A Physics-Informed, Oscillatory Network Strategy for Aneurysm Detection: A Research and Implementation Guide

Part I: Foundational Assets and Pre-training Strategy

This section details the critical preparatory steps for developing the proposed aneurysm detection pipeline. It provides a definitive strategy for pre-training the core feature extractor, establishing the quality requirements for auxiliary models, and curating a high-impact pre-training data corpus.

1. Pre-training the WaveFormer Backbone

The performance of the entire detection pipeline is contingent upon the quality of the feature representations learned by the WaveFormer backbone. A robust self-supervised learning (SSL) pre-training strategy is therefore not an optional enhancement but a foundational requirement. This sub-section evaluates the leading SSL paradigms and recommends an optimal approach tailored to the specific challenges of 3D medical imaging and the constraints of the competition.

1.1 A Definitive Comparison: SparK vs. Contrastive Learning for 3D Medical Pre-training

The choice between generative and contrastive SSL methods is a pivotal strategic decision. Generative methods, such as Masked Autoencoders (MAE) and their derivatives like SparK, learn by reconstructing masked or corrupted portions of an input. Contrastive methods, such as MoCo or SimCLR, learn by maximizing the similarity between different augmented views of the same image while minimizing similarity with other images.

Analysis of Methodologies:

- **Contrastive Learning:** This paradigm has demonstrated significant success in learning powerful, global semantic representations by enforcing instance-level discrimination.[1] The core strength lies in its ability to learn features that are invariant to a range of data augmentations, which is effective for high-level classification tasks.[3] However, this strength is also a potential vulnerability. The performance of contrastive methods is heavily dependent on the design of these augmentations, which must be carefully tailored to create meaningful positive and negative pairs.[1] In the context of medical imaging, where instance diversity can be limited and fine-grained structural details are paramount, contrastive approaches may struggle to capture the subtle anatomical features necessary for aneurysm detection.[3]

- **Generative Learning (SparK):** Generative methods, particularly those based on masked image modeling (MIM), operate on a more fundamental principle: learning a model of the data's structure by predicting missing information. SparK successfully adapts the MAE framework, originally designed for Vision Transformers (ViTs), to convolutional and hybrid architectures by employing sparse convolutions to efficiently process only the unmasked regions of the input.[4] This reconstruction-based pretext task compels the model to learn rich local and contextual information, which is crucial for understanding the fine-grained anatomical structures present in medical scans.[3]

Robustness in Data-Scarce Scenarios:

A crucial differentiator for this project is the robustness of the pre-training method to the size and characteristics of the downstream, fine-tuning dataset. The RSNA challenge dataset, while substantial, may represent a data-scarce environment for rare aneurysm subtypes. Research comparing SparK with leading contrastive methods (BYOL, MoCoV2, SwAV) on CT classification tasks provides a clear verdict in this regard.[5] While all SSL methods outperform training from scratch, SparK demonstrates superior robustness when the fine-tuning dataset is small. One study found that SparK's performance remained constant even when the downstream training data was reduced by 50%, whereas the performance of top contrastive methods degraded significantly under the same conditions.[5] The conclusion was that for tasks with fewer than approximately 150 samples per class, SparK is the more reliable and higher-performing choice.[5]

This empirical evidence is strategically vital. The unseen private test set in the competition is, by definition, a scenario of data scarcity and potential domain shift. A pre-training method that is less sensitive to the specific distribution and size of the fine-tuning data is inherently lower-risk and more likely to generalize well. SparK's focus on learning fundamental local structures provides a more resilient foundation than a contrastive approach that relies on discriminating between specific instances and augmentations.

Definitive Recommendation:

The WaveFormer backbone should be pre-trained using the **SparK** methodology. Its proven ability to learn fine-grained anatomical features and, most critically, its superior robustness in data-scarce fine-tuning scenarios make it the optimal choice for de-risking the project and maximizing the potential for generalization to the unseen competition test set.

Table 1: Comparative Analysis of SSL Pre-training Methods

Method	Representation Type	Primary Strength	Primary Weakness	Robustness to Data Scarcity	Recommendation for Project
Contrastive Learning	Global / Instance-level	Strong semantic discrimination for high-level tasks [3]	Relies on augmentations; struggles with fine-grained detail [1]	Lower	Not Recommended as primary
SparK (Generative)	**Local / Contextual**	**Learns fine-grained structure; robust to data scarcity** [5]	**May capture less global semantic meaning than contrastive methods**	**Higher**	**Strongly Recommended**
Hybrid (e.g., CM-UNet)	Balanced Global/Local	Combines strengths of both paradigms [3]	Increased implementation and training complexity	High	Viable but more complex alternative

1.2 State-of-the-Art 3D Masking Strategies for Anatomically-Aware Representation Learning

The effectiveness of a MIM-based pre-training approach like SparK is highly dependent on the masking strategy. For 3D medical volumes, which are often characterized by a large proportion of background or uniform tissue, a simple random masking strategy is computationally inefficient and can lead to suboptimal representations.[1] The model expends significant capacity learning to reconstruct irrelevant or low-information regions. State-of-the-art research has therefore moved towards more intelligent, anatomy-aware masking techniques.

Analysis of Advanced Masking Strategies:

- **Anatomy-Guided Masking (AnatoMask):** This approach refines random masking by dynamically identifying and prioritizing anatomically significant regions for the reconstruction task.[1] The core innovation is a self-distillation framework where a "teacher" network identifies regions with high reconstruction loss, which are assumed to be anatomically complex.[9] These high-loss regions are preferentially masked for a "student" network to reconstruct. To prevent the model from focusing on overly difficult regions too early, a "masking dynamics function" implements an easy-to-hard curriculum, gradually increasing the proportion of anatomy-guided masks as training progresses.[1] This strategy has been shown to improve pre-training efficiency and downstream performance compared to standard random masking.[1]
- **Hierarchical Masking (MiM: Mask in Mask):** This framework introduces a more profound architectural change to the masking process itself, designed to explicitly model the hierarchical nature of anatomical structures.[10] Instead of a single-level mask, MiM creates multi-level masked inputs. For instance, a coarse mask is applied at Level 1. The masked regions from Level 1 then form the input volume for Level 2, where a finer-grained mask is applied.[11] The model is tasked with reconstructing the masked tokens at all levels simultaneously. Furthermore, a cross-level alignment loss, typically a contrastive loss, is used to enforce semantic consistency between features from adjacent scales.[10] This entire process forces the model to learn not only fine-grained details but also their relationship to the broader anatomical context from which they were derived.[11]

Strategic Rationale for Hierarchical Context:

The detection of an intracranial aneurysm is an intrinsically multi-scale problem. It requires the identification of a small, local morphological anomaly (the aneurysm sac) and the simultaneous understanding of its global topological context within the cerebrovascular tree (e.g., its location at a bifurcation of a major artery). While AnatoMask improves the *semantic content* of the masked regions, its single-scale approach does not explicitly teach the model these crucial multi-scale relationships.

The MiM framework, by its very design, provides a powerful inductive bias for learning about tree-like, hierarchical structures like the vasculature. The nested masking scheme and cross-level alignment loss compel the model to understand how a small, detailed structure (e.g., a distal vessel segment) relates to the larger, coarser structure it is a part of (e.g., a proximal parent artery). This pre-training objective is perfectly aligned with the downstream task of identifying a localized anomaly within a complex, branching system.

Definitive Recommendation:

Implement the **Mask in Mask (MiM)** hierarchical masking framework for pre-training the WaveFormer backbone. Its ability to explicitly model multi-scale anatomical relationships provides a superior inductive bias for learning vascular features compared to single-scale anatomy-guided methods. This approach will equip the backbone with a more profound understanding of both local vessel morphology and global vascular topology, which is essential for accurate aneurysm detection.

1.3 Curating a High-Impact Pre-training Corpus from Public Datasets

The success of any SSL strategy is directly proportional to the scale and diversity of the unlabeled pre-training data. Assembling such a corpus from disparate public sources can be a significant data engineering challenge, involving discovery, access, and, most critically, harmonization. The strategy must therefore prioritize high-volume, high-quality, and low-friction data sources.

Survey of Major Public Data Repositories:

- **The OpenMind Dataset (from the SSL3D Challenge):** This is the most significant and strategically important resource identified. It is a massive, pre-curated aggregation of over 800 smaller datasets from the OpenNeuro platform.[13] Comprising 114,570 3D MRI volumes from over 34,000 patients, it includes a wide variety of sequences such as T1w, T2w, and FLAIR.[13] Crucially, the data has already been cleaned and standardized, which dramatically reduces the data engineering overhead and makes it an ideal foundational component for the pre-training corpus.[14]
- **The Cancer Imaging Archive (TCIA):** TCIA is a premier repository for oncology imaging data and contains several collections relevant to head and neck anatomy.[16] Notable collections include
  HNSCC (Head and Neck Squamous Cell Carcinoma), with 627 subjects featuring both CT and MR scans, and **HEAD-NECK-PET-CT**, with 298 subjects providing CT and PET data.[17] While invaluable for adding CT and CTA modality diversity, these datasets often have "Limited" access protocols and require dedicated efforts for standardization to match other data sources.[19]

- **DeepLesion:** This NIH dataset provides over 32,000 annotated lesions from 4,400 patients, all derived from CT scans.[20] While not specific to vascular anatomy, its sheer scale and diversity of radiological findings make it an excellent resource for teaching a model general-purpose feature extraction on CT images.
- **Other Repositories:** The Stanford AIMI Center shares several relevant datasets, including SinoCT (head CTs).[21] OpenNeuro itself is the source for OpenMind and contains a vast number of raw datasets, though navigating and processing them individually is a major undertaking.[22]

A Strategic Approach to Corpus Curation:

A pragmatic strategy is to adopt a tiered approach, prioritizing pre-processed, large-scale datasets to form a "core" corpus, which can then be augmented with more specialized datasets. The OpenMind dataset, due to its scale and pre-processed nature, is the clear choice for the core MRI component. The primary effort should then be focused on sourcing and standardizing the largest available CT/CTA datasets to complement it.

Definitive Recommendation:

The pre-training corpus should be constructed as follows:

1. **Core MRI Corpus:** Utilize the **OpenMind** dataset as the foundational source for all MRI modalities (T1, T2, FLAIR). Its massive scale and pre-standardized format provide the highest return on investment.
2. **Core CT/CTA Corpus:** Augment the MRI data with the **TCIA HNSCC** collection.[17] This is the largest and most relevant TCIA collection containing both CT and MR modalities, providing a strong foundation for CT-based pre-training.
3. **Supplemental Diversity:** If project timelines and resources permit, further supplement the corpus with the **TCIA HEAD-NECK-PET-CT** collection [18] to add modality diversity and the
   DeepLesion dataset [20] to improve the model's ability to represent a wide range of radiological abnormalities. All supplemental data must be harmonized (resampled, intensity-normalized, and converted to a consistent format like NIfTI) to match the specifications of the core corpus.

Table 2: Recommended Public Datasets for Pre-training Corpus

Dataset Name	Primary Modalities	Scale (Subjects/Volumes)	Key Characteristics	Strategic Role
OpenMind (SSL3D)	MRI (T1w, T2w, FLAIR, etc.)	34k / 114k	Pre-curated, standardized, massive scale [13]	**Core MRI Corpus**
TCIA HNSCC	CT, MR	627 / >1k	Real clinical data with RT structures [17]	**Core CT/CTA Corpus**
TCIA HEAD-NECK-PET-CT	CT, PT	298 / >500	Adds PET modality for diversity [18]	**Supplemental CT Corpus**
DeepLesion	CT	4.4k / 32k lesions	High diversity of radiological findings [20]	**Supplemental Feature Diversity**

2. Auxiliary Vessel Segmentation Model

The Stage 2 Expert Classifier operates on regions of interest defined by a vessel mask. This mask is generated by an auxiliary U-Net model. The quality of this segmentation is not an end in itself but is a critical factor influencing the performance of the entire downstream pipeline. Therefore, it is essential to understand the sensitivity of the Stage 2 model to imperfections in this mask to define an appropriate accuracy target.

2.1 Sensitivity Analysis: Defining Accuracy Requirements and Impact on Downstream Performance

The relationship between segmentation accuracy and the performance of a subsequent classification task is complex and non-linear. Standard segmentation metrics like the Dice

Similarity Coefficient (DSC), which measures spatial overlap, do not always correlate directly with downstream task success.[23] For instance, a study on virtual diffusion sequences found that despite low average Dice scores (0.43), the scan-level classification accuracy for detecting acute infarcts was reasonably high (72%).[24] This suggests that as long as the clinically salient features are captured, perfect voxel-wise overlap is not a strict prerequisite.

However, the introduction of segmentation errors can also be highly detrimental. Using synthetic data containing artifacts for training a segmentation model can be viewed as a form of "data poisoning," leading to significant performance degradation in downstream tasks.[25] This highlights that the

type of error matters as much as its magnitude.

In the context of this two-stage pipeline, the vessel mask serves two functions: it defines the input region for the Expert Classifier and the computational domain for the PINN's physics loss. This dual role creates a profound and asymmetric sensitivity to segmentation errors.

- **Impact of False Negatives (Missed Vessels):** A false negative, where the auxiliary U-Net fails to segment a vessel, is a catastrophic failure. If a small, aneurysm-bearing vessel is missed, that entire region is excluded from the input to the Stage 2 model. The classifier will never see the anomaly, and the PINN loss will not be calculated in that domain. The probability of detecting that aneurysm drops to zero. This represents an unrecoverable, mission-critical error that propagates through the entire system.
- **Impact of False Positives (Over-segmentation):** A false positive, where the mask is slightly larger than the actual vessel, is a suboptimal but non-catastrophic error. The classifier receives the true vessel plus a cuff of surrounding non-vessel tissue. This introduces noise into the feature extraction process but does not eliminate the signal. Similarly, the PINN will attempt to enforce the Navier-Stokes equations on static tissue, which will result in a non-zero residual and add a small, potentially noisy penalty to the overall loss. While this will degrade performance, the model still has the opportunity to make a correct classification.

This analysis reveals that the system's performance is far more sensitive to recall (minimizing false negatives) than to precision (minimizing false positives). The primary requirement for the auxiliary segmentation model is not to achieve a perfect Dice score, but to ensure near-complete capture of all potential vascular structures, no matter how small.

Definitive Recommendation:

- **Required Accuracy Level:** The primary target for the auxiliary U-Net is a **vessel recall (sensitivity) of >98%**. A Dice score of approximately 0.90 is a reasonable secondary target, but it should not be achieved at the expense of recall.
- **Sensitivity Profile:** The Stage 2 model is **critically sensitive** to false negatives in the vessel mask and only **moderately sensitive** to false positives.
- Implementation Strategy: To align the training objective with this sensitivity profile, the auxiliary U-Net should be trained using a loss function that heavily penalizes false negatives. The Tversky loss is an excellent candidate. It is a generalization of the Dice loss that allows for weighting false positives and false negatives differently. The Tversky index (TI) is given by:

  $TI(\alpha,\beta)=TP+\alpha FN+\beta FPTP$

  To prioritize recall, the parameter α should be set higher than β (e.g., $\alpha=0.7,\beta=0.3$). Training the U-Net to minimize $1-TI$ will explicitly optimize the model to avoid missing vessel structures, thereby mitigating the most critical failure mode for the downstream classifier.

Part II: Stage 1 Implementation Strategy: The "Smart Net"

The first stage of the pipeline, the "Smart Net," is responsible for generating high-quality candidate regions. This stage moves beyond simple blob detection by integrating SegFormer3D with Vessel Graph Network (VGN) principles to perform a topologically-aware analysis, identifying regions that are not just dense but structurally anomalous.

3. Integrating Vessel Graph Networks with SegFormer3D

The core task of this stage is to translate the dense, voxel-based probability map produced by the SegFormer3D model into a sparse, structured graph representation of the vasculature. This graph can then be analyzed for topological irregularities indicative of an aneurysm.

3.1 From Voxel to Vertex: A Practical Guide to Graph Construction and Topological Anomaly Detection

The conversion from a 3D image volume to a mathematical graph is a multi-step process involving segmentation, skeletonization, and graph extraction.[26] Once the graph is constructed, Graph Neural Networks (GNNs) provide a powerful framework for learning its properties and detecting anomalies.[27]

Implementation Pipeline:

1. **Probability Map to Binary Mask:** The initial output from the SegFormer3D model is a 3D tensor where each voxel value represents the probability of it being part of a vessel. This continuous probability map must be converted into a discrete binary mask. This is achieved by applying a confidence threshold. A relatively high threshold (e.g., probability > 0.8) should be used to ensure that the initial mask represents the high-confidence core of the vasculature, minimizing noise that could corrupt the subsequent topological analysis.
2. **Binary Mask to Skeleton:** The next step is to extract the one-dimensional centerline, or skeleton, of the 3D vessel mask. This process, known as skeletonization or thinning, reduces the volumetric representation to a 1-voxel-thick line structure that preserves the essential connectivity and topology of the vessel tree.[28] This is a critical step, as the skeleton forms the direct basis for the graph. Robust 3D morphological thinning algorithms, such as the 3D implementation of the Zhang-Suen algorithm available in libraries like
scikit-image (skimage.morphology.skeletonize_3d), are well-suited for this task.
3. **Skeleton to Graph Construction:** The resulting 3D skeleton is traversed to identify key topological points, which are then converted into the nodes and edges of a graph:

- **Nodes:** Graph nodes are defined at critical points in the skeleton. These are primarily **junction points** (voxels with more than two neighboring skeleton voxels) and **end points** (voxels with only one neighbor).[26]
- **Edges:** The continuous paths of skeleton voxels that connect any two nodes are defined as the graph edges.
- **Feature Extraction:** To enrich the graph representation, features should be extracted for both nodes and edges. A distance transform can be applied to the original binary mask to efficiently calculate the vessel radius at any point.

- **Node Features:** Key features for each node should include its 3D coordinates, the number of connected edges (node degree), and the average vessel radius in the immediate vicinity of the node.
- **Edge Features:** Key features for each edge should include its length (in voxels or millimeters), an estimate of its average curvature, and the average vessel radius along its path.

Topological Anomaly Detection with Graph Attention Networks (GATs):

With the vessel network represented as an attributed graph, the task is to identify nodes or subgraphs that represent an aneurysm. An aneurysm is a distinct topological anomaly—a localized, abnormal expansion that violates the expected geometric and connectivity patterns of a healthy vascular tree. While standard GNNs are effective for learning on graphs, Graph Attention Networks (GATs) are particularly well-suited for this anomaly detection task.[29]

A standard Graph Convolutional Network (GCN) updates a node's representation by aggregating features from its neighbors with fixed, pre-defined weights (often based on node degree). This isotropic aggregation can "smooth over" the features of a localized anomaly, making it harder to detect. In contrast, a GAT introduces an attention mechanism, allowing the model to learn to assign different importance weights to different neighbors during the aggregation process.[30]

This is a powerful advantage for aneurysm detection. A GAT can learn the complex, context-dependent rules of healthy vascular topology. For example, it can learn that at a normal bifurcation node, its two child vessel segments are highly important neighbors. If an aneurysm is present, it might appear as a third, highly unusual neighbor (e.g., a node with a very high radius and connected by a very short edge). The GAT's attention mechanism can learn to assign a very high attention weight to this anomalous neighbor, causing the resulting node embedding to be significantly different from that of a healthy bifurcation. This makes the anomalous node easily separable in the feature space.

Definitive Recommendation:

The final step of the Smart Net should employ a **Graph Attention Network (GAT)** trained for node-level anomaly detection. The GAT will process the constructed vessel graphs, and its output will be an anomaly score for each node. Nodes (and their corresponding 3D regions) that exceed a certain anomaly score threshold will be selected as the final candidate proposals to be passed to the Stage 2 Expert Classifier. This approach leverages the GAT's ability to learn context-aware topological rules, making it a highly effective method for identifying structurally anomalous regions that are likely to be aneurysms.

Part III: Stage 2 Implementation Strategy: The "Expert Classifier"

The second stage of the pipeline, the "Expert Classifier," is designed to perform the final, high-precision classification of the candidate regions generated by the Smart Net. This stage employs a sophisticated hybrid architecture that integrates novel components—Artificial Kuramoto Oscillatory Neurons (AKOrN) and a Physics-Informed Neural Network (PINN) head—to move beyond simple pattern recognition and incorporate principles from dynamical systems and fluid dynamics.

4. Architectural Integration of Artificial Kuramoto Oscillatory Neuron (AKOrN) Blocks

AKOrN units are a novel alternative to standard activation functions like ReLU or GeLU. They are based on the Kuramoto model of coupled oscillators, a mathematical model that describes synchronization phenomena in physical systems.[31] By replacing static activation units with these dynamic, interacting oscillators, the network can learn to "bind" related features together through synchronization, potentially leading to more robust and abstract representations.[32]

4.1 Mathematical Formulation and Spatial Data Application

The AKOrN model generalizes the original Kuramoto model to operate on multi-dimensional vectors on a hypersphere. Each oscillator, xi, is an N-dimensional unit vector, i.e., $xi \in RN$ with $\|xi\|2=1$, where N is the oscillator_dim. The dynamics of these oscillators are governed by the following differential equation [31]:

$\dot{x}i=\Omega ixi+Projxi(ci+j\sum Jijxj)$

where:

- $\dot{x}i$ is the time derivative of the oscillator state xi.
- Ωi is an N×N anti-symmetric matrix representing the oscillator's **natural frequency**, which dictates its intrinsic rotation.
- $ci \in RN$ is a **conditional stimulus** vector derived from the input data or the previous layer's activations. It acts as an external field, biasing the oscillator's phase.

- Jij represents the **coupling strength** between oscillator i and oscillator j. These are the learnable weights of the network layer (e.g., the weights of a convolutional kernel or an attention matrix).
- Projxi(y)=y−⟨y,xi⟩xi is a projection operator that maps a vector y onto the tangent space of the sphere at point xi. This ensures that the update step keeps the oscillator on the unit sphere.

For practical implementation in a neural network, this differential equation is discretized and solved iteratively for a fixed number of steps, T.

When applied to **spatial image data**, the oscillators xi and stimuli ci are arranged on a grid, indexed by channel, height, and width, e.g., xc,h,w. The coupling term ∑jJijxj is implemented as a standard convolution. The update direction for an oscillator at position (c,h,w) is computed by convolving the oscillator states X with a learnable kernel J, adding the conditional stimuli C, and then applying the projection and update steps.[31] This allows the AKOrN layer to learn spatio-temporal patterns, with evidence showing that shallow layers learn high-frequency local features while deeper layers generate more global, low-frequency waves of synchronization.[32]

4.2 Optimal Architectural Placement within a 3D Transformer Layer

A standard Transformer block consists of a Multi-Head Self-Attention (MSA) module followed by a feed-forward Multi-Layer Perceptron (MLP), with layer normalization and residual connections applied to each. The leading hypothesis is to place the AKOrN block in "Sequential Augmentation" after the MLP block. This placement is strategically optimal.

The MSA module's role is to aggregate information across all spatial locations, allowing each token to gather global context. The subsequent MLP block then acts as a feature transformation step, processing each token's aggregated information independently to learn richer, channel-wise representations.

Placing the AKOrN block *after* the MLP allows the oscillatory dynamics to operate on these highly processed, context-aware feature vectors. The synchronization and binding process of the Kuramoto dynamics can then group and refine these abstract features. For example, features corresponding to a vessel wall and features corresponding to turbulent flow within it, having been brought together by MSA and refined by the MLP, can be "bound" into a coherent representation of an aneurysm by the AKOrN block. This bound representation is then passed to the next Transformer layer's MSA module. This creates a powerful processing cycle within each layer: **global spatial mixing (MSA) -> per-token feature transformation (MLP) -> dynamic**

feature binding (AKOrN). Placing it before the MLP would mean the oscillators operate on less refined features, and placing it within the MSA block would overly complicate the attention mechanism itself.

4.3 Data Conversion and Hyperparameters

The AKOrN layer operates on a set of oscillators. When integrated into a convolutional or transformer architecture processing a 3D feature map of shape (Batch, Channels, Depth, Height, Width), this feature map must be converted into a 1D sequence of oscillator states. A naive flattening would result in an extremely long sequence, making the computation of all-to-all interactions (as in a fully-connected or attentive AKOrN) or even large-kernel convolutions computationally infeasible, especially given the 16-24GB VRAM constraint.

Optimal Data Conversion Method:

To manage computational complexity, an **Adaptive 3D Pooling** layer should be applied to the input feature map before the AKOrN block. For example, an adaptive average pooling layer can downsample the spatial dimensions (Depth, Height, Width) to a fixed, manageable size (e.g., 8×8×8). The resulting feature map is then flattened into a sequence of 8×8×8=512 tokens. This drastically reduces the number of oscillators the Kuramoto simulation must handle, making it tractable within VRAM limits while preserving the essential spatial information from the feature map.

Hyperparameter Tuning: oscillator_dim (N):

The oscillator_dim hyperparameter, which corresponds to the dimension N of each oscillator vector xi, is a critical parameter with a significant trade-off. It defines the richness of the phase space in which each neuron can operate.

- **Risk of Small N:** If N is too small (e.g., N=2, the original Kuramoto model), the oscillator's state is limited to a simple phase angle on a circle. This may be insufficient to represent the complex phase relationships and features needed for medical image analysis.
- **Risk of Large N:** If N is too large, the computational cost of the vector operations (projections, matrix-vector products) within the Kuramoto update increases significantly. This directly impacts both training time and VRAM consumption. Furthermore, a very high-dimensional phase space can make the optimization landscape more complex and potentially increase the risk of overfitting.

Tuning Approach:

Given the VRAM constraints, the tuning of oscillator_dim should be approached conservatively. Start with a small value, such as **N=4 or N=8**. Monitor the model's performance and VRAM usage. Incrementally increase N and observe the marginal gain in the validation AUC versus the increase in computational cost. It is likely that a point of diminishing returns will be reached where a larger oscillator_dim provides negligible performance improvement while consuming significantly more resources. This empirical approach will find the optimal balance for the competition environment.

5. Hemodynamic Regularization with a Physics-Informed Neural Network (PINN) Head

The integration of a PINN as an auxiliary head is a core innovation of this project, designed to regularize the model by ensuring its predictions are consistent with the physical laws of fluid dynamics. A PINN embeds a Partial Differential Equation (PDE) into the loss function of a neural network. The network is trained not only to fit data but also to minimize the PDE residual—the degree to which its output violates the governing physical equations.

5.1 A Practical Framework for Mitigating PINN Training Failures

Training PINNs is notoriously challenging. A common "failure mode" occurs when the PDE residual loss converges to a low value, but the learned solution is physically incorrect and has a large error compared to the ground truth.[34] Historically, this has been attributed to the optimizer getting trapped in poor local minima separated by steep loss barriers.[34] However, recent state-of-the-art research has provided a more fundamental understanding and, consequently, more effective mitigation strategies.

Concrete Mitigation Steps:

1. **Utilize Double-Precision (FP64) Arithmetic:** The most critical step to mitigate failure modes is to perform all PINN-related computations using 64-bit floating-point precision. Groundbreaking recent work has demonstrated that many so-called failure modes are not inescapable local minima but are "precision-induced stalls".[35] With standard 32-bit precision (FP32), second-order optimizers like L-BFGS can prematurely satisfy their convergence criteria due to floating-point inaccuracies in gradient calculations, causing training to halt in a "failure phase".[35] Simply switching to FP64 provides the necessary numerical precision for the optimizer to navigate these regions and continue converging

towards the correct solution. This single change has been shown to enable vanilla PINNs to solve problems that were previously considered to exhibit failure modes, outperforming more complex, specialized architectures.[34]

2. **Implement an Adaptive Sampling Strategy:** Uniformly sampling collocation points within the domain is inefficient. The PDE residual is often highly non-uniform, with large errors concentrated in specific regions (e.g., areas of high curvature or turbulence).[38] An adaptive sampling strategy focuses computational effort where it is most needed. The **Retain-Resample-Release (R3) algorithm** is a state-of-the-art approach for this.[38] The R3 method works by:

- **Retaining:** Keeping a subset of collocation points from the previous iteration that have the highest PDE residuals.
- **Resampling:** Replacing the points with low residuals with new points sampled uniformly from the domain.
- Releasing: Automatically dropping points from the "retained" set once the model learns to resolve the high residuals in that area.
  This dynamically concentrates sampling in problematic regions without incurring significant computational overhead.38

3. **Employ Curriculum Learning:** For particularly stiff or complex problems, a curriculum learning strategy can stabilize training.[40] This involves starting with a simpler version of the physics loss and gradually increasing its complexity. For the Navier-Stokes equations, this could mean:

- Starting with a low Reynolds number (viscosity-dominated flow) and gradually increasing it towards the physically realistic value.
- Initially training only on the boundary conditions and the continuity equation, and then introducing the momentum equation terms into the loss function after a certain number of epochs.

By combining these three concrete steps—FP64 precision as the foundation, R3 sampling for efficiency, and curriculum learning for stability—the risk of PINN training failure can be substantially mitigated.

5.2 Implementation of the 3D Navier-Stokes Residual Loss in PyTorch

The physics loss is calculated from the residuals of the incompressible Navier-Stokes equations. For a steady-state, incompressible, Newtonian fluid, these equations consist of the continuity equation (conservation of mass) and the momentum equations (conservation of momentum).

The network, fθ, takes spatial coordinates (x,y,z) as input and outputs the velocity vector field (u,v,w) and the pressure field p. The physics loss, Lphysics, is the mean squared error of the PDE residuals over a set of collocation points {(xi,yi,zi)}i=1Nc within the vessel mask.

Continuity Equation (Mass Conservation):

$$ R_{cont} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 $$

Momentum Equations (assuming zero body forces):

$$ R_u = \rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) $$

$$ R_v = \rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) $$

$$ R_w = \rho \left(u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) $$

where ρ is the fluid density and μ is the dynamic viscosity (these are physical constants for blood).

The following PyTorch code snippet demonstrates how to calculate these residuals using torch.autograd.grad to compute the necessary high-order derivatives.

Python

```
import torch

def compute_navier_stokes_residuals(network_output, coords, rho=1060.0, mu=0.0035):
    """
    Computes the residuals of the 3D incompressible Navier-Stokes equations.
    Args:
        network_output (torch.Tensor): Tensor of shape (N, 4) containing [u, v, w, p].
        coords (torch.Tensor): Tensor of shape (N, 3) containing [x, y, z] coordinates.
                               Requires grad to be True.
        rho (float): Fluid density (e.g., for blood, in kg/m^3).
        mu (float): Dynamic viscosity (e.g., for blood, in Pa*s).

    Returns:
        Tuple: Residuals for continuity, u-momentum, v-momentum, w-momentum.
    """
    # Unpack network outputs
    u = network_output[:, 0:1]
    v = network_output[:, 1:2]
    w = network_output[:, 2:3]
    p = network_output[:, 3:4]

    # --- Compute First-Order Derivatives ---
    # We use torch.autograd.grad to compute gradients of outputs with respect to inputs.
    # create_graph=True is essential to allow for computing higher-order derivatives (gradients of
gradients).

    # Gradient of u with respect to (x, y, z)
    grad_u = torch.autograd.grad(u, coords, grad_outputs=torch.ones_like(u),
create_graph=True)
    u_x = grad_u[:, 0:1]
    u_y = grad_u[:, 1:2]
    u_z = grad_u[:, 2:3]

    # Gradient of v with respect to (x, y, z)
    grad_v = torch.autograd.grad(v, coords, grad_outputs=torch.ones_like(v), create_graph=True)
    v_x = grad_v[:, 0:1]
    v_y = grad_v[:, 1:2]
    v_z = grad_v[:, 2:3]

    # Gradient of w with respect to (x, y, z)
    grad_w = torch.autograd.grad(w, coords, grad_outputs=torch.ones_like(w),
create_graph=True)
    w_x = grad_w[:, 0:1]
    w_y = grad_w[:, 1:2]
    w_z = grad_w[:, 2:3]

    # Gradient of p with respect to (x, y, z)
    grad_p = torch.autograd.grad(p, coords, grad_outputs=torch.ones_like(p),
create_graph=True)
    p_x = grad_p[:, 0:1]
    p_y = grad_p[:, 1:2]
    p_z = grad_p[:, 2:3]

    # --- Compute Second-Order Derivatives (Laplacians) ---
    # To get the second derivative, we take the gradient of the first derivative.

    # Laplacian of u
    u_xx = torch.autograd.grad(u_x, coords, grad_outputs=torch.ones_like(u_x),
create_graph=True)[:, 0:1]
    u_yy = torch.autograd.grad(u_y, coords, grad_outputs=torch.ones_like(u_y),
create_graph=True)[:, 1:2]
    u_zz = torch.autograd.grad(u_z, coords, grad_outputs=torch.ones_like(u_z),
create_graph=True)[:, 2:3]
    laplacian_u = u_xx + u_yy + u_zz

    # Laplacian of v
    v_xx = torch.autograd.grad(v_x, coords, grad_outputs=torch.ones_like(v_x),
create_graph=True)[:, 0:1]
    v_yy = torch.autograd.grad(v_y, coords, grad_outputs=torch.ones_like(v_y),
create_graph=True)[:, 1:2]
    v_zz = torch.autograd.grad(v_z, coords, grad_outputs=torch.ones_like(v_z),
create_graph=True)[:, 2:3]
    laplacian_v = v_xx + v_yy + v_zz

    # Laplacian of w
    w_xx = torch.autograd.grad(w_x, coords, grad_outputs=torch.ones_like(w_x),
create_graph=True)[:, 0:1]
    w_yy = torch.autograd.grad(w_y, coords, grad_outputs=torch.ones_like(w_y),
create_graph=True)[:, 1:2]
    w_zz = torch.autograd.grad(w_z, coords, grad_outputs=torch.ones_like(w_z),
create_graph=True)[:, 2:3]
```

```
    laplacian_w = w_xx + w_yy + w_zz

    # --- Assemble the Residuals ---

    # Continuity equation residual (mass conservation)
    residual_continuity = u_x + v_y + w_z

    # u-momentum equation residual
    # R_u = rho * (u*u_x + v*u_y + w*u_z) + p_x - mu * laplacian_u
    advection_u = u * u_x + v * u_y + w * u_z
    residual_u_momentum = rho * advection_u + p_x - mu * laplacian_u

    # v-momentum equation residual
    # R_v = rho * (u*v_x + v*v_y + w*v_z) + p_y - mu * laplacian_v
    advection_v = u * v_x + v * v_y + w * v_z
    residual_v_momentum = rho * advection_v + p_y - mu * laplacian_v

    # w-momentum equation residual
    # R_w = rho * (u*w_x + v*w_y + w*w_z) + p_z - mu * laplacian_w
    advection_w = u * w_x + v * w_y + w * w_z
    residual_w_momentum = rho * advection_w + p_z - mu * laplacian_w

    return residual_continuity, residual_u_momentum, residual_v_momentum, residual_w_momentum
```

5.3 Coordinate Sampling Strategy

The choice of collocation points {(xi,yi,zi)} within the vessel mask where the physics loss is evaluated is critical for both accuracy and computational efficiency. Evaluating the loss on every voxel inside the mask is computationally prohibitive.

Comparison of Strategies:

- **Uniform Random Sampling:** The simplest approach is to randomly sample Nc points from a continuous uniform distribution within the bounding box of the vessel mask at each training iteration.[41] While easy to implement, this method is inefficient as it does not account for the fact that the PDE residual is often highly non-uniform.

- **Adaptive Importance Sampling:** More advanced methods adapt the sampling distribution during training to focus on regions where the model is performing poorly (i.e., where the PDE residual is high).[42] Several variants exist:
  - **Residual-Based Rejection Sampling:** Sample a large number of points uniformly, compute the residual for all of them, and then select a training batch by sampling from this set with probabilities proportional to their residual magnitude.[41] This is more accurate but computationally expensive.
  - **Retain-Resample-Release (R3) Sampling:** As described in Section 5.1, this method provides an efficient way to incrementally accumulate points in high-residual regions without the overhead of evaluating residuals on a massive point cloud at every step.[38] It maintains a dynamic population of collocation points, retaining those with high error and replacing those with low error.

Definitive Recommendation:

The most effective strategy that balances computational cost and accuracy is an **adaptive importance sampling scheme based on the Retain-Resample-Release (R3) algorithm**. The implementation would proceed as follows:

1. **Initialization:** At the start of training, sample an initial set of Nc collocation points uniformly at random from within the 3D vessel mask.
2. Periodic Adaptation: Every K training steps (e.g., K=100), perform an adaptation step:
a. Evaluate: Compute the magnitude of the total Navier-Stokes residual, ‖RNS‖, for all Nc points in the current set.
b. Retain: Identify the top k% of points with the highest residuals. These points are retained for the next training phase.
c. Resample: Replace the remaining (100−k)% of points with new points sampled uniformly at random from within the vessel mask.
3. **Training:** Use this new, adapted set of Nc points for the next K training steps.

This strategy ensures that the model's training is continuously focused on the regions of the fluid domain that are most difficult to approximate, leading to faster convergence and a more accurate final solution without the prohibitive cost of dense residual evaluation at every step.

Part IV: End-to-End Training, Fusion, and Advanced Ensembling

This final part addresses the overarching strategies required to train the complete pipeline, effectively fuse information from multiple sources, and construct a powerful ensemble for the final submission. These strategies are tailored to the specific multi-task, multi-modal nature of the model and the computational constraints of the competition environment.

6. Multi-Task Loss Balancing

The Expert Classifier is trained with a multi-task objective, minimizing a composite loss function with at least four terms: $Ltotal=w1Lclass+w2Lseg+w3Lloc+w4Lphysics$. The relative weighting (wi) of these terms is critical; improper balancing can lead to one task dominating the learning process, harming overall performance.

6.1 A Comparative Analysis of Loss Weighting Strategies for Computationally Constrained Environments

Two primary families of dynamic loss weighting methods exist: gradient-based methods and uncertainty-based methods.

- **Gradient-Based Methods:** These methods, such as Multiple-Gradient Descent Algorithm (MGDA), directly manipulate the task-specific gradients to find a Pareto-optimal update direction that improves all tasks simultaneously.[44] They often involve computing task-wise gradients for the shared layers and solving a small optimization problem to find the optimal weights or projecting conflicting gradients.[44] The primary disadvantage of these methods is their **high computational cost**. They typically require access to per-task gradients, which can involve multiple backward passes or complex operations on the gradient vectors at each training step, making them poorly suited for a time-constrained environment like a 12-hour Kaggle notebook.[44]
- **Uncertainty Weighting (UW):** This approach frames multi-task learning from a probabilistic perspective, weighting each task's loss based on its homoscedastic uncertainty—a measure of the task's inherent, observation-independent noise.[45] In practice, this is implemented by having the model predict a log-variance term, $σi2$, for each task. The total loss is then formulated as:

Ltotal=i∑2σi21Li+log(σi)

The model learns to balance the tasks by adjusting the σi parameters. A task with high uncertainty (large σi) will have its loss term down-weighted, while the log(σi) term acts as a regularizer to prevent the uncertainties from growing infinitely large. The key advantage of UW is its computational efficiency. It adds only one learnable scalar parameter per task and does not require any special gradient manipulations or multiple backward passes.45 Recent work has proposed an analytical variant, Soft Optimal Uncertainty Weighting (UW-SO), which computes weights directly as the softmax of the inverse losses, further simplifying the process and improving performance over the original UW.44

Definitive Recommendation:

For the computationally constrained Kaggle environment, **Uncertainty Weighting (specifically, the UW-SO variant)** is the definitively recommended method for balancing the four loss terms. Its negligible computational overhead compared to the expensive gradient manipulations required by methods like MGDA makes it the only practical choice for maximizing performance within the 12-hour runtime limit.

7. Advanced Data Fusion Architectures

The model must effectively integrate information from two distinct sources: the four different 3D imaging modalities (CTA, MRA, T1, T2) and the non-imaging patient metadata (e.g., age, sex).

7.1 A Cross-Attention Architecture for Multi-Modal Image Fusion

A simple approach to multi-modal fusion is early fusion (stacking modalities as input channels) or late fusion (averaging predictions). However, a more powerful approach is intermediate fusion using a cross-attention mechanism, which allows the model to learn how to dynamically integrate information between modalities.[48]

Proposed Architecture:

A dedicated **Multi-Modal Cross-Attention Fusion Block** should be integrated into the model's encoder. The architecture would be as follows:

1. **Separate Initial Encoders:** Each of the four imaging modalities (CTA, MRA, T1, T2) is passed through a few initial, separate convolutional layers (a "stem") to extract low-level features specific to that modality. This results in four parallel feature maps: FCTA,FMRA ,FT1,FT2.
2. **Iterative Cross-Attention:** A cross-attention module is then used to fuse these feature maps. Let's designate CTA as the "primary" modality for this example. The fusion process is:

- The feature map FCTA is used to generate the **Query** vector (QCTA).
- The feature maps from the other three modalities are concatenated and used to generate the **Key** (Kother) and **Value** (Vother) vectors.
- The cross-attention operation, Attention(QCTA,Kother,Vother), computes an updated feature map, FCTA′, where the CTA features have been enhanced with contextually relevant information queried from the other modalities.

3. **Symmetric Fusion:** This process is performed symmetrically, with each modality taking a turn as the "primary" query source. This results in four updated feature maps, {FCTA′,FMRA′ ,FT1′,FT2′}, each enriched with information from all other modalities.

4. **Final Aggregation:** These four enriched feature maps are then concatenated or summed and passed to the deeper layers of the main WaveFormer backbone.

This architecture is superior to simple concatenation because it allows the model to learn, for example, that T1-weighted images provide excellent soft-tissue context that can help disambiguate a complex vascular structure seen in the CTA scan.

7.2 Beyond Late Fusion: Advanced Methods for Integrating Patient Metadata

Late fusion, where metadata is simply concatenated with flattened image features just before the final classification layer, is a common but suboptimal strategy. It provides the model with very little opportunity to learn the complex, non-linear interactions between patient characteristics and image features.

Proposed Advanced Fusion Method: Gated Attention Fusion

A more advanced and effective approach is an intermediate fusion strategy using a **gated attention mechanism**.[50] This method allows the metadata to modulate the processing of the image features at a deeper level.

Architecture:

1. **Metadata Embedding:** The patient metadata (e.g., age, sex) is first processed by a small MLP to produce a low-dimensional embedding vector, Emeta.
2. **Gating Vector Generation:** This embedding vector Emeta is then used to generate a **gating vector**, G, via another linear layer followed by a sigmoid activation function. The dimension of G is equal to the number of channels in the image feature map, Fimage, at a chosen layer in the backbone.
3. Feature Modulation: The gating vector G is then used to modulate the image features via element-wise multiplication:

   $$Fimage' = Fimage \odot G$$

   This operation, similar to a Squeeze-and-Excitation block, allows the model to use the patient's age and sex to dynamically up-weight or down-weight entire feature channels. For example, it might learn that certain textural features are more indicative of an aneurysm in older patients and therefore amplify those channels when the "age" metadata value is high.

This gated fusion mechanism is a form of feature-wise linear modulation (FiLM) and provides a much richer way for the model to integrate metadata, moving beyond simple concatenation to a more powerful conditional computation.

8. A Novel Stacking Ensemble Strategy

The final step in a competitive machine learning project is ensembling. Stacking is a powerful ensemble technique where the predictions of multiple base models (level-1 models) are used as input features for a meta-model (level-2 model) that makes the final prediction.[51] A standard stacking approach would simply use the predicted probabilities from several folds of the Expert Classifier as features for an XGBoost meta-learner. However, the unique architecture of our model allows for a more sophisticated approach.

8.1 Leveraging Interpretability Metrics as Meta-Features for a Level-2 Ensemble

The AKOrN and PINN components of the Expert Classifier produce not only predictions but also valuable, interpretable metrics about the *nature* of those predictions. These metrics can be harvested and used as powerful meta-features to give the level-2 model deeper insight into the reliability of the level-1 predictions.[51]

Proposed Stacking Architecture:

1. **Level-1 Model Training:** Train the full two-stage pipeline using 5-fold cross-validation. This will result in 5 trained Expert Classifier models.
2. **Meta-Feature Generation:** For each sample in the training set (using the out-of-fold predictions) and for each sample in the test set, generate the following outputs from each of the 5 level-1 models:

- **Prediction Probability:** The standard output probability for the "Aneurysm Present" class.
- **AKOrN Order Parameter:** For each candidate, calculate the Kuramoto order parameter, r=C1‖∑i=1Cxi‖, from the final state of the oscillators in the last AKOrN block. This value, ranging from 0 (incoherence) to 1 (full synchronization), is a direct measure of the model's internal "certainty" or "consensus" about the features it has bound together.[32] A highly synchronized state might indicate a confident prediction.
- **PINN Physics Loss:** For each candidate, calculate the magnitude of the final, minimized Navier-Stokes residual loss, Lphysics. This value represents how physically plausible the model's internal representation of the blood flow is. A very low residual suggests the prediction is consistent with fluid dynamics, while a high residual suggests a physically implausible solution, potentially indicating an erroneous prediction.

3. **Level-2 Meta-Model Training:** Create a new feature set for each sample. If using 5 folds, the feature set for a single sample would be a 15-dimensional vector: [prob_1, order_param_1, pinn_loss_1, prob_2,..., pinn_loss_5]. Train a robust meta-learner, such as **XGBoost**, on this enriched feature set to predict the final aneurysm probability.

This novel ensembling strategy allows the meta-model to learn not just *what* the base models are predicting, but also *how* they are arriving at their predictions. It can learn patterns such as "distrust predictions from model 3 if its PINN loss is high, even if its probability is also high," or "give more weight to predictions where all models show a high degree of oscillator synchronization." This leverages the unique interpretability of the proposed architecture to create a more intelligent and robust final ensemble.

Works cited

1. arxiv.org, accessed August 10, 2025, https://arxiv.org/html/2407.06468v1
2. SpineCLUE: Automatic Vertebrae Identification Using Contrastive Learning and Uncertainty Estimation - arXiv, accessed August 10, 2025, https://arxiv.org/html/2401.07271v1
3. CM-UNet: A Self-Supervised Learning-Based Model for Coronary Artery Segmentation in X-Ray Angiography - arXiv, accessed August 10, 2025, https://arxiv.org/html/2507.17779
4. arxiv.org, accessed August 10, 2025, https://arxiv.org/html/2408.05815v1
5. Self-Supervised Pre-Training with Contrastive and Masked ..., accessed August 10, 2025, https://arxiv.org/pdf/2308.06534
6. Masked Image Modeling Advances 3D Medical Image Analysis - CVF Open Access, accessed August 10, 2025, https://openaccess.thecvf.com/content/WACV2023/papers/Chen_Masked_Image_Modeling_Advances_3D_Medical_Image_Analysis_WACV_2023_paper.pdf
7. Masked Image Modeling Advances 3D Medical Image Analysis | Request PDF - ResearchGate, accessed August 10, 2025, https://www.researchgate.net/publication/368314169_Masked_Image_Modeling_Advances_3D_Medical_Image_Analysis
8. AnatoMask: Enhancing Medical Image Segmentation with Reconstruction-guided Self-masking | Request PDF - ResearchGate, accessed August 10, 2025, https://www.researchgate.net/publication/382111539_AnatoMask_Enhancing_Medical_Image_Segmentation_with_Reconstruction-guided_Self-masking
9. AnatoMask: Enhancing Medical Image Segmentation with ..., accessed August 10, 2025, https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/07731.pdf
10. arxiv.org, accessed August 10, 2025, https://arxiv.org/html/2404.15580v2
11. MiM: Mask in Mask Self-Supervised Pre-Training for 3D ... - arXiv, accessed August 10, 2025, https://arxiv.org/pdf/2404.15580
12. MiM: Mask in Mask Self-Supervised Pre-Training for 3D Medical Image Analysis - arXiv, accessed August 10, 2025, https://arxiv.org/html/2404.15580v1

13. Self-Supervised Learning for 3D Medical Imaging Challenge - Home, accessed August 10, 2025, https://ssl3d-challenge.dkfz.de/

14. Self-Supervised Learning for 3D Medical Imaging (SSL3D) Challenge - Zenodo, accessed August 10, 2025, https://zenodo.org/records/15077453

15. An OpenMind for 3D medical vision self-supervised learning - arXiv, accessed August 10, 2025, https://arxiv.org/html/2412.17041v1

16. The Cancer Imaging Archive, accessed August 10, 2025, https://www.cancerimagingarchive.net/

17. HNSCC - The Cancer Imaging Archive (TCIA), accessed August 10, 2025, https://www.cancerimagingarchive.net/collection/hnscc/

18. HEAD-NECK-PET-CT - The Cancer Imaging Archive (TCIA), accessed August 10, 2025, https://www.cancerimagingarchive.net/collection/head-neck-pet-ct/

19. Browse Collections - The Cancer Imaging Archive (TCIA), accessed August 10, 2025, https://www.cancerimagingarchive.net/browse-collections/

20. NIH Clinical Center releases dataset of 32000 CT images, accessed August 10, 2025, https://www.nih.gov/news-events/news-releases/nih-clinical-center-releases-dataset-32000-ct-images

21. Shared Datasets | Center for Artificial Intelligence in Medicine & Imaging, accessed August 10, 2025, https://aimi.stanford.edu/shared-datasets

22. OpenNeuro, accessed August 10, 2025, https://openneuro.org/

23. Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index: Scientific Reports - PMC - PubMed Central, accessed August 10, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC1415224/

24. DICE Score vs Radiologist-Visual quantification of Virtual Diffusion Sequences–pitfalls of lesion segmentation-based approach as compared to clinical relevance-based - CARPL.ai, accessed August 10, 2025, https://carpl.ai/research/dice-score-vs-radiologist-visual-quantification-of-virtual-diffusion-sequencespitfalls-of-lesion-segmentation-based-approach-as-compared-to-clinical-relevance-based

25. The Impact of Poisoned MRI Image on U-Net Brain Tumor Segmentation - arXiv, accessed August 10, 2025, https://arxiv.org/html/2502.03825v1

26. (PDF) Graph-Based Methods for Vessel Skeletonization - ResearchGate, accessed August 10, 2025, https://www.researchgate.net/publication/390280343_Graph-Based_Methods_for_Vessel_Skeletonization

27. Rethinking Graph Neural Networks for Anomaly Detection - Proceedings of Machine Learning Research, accessed August 10, 2025, https://proceedings.mlr.press/v162/tang22b/tang22b.pdf

28. Topological Analysis of Vascular Networks: A Proof-of-Concept Study in Cerebral Angiography - ResearchGate, accessed August 10, 2025, https://www.researchgate.net/publication/388976612_Topological_Analysis_of_Vascular_Networks_A_Proof-of-Concept_Study_in_Cerebral_Angiography

29. The model of vessel trajectory abnormal behavior detection based on graph attention prediction and reconstruction network - Bohrium, accessed August 10, 2025, https://www.bohrium.com/paper-details/the-model-of-vessel-trajectory-abnormal-behavior-detection-based-on-graph-attention-prediction-and-reconstruction-network/933191783087604316-3885

30. GAT-AD: Graph Attention Networks for contextual anomaly detection in network monitoring - UPCommons, accessed August 10, 2025, https://upcommons.upc.edu/bitstream/handle/2117/423072/1-s2.0-S0360835224009525-main.pdf?sequence=1

31. ARTIFICIAL KURAMOTO OSCILLATORY NEURONS - Andreas Geiger, accessed August 10, 2025, https://www.cvlibs.net/publications/Miyato2025ICLR.pdf

32. Artificial Kuramoto Oscillatory Neurons (AKOrN) - ICLR 2025 - Takeru Miyato, accessed August 10, 2025, https://takerum.github.io/akorn_project_page/

33. Track: Oral Session 1F - ICLR 2026, accessed August 10, 2025, https://iclr.cc/virtual/2025/session/31940

34. FP64 is All You Need: Rethinking Failure Modes in Physics-Informed Neural Networks, accessed August 10, 2025, https://arxiv.org/html/2505.10949v1

35. (PDF) FP64 is All You Need: Rethinking Failure Modes in Physics-Informed Neural Networks - ResearchGate, accessed August 10, 2025, https://www.researchgate.net/publication/391856212_FP64_is_All_You_Need_Rethinking_Failure_Modes_in_Physics-Informed_Neural_Networks

36. FP64 is All You Need: Rethinking Failure Modes in Physics-Informed Neural Networks - ChatPaper, accessed August 10, 2025, https://chatpaper.com/paper/137529

37. [2505.10949] FP64 is All You Need: Rethinking Failure Modes in Physics-Informed Neural Networks - arXiv, accessed August 10, 2025, https://arxiv.org/abs/2505.10949
38. Mitigating Propagation Failures in Physics-informed Neural ..., accessed August 10, 2025, https://proceedings.mlr.press/v202/daw23a/daw23a.pdf
39. Mitigating Propagation Failures in Physics-Informed Neural Networks Using Retain-Resample-Release (R3) Sampling (Conference Paper) - NSF Public Access Repository, accessed August 10, 2025, https://par.nsf.gov/biblio/10470292-mitigating-propagation-failures-physics-informed-neural-networks-using-retain-resample-release-r3-sampling
40. Characterizing possible failure modes in physics-informed neural networks, accessed August 10, 2025, https://users.cs.utah.edu/~zhe/pdf/characterizing_possible_failur.pdf
41. Moving Sampling Physics-informed Neural Networks induced by Moving Mesh PDE - arXiv, accessed August 10, 2025, https://arxiv.org/html/2311.16167v3
42. Adaptive Self-Supervision Algorithms for Physics-Informed Neural Networks, accessed August 10, 2025, https://www.stat.berkeley.edu/~mmahoney/pubs/subramanian_ECAI23.pdf
43. An Importance Sampling Method for Generating Optimal Interpolation Points in Training Physics-Informed Neural Networks - MDPI, accessed August 10, 2025, https://www.mdpi.com/2227-7390/13/1/150
44. Analytical Uncertainty-Based Loss Weighting in Multi-Task Learning - arXiv, accessed August 10, 2025, https://arxiv.org/html/2408.07985v1
45. Multi-Task Learning Using Uncertainty to Weigh Losses for Heterogeneous Face Attribute Estimation - arXiv, accessed August 10, 2025, https://arxiv.org/html/2403.00561v1
46. [Literature Review] Analytical Uncertainty-Based Loss Weighting in, accessed August 10, 2025, https://www.themoonlight.io/en/review/analytical-uncertainty-based-loss-weighting-in-multi-task-learning
47. Analytical Uncertainty-Based Loss Weighting in Multi-Task Learning - ResearchGate, accessed August 10, 2025, https://www.researchgate.net/publication/383153166_Analytical_Uncertainty-Based_Loss_Weighting_in_Multi-Task_Learning
48. Cross-Attention Transformer-Based Visual-Language Fusion for ..., accessed August 10, 2025, https://www.preprints.org/manuscript/202502.2255/v1

49. A multimodal vision transformer for interpretable fusion of functional and structural neuroimaging data - PubMed Central, accessed August 10, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11599617/

50. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines, accessed August 10, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC7567861/

51. An interpretable stacking ensemble model for high-entropy alloy mechanical property prediction - Frontiers, accessed August 10, 2025, https://www.frontiersin.org/journals/materials/articles/10.3389/fmats.2025.1601874/full

52. Enhanced and Interpretable Prediction of Multiple Cancer Types Using a Stacking Ensemble Approach with SHAP Analysis - MDPI, accessed August 10, 2025, https://www.mdpi.com/2306-5354/12/5/472

53. Chapter 16 Interpretable Machine Learning - · Bradley Boehmke, accessed August 10, 2025, https://bradleyboehmke.github.io/HOML/iml.html