Latent Visual Diffusion Reasoning for Explainable Pathology Grading

Faris H. Rizk[1] FARIS.HAMDI.RIZK@GMAIL.COM
[1] *Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, Egypt*

Nan Xi[2] NANXI@BUFFALO.EDU
[2] *University at Buffalo*

Junsong Yuan[2] JSYUAN@BUFFALO.EDU
[2] *University at Buffalo*

Editors: Under Review for MIDL 2026

Abstract

Pathology grading (e.g., tumor severity assessment) is a fundamental task in clinical workflows and plays a critical role in guiding diagnosis, treatment planning, and prognosis. However, achieving accurate grading requires extensive domain expertise and a structured, multi-step reasoning process that integrates subtle visual cues across the imaging volume. Most existing computational approaches simplify this task into a direct regression or classification problem, predicting only the final grade without revealing the intermediate reasoning steps that lead to the decision. This lack of transparency limits their utility in clinical settings, where interpretability and alignment with human reasoning are essential.

To address these limitations, we propose Latent Visual Diffusion Reasoning (LVDR), a framework that explicitly models the visual reasoning process in a latent reasoning space. LVDR formulates pathology reasoning as a conditional diffusion process that progressively refine latent reasoning space embeddings, evolving sequentially along the temporal axis of the scan. In essence, the model progressively refine through subsequent slices to produce a coherent reasoning trajectory. This formulation enables LVDR to learn step-by-step, interpretable reasoning paths that uncover how the model integrates visual evidence across slices to reach the final grading decision.

Extensive experiments on three pathology grading benchmarks demonstrate that LVDR achieves performance on par with state-of-the-art methods while providing substantially more interpretable visual reasoning trajectories. These trajectories illuminate the internal decision-making process and offer a level of transparency that is critical for clinical adoption. All codes and models will be public available upon the finish of the review process.

Keywords: Latent Visual Diffusion Reasoning, Explainable Pathology Grading

1. Introduction

Pathology grading serves as a cornerstone of clinical decision-making, directly informing diagnosis, treatment planning, and patient prognosis. For tasks such as tumor severity assessment, clinicians rely on a structured and highly specialized reasoning process: they scrutinize subtle morphological cues, compare patterns across spatially adjacent CT/MRI slices, and integrate multi-scale contextual information before reaching a final grade. This multi-step visual reasoning is essential for ensuring diagnostic reliability.

Despite recent progress in medical image analysis, most approaches still formulate pathology grading as a direct regression or classification task. Such models predict only the final grade, without exposing the intermediate reasoning steps that lead to the decision. Consequently, their outputs remain largely opaque, offering limited insight into how evidence distributed across the imaging volume influences the final prediction. This gap between model behavior and human diagnostic reasoning presents a major obstacle for clinical adoption, where interpretability and transparency are critical.

Recent work has begun addressing this challenge, such as using diffusion autoencoders to generate counterfactual explanations (Atad et al., 2024). However, these counterfactual methods operate on single images and do not account for multi-slice information. In practice, clinicians continuously update their diagnostic hypotheses as they scroll through image stacks: some slices weaken an initial impression, others reinforce it, and occasionally a later slice completely overturns the earlier assessment. Therefore, what is missing is a representation of the model's *reasoning trajectory* across slices to capture how evidence is accumulated, revised, or discarded throughout the volume.

In this work, we introduce Latent Visual Diffusion Reasoning (LVDR), a framework that bridges this gap by explicitly modeling step-by-step visual reasoning in a latent reasoning space. This reasoning space is defined by the essential semantics of the entire patient volume data (CT/MRI scans), such as its global label. Embeddings within this space evolve along continuous trajectories that represent valid and interpretable reasoning paths. Inspired by the generative principles of diffusion models, LVDR formulates pathological grading as a *conditional diffusion process* that unfolds over time. This formulation reflects the intuition that visual reasoning should follow a coherent *latent trajectory* characterized by a gradual refinement of semantic understanding. As the trajectory evolves, it progressively approaches the target distribution, such as the ground-truth label, naturally capturing the progressive structure of human-like reasoning. Specifically, LVDR operates conditional diffusion process in the latent reasoning space. Instead of directly inferring a grade from the full volume, LVDR processes the scan sequentially: it begins with an initial latent state and iteratively refines it as additional slices are observed. This evolution forms a coherent reasoning trajectory that reveals how the model accumulates, updates, and integrates evidence over time. By grounding the decision-making process in an interpretable latent evolution, LVDR naturally aligns with the structured workflow of human experts.

We evaluate LVDR on three pathology grading benchmarks and show that it achieves performance comparable to state-of-the-art methods while offering substantially improved interpretability. The resulting reasoning trajectories make the model's decision-making process explicit, revealing how individual slices contribute to the final prediction and how the model's hypothesis evolves across the imaging volume. Such transparency is essential for building clinically trustworthy AI systems and represents an important step toward deploying computational pathology tools in real-world medical workflows.

In summary, our main contributions are as follows:

- We introduce a new formulation of visual reasoning that models step-by-step inference as a conditional diffusion process in a latent reasoning space, enabling interpretable latent evolution trajectories.

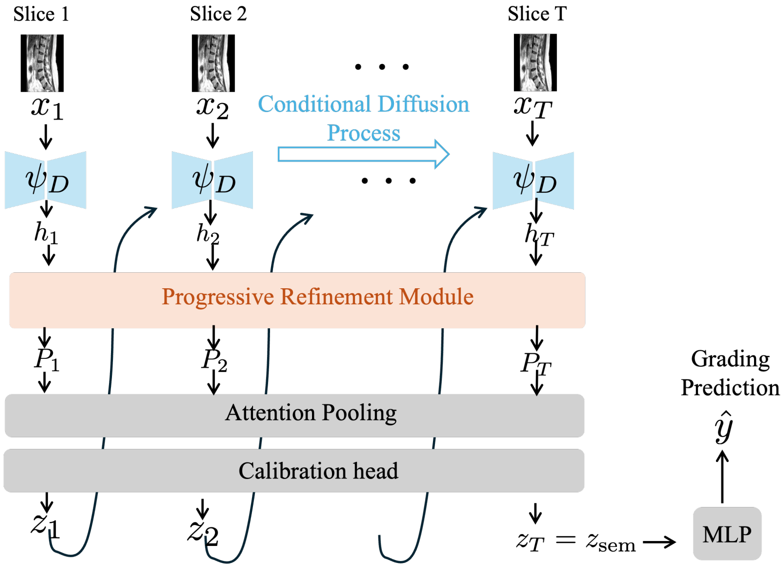

Figure 1: Latent Visual Diffusion Reasoning architecture. A sequence of slices from a patient is reconstructed with diffusion autoencoder ψ_D. Slice-level latent features extracted from ψ_D are further refined by progressive refinement module to yield a latent trajectory $(P_1, \dots, P_T)$. $\{P_i\}_{i=1}^T$ are further updated into $\{z_i\}_{i=1}^T$, which are taken as conditions for the conditional diffusion process.

- We propose LVDR, a novel framework that incrementally refines reasoning states across slices to arrive at the final pathology grade.

- We conduct extensive experiments on multiple pathology grading datasets and demonstrate that LVDR delivers competitive performance while providing interpretability through explicit reasoning trajectories.

2. Method

2.1. Problem Formulation

For a given patient's CT/MRI scan $\mathcal{X} = \{x_1, x_2, \cdots, x_T\}$ containing T slices, pathology grading aims to predict the patient's pathological grade $\hat{y}$ using a learned model $\mathcal{F}_\theta$ parameterized by θ. Formally, $\hat{y} = \mathcal{F}_\theta(\mathcal{X})$, where $\hat{y}$ is belonged to a finite set of discrete categories indicating different levels of pathological severity.

Our LVDR framework consists of two major components: (1) Conditional Diffusion Model ψ_D; (2) Progressive Refinement module ψ_P that progressively refine features generated from the diffusion model, producing latent space embeddings $\{P_i\}_{i=1}^T$.

2.2. Conditional Diffusion Model

The conditional diffusion model consists of two training stages. At the first stage, for each slice x_i, we train the conditional diffusion model ψ_D to reconstruct the slice. After training, we employ ψ_D to extract a slice-level feature representation: $h_i = \psi_D(x_i) \in \mathbb{R}^d$, where d denotes the feature dimensionality. The set of slice features $\{h_i\}_{i=1}^T$ is then refined by the Progressive Refinement Module (described in the next section), producing updated representations $\{z_i\}_{i=1}^T$. These refined features $\{z_i\}_{i=1}^T$ serve as conditioning inputs for the conditional diffusion process used to train the diffusion-based reasoning model in the second stage.

2.3. Progressive Refinement Module

To expose disc-level reasoning as an explicit internal process rather than a hidden side effect of feature aggregation, we adopt a slot-based memory architecture inspired by object-centric models (Locatello et al., 2020). With the extracted slice-level features, we design a Progressive Refinement Module (PRM) that progressively evolves the reasoning trajectory within a latent reasoning space. This latent space is intended to approximate the internal reasoning process of clinicians during decision making, thereby enabling the model to ground its final prediction in an interpretable evolution of latent states.

At the start of each sequence, the model initializes a set of S learnable slot vectors, $M_0 = \{s_1^{(0)}, \ldots, s_S^{(0)}\}, \qquad s_i^{(0)} \in \mathbb{R}^d$, shared across all discs and volumes. At each time step $t \in \{1, \cdots, T\}$, the progressive encoder updates this memory by attending to the current slice tokens. Let $\tilde{U}_t = \{\tilde{u}_{t,1}, \ldots, \tilde{u}_{t,K}\}$ denote the depth-augmented patch tokens for slice t, and let $M_{t-1} = \{s_1^{(t-1)}, \ldots, s_S^{(t-1)}\}$ be the slots before seeing slice t. The update proceeds in three consecutive stages: Cross-attention (CA) + Self-attention (SA) + Feed-forward Network (FFN).

We denote the resulting slot set after processing slice t by $M_t = \{s_1^{(t)}, \ldots, s_S^{(t)}\}$. All subsequent operations, including trajectory extraction and disc-level summarization, operate on this evolving memory. Because the same update mechanism is applied at every time step with shared parameters, the encoder learns a stationary update rule for how disc-level hypotheses should be revised when new evidence arrives.

To turn this internal memory into an explicit reasoning trajectory, we aggregate the slots at each step into a single vector $P_t = \text{pool}(M_t) \in \mathbb{R}^d$, where $\text{pool}(\cdot)$ denotes a simple permutation-invariant operation, such as averaging over slots. The sequence $(P_1, \ldots, P_T)$ is what we refer to as the *latent reasoning trajectory*. P_1 captures the model's initial state after seeing the first slice; intermediate points P_t reflect incremental updates as more slices are incorporated; P_T is the final internal state after all available slices have been processed. In later sections we will project these trajectories into low-dimensional spaces and analyze their norms to study how quickly and how confidently the model converges to a disc-level representation.

While the trajectory (P_t) is the main object of interest for explainability, downstream tasks ultimately operate on a single disc-level representation. After the last slice has been processed and the final slot set M_T has been obtained, we therefore apply an attention-based pooling mechanism to extract a compact semantic summary. Concretely, we introduce a small set of learned pooling queries $\{q_1, \ldots, q_Q\} \in \mathbb{R}^d$, and attend from these queries to the slots in M_T. This is analogous in spirit to the set-based pooling used in Set Transformers (Lee et al., 2019): different queries can specialize to different aspects of the disc-level state, for example one query focusing on central disc signal and another on surrounding bone and soft tissue. The outputs of this pooling layer are then aggregated (for instance by concatenation followed by a linear projection) into a single vector $\tilde{z} \in \mathbb{R}^d$ that serves as a noise-robust summary of the final slot configuration. Empirically, this attention-based pooling yields more stable and informative disc-level vectors than plain averaging, especially when the number of slots is moderate and different slots specialize to distinct disc components.

The pooled representation $\tilde{z}$ is passed through a lightweight calibration head to obtain the final disc-level semantic latent, $z_{\text{sem}} = W\,\text{LN}(\tilde{z}) + b \in \mathbb{R}^d$, where $\text{LN}(\cdot)$ denotes layer normalization and W, b are learned parameters.

2.4. Training Objectives

The progressive encoder is trained with a combination of generative and discriminative objectives, together with an explicit regularizer on the trajectory geometry. The generative term ensures that the disc-level latent z_{sem} retains sufficient information to reconstruct diverse slices from the same disc; the discriminative terms tie this latent to clinically meaningful labels; the trajectory regularizer encourages the intermediate states P_t to evolve in a stable, interpretable manner.

For each disc-level sequence, we first sample a small number of valid slice indices $\mathcal{K} \subset \{1, \ldots, T\}$ (three in our experiments). For each $k \in \mathcal{K}$, we form a noisy version of slice x_k at a randomly sampled diffusion time step τ using the forward noising process of the diffusion autoencoder. The conditional decoder, now conditioned on the disc-level latent z_{sem} (as shown in Figure 1) rather than on a slice-specific latent, is trained to predict the injected noise. Denoting the noise prediction network by $\epsilon_\theta(\cdot\,; z_{\text{sem}}, \tau)$ and the true noise by ϵ, the reconstruction loss takes the form

$$L_{\text{recon}} = \mathbb{E}_{k \in \mathcal{K},\, \tau,\, \epsilon} \left\| \epsilon - \epsilon_\theta(x_k, \tau; z_{\text{sem}}) \right\|_2^2, \tag{1}$$

mirroring the usual diffusion objective (Ho et al., 2020; Preechakul et al., 2022). Conditioning on z_{sem} for multiple, randomly selected slices forces this latent to encode global disc-level semantics that are simultaneously useful for reconstructing different parts of the sequence, rather than overfitting to any single slice.

On top of this generative term, we attach lightweight prediction heads to z_{sem} where disc- or volume-level labels are available. For the SPIDER dataset, Pfirrmann grades are ordinal. We therefore use an ordinal regression head that outputs logits $\ell \in \mathbb{R}^C$ over the C possible grades and define a combined loss

$$L_{\text{ord}} = L_{\text{CE}}(y, \ell) \;+\; \alpha \left(\mathbb{E}[\hat{y}] - y \right)^2, \tag{2}$$

where $y \in \{0, \ldots, C-1\}$ is the true grade, L_{CE} is the cross-entropy loss, $\hat{y}$ is the random variable corresponding to the predicted grade under the softmax distribution over ℓ, and α controls the strength of the auxiliary expected-grade penalty. This second term penalizes large deviations between the expected predicted grade and the true grade, encouraging the network to exploit the ordinal structure rather than treating grades as unrelated classes.

For BraTS, where the task is to detect the presence of peritumoral edema at a coarse volume level, we employ a binary classifier on top of z_{sem} with a standard binary loss, such as cross-entropy or focal loss (Lin et al., 2020), depending on the class imbalance in the split under consideration. In both cases, the predictive losses are introduced primarily to ensure that the learned trajectories and disc-level semantics remain aligned with clinically meaningful distinctions; they are not tuned with the sole aim of surpassing strong slice-level baselines.

To encourage trajectories that are stable, convergent, and visually interpretable, we add an explicit regularizer on the sequence of latent states $(P_1, \dots, P_T)$. Intuitively, we prefer trajectories that move purposefully toward a final representation P_T rather than wandering in latent space before finally collapsing. We capture this preference by penalizing the cosine distance between intermediate states and the final state:

$$L_{\text{traj}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \big(1 - \cos(P_t, P_T)\big), \tag{3}$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity. When P_t and P_T are well aligned, the penalty is small; large angular deviations incur a higher cost. This term does not enforce strict monotonicity, but it biases the model toward trajectories that curve gently toward their endpoint, which in practice makes them substantially easier to analyze and relate to the underlying slices.

The full training objective combines these components as

$$L = L_{\text{recon}} + \lambda_{\text{traj}} L_{\text{traj}} + \lambda_{\text{ord}} L_{\text{ord}} + \lambda_{\text{cls}} L_{\text{cls}}, \tag{4}$$

where L_{cls} collects all classification or grading losses (ordinal on SPIDER, binary on BraTS) and $\lambda_{\text{traj}}, \lambda_{\text{ord}}, \lambda_{\text{cls}} \geq 0$ are scalar weights. These hyperparameters are chosen to balance reconstruction fidelity, trajectory smoothness, and predictive alignment. In particular, we select λ_{cls} large enough that disc-level predictions are non-trivial yet not so large that the model collapses to a purely discriminative solution at the expense of generative explainability. The exact values used for each dataset are reported in an appendix to support reproducibility.

3. Experiments and Results

3.1. Datasets

Our evaluation spans three complementary datasets: (1) SPIDER discs for the primary disc-level reasoning task; (2) BraTS brain tumor volumes for edema-related volumetric analysis; (3) RetinaMNIST for a purely slice-level comparison benchmark.

The SPIDER lumbar spine dataset (van der Graaf et al., 2024) provides multi-slice sagittal MR images with disc-level annotations. Each lumbar disc is assigned a Pfirrmann grade from 0 (healthy) to 4 (severely degenerated) (Pfirrmann et al., 2001). For volumetric tumor imaging, we use the BraTS benchmark (Menze et al., 2015), focusing on the FLAIR modality where peritumoral edema is most prominent. Volumes are treated as axial sequences of slices, again with variable length. RetinaMNIST is drawn from the MedMNIST v2 collection (Yang et al., 2023) and consists of single-slice fundus photographs with ordinal diabetic retinopathy severity labels (five classes). More detailed illustration of datasets can be found in supplement materials.

3.2. Evaluation Metrics

Because SPIDER and RetinaMNIST provide ordinal labels, we adopt metrics that are sensitive to the graded structure rather than treating categories as unordered. On both datasets, we report mean absolute error (MAE) between predicted and true grades, macro F1 to capture class-wise balance, and quadratic weighted Cohen's kappa (Cohen, 1968) as a measure of agreement that explicitly penalizes larger ordinal disagreements more heavily than smaller ones. For SPIDER, we additionally derive a binary degeneration indicator (e.g., grade > 0 versus grade $= 0$) and report area under the ROC curve (AUC) and F1 for this binary task; this provides a simple, clinically intuitive view of how well models separate healthy from degenerated discs.

For the BraTS edema-related task, labels are binary by construction (edema present versus absent). We therefore focus on AUC and F1, complemented by balanced accuracy where the class distribution is notably skewed. More detailed metric introduction is presented in supplement materials.

3.3. Quantitative Results

Model	**Dataset**	**Pathological Category (num of grade)**	**AUC (bin)↑**	**F_1 (bin)↑**	**MAE (ord)↓**	**F_1 (ord)↑**
DiffAE + SVM (Atad et al., 2024)	SPIDER	Pfirrmann grade (5)	0.67	0.93	0.89	0.25
DiffAE + LR (Atad et al., 2024)	SPIDER	Pfirrmann grade (5)	0.65	0.93	0.87	0.33
DenseNet121 (Huang et al., 2017)	SPIDER	Pfirrmann grade (5)	0.64	0.92	**0.83**	0.30
Qwen 7B (zero-shot) (Yang et al., 2025)	SPIDER	Pfirrmann grade (5)	0.50	0.92	1.00	0.0886
Qwen 7B (fine-tuned) (Yang et al., 2025)	SPIDER	Pfirrmann grade (5)	0.69	0.93	1.00	0.28
Qwen 3B (zero-shot) (Yang et al., 2025)	SPIDER	Pfirrmann grade (5)	0.50	0.90	1.03	0.20
Qwen 3B (fine-tuned) (Yang et al., 2025)	SPIDER	Pfirrmann grade (5)	0.66	**0.93**	0.94	**0.36**
LVDR (Ours)	SPIDER	Pfirrmann grade (5)	**0.71**	0.85	1.17	0.33

Table 1: Quantitative performance on the SPIDER dataset. For SPIDER and RetinaMNIST, the binary (bin) columns (AUC and F_1 (bin)) correspond to degeneration or disease presence (e.g., SPIDER: grade >0 vs. grade $=0$), while the ordinal (ord) columns (MAE and F_1 (ord)) correspond to the full multi-grade tasks (5-class Pfirrmann or DR severity).

Model	**Dataset**	**Pathological Category (num of grade)**	**AUC (bin)↑**	**F_1 (bin)↑**	**MAE (ord)↓**	**F_1 (ord)↑**
DiffAE + LR (Atad et al., 2024)	RetinaMNIST	DR severity (5)	0.82	0.86	0.73	0.25
DenseNet121 (Huang et al., 2017)	RetinaMNIST	DR severity (5)	0.81	0.86	0.75	0.32
Qwen 7B (zero-shot) (Yang et al., 2025)	RetinaMNIST	DR severity (5)	0.48	0.62	1.24	0.13
Qwen 7B (fine-tuned) (Yang et al., 2025)	RetinaMNIST	DR severity (5)	0.80	0.86	0.78	0.36
Qwen 3B (zero-shot) (Yang et al., 2025)	RetinaMNIST	DR severity (5)	0.50	0.70	1.45	0.12
Qwen 3B (fine-tuned) (Yang et al., 2025)	RetinaMNIST	DR severity (5)	0.79	0.85	0.82	**0.33**
LVDR (Ours)	RetinaMNIST	DR severity (5)	**0.83**	**0.87**	**0.72**	0.31

Table 2: Quantitative performance on the RetinaMNIST dataset.

Table 1 shows that in the SPIDER dataset, the progressive framework achieves a binary AUC of 0.71 and a binary F_1 of 0.85 when distinguishing degenerated discs (grade > 0) from healthy ones, with an ordinal MAE of 1.17 and ordinal F_1 of 0.33. Compared to slice-level DiffAE baselines from Atad et al. (2024), the progressive encoder is competitive on binary AUC (0.71 versus 0.67 for DiffAE+SVM and 0.65 for DiffAE+LR) but trails these models on binary F_1, which remains very high for all slice-based methods. Its ordinal MAE is higher than that of DenseNet121 and the DiffAE classifiers, indicating that the exact Pfirrmann

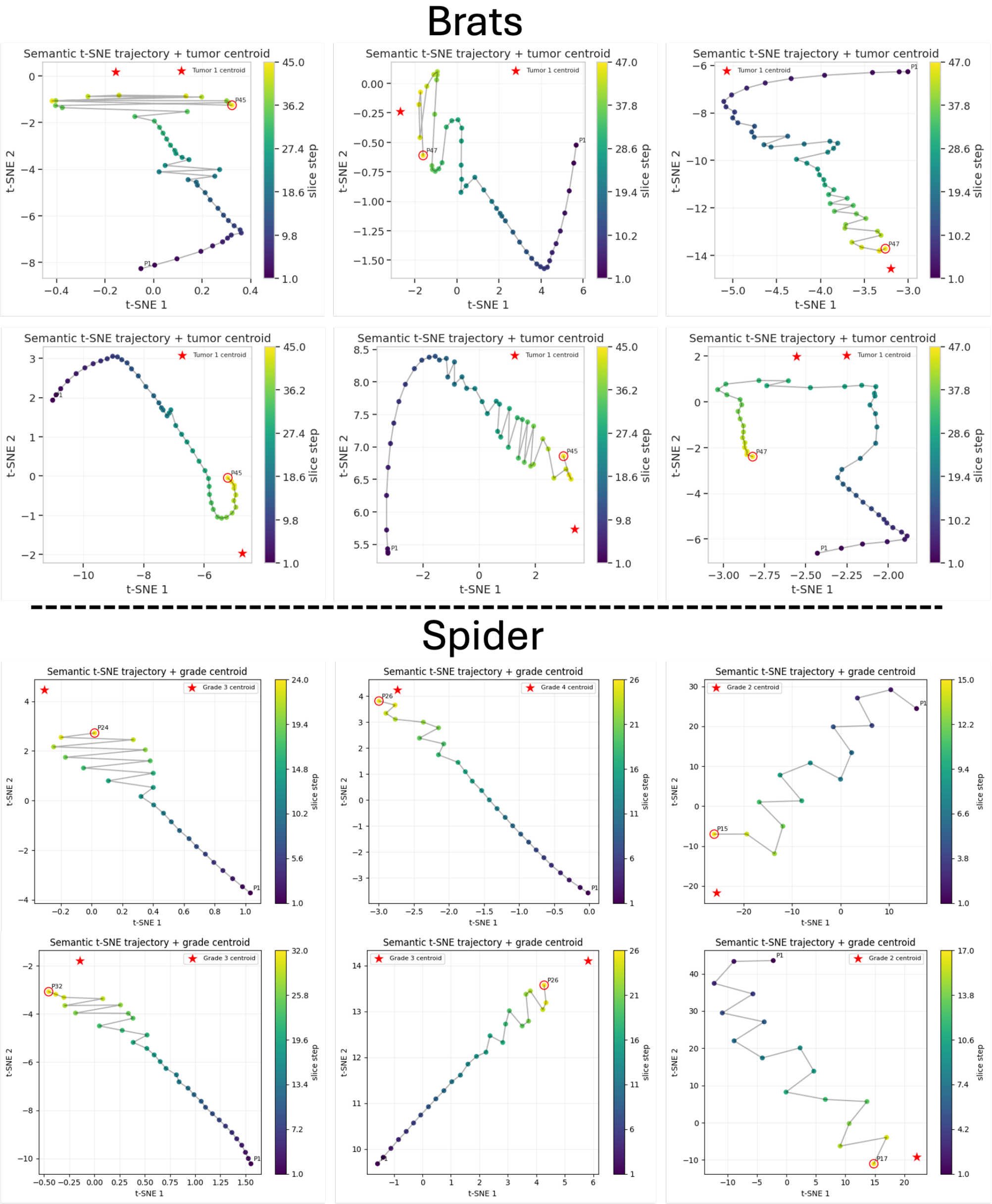

Figure 2: Latent Reasoning Space trajectory for the BRATS and SPIDER dataset. Each dot corresponds to a latent state P_t, color-coded by slice index; the first and last points are annotated, and the centroids of trajectories for discs with different grades are shown as a red star for reference.

grade is predicted less precisely, although the ordinal F_1 is similar to DiffAE+LR and within range of the best Qwen baseline. By the same token, fine-tuned Qwen models benefit from

Model	**Dataset**	**Pathological Category (num of grade)**	**AUC (bin)↑**	**F_1 (bin)↑**	**MAE (ord)↓**	**F_1 (ord)↑**
DiffAE + LR (Atad et al., 2024)	BraTS	Edema (2)	**0.63**	0.94	–	–
DenseNet121 (Huang et al., 2017)	BraTS	Edema (2)	0.49	**0.96**	–	–
Qwen 7B (zero-shot) (Yang et al., 2025)	BraTS	Edema (2)	0.50	0.42	–	–
Qwen 7B (fine-tuned) (Yang et al., 2025)	BraTS	Edema (2)	0.75	0.77	–	–
Qwen 3B (zero-shot) (Yang et al., 2025)	BraTS	Edema (2)	0.50	0.42	–	–
Qwen 3B (fine-tuned) (Yang et al., 2025)	BraTS	Edema (2)	0.68	0.66	–	–
LVDR (Ours)	BraTS	Edema (2)	0.57	0.94	–	–

Table 3: Performance on the BraTS dataset. FFor BraTS, labels are binary and only AUC and F_1 (bin) are applicable. DiffAE+SVM, DiffAE+LR, and DenseNet121 results are taken from Atad et al. (2024); Qwen baselines and the progressive framework are from our experiments.

their large capacity and slice-level supervision: the 3B variant attains the highest ordinal F_1 (0.36), while the 7B variant provides the strongest binary AUC among non-progressive baselines (0.69). These results are consistent with our design goal: the progressive encoder is not tuned to dominate slice-level models numerically but to provide disc-level reasoning with acceptable performance.

In the RetinaMNIST, in contrast, is explicitly a slice-level benchmark, and we do not apply the progressive encoder there. Slice-based baselines perform strongly: DiffAE+SVM reaches an AUC of 0.83 and MAE of 0.72, with DenseNet121 close behind. Zero-shot Qwen models struggle on this task, especially in terms of ordinal MAE and F_1, which is not surprising given the subtlety of diabetic retinopathy grading and the absence of task-specific tuning. Fine-tuning substantially improves Qwen performance—particularly for the 7B model, which matches DenseNet121 on ordinal F_1 and achieves a competitive AUC of 0.80—but these models still do not clearly surpass the simpler diffusion-latent or CNN baselines. This again reinforces a cautious view: large vision–language models can be competitive given sufficient supervision, but they are not a silver bullet for medical grading tasks.

For BraTS, all reported methods operate at the volume level, but labels are binary and derived from segmentation masks. Here, fine-tuned Qwen 7B achieves the strongest AUC (0.75), followed by Qwen 3B (0.68) and DiffAE+LR (0.63). DenseNet121 obtains the highest F_1 (0.96) but with a relatively low AUC (0.49), suggesting that it may be overconfident on the majority class. We deliberately refrain from training a full disc-level progressive model on BraTS because the official training split contains only tumor patients at the case level, so a balanced volume-level edema task is ill-posed. Instead, we use BraTS primarily for trajectory analyses and qualitative visualization of volumetric reasoning, treating the baseline numbers as a rough reference rather than as a target to beat.

Taken together, these quantitative results show that the progressive disc-level encoder achieves reasonable predictive performance on its primary target task (SPIDER Pfirrmann grading) despite operating on far fewer effective training examples than slice-based models. DiffAE classifiers and DenseNet121 have access to many more labeled slices and, not surprisingly, often obtain lower MAE or higher F_1 on pure grading metrics. Qwen models demonstrate that large-scale vision–language pretraining can be leveraged for these tasks, particularly after fine-tuning, but their zero-shot performance is uneven. In this context, the progressive encoder's value lies less in squeezing out marginal gains on established metrics and more in providing an explicit, analyzable reasoning trajectory at the disc level—one that can be interrogated via PCA projections, norm profiles, and reconstructions in a way that slice-level models simply do not support.

3.4. Explainability via Latent Trajectories

To examine how the visual reasoning process evolves in the latent reasoning space, we project the latent states P_t into a two-dimensional space using principal component analysis (PCA). Figure 2 shows representative trajectory samples from the SPIDER and BRATS datasets. Each point corresponds to one step P_t, color-coded by slice index, with the initial state P_1 and final state P_T highlighted. We additionally overlay the trajectory centroids for samples with a particular pathology grade, computed over the training set to provide a coarse, label-specific semantic reference in the latent space. As shown in Figure 2, the trajectory exhibits a smooth evolution from the first to the last slice, and the final latent state lies close to the label-specific semantic reference embedding.

The design of the progressive encoder is motivated by explainability: we want not only a final patient-level prediction but also a transparent description of how it is reached. The latent trajectory (P_t) provides this description. To analyze and visualize these trajectories, we derive three complementary diagnostics.

First, to visualize the latent reasoning process across many discs, we project trajectory points into two dimensions using principal component analysis (PCA). Concretely, we collect all P_t from a subset of slices, fit a PCA transformation on this set, and then plot the path formed by $\{\mathrm{PCA}(P_t)\}_{t=1}^{T}$. Points along the path are color-coded either by slice index (early to late) or by patient label (e.g., Pfirrmann grade).

Second, we monitor the evolution of the trajectory norm. For each disc, we track the Euclidean norm $\|P_t\|_2$ as a function of t. Stabilizing norms suggest that the model's belief has consolidated and that additional slices contribute little new information; sharp changes or non-monotonic behaviour often correspond to slices containing strong or conflicting evidence. By aligning peaks and inflection points in $\|P_t\|$ with the actual slice indices, we can inspect the corresponding images and ask whether the changes in the latent state coincide with anatomically salient features.

4. Conclusion

We introduced Latent Visual Diffusion Reasoning (LVDR), a framework that models pathology grading as a progressive refinement process in a latent diffusion space, enabling interpretable, slice-by-slice reasoning trajectories that expose how visual evidence is integrated to reach the final grade. LVDR matches state-of-the-art performance across three benchmarks while offering substantially clearer and more faithful reasoning paths suited for clinical use. Looking ahead, LVDR opens several promising research directions. The first one is integrating multi-modal clinical data to enrich the latent reasoning space, extending diffusion-based reasoning to other volumetric and temporal medical tasks. The other one is exploring human–AI co-reasoning frameworks where clinicians can inspect, guide, or correct the model's evolving trajectory. These directions highlight the potential of diffusion-driven latent reasoning as a foundation for building transparent and trustworthy medical AI systems.

References

Matan Atad, David Schinz, Hendrik Moeller, Robert Graf, Benedikt Wiestler, Daniel Rueckert, Nassir Navab, Jan S. Kirschke, and Matthias Keicher. Counterfactual explanations for medical image classification and regression using diffusion autoencoder. *Machine Learning for Biomedical Imaging*, 2(iMIMIC 2023 special issue):2103–2125, 2024. doi: 10.59275/j.melba.2024-4862. URL https://www.melba-journal.org/papers/2024:024.html.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. OpenReview (submitted to ICLR 2024), 2023. URL https://openreview.net/forum?id=qrGjFJVl3m.

Shuai Bai, Peng Wang, Sinan Tan, et al. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Jacob Cohen. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968. doi: 10.1037/h0026256. URL https://doi.org/10.1037/h0026256.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017. doi: 10.1109/CVPR.2017.243. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826. URL https://doi.org/10.1109/TPAMI.2018.2858826. Preprint: arXiv:1708.02002.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 11525–11538, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8511df98c02ab60aea1b2356c013bc0f-Abstract.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015. doi: 10.1109/TMI.2014.2377694. URL https://pubmed.ncbi.nlm.nih.gov/25494501/.

Christian W. A. Pfirrmann, Armin Metzdorf, Marco Zanetti, Juerg Hodler, and Norbert Boos. Magnetic resonance classification of lumbar intervertebral disc degeneration. *Spine*, 26(17):1873–1878, 2001. doi: 10.1097/00007632-200109010-00011. URL https://pubmed.ncbi.nlm.nih.gov/11568697/.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10619–10629, 2022. doi: 10.1109/CVPR52688.2022.01036. URL https://openaccess.thecvf.com/content/CVPR2022/html/Preechakul_Diffusion_Autoencoders_Toward_a_Meaningful_and_Decodable_Representation_CVPR_2022_paper.html.

Jasper W. van der Graaf, Miranda L. van Hooff, Constantinus F. M. Buckens, Matthieu Rutten, Job L. C. van Susante, Robert Jan Kroeze, Marinus de Kleuver, Bram van Ginneken, and Nikolas Lessmann. Lumbar spine segmentation in MR images: A dataset and a public benchmark. *Scientific Data*, 11(1):264, 2024. doi: 10.1038/s41597-024-03090-w. URL https://www.nature.com/articles/s41597-024-03090-w.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2: A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023. doi: 10.1038/s41597-022-01721-8. URL https://www.nature.com/articles/s41597-022-01721-8.

Appendix A. More on the Datasets

For the SPIDER dataset, we group sagittal slices into disc-level sequences by anatomical level, yielding variable-length stacks $(x_1, \ldots, x_T)$ where T depends on acquisition and field-of-view. These disc sequences form the natural input to the progressive encoder and the basis for all trajectory analyses. Following the protocol used in prior work on diffusion-autoencoder-based counterfactuals (Atad et al., 2024), we derive our splits from the official SPIDER training and validation partitions: the official training set is further divided into training and validation sets via stratified sampling (10% for validation), while the official validation set is held out as a test set. SPIDER is the primary benchmark on which we evaluate disc-level reasoning trajectories and their alignment with Pfirrmann grades.

For the BraTS dataset (Menze et al., 2015), to obtain a coarse binary edema label at the case or volume level, we leverage the provided segmentation masks: cases with non-zero edema labels in the reference segmentation are considered edema-positive, while volumes without edema in the relevant channels are considered edema-negative. A known caveat is that the official BraTS training split contains only tumor patients at the case level—there are no truly healthy volumes—so fully balanced disc- or volume-level training is not straightforward. We therefore use BraTS primarily to study progressive reasoning in volumetric edema patterns and to report an auxiliary binary classification task, while making the limitations of this setup explicit in the Results and Limitations sections.

For the RetinaMNIST dataset, each case is just one image ($T = 1$), so there is no meaningful notion of a trajectory over depth. We therefore do not apply the progressive encoder in its intended form. Instead, we evaluate only slice-level baselines—diffusion latent classifiers, DenseNet-style convolutional networks, and Qwen-based vision–language models—on this dataset. By the same token, RetinaMNIST complements SPIDER and BraTS by serving as a standard slice-level benchmark where our method is deliberately out of scope; it helps contextualize the behaviour of the shared diffusion backbone and the baselines outside multi-slice settings.

Appendix B. Baselines

Our baselines are designed to probe two questions: how much information about disc-level labels is already present in slice-level diffusion latents and standard CNN features, and how a large vision–language model behaves when asked to perform the same tasks either zero-shot or after fine-tuning. Crucially, all diffusion-autoencoder and CNN baseline numbers reported in the main tables are taken directly from Atad et al. (Atad et al., 2024), who evaluate them on SPIDER, RetinaMNIST, and BraTS; our own contributions are the progressive encoder and the Qwen-based baselines.

The first family of baselines consists of classifiers trained on top of diffusion autoencoder latents. Following Atad et al. (2024), we use the frozen DiffAE encoder (Preechakul et al., 2022) to obtain a latent vector for each slice and then fit simple linear and SVM classifiers on these latents. For SPIDER and BraTS, disc- or volume-level predictions are obtained by aggregating slice-level predictions—for example, by averaging class probabilities across all slices belonging to a disc—and then thresholding as appropriate. For RetinaMNIST, each image is treated as a single slice and predictions are per-image. These DiffAE+LR and DiffAE+SVM baselines share the same semantic latent space as our progressive encoder but do not model sequential reasoning at all: each slice is processed independently, and any disc-level decision arises from a post-hoc aggregation rule.

As a strong convolutional baseline, we rely on DenseNet121 (Huang et al., 2017), again using the results reported in Atad et al. (2024). DenseNet121 is trained on individual slices with standard supervisory signals for each dataset. For SPIDER and BraTS, disc- or volume-level labels are obtained via simple aggregation of slice-level predictions, analogous to the DiffAE baselines; for RetinaMNIST, the model directly predicts the diabetic retinopathy severity of each fundus image. This baseline reflects the performance of a widely used discriminative architecture that has no explicit generative component and no mechanism for exposing a multi-slice reasoning process.

The final family of baselines uses large vision–language models from the Qwen series (Bai et al., 2023; Wang et al., 2024; Bai et al., 2025). We consider two model sizes (3B and 7B parameters) and evaluate them in both zero-shot and fine-tuned modes. In the zero-shot setting, each slice or disc is converted into a multimodal prompt—an image paired with a textual instruction such as "Predict the Pfirrmann grade (0–4) for this intervertebral disc"—and the model's textual output is parsed into a discrete label. For disc-level tasks, we either prompt the model with a representative slice or with a collage of multiple slices; in both cases, no gradient updates are performed. In the fine-tuning setting, we adapt the Qwen models on the training splits of SPIDER, RetinaMNIST, and BraTS using supervised multimodal instruction tuning, again framing each task as a question–answer problem. All Qwen baselines are implemented and trained by us; they are not part of the original DiffAE study.

A structural asymmetry runs through these comparisons. The diffusion-latent classifiers, DenseNet, and Qwen baselines all operate (at least during training) on slice-level samples. This means they effectively see far more labeled examples than our disc-level progressive encoder, which processes each disc as a single training instance regardless of how many slices it contains. We acknowledge this sample-count advantage explicitly and interpret performance differences in that light. The aim is not to claim that the progressive encoder dominates slice-level models on pure metrics, but to demonstrate that it achieves competitive performance while offering a qualitatively richer, trajectory-based explanation of its disc-level decisions.

Appendix C. Implementation Details

The progressive encoder shares a single configuration across all SPIDER experiments, with minor adaptations for BraTS driven primarily by memory considerations. Unless otherwise stated, the latent dimension is fixed at $d = 512$, matching the diffusion autoencoder backbone (Preechakul et al., 2022). The slot-based memory consists of $S = 16$ slots, each in $\mathbb{R}^d$, updated by a stack of two attention blocks with eight attention heads per block. We use attention-based pooling with $Q = 4$ learned queries at the final step, followed by a lightweight calibration head as described in Section 2.5. The maximum effective sequence length is set to $T_{\max} = 32$; longer sequences are subsampled along the depth axis, while shorter sequences use all available slices. Dropout in the attention and feed-forward sublayers is set to 0.1, slot dropout to 0.05, and stochastic depth to 0.1. These hyperparameters—together with patch-level tokenization on an 8×8 grid and sinusoidal depth embeddings—were selected based on preliminary experiments on SPIDER and kept fixed for all subsequent SPIDER runs. For BraTS, we reuse the same progressive architecture and hyperparameters, only adjusting batch size when necessary.

Optimization follows a standard but carefully controlled recipe. We use AdamW (Loshchilov and Hutter, 2019) with learning rate 10^{-4}, weight decay 10^{-2}, and default β parameters. The batch size for SPIDER is set to 12 disc-level sequences, with gradient clipping at a global norm of 1.0 to stabilize training in the presence of noisy or very long trajectories. Where supported by the hardware, we enable mixed-precision training (fp16) to reduce memory footprint and improve throughput. To promote reproducibility, we fix random seeds at the framework level, initialize NumPy and PyTorch RNGs deterministically, and enable deterministic backends where this does not incur prohibitive slowdowns; details of the exact settings are summarized in the appendix.

The training schedule is governed by a fixed number of optimization steps rather than a fixed number of passes over the data, consistent with the diffusion autoencoder codebase we build on. For SPIDER, we monitor performance on a held-out validation split and retain the checkpoint with the best validation ordinal performance (mean absolute error on grades) under the full training objective. BraTS models are selected analogously based on validation AUC for the edema detection task. RetinaMNIST baselines are trained with the same optimizer hyperparameters but without the progressive encoder, using early stopping on validation MAE. All models are trained on a single NVIDIA L4 GPU setup in our experiments.

Appendix D. More on the Evaluation Metrics

There is, at present, no widely accepted scalar metric for the interpretability of latent trajectories. We therefore treat explainability as an analysis dimension rather than as a single number. In the Results section, we qualitatively examine (i) how trajectories cluster by Pfirrmann grade or edema label when projected into a low-dimensional space, (ii) how quickly and smoothly they converge as more slices are incorporated, and (iii) how large trajectory steps align with visually salient slices that change the apparent severity of degeneration or extent of edema. Predictive metrics then play a supporting role: they act as sanity checks that the trajectories and disc-level latents remain informative about clinically relevant decisions, but they are not the primary criterion by which we assess the usefulness of the progressive encoder.