SliceWorld: A Predictive and Controllable World-State Model for CT Report Generation

Yuanhe Tian[♥]
[♥]Zhongguancun Academy
[♥]yhtian94@gmail.com

Yan Song[♠*]
[♠]University of Science and Technology of China
[♠]clksong@gmail.com

Abstract

CT report generation (CTRG) requires models to summarize three-dimensional anatomical context and pathological findings from hundreds of axial slices. Existing methods typically learn a direct image-to-text mapping, providing limited mechanisms for modeling how CT evidence evolves across slices or how reports respond to controlled changes in latent lesion-related factors. We propose SliceWorld, a CT-specific world-state framework that treats an axial CT scan as an ordered sequence along the z-axis. SliceWorld encodes prefix CT evidence into factor-aware latent states containing anatomy, lesion, and uncertainty components, and projects these states into world tokens used for multi-step future-slice feature prediction, lesion-factor intervention, and LLM-based report generation. The model is first pretrained on CT slice sequences with predictive, factor-aware, and counterfactual objectives, and is then fine-tuned on paired CT-report data. Experiments on M3D-Cap and CT-RATE show that SliceWorld improves natural language generation metrics and clinically oriented automatic evaluation. Further analyses demonstrate multi-horizon future-slice prediction, measurable factor alignment, reduced-slice robustness, and selective lesion-sensitive report modulation.[2]

1 Introduction

Reports are the primary medium for communicating computed tomography (CT) findings and are central to diagnosis, treatment planning, and disease monitoring. Automated reporting has therefore become an active research area, yet existing approaches often treat CTRG as direct image-to-text mapping and overlook the spatial and axial-sequential structure revealed by volumetric CT [1–6].

Existing CT reporting approaches mainly follow two routes: global volume encoding for language-model generation [2, 7], or sequential aggregation of axial slice features [8, 9]. The former dilutes fine-grained anatomical localization, while the latter respects slice order but still treats the hidden state as a report-optimized feature vector rather than a predictive and intervenable CT state. In parallel, the concept of *world models* has emerged in reinforcement learning and video prediction as a powerful paradigm for building internal representations that capture the dynamics of an environment [10, 11]. A world model learns not only to encode observations, but also to predict future observations, support intervention analysis, and provide a structured internal state for downstream decision making. Translating this idea to medical imaging, specifically to CT report generation (CTRG), offers a compelling opportunity: we aim to construct an internal state that captures predictive slice-level visual evidence across adjacent axial slices and supports latent lesion-factor manipulation for report-level analysis.

In this paper, we propose **SliceWorld**, a framework that frames CTRG as a world-modeling problem over the axial slice sequence. Rather than claiming a general-purpose physical simulator, Slice-

*Corresponding author.
[2]Code is available at `https://github.com/synlp/SliceWorld`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

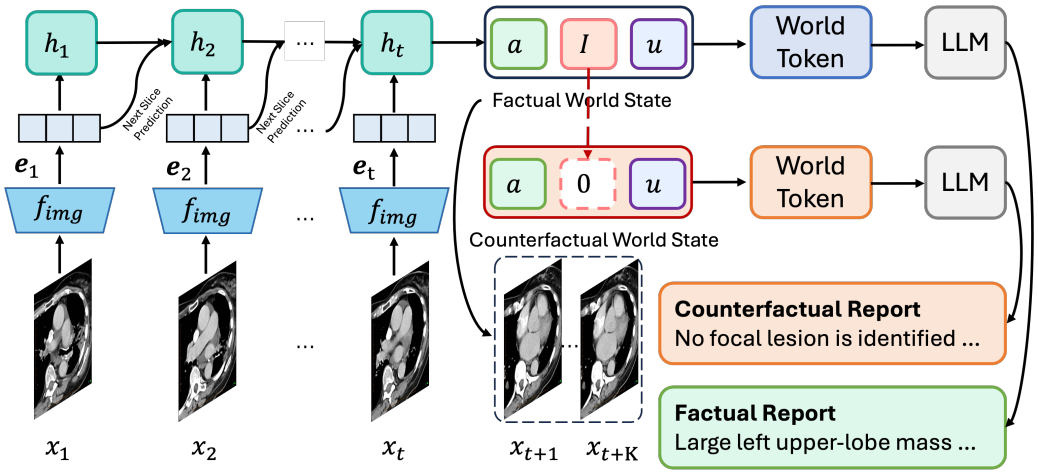


Figure 1: Overview of SliceWorld. Ordered axial CT slices are encoded into prefix states $\mathbf{h}_t = f_{\text{seq}}(\mathbf{e}_{1:t})$ without accessing future slices. Each $\mathbf{h}_t$ is decomposed into anatomy $\mathbf{a}_t$, lesion $\mathbf{l}_t$, and uncertainty $\mathbf{u}_t$ factors, projected into world tokens for LLM-based report generation, and optionally modified through lesion-factor neutralization to produce a latent lesion-neutral report variant.

World studies a narrower but testable form of CT world modeling, where ordered axial evidence is compressed into a predictive and factor-aware state for report generation and controlled latent analysis. Different from conventional sequential aggregators, SliceWorld learns a factor-aware state representation that encourages specialization of anatomy-, lesion-, and uncertainty-related signals. The state evolution follows anatomical continuity priors and is trained with a suite of self- and weakly supervised world-model objectives, including multi-step future-slice feature prediction, factor-aware state learning, and counterfactual future-evidence prediction under lesion-factor intervention. This design provides a structured interface between volumetric CT evidence and language generation, yielding three key capabilities: (i) diagnostic analysis, as the state components are probed for anatomy-, lesion-, and uncertainty-related signals; (ii) controllable analysis, where latent lesion-factor interventions produce lesion-suppressed report variants that are evaluated for non-target preservation; and (iii) partial-observation robustness, where the learned world state remains useful when only a subset of slices is available. We evaluate SliceWorld not only with report-generation metrics, but also with future-slice feature prediction, factor-alignment probing, and lesion-factor intervention, each corresponding to a property implied by the proposed world-state formulation. The results on benchmark CTRG datasets demonstrate the effectiveness of SliceWorld.

Our contributions are summarized as threefold:

- We propose **SliceWorld**, a CT world model for CTRG, which learns a factor-aware state space over the axial slice sequence.
- We introduce a training framework that combines multi-step future-slice prediction, factor-aware state learning, and lesion-factor counterfactual prediction to encourage predictive and intervention-sensitive representations of axial CT structure.
- Extensive experiments demonstrate that SliceWorld achieves strong CTRG results, predicts future slice-level visual evidence beyond persistence baselines, shows measurable factor alignment, and supports lesion-sensitive latent report modulation.

2 The Approach

SliceWorld formulates CTRG as a world-modeling problem over ordered axial CT observations.[3] Given a CT study $V = \{x_t\}_{t=1}^{T}$, where x_t denotes the t-th axial slice and T is the number of observed slices, SliceWorld first encodes the slice sequence into factor-aware world states. At step t,

[3] In this CT-specific formulation, the environment is the patient-specific three-dimensional anatomical and pathological condition, and each axial slice provides a partial observation. The latent state summarizes the observed axial prefix, predicts unobserved slice features, and supports lesion-factor intervention. Prediction and intervention operate in feature and latent spaces, respectively.

the observation is the slice-level visual feature $\mathbf{e}_t$, and the world state $\mathbf{s}_t = [\mathbf{a}_t; \mathbf{l}_t; \mathbf{u}_t]$ summarizes the observed prefix, where $\mathbf{a}_t$, $\mathbf{l}_t$, and $\mathbf{u}_t$ denote the anatomy, lesion, and uncertainty factors, respectively. SliceWorld then projects these states into a sequence of world tokens and generates the CT report from the projected tokens with an LLM. Figure 1 illustrates this overall pipeline, including causal prefix encoding, factor-aware state construction, world-token projection, report generation, and lesion-factor intervention. Formally, the overall generation process is expressed as

$$\widehat{R} = f_{\mathrm{gen}}\big(f_{\mathrm{tok}}(f_{\mathrm{wm}}(V))\big) \tag{1}$$

where $\widehat{R}$ denotes the generated CT report, f_{wm} denotes the world-state encoder that constructs factor-aware world states from the CT slice sequence, f_{tok} projects the world states into interface tokens, and f_{gen} denotes autoregressive report generation from the projected token sequence. To strengthen SliceWorld before report-generation fine-tuning, we introduce a world-model pretraining stage on CT slice sequences. This stage uses predictive and counterfactual objectives to learn representations that capture slice dynamics, semantic factors, and intervention-sensitive behavior. The following subsections describe CT encoding and world-state construction, report generation from world tokens, and world-model pretraining with CTRG fine-tuning.

2.1 CT Encoding and World-State Construction

The world-state encoder f_{wm} aims to map the input CT slice sequence $\{x_t\}_{t=1}^{T}$ to a sequence of factor-aware world states. Specifically, for each slice x_t, we apply an image encoder f_{img} to extract a slice-level visual representation by $\mathbf{e}_t = f_{\mathrm{img}}(x_t)$, where $\mathbf{e}_t$ denotes the visual feature of the t-th slice. We then add a positional encoding $\mathbf{p}_t$ to preserve the relative axial location of each slice and feed the prefix sequence into a causal slice-sequence encoder by

$$\mathbf{h}_t = f_{\mathrm{seq}}(\mathbf{e}_{1:t} + \mathbf{p}_{1:t}) \quad t = 1, \ldots, T \tag{2}$$

where $\mathbf{h}_t \in \mathbb{R}^{d_h}$ denotes the contextualized representation at slice position t. This causal construction ensures that $\mathbf{h}_t$ is computed only from $x_{1:t}$ and does not access future slices $x_{t+1:T}$ when it is used for multi-step future-slice prediction or downstream world-state construction.

Based on $\mathbf{h}_t$, SliceWorld constructs a factor-aware world state $\mathbf{s}_t = [\mathbf{a}_t; \mathbf{l}_t; \mathbf{u}_t]$ that contains anatomy state $\mathbf{a}_t$, lesion state $\mathbf{l}_t$, and uncertainty state $\mathbf{u}_t$ by

$$\mathbf{a}_t = f_{\mathrm{anat}}(\mathbf{h}_t) \quad \mathbf{l}_t = f_{\mathrm{les}}(\mathbf{h}_t) \quad \mathbf{u}_t = f_{\mathrm{unc}}(\mathbf{h}_t) \tag{3}$$

To make lesion-related evidence explicit in the world model, we further attach a lesion-presence head to the lesion factor by

$$\widehat{m}_t = \sigma(f_{\mathrm{occ}}(\mathbf{l}_t)) \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function and $\widehat{m}_t \in (0, 1)$ denotes the predicted probability that slice t is lesion-related. This variable provides an explicit estimate of whether the current slice contains lesion-related evidence, which is later used in supervision and counterfactual learning. Lesion annotations anchor the lesion factor, whereas the multi-step future-slice prediction objective uses feature targets without lesion labels. The report generator receives the complete three-factor state, retaining anatomical context and prediction-difficulty information alongside lesion-related evidence.

2.2 Report Generation from World Tokens

After constructing the factor-aware world state $\mathbf{s}_t$ at each slice position, SliceWorld projects it into a world token $\mathbf{w}_t$ by

$$\mathbf{w}_t = f_{\mathrm{tok}}(\mathbf{s}_t) \tag{5}$$

where $\mathbf{w}_t$ denotes the projected interface token used for language generation.[4] A two-layer multilayer perceptron f_{lm} further maps each world token into the hidden space of the LLM by $\mathbf{z}_t = f_{\mathrm{lm}}(\mathbf{w}_t)$, where $\mathbf{z}_t \in \mathbb{R}^{d_z}$ denotes the language-space token at slice position t. The resulting token sequence $\mathbf{z}_{1:T}$ is fed into the LLM as visual-prefix context, and the report $\widehat{R}$ is generated through the standard autoregressive process.

[4]The state $\mathbf{s}_t$ is the factor-aware world state that stores anatomy, lesion, and uncertainty information, while $\mathbf{w}_t$ is its tokenized representation passed to the language interface. Thus, $\mathbf{w}_t$ is not an additional recurrent state.

The same state-to-token interface also supports lesion-factor intervention at inference time. Given the factual world state $\mathbf{s}_t = [\mathbf{a}_t; \mathbf{l}_t; \mathbf{u}_t]$, SliceWorld constructs a latent lesion-neutral state by modifying only the lesion factor while keeping the anatomy and uncertainty factors unchanged by $\mathbf{s}_t^{\text{lesion-zero}} = [\mathbf{a}_t; \mathbf{0}; \mathbf{u}_t]$. This latent state is then projected by the same token projector through $\mathbf{w}_t^{\text{lesion-zero}} = f_{\text{tok}}(\mathbf{s}_t^{\text{lesion-zero}})$, where $\mathbf{w}_t^{\text{lesion-zero}}$ denotes the world token produced after lesion-factor neutralization. During factual and lesion-neutral generation, the decoder, prompt, decoding strategy, and input CT evidence are kept identical. Thus, differences between the two generated reports are attributed to the latent lesion-factor modification. Here, counterfactual analysis refers to the paired factual and lesion-factor-intervened latent states used to generate the two reports.

2.3 World-Model Pretraining and Report Fine-Tuning

SliceWorld is optimized with an overall objective composed of four terms: multi-step future-slice prediction (MFP), factor-aware state learning (FAS), counterfactual prediction (CF), and CTRG. Therefore, the overall training loss $\mathcal{L}$ is written as

$$\mathcal{L} = \alpha_{\text{mfp}}\mathcal{L}_{\text{mfp}} + \alpha_{\text{fas}}\mathcal{L}_{\text{fas}} + \alpha_{\text{cf}}\mathcal{L}_{\text{cf}} + \alpha_{\text{ctrg}}\mathcal{L}_{\text{ctrg}} \tag{6}$$

where $\mathcal{L}_{\text{mfp}}$ denotes the multi-step feature prediction loss, $\mathcal{L}_{\text{fas}}$ denotes the factor-aware state learning loss, $\mathcal{L}_{\text{cf}}$ denotes the counterfactual prediction loss, and $\mathcal{L}_{\text{ctrg}}$ denotes the CT report generation loss. The coefficients α_{mfp}, α_{fas}, α_{cf}, and α_{ctrg} control which objectives are active in each training stage. The first three terms are used during world-model pretraining on CT slice sequences without text supervision, so that the factor-aware world states capture slice dynamics, lesion-related signals, and the intervention-sensitive effect of lesion factors on future observations.[5] The CTRG term $\mathcal{L}_{\text{ctrg}}$ is then used during report fine-tuning to adapt the projected world-token interface to autoregressive report generation, where $\alpha_{\text{mfp}} = \alpha_{\text{fas}} = \alpha_{\text{cf}} = 0$ with the pretrained model parameters frozen, so the optimization focuses on aligning the pretrained world-token interface with paired report supervision rather than continuing to optimize auxiliary feature-space objectives. The following texts present the four loss terms, where $\widehat{\mathbf{e}}_{t+k}^{(\cdot,k)}$ denotes a prediction of the future slice feature $\mathbf{e}_{t+k}$ with k the future-prediction horizon, and the superscripts ((h), (w), and (w, cf)) specify whether the prediction is produced from the hidden state, the world token, or the counterfactual world token, respectively.

Multi-step Future-slice Prediction Loss. The MFP term trains SliceWorld to predict future slice-level visual features from the current prefix state. It combines two prediction losses by

$$\mathcal{L}_{\text{mfp}} = \lambda_h\mathcal{L}_{\text{mfp}}^{h} + \lambda_w\mathcal{L}_{\text{mfp}}^{w} \tag{7}$$

where $\mathcal{L}_{\text{mfp}}^{h}$ denotes the prediction loss from the sequence representation $\mathbf{h}_t$, $\mathcal{L}_{\text{mfp}}^{w}$ denotes the prediction loss from the world token $\mathbf{w}_t$, and λ_h and λ_w are their corresponding loss weights. A world model is expected to predict how the ordered observation sequence evolves after the current observation. In SliceWorld, this property is enforced by predicting the next K slice-level visual features from the current prefix state. The first pathway predicts future slice features from the sequence representation:

$$\mathcal{L}_{\text{mfp}}^{h} = \frac{1}{K(T-K)} \sum_{t=1}^{T-K} \sum_{k=1}^{K} \left\| \widehat{\mathbf{e}}_{t+k}^{(h,k)} - \mathbf{e}_{t+k} \right\|_2^2 \tag{8}$$

where $\widehat{\mathbf{e}}_{t+k}^{(h,k)} = f_{\text{pred},k}^{h}(\mathbf{h}_t)$ denotes the predicted feature of the k-step future slice from $\mathbf{h}_t$. This objective trains the slice-sequence encoder to capture multi-step axial dynamics. The second pathway predicts the same future slice features from the world token:

$$\mathcal{L}_{\text{mfp}}^{w} = \frac{1}{K(T-K)} \sum_{t=1}^{T-K} \sum_{k=1}^{K} \left\| \widehat{\mathbf{e}}_{t+k}^{(w,k)} - \mathbf{e}_{t+k} \right\|_2^2 \tag{9}$$

where $\widehat{\mathbf{e}}_{t+k}^{(w,k)} = f_{\text{pred},k}^{w}(\mathbf{w}_t)$ denotes the predicted feature of the k-step future slice from $\mathbf{w}_t$. This objective preserves future-prediction ability in the projected token interface, so that the sequence later used for report generation remains tied to the predictive world state.

[5] In other words, $\alpha_{\text{ctrg}} = 0$ in the pretraining stage.

Factor-aware State Loss. The FAS term encourages functional specialization of the anatomy, lesion, and uncertainty factors for stable anatomical context, lesion-related evidence, and prediction difficulty, respectively. It is defined as

$$\mathcal{L}_{\text{fas}} = \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}} + \lambda_{\text{unc}}\mathcal{L}_{\text{unc}} + \lambda_{\text{occ}}\mathcal{L}_{\text{occ}} \tag{10}$$

where $\mathcal{L}_{\text{smooth}}$ regularizes anatomical continuity, $\mathcal{L}_{\text{sparse}}$ regularizes lesion-factor sparsity, $\mathcal{L}_{\text{unc}}$ aligns the uncertainty factor with future-prediction difficulty, and $\mathcal{L}_{\text{occ}}$ supervises slice-level lesion presence. The coefficients λ_{smooth}, λ_{sparse}, λ_{unc}, and λ_{occ} are the corresponding loss weights. The anatomical continuity term encourages the anatomy factor to vary smoothly across adjacent slices through

$$\mathcal{L}_{\text{smooth}} = \frac{1}{T-1}\sum_{t=2}^{T}\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2^2 \tag{11}$$

The lesion sparsity term encourages the lesion factor to remain sparse across the slice sequence through

$$\mathcal{L}_{\text{sparse}} = \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{l}_t\|_1 \tag{12}$$

The uncertainty-alignment term maps the uncertainty factor $\mathbf{u}_t$ to a scalar prediction-difficulty estimate and aligns it with the averaged future-prediction error through

$$\mathcal{L}_{\text{unc}} = \frac{1}{T-K}\sum_{t=1}^{T-K}\left\| g_{\text{unc}}(\mathbf{u}_t) - \text{sg}\left[\frac{1}{K}\sum_{k=1}^{K}\text{Err}\big(\widehat{\mathbf{e}}_{t+k}^{(h,k)}, \mathbf{e}_{t+k}\big)\right]\right\|_2^2 \tag{13}$$

where $g_{\text{unc}}(\cdot)$ maps the uncertainty factor to a scalar, $\text{Err}(\cdot,\cdot)$ denotes the normalized prediction error at each future horizon, and $\text{sg}[\cdot]$ stops gradients through the averaged prediction-error target. Finally, the lesion-presence term explicitly anchors the lesion factor to slice-level lesion-related evidence. Let $m_t \in \{0,1\}$ denote the lesion-presence label for slice t. We train the lesion-presence head with

$$\mathcal{L}_{\text{occ}} = -\frac{1}{T}\sum_{t=1}^{T}\big(m_t \log \widehat{m}_t + (1-m_t)\log(1-\widehat{m}_t)\big) \tag{14}$$

where $\widehat{m}_t$ is the lesion-presence probability predicted from the lesion factor $\mathbf{l}_t$.

Counterfactual Prediction Loss. The CF term trains SliceWorld to produce intervention-sensitive future predictions by comparing factual prediction with lesion-neutralized prediction. It consists of an invariance loss on non-lesion slices and an effect loss on lesion-related slices:

$$\mathcal{L}_{\text{cf}} = \lambda_{\text{inv}}\mathcal{L}_{\text{cf-inv}} + \lambda_{\text{eff}}\mathcal{L}_{\text{cf-eff}} \tag{15}$$

where $\mathcal{L}_{\text{cf-inv}}$ denotes the counterfactual invariance loss on non-lesion slices, $\mathcal{L}_{\text{cf-eff}}$ denotes the counterfactual effect loss on lesion-related slices, and λ_{inv} and λ_{eff} are the corresponding loss weights. A world model is expected not only to predict future observations, but also to respond consistently to interventions on its internal factors. Given the factual world state $\mathbf{s}_t = [\mathbf{a}_t; \mathbf{l}_t; \mathbf{u}_t]$, SliceWorld constructs a counterfactual state by neutralizing only the lesion factor $\mathbf{s}_t^{\text{cf}} = [\mathbf{a}_t; \mathbf{0}; \mathbf{u}_t]$, where $\mathbf{s}_t^{\text{cf}}$ denotes the lesion-neutralized counterfactual state. This state is then projected into a counterfactual world token by $\mathbf{w}_t^{\text{cf}} = f_{\text{tok}}(\mathbf{s}_t^{\text{cf}})$, where $\mathbf{w}_t^{\text{cf}}$ denotes the world token after lesion-factor neutralization. The same multi-step prediction head is applied to the counterfactual token $\widehat{\mathbf{e}}_{t+k}^{(w,\text{cf},k)} = f_{\text{pred},k}^{w}(\mathbf{w}_t^{\text{cf}})$, where $\widehat{\mathbf{e}}_{t+k}^{(w,\text{cf},k)}$ denotes the counterfactual prediction of the k-step future slice feature from the counterfactual world token. For non-lesion slices, lesion-factor neutralization is expected to preserve the future prediction. The invariance loss penalizes the discrepancy between factual and counterfactual predictions on non-lesion slices where $m_t = 0$

$$\mathcal{L}_{\text{cf-inv}} = \frac{\sum_{t=1}^{T-K}\sum_{k=1}^{K}(1-m_t)\left\|\widehat{\mathbf{e}}_{t+k}^{(w,k)} - \widehat{\mathbf{e}}_{t+k}^{(w,\text{cf},k)}\right\|_2^2}{K\sum_{t=1}^{T-K}(1-m_t)+\epsilon} \tag{16}$$

where ϵ is a small constant for numerical stability. For lesion-related slices, lesion-factor neutralization is expected to reduce lesion-relevant predictive information and increase future-prediction error. Since

DeepLesion provides lesion-related slice ranges, m_t indicates whether the current predictive state lies in a lesion-relevant interval, which reduces boundary mismatch when predicting future slice features $\mathbf{e}_{t+1:t+K}$. We define the factual prediction error as

$$\ell_{t,k}^{\text{fact}} = \left\| \widehat{\mathbf{e}}_{t+k}^{(w,k)} - \mathbf{e}_{t+k} \right\|_2^2 \tag{17}$$

and the counterfactual prediction error as

$$\ell_{t,k}^{\text{cf}} = \left\| \widehat{\mathbf{e}}_{t+k}^{(w,\text{cf},k)} - \mathbf{e}_{t+k} \right\|_2^2 \tag{18}$$

where $\ell_{t,k}^{\text{fact}}$ and $\ell_{t,k}^{\text{cf}}$ denote the factual and counterfactual prediction errors for the k-step future slice, respectively. We define the effect loss as a margin-based objective

$$\mathcal{L}_{\text{cf-eff}} = \frac{\sum_{t=1}^{T-K} \sum_{k=1}^{K} m_t \max\left(0, \delta + \ell_{t,k}^{\text{fact}} - \ell_{t,k}^{\text{cf}}\right)}{K \sum_{t=1}^{T-K} m_t + \epsilon} \tag{19}$$

where δ is the margin that requires the counterfactual prediction error to be larger than the factual prediction error on lesion-related slices. This objective encourages the lesion factor to act as an intervention-sensitive component of future prediction rather than as a passive feature channel.

CTRG Fine-tuning Loss. The CTRG term is the standard autoregressive negative log-likelihood over the target report $R = (r_1, \ldots, r_N)$. This term optimizes the language decoder to generate the target report from the world-token sequence.

3 Experiment Settings

3.1 Dataset

We use three datasets with different roles in the overall pipeline. DeepLesion [12] is used for world-model pretraining, while M3D-Cap [13] and CT-RATE [14] are used for CTRG. These datasets provide CT data in different formats. DeepLesion provides lesion-centric axial CT slices with lesion annotations and ordered slice context. M3D-Cap provides CT image-text pairs across multiple anatomical regions and CT views. CT-RATE provides chest CT volumes in HU-valued NIfTI format, paired with radiology reports and abnormality labels. Since SliceWorld takes an ordered image sequence as input, we convert each dataset into this unified representation. For DeepLesion, each CT series is sorted by axial order and used as a slice sequence for world-model pretraining. For M3D-Cap, we follow the official image-text construction and use the ordered CT image group associated with each annotated sample when multiple images are available; samples with a single image are treated as length-one sequences. For CT-RATE, we convert each HU-valued NIfTI volume into a sequence of windowed RGB slice images before feeding it to the visual encoder. DeepLesion is used only for world-model pretraining because it does not provide paired full radiology reports. Its lesion annotations and slice ranges support learning slice dynamics, factor-aware states, slice-level lesion presence, and lesion-factor counterfactual prediction. M3D-Cap evaluates adaptation to heterogeneous CT image-text supervision, while CT-RATE evaluates volume-level chest CT report generation. For all datasets, we follow the official split protocols. The CT-RATE HU-to-image preprocessing protocol is described in Appendix A. DeepLesion series construction and slice-level lesion-label construction are described in Appendix B. Additional details for the report datasets, including M3D-Cap and CT-RATE split usage, are reported in Appendix C.

3.2 Baselines

We compare SliceWorld against strong baselines from recent CT and medical vision-language generation studies. The compared approaches include general medical multimodal models, such as MedVInT-TD [15], Med-Flamingo [16], and RadFM [17], as well as CT-specific report generation approaches, including M3D [13], CT2Rep [2], CT2-CHAT [18], and S-LMR [9]. These baselines cover global volume-to-text modeling, slice-sequential report generation, multimodal foundation models, and anatomy-lesion decomposition-based report generation.

Table 1: Results of existing studies and our approach with different backbone LLMs on M3D-Cap. "†" marks scores reported in Tian et al. [9], and models without any marker use their original reported scores. LLM size reports the language backbone size when applicable.

	LLM size	BLEU-1	ROUGE-1	METEOR	BERT-score	RaTEScore (F1)
MedVInT-TD† [15]	7B	11.7	16.2	11.4	87.7	–
Med-Flamingo† [16]	9B	12.4	16.8	11.7	88.0	–
RadFM† [17]	14B	13.1	17.4	13.0	88.3	–
M3D [13]	7B	15.2	19.6	14.4	88.5	–
CT2Rep† [2]	–	16.2	18.8	14.1	88.5	–
S-LMR [9]	2B	17.4	19.3	14.8	88.8	36.0
Ours (Qwen3-1.7B)	2B	18.1	20.5	15.3	89.3	36.7
Ours (Qwen3-4B)	4B	**18.5**	**20.8**	**15.6**	**89.6**	**37.1**
Ours (Ministral-3-3B)	3B	18.3	20.7	15.4	89.4	36.9

Table 2: Results of existing studies and our approach with different backbone LLMs on CT-RATE. "†" and "‡" mark scores reported in Tian et al. [9] and Hamamci et al. [14], respectively. Models without any marker use their original reported scores. LLM size reports the language backbone size when applicable.

	LLM size	BL-1	BL-2	BL-3	BL-4	R-L	MT	Green	RaTEScore (F1)
MedVInT-TD† [15]	7B	33.9	26.7	21.8	18.3	30.2	18.4	40.2	12.1
Med-Flamingo† [16]	9B	34.9	27.4	22.5	19.0	31.1	19.1	42.4	13.8
RadFM† [17]	14B	36.4	28.0	22.9	19.3	31.7	19.9	43.3	15.4
CT2Rep‡ [2]	–	37.2	29.2	24.4	21.3	36.2	19.7	48.7	16.0
CT2-CHAT‡ [18]	8B	37.3	28.4	23.1	19.8	32.6	21.5	43.6	18.4
S-LMR† [9]	2B	38.1	29.6	24.9	21.7	37.0	21.9	49.4	18.9
Ours (Qwen3-1.7B)	2B	38.7	30.4	25.4	22.2	37.7	22.4	50.1	20.1
Ours (Qwen3-4B)	4B	**39.4**	**30.9**	**26.0**	**22.6**	**38.2**	**23.0**	**50.7**	**20.6**
Ours (Ministral-3-3B)	3B	39.0	30.6	25.7	22.4	38.0	22.7	50.4	20.2

3.3 Implementation Details

SliceWorld uses DINOv2 ViT [19] as the slice-level visual encoder and a randomly initialized Mamba sequence encoder [20] for axial sequence modeling. A cross-attention compression module aggregates DINOv2 patch tokens into one representation per observed slice. The Mamba encoder is implemented as a unidirectional causal scan. For language decoding, the world-token sequence is mapped to the decoder hidden space by a two-layer multilayer perceptron and used as a continuous prefix to the textual input. We use Qwen3-1.7B, Qwen3-4B [21], and Ministral-3-3B [22] as decoder backbones. During report fine-tuning, we freeze the pretrained world-state encoder, apply LoRA [23] to the LLMs, train the projection layers that connect world tokens to the decoder, and use greedy decoding at inference time. All model hyperparameters, LLM backbone details, loss weights, and training settings are reported in Appendix D, with computational cost summarized in Appendix E. All experiments are conducted on NVIDIA A40 GPUs.

We evaluate model performance with two metric categories: standard natural language generation metrics and clinically oriented metrics. The first measures surface-form and semantic similarity, while the second measures whether clinically relevant findings are preserved. For M3D-Cap, we report BLEU-1, ROUGE-1, METEOR, BERTScore, and RaTEScore (F1); for CT-RATE, we report BLEU-1–4, ROUGE-L, METEOR, GREEN, and RaTEScore (F1). In addition, we also perform human evaluation, where we randomly sample 100 cases from the test set and ask two experts to assign a score ranging from 1 to 5 (higher is better) to the report based on its quality, and report the average score. The detailed evaluation protocol, metric definitions, and rationale for dataset-specific metric choices are provided in Appendix F.

Table 3: Ablation study of our approach on CT-RATE with Qwen3-1.7B. All settings share the same architecture and report-generation training setup; +Recon, +MFP, +FAS, and Full SliceWorld use the same DeepLesion data. "*" marks paired bootstrap significance over Direct CTRG ($p \leq 0.05$).

	BL-1	BL-2	BL-3	BL-4	R-L	MT	Green	RaTEScore (F1)	Expert (1–5)
Direct CTRG	36.9	28.5	23.4	20.1	35.4	20.8	47.2	17.5	3.3
+Recon	37.3	29.0	23.9	20.6	36.0	21.2	47.8	18.0	3.3
+MFP	37.8	29.4	24.4	21.4	36.5	21.5	48.5	18.5	3.5
+FAS	38.2	30.1	25.1	22.0	37.2	22.0	49.4	19.3	3.5
SliceWorld	**38.7***	**30.4***	**25.4***	**22.2***	**37.7***	**22.4***	**50.1***	**20.1***	**3.7**

4 Results and Analyses

4.1 Overall Results

The CTRG results of our approach and existing representative models on M3D-Cap and CT-RATE are in Table 1 and Table 2, respectively. We make two observations. First, SliceWorld outperforms the compared representative CTRG approaches on both datasets. Most prior approaches mainly treat CTRG as direct image-to-text generation or slice-feature aggregation, without explicitly learning a predictive and intervention-sensitive world state. The improvements over these baselines therefore suggest that world-modeling provides an effective representation interface for CTRG. Notably, SliceWorld also surpasses several baselines with substantially larger LLM backbones, such as Med-Flamingo (9B), RadFM (14B), and CT2-CHAT (8B), indicating that the gain is not merely a result of scaling the decoder. Second, SliceWorld obtains strong results with different backbone LLMs. Across M3D-Cap and CT-RATE, these variants remain competitive or superior on both standard NLG metrics and clinically oriented automatic metrics.

4.2 Ablation Study

To examine the contribution of each component in SliceWorld, we conduct an ablation study of our approach with Qwen3-1.7B on CT-RATE[6] with different model variants under the same settings. *Direct CTRG* removes the DeepLesion world-model pretraining stage and fine-tunes the same SliceWorld architecture end-to-end on the CTRG dataset, with all model components updated rather than randomly frozen. Starting from Direct CTRG, we then add different pretraining objectives on DeepLesion to form the remaining variants. All pretrained variants use the same DeepLesion series, preprocessing, and pretraining budget, and keep the pretrained visual/world-state components frozen during report fine-tuning. *+Recon*[7] adds a data-matched reconstruction-pretraining stage that uses an MAE-style current-slice reconstruction target instead of MFP. *+MFP* replaces this reconstruction target with multi-step future-slice prediction while retaining the same three-factor architecture and disabling factor-aware objectives, lesion-specific supervision, and counterfactual learning. *+FAS* builds on +MFP by further adding anatomy smoothness, lesion sparsity, uncertainty calibration, and lesion-presence supervision. *Full SliceWorld* builds on +FAS by additionally adding factual/lesion-zero counterfactual future prediction. Table 3 shows a consistent step-by-step improvement. Compared with Direct CTRG, +Recon tests whether gains are explained by the additional DeepLesion data alone. +Recon improves over Direct CTRG, but +MFP performs better with the same DeepLesion data and number of pretraining epochs, indicating a contribution from predictive learning beyond reconstruction. Because +MFP uses no lesion-specific labels, the comparison also shows that part of the reporting gain arises without lesion supervision. Under the same DeepLesion pretraining data, +FAS further improves performance, suggesting that factor-aware learning improves clinically relevant consistency, and Full SliceWorld performs best after adding CF, showing that counterfactual learning complements the predictive and factor-aware objectives. This monotonic trend supports that each component of the proposed world-model training contributes to final CTRG performance, which is further analyzed through prediction, factor-alignment, and intervention studies below.

[6]We also report the results on M3D-Cap under the same setting in Appendix Table 7.

[7]To separate architectural gains from additional pretraining data, the reconstruction and predictive variants use the same DeepLesion budget and differ only in whether the auxiliary objective reconstructs current-slice evidence or predicts future axial evidence.

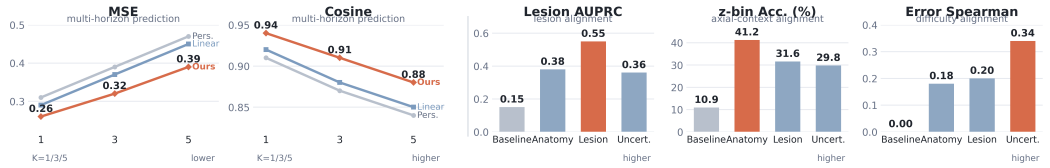


Figure 2: State-level diagnostics of SliceWorld with Qwen3-1.7B. The left two panels show multi-horizon prefix-only future-slice feature prediction on CT-RATE for $K \in \{1, 3, 5\}$. The right three panels show factor-alignment diagnostics on held-out DeepLesion sequences by comparing each state factor with simple statistical baselines. Higher is better except for MSE.

4.3 Effect of CT World Modeling

One of the most essential characteristics of a world model is its capability to predict future observations, which correspond to upcoming slices in the CT volume in our setting. Therefore, we explore whether our approach preserves multi-step future-slice prediction ability. That is, for each position t, the model receives only the prefix slices $x_{1:t}$ and predicts the visual features of future slices $x_{t+1:t+K}$. Since no future slice is encoded, good performance across multiple future horizons indicates that the world state carries predictive spatial information rather than only serving the CTRG decoder. Considering that adjacent CT slices are naturally similar, we compare SliceWorld (using the Qwen3-1.7B backbone) with two CT-continuity baselines, i.e., the persistence baseline and the linear extrapolation baseline. The persistence baseline directly uses the current slice feature for every future horizon, i.e., $\widehat{\mathbf{e}}_{t+k} = \mathbf{e}_t$. The linear extrapolation baseline assumes that the recent feature displacement continues linearly, i.e., $\widehat{\mathbf{e}}_{t+k} = \mathbf{e}_t + k(\mathbf{e}_t - \mathbf{e}_{t-1})$. We report mean squared error (MSE), where lower is better, and cosine similarity, where higher is better, between the predicted and ground-truth visual features. Figure 2 visualizes the multi-horizon prediction trend in the left two panels. SliceWorld achieves lower MSE and higher cosine similarity than both baselines, which suggests that the final CTRG model retains a predictive representation of axial CT structure, rather than learning only a direct mapping from CT features to report text. These comparisons assess prediction beyond simple axial-continuity rules, while Table 3 examines how the training objectives affect downstream report generation.

4.4 Effect of Factor-Aware State Learning

We test whether factor-aware learning assigns clinically relevant information to the intended factors rather than merely improving CTRG scores. Keeping SliceWorld fixed, we extract $\mathbf{a}_t$, $\mathbf{l}_t$, and $\mathbf{u}_t$ on held-out DeepLesion sequences and train the same shallow probe family for each factor. The lesion probe predicts slice-level lesion labels m_t and is evaluated by the area under the precision-recall curve (AUPRC), while the axial-context probe predicts ten normalized z-bins and is evaluated by accuracy. The uncertainty probe estimates the averaged future-prediction difficulty $d_t = \frac{1}{K}\sum_{k=1}^{K} \|\widehat{\mathbf{e}}_{t+k}^{(w,k)} - \mathbf{e}_{t+k}\|_2^2$ and is evaluated by Spearman correlation. The right three panels of Figure 2 show a consistent diagonal trend above simple statistical baselines: $\mathbf{l}_t$ is best for lesion AUPRC, $\mathbf{a}_t$ for z-bin accuracy, and $\mathbf{u}_t$ for error Spearman. All three factors are projected from the same causal hidden state $\mathbf{h}_t$, while the factor-aware objectives encourage distinct functional roles. The nonzero off-diagonal scores indicate residual information sharing among the factors, so the probes support preferential functional specialization rather than statistical independence. Appendix Figure 5 provides qualitative examples.

4.5 Selective Lesion Intervention

We further test whether manipulating the lesion factor produces selective clinical report changes. For each CT-RATE study, we compare a factual report from $\mathbf{w}_{1:T}$ with lesion-zero and uncertainty-zero variants, where $\mathbf{s}_t^{\text{lesion-zero}} = [\mathbf{a}_t; \mathbf{0}; \mathbf{u}_t]$ and $\mathbf{s}_t^{\text{unc-zero}} = [\mathbf{a}_t; \mathbf{l}_t; \mathbf{0}]$. We measure target change and mention removal on a narrow focal-lesion target set, and preservation/hallucination on non-target findings.[8] Table 4 shows that lesion-zero moderately but selectively reduces focal-lesion content in lesion-positive cases, while behaving approximately as a no-op in lesion-negative cases and

[8]We interpret lesion-zero generation as latent report modulation rather than physical lesion removal, and therefore evaluate it through target change, non-target preservation, and negative-control perturbations rather than as clinical counterfactual ground truth.

Table 4: Selective lesion-factor intervention on CT-RATE. Target metrics use a narrow focal-lesion target set; preservation and hallucination are computed on non-target findings. Uncertainty-zero is included as a negative control.

Group	Target chg. (%)	Mention rem. (%)	Preserv. (%)	Halluc. (%)
Positive, lesion-zero	58.4	52.7	94.8	2.2
Positive, uncertainty-zero	9.6	6.8	96.7	1.4
Negative, lesion-zero	4.1	3.0	**97.3**	**1.2**
Negative, uncertainty-zero	2.9	2.2	**97.9**	**1.1**

preserving most non-target findings. The uncertainty-zero control produces much smaller target changes, indicating that the effect is tied to the lesion factor rather than arbitrary latent perturbation. The negative controls test factor selectivity, while non-target preservation and hallucination rates measure the breadth of the induced report changes. To further evaluate whether lesion-factor modulation introduces unsafe report changes, Appendix Table 11 reports automatic clinical-validity proxies based on CT-RATE labels and RadBERT-CT extraction. These proxies show that lesion-zero keeps the overall clinical-label agreement nearly unchanged, has low false-negative risk on non-target findings, and rarely flips non-target labels, supporting that the intervention changes focal-lesion content without broadly corrupting the report. Appendix Figure 4 gives illustrative examples.

5 Related Work

Radiology report generation has been studied extensively in chest X-ray and is increasingly explored in CT, where recent approaches improve image-text alignment through volume-to-text modeling, slice-sequential encoding, anatomy-guided structures, multimodal foundation models, or selective state-space backbones [24–26, 1, 27, 28, 3, 13, 29, 2, 7, 30, 9, 31, 8, 6, 4, 5, 32]. These studies improve report quality by better encoding volumetric context or by separating anatomical and pathological cues, but they still mainly learn a direct mapping from CT representations to text rather than an explicit predictive world representation. More broadly, general vision-language studies model relations among image-text samples [33] and predict structured concepts for image captioning [34]. In parallel, world models have been developed for simulation, prediction, and counterfactual reasoning in sequential environments [10, 11], while their use in medical imaging remains limited and has mostly focused on representation learning rather than report generation [35]. SliceWorld connects these lines of studies by learning a lesion-aware world representation over axial CT slices that is predictive of future observations and usable for latent lesion-factor report modulation.

6 Conclusion

We presented SliceWorld, a CT world model for report generation that treats an axial CT scan as an ordered observation sequence and learns a predictive, factor-aware latent state over slice-level visual evidence. By combining multi-step future-slice feature prediction, factor-aware state learning, lesion-factor counterfactual prediction, and an LLM world-token interface, SliceWorld moves CTRG beyond direct volume-to-text mapping toward a predictive and controllable representation interface. Experiments on M3D-Cap and CT-RATE show consistent improvements over representative CTRG baselines across standard NLG metrics and clinically oriented evaluations. Further ablations and analyses show that the learned state retains prefix-only future-slice prediction ability, exhibits measurable factor alignment, improves robustness under partial slice observation, and supports selective lesion-factor report modulation. These results suggest that CT-specific world-state modeling provides a useful framework for improving volumetric report generation and for analyzing how latent lesion-related evidence affects generated reports.

Acknowledgements

This work is supported by the Key Research and Development Program of Anhui Province, China (Grant No. 202423k09020030) and Zhongguancun Academy (Grant No. XTS0043).

References

[1] Phillip Sloan, Philip Clatworthy, Edwin Simpson, and Majid Mirmehdi. Automated radiology report generation: A review of recent advances. *IEEE Reviews in Biomedical Engineering*, 18: 368–387, 2024.

[2] Ibrahim Ethem Hamamci, Sezgin Er, and Bjoern Menze. CT2Rep: Automated radiology report generation for 3d medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 476–486, 2024.

[3] Xiwei Deng, Xianchun He, Jianfeng Bao, Yudan Zhou, Shuhui Cai, Congbo Cai, and Zhong Chen. Mvketr: chest ct report generation with multi-view perception and knowledge enhancement. *IEEE Journal of Biomedical and Health Informatics*, 2025.

[4] Yilin Li, Chao Kong, Guosheng Zhao, and Zijian Zhao. Automatic radiology report generation with deep learning: a comprehensive review of methods and advances. *Artificial Intelligence Review*, 58(11):344, 2025.

[5] Xinyi Wang, Grazziela Figueredo, Ruizhe Li, Wei Emma Zhang, Weitong Chen, and Xin Chen. A survey of deep-learning-based radiology report generation using multimodal inputs. *Medical Image Analysis*, 103:103627, 2025.

[6] Hamza Kalisch, Fabian Hörst, Jens Kleesiek, Ken Herrmann, and Constantin Seibold. Ct-graph: Hierarchical graph attention network for anatomy-guided ct report generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6775–6784, 2025.

[7] Zhixuan Chen, Yequan Bie, Haibo Jin, and Hao Chen. Large language model with region-guided referring and grounding for ct report generation. *IEEE Transactions on Medical Imaging*, 2025.

[8] Yuanhe Tian, Lei Mao, and Yan Song. Recurrent visual feature extraction and stereo attentions for ct report generation. *arXiv preprint arXiv:2506.19665*, 2025.

[9] Yuanhe Tian and Yan Song. Feature decomposition via shared low-rank matrix recovery for ct report generation. *IEEE Transactions on Medical Imaging*, 2025.

[10] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

[11] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[12] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M Summers. Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *Journal of medical imaging*, 5(3):036501–036501, 2018.

[13] Fan Bai, Yuxin Du, Tiejun Huang, Max Q-H Meng, and Bo Zhao. M3D: Advancing 3d medical image analysis with multi-modal large language models. *arXiv preprint arXiv:2404.00578*, 2024.

[14] Ibrahim Ethem Hamamci, Sezgin Er, Chenyu Wang, Furkan Almas, Ayse Gulnihan Simsek, Sevval Nil Esirgun, Irem Dogan, Omer Faruk Durugol, Benjamin Hou, Suprosanna Shit, et al. Generalist foundation models from a multimodal dataset for 3d computed tomography. *Nature Biomedical Engineering*, pages 1–19, 2026.

[15] Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023.

[16] Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Yash Dalmia, Jure Leskovec, Cyril Zakka, Eduardo Pontes Reis, and Pranav Rajpurkar. Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*, pages 353–367, 2023.

[17] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Towards generalist foundation model for radiology. *arXiv preprint arXiv:2308.02463*, 2023.

[18] Ibrahim Ethem Hamamci, Sezgin Er, Furkan Almas, Ayse Gulnihan Simsek, Sevval Nil Esirgun, Irem Dogan, Muhammed Furkan Dasdelen, Omer Faruk Durugol, Bastian Wittmann, Tamaz Amiranashvili, et al. Developing generalist foundation models from a multimodal dataset for 3d computed tomography. 2024.

[19] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[20] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[21] 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.

[22] Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. volume 1, page 3, 2022.

[24] Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. Generating radiology reports via memory-driven transformer. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1439–1449, 2020.

[25] Zhihong Chen, Yaling Shen, Yan Song, and Xiang Wan. Cross-modal memory networks for radiology report generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 5904–5914, 2021.

[26] Chang Liu, Yuanhe Tian, and Yan Song. A systematic review of deep learning-based research on radiology report generation. *arXiv preprint arXiv:2311.14199*, 2023.

[27] Chang Liu, Yuanhe Tian, Weidong Chen, Yan Song, and Yongdong Zhang. Bootstrapping large language models for radiology report generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 18635–18643, 2024.

[28] Yongheng Sun, Yueh Z Lee, Genevieve A Woodard, Hongtu Zhu, Chunfeng Lian, and Mingxia Liu. R2gen-mamba: A selective state space model for radiology report generation. pages 1–4, 2025.

[29] Yuanhe Tian, Fei Xia, and Yan Song. Diffusion networks with task-specific noise control for radiology report generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1771–1780, 2024.

[30] Cris Claessens, Christiaan Viviers, Giacomo D'Amicantonio, Egor Bondarev, and Fons van der Sommen. Scaling self-supervised and cross-modal pretraining for volumetric ct transformers. *arXiv preprint arXiv:2511.17209*, 2025.

[31] Yuanhe Tian, Zexuan Yan, Nenan Lyu, and Yan Song. Extractive radiology reporting with memory-based cross-modal representations. *IEEE Transactions on Medical Imaging*, 2025.

[32] Louis Blankemeier, Ashwin Kumar, Joseph Paul Cohen, Jiaming Liu, Longchao Liu, Dave Van Veen, Syed Jamal Safdar Gardezi, Hongkun Yu, Magdalini Paschali, Zhihong Chen, et al. Merlin: a computed tomography vision–language foundation model and dataset. *Nature*, pages 1–11, 2026.

[33] Zheren Fu, Zhendong Mao, Yan Song, and Yongdong Zhang. Learning semantic relationship among instances for image-text matching. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15159–15168, 2023.

[34] Ting Wang, Weidong Chen, Yuanhe Tian, Yan Song, and Zhendong Mao. Improving image captioning via predicting structured concepts. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 360–370, 2023.

[35] Yang Yue, Yulin Wang, Chenxin Tao, Pan Liu, Shiji Song, and Gao Huang. Chexworld: Exploring image world modeling for radiograph representation learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20778–20788, 2025.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[37] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004.

[38] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[39] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[40] Weike Zhao, Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Ratescore: A metric for radiology report generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15004–15019, 2024.

[41] Sophie Ostmeier, Justin Xu, Zhihong Chen, Maya Varma, Louis Blankemeier, Christian Bluethgen, Arne Edward Michalson, Michael Moseley, Curtis Langlotz, Akshay S Chaudhari, et al. Green: Generative radiology report evaluation and error notation. *arXiv preprint arXiv:2405.03595*, 2024.

A CT-RATE HU-to-Image Preprocessing

CT-RATE provides 3D NIfTI volumes whose voxel values are already stored in Hounsfield units (HU) as int16 values. We therefore read the NIfTI array directly as HU and do not reapply the DICOM rescale intercept or slope. For each volume, we identify the axial dimension, iterate over all slices, and convert each slice into a three-channel uint8 image using fixed CT windows: lung window $(\mathrm{WL} = -600, \mathrm{WW} = 1500)$ as the red channel, soft-tissue window $(\mathrm{WL} = 40, \mathrm{WW} = 400)$ as the green channel, and bone window $(\mathrm{WL} = 400, \mathrm{WW} = 1500)$ as the blue channel. For each window, HU values are clipped to $[\mathrm{WL} - \mathrm{WW}/2, \mathrm{WL} + \mathrm{WW}/2]$, linearly scaled to $[0, 255]$, and cast to uint8. Each slice is resized to 224×224 using area interpolation, and all slices are saved as an array of shape $(N, 224, 224, 3)$.

B DeepLesion Pretraining Data Construction

DeepLesion contains 32,120 axial CT slices from 10,594 studies of 4,427 patients, with 32,735 lesion annotations in total. In addition to lesion bounding boxes and size measurements, it provides patient-, study-, and series-level metadata, key slices, and lesion-related slice ranges. For DeepLesion pretraining, each CT series is sorted by axial slice order and treated as one observation sequence. We follow the official patient-level split and organize each $(\mathrm{Patient}, \mathrm{Study}, \mathrm{Series})$ tuple as one ordered axial slice sequence. We assign the slice-level lesion label $m_t = 1$ when slice t contains an annotated lesion box or falls within the lesion-related slice range of an annotated lesion in the same series. Slices outside annotated lesion ranges are used as weak negatives with $m_t = 0$. Because DeepLesion is lesion-centric and absence of annotation does not guarantee absence of pathology, we use these labels only for world-model pretraining and factor-alignment diagnostics, and we describe the resulting factors as aligned rather than perfectly disentangled.

C Report Dataset Details

M3D-Cap is a large-scale CT image-text dataset with 120,092 image-text pairs and official train/validation/test splits of 116,092/2,000/2,000. Its samples cover multiple anatomical regions and CT views, providing diverse text supervision for adapting the learned world representation to report generation. Unlike CT-RATE, M3D-Cap is not organized as full-volume radiology reports for every sample. We therefore follow its official image-text benchmark protocol: when a sample provides multiple ordered CT images from the same annotated case/view, we treat them as an ordered slice group; when only one image is available, we use it as a length-one sequence. This keeps the evaluation consistent with prior M3D-Cap studies while allowing SliceWorld to use the same ordered-sequence interface. CT-RATE is a chest CT-report dataset that pairs thoracic CT volumes with radiology reports and abnormality labels. We follow the released split protocol for CT-RATE and use the corresponding training, validation, and test partitions for model development and evaluation. The CT-RATE HU-to-image conversion is described separately in Appendix A.

D Additional Implementation Details

Backbone details. The DINOv2 ViT slice encoder has 24 Transformer blocks, hidden dimension 1,024, and 16 attention heads. The Mamba sequence encoder has 12 layers with hidden dimension $d_{\text{model}} = 512$, state dimension $d_{\text{state}} = 16$, convolution width $d_{\text{conv}} = 4$, and expansion factor 2. The state-factor dimensions are $d_{\text{anat}} = 256$, $d_{\text{lesion}} = 192$, and $d_{\text{unc}} = 64$, and the world-token dimension is 512. The decoder backbones are Qwen3-1.7B, Qwen3-4B, and Ministral-3-3B. Qwen3-1.7B has 28 Transformer layers with hidden size 2,048, 16 attention heads, and 8 key-value heads; Qwen3-4B has 36 layers with hidden size 2,560, 32 attention heads, and 8 key-value heads; Ministral-3-3B has 26 layers with hidden size 3,072, 32 attention heads, and 8 key-value heads.

Slice-token compression. For each CT slice x_t, DINOv2 outputs patch tokens $\mathbf{P}_t \in \mathbb{R}^{M \times d_v}$. We use a learnable slice query $\mathbf{q}_{\text{slc}}$ to attend to these patch tokens through a multi-head cross-attention module:

$$\widetilde{\mathbf{e}}_t = \text{CrossAttn}(\mathbf{q}_{\text{slc}}, \mathbf{P}_t, \mathbf{P}_t) \quad \mathbf{e}_t = W_e \text{LN}(\widetilde{\mathbf{e}}_t) \tag{20}$$

This produces one slice-level feature $\mathbf{e}_t$ per observed axial slice before sequence modeling.

LLM prefix construction. After factor-aware states are projected into world tokens, we map them to the LLM hidden dimension as $\mathbf{z}_{1:T} = f_{\text{lm}}(\mathbf{w}_{1:T})$. During report fine-tuning, these continuous embeddings are prepended to the textual prompt embeddings and then followed by target report tokens:

$$\mathbf{E}_{\text{in}} = [\mathbf{z}_{1:T}; \mathbf{E}_{\text{prompt}}; \mathbf{E}_{\text{report}}] \tag{21}$$

The autoregressive loss is applied only to report tokens, while the world-token prefix and prompt tokens serve as conditioning context.

Training hyperparameters. World-model pretraining uses AdamW with learning rate 1×10^{-4}, weight decay 0.01, batch size 8, dropout 0.1, maximum sequence length 256, maximum sampled slices 196, and 3 epochs. For MFP, the future-prediction horizon is set to $K = 5$, i.e., each valid prefix state predicts the next five slice-level DINOv2 features. Report fine-tuning uses AdamW with learning rate 2×10^{-5}, weight decay 0.01, LoRA rank 16, LoRA $\alpha = 32$, and LoRA dropout 0.05. For M3D-Cap, we use batch size 4, maximum sampled slices 196, and maximum text length 256. For CT-RATE, we use batch size 1 with gradient accumulation 2, maximum sampled slices 480, and maximum text length 384.

Loss weights. During world-model pretraining, the active top-level objectives use $\alpha_{\text{mfp}} = \alpha_{\text{fas}} = \alpha_{\text{cf}} = 1$. During report fine-tuning, the active report-generation objective uses $\alpha_{\text{ctrg}} = 1$. All component weights inside the active objectives are set to 1; the α coefficients indicate whether an objective is active in the current optimization phase. For the MFP term, $\lambda_h = \lambda_w = 1$. For the FAS term, $\lambda_{\text{smooth}} = \lambda_{\text{sparse}} = \lambda_{\text{unc}} = \lambda_{\text{occ}} = 1$, and the lesion-presence BCE uses unit positive-class weight. For the CF term, $\lambda_{\text{inv}} = \lambda_{\text{eff}} = 1$, and the hinge margin is $\delta = 0.1$.

E Computational Cost

All reported runs use NVIDIA A40 GPUs. Table 5 summarizes the cost of the CT-RATE ablation settings in Table 3. The main additional cost of SliceWorld comes from the world-model pretraining stage, while report fine-tuning cost remains similar across ablation variants because the same DINOv2 encoder, Mamba encoder, Qwen3-1.7B decoder, LoRA configuration, and CT-RATE split are used.

Table 5: Computational cost of the CT-RATE ablation settings with Qwen3-1.7B on A40 GPUs.

Training setting	Pretrain GPU h	Fine-tune GPU h	Total GPU h	Peak memory (GB)
Direct CTRG	0.0	38.5	38.5	36.2
+Recon	20.8	38.7	59.5	36.9
+MFP	21.4	38.7	60.1	37.0
+FAS	24.9	38.9	63.8	37.5
Full SliceWorld	28.6	39.0	67.6	38.1

Table 6 reports the computational cost across the three LLM backbones. Since the visual encoder and world-state encoder are shared, larger language backbones mainly increase LoRA fine-tuning memory and decoding latency. The corresponding report-generation performance is already reported in Table 1 and Table 2.

Table 6: Backbone-level computational cost. Memory and time are measured on A40 GPUs under the CT-RATE fine-tuning and inference setting.

Backbone	Params	Fine-tune GPU h	Peak memory (GB)	Sec./study
Qwen3-1.7B	1.7B	39.0	38.1	2.4
Qwen3-4B	4.0B	62.5	57.8	4.1
Ministral-3-3B	3.3B	54.2	51.6	3.5

F Additional Evaluation Details

Report-generation metrics. We use standard natural language generation metrics, including BLEU [36], ROUGE [37], METEOR [38], and BERTScore [39]. BLEU and ROUGE mainly measure lexical overlap, METEOR further accounts for stemming and synonym matching, and BERTScore measures semantic similarity with contextualized token embeddings. These metrics evaluate surface-form and semantic agreement, but they do not fully capture clinical correctness.

Clinically oriented metrics. For M3D-Cap, we additionally report RaTEScore [40], following the common evaluation practice on this benchmark. For CT-RATE, we report GREEN [41], which is designed for clinically oriented automatic evaluation in chest imaging, and RaTEScore (F1) for finding-level report consistency. We compute abnormality-level precision, recall, and F1 for CT-RATE. We do not apply GREEN to M3D-Cap because GREEN is developed for chest imaging, whereas M3D-Cap contains CT data from multiple anatomical regions.

Expert evaluation protocol. To complement automatic clinical metrics, we perform a blinded expert evaluation protocol on 100 randomly sampled CT-RATE studies. Two clinical experts independently score each generated report on a 1–5 clinical correctness scale, where 1 indicates mostly incorrect or unsafe output and 5 indicates clinically accurate and complete output. The system identity is hidden and report order is randomized.

Significance testing. For results marked with * in the ablation study, we use paired bootstrap resampling over the test studies with 10,000 resamples and compare Full SliceWorld against Direct CTRG under the same evaluation pipeline. We mark significance for the metrics when the one-sided bootstrap p-value is at most 0.05. We do not use * markers in the SOTA comparison tables because several prior scores are taken from existing studies and do not provide paired per-study outputs under our pipeline.

G Additional Experiment Results

We include additional experiment results that complement the main analyses, including backbone-controlled CTRG checks, partial-observation robustness, and sensitivity analyses for the world-model training design.

Same-backbone M3D-Cap ablation. Table 7 compares direct report fine-tuning with progressive SliceWorld training variants on M3D-Cap under the same Qwen3-1.7B decoder, DINOv2 encoder, Mamba sequence encoder, LoRA configuration, split, and decoding setting. Direct CTRG removes the DeepLesion world-model pretraining stage and updates all model components end-to-end on M3D-Cap, while +MFP, +FAS, and Full SliceWorld use the same DeepLesion series, preprocessing, and pretraining budget before freezing the pretrained visual/world-state components during report fine-tuning. The comparison shows that world-model pretraining improves CTRG beyond the shared visual and language backbone.

Table 7: Same-backbone CTRG ablation on M3D-Cap with Qwen3-1.7B.

Training setting	BLEU-1	ROUGE-1	METEOR	BERTScore	RaTEScore (F1)
Direct CTRG	16.9	19.3	14.5	88.7	35.5
+MFP	17.5	19.9	14.9	89.0	36.1
+FAS	17.8	20.2	15.1	89.1	36.4
Full SliceWorld	**18.1**	**20.5**	**15.3**	**89.3**	**36.7**

M3D-Cap single-image and multi-slice groups. Because M3D-Cap contains both length-one image-text samples and samples with multiple ordered CT images, we further split its test set into single-image and multi-slice input groups. This analysis tests whether the axial world-state interface remains useful when multiple ordered observations are available, while respecting that M3D-Cap is not a full-volume report dataset. As shown in Table 8, SliceWorld improves over Direct CTRG in both groups, with a slightly larger gain in the multi-slice group where axial evidence is available. Since the single-image group is small, we treat this as a supporting analysis rather than the main evidence.

Table 8: M3D-Cap single-image versus multi-slice input-group analysis with Qwen3-1.7B.

Model	Group	N	BLEU-1	ROUGE-1	METEOR	RaTEScore
Direct CTRG	single-image	65	10.2	15.1	13.0	31.2
Full SliceWorld	single-image	65	10.8	15.8	13.6	32.0
Direct CTRG	multi-slice	1935	17.1	19.4	14.6	35.6
Full SliceWorld	multi-slice	1935	18.3	20.7	15.4	36.9

Reduced-slice CT-RATE inference. Table 9 evaluates CT-RATE report generation when only a uniformly sampled subset of slices is provided at inference time. Both models use the same Qwen3-1.7B decoder and the same decoding protocol. SliceWorld degrades more slowly as the slice budget decreases, supporting the robustness of the learned world state under partial axial observation.

Table 9: Reduced-slice CTRG robustness on CT-RATE with Qwen3-1.7B.

Model	Slice budget	BL-1	GREEN	RaTEScore (F1)
Direct CTRG	100%	36.9	47.2	17.5
Direct CTRG	50%	35.8	46.1	16.4
Direct CTRG	25%	34.6	44.8	15.3
Full SliceWorld	100%	**38.7**	**50.1**	**20.1**
Full SliceWorld	50%	**37.8**	**49.0**	**19.2**
Full SliceWorld	25%	**36.6**	**47.6**	**18.1**

Loss-weight sensitivity. We evaluate CT-RATE sensitivity to the top-level FAS and CF weights while keeping the main experiment setting fixed. We keep $\alpha_{\text{mfp}} = 1$ because multi-step future-slice prediction is the core predictive objective, and keep $\alpha_{\text{ctrg}} = 1$ during report fine-tuning. Table 10 sweeps $\alpha_{\text{fas}}, \alpha_{\text{cf}} \in \{0.5, 0.8, 1.0\}$ using the same CT-RATE setup as Table 3. The results improve as either FAS or CF receives larger weight, and the default $(\alpha_{\text{fas}}, \alpha_{\text{cf}}) = (1.0, 1.0)$ achieves the strongest result, supporting the usefulness of both losses.

Table 10: CT-RATE sensitivity to FAS and CF loss weights with Qwen3-1.7B. α_{mfp} and α_{ctrg} are fixed to 1 in their active training stages.

α_{fas}	α_{cf}	BL-1	GREEN	RaTEScore (F1)
0.5	0.5	37.9	48.9	18.9
0.5	0.8	38.1	49.2	19.2
0.5	1.0	38.2	49.4	19.3
0.8	0.5	38.2	49.3	19.2
0.8	0.8	38.4	49.6	19.6
0.8	1.0	38.5	49.8	19.8
1.0	0.5	38.4	49.7	19.6
1.0	0.8	38.6	49.9	19.9
1.0	1.0	**38.7**	**50.1**	**20.1**

Future-horizon sensitivity. We also evaluate how the multi-step future-prediction horizon affects downstream CT-RATE report generation. Figure 3 reports the result for $K \in \{1, 2, 3, 5, 10\}$ using one standard NLG metric and one clinically oriented metric. Short horizons provide a useful predictive signal, but $K = 5$ performs best because it encourages the state to capture a slightly broader axial context without forcing predictions too far away from the observed prefix. The mild decrease at $K = 10$ suggests that very long feature-space prediction can introduce unnecessary difficulty while still remaining competitive with shorter horizons.

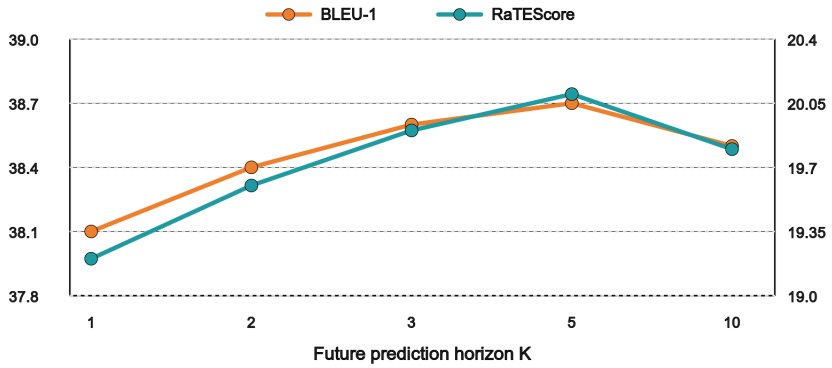


Figure 3: CT-RATE sensitivity to the future-prediction horizon K. We report BLEU-1 and RaTEScore with Qwen3-1.7B. The default $K = 5$ gives the strongest balance between report-generation quality and clinically oriented correctness.

Intervention-validity proxies. Table 11 complements the main selective lesion-intervention results with automatic clinical-validity proxies. We compare factual and intervention reports using RadBERT-CT labels and CT-RATE gold labels, and compute false-negative risk and contradiction on non-target findings. These auxiliary results check whether lesion-factor modulation preserves non-target clinical content rather than merely changing focal-lesion words.

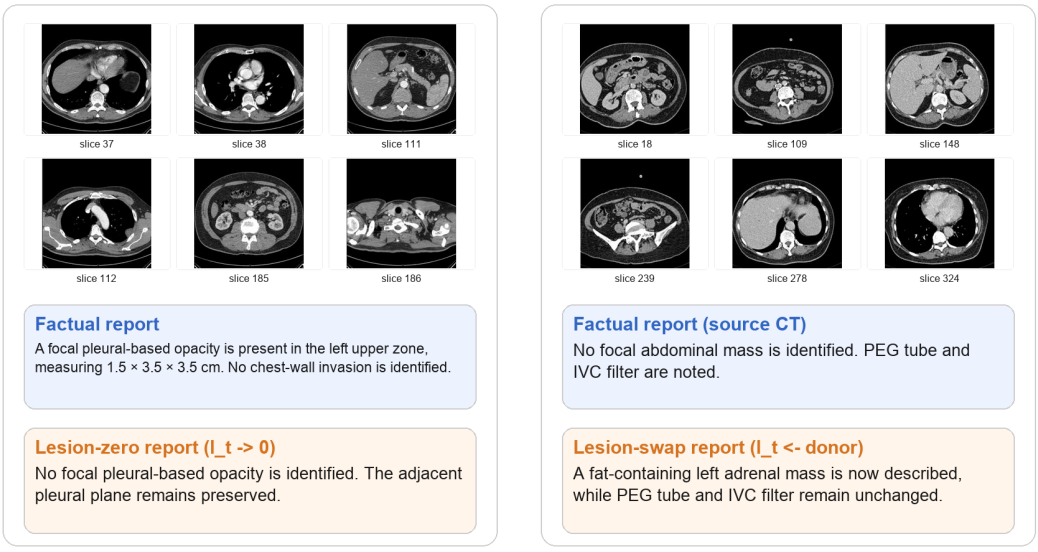
H Additional Qualitative Cases

Figure 4 complements the selective lesion intervention analysis in Section 4.5 by showing representative lesion-zero and lesion-factor insertion examples. Figure 5 complements the factor-aware

Table 11: Automatic intervention-validity proxies on CT-RATE. Clinical F1 Δ compares intervention reports against factual reports using RadBERT-CT labels and CT-RATE gold labels; false-negative risk and contradiction are computed on non-target findings.

Intervention	Clinical F1 Δ	Non-target F1 Δ	FN risk (%)	Contradiction (%)
Lesion-zero	-0.002	-0.001	1.8	4.9
Uncertainty-zero	-0.001	-0.001	1.2	3.7

state analysis in Section 4.4 by visualizing representative CT slices, the lesion-factor trajectory, the generated report, and the reference report. These qualitative figures are provided as illustrative examples; the corresponding aggregate claims are supported by the quantitative metrics in the main tables.

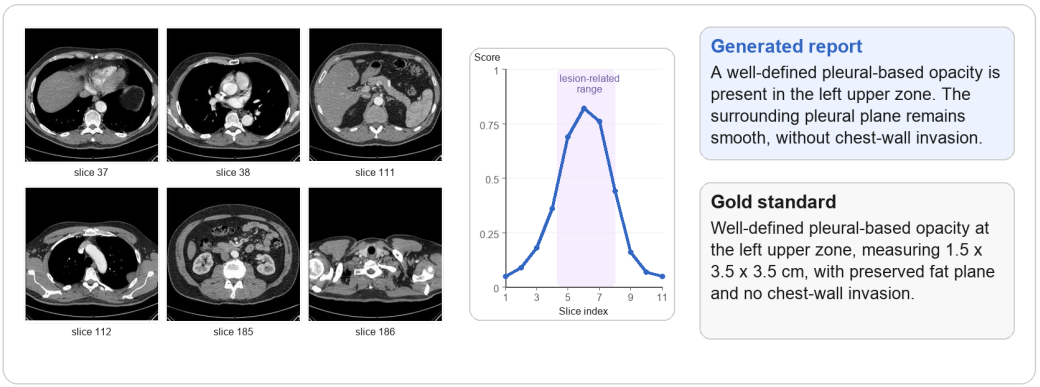


(a) (b)

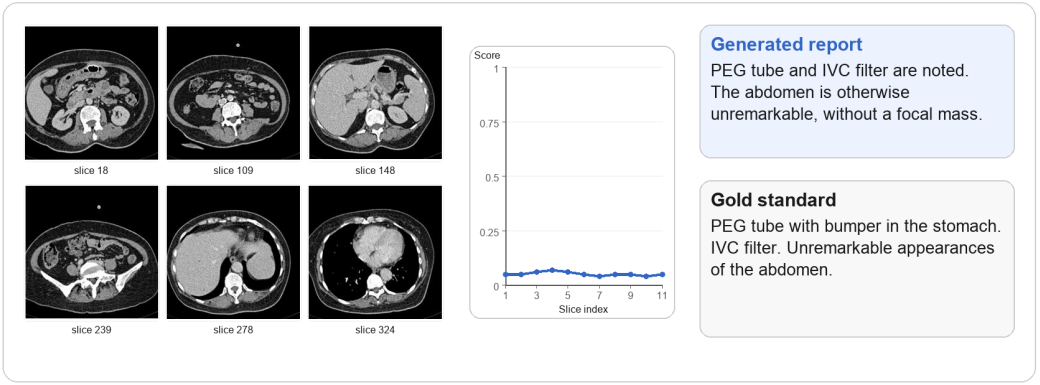
Figure 4: Qualitative counterfactual lesion intervention examples. Panel (a) shows a lesion-positive case where lesion-zero removes a focal pleural-based opacity statement while preserving surrounding anatomical context. Panel (b) shows a lesion-factor insertion case where the inserted lesion factor introduces a donor-like adrenal-mass statement while keeping unrelated source findings unchanged.

I Limitations

SliceWorld uses the term world model in a restricted CT-specific sense: it learns a latent predictive state over ordered axial observations, predicts future slice-level visual evidence in feature space, and exposes factor-level interventions for report analysis. It is not a HU-valued physical CT simulator, an action-conditioned RL environment model, or a model of disease progression over real time. This operational definition is meaningful for CTRG because report generation depends on how local findings appear, persist, and disappear across the z-axis, and our study explores whether such axial evidence can be represented as a predictive and intervention-sensitive state rather than a direct volume-to-text feature. Lesion-zero intervention should therefore be interpreted as latent lesion-factor neutralization rather than physical lesion removal from the image. The factor-alignment analyses rely on available weak lesion labels, axial-position proxies, and automatic report labelers, so they support factor specialization rather than perfect disentanglement. Our clinical evaluation combines retrospective public benchmarks, automatic metrics, and a 100-case evaluation by two clinical experts. Multi-center external validation and prospective radiologist or workflow evaluation remain future directions.



(a)



(b)

Figure 5: Additional qualitative examples for factor-aware state behavior. The lesion-positive example shows a lesion trajectory rising over the slice interval associated with a focal pleural-based opacity, while the control example keeps the trajectory low and avoids hallucinating focal lesion terms.

J Ethics, Assets, and Release

This work uses publicly available, de-identified retrospective CT datasets and does not involve new data collection, crowdsourcing, or prospective human-subject experiments. The method is intended as a research prototype for CT report generation and state-space analysis, not as a standalone clinical decision-making system. Generated reports and counterfactual outputs should be reviewed by qualified clinicians before any practical use. Potential risks include incorrect or incomplete generated findings, over-reliance on automated reports, and misinterpretation of latent lesion-factor interventions as physical image edits. We mitigate these risks by evaluating clinically oriented metrics, reporting limitations, and framing latent lesion-factor report modulation as an analysis tool rather than a diagnostic intervention.

We use existing datasets and model components, including DeepLesion, M3D-Cap, CT-RATE, DINOv2, Mamba, Qwen3, and Ministral, and cite the corresponding original sources. All assets are used under their original licenses, access terms, and dataset usage agreements. We do not redistribute raw medical images, reports, or pretrained third-party model checkpoints.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state that SliceWorld is a world-model framework for CT report generation, and the results evaluate report generation, predictive state behavior, factor alignment, and lesion-factor intervention. The scope is further bounded in the Limitations section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper includes a Limitations section in the appendix discussing latent-space intervention, noisy labels/proxies/automatic labelers, retrospective evaluation, and missing visual evidence.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: The paper does not present theoretical theorems or proof-based claims. The mathematical expressions define the model and training objectives.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The paper describes the architecture, losses, datasets, splits, preprocessing, training settings, evaluation metrics, and compute resources in the main text and appendices. An anonymized supplementary code package is also described in Appendix J.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The submission includes an anonymized code package with core model, preprocessing, training, and evaluation scripts. The medical datasets and third-party checkpoints are not redistributed; users are instructed to obtain them from their official sources under the original access terms.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Dataset roles and splits are described in Section 3.1, implementation details are summarized in Section 3.3, and full hyperparameters, loss weights, preprocessing, and evaluation details are reported in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Results marked with * are tested with paired bootstrap resampling over test studies. The resampling procedure and significance threshold are described in Appendix F.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Section 3.3 states that experiments use NVIDIA A40 GPUs. Appendix E reports GPU hours, peak memory, and backbone-level inference cost.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses publicly available de-identified retrospective datasets and does not involve new human-subject data collection. Appendix J discusses responsible use, asset handling, and release safeguards.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes].

Justification: Appendix J discusses intended research use, potential benefits for CT report generation research, and risks such as incorrect generated findings, over-reliance, and misinterpretation of latent interventions.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes].

Justification: The release does not redistribute raw medical images, reports, or third-party pretrained checkpoints. Appendix J states that the code package is anonymized, contains no patient data, and requires users to obtain datasets/checkpoints from official sources.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: The paper cites the original sources for DeepLesion, M3D-Cap, CT-RATE, DINOv2, Mamba, Qwen3, and Ministral. Appendix J states that all assets are used under their original licenses, access terms, and dataset usage agreements, and are not redistributed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes].

Justification: The new release asset is the anonymized SliceWorld code package. Appendix J describes its scope, and the package includes a README and configuration files documenting model training and evaluation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, user studies, or newly recruited human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The study uses public, de-identified retrospective datasets and does not collect new data from human subjects. No new human-subject experiment is conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: LLMs are part of the core CTRG method. The paper specifies the decoder backbones, how world tokens are projected into the LLM hidden space, and how report generation is trained and evaluated.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.