Architectural Paradigms for Disentangled Spatiotemporal Tumor Modeling: From Baseline 3D Variational Autoencoders to Physics-Informed Neural ODE Integration

Executive Summary

The convergence of generative deep learning and dynamical systems theory offers a transformative approach to computational oncology, particularly in the forecasting of solid tumor growth from longitudinal Magnetic Resonance Imaging (MRI). This report delineates a comprehensive, incremental experimental framework for implementing a 3D Variational Autoencoder (VAE) tailored to the complexities of volumetric medical data. The ultimate objective is to construct a latent representation that is not merely a compressed code, but a disentangled state vector suitable for driving downstream Neural Ordinary Differential Equations (Neural ODEs). By rigorously progressing from a baseline 3D convolutional architecture to advanced variants incorporating Total Correlation penalties and semi-supervised physics constraints, we establish a pathway to overcome the curse of dimensionality inherent in voxel-wise modeling.

The analysis synthesizes literature on state-of-the-art VAE architectures—including β-TCVAE, DIP-VAE, and Spatial Broadcast Decoders—with the specific requirements of tumor growth dynamics modeled by Gompertzian and reaction-diffusion laws. We identify that standard VAEs suffer from latent entanglement, rendering them unsuitable for differential equation modeling where state variables must possess distinct physical interpretations (e.g., volume versus shape). To remediate this, we propose three distinct experimental phases: establishing reconstruction fidelity with Group-Normalized 3D ResNets; inducing unsupervised disentanglement via coordinate-aware decoding and statistical independence constraints; and finally, enforcing semantic alignment through semi-supervised regularization. This report provides exhaustive implementation details for the PyTorch deep learning framework, addressing critical engineering challenges such as memory-constrained 3D convolution, gradient estimation bias in small batches, and the stabilization of the rate-distortion trade-off via cyclic annealing schedules.

1. Theoretical Foundations: The State-Space Hypothesis in Medical Imaging

1.1 The Curse of Dimensionality in Spatiotemporal Forecasting

The fundamental challenge in modeling tumor evolution lies in the massive dimensionality of the observational space. A single multi-modal MRI scan—comprising T1-weighted, T1-weighted with gadolinium contrast (T1-Gd), T2-weighted, and Fluid Attenuated Inversion Recovery (FLAIR) sequences—is typically represented as a 4D tensor $X \in \mathbb{R}^{C \times D \times H \times W}$. For a standard resolution of 128^3 voxels and 4 channels, the input dimension exceeds 8 million variables.[1]

Attempting to model the temporal trajectory of this system directly in voxel space via Partial Differential Equations (PDEs) or raw Video Prediction networks is computationally intractable and statistically ill-posed given the scarcity of longitudinal medical datasets. Clinical trials often provide only sparse temporal snapshots (e.g., scans every 2–3 months), insufficient for learning pixel-wise dynamics without strong inductive biases.[2] Furthermore, voxel intensities are prone to scanner-induced stochasticity, bias field artifacts, and non-biological noise, which can destabilize differential equation solvers sensitive to initial conditions.

1.2 The Manifold Assumption and Neural ODEs

To surmount these limitations, we adopt the manifold hypothesis: the high-dimensional anatomical data resides on a lower-dimensional manifold governed by a compact set of latent factors, $\mathbf{z} \in \mathbb{R}^d$, where $d \ll 8 \times 10^6$. If these latent factors capture the meaningful biological variation—such as tumor burden, edema extension, and necrotic core composition—they can serve as the state variables for a dynamical system.

This motivates the integration of VAEs with Neural ODEs. In this paradigm, the VAE functions as a non-linear state estimator, mapping the observation space $\mathcal{X}$ to the latent state space $\mathcal{Z}$. The temporal evolution is then modeled entirely within $\mathcal{Z}$ via an ODE:

$$\frac{d\mathbf{z}(t)}{dt} = f_\theta(\mathbf{z}(t), t)$$

where f_θ is a neural network parameterized to approximate the underlying biological growth laws.4 The solution to this ODE, $\mathbf{z}(t) = \mathbf{z}(t_0) + \int_{t_0}^t f_\theta(\mathbf{z}(\tau), \tau) d\tau$, allows for continuous-time forecasting of the tumor state, which can then be decoded back to the image space for clinical visualization.5

1.3 The Necessity of Disentanglement

Crucially, the efficacy of the Neural ODE is strictly contingent on the geometric properties of the latent space $\mathcal{Z}$. A standard VAE, trained solely to minimize reconstruction error, typically yields an *entangled* representation where individual dimensions of $\mathbf{z}$ encode complex mixtures of factors (e.g., z_1 represents "tumor size minus skull thickness").

Such entanglement is catastrophic for dynamical modeling. If the state variable z_1

conflates a dynamic biological factor (tumor size) with a static anatomical factor (skull size), the ODE function $f_\theta$ must learn highly complex, non-smooth vector fields to separate these influences. Conversely, a *disentangled* representation—where one subspace encodes "volume" and another "spatial location"—allows the ODE to learn simpler, more robust dynamics (e.g., exponential or Gompertzian growth on the volume dimension, translational invariance on the location dimensions).[1] Thus, the architectural design of the VAE is not merely a matter of image quality, but of ensuring the *identifiability* and *stability* of the downstream physical model.

2. Experiment 1: The Baseline 3D Convolutional VAE

The initial experimental phase focuses on establishing the baseline infrastructure for processing 3D volumetric data. The primary objective is to achieve high-fidelity reconstruction of multi-modal brain MRIs. Without the ability to accurately reproduce fine anatomical details—such as the irregular boundaries of a glioblastoma or the subtle intensity gradients of peritumoral edema—any subsequent disentanglement or dynamic modeling is futile.

2.1 Data Preprocessing and Tensor Specification

The input to the model consists of 3D MRI volumes derived from datasets such as BraTS (Brain Tumor Segmentation). The data pipeline must handle the heterogeneity of MRI acquisition.

Intensity Normalization: Unlike CT scans which have absolute Hounsfield units, MRI signal intensities are relative and vary across scanners. Z-score normalization (subtracting the mean and dividing by the standard deviation of brain voxels per subject) is mandatory to standardize the input distribution.[7]

$$x_{norm} = \frac{x - \mu_{brain}}{\sigma_{brain}}$$

This normalization ensures that the VAE models structural variation rather than arbitrary intensity shifts.

Input Dimensions: We standardize inputs to crops of size $128 \times 128 \times 128$ voxels. This represents a trade-off between anatomical context and GPU memory constraints. With 4 modalities (channels), the input tensor $X$ has shape $(B, 4, 128, 128, 128)$.

2.2 Encoder Architecture: The 3D ResNet Backbone

Standard 2D convolutional architectures (like VGG or simple stacked CNNs) scale poorly to 3D due to the vanishing gradient problem in deep volumetric networks. We therefore implement a **3D Residual Network (ResNet-18)** backbone, adapted for variational inference.[1]

PyTorch Implementation Components:

- **Initial Convolution:** The network begins with a $7 \times 7 \times 7$ convolution (stride 2) to rapidly reduce spatial resolution and memory footprint, followed by a max-pooling layer. However, for finer detail preservation in segmentation tasks, a $3 \times 3 \times 3$ convolution with stride 1 or 2 is often preferred to avoid aggressive information loss at the input stage.[8]

- **Residual Blocks:** The core comprises four stages of BasicBlocks. In 3D, a BasicBlock consists of two $3 \times 3 \times 3$ convolutions with a skip connection: $y = \mathcal{F}(x, \{W_i\}) + x$. The skip connection facilitates the flow of gradients during backpropagation, essential for training deeper 3D networks.[9]
- **Dimensionality Reduction:** Feature map dimensions are progressively halved (from 64^3 to 8^3) while channel width is doubled (e.g., 32 $\to$ 64 $\to$ 128 $\to$ 256).
- **Latent Bottleneck:** The final feature volume (e.g., $256 \times 8 \times 8 \times 8$) is flattened. Two parallel linear layers project this vector to the latent parameters: the mean $\boldsymbol{\mu} \in \mathbb{R}^{128}$ and log-variance $\log \boldsymbol{\sigma}^2 \in \mathbb{R}^{128}$.

2.3 The Normalization Dilemma: GroupNorm vs. BatchNorm

A critical engineering decision in 3D VAEs is the choice of normalization layer. Batch Normalization (BN) is the standard in 2D computer vision, but it fails in the 3D medical domain due to the "small batch" problem.

Reasoning:

Processing 128^3 volumes with 4 channels and deep 3D filters consumes massive GPU memory. On a standard high-end GPU (e.g., NVIDIA A100 40GB), the maximum feasible batch size (B) is often limited to 2, 4, or 8. BN estimates the population mean and variance using the current mini-batch statistics. When B is small, these estimates become highly noisy, introducing stochasticity that destabilizes training and leads to significant errors during inference (where running statistics are used).10

Implementation:

We replace all instances of nn.BatchNorm3d with Group Normalization (nn.GroupNorm).

- **Mechanism:** GN divides the channels into groups (e.g., 32 groups) and computes statistics (mean and variance) within each group for *each sample independently*. This makes the normalization calculation independent of batch size B.[10]
- PyTorch Syntax: nn.GroupNorm(num_groups=8, num_channels=C).
  Studies have consistently shown that GN outperforms BN in 3D medical segmentation and reconstruction tasks when $B < 16$.13

2.4 The Baseline Loss Function: ELBO

The objective function is the negative Evidence Lower Bound (ELBO), composed of a reconstruction term and a regularization term:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}}$$

Reconstruction Loss ($\mathcal{L}_{\text{recon}}$):

Assuming a Gaussian likelihood for the decoder $p_\theta(x|\mathbf{z})$, the negative log-likelihood is equivalent to the Mean Squared Error (MSE).

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^N \| x_i - \hat{x}_i \|^2$$

- **Implementation Note:** Use reduction='sum' in PyTorch's MSELoss. Using mean scales the loss by the number of voxels ($128^3 \approx 2 \times 10^6$), making it vanishingly small relative to the KL term. To balance the optimization, the sum over all pixels maintains the magnitude of the data fidelity term against the regularizer.[15]

KL Divergence ($\mathcal{L}_{\text{KL}}$):
This term forces the approximate posterior $q_\phi(\mathbf{z}|x)$ to match the prior $p(\mathbf{z}) = \mathcal{N}(0, I)$.
$$ \mathcal{L}{\text{KL}} = D{\text{KL}}(\mathcal{N}(\mu, \sigma^2) |
| \mathcal{N}(0, I)) = -\frac{1}{2} \sum_{j=1}^{d} (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2) $$
This closed-form solution is computationally efficient and standard for Gaussian VAEs.1

2.5 Analysis of Experiment 1

The expected outcome of Experiment 1 is a model capable of compressing MRI scans and reconstructing them with reasonable visual fidelity. However, the latent space will exhibit significant **entanglement**. Since the objective function ($\text{MSE} + \text{KL}$) contains no incentives to separate factors of variation, the network will encode information in the most compact way possible. A single latent dimension might simultaneously control the tumor's size and its position in the left hemisphere.

This entanglement poses a fatal problem for the downstream Neural ODE. If we define a differential equation on this latent space, the dynamics of "tumor growth" would be mathematically coupled with "tumor location," implying that as a tumor grows, it must also physically translate across the brain in a specific trajectory dictated by the encoder's bias. To rectify this, we must enforce disentanglement.

3. Experiment 2 (Candidate 1): Unsupervised Disentanglement via β-TCVAE and Spatial Broadcasting

The second experimental phase introduces "Candidate 1," an architecture designed to disentangle latent factors without relying on explicit labels. We target two specific forms of entanglement: **spatial entanglement** (where position is mixed with content) and **statistical entanglement** (where independent factors are correlated).

3.1 Architectural Innovation: The Spatial Broadcast Decoder (SBD)

Standard VAE decoders typically use a dense (fully connected) layer to project the latent vector $\mathbf{z}$ into a 3D volume, followed by transposed convolutions. This design is spatially agnostic; the latent vector $\mathbf{z}$ must explicitly encode spatial coordinates to tell the decoder "where" to place features (e.g., "activate the top-left corner"). This forces positional information into $\mathbf{z}$, entangling it with object identity (content).

To decouple "where" from "what," we implement the **Spatial Broadcast Decoder (SBD)**.[1]
Mechanism and Reasoning:

The SBD removes the requirement for $\mathbf{z}$ to carry coordinate information by providing explicit coordinate grids to the decoder.

1. **Broadcasting:** The latent vector $\mathbf{z} \in \mathbb{R}^{B \times d}$ is tiled (replicated) across the spatial dimensions of a tensor with height $H'$, width $W'$, and depth $D'$.
2. **Coordinate Concatenation:** Fixed coordinate tensors representing the $x, y, z$ positions are concatenated to the tiled latents. These coordinates are typically normalized to the range $[-1, 1]$.[18]
3. **Convolutional Decoding:** The resulting tensor is processed by standard convolutional layers (keeping spatial resolution constant or upsampling) to generate the image.

By supplying explicit Cartesian coordinates, the decoder can function as a "rendering engine" that decides *what* to draw based on $\mathbf{z}$ and *where* to draw it based on the coordinate channels. Theoretical and empirical results suggest this architectural bias encourages $\mathbf{z}$ to become translation-invariant, effectively disentangling position from tumor morphology.[1]

PyTorch Implementation Detail:

Python

```
# Assuming z is shape (B, 128)
# Target spatial resolution for broadcast: (8, 8, 8)
z_tiled = z.view(B, 128, 1, 1, 1).expand(-1, -1, 8, 8, 8)

# Generate Coordinate Grids
x = torch.linspace(-1, 1, 8)
y = torch.linspace(-1, 1, 8)
d = torch.linspace(-1, 1, 8)
grid_d, grid_h, grid_w = torch.meshgrid(d, y, x, indexing='ij')

# Stack and Expand
coords = torch.stack([grid_d, grid_h, grid_w], dim=0) # (3, 8, 8, 8)
coords = coords.unsqueeze(0).expand(B, -1, -1, -1, -1) # (B, 3, 8, 8, 8)

# Concatenate: Input to decoder has 128 + 3 = 131 channels
decoder_input = torch.cat([z_tiled, coords], dim=1)
```

Note: The use of indexing='ij' in torch.meshgrid ensures the coordinate axes align correctly with the tensor dimensions $(D, H, W)$.

3.2 Loss Function: $\beta$-TCVAE Decomposition

While SBD addresses spatial entanglement, it does not guarantee that the remaining content factors (e.g., texture vs. shape) are separated. To enforce statistical independence among these factors, we adopt the **β-Total Correlation VAE (β-TCVAE)** objective.[1]

The TC-Decomposed ELBO:

The standard KL divergence term can be mathematically decomposed into three components:
$$ \mathbb{E}{p(x)} = \underbrace{I(x; \mathbf{z})}{\text{Mutual Info}} +
\underbrace{D_{\text{KL}}(q(\mathbf{z}) |
| \prod_j q(z_j))}{\text{Total Correlation}} + \underbrace{\sum_j D{\text{KL}}(q(z_j) |
| p(z_j))}_{\text{Dimension-wise KL}} $$

The critical term is the **Total Correlation (TC)**. Minimizing TC forces the aggregated posterior distribution $q(\mathbf{z}) = \mathbb{E}_{p(x)}[q(\mathbf{z}|x)]$ to factorize, implying that the latent variables $z_1, z_2, \dots$ are statistically independent. The β-TCVAE algorithm weights this term heavily with a hyperparameter $\beta > 1$ (e.g., $\beta \approx 6$) to penalize dependence.[19]

The Minibatch Estimation Challenge:

Calculating the TC term requires evaluating the aggregated posterior $q(\mathbf{z})$, which involves a sum over the entire training dataset—an intractable operation during training. The standard solution is the Minibatch Weighted Sampling (MWS) estimator.6 This estimator approximates $q(\mathbf{z})$ using the samples in the current minibatch.

However, MWS is a biased estimator, and the bias is inversely proportional to the batch size M.[19] In 3D MRI, where memory constraints limit batches to $M=4$ or 8, the TC estimate becomes highly inaccurate. The variance of the gradient estimator explodes, potentially leading to negative KL estimates or unstable training dynamics.[21]

Mitigation Strategies:

1. **Gradient Checkpointing:** To enable larger physical batch sizes (e.g., 16 or 32) on limited VRAM, we employ activation checkpointing (re-computing activations during the backward pass instead of storing them).[23] This trades compute time for memory, allowing for a more representative batch in the TC estimator.
2. **Memory Banks:** A more advanced solution involves maintaining a "memory bank" or queue of latent samples from previous iterations (similar to contrastive learning approaches like MoCo).[24] By using samples from the memory bank to estimate the marginals $q(z_j)$, we effectively increase the sample size for the TC calculation to thousands, significantly reducing bias without increasing the immediate memory footprint of the forward pass.

3.3 Training Schedule and Annealing

Imposing a strong TC penalty ($\beta \gg 1$) early in training can cause **posterior collapse**, where the encoder sets $q(\mathbf{z}|x) \approx p(\mathbf{z})$ to minimize the KL term to zero, effectively ignoring the input data.[1]

To prevent this, we implement **Monotonic or Cyclic Annealing** of the β parameter.

- **Monotonic Schedule:** β starts at 0 and linearly ramps up to its target value (e.g., 6) over a fixed number of epochs (e.g., 20% of training). This allows the reconstruction term to dominate early learning, establishing a meaningful latent space before the independence constraints are tightened.[27]

- **Cyclic Annealing:** We repeat the ramp-up process multiple times during training. This periodically "releases" the regularization pressure, allowing the model to escape local minima where latent dimensions are effectively shut off (KL vanishing).[29] For Candidate 1, a linear warm-up is the minimum requirement.

4. Experiment 3 (Candidate 2): Semi-Supervised Physics-Informed VAE

While Candidate 1 produces independent factors, it does not guarantee **semantic interpretability**. The latent variable $z_1$ might represent "texture," but there is no guarantee it corresponds specifically to "tumor volume." For Neural ODEs modeling physical laws like Gompertzian growth ($dV/dt$), we specifically need a state variable that linearly maps to tumor volume. Candidate 2 enforces this alignment via semi-supervised learning.

4.1 Hybrid Latent Architecture

We partition the 128-dimensional latent space into semantically designated subspaces:

- $\mathbf{z}_{\text{vol}} \in \mathbb{R}^1$: Explicitly reserved for the log-tumor volume.
- $\mathbf{z}_{\text{loc}} \in \mathbb{R}^3$: Reserved for the tumor centroid coordinates $(x, y, z)$.
- $\mathbf{z}_{\text{shape}} \in \mathbb{R}^{k}$: Reserved for morphological features (e.g., sphericity, irregularity).
- $\mathbf{z}_{\text{bg}} \in \mathbb{R}^{128 - 4 - k}$: Unsupervised dimensions capturing background anatomy and scanner variations.

4.2 Loss Function: DIP-VAE + Auxiliary Regression

Candidate 2 employs a multi-task loss function that combines reconstruction, disentanglement regularization, and supervised regression.

1. Unsupervised Regularization: DIP-VAE

For the unsupervised dimensions ($\mathbf{z}_{\text{shape}}, \mathbf{z}_{\text{bg}}$), we employ the Disentangled Inferred Prior (DIP-VAE) objective rather than TCVAE.

- **DIP-VAE Formulation:** This method penalizes the mismatch between the covariance of the aggregated posterior and the identity matrix.[1]
  $$\mathcal{L}_{\text{DIP}} = \lambda_{od} \sum_{i \neq j} [\text{Cov}_{q(\mathbf{z})}(z_i, z_j)]^2 + \lambda_{d} \sum_{i} [\text{Var}_{q(\mathbf{z})}(z_i) - 1]^2$$
- **Reasoning:** DIP-VAE relies on moment matching (covariances) rather than density estimation. Covariance matrices can be estimated more robustly on smaller batches than the full joint density required by TCVAE, making DIP-VAE practically superior for 3D MRI tasks where batch sizes are constrained.[1] We specifically use the **DIP-VAE-II** variant, which penalizes both the off-diagonal (decorrelation) and diagonal (variance unit) terms to ensure the latent space remains standard normal.

2. Supervised Auxiliary Losses:
We utilize the segmentation masks available in datasets like BraTS to compute ground truth scalar labels: tumor volume $v_{gt}$ and centroid $c_{gt}$. We impose regression penalties on the designated latents:

- **Volume Regression:** $\mathcal{L}_{\text{vol}} = \| \mathbf{z}_{\text{vol}} - \log(v_{gt}) \|^2$. (Log-transform is used to handle the exponential scale of tumor sizes).
- **Location Regression:** $\mathcal{L}_{\text{loc}} = \| \mathbf{z}_{\text{loc}} - c_{gt} \|^2$.

Total Loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ELBO}} + \gamma_{\text{sup}} (\mathcal{L}_{\text{vol}} + \mathcal{L}_{\text{loc}}) + \lambda_{\text{DIP}} \mathcal{L}_{\text{DIP}}$$

4.3 Implementation: Balancing Competing Objectives

A critical implementation detail is the weighting $\gamma_{\text{sup}}$. The reconstruction loss $\mathcal{L}_{\text{recon}}$ sums over millions of voxels, resulting in magnitudes of $10^5$ or $10^6$. The regression losses are scalar errors, typically $\approx 10^0$ or $10^{-1}$.

- **Weighting Strategy:** $\gamma_{\text{sup}}$ must be large (e.g., $10^3$ to $10^4$) to be visible to the gradient descent optimizer.
- Uncertainty Weighting: A robust, automated way to tune these weights is to use Homoscedastic Uncertainty weighting.33 The model learns a trainable parameter $\sigma_k$ for each task loss $\mathcal{L}_k$, optimizing:

  $$\mathcal{L}_{\text{total}} = \sum_k \frac{1}{2\sigma_k^2}\mathcal{L}_k + \log \sigma_k$$

  This allows the network to dynamically balance the reconstruction and regression trades-offs during training without manual tuning of $\gamma$.

5. Downstream Integration: The Neural ODE Interface

The ultimate validation of these architectures is their integration with the Neural ODE. Candidate 2 provides a "plug-and-play" state vector for this purpose.

5.1 Physics-Informed Dynamics

We model the tumor growth using a differential equation that incorporates the Gompertz law, a standard model for tumor kinetics 4:

$$\frac{dV}{dt} = \alpha V \ln\left(\frac{K}{V}\right)$$

where $V$ is volume, $\alpha$ is the proliferation rate, and $K$ is the carrying capacity. Using the disentangled VAE, we define the state of the system as $\mathbf{z}(t)$. We construct a Neural ODE where the derivative is defined as:

1. Volume Dynamics: The derivative of the volume latent $\mathbf{z}_{\text{vol}}$ is explicitly constrained to follow the Gompertz form (or a neural approximation thereof).

   $$\frac{d\mathbf{z}_{\text{vol}}}{dt} \approx \text{NeuralNet}_1(\mathbf{z}_{\text{shape}}) \cdot \mathbf{z}_{\text{vol}} \dots$$

   Here, the proliferation rate $\alpha$ is not a fixed constant but is predicted from the shape/texture latents $\mathbf{z}_{\text{shape}}$. This captures the biological intuition that irregular, necrotic tumors (encoded in texture latents) grow differently than spherical, homogenous ones.35
2. Morphological Dynamics: The evolution of the shape latents themselves can be modeled by a separate neural network component, capturing how the tumor's geometry evolves (e.g., becoming more irregular over time):

   $$\frac{d\mathbf{z}_{\text{shape}}}{dt} = \text{NeuralNet}_2(\mathbf{z}_{\text{shape}})$$

5.2 Implementation with torchdiffeq

The integration uses the adjoint sensitivity method to backpropagate through the ODE solver with constant memory cost.[4]

Python

```python
# Pseudo-code for ODE Forward Pass
def forward(self, t, z_state):
    z_vol = z_state[:, 0]
    z_shape = z_state[:, 1:]

    # Predict growth parameters from shape
    alpha, K = self.hyper_net(z_shape)

    # Gompertz dynamics for volume
    d_vol = alpha * z_vol * torch.log(K / z_vol)

    # Neural dynamics for shape
    d_shape = self.shape_ode_net(z_shape)
    return torch.cat([d_vol, d_shape], dim=1)
```

This architecture ensures that the ODE is grounded in physics (Gompertz law) while leveraging the rich, patient-specific phenotypic information extracted by the VAE.

6. Conclusion

This report has structured a rigorous experimental path for developing a 3D MRI VAE suitable for tumor growth modeling. We identified that the primary obstacle is latent entanglement, which decouples the learned representation from the physical reality of tumor dynamics.

- **Experiment 1** established the necessary baseline, emphasizing the critical role of Group Normalization for training stability in memory-constrained 3D environments.
- **Experiment 2 (Candidate 1)** introduced the Spatial Broadcast Decoder and $\beta$-TCVAE to achieve unsupervised disentanglement, solving the specific problem of positional invariance required for robust ODE modeling.
- **Experiment 3 (Candidate 2)** represents the optimal configuration for this domain. By combining DIP-VAE regularization with semi-supervised regression penalties, it forces the latent space to align with the specific state variables (volume, location) required by biological growth laws (Gompertz).

This hybrid approach—using deep learning for feature extraction and differential equations for temporal extrapolation—represents the state-of-the-art in predictive oncology. By implementing the recommended Candidate 2 architecture, researchers can construct a model that is both data-efficient and biologically interpretable, offering a robust tool for clinical forecasting.

Obras citadas

1. Variational Autoencoders for Disentangled 3D MRI Representation – Literature Review.pdf
2. Bridging pharmacology and neural networks: A deep dive into neural ordinary differential equations - PMC - NIH, fecha de acceso: diciembre 17, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11330178/
3. Early Prediction of Overall Survival in Oncology Trials Using Tumor Dynamic Neural-ODE - AI4D3 NeurIPS 2025, fecha de acceso: diciembre 17, 2025, https://ai4d3.github.io/2025/papers/14_Early_Prediction_of_Overall.pdf
4. Adaptive Tumor Growth Forecasting via Neural & Universal ODEs - arXiv, fecha de acceso: diciembre 17, 2025, https://arxiv.org/html/2511.22292v1
5. Explainable deep learning for tumor dynamic modeling and overall survival prediction using Neural-ODE - PMC - NIH, fecha de acceso: diciembre 17, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10657412/
6. Isolating Sources of Disentanglement in Variational Autoencoders - NIPS papers, fecha de acceso: diciembre 17, 2025, http://papers.neurips.cc/paper/7527-isolating-sources-of-disentanglement-in-variational-autoencoders.pdf
7. benniatli/BrainAgePredictionResNet: 3D residual neural network that predicts age

based on brain MRI images (implemented using TensorFlow 2). - GitHub, fecha de acceso: diciembre 17, 2025, https://github.com/benniatli/BrainAgePredictionResNet

8. [1810.11654] 3D MRI brain tumor segmentation using autoencoder regularization - ar5iv, fecha de acceso: diciembre 17, 2025, https://ar5iv.labs.arxiv.org/html/1810.11654
9. 3D ResNet: Volumetric Deep Learning - Emergent Mind, fecha de acceso: diciembre 17, 2025, https://www.emergentmind.com/topics/3d-resnet
10. Group Normalization - CVF Open Access, fecha de acceso: diciembre 17, 2025, https://openaccess.thecvf.com/content_ECCV_2018/papers/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.pdf
11. comparing normalization methods for limited batch size segmentation neural networks - arXiv, fecha de acceso: diciembre 17, 2025, https://arxiv.org/pdf/2011.11559
12. Review — Group Norm (GN): Group Normalization (Image Classification) | by Sik-Ho Tsang, fecha de acceso: diciembre 17, 2025, https://sh-tsang.medium.com/review-group-norm-gn-group-normalization-image-classification-5f7fe0f58eb6
13. Comparative Analysis of Brain Tumor Image Segmentation Performance of 2D U-Net and 3D U-Nets with Alternative Normalization Methods - JMIS(Journal of Multimedia Information System), fecha de acceso: diciembre 17, 2025, https://www.jmis.org/archive/view_article?pid=jmis-11-2-157
14. Comparing Normalization Methods for Limited Batch Size Segmentation Neural Networks, fecha de acceso: diciembre 17, 2025, https://www.researchgate.net/publication/346143341_Comparing_Normalization_Methods_for_Limited_Batch_Size_Segmentation_Neural_Networks
15. Correct implementation of VAE loss - PyTorch Forums, fecha de acceso: diciembre 17, 2025, https://discuss.pytorch.org/t/correct-implementation-of-vae-loss/146750
16. [1901.07017] Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs - arXiv, fecha de acceso: diciembre 17, 2025, https://arxiv.org/abs/1901.07017
17. Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs. - SciSpace, fecha de acceso: diciembre 17, 2025, https://scispace.com/pdf/spatial-broadcast-decoder-a-simple-architecture-for-learning-33fx4sq7h4.pdf
18. [1901.07017] Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs - ar5iv, fecha de acceso: diciembre 17, 2025, https://ar5iv.labs.arxiv.org/html/1901.07017
19. Isolating Sources of Disentanglement in Variational Autoencoders - NIPS papers, fecha de acceso: diciembre 17, 2025, https://papers.nips.cc/paper/7527-isolating-sources-of-disentanglement-in-variational-autoencoders
20. [1912.05184] Variational Learning with Disentanglement-PyTorch - arXiv, fecha de acceso: diciembre 17, 2025, https://arxiv.org/abs/1912.05184

21. negetive total correlation loss for btc-vae · Issue #54 · YannDubs/disentangling-vae - GitHub, fecha de acceso: diciembre 17, 2025, https://github.com/YannDubs/disentangling-vae/issues/54
22. How to estimate total correlation KL[q(z)||Πjq(zj)] of VAE after training (useful for latents disentanglement evaluation), fecha de acceso: diciembre 17, 2025, https://datascience.stackexchange.com/questions/63976/how-to-estimate-total-correlation-klqz%CE%A0jqzj-of-vae-after-training-usefu
23. Memory efficient model based deep learning reconstructions for high spatial resolution 3D non-cartesian acquisitions - NIH, fecha de acceso: diciembre 17, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10034748/
24. A Gradient Accumulation Method for Dense Retriever under Memory Constraint - NIPS papers, fecha de acceso: diciembre 17, 2025, https://proceedings.neurips.cc/paper_files/paper/2024/file/15ba84c1e19b0eb75f96922f5da0a021-Paper-Conference.pdf
25. 16-726 Project - andrew.cmu.ed, fecha de acceso: diciembre 17, 2025, https://www.andrew.cmu.edu/course/16-726/projects/zijieli/project/
26. Less pain, more gain: A simple method for VAE training with less of that KL-vanishing agony, fecha de acceso: diciembre 17, 2025, https://www.microsoft.com/en-us/research/blog/less-pain-more-gain-a-simple-method-for-vae-training-with-less-of-that-kl-vanishing-agony/
27. Towards Latent Space Disentanglement of Variational AutoEncoders for Language - uu .diva, fecha de acceso: diciembre 17, 2025, https://uu.diva-portal.org/smash/get/diva2:1682344/FULLTEXT01.pdf
28. Learning Dynamics in Linear VAE: Posterior Collapse Threshold, Superfluous Latent Space Pitfalls, and Speedup with KL Annealing, fecha de acceso: diciembre 17, 2025, https://proceedings.mlr.press/v238/ichikawa24a/ichikawa24a.pdf
29. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing - GitHub, fecha de acceso: diciembre 17, 2025, https://github.com/haofuml/cyclical_annealing
30. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing | Request PDF, fecha de acceso: diciembre 17, 2025, https://www.researchgate.net/publication/331987117_Cyclical_Annealing_Schedule_A_Simple_Approach_to_Mitigating_KL_Vanishing
31. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, fecha de acceso: diciembre 17, 2025, https://openreview.net/forum?id=H1kG7GZAW
32. Isolating Sources of Disentanglement in Variational Autoencoders - Ricky T. Q. Chen, fecha de acceso: diciembre 17, 2025, https://rtqichen.github.io/posters/tcvae_poster.pdf
33. Balance between 2 MSE losses - PyTorch Forums, fecha de acceso: diciembre 17, 2025, https://discuss.pytorch.org/t/balance-between-2-mse-losses/185315
34. Population modeling of tumor growth curves and the reduced Gompertz model improve prediction of the age of experimental tumors - NIH, fecha de acceso: diciembre 17, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC7059968/
35. (PDF) Adaptive tumor growth forecasting via neural & universal ODEs -

ResearchGate, fecha de acceso: diciembre 17, 2025, https://www.researchgate.net/publication/398134446_Adaptive_tumor_growth_forecasting_via_neural_universal_ODEs