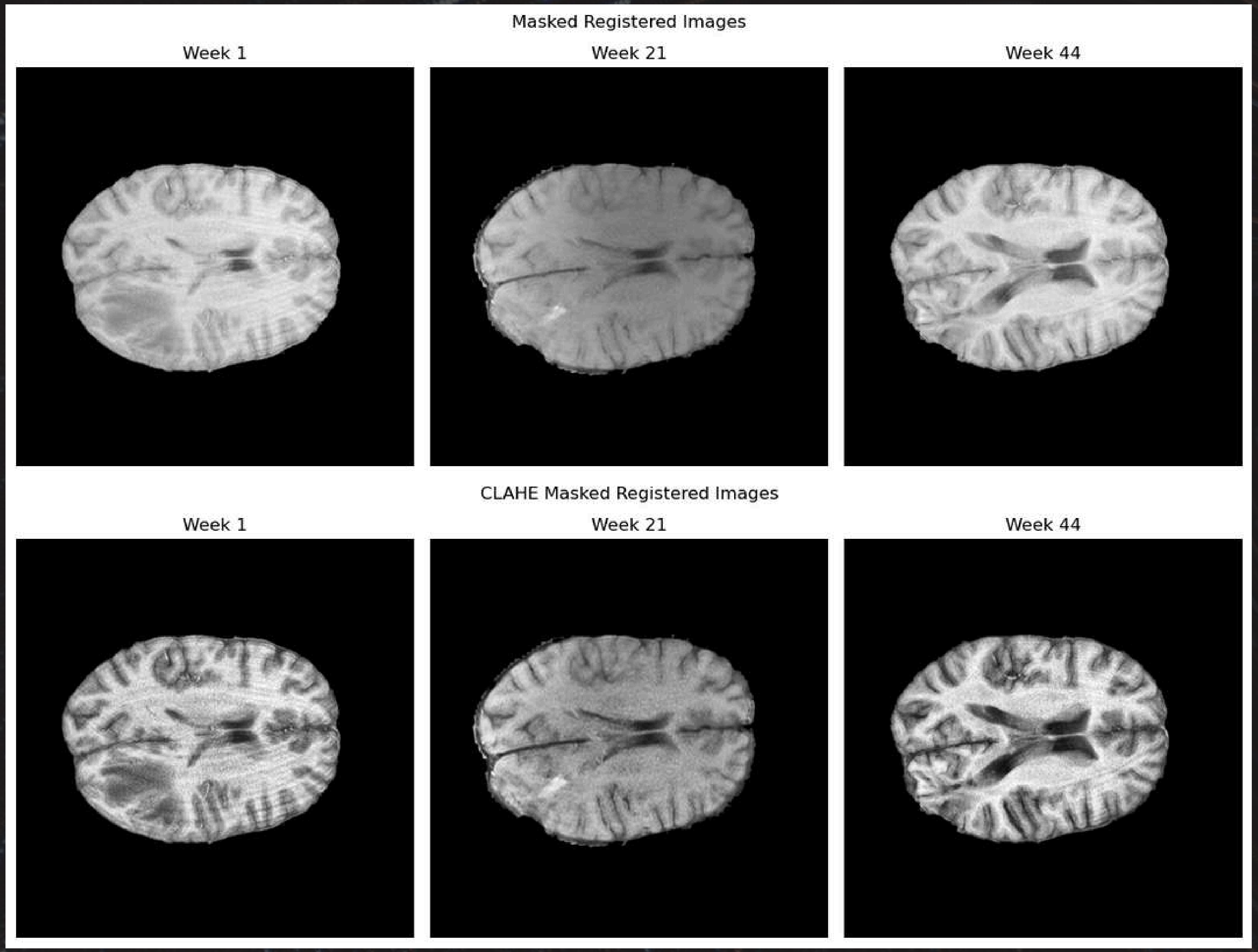
$$\text{Labels} = \text{ConnectedComponents}(M_{\text{filled}})$$

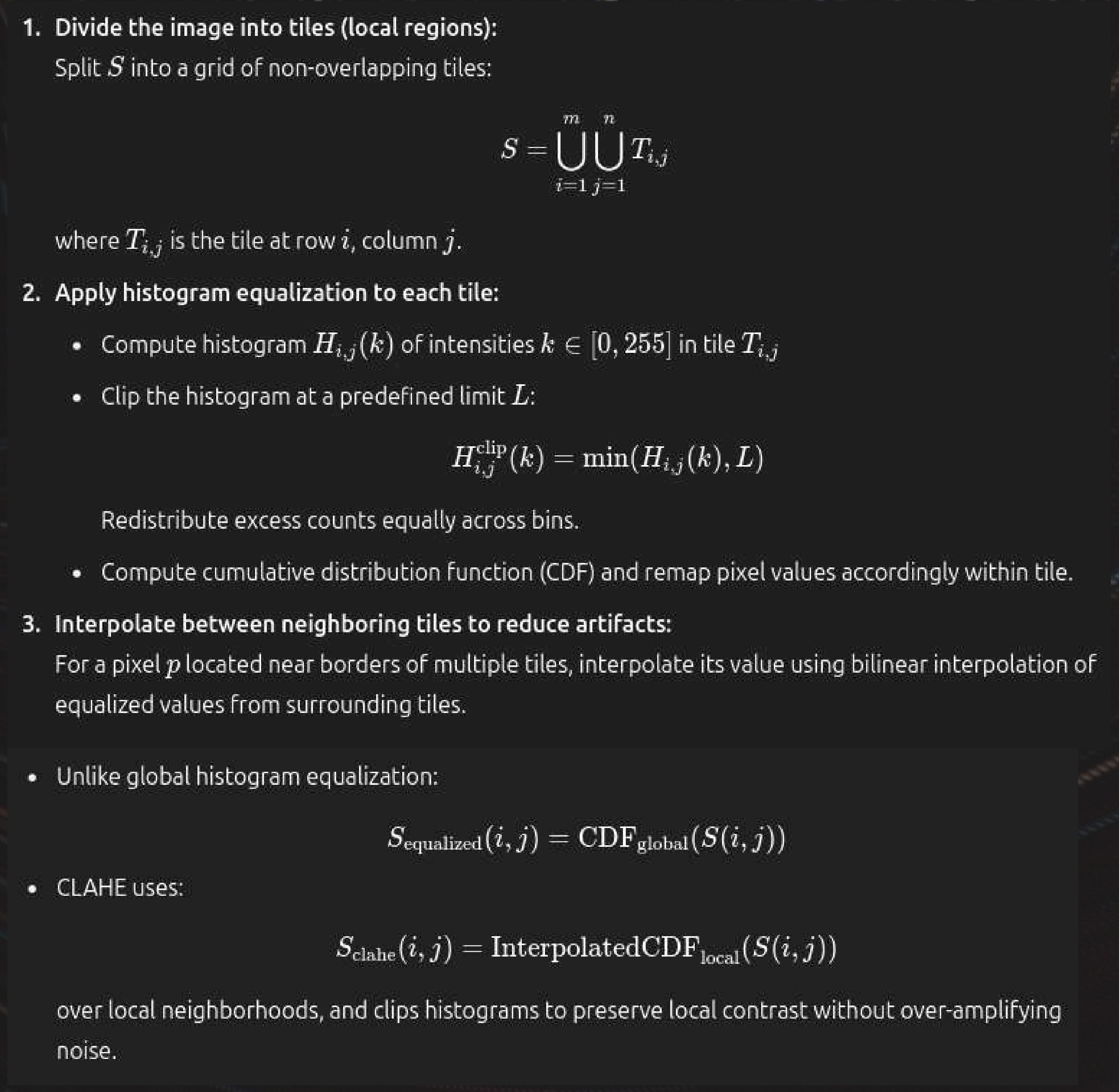
- Let L^* be the label with the maximum area (excluding background).

$$M(i,j) = \begin{cases} 1 & \text{if Labels}(i,j) = L^* \\ 0 & \text{otherwise} \end{cases}$$

STEP 3(a): CLAHE

Contrast Limited Adaptive Histogram Equalization





1. **Divide the image into tiles (local regions):**
   Split S into a grid of non-overlapping tiles:

$$S = \bigcup_{i=1}^{m} \bigcup_{j=1}^{n} T_{i,j}$$

   where $T_{i,j}$ is the tile at row i, column j.

2. **Apply histogram equalization to each tile:**
   - Compute histogram $H_{i,j}(k)$ of intensities $k \in [0, 255]$ in tile $T_{i,j}$
   - Clip the histogram at a predefined limit L:

$$H_{i,j}^{\text{clip}}(k) = \min(H_{i,j}(k), L)$$

     Redistribute excess counts equally across bins.
   - Compute cumulative distribution function (CDF) and remap pixel values accordingly within tile.

3. **Interpolate between neighboring tiles to reduce artifacts:**
   For a pixel p located near borders of multiple tiles, interpolate its value using bilinear interpolation of equalized values from surrounding tiles.

- Unlike global histogram equalization:

$$S_{\text{equalized}}(i, j) = \text{CDF}_{\text{global}}(S(i, j))$$

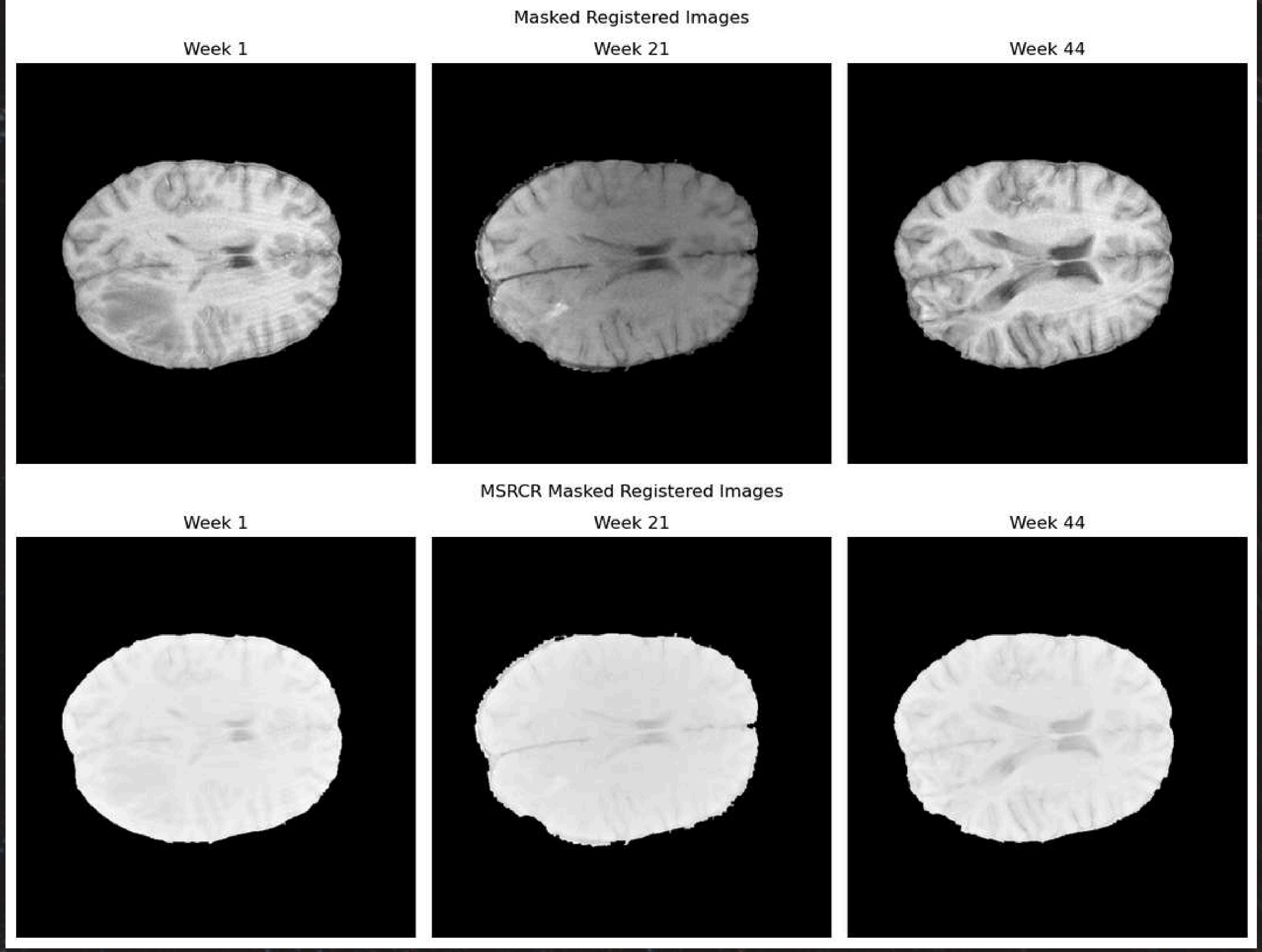
- CLAHE uses:

$$S_{\text{clahe}}(i, j) = \text{InterpolatedCDF}_{\text{local}}(S(i, j))$$

  over local neighborhoods, and clips histograms to preserve local contrast without over-amplifying noise.

STEP 3(b): MSRCR

Multi-scale retinex with color restoration

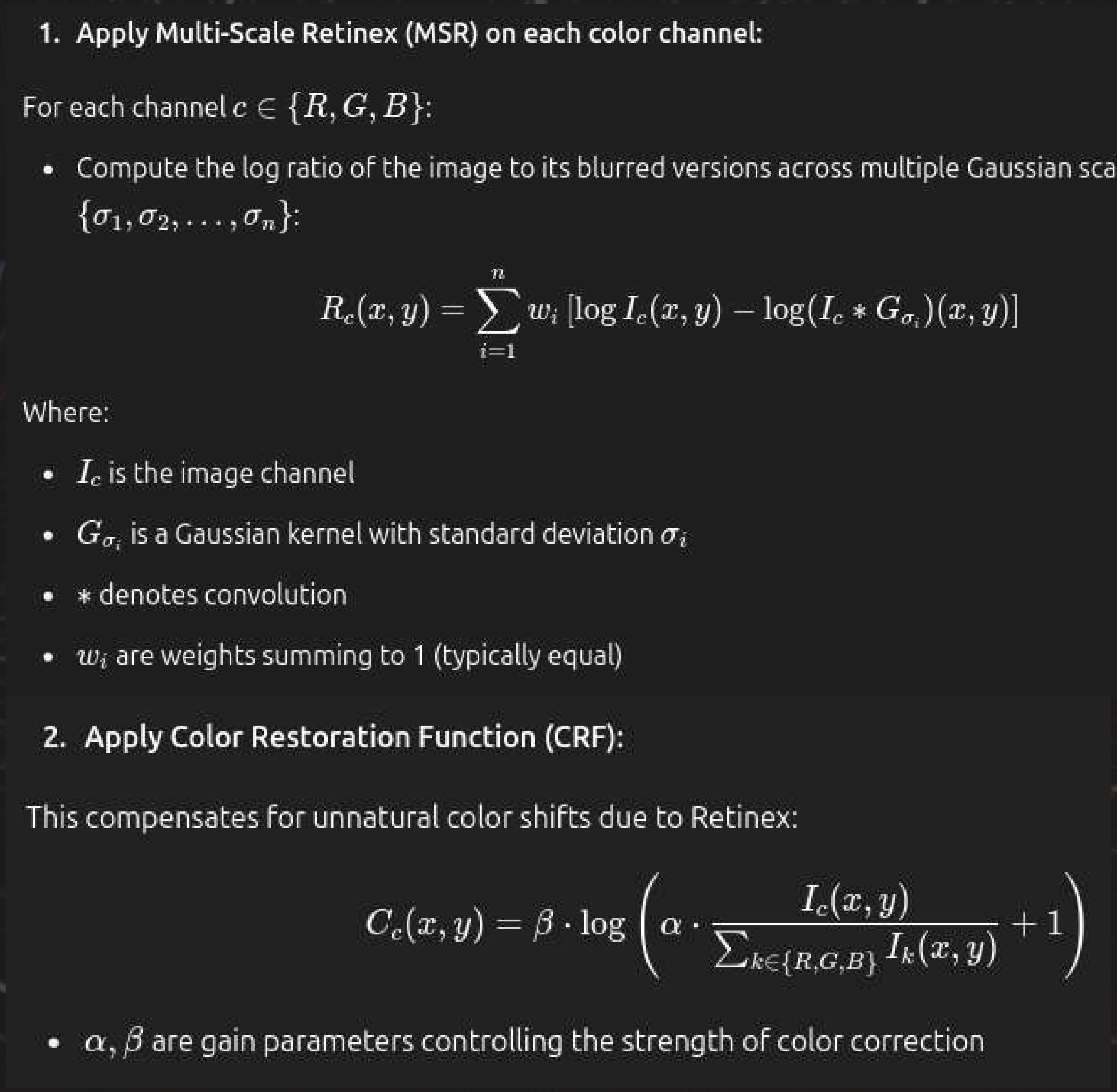


Masked Registered Images

Week 1 Week 21 Week 44

MSRCR Masked Registered Images

Week 1 Week 21 Week 44



1. **Apply Multi-Scale Retinex (MSR) on each color channel:**

For each channel $c \in \{R, G, B\}$:

- Compute the log ratio of the image to its blurred versions across multiple Gaussian scales $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$:

$$R_c(x, y) = \sum_{i=1}^{n} w_i \left[\log I_c(x, y) - \log(I_c * G_{\sigma_i})(x, y)\right]$$

Where:

- I_c is the image channel
- G_{σ_i} is a Gaussian kernel with standard deviation σ_i
- $*$ denotes convolution
- w_i are weights summing to 1 (typically equal)

2. **Apply Color Restoration Function (CRF):**

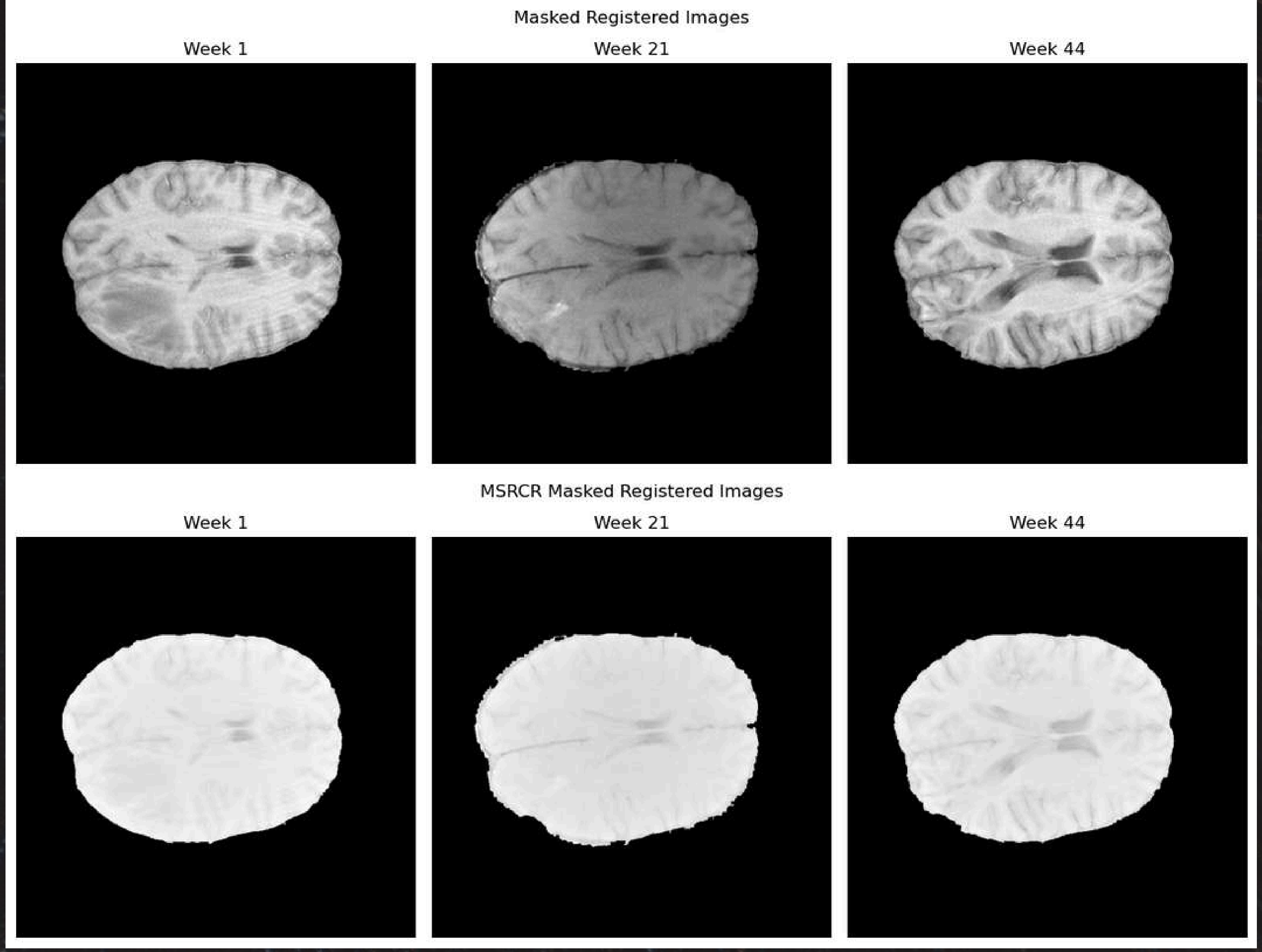
This compensates for unnatural color shifts due to Retinex:

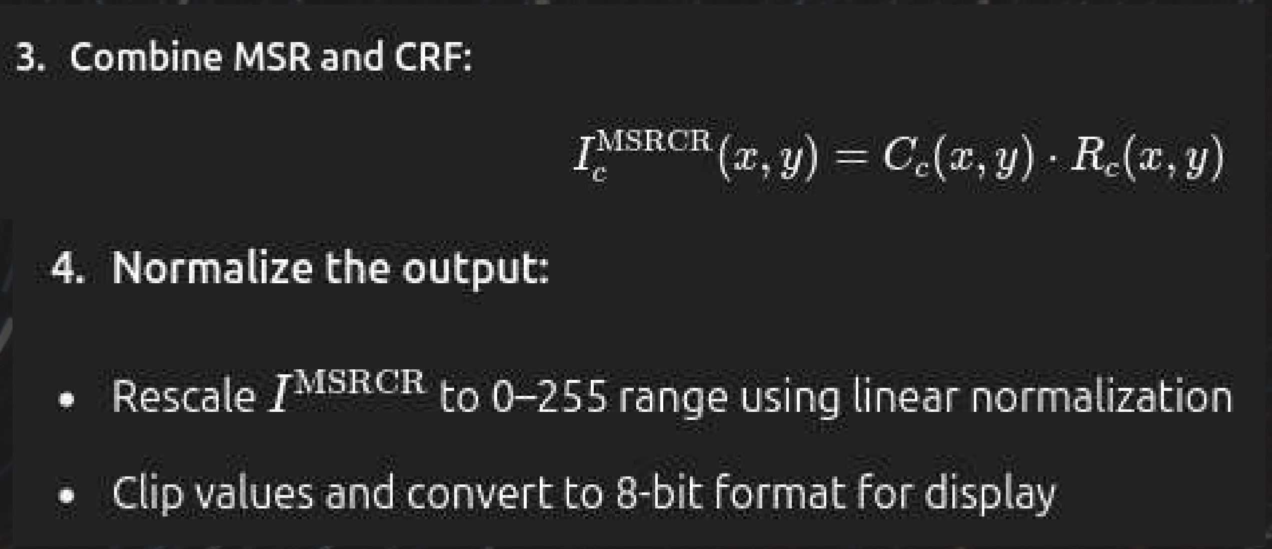
$$C_c(x, y) = \beta \cdot \log\left(\alpha \cdot \frac{I_c(x, y)}{\sum_{k \in \{R,G,B\}} I_k(x, y)} + 1\right)$$

- α, β are gain parameters controlling the strength of color correction

STEP 3(b): MSRCR

Multi-scale retinex with color restoration





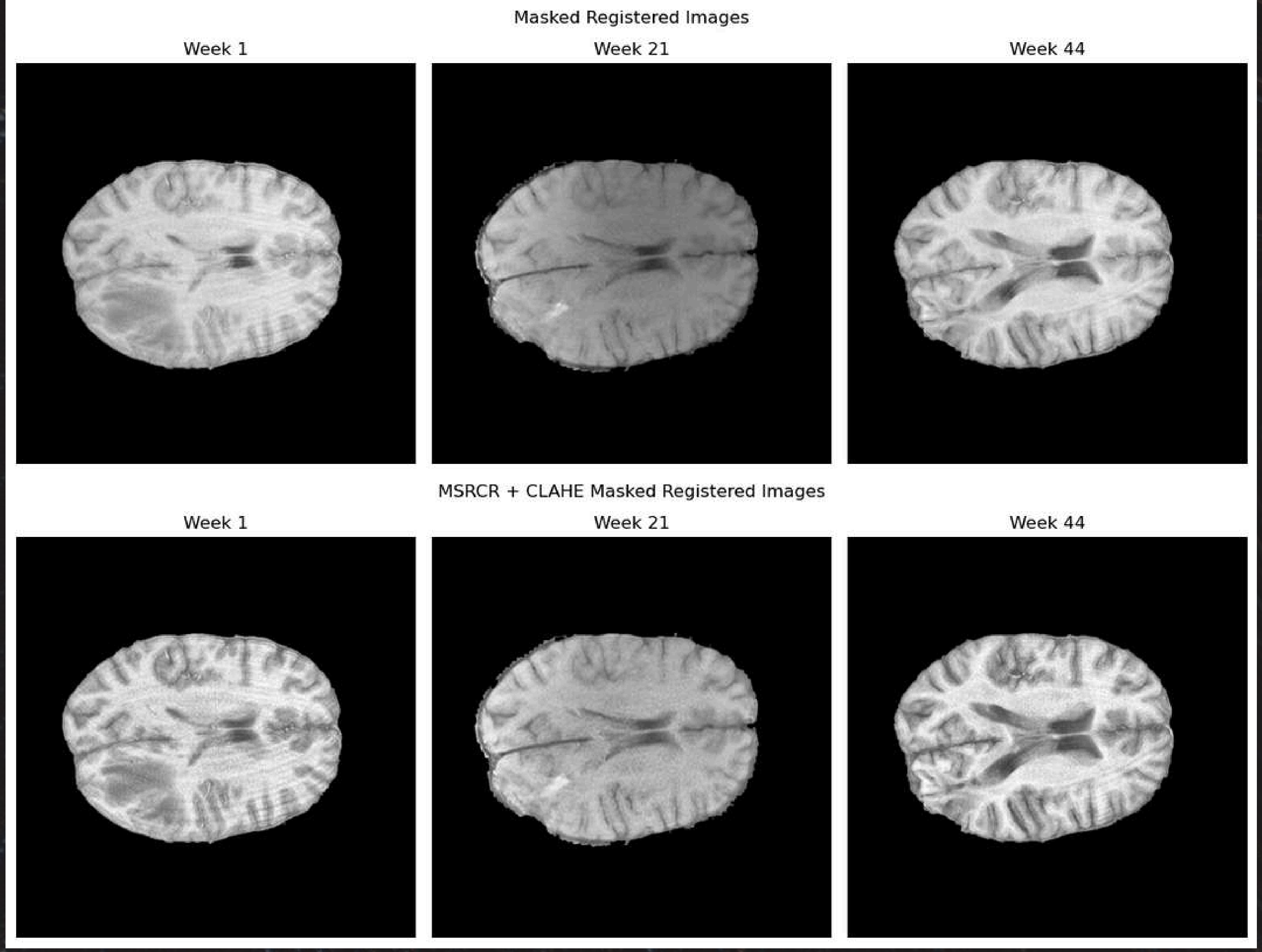
3. **Combine MSR and CRF:**

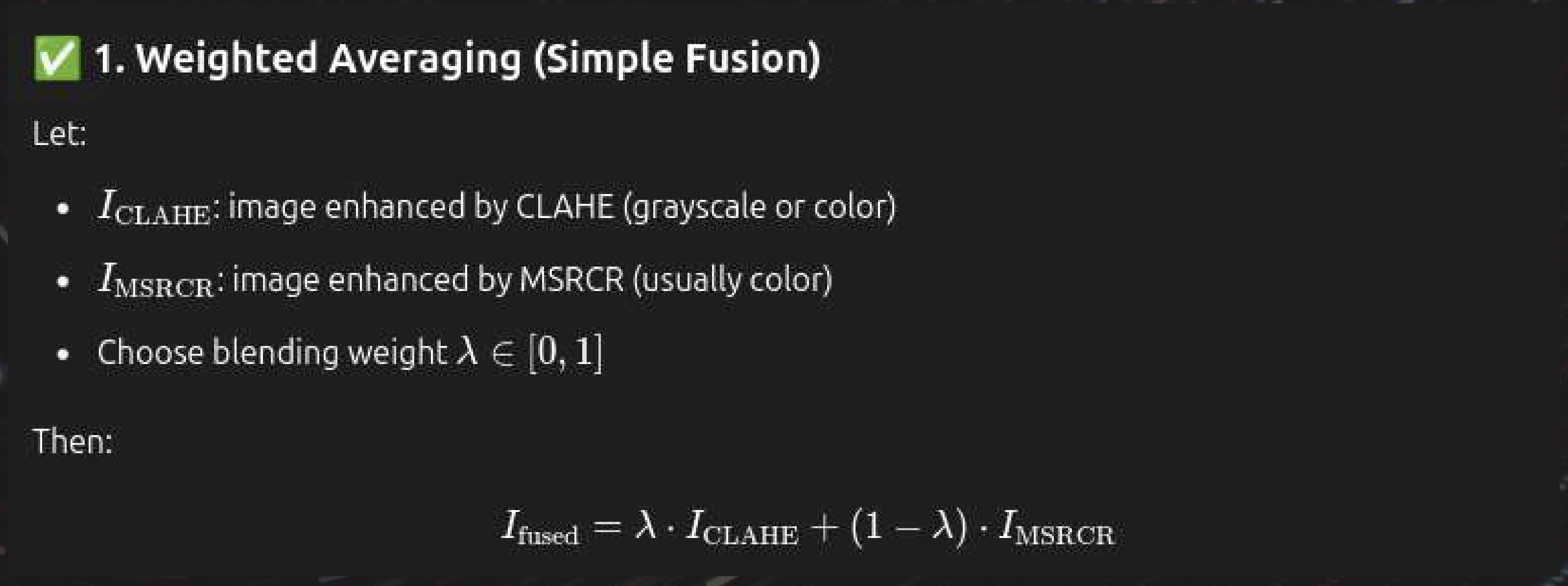
$$I_c^{\mathrm{MSRCR}}(x, y) = C_c(x, y) \cdot R_c(x, y)$$

4. **Normalize the output:**

- Rescale I^{MSRCR} to 0–255 range using linear normalization
- Clip values and convert to 8-bit format for display

STEP 3(c): MSRCR + CLAHE





✅ **1. Weighted Averaging (Simple Fusion)**

Let:

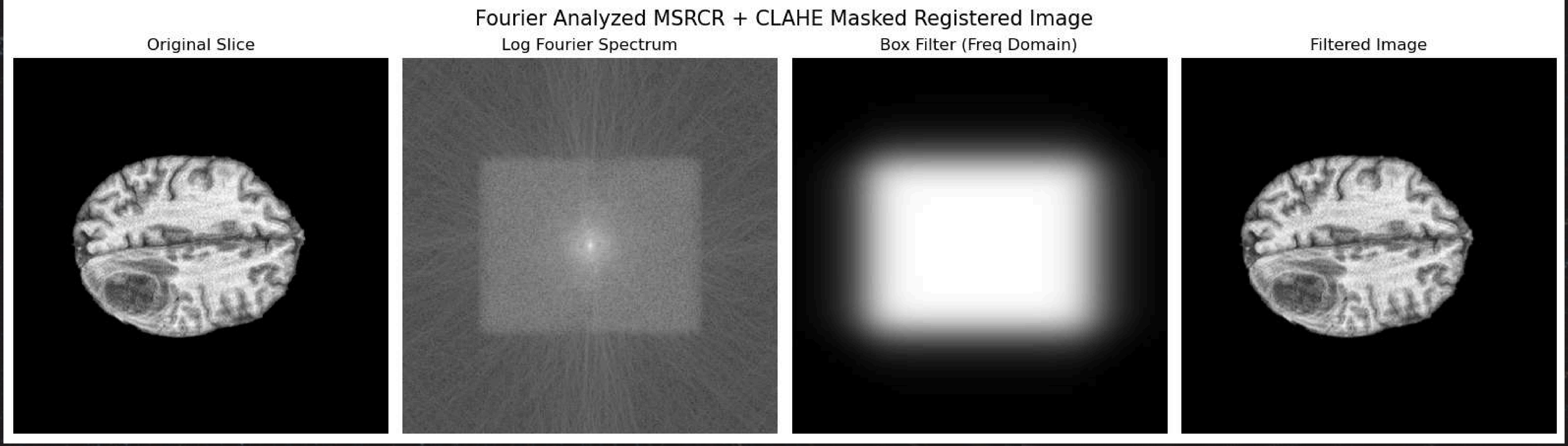
- I_{CLAHE}: image enhanced by CLAHE (grayscale or color)
- I_{MSRCR}: image enhanced by MSRCR (usually color)
- Choose blending weight $\lambda \in [0, 1]$

Then:

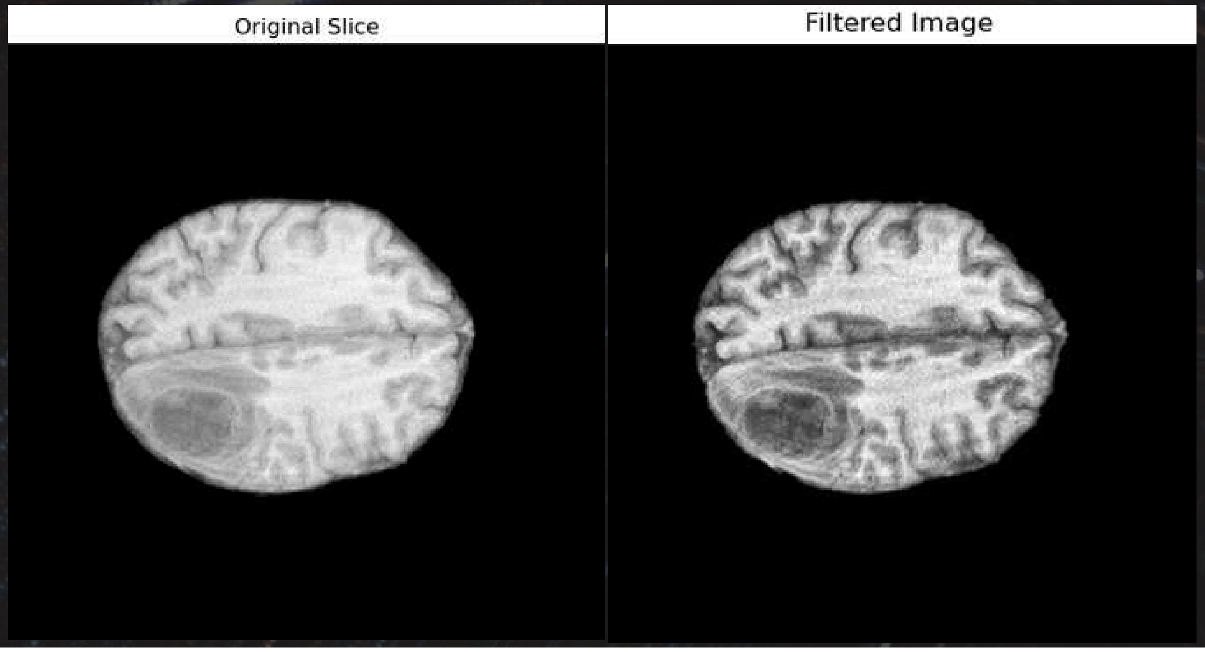
$$I_{\text{fused}} = \lambda \cdot I_{\text{CLAHE}} + (1 - \lambda) \cdot I_{\text{MSRCR}}$$

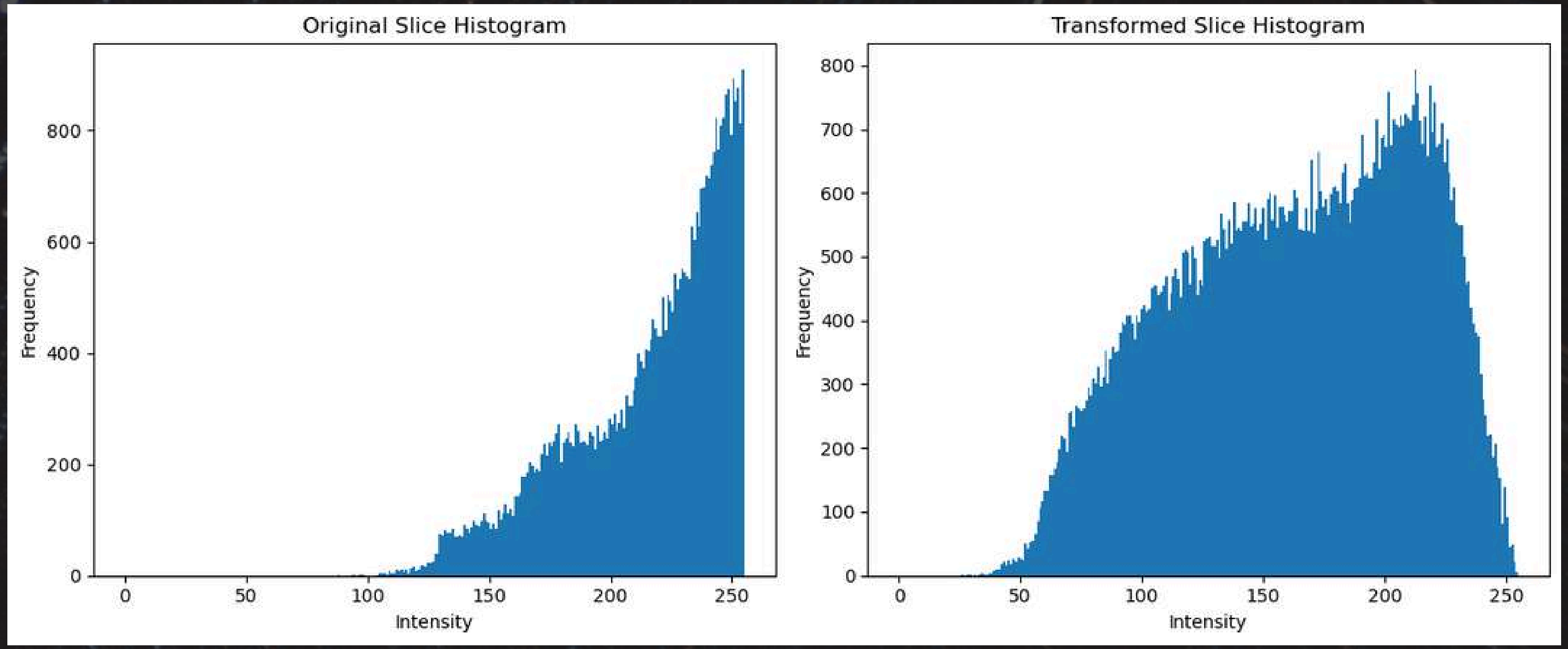
Chosen λ: 0.6

STEP 4: Fourier Analysis



STEP 5: Visual Inspection





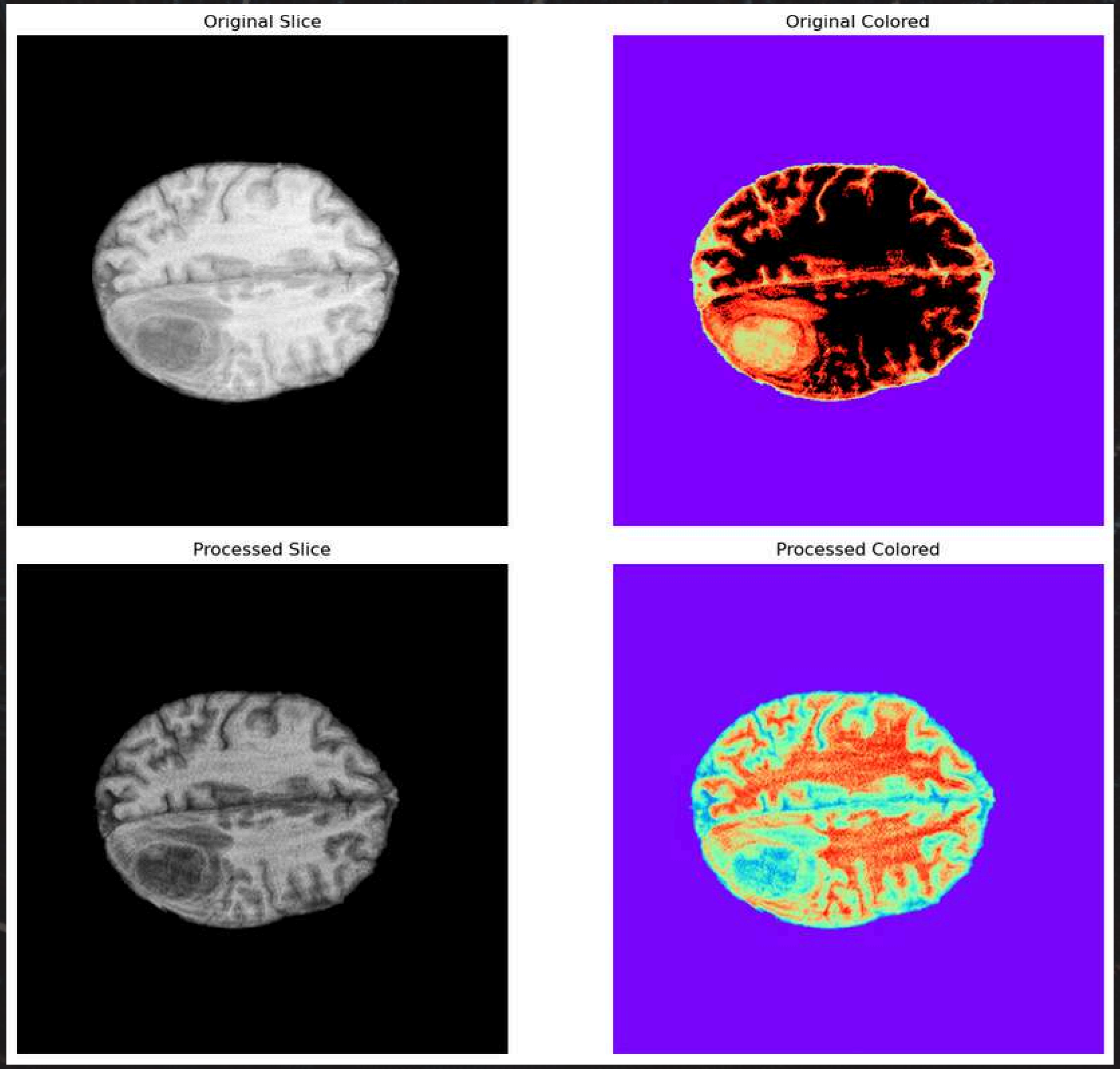
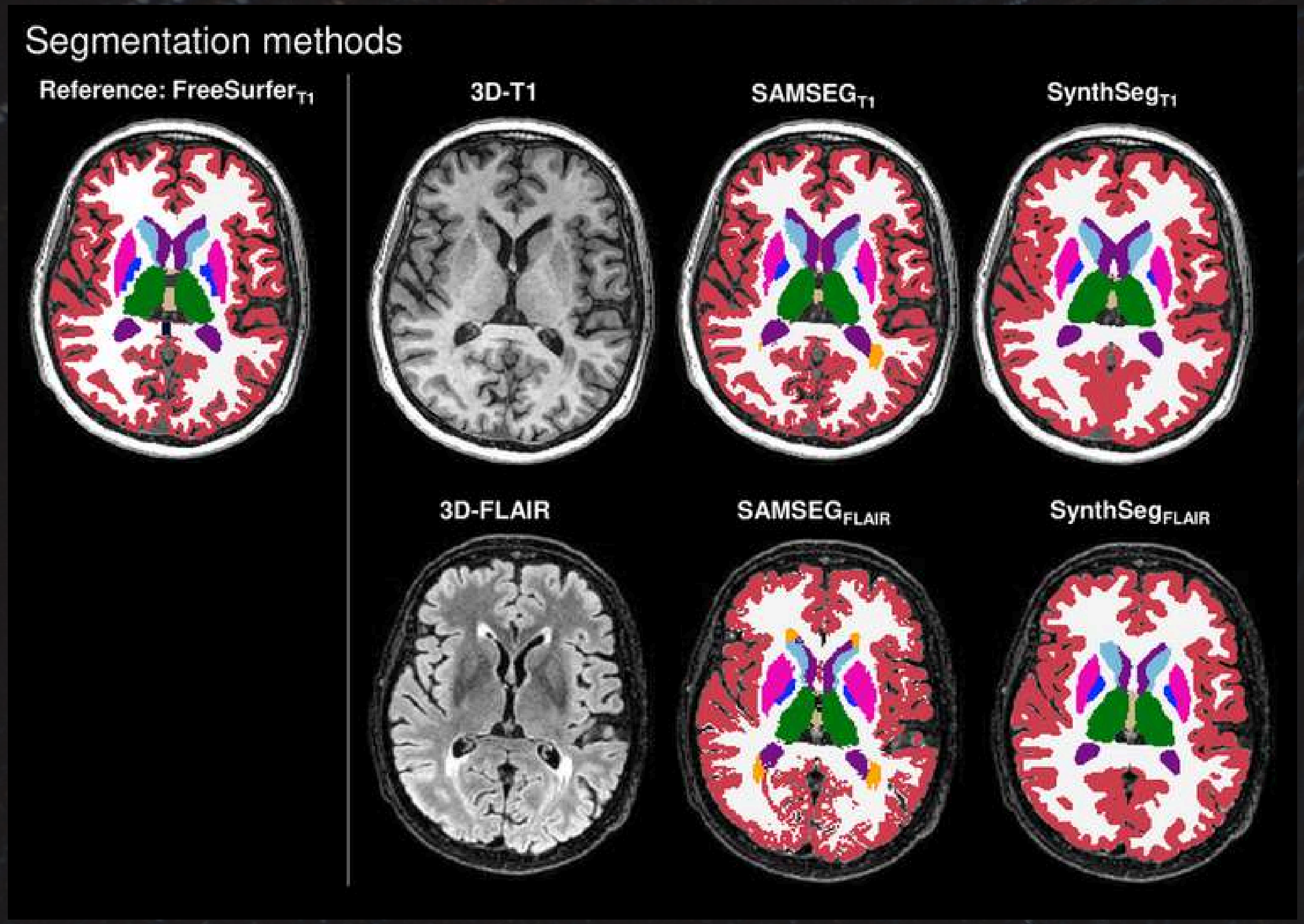
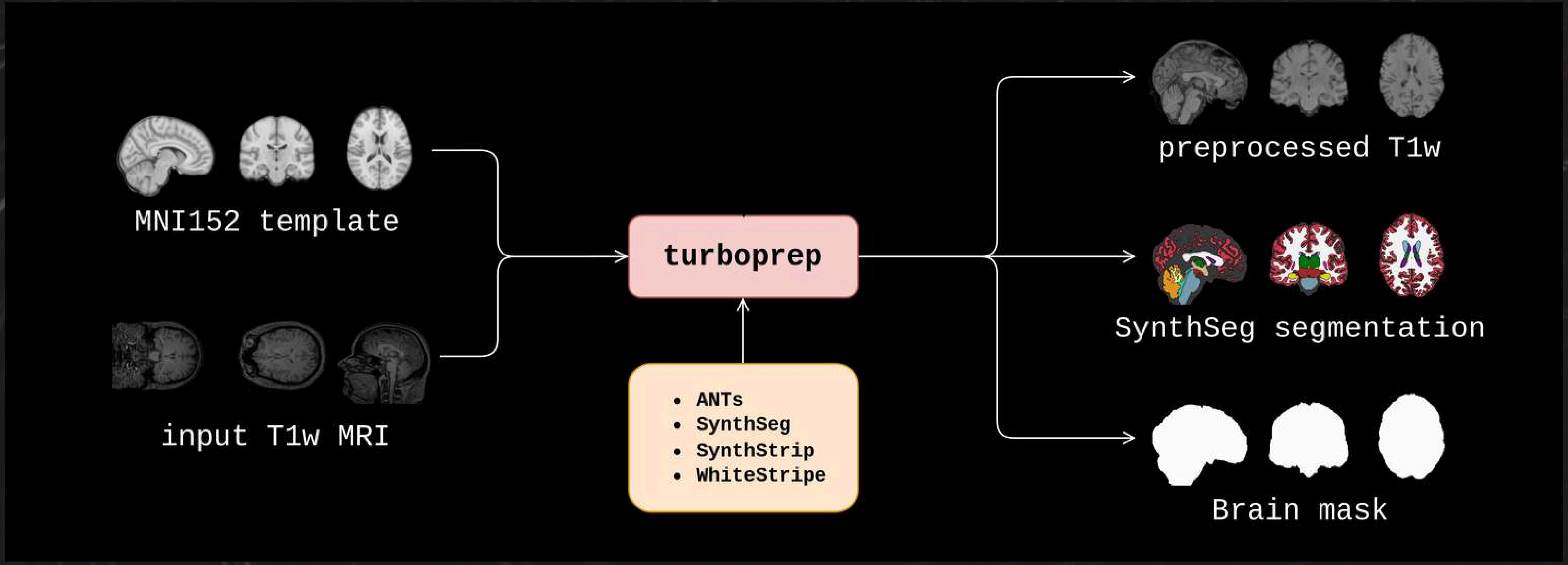


IMAGE PREPROCESSING

PART 2: SEGMENTATION MASK (TURBOPREP)

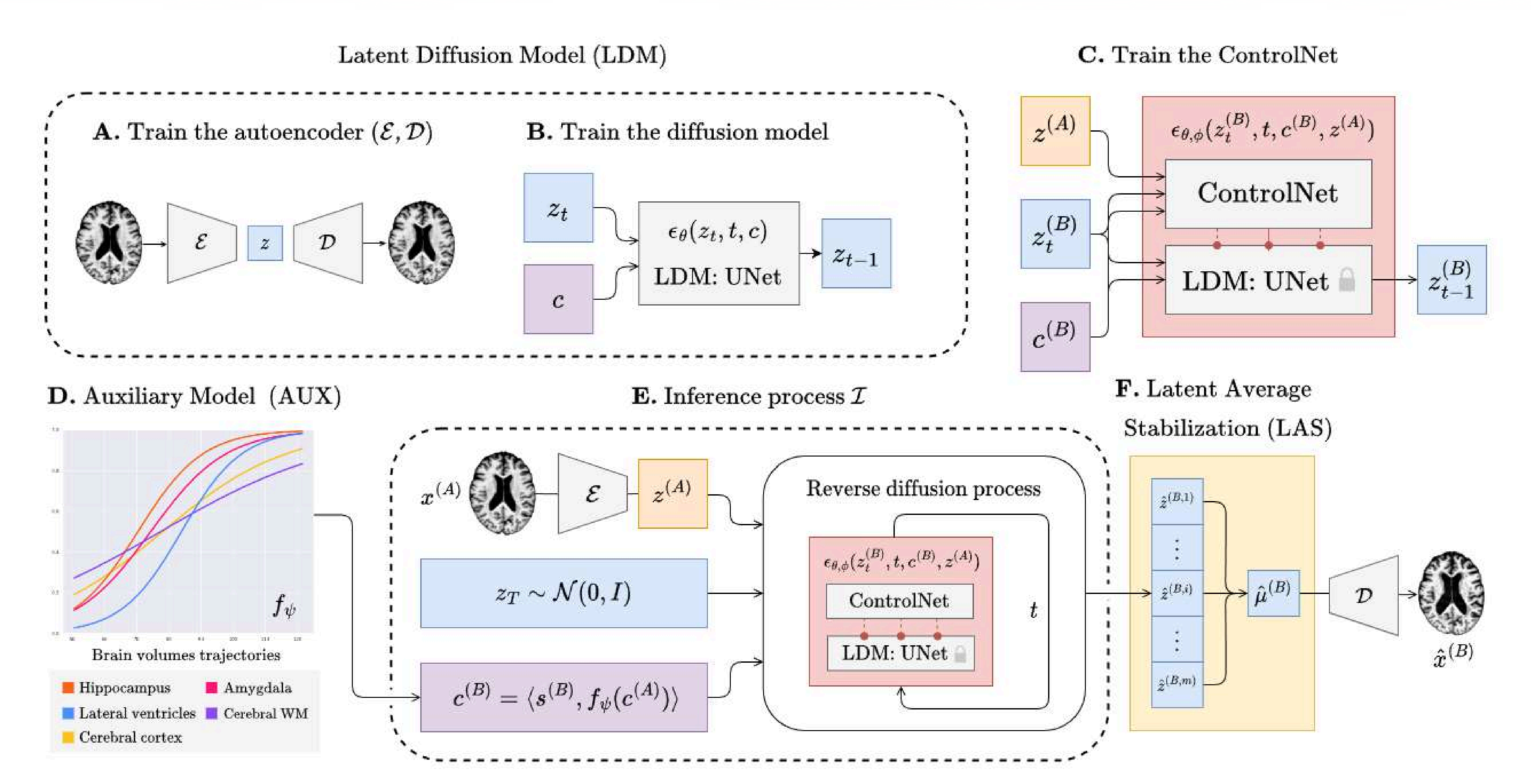




- ANTs
- SynthSeg
- SynthStrip
- WhiteStripe

* SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining

BRAIN LATENT PROGRESSION (BRLP)


Latent Diffusion Model (LDM)
A. Train the autoencoder $(\mathcal{E}, \mathcal{D})$
$\mathcal{E}$
z
$\mathcal{D}$
B. Train the diffusion model
z_t
c
$\epsilon_\theta(z_t, t, c)$
LDM: UNet
z_{t-1}
C. Train the ControlNet
$z^{(A)}$
$z_t^{(B)}$
$c^{(B)}$
$\epsilon_{\theta,\phi}(z_t^{(B)}, t, c^{(B)}, z^{(A)})$
ControlNet
LDM: UNet
$z_{t-1}^{(B)}$
D. Auxiliary Model (AUX)
f_ψ
Brain volumes trajectories
Hippocampus
Amygdala
Lateral ventricles
Cerebral WM
Cerebral cortex
E. Inference process $\mathcal{I}$
$x^{(A)}$
$\mathcal{E}$
$z^{(A)}$
$z_T \sim \mathcal{N}(0, I)$
$c^{(B)} = \langle s^{(B)}, f_\psi(c^{(A)}) \rangle$
Reverse diffusion process
$\epsilon_{\theta,\phi}(z_t^{(B)}, t, c^{(B)}, z^{(A)})$
ControlNet
LDM: UNet
t
F. Latent Average Stabilization (LAS)
$\hat{z}^{(B,1)}$
$\hat{z}^{(B,i)}$
$\hat{z}^{(B,m)}$
$\hat{\mu}^{(B)}$
$\mathcal{D}$
$\hat{x}^{(B)}$

AUXILIARY MODEL

The brain typically shrinks to some degree in healthy aging, but surprisingly does not lose neurons in large numbers. In alzheimer’s disease typically show approximately 7% lower overall brain volumes compared to healthy elderly individuals [3].

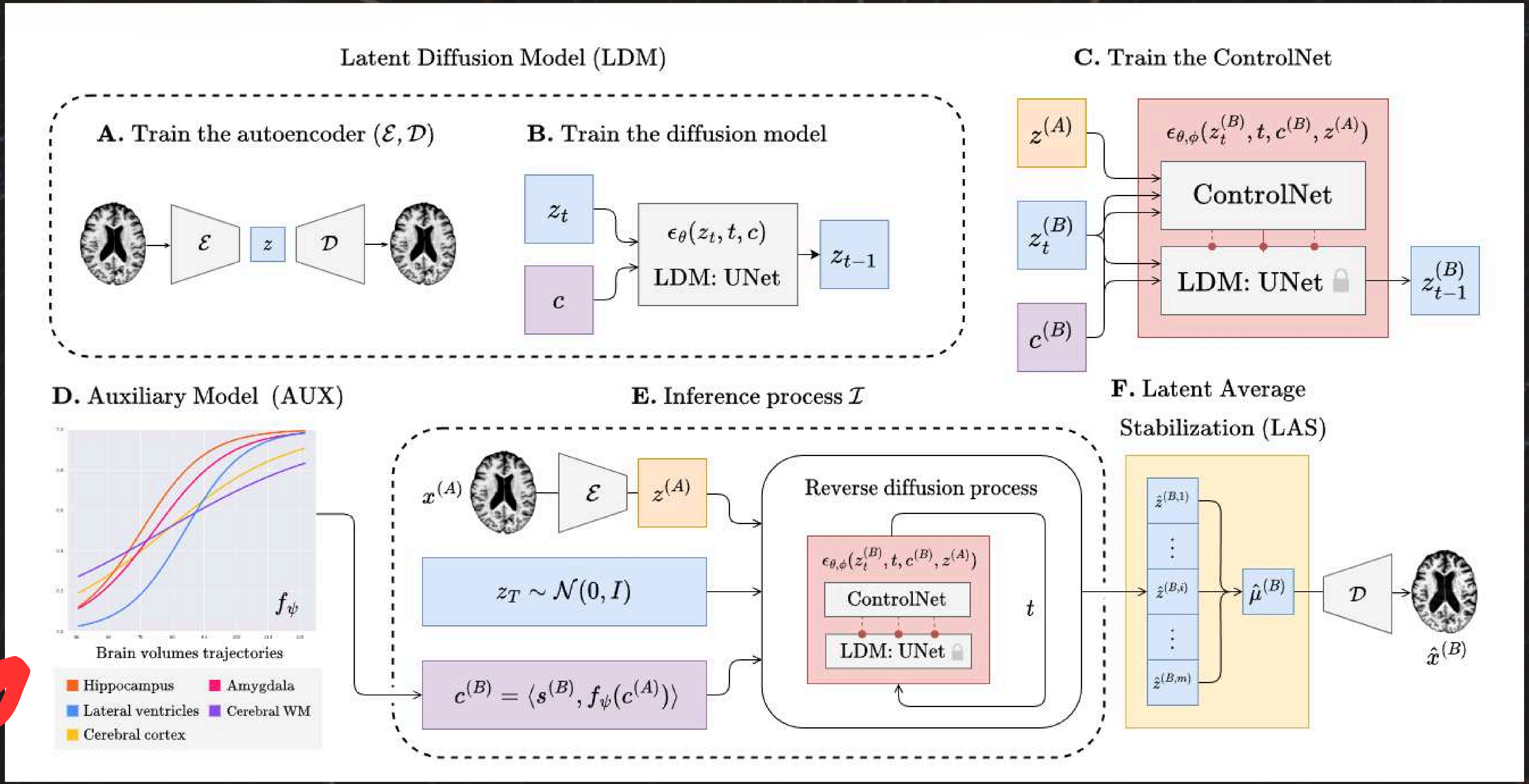
Brain of a 75 year old person

Brain of a person suffering from alzheimer's disease

AUXILIARY MODEL

BrLP [2] uses an Auxilary model to model this shrinkage over time.

It uses a rather restrictive model that uses Logistic Regression



Latent Diffusion Model (LDM)
A. Train the autoencoder $(\mathcal{E}, \mathcal{D})$
$\mathcal{E}$
z
$\mathcal{D}$
B. Train the diffusion model
z_t
c
$\epsilon_\theta(z_t, t, c)$
LDM: UNet
z_{t-1}
C. Train the ControlNet
$z^{(A)}$
$z_t^{(B)}$
$c^{(B)}$
$\epsilon_{\theta,\phi}(z_t^{(B)}, t, c^{(B)}, z^{(A)})$
ControlNet
LDM: UNet
$z_{t-1}^{(B)}$
D. Auxiliary Model (AUX)
f_ψ
Brain volumes trajectories
🟧 Hippocampus
🟥 Amygdala
🟦 Lateral ventricles
🟪 Cerebral WM
🟨 Cerebral cortex
E. Inference process $\mathcal{I}$
$x^{(A)}$
$\mathcal{E}$
$z^{(A)}$
$z_T \sim \mathcal{N}(0, I)$
$c^{(B)} = \langle s^{(B)}, f_\psi(c^{(A)}) \rangle$
Reverse diffusion process
$\epsilon_{\theta,\phi}(z_t^{(B)}, t, c^{(B)}, z^{(A)})$
ControlNet
LDM: UNet
t
F. Latent Average Stabilization (LAS)
$\hat{z}^{(B,1)}$
$\hat{z}^{(B,i)}$
$\hat{z}^{(B,m)}$
$\hat{\mu}^{(B)}$
$\mathcal{D}$
$\hat{x}^{(B)}$

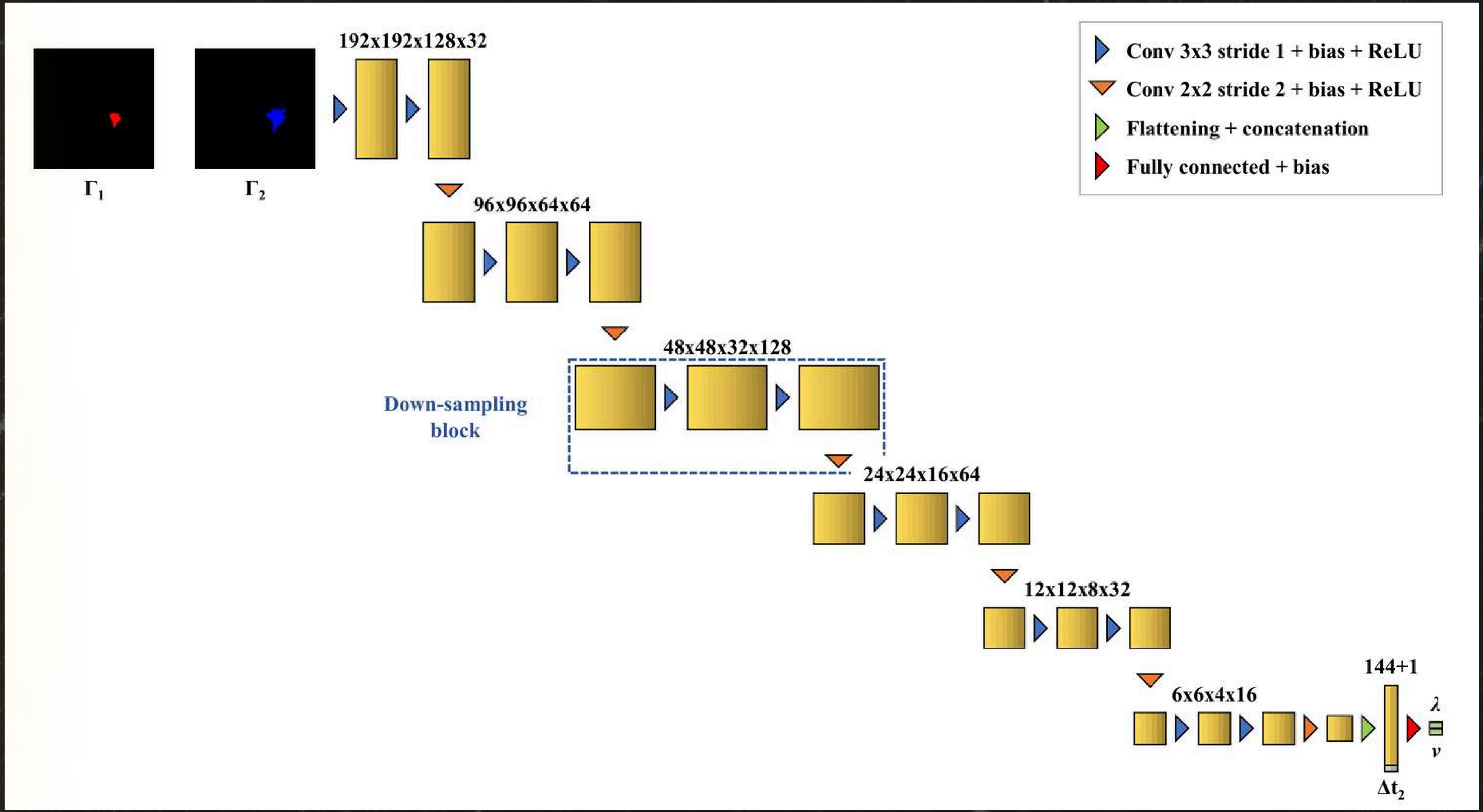
AUXILIARY MODEL

This is a reaction diffusion model that we use to better model the growth of tumor using a differential equation framework that incorporates both cell migration (diffusion) and population growth (reaction). They are Described through these 3 equations

$$
\begin{aligned}
\frac{\partial c(x,t)}{\partial t} &= \nabla \quad (D(x)\,\nabla c(x,t)) + \rho\, c(x,t)(1 - c(x,t)) && \forall x \in \Omega,\ \forall t > 0 \\
c(x,0) &= c_0(x) && \forall x \in \Omega \\
D(x)\,\nabla c(x,t) \cdot n_{\partial\Omega}(x) &= 0 && \forall x \in \partial\Omega
\end{aligned}
$$

The terms that are our parameters in the following equations are the D(x) which represents diffusion rate and pc(x,t) representing the reaction term

SOLUTION

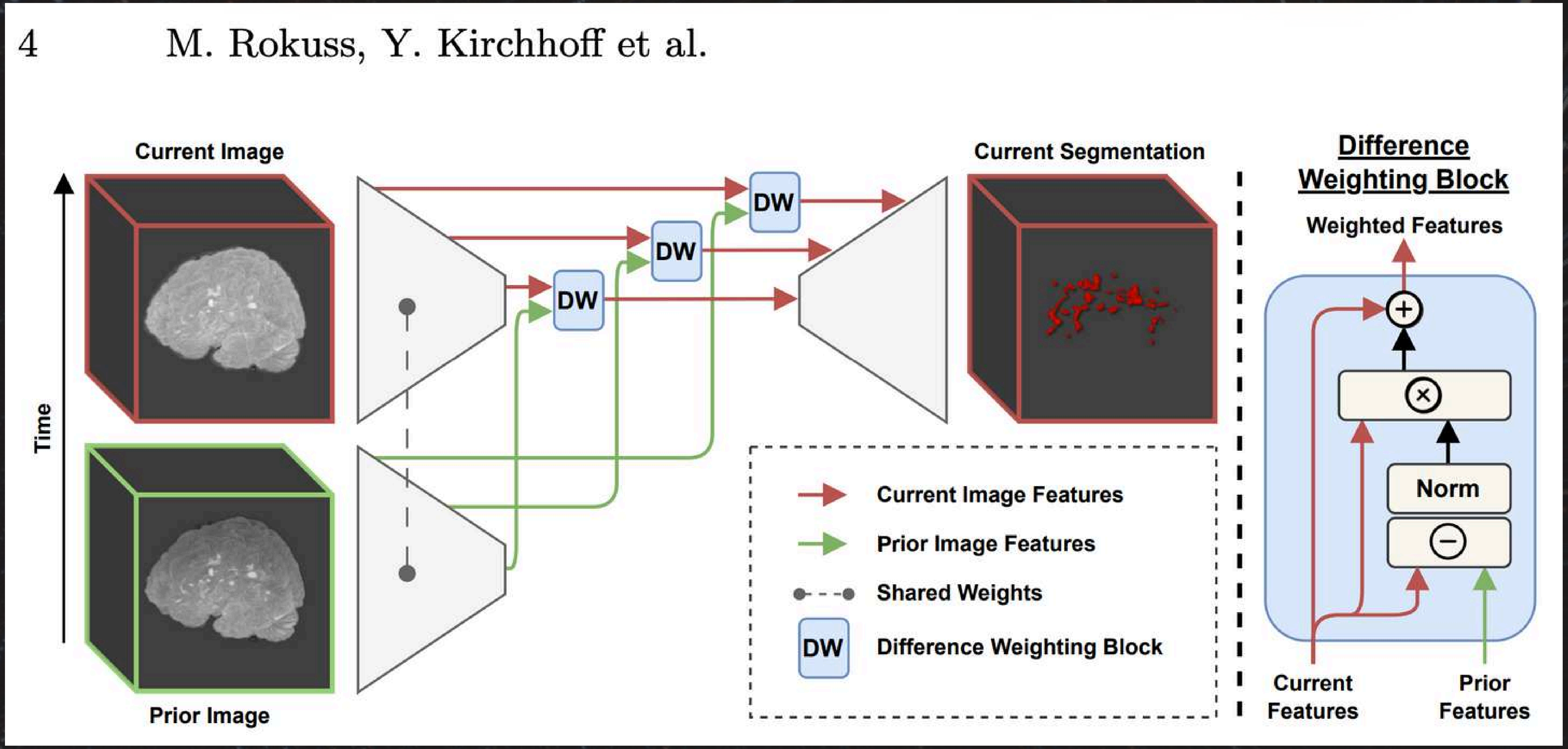


$$d_{\text{white}} = \frac{\lambda\, v}{2}$$

$$\rho = \frac{v}{2\,\lambda}$$

DIFFERENCE WEIGHTING BLOCK

THE AUTHOR'S APPROACH [1]

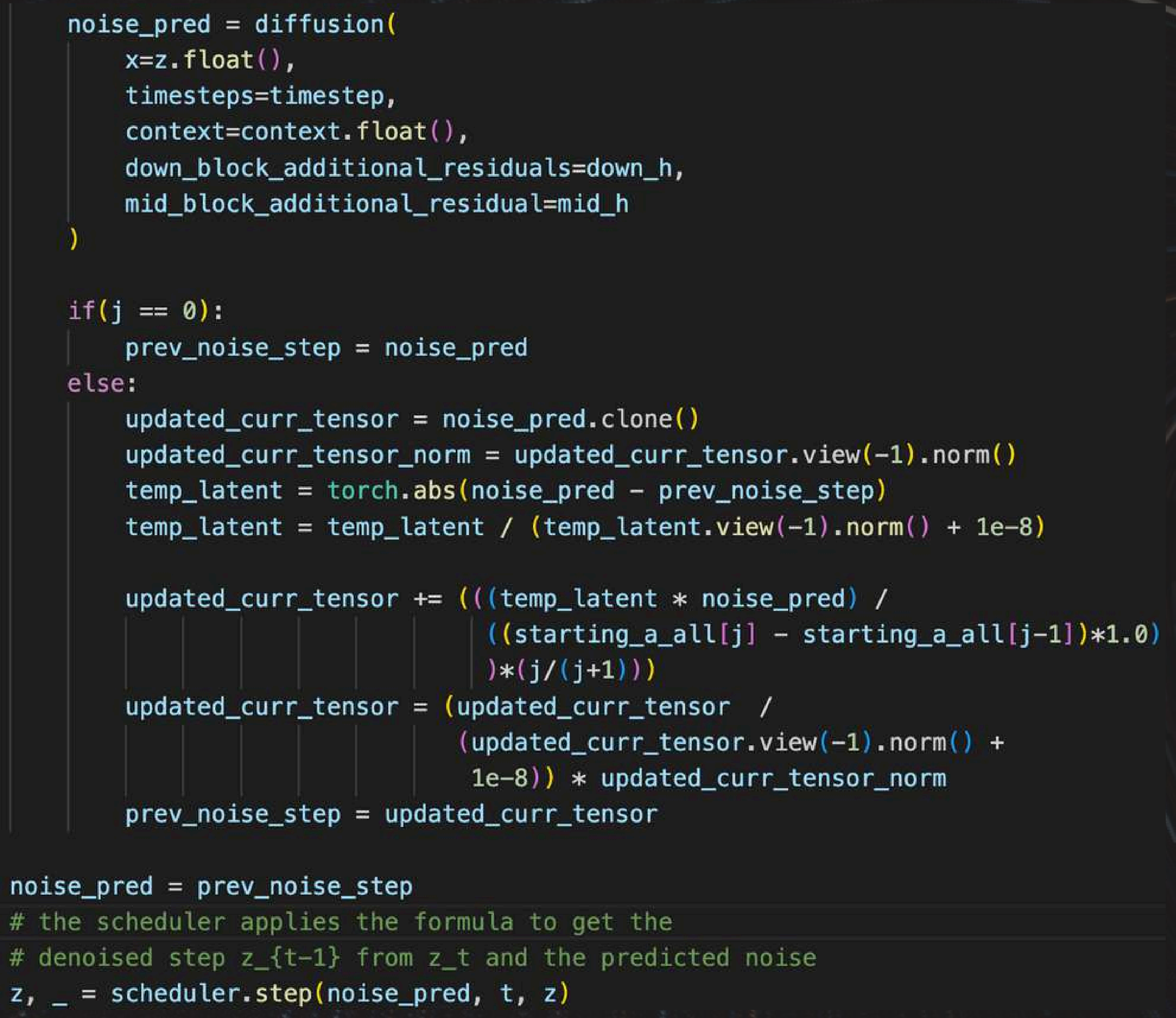


4 M. Rokuss, Y. Kirchhoff et al.

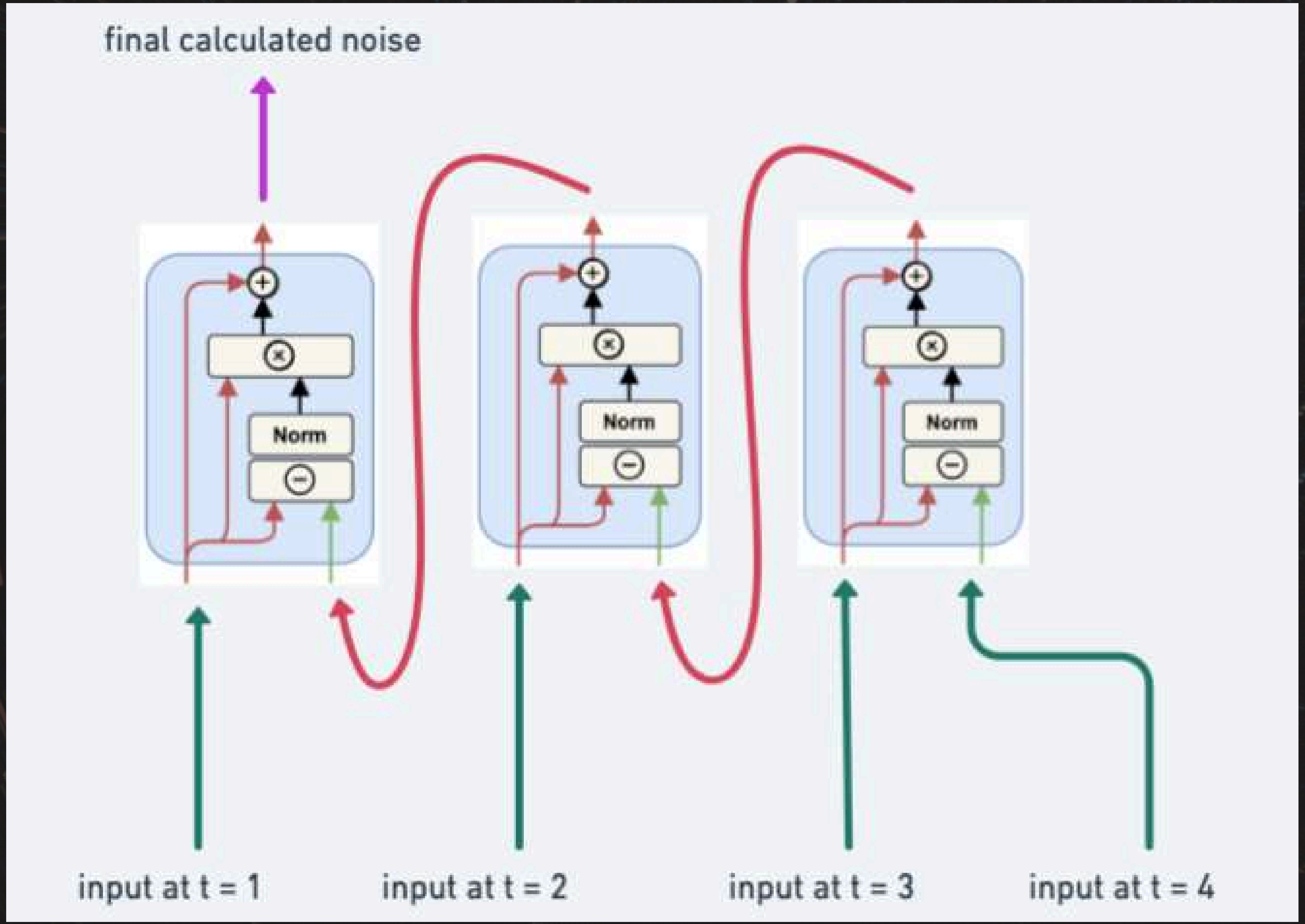
Incorporate explicit architectural bias to fully leverage the benefits of longitudinal information

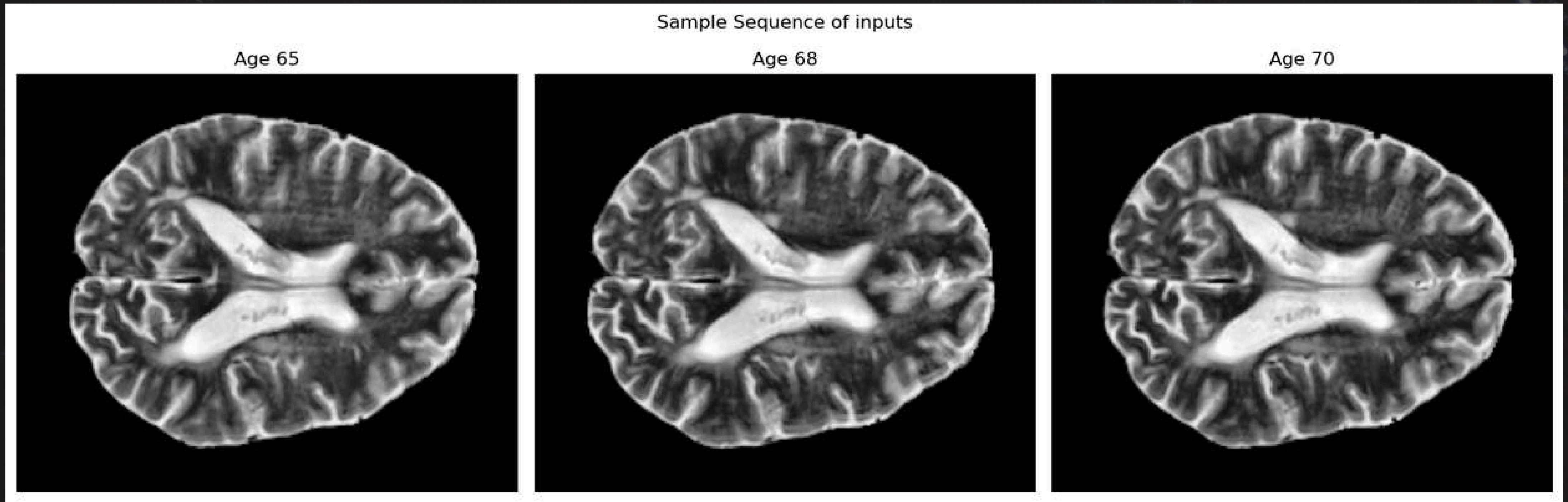
We extend this idea in the BrLP[2] pipeline to include variable number of past scans (not limited to just 1)

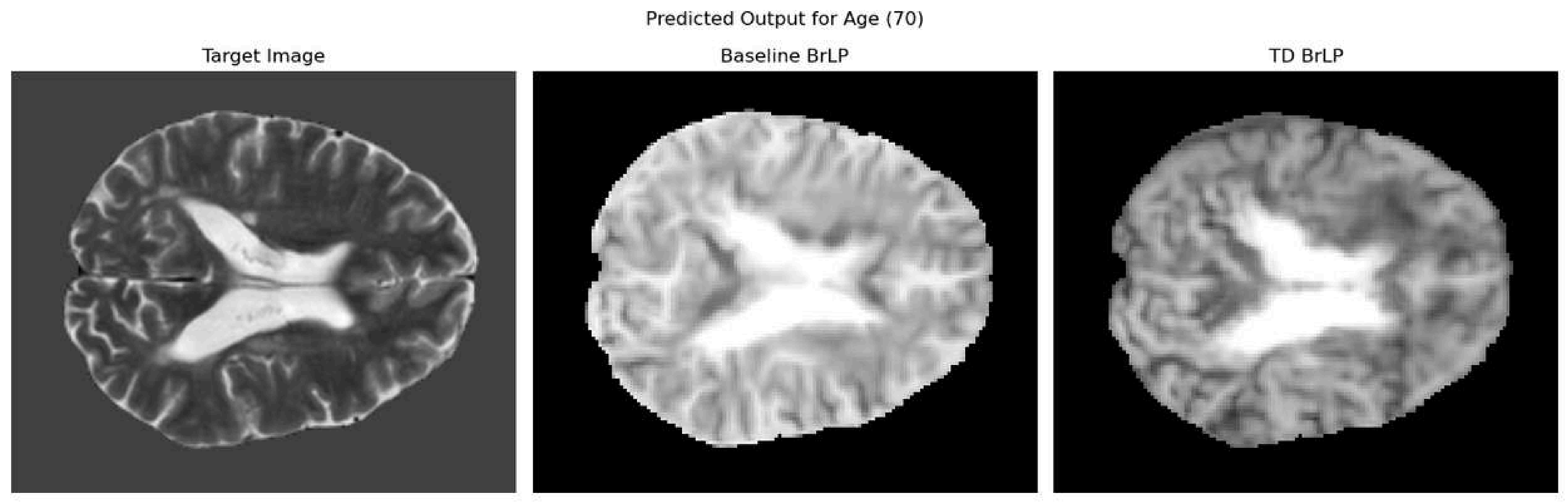
DIFFERENCE WEIGHTING BLOCK

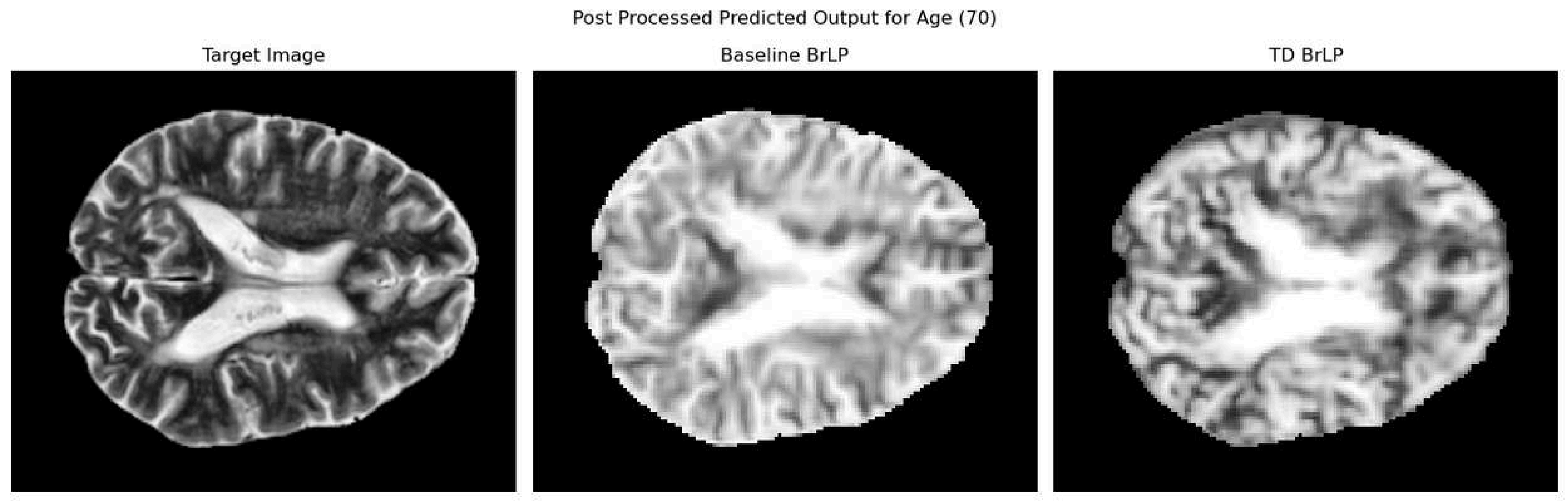


```
    noise_pred = diffusion(
        x=z.float(),
        timesteps=timestep,
        context=context.float(),
        down_block_additional_residuals=down_h,
        mid_block_additional_residual=mid_h
    )

    if(j == 0):
        prev_noise_step = noise_pred
    else:
        updated_curr_tensor = noise_pred.clone()
        updated_curr_tensor_norm = updated_curr_tensor.view(-1).norm()
        temp_latent = torch.abs(noise_pred - prev_noise_step)
        temp_latent = temp_latent / (temp_latent.view(-1).norm() + 1e-8)

        updated_curr_tensor += (((temp_latent * noise_pred) /
                                 ((starting_a_all[j] - starting_a_all[j-1])*1.0)
                                )*(j/(j+1)))
        updated_curr_tensor = (updated_curr_tensor  /
                               (updated_curr_tensor.view(-1).norm() +
                                1e-8)) * updated_curr_tensor_norm
        prev_noise_step = updated_curr_tensor

noise_pred = prev_noise_step
# the scheduler applies the formula to get the
# denoised step z_{t-1} from z_t and the predicted noise
z, _ = scheduler.step(noise_pred, t, z)
```









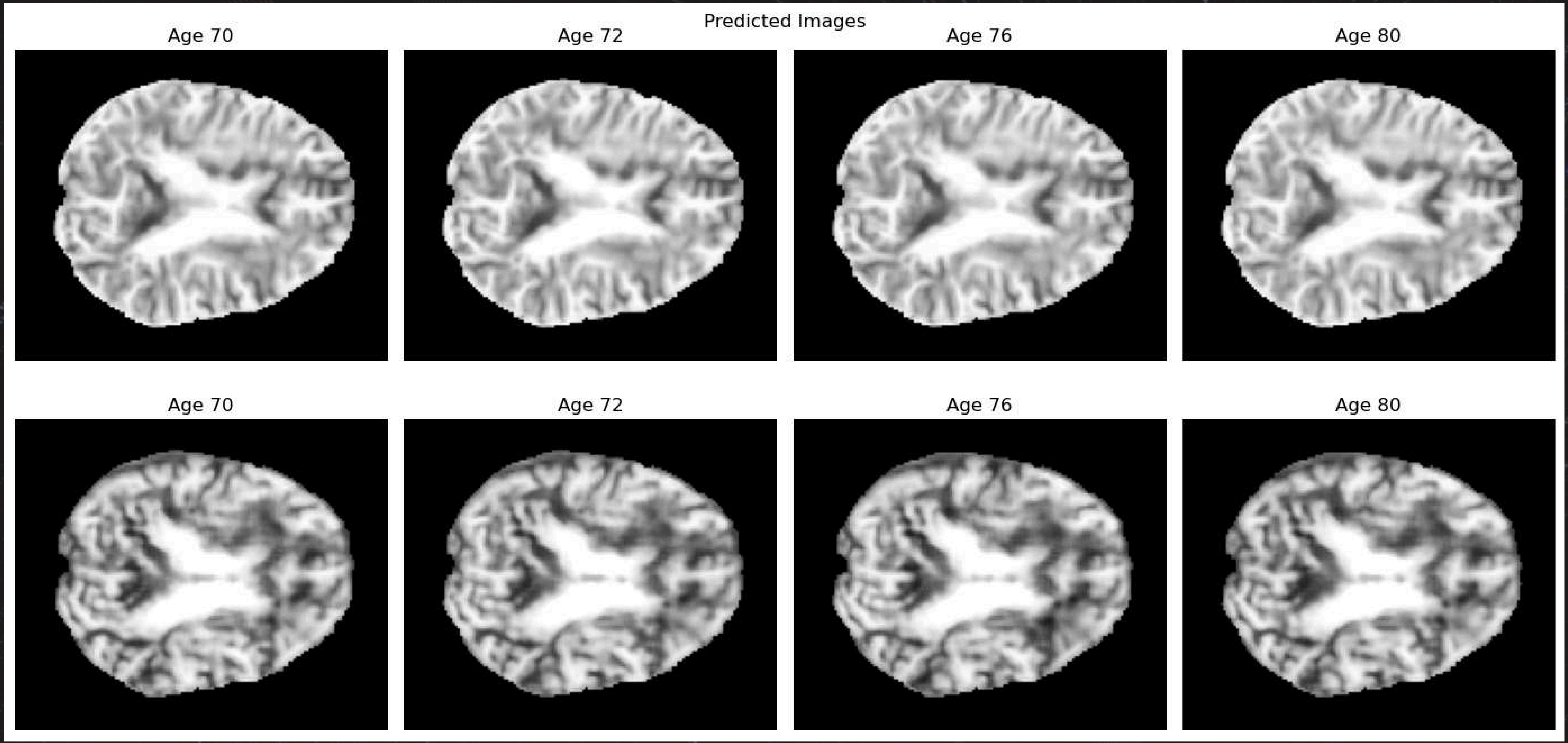
MS OUTPUTS

average inference time original: 5 sec

for each extra past scan 4-5 sec increase in inference time

MSE
original: 8.34077868e+02
ours: 8.09714383e+02

no. of params same as the original model (mathematical block)


Predicted Images
Age 70
Age 72
Age 76
Age 80
Age 70
Age 72
Age 76
Age 80

OTHER THINGS WE TESTED AND FAILURES

- Could not find enough open source data available to fine-tune the model thus shifted our focus to brain tumors instead.

- We implemented and modified the base BrLP code for training using our methods

- However, the fine-tuning did not yield fruitful results due to :-
    a. The resolution of data being too low compared to whats expected by model thus causing significant distortion and blurring
    b. The brain tumor sequences were not able to register well thus causing undesirable effects and tranformations.

FUTURE SCOPE

- Fixing the resolution and image sequence registration issues in the data set
- Build upon the temporal difference idea and implement Eligibility trace based and monte carlo based methods (from reinforcement learning) to improve the model
- Explore other approached under the history weighting category apart from TD learning as done in RL

REFERENCES

[1] https://arxiv.org/pdf/2409.13416

[2] https://papers.miccai.org/miccai-2024/paper/0511_paper.pdf

[3] https://pubs.rsna.org/doi/10.1148/radiology.178.1.1984290

[4]https://github.com/MIC-DKFZ/BraTPRO/blob/master/dataset_download/download_and_convert_dataset.py

THANK YOU
GOOD SIR