CROSS-SCALE LATENT PREDICTIVE PRETRAINING FOR WHOLE-SLIDE IMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Whole-slide images (WSIs) contain diagnostic information across magnifications, from tissue architecture to cellular morphology. Multi-scale MIL methods often encode patches from different magnifications independently and combine their representations *post hoc*, so the patch representations may not fully integrate the complementary information carried by different magnifications. Using both scales also requires fine patch encoding at inference. Cross-scale self-supervised learning (SSL) offers an alternative by introducing interaction during pretraining, but direct representation matching or fine-patch reconstruction does not account for *cross-scale asymmetry* because fine views contain details unavailable from coarse inputs. We present **Patch-JEPA**, a cross-scale latent predictive pretraining framework. A frozen fine-scale target encoder produces an ordered fine-scale latent field from aligned fine patches. Separate prediction branches map coarse representations to this field and an aggregated coarse-scale target, training the coarse-scale context encoder without direct cross-scale matching or fine-patch reconstruction. Fine-scale representations are used only as pretraining targets, and the retained context encoder uses only coarse patches for *coarse-only inference*. Patch-JEPA is competitive with standard SSL, cross-scale SSL, and post-hoc multi-scale WSI pipelines on four pathology benchmarks. On RCC, ridge probing yields $R^2 = 0.450$ for aligned fine-scale latent-field prediction, compared with 0.231 for Cross-MAE. Model-side inference cost remains close to a single-scale baseline under a fixed-bag timing protocol.

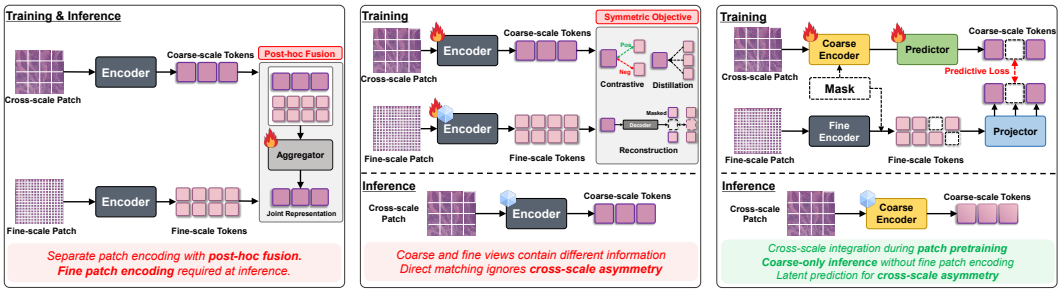
1 INTRODUCTION

Whole-slide image (WSI) analysis is a major task in computational pathology, supporting diagnosis, subtyping, and prognosis (Tizhoosh & Pantanowitz, 2018; Van der Laak et al., 2021; Cui & Zhang, 2021). WSIs contain diagnostically relevant information across multiple magnifications. Low magnification captures tissue architecture and spatial organization, whereas high magnification reveals cellular and nuclear morphology (Gurcan et al., 2009). Pathologists combine information across magnifications, using tissue-level context to interpret cellular patterns and cellular morphology to refine tissue-level interpretation (Bera et al., 2019; Chen et al., 2022b). We use *low/high magnification* to denote viewing levels and *coarse/fine* to describe their roles in representation learning.

Current WSI classification pipelines typically perform cross-scale integration after patch encoding. Multi- or cross-scale multiple instance learning (MIL) methods encode patches from different magnifications separately and combine their representations *post hoc* for slide-level prediction (Li et al., 2021; Zhang et al., 2022; Deng et al., 2024; Xiang et al., 2023). Other methods learn slide-level representations from fixed patch representations using self-supervised or unsupervised objectives (Chen et al., 2022b;a; Hou et al., 2024; Jaume et al., 2024). In *post-hoc cross-scale integration* (Figure 1(A)), coarse and fine patches are encoded independently before fusion. Coarse encoding can fail to preserve cues predictive of fine-scale morphology that cannot be recovered after the representation has been fixed. Using both magnifications downstream also requires cross-scale computation, including fine patch encoding, during inference.

Cross-scale supervision can instead be applied during patch pretraining, before slide-level modeling, so that aligned fine patches directly shape the coarse representation. This requires accounting for the relation between the two views. Coarse and fine views of the same location are spatially aligned

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Cross-scale MIL (b) Cross-scale SSL (c) Patch-JEPA

Figure 1: Cross-scale integration strategies. (a) Cross-scale MIL methods fuse separately encoded coarse and fine representations *post hoc*. (b) Cross-scale SSL couples aligned magnification views during pretraining using distillation, reconstruction, or contrastive objectives. (c) Patch-JEPA predicts fine-scale latent targets from coarse inputs and trains the retained coarse-scale context encoder for coarse-only inference.

but do not contain identical information. Cellular and nuclear details visible in the fine view are not all discernible in the coarse view, whereas the coarse view provides broader tissue context. We refer to this difference as *cross-scale asymmetry*.

Existing SSL objectives are not designed for this asymmetric relation (Figure 1(B)). Contrastive objectives bring the representations of the two views closer (Chhipa et al., 2023), which can emphasize information observable in both views. Distillation-based approaches match the coarse representation to a target derived from the fine view (Su et al., 2025), although the target may contain details that are difficult to infer from the coarse patch alone. Reconstruction objectives require the coarse view to reproduce fine patch content that it does not reveal. Cross-scale pretraining instead requires a target that captures fine-scale structure predictable from the coarse view.

A cross-scale pretraining objective for this setting should satisfy three requirements. **(1)** Information from fine patches should supervise coarse representation learning before downstream slide-level modeling, rather than enter only through post-hoc cross-scale integration. **(2)** Using fine patches during training should not require fine patch encoding at inference. **(3)** The objective should account for cross-scale asymmetry by learning fine-scale structure predictable from the coarse view rather than enforcing direct equality between coarse and fine representations.

Patch-JEPA is a cross-scale latent predictive pretraining framework for this setting. In Figure 1(C), the coarse-scale context encoder maps a masked coarse patch to a masked coarse representation. A frozen fine-scale target encoder encodes aligned fine patches from the same region, and the resulting fine representations are arranged in spatial order to form an ordered fine-scale latent field. The fine-scale prediction branch predicts this field from the masked coarse representation. The coarse-scale prediction branch predicts a stop-gradient coarse-scale target obtained by aggregating the field to coarse resolution. Both prediction losses train the coarse-scale context encoder. Section 2 describes the architecture and training objective.

(1) Both prediction losses supervise the coarse-scale context encoder during patch pretraining, so cues predictive of fine-scale morphology shape the coarse representation before slide-level modeling instead of entering only through post-hoc cross-scale integration. **(2)** Fine-scale representations are used only as pretraining targets, and the retained coarse-scale context encoder takes only coarse patches, so *coarse-only inference* requires no fine patch encoding. **(3)** Latent prediction handles cross-scale asymmetry by predicting latent targets from the masked coarse representation without directly matching coarse and fine representations or reconstructing fine patches. The fixed coarse-to-fine geometry preserves spatial correspondence across scales.

Our contributions are as follows.

- We shift cross-scale supervision from slide-level inference to patch pretraining, so aligned fine-scale information shapes the coarse representation before downstream WSI modeling.
- We introduce a one-to-many latent predictive objective for cross-scale asymmetry, predicting fine-scale structure from coarse representations without direct cross-scale matching or fine patch reconstruction.

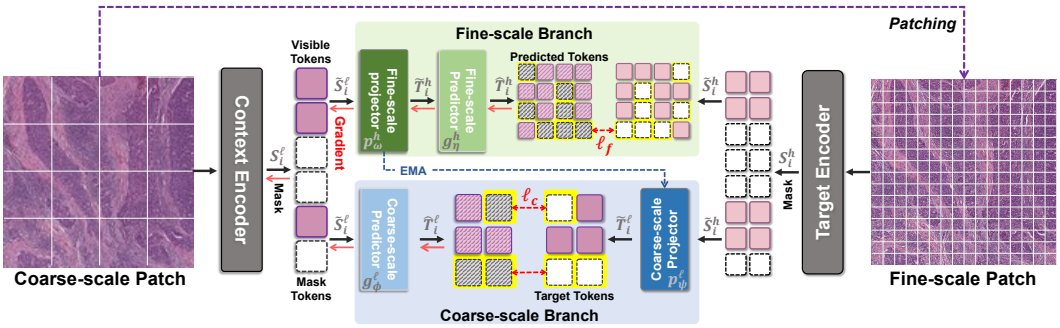


Figure 2: Patch-JEPA overview. During pretraining, a coarse-scale context encoder processes masked coarse patches and predicts aligned fine-scale latent targets from the fine-scale target encoder through the coarse-scale prediction branch and fine-scale prediction branch. The fine-scale prediction loss is evaluated only at fine tokens corresponding to masked coarse tokens. During inference, the slide-level model uses only the coarse representations.

- Across four pathology benchmarks, Patch-JEPA is competitive with standard SSL, cross-scale SSL, and WSI methods using explicit multi-scale inference, while supporting coarse-only inference without fine patch encoding or cross-scale computation.

2 METHOD

Framework overview. Figure 2 presents Patch-JEPA's cross-scale integration during pretraining for WSI analysis. Fine-scale supervision during pretraining shapes the representations of the coarse-scale context encoder to support prediction of aligned fine-scale structure for downstream slide-level models. A fine-scale target encoder supplies high-magnification latent targets. A coarse-scale prediction branch predicts an aggregated fine-scale latent target, while a fine-scale prediction branch predicts the ordered fine-scale latent field. Table 6 reports ablations of these components.

Cross-scale input formulation. For site i, the input is a coarse patch x_i^c and its aligned fine-patch field $X_i^f = [x_{i,j}^f]_{j=1}^{16}$, where each coarse patch aligns to 16 ordered fine patches under a fixed 1-to-16 relation. The two views are spatially aligned but informationally asymmetric. The fine scale contains details unavailable from the coarse scale alone. Here, *fine-scale structure* refers to aligned fine-scale information predictable from local coarse context under this fixed correspondence. Patch-JEPA does not reconstruct all fine-scale detail. Instead, it uses the predictable component to shape the coarse representation through latent prediction. In our experiments, the coarse and fine scales correspond to $1.25\times$ and $5\times$ magnification, respectively. Appendix B.2 describes patch extraction and alignment, and Appendix D gives implementation details.

2.1 LATENT ENCODERS

Fine-scale target encoder ($f_{\theta_T}^f$). Patch-JEPA uses an asymmetric encoder pair. The coarse-scale context encoder is retained for inference, whereas the frozen fine-scale target encoder defines the pretraining target space. For compactness, the main-text notation suppresses token-grid axes. S_i^c denotes a d-channel coarse token grid and S_i^f its aligned field of 16 fine token grids. In the reported configuration, the fine-scale target encoder is a pretrained Vision Transformer (ViT) whose weights θ_T remain frozen throughout pretraining. Unlike the standard JEPA target encoder, which is an exponential moving average (EMA) of the online encoder (Assran et al., 2023; Bardes et al., 2024), it operates at a different magnification and observes information unavailable to the coarse-scale prediction branch. Updating it by EMA with the coarse-scale context encoder could let its latent space drift with the coarse representation and absorb prediction errors. The fine-scale target encoder is kept as a fixed reference space, so training optimizes the coarse representation against predictable fine-scale targets rather than a moving target. Each fine patch is encoded as

$$S_i^f = \left[S_{i,j}^f\right]_{j=1}^{16}, \quad S_{i,j}^f = f_{\theta_T}^f(x_{i,j}^f). \tag{1}$$

The reported token-grid shapes are given in Appendix D.

Coarse-scale context encoder (f_θ^c). The coarse-scale context encoder maps the coarse patch x_i^c to a latent representation $S_i^c = f_\theta^c(x_i^c)$ that must support prediction of fine-scale structure predictable from local tissue context.

2.2 Cross-Scale Prediction Branches

Standard JEPA uses context and target from the same view. In Patch-JEPA, aligned coarse and fine views define the context and target, so we use a coarse-scale prediction branch and a fine-scale prediction branch. For a sample X with N coarse sites, let $\Omega_X \subseteq \{1, \dots, N\}$ denote the nonempty masked site set:

$$\widetilde{x}_i^c = \mathcal{M}_c(x_i^c; \mathbf{1}[i \in \Omega_X]), \quad \widetilde{S}_i^c = f_\theta^c(\widetilde{x}_i^c), \quad \widetilde{S}_i^f = \mathcal{M}_f(S_i^f; \mathbf{1}[i \in \Omega_X]), \quad i \in \{1, \dots, N\}. \tag{2}$$

Here, $\mathcal{M}_c$ corrupts selected coarse inputs before context encoding, whereas $\mathcal{M}_f$ selects the corresponding ordered fine-scale latent fields after fine patch encoding as targets. It does not apply token-level masking to the stop-gradient fine-scale targets. Ω_X determines the sites on which both losses are evaluated, while token/block masking occurs only within their coarse token grids. Appendix D summarizes the masking procedure and EMA-tied projector update.

For each selected site, let m_i^c denote its binary coarse token/block mask and let $m_i^f = \mathcal{U}_4(m_i^c)$ be its stride-4 expansion to the aligned fine-token grid, where $\mathcal{U}_4$ replicates each coarse-mask entry over its corresponding 4×4 fine-token block. The stop-gradient fine-scale target remains complete, but the fine-scale prediction loss is evaluated only at entries selected by m_i^f.

2.2.1 Fine-Scale Prediction Branch

Fine-scale projector (p_ω^f). The fine-scale prediction branch predicts the ordered fine-scale latent field from the masked coarse representation, while the stop-gradient fine-scale target is formed after fine patch encoding on the corresponding selected sites. Its projector maps the masked coarse representation into the aligned fine-token-grid geometry:

$$\widetilde{T}_i^f = p_\omega^f(\widetilde{S}_i^c). \tag{3}$$

The fine-scale projector maps the coarse representation into the aligned fine-token-grid geometry of the same site, so the pretraining signal is location-specific rather than pooled. Without the ordered fine-scale latent field, the objective could match pooled fine content without making prediction sensitive to the spatial arrangement of fine-scale structure that downstream WSI models may need. This projector is separate from the pretraining-only predictor g_η^f.

Fine-scale predictor (g_η^f). The fine-scale predictor maps the fine-scale projector output into the target latent space

$$\widehat{T}_i^f = g_\eta^f(\widetilde{T}_i^f). \tag{4}$$

The prediction is supervised against the stop-gradient fine-scale target $\widetilde{S}_i^f$. In the reported model, g_η^f is a token-wise linear layer applied independently at each fine-token location.

2.2.2 Coarse-Scale Prediction Branch

Coarse-scale projector (p_ψ^c). The coarse-scale prediction branch is the main training signal. It places the compression target directly on the retained coarse-scale context encoder by predicting the stop-gradient coarse-scale target, while the fine-scale prediction branch provides auxiliary supervision through an expanded pretraining head. Its projector maps the stop-gradient fine-scale target $\widetilde{S}_i^f$ to the stop-gradient coarse-scale target $\widetilde{T}_i^c$ using a Conv2D-based aggregation module

$$\widetilde{T}_i^c = p_\psi^c(\widetilde{S}_i^f), \quad i \in \{1, \dots, N\}, \qquad \psi \leftarrow \alpha\, \psi + (1 - \alpha)\, \mathcal{P}(\omega). \tag{5}$$

Its parameters ψ are updated by an exponential moving average of $\mathcal{P}(\omega)$ with momentum $\alpha \in [0, 1)$. This is possible because the transposed-convolution kernel and its Conv2D counterpart share the same weight shape up to channel permutation. The fine-scale projector expands one coarse representation into the aligned fine-token-grid geometry, and the coarse-scale projector summarizes

that layout back to coarse resolution. Here, $\mathcal{P}(\omega)$ denotes the channel-permuted transpose of the fine-scale projector weights used to form the Conv2D coarse-scale projector. This branch explicitly compresses the ordered fine-scale latent field back to the stop-gradient coarse-scale target, so the retained encoder is trained against an aggregated fine representation rather than the expanded field alone. The projector produces the stop-gradient coarse-scale target $\widetilde{T}_i^c$ from $\widetilde{S}_i^f$, against which the coarse-scale predictor is trained; EMA updates and stop-gradient prevent co-adaptation to the online coarse-scale predictor. Appendix D gives the tensor-axis convention.

Coarse-scale predictor (g_ϕ^c). The coarse-scale predictor maps the masked coarse representation to a coarse-scale prediction using a lightweight ViT predictor

$$\widehat{T}_i^c = g_\phi^c(\widetilde{S}_i^c). \tag{6}$$

This prediction is matched to the stop-gradient coarse-scale target $\widetilde{T}_i^c$ in ℓ_c. Both predictors are discarded after pretraining. Only the coarse-scale context encoder is used downstream.

2.3 Training Objective

Patch-JEPA is trained with a main coarse-scale prediction loss between $\widehat{T}_i^c$ and $\widetilde{T}_i^c$ and an auxiliary fine-scale prediction loss between $\widehat{T}_i^f$ and $\widetilde{S}_i^f$

$$\mathcal{L} = \mathbb{E}_{X\sim p(X)}\mathbb{E}_{\Omega_X\sim p(\cdot|X)}\left[\frac{1}{|\Omega_X|}\sum_{i\in\Omega_X}\left(\ell_c\Big(\widehat{T}_i^c, \mathrm{sg}(\widetilde{T}_i^c)\Big) + \lambda\,\ell_f\Big(\widehat{T}_i^f, \mathrm{sg}(\widetilde{S}_i^f); m_i^f\Big)\right)\right]. \tag{7}$$

Here, Ω_X denotes the coarse-site indices sampled from X for masking. For each $i \in \Omega_X$, the context-corruption operator $\mathcal{M}$ applies token/block masking within the coarse token grid to form $\widetilde{S}_i^c$; sites outside Ω_X do not contribute to the loss. We set $\lambda = 0.1$ so that fine-scale prediction is auxiliary to the coarse objective. For each selected site, $\ell_f(\cdot,\cdot;m_i^f)$ is mean absolute error computed only over fine-token entries selected by the stride-4 expansion of the coarse mask, while ℓ_c compares the predicted coarse-token grid $\widehat{T}_i^c$ with the stop-gradient coarse-scale target $\widetilde{T}_i^c$. The masked-subset fine-scale prediction loss is restricted to fine tokens corresponding to masked coarse tokens; the stop-gradient fine-scale target itself is not additionally masked.

We use mean absolute error for latent regression because multiple fine-scale configurations can match the same coarse context. The operator $\mathrm{sg}(\cdot)$ denotes stop-gradient. We optimize the coarse-scale context encoder θ, coarse-scale predictor ϕ, fine-scale predictor η and fine-scale projector ω.

2.4 Inference

At inference, only the coarse-scale context encoder is used. Given coarse patches $\{x_i^c\}_{i=1}^N$ from a WSI, the slide-level representation is computed as

$$s_{\mathrm{slide}} = \mathcal{H}\big(\{S_i^c\}_{i=1}^N\big)\,. \tag{8}$$

We use mean pooling for $\mathcal{H}$ and a fixed linear probe in the current evaluation to isolate the pretrained coarse representation. Cross-scale supervision is used only during pretraining, and no fine-scale input or test-time adaptation is required.

3 Experiments

We organize the empirical evaluation around the four claims introduced in Section 1:

- **RQ1.** Can pretraining against predictable fine-scale targets shape coarse representations, enabling competitive coarse-only inference against post-hoc multi-scale WSI pipelines?
- **RQ2.** Does latent prediction better match asymmetric coarse–fine pathology views than alignment, distillation, or reconstruction-based SSL objectives?
- **RQ3.** Are the gains due to correct coarse–fine correspondence rather than merely exposing JEPA to multiple magnification levels?
- **RQ4.** Do pretrained coarse representations support prediction of aligned fine-scale structure under coarse-only inference?

Table 1: Comparison on four datasets. Cross-scale MIL baselines use multi-scale inputs at inference, while slide-level foundation models use their published inference pipelines. Patch-JEPA uses cross-scale supervision only during pretraining. DS denotes trainable downstream parameters. ACC, AUROC, and F1 denote accuracy, area under the ROC curve, and macro-F1, respectively. The **best** and second-best values are marked.

		RCC			BRCA			UCEC			Camelyon16		
Method	DS	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1
Single-scale MIL													
ABMIL (Ilse et al., 2018)	0.6M	0.897±0.02	0.970±0.01	0.882±0.03	0.840±0.03	0.812±0.05	0.653±0.07	0.749±0.03	0.752±0.04	0.601±0.04	0.669±0.05	0.701±0.09	0.634±0.06
CLAM (Lu et al., 2021)	0.4M	0.896±0.03	0.971±0.01	0.882±0.03	0.840±0.05	0.810±0.05	0.672±0.05	0.760±0.03	0.760±0.04	0.603±0.04	0.664±0.05	0.706±0.08	0.625±0.06
TransMIL (Shao et al., 2021)	2.3M	0.885±0.03	0.966±0.02	0.867±0.04	0.837±0.02	0.742±0.05	0.599±0.06	0.762±0.04	0.748±0.07	0.564±0.10	0.662±0.07	0.677±0.09	0.577±0.13
Cross-scale MIL													
DSMIL (Li et al., 2021)	0.6M	0.895±0.02	0.979±0.01	0.872±0.02	0.858±0.04	0.860±0.05	0.716±0.05	0.780±0.05	0.760±0.06	0.553±0.09	0.717±0.11	0.757±0.10	**0.692±0.11**
MRAN (Xiang et al., 2023)	0.9M	0.884±0.01	0.980±0.01	0.863±0.02	0.848±0.03	0.863±0.05	0.728±0.04	0.762±0.04	0.775±0.04	**0.625±0.08**	0.664±0.09	0.709±0.10	0.629±0.11
CSMIL (Deng et al., 2024)	0.9M	0.897±0.02	0.980±0.01	0.885±0.03	0.858±0.03	0.862±0.05	**0.740±0.05**	0.751±0.05	0.767±0.05	0.586±0.10	0.651±0.07	0.684±0.08	0.610±0.07
MSPN (Wu et al., 2026)	4.8M	0.891±0.01	0.976±0.01	0.872±0.02	0.859±0.03	0.862±0.06	0.718±0.07	0.750±0.04	0.769±0.05	0.560±0.10	0.707±0.04	**0.775±0.03**	0.682±0.05
MARBLE (Dwarampudi et al., 2026)	2.4M	0.888±0.02	0.977±0.01	0.867±0.02	0.853±0.04	0.842±0.04	0.703±0.05	0.760±0.04	0.784±0.04	0.589±0.07	0.664±0.02	0.709±0.07	0.629±0.03
Cross-scale slide-level foundation models													
HIPT (Chen et al., 2022b)	3.1K	0.859±0.03	0.940±0.02	0.854±0.04	0.830±0.02	0.780±0.03	0.569±0.05	0.759±0.02	0.703±0.08	0.450±0.04	0.692±0.05	0.705±0.05	0.620±0.06
CHIEF (Wang et al., 2024)	6.1K	0.878±0.02	0.967±0.01	0.871±0.01	0.860±0.02	0.870±0.02	0.658±0.05	0.763±0.02	0.774±0.04	0.504±0.04	0.694±0.02	0.701±0.06	0.666±0.06
GigaPath (Xu et al., 2024)	12K	0.914±0.02	**0.984±0.01**	0.892±0.02	**0.866±0.03**	**0.895±0.03**	0.694±0.05	0.775±0.04	**0.787±0.03**	0.608±0.07	0.697±0.04	0.733±0.06	0.655±0.04
Patch-JEPA	1.5K	**0.918±0.01**	0.983±0.01	**0.906±0.02**	0.862±0.02	0.872±0.04	0.736±0.03	**0.782±0.05**	0.775±0.07	0.592±0.09	**0.719±0.03**	0.743±0.06	0.683±0.05

3.1 Experimental Settings

Datasets. We evaluate on TCGA-RCC, TCGA-BRCA, TCGA-UCEC, and the Camelyon16 benchmark. Dataset statistics and preprocessing are in Appendix B.

Training and evaluation details. We freeze the pretrained coarse-scale context encoder and train only a lightweight downstream head, while MIL baselines retain their original slide-level architectures. This keeps the downstream comparison focused on where cross-scale supervision enters the pipeline rather than on differences in slide-level readout capacity. For the TCGA cohorts, we use patient-level five-fold stratified cross-validation so that no patient contributes slides to more than one fold. For Camelyon16, we use five-fold slide-level splits because patient identifiers are not used in the same way. Self-supervised pretraining is performed once on unlabeled slides in each corpus. Slide-level labels and clinical endpoints are not used during self-supervised pretraining. Downstream evaluation follows the shared single-branch protocol. Detailed comparison settings, pretraining design choices, and implementation specifics are provided in Appendices C and D.

3.2 **RQ1.** System-Level Comparison

Setup. RQ1 compares complete Patch-JEPA and post-hoc multi-scale WSI pipelines at the system level. We include single-scale MIL (Ilse et al., 2018; Lu et al., 2021; Shao et al., 2021), post-hoc cross-scale MIL (Li et al., 2021; Xiang et al., 2023; Deng et al., 2024; Wu et al., 2026; Dwarampudi et al., 2026), and slide-level foundation model baselines (Chen et al., 2022b; Wang et al., 2024; Xu et al., 2024). Cross-scale MIL baselines use multi-scale inputs at inference, whereas slide-level foundation models use their published pipelines and weights. Patch-JEPA uses cross-scale supervision only during pretraining. MIL baselines use a pretrained ViT-S/16 patch encoder, and foundation baselines use published weights. Appendix C.2 provides full setup details and additional results, including survival and stage prediction.

Results. Table 1 provides a system-level comparison between post-hoc multi-scale inference and pretraining-time cross-scale shaping of coarse representations. Patch-JEPA remains competitive with post-hoc multi-scale inference pipelines, suggesting that predictable fine-scale targets effectively shape coarse patch representations during pretraining. Additional system-level results with ViT-Base backbones, in-domain SSL pretraining, and magnification extension are provided in Appendix C.2.3.

3.3 **RQ2.** Objective Comparison

Setup. RQ2 tests the objective-level hypothesis. Because aligned coarse and fine pathology views are related but asymmetric, they should not be treated as interchangeable augmentations or as fully reconstructable targets. In the main-text RQ2 comparison, we initialize the compared SSL backbones from ImageNet-1K weights and match the trainable encoder family, in-domain pretraining

data, downstream protocol, and shared evaluation path so that RQ2 isolates objective choice under the same downstream WSI evaluation setting. We reimplement all compared objectives under a shared protocol. The comparison includes single-scale SSL (Chen et al., 2020; Oquab et al., 2023; He et al., 2022), existing cross-scale SSL (Chhipa et al., 2023; Su et al., 2025), and Cross-MAE. Appendices C.3.1 and C.3.3 report additional results from ImageNet-1K-initialized and randomly initialized SSL encoders.

Table 2: Objective comparison under coarse-only inference; classification accuracy.

Method	RCC	BRCA	UCEC	Camelyon16
Single-scale objectives				
SimCLR (Chen et al., 2020)	0.630±0.02	0.841±0.03	0.749±0.02	0.616±0.03
DINOv2 (Oquab et al., 2023)	0.825±0.03	0.814±0.03	0.753±0.03	0.654±0.02
MAE (He et al., 2022)	0.589±0.02	0.822±0.03	0.757±0.03	0.602±0.01
Cross-scale objectives				
MPCS (Chhipa et al., 2023)	0.713±0.02	0.816±0.03	0.757±0.02	0.664±0.03
X-MAG (Su et al., 2025)	0.856±0.01	0.845±0.03	0.689±0.05	0.613±0.03
Cross-MAE	0.605±0.02	0.805±0.04	0.761±0.03	0.607±0.05
Patch-JEPA	**0.918**±0.01	**0.862**±0.02	**0.782**±0.05	**0.719**±0.04

Table 3: Effect of cross-scale supervision under coarse-only inference; classification accuracy.

Variant	RCC	BRCA	UCEC	Camelyon16
Single-scale				
I-JEPA (coarse)	0.771±0.02	0.817±0.03	0.720±0.02	0.637±0.04
I-JEPA (fine)	0.798±0.02	0.837±0.02	0.739±0.02	0.622±0.04
Cross-scale				
Joint	0.908±0.02	0.859±0.03	0.760±0.02	0.669±0.06
Unaligned	0.852±0.01	0.825±0.03	0.741±0.02	0.599±0.01
Patch-JEPA	**0.918**±0.01	**0.862**±0.02	**0.782**±0.05	**0.719**±0.04

Results. Table 2 shows that Patch-JEPA remains competitive overall, with cross-scale objectives generally outperforming single-view objectives. Cross-scale supervision alone is not sufficient. Objectives that align coarse and fine views or reconstruct fine-scale content remain mismatched to the information available in the coarse input, because many cellular details are not fully recoverable from low magnification alone. Patch-JEPA predicts the fine-scale latent structure that is conditionally predictable from coarse context, which better fits the asymmetric relation between the two views. These results support cross-scale latent prediction as a way to shift useful cross-scale information into coarse representations during pretraining.

3.4 **RQ3.** Effect of Cross-Scale Supervision

Setup. RQ3 asks whether Patch-JEPA improves simply because JEPA is exposed to multiple scales during pretraining, or because the model preserves the correct coarse–fine correspondence between them. Under the same in-domain pretraining data, encoder family, and coarse-only inference protocol, we compare four patch-pretraining variants. These are I-JEPA (coarse), I-JEPA (fine), an unaligned control that keeps the same scale exposure but deliberately breaks cross-scale alignment, and aligned joint-scale training (coarse and fine views as context). All RQ3 variants, including Joint and I-JEPA (fine), are evaluated using only coarse-scale slide inputs. The main table reports classification accuracy. Appendix C.4 gives the setup and baselines.

Results. Table 3 shows that aligned joint-scale training is the strongest non-Patch-JEPA baseline, while the unaligned control drops despite using the same two magnification levels. Simply turning JEPA into a multi-scale pretraining setup is not enough once coarse–fine correspondence is broken. The aligned-versus-unaligned comparison supports the claim that the gains come from preserving cross-scale correspondence rather than from JEPA seeing both coarse and fine patches during pretraining. Because every variant is read out from coarse inputs, this comparison concerns how the pretraining construction transfers to the same coarse-only inference path.

3.5 **RQ4.** Predicting Fine-Scale Structure from Coarse Representations

Setup. RQ4 asks whether the frozen coarse representation learned by Patch-JEPA supports prediction of its ordered fine-scale latent field without receiving fine-scale input at evaluation. We compare single-scale SSL baselines (SimCLR, MAE), cross-scale SSL baselines (MPCS, X-MAG), Cross-MAE, and Patch-JEPA under the same coarse-only evaluation setting. A ridge probe maps frozen coarse representations to the aligned fine-scale latent field. We report cosine similarity, R^2, Recall@1 (R@1), and Recall@5 (R@5). Table 4 reports TCGA-RCC results; Appendix C.5 gives the probe construction and additional RCC analyses.

Results. Patch-JEPA achieves the strongest aligned fine-scale latent-field prediction in Table 4, with the highest cosine similarity, R^2, Recall@1, and Recall@5. These results show that coarse representations pretrained with aligned cross-scale latent targets retain fine-scale information that

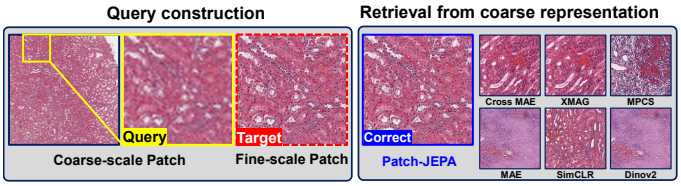


Figure 3: Qualitative RQ4 example on BRCA. The ridge probe predicts aligned fine-scale structure from the coarse representation. On BRCA, the predicted fine-scale structure follows the spatial organization of the aligned target.

Table 4: Aligned fine-scale latent-field prediction on RCC. A ridge probe maps frozen coarse representations to aligned fine-scale latent fields.

Method	Cosine ↑	R^2 ↑	R@1 ↑	R@5 ↑
SimCLR	0.739	0.187	13.6%	51.8%
MAE	0.743	0.200	14.4%	53.1%
MPCS	0.729	0.161	12.8%	50.3%
X-MAG	0.732	0.169	12.9%	50.2%
Cross-MAE	0.754	0.231	16.3%	55.9%
Patch-JEPA	**0.830**	**0.450**	**18.0%**	**60.4%**

supports position-sensitive prediction and retrieval under coarse-only inference. Figure 3 provides a qualitative example. Same-slide and same-patient exclusion tests and shuffled-order controls are reported in Appendix C.5.4.

3.6 Auxiliary Analyses

Cross-domain transfer. Table 5 compares two transfer settings. Cross-cohort transfer from Camelyon16 to TCGA-RCC combines cohort, institution, scanner, and task shifts; cross-organ transfer from RCC to BRCA evaluates transfer across organs within TCGA. Patch-JEPA is strongest in both settings and uses aligned fine-scale supervision during pretraining to learn morphology predictable from coarse context.

Table 5: Cross-domain results.

Method	Cross-cohort ACC	Cross-cohort AUROC	Cross-organ ACC	Cross-organ AUROC
SimCLR	0.631±0.04	0.708±0.06	0.825±0.03	0.615±0.04
MAE	0.571±0.03	0.649±0.05	0.825±0.03	0.659±0.03
MPCS	0.593±0.02	0.697±0.05	0.824±0.03	0.641±0.04
Cross-MAE	0.602±0.02	0.688±0.05	0.822±0.03	0.653±0.04
Patch-JEPA	**0.737**±0.02	**0.856**±0.02	**0.843**±0.02	**0.811**±0.06

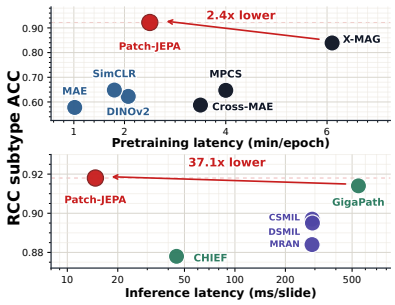


Figure 4: Efficiency on TCGA-RCC. Top pretraining time. Bottom model-side inference time.

Efficiency. Figure 4 compares RCC subtype accuracy with pretraining time and model-side inference cost. **Top.** Patch-JEPA moves cross-scale computation from inference to pretraining. **Bottom.** Post-hoc multi-scale methods require fine patch encoding and coarse–fine fusion at every inference pass, which raises model-side inference cost. **Patch-JEPA.** Aligned fine-scale supervision during pretraining incorporates fine-scale morphology into the coarse representation, which is used with a lightweight slide-level head downstream. Appendix C.6 gives the full timing protocol.

Module ablation. Table 6 ablates the main components of Patch-JEPA. The full Patch-JEPA model performs best. Within the reported design, updating the fine-scale target encoder by EMA instead of keeping it frozen lowers performance. The coarse input lacks fine-scale detail, so the frozen fine-scale target encoder provides a fixed reference space for learning fine-scale structure predictable from coarse context. An EMA target partially follows the online coarse representation and lets the fine-scale latent target drift with the coarse-scale prediction branch, weakening the intended cross-scale asymmetry. Removing the coarse-scale prediction branch weakens direct supervision on the retained coarse representation. Removing the fine-scale prediction branch removes aligned fine-scale prediction. Removing the projectors removes the coarse–fine geometry-aware latent mapping. Removing the predictor weakens latent-target prediction.

Table 6: Ablation on BRCA and UCEC.

Variant	BRCA ACC	BRCA AUROC	UCEC ACC	UCEC AUROC
EMA fine-scale encoder	0.853±0.01	0.863±0.02	0.758±0.02	0.722±0.03
w/o coarse-scale branch	0.851±0.01	0.856±0.02	0.773±0.03	0.773±0.06
w/o fine-scale branch	0.851±0.01	0.844±0.02	0.747±0.04	0.752±0.06
w/o projectors	0.847±0.01	0.836±0.02	0.760±0.03	0.718±0.07
w/o predictor	0.848±0.01	0.840±0.02	0.759±0.03	0.727±0.06
Patch-JEPA	**0.862**±0.02	**0.872**±0.04	**0.782**±0.05	**0.775**±0.07

4 Related Work

Multi-scale WSI learning. Multi-scale WSI methods often encode patches at different magnifications and combine the resulting representations during slide-level modeling. MIL approaches perform this combination through dual-stream aggregation, hierarchical attention, cross-scale interaction, graph-based modeling, or progressive pyramidal encoding (Li et al., 2021; Zhang et al., 2022; Deng et al., 2024; Xiang et al., 2023; Bontempo et al., 2024; Wu et al., 2026; Dwarampudi et al., 2026). Slide-level representation methods also build WSI representations from encoded patches using hierarchical self-supervision, retrieval objectives, or unsupervised prototyping (Chen et al., 2022b;a; Song et al., 2024; Jaume et al., 2024). In both cases, cross-scale interaction occurs after patch representations have been formed. We move this interaction into patch pretraining, where aligned fine-scale information supervises the coarse representation before slide-level modeling.

Cross-scale representation learning in pathology. Pathology foundation models learn transferable patch representations from large pathology corpora (Filiot et al., 2023; Chen et al., 2024; Zimmermann et al., 2024; Filiot et al., 2024). Cross-scale SSL additionally uses relationships between magnifications during patch pretraining. Contrastive methods encourage agreement between aligned views (Chhipa et al., 2023), whereas cross-magnification distillation transfers target representations across magnifications (Su et al., 2025). These objectives directly couple representations from views that differ in observable detail. Patch-JEPA formulates the coarse-to-fine relation as latent prediction, using fine-scale targets to supervise the coarse representation without direct representation matching or fine-patch reconstruction.

Joint-embedding predictive architectures. JEPA learns representations by predicting latent targets rather than reconstructing input pixels (LeCun, 2022). I-JEPA applies latent prediction to images (Assran et al., 2023), with extensions to video, 3D data, and neural signals (Bardes et al., 2024; Assran et al., 2025; Hu et al., 2024; Dong et al., 2024). Frozen-target variants modify how the target representation is defined (Li et al., 2025; Radhachandran et al., 2026), and CrossJEPA applies latent prediction across modalities (Perera et al., 2025). Patch-JEPA considers a different setting. Spatially aligned coarse and fine pathology views contain different observable information and are not both available at inference. The context is a coarse patch, the target is derived from its aligned fine-patch field, and only the coarse-scale context encoder is retained for downstream inference.

5 Discussion

The objective comparison and alignment ablation point to a common principle: the coarse encoder benefits from fine-scale supervision when the target preserves spatial correspondence and focuses on structure predictable from the coarse view. The fine-scale prediction probe supports this interpretation by showing that the frozen coarse representation retains information about the aligned fine-scale latent field. At the system level, using fine views during pretraining allows Patch-JEPA to use a coarse-only downstream path while remaining competitive with post-hoc multi-scale pipelines.

Limitations and future directions. Patch-JEPA requires aligned coarse–fine regions during pretraining, and coarse-only inference cannot recover cellular details absent from the coarse view. These limits motivate extending latent prediction to broader WSI context and investigating selective fine-scale analysis when a coarse view is insufficient. Evaluation on larger pretraining corpora, backbones, and multi-institutional cohorts would clarify how this approach scales and transfers.

Conclusion. Patch-JEPA learns from aligned fine-scale latent targets while retaining a coarse encoder for downstream WSI analysis. Across four pathology benchmarks, the complete pipeline achieves competitive system-level performance without requiring fine-scale inputs at inference. The objective and alignment analyses further indicate that spatially matched coarse–fine supervision helps the coarse representation support downstream tasks and fine-scale target prediction. These findings make cross-scale pretraining a promising complement to slide-level multi-scale modeling when repeated high-magnification processing is costly.

AI USE STATEMENT

We used generative AI tools, including OpenAI Codex, to assist with experimental-design feedback, implementation and debugging of model and analysis code, construction of figures and tables, and interpretation of results. These tools also assisted with literature discovery, citation checks, translation, and drafting and editing parts of the manuscript. The authors decided which suggestions and experiments to adopt, reviewed the AI-assisted outputs, and take responsibility for the final methods, results, claims, and text. .

ETHICS STATEMENT

This study uses publicly available whole-slide image datasets and does not involve new data collection or direct interaction with human participants. Patch-JEPA supports computational pathology research through cross-scale representation learning from aligned whole-slide image views.

REPRODUCIBILITY STATEMENT

Section 2 describes the Patch-JEPA architecture, target construction, masking procedure, training objective, and inference protocol. Appendix B describes the datasets, data splits, preprocessing, and aligned coarse–fine patch extraction. Appendix C specifies the downstream evaluation protocols, baseline settings, and experiment-specific procedures. Appendix D provides the implementation details, including tensor shapes, projector and predictor architectures, optimization hyperparameters, EMA updates, and the complete pretraining algorithm. Additional experiments, ablations, and the compute and timing protocol are reported in the appendix. Code and experiment configurations will be released publicly upon acceptance.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499. URL https://doi.org/10.1109/CVPR52729.2023.01499.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2.

Kaustav Bera, Kurt A. Schalper, David L. Rimm, Vamsidhar Velcheti, and Anant Madabhushi. Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology. *Nature Reviews Clinical Oncology*, 16(11):703–715, 2019. doi: 10.1038/s41571-019-0252-y. URL https://doi.org/10.1038/s41571-019-0252-y.

Gianpaolo Bontempo, Federico Bolelli, Angelo Porrello, Simone Calderara, and Elisa Ficarra. A graph-based multi-scale approach with knowledge distillation for wsi classification. *IEEE Transactions on Medical Imaging*, 43(4):1412–1421, 2024. doi: 10.1109/TMI.2023.3337549. URL https://doi.org/10.1109/TMI.2023.3337549.

Chengkuan Chen, Ming Y. Lu, Drew F. K. Williamson, Tiffany Y. Chen, Andrew J. Schaumberg, and Faisal Mahmood. Fast and scalable search of whole-slide images via self-supervised

deep learning. *Nature Biomedical Engineering*, 6(12):1420–1434, 2022a. doi: 10.1038/s41551-022-00929-8. URL https://doi.org/10.1038/s41551-022-00929-8.

Richard J. Chen, Chengkuan Chen, Yicong Li, Tiffany Y. Chen, Andrew D. Trister, Rahul G. Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16123–16134, 2022b. doi: 10.1109/CVPR52688.2022.01567. URL https://doi.org/10.1109/CVPR52688.2022.01567.

Richard J. Chen, Tong Ding, Ming Y. Lu, Drew F. K. Williamson, Guillaume Jaume, Andrew H. Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, Mane Williams, Lukas Oldenburg, Luca L. Weishaupt, Judy J. Wang, Anurag Vaidya, Long Phi Le, Georg Gerber, Sharifa Sahai, Walt Williams, and Faisal Mahmood. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3):850–862, 2024. doi: 10.1038/s41591-024-02857-3. URL https://doi.org/10.1038/s41591-024-02857-3.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607, 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Prakash Chandra Chhipa, Richa Upadhyay, Gustav Grund Pihlgren, Rajkumar Saini, Seiichi Uchida, and Marcus Liwicki. Magnification prior: a self-supervised method for learning representations on breast cancer histopathological images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2717–2727, 2023.

Miao Cui and David Y Zhang. Artificial intelligence and computational pathology. *Laboratory Investigation*, 101(4):412–422, 2021.

Ruining Deng, Can Cui, Lucas W. Remedios, Shunxing Bao, R. Michael Womick, Sophie Chiron, Jia Li, Joseph T. Roland, Ken S. Lau, Qi Liu, Keith T. Wilson, Yaohong Wang, Lori A. Coburn, Bennett A. Landman, and Yuankai Huo. Cross-scale multi-instance learning for pathological image diagnosis. *Medical Image Analysis*, 94:103124, 2024. doi: 10.1016/j.media.2024.103124. URL https://doi.org/10.1016/j.media.2024.103124.

Zijian Dong, Ruilin Li, Yilei Wu, Thuan Tinh Nguyen, Joanna Su Xian Chong, Fang Ji, Nathanael Ren Jie Tong, Christopher Li Hsian Chen, and Juan Helen Zhou. Brain-JEPA: Brain dynamics foundation model with gradient positioning and spatiotemporal masking. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2732. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9c3828adf1500f5de3c56f6550dfe43c-Abstract-Conference.html.

Jagan Mohan Reddy Dwarampudi, Joshua Wong, Hien Van Nguyen, and Tania Banerjee. A multi-scale linear-time encoder for whole-slide image analysis. In *2026 IEEE 23rd International Symposium on Biomedical Imaging (ISBI)*, pp. 1–4, 2026. doi: 10.1109/ISBI61048.2026.11516046. URL https://doi.org/10.1109/ISBI61048.2026.11516046.

Alexandre Filiot, Ridouane Ghermi, Antoine Olivier, Paul Jacob, Lucas Fidon, Axel Camara, Alice Mac Kain, Charlie Saillard, and Jean-Baptiste Schiratti. Scaling self-supervised learning for histopathology with masked image modeling. *medRxiv*, 2023. doi: 10.1101/2023.07.21.23292757. URL https://doi.org/10.1101/2023.07.21.23292757.

Alexandre Filiot, Paul Jacob, Alice Mac Kain, and Charlie Saillard. Phikon-v2, a large and public feature extractor for biomarker prediction, 2024. URL https://arxiv.org/abs/2409.09173.

Metin N Gurcan, Laura E Boucheron, Ali Can, Anant Madabhushi, Nasir M Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://doi.org/10.1109/CVPR52688.2022.01553.

Xinhai Hou, Cheng Jiang, Akhil Kondepudi, Yiwei Lyu, Asadur Chowdury, Honglak Lee, and Todd C. Hollon. A self-supervised framework for learning whole slide representations, 2024. URL https://arxiv.org/abs/2402.06188.

Naiwen Hu, Haozhe Cheng, Yifan Xie, Shiqi Li, and Jihua Zhu. 3D-JEPA: A joint embedding predictive architecture for 3D self-supervised representation learning, 2024. URL https://arxiv.org/abs/2409.15803.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2127–2136, 2018. URL https://proceedings.mlr.press/v80/ilse18a.html.

Guillaume Jaume, Anurag Vaidya, Andrew Zhang, Andrew H. Song, Richard J. Chen, Sharifa Sahai, Dandan Mo, Emilio Madrigal, Long Phi Le, and Faisal Mahmood. Multistain pretraining for slide representation learning in pathology. In *Computer Vision – ECCV 2024*, pp. 19–37, 2024. doi: 10.1007/978-3-031-73414-4_2. URL https://doi.org/10.1007/978-3-031-73414-4_2.

Yann LeCun. A path towards autonomous machine intelligence, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

Bin Li, Yin Li, and Kevin W. Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323, 2021. doi: 10.1109/CVPR46437.2021.01409. URL https://doi.org/10.1109/CVPR46437.2021.01409.

Xianhang Li, Chen Huang, Chun-Liang Li, Eran Malach, Josh Susskind, Vimal Thilak, and Etai Littwin. Rethinking JEPA: Compute-efficient video ssl with frozen teachers, 2025. URL https://arxiv.org/abs/2509.24317.

Ming Y. Lu, Drew F. K. Williamson, Tiffany Y. Chen, Richard J. Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5(6):555–570, 2021. doi: 10.1038/s41551-020-00682-w. URL https://doi.org/10.1038/s41551-020-00682-w.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Yann LeCun. DINOv2: Learning robust visual features without supervision, 2023. URL https://arxiv.org/abs/2304.07193.

Avishka Perera, Kumal Hewagamage, Saeedha Nazar, Kavishka Abeywardana, Hasitha Gallella, Ranga Rodrigo, and Mohamed Afham. CrossJEPA: Cross-modal joint-embedding predictive architecture for efficient 3D representation learning from 2D images, 2025. URL https://arxiv.org/abs/2511.18424.

Ashwath Radhachandran, Vedrana Ivezić, Shreeram Athreya, Corey W. Arnold, and William Speier. US-JEPA: A joint embedding predictive architecture for ultrasound, 2026. URL https://arxiv.org/abs/2602.19322.

Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, and Yongbing Zhang. TransMIL: Transformer based correlated multiple instance learning for whole slide image classification. In *Advances in Neural Information Processing Systems*, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/10c272d06794d3e5785d5e7c5356e9ff-Abstract.html.

Andrew H. Song, Richard J. Chen, Tong Ding, Drew F. K. Williamson, Guillaume Jaume, and Faisal Mahmood. Morphological prototyping for unsupervised slide representation learning in computational pathology. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11566–11578, 2024. doi: 10.1109/CVPR52733.2024.01099. URL https://doi.org/10.1109/CVPR52733.2024.01099.

Ziyu Su, Abdul Rehman Akbar, Usama Sajjad, Anil V. Parwani, and Muhammad Khalid Khan Niazi. Streamline pathology foundation model by cross-magnification distillation, 2025. URL https://arxiv.org/abs/2509.23097.

Hamid Reza Tizhoosh and Liron Pantanowitz. Artificial intelligence and digital pathology: challenges and opportunities. *Journal of Pathology Informatics*, 9(1):38, 2018.

Jeroen Van der Laak, Geert Litjens, and Francesco Ciompi. Deep learning in histopathology: the path to the clinic. *Nature Medicine*, 27(5):775–784, 2021.

Xiyue Wang, Junhan Zhao, Eliana Marostica, Wei Yuan, Jietian Jin, Jiayu Zhang, Ruijiang Li, Hongping Tang, Kanran Wang, Yu Li, Fang Wang, Yulong Peng, Junyou Zhu, Jing Zhang, Christopher R. Jackson, Jun Zhang, Deborah Dillon, Nancy U. Lin, Lynette Sholl, Thomas Denize, David Meredith, Keith L. Ligon, Sabina Signoretti, Shuji Ogino, Jeffrey A. Golden, MacLean P. Nasrallah, Xiao Han, Sen Yang, and Kun-Hsing Yu. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, 634(8035):970–978, 2024. doi: 10.1038/s41586-024-07894-z. URL https://doi.org/10.1038/s41586-024-07894-z.

Bryan Wong, Jong Woo Kim, Huazhu Fu, and Mun Yong Yi. Few-shot learning from gigapixel images via hierarchical vision-language alignment and modeling. In *Proceedings of the 39th Conference on Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=ZC2rbIYWfy.

Shuyang Wu, Yifu Qiu, Ines P. Nearchou, Sandrine Prost, Jonathan A. Fallowfield, Hakan Bilen, and Timothy J. Kendall. Enabling progressive whole-slide image analysis with multi-scale pyramidal network, 2026. URL https://arxiv.org/abs/2602.01951.

Hangchen Xiang, Junyi Shen, Qingguo Yan, Meilian Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Multi-scale representation attention based deep multiple instance learning for gigapixel whole slide image analysis. *Medical Image Analysis*, 89:102890, 2023. doi: 10.1016/j.media.2023.102890. URL https://doi.org/10.1016/j.media.2023.102890.

Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, Yanbo Xu, Mu Wei, Wenhui Wang, Shuming Ma, Furu Wei, Jianwei Yang, Chunyuan Li, Jianfeng Gao, Jaylen Rosemon, Tucker Bower, Soohee Lee, Roshanthi Weerasinghe, Bill J. Wright, Ari Robicsek, Brian Piening, Carlo Bifulco, Sheng Wang, and Hoifung Poon. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 630(8015):181–188, 2024. doi: 10.1038/s41586-024-07441-w. URL https://doi.org/10.1038/s41586-024-07441-w.

Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E. Coupland, and Yalin Zheng. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18780–18790, 2022. doi: 10.1109/CVPR52688.2022.01824. URL https://doi.org/10.1109/CVPR52688.2022.01824.

Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, David Klimstra, Razik Yousfi, Thomas Fuchs, Nicolo Fusi, Siqi Liu, and Kristen Severson. Virchow2: Scaling self-supervised mixed magnification models in pathology, 2024. URL https://arxiv.org/abs/2408.00738.

A APPENDIX

NOTATION SUMMARY

Table 7: Core notation used throughout the Patch-JEPA formulation.

Symbol	Meaning
Inputs and indexing	
c, f	Coarse and fine scales, respectively.
i, j	Coarse-site index and fine-patch index within the aligned 4×4 fine-patch grid.
x_i^c	Coarse patch at site i.
$x_{i,j}^f$	Fine patch at fine location j aligned to coarse site i.
$X_i^f = [x_{i,j}^f]_{j=1}^{16}$	Complete fine-patch field aligned to x_i^c, indexed in fixed row-major fine-patch-grid order.
$X = \{(x_i^c, X_i^f)\}_{i=1}^N$	One sample consisting of N aligned coarse–fine sites.
Ω_X	Masked coarse-site index set sampled for training sample X, and both loss terms are evaluated only for $i \in \Omega_X$.
N	Number of coarse sites (i.e., WSI-level crop locations) in one training sample, not the number of internal ViT tokens per crop.
d	Latent representation dimension.
Encoders and latent states	
f_θ^c	Trainable coarse-scale context encoder retained for downstream inference.
$f_{\theta_T}^f$	Frozen fine-scale target encoder used only in pretraining to define a fixed reference space rather than a co-adapting EMA target.
$\widetilde{x}_i^c = \mathcal{M}_c(x_i^c; \mathbf{1}[i \in \Omega_X])$	Corrupted coarse input produced before context encoding for selected sites.
S_i^c	Coarse-scale context encoder output for crop x_i^c, with shape $\mathbb{R}^d$. In the reported implementation, it is carried by a $14 \times 14 \times d$ token grid before downstream pooling.
$S_{i,j}^f$	Fine-scale target encoder output for fine patch $x_{i,j}^f$, with shape $\mathbb{R}^d$. In the reported implementation, it is carried by a $14 \times 14 \times d$ token grid.
$S_i^f = [S_{i,j}^f]_{j=1}^{16}$	Ordered fine-scale latent field of 16 fine representations in fixed row-major order, with shape $\mathbb{R}^{16 \times d}$. In the reported implementation, it corresponds to a $56 \times 56 \times d$ fine-token grid after arranging the 16 aligned grids.
$\widetilde{S}_i^f = \mathcal{M}_f(S_i^f; \mathbf{1}[i \in \Omega_X])$	Stop-gradient fine-scale target formed by applying the selected-site mask after fine patch encoding, and in the reported implementation it shares the same $56 \times 56 \times d$ grid as S_i^f.
m_i^c, m_i^f	Binary coarse token/block mask and its stride-4 expansion to the aligned fine-token grid. The latter selects the support of the masked-subset fine-scale prediction loss.
$\widetilde{S}_i^c = f_\theta^c(\widetilde{x}_i^c)$	Coarse representation obtained by encoding the corrupted coarse input. In the reported implementation, it is carried by a $14 \times 14 \times d$ token grid.
Projectors, predictors, and targets	
p_ω^f	Fine-scale projector that expands $\widetilde{S}_i^c$ to the aligned fine-token-grid geometry via stride-4 transposed convolution in the reported implementation, keeping fine-scale supervision location-specific rather than pooled.
$\widetilde{T}_i^f = p_\omega^f(\widetilde{S}_i^c)$	Fine-scale projector output before the fine-scale predictor, with shape $\mathbb{R}^{56 \times 56 \times d}$ in the reported implementation.
g_η^f	Fine-scale predictor used only in pretraining.
$\widehat{T}_i^f = g_\eta^f(\widetilde{T}_i^f)$	Predicted fine-scale latent field matched to $\widetilde{S}_i^f$ for masked coarse sites $i \in \Omega_X$, with shape $\mathbb{R}^{56 \times 56 \times d}$ in the reported implementation.
p_ψ^c	Coarse-scale projector that aggregates the stop-gradient fine-scale target back to coarse resolution via convolution aligned with p_ω^f, defining the stop-gradient coarse-scale target for the retained encoder.
$\widetilde{T}_i^c = p_\psi^c(\widetilde{S}_i^f)$	Stop-gradient coarse-scale target derived from the stop-gradient fine-scale target, and it is the main target matched by the coarse-scale prediction branch supervising the retained encoder. In the reported implementation, the shape is $\mathbb{R}^{14 \times 14 \times d}$.
$\mathcal{P}(\omega)$	Channel-permuted transpose of the fine-scale projector weights ω, used to form the Conv2D coarse-scale projector update.
g_ϕ^c	Coarse-scale predictor used only in pretraining.
$\widehat{T}_i^c = g_\phi^c(\widetilde{S}_i^c)$	Coarse-scale prediction matched to $\widetilde{T}_i^c$ for masked coarse sites $i \in \Omega_X$; in the reported implementation, shape $\mathbb{R}^{14 \times 14 \times d}$.
Objective and inference	
$\mathcal{M}_c(\cdot;\cdot)$	Coarse-input corruption operator applied before context encoding for selected sites $i \in \Omega_X$.
$\mathcal{M}_f(\cdot;\cdot)$	Target-side site-mask operator applied after fine patch encoding to form $\widetilde{S}_i^f$.
$\mathrm{sg}(\cdot)$	Stop-gradient operator in the pretraining loss.
ℓ_c, ℓ_f	Coarse- and fine-scale prediction losses, where ℓ_c is the main compression loss on the retained encoder and ℓ_f is an auxiliary fine-scale prediction loss. ℓ_c is mean absolute error on the coarse-token grid; ℓ_f is mean absolute error computed only on fine-token entries selected by m_i^f, for masked coarse sites $i \in \Omega_X$.
λ	Weight on the fine-scale prediction loss ℓ_f; set to 0.1 in the reported configuration.
α	EMA momentum used to update ψ from $\mathcal{P}(\omega)$.
$\mathcal{L}$	Total Patch-JEPA pretraining objective.
$\mathcal{H}$	Slide-level aggregation operator applied to coarse representations at inference.

Table 8: External dataset and implementation sources used in the reported experiments.

Item	Source
Dataset	
TCGA	GDC Data Portal (https://gdc.cancer.gov/access-data/gdc-data-portal); download client: https://github.com/NCI-GDC/gdc-client
Camelyon16	CAMELYON16 Grand Challenge (https://camelyon16.grand-challenge.org/)
Single-scale MIL	
ABMIL	https://github.com/AMLab-Amsterdam/AttentionDeepMIL
CLAM	https://github.com/mahmoodlab/CLAM
TransMIL	https://github.com/szc19990412/TransMIL
Cross-scale MIL	
CSMIL	https://github.com/hrlblab/CS-MIL
DSMIL	https://github.com/binli123/dsmil-wsi
MRAN	https://github.com/xhangchen/MRAN
MSPN	https://arxiv.org/abs/2602.01951
MARBLE	https://arxiv.org/abs/2602.02918
Cross-scale SSL	
X-MAG	https://github.com/AI4Path-Lab/XMAG
MPCS	https://github.com/prakashchhipa/Magnification-Prior-Self-Supervised-Method
Slide-level foundation models	
HIPT	https://github.com/mahmoodlab/HIPT
CHIEF	https://github.com/hms-dbmi/CHIEF
GigaPath	https://github.com/prov-gigapath/prov-gigapath

B Dataset and Preprocessing Details

B.1 Dataset Descriptions

We use three public TCGA WSI datasets, RCC, BRCA, and UCEC,[1] together with Camelyon16 for metastasis classification. Table 9 summarizes the number of labeled slides and extracted hierarchical patches used in our experiments.

Table 9: Summary of the WSI datasets. Coarse- and fine-patch counts use the reported $1.25\times \rightarrow 5\times$ hierarchy.

Dataset	Subtype (Abbr.)	# Slides	# Patches (coarse)	# Patches (fine)
TCGA-RCC	Clear Cell Renal Cell Carcinoma (CCRCC)	518	44,606	713,696
	Papillary Renal Cell Carcinoma (PRCC)	299	25,561	408,976
	Chromophobe Renal Cell Carcinoma (CHRCC)	121	10,536	168,576
TCGA-BRCA	Invasive Ductal Carcinoma (IDC)	747	53,870	861,920
	Invasive Lobular Carcinoma (ILC)	159	10,801	172,816
TCGA-UCEC	Endometrioid Carcinoma	400	39,582	633,312
	Serous Carcinoma	126	11,653	186,448
Camelyon16	Normal	239	22,401	358,416
	Tumor	160	20,286	324,576

B.2 Preprocessing and Hierarchical Patch Extraction

Each WSI is first processed to identify tissue-containing regions and remove background areas, and patch extraction is performed only within the retained tissue regions. Following a HiVE-MIL-style coarse–fine patching convention (Wong et al., 2025), we define crop coordinates in level-0 WSI pixel space and bind each accepted coarse site to its aligned fine patches as one structured coarse–fine unit. In the main reported setup, the target nominal magnifications are $1.25\times$ and $5\times$. For each nominal magnification, we select the nearest available pyramid level using the slide's objective-power metadata and pyramid downsample factors, then bilinearly resize the extracted crop to the fixed network input size when the available level does not exactly match the target sampling. For each tissue-containing coarse-scale location, we identify one level-0 bounding box and read two

[1] https://portal.gdc.cancer.gov/

views from the same field of view. These are a 224×224 coarse patch at $1.25\times$ and an aligned 896×896 fine-scale region at $5\times$. The fine-scale region is divided into a fixed 4×4 grid of 224×224 fine patches, yielding 16 fine patches per coarse site. We use row-major ordering with top-left origin, so if (r, c) denotes a zero-based fine-patch-grid coordinate, then $j = 4r + c + 1$, and we write the resulting aligned fine-patch field as $X_i^f = [x_{i,j}^f]_{j=1}^{16}$. This ordered field lets the fine-scale prediction branch supervise spatial arrangement rather than only pooled fine content. With this construction, every retained fine patch traces back to a unique coarse patch, and the coarse patch and fine-scale region share the same underlying level-0 bounding box.

The 16 elements of the fine-patch field refer to WSI-level fine-scale crops, not internal ViT tokens. Each 224×224 fine patch may be further decomposed into an internal token grid by the encoder. Equivalently, the implementation may materialize the aligned fine view as one larger crop and partition its internal token grid into 16 fine-patch-aligned regions. This distinction separates the data-level coarse–fine hierarchy from the encoder-level tokenization. Coarse sites are sampled from tissue-containing regions, and the implementation carries a fine-location validity mask after tissue/background filtering and slide-boundary checks so that invalid fine locations are excluded without changing the deterministic fine index ordering. For the TCGA cohorts, folds are split at the patient level so that no patient contributes slides to more than one fold. For Camelyon16, folds are split at the slide level because patient identifiers are not used in the same way. Subtype classification is evaluated for RCC, BRCA, and UCEC, while Camelyon16 is evaluated as a metastasis classification task; the TCGA cohorts also use stage and survival tasks when the corresponding clinical annotations are available.

C Downstream and RQ-Specific Setup Details

Unless otherwise noted, the controlled pretraining comparisons in RQ2–RQ3 and the Patch-JEPA side of RQ1 use in-domain pathology pretraining. RQ1 also includes MIL baselines evaluated with a shared frozen ViT-S/16 patch encoder and slide-level foundation model baselines evaluated with their published weights and pipelines. We first describe the shared downstream protocol, then give the setup choices specific to each research question.

C.1 Shared Downstream Protocol

Self-supervised pretraining uses all available unlabeled WSIs in each corpus. Slide-level labels and clinical endpoints are introduced after the downstream fold split for supervised probing, validation, and model selection.

Linear-probing protocol. Main downstream evaluations use a unified linear-probing entry point. For both classification and survival, we freeze the pretrained patch encoder, aggregate slide-level representations, and optimize only a linear head with AdamW during downstream supervised probing. The default downstream hyperparameters are 20 epochs, learning rate 10^{-4}, weight decay 10^{-5}, patience 5, and device `cuda`. For classification, the downstream table uses `slide_id` and `label`. For survival, it uses `slide_id`, `time`, and `event`; for RCC subtype-specific survival splits, it uses `os_survival_days` and `os_censorship`. The downstream optimizer operates on full split tensors rather than on a separate mini-batch data loader.

MIL downstream protocol. MIL downstream experiments use the same five downstream folds and metadata fields as the linear-probing experiments (patient-level for the TCGA cohorts and slide-level for Camelyon16). The patch encoder is frozen, and only the MIL aggregation module and task head are trained. For the RQ1 single-scale and cross-scale MIL baselines, this frozen patch encoder is the same ViT-S/16 checkpoint used to initialize and freeze Patch-JEPA's fine-scale target encoder, namely `vit_small_patch16_224.augreg_in1k`, and the identical checkpoint is reused unchanged across all MIL methods. Single-scale MIL baselines use frozen $5\times$ patch bags, while cross-scale MIL baselines use aligned $1.25\times$–$5\times$ coarse–fine bags; these two input regimes are reported separately. All MIL models are trained with AdamW for 20 epochs with learning rate 10^{-4}, weight decay 10^{-5}, shuffled training samples, no class weighting, and checkpoint selection by validation performance. Classification reports accuracy (ACC), balanced accuracy (BAcc), AUROC, and macro-F1 (F1); survival reports C-index, averaged over five folds.

C.2 RQ1. Stage of Cross-Scale Integration

C.2.1 Detailed Setup

RQ1 asks whether cross-scale computation can be shifted from slide-level inference into patch pretraining by comparing Patch-JEPA with post-hoc multi-scale WSI pipelines in Tables 1 and 11. The comparison includes single-scale MIL methods (ABMIL, CLAM, and TransMIL), cross-scale MIL methods that fuse separately encoded magnifications during slide-level modeling (DSMIL, MRAN, CSMIL, MSPN, and MARBLE), and hierarchical or slide-level foundation model pipelines (HIPT, CHIEF, and GigaPath). Within the MIL subset, every method reuses the same frozen ViT-S/16 patch encoder, `vit_small_patch16_224.augreg_in1k`, with no additional MIL-specific pretraining; Patch-JEPA is evaluated with the shared linear-probing protocol with a frozen encoder, whereas foundation baselines use their published pipelines and weights.

Patch-level foundation encoders such as Phikon (Filiot et al., 2023), UNI (Chen et al., 2024), Virchow2 (Zimmermann et al., 2024), and Phikon-v2 (Filiot et al., 2024) are discussed in related work. RQ1 compares the stage at which cross-scale information is incorporated. DS denotes downstream trainable parameters. Appendix C.1 gives the shared downstream protocol.

C.2.2 Baseline Groups

Below we briefly summarize the tabulated RQ1 baselines at the model level.

- **ABMIL** (Ilse et al., 2018) is a canonical single-scale attention MIL baseline that aggregates patch representations into a slide-level representation without explicit cross-scale integration.
- **CLAM** (Lu et al., 2021) is a single-scale MIL baseline that uses clustering-constrained attention to focus on the most discriminative instances within a slide bag.
- **TransMIL** (Shao et al., 2021) is a single-scale transformer MIL baseline that models long-range correlations among patch representations through self-attention.
- **DSMIL** (Li et al., 2021) is a dual-stream MIL baseline in which multi-instance evidence is fused at the slide-model stage rather than in patch pretraining.
- **MRAN** (Xiang et al., 2023) is a multi-scale representation attention baseline that combines magnification-specific representations after patch encoding through attention-based slide reasoning.
- **CSMIL** (Deng et al., 2024) is a cross-scale MIL baseline that explicitly models interactions between magnification-specific patch representations during slide aggregation.
- **MSPN** (Wu et al., 2026) is a progressive multi-scale slide model that introduces pyramidal cross-scale structure at the slide-encoding stage rather than through patch-level predictive pretraining.
- **HIPT** (Chen et al., 2022b) is a hierarchical slide-level foundation model baseline that builds Slide-level representations from multi-level patch representations and slide-level transformers.
- **CHIEF** (Wang et al., 2024) is a slide-level foundation model baseline that combines strong patch representations with whole-slide pretraining.
- **GigaPath** (Xu et al., 2024) is a large-scale pathology slide-level foundation model baseline trained to transfer slide-level representations from broad pathology corpora.
- **MARBLE** (Dwarampudi et al., 2026) is a recent cross-scale MIL-style slide encoder that introduces multi-scale structure after patch encoding, providing another post-hoc comparison point within the post-hoc slide-modeling group.

C.2.3 Additional Experiments

$5\times \rightarrow 20\times$ **full-frame evaluation.**

Setup. The magnification-extension experiment evaluates Patch-JEPA with $5\times$ coarse views and aligned $20\times$ fine targets on BRCA and UCEC. Inference uses only the $5\times$ context encoder. Table 10 compares the complete systems using the baseline families of Table 1.

Table 10: Full-frame BRCA and UCEC classification with $5\times \rightarrow 20\times$ Patch-JEPA pretraining. ACC and AUROC are reported.

Cohort	Metric	ABMIL	CLAM	TransMIL	DSMIL	MRAN	CSMIL	MSPN	MARBLE	HIPT	CHIEF	GigaPath	**Patch-JEPA**
BRCA	ACC	0.852	0.856	0.849	0.863	0.868	0.854	0.837	0.857	0.833	0.862	**0.878**	0.874
	AUROC	0.848	0.859	0.813	0.865	0.869	0.843	0.833	0.861	0.812	0.856	**0.882**	0.881
UCEC	ACC	0.779	0.776	0.760	0.790	**0.813**	0.764	0.776	0.792	0.785	0.788	0.799	0.805
	AUROC	0.786	0.759	0.713	0.791	**0.819**	0.756	0.763	0.782	0.792	0.795	0.795	0.809

Survival and stage results. Table 11 complements Table 1 with the appendix-only survival and stage endpoints under the same RQ1 grouping. The outcome remains dataset-dependent. Patch-JEPA leads both RCC endpoints, BRCA splits between MARBLE on stage and GigaPath on survival, and UCEC is more mixed, with several MIL baselines matching or exceeding Patch-JEPA. Taken together, the endpoint results suggest that the benefit of shifting cross-scale integration into patch pretraining is clearest on RCC, while UCEC remains more sensitive to cohort effects and downstream slide-level modeling.

Table 11: Survival and stage results (C-index / F1).

	RCC		BRCA		UCEC	
Method	Surv.	Stage	Surv.	Stage	Surv.	Stage
Single-scale MIL						
ABMIL (Ilse et al., 2018)	0.652±0.07	0.272±0.03	0.561±0.05	0.200±0.01	0.675±0.11	0.204±0.01
CLAM (Lu et al., 2021)	0.651±0.07	0.275±0.03	0.589±0.04	0.209±0.02	0.661±0.08	0.198±0.01
TransMIL (Shao et al., 2021)	0.650±0.07	0.279±0.04	0.582±0.05	0.213±0.02	0.621±0.08	0.198±0.01
Cross-scale MIL						
DSMIL (Li et al., 2021)	0.687±0.08	0.306±0.05	0.598±0.03	0.228±0.04	0.669±0.09	0.196±0.01
MRAN (Xiang et al., 2023)	0.686±0.08	0.276±0.07	0.583±0.05	0.221±0.03	0.673±0.08	0.202±0.01
CSMIL (Deng et al., 2024)	0.652±0.07	0.290±0.08	0.589±0.03	0.208±0.02	0.675±0.05	0.208±0.02
MSPN (Wu et al., 2026)	0.676±0.09	0.308±0.07	0.578±0.05	0.226±0.03	0.670±0.09	**0.217±0.03**
MARBLE (Dwarampudi et al., 2026)	0.684±0.07	0.283±0.05	0.596±0.05	**0.240±0.02**	0.632±0.10	0.202±0.01
Cross-scale slide-level foundation models						
HIPT (Chen et al., 2022b)	0.647±0.04	0.191±0.02	0.573±0.08	0.223±0.03	0.638±0.09	0.183±0.01
CHIEF (Wang et al., 2024)	0.661±0.06	0.211±0.03	0.592±0.07	0.210±0.03	0.654±0.08	0.194±0.01
GigaPath (Xu et al., 2024)	0.625±0.06	0.317±0.04	**0.625±0.07**	0.231±0.03	0.667±0.07	0.205±0.02
Patch-JEPA (ViT-S/16)	**0.694±0.07**	**0.330±0.03**	0.596±0.03	0.237±0.02	**0.676±0.06**	0.209±0.01

ViT-B/16 backbone sensitivity. Tables 13 and 12 provide a ViT-B/16 sensitivity check for the appendix MIL comparison rather than a literal duplicate. The non-JEPA rows use frozen representations from the ImageNet-1K-initialized ViT-B/16 checkpoint `vit_base_patch16_224.augreg_in1k`, while Table 13 also includes the matched ViT-B/16 Patch-JEPA model evaluated with its learned coarse-scale context encoder at inference. This keeps the backbone family fixed at ViT-B/16 while separating representations from a frozen encoder from the ViT-B/16 Patch-JEPA reference. The results remain mixed across cohorts and heads rather than collapsing to a single dominant MIL model. RCC still favors MARBLE or DSMIL in several subtype metrics, while BRCA and UCEC remain more distributed across heads and survival and stage trends continue to vary by cohort. Table 12 uses the same stage F1 metric as the RQ1 appendix survival and stage table, and Table 13 reuses the available five-fold standard deviation for the ViT-B/16 Patch-JEPA row from the corresponding reference run.

Table 14: Survival and stage metrics across SSL objectives and MIL heads. Values are mean ± standard deviation over five folds. Survival uses C-index and stage uses F1.

MIL head	RCC		BRCA		UCEC	
	Surv.	Stage	Surv.	Stage	Surv.	Stage
MAE (He et al., 2022) + random init.						
ABMIL	0.574±0.09	0.179±0.01	0.564±0.07	0.192±0.01	0.555±0.06	0.198±0.01
CLAM	0.555±0.07	0.175±0.01	0.566±0.08	0.194±0.01	0.557±0.08	0.198±0.01
TransMIL	0.575±0.07	0.177±0.01	0.532±0.09	0.190±0.01	0.608±0.08	0.198±0.01
DSMIL	0.666±0.06	0.302±0.04	0.562±0.06	0.187±0.01	0.605±0.11	0.204±0.01
MRAN	0.663±0.08	0.289±0.03	0.568±0.08	0.190±0.01	0.585±0.07	0.204±0.01
CSMIL	0.638±0.09	0.285±0.06	0.557±0.08	0.187±0.01	0.618±0.05	0.203±0.01
MSPN	0.658±0.05	0.302±0.03	0.531±0.06	0.191±0.01	0.602±0.12	0.209±0.01
MARBLE	0.667±0.05	0.291±0.03	0.582±0.09	0.196±0.02	0.599±0.07	0.204±0.01
SimCLR (Chen et al., 2020) + random init.						
ABMIL	0.539±0.04	0.178±0.01	0.466±0.06	0.187±0.01	0.539±0.05	0.198±0.01
CLAM	0.560±0.06	0.176±0.01	0.503±0.09	0.192±0.01	0.543±0.05	0.198±0.01
TransMIL	0.600±0.06	0.175±0.01	0.553±0.08	0.200±0.01	0.542±0.04	0.198±0.01
DSMIL	0.675±0.07	0.311±0.05	0.546±0.09	0.196±0.01	0.577±0.08	0.204±0.01
MRAN	0.666±0.06	0.311±0.03	0.518±0.07	0.194±0.01	0.584±0.09	0.209±0.01
CSMIL	0.661±0.06	0.310±0.05	0.539±0.11	0.187±0.01	0.610±0.09	**0.217**±0.04
MSPN	0.637±0.05	0.279±0.06	0.517±0.05	0.187±0.01	0.574±0.13	0.207±0.02
MARBLE	0.638±0.05	0.262±0.05	0.538±0.09	0.208±0.02	0.573±0.09	0.205±0.01
Patch-JEPA	**0.694**±0.07	0.330±0.03	**0.596**±0.03	**0.237**±0.02	**0.676**±0.06	0.209±0.01

MIL head	RCC		BRCA		UCEC	
	Surv.	Stage	Surv.	Stage	Surv.	Stage
Cross-MAE + random init.						
ABMIL	0.569±0.08	0.175±0.01	0.574±0.07	0.185±0.01	0.576±0.08	0.198±0.01
CLAM	0.569±0.07	0.175±0.01	0.569±0.06	0.185±0.01	0.604±0.08	0.198±0.01
TransMIL	0.560±0.07	0.176±0.01	0.574±0.07	0.187±0.01	0.611±0.10	0.198±0.01
DSMIL	0.658±0.05	0.294±0.03	0.585±0.06	0.194±0.01	0.588±0.11	0.204±0.01
MRAN	0.652±0.05	0.304±0.01	0.577±0.06	0.187±0.01	0.570±0.10	0.207±0.03
CSMIL	0.643±0.07	**0.335**±0.05	0.572±0.08	0.187±0.01	0.570±0.11	0.207±0.02
MSPN	0.656±0.05	0.305±0.02	0.549±0.07	0.187±0.01	0.596±0.14	0.207±0.02
MARBLE	0.636±0.02	0.291±0.07	0.591±0.08	0.201±0.01	0.610±0.07	0.204±0.01
MPCS (Chhipa et al., 2023) + random init.						
ABMIL	0.511±0.06	0.191±0.01	0.500±0.06	0.185±0.01	0.604±0.07	0.198±0.01
CLAM	0.505±0.06	0.189±0.01	0.528±0.03	0.185±0.01	0.612±0.08	0.198±0.01
TransMIL	0.539±0.04	0.184±0.01	0.590±0.08	0.187±0.01	0.605±0.04	0.198±0.01
DSMIL	0.667±0.06	0.293±0.03	0.547±0.09	0.187±0.01	0.578±0.12	0.204±0.01
MRAN	0.662±0.06	0.287±0.03	0.547±0.07	0.187±0.01	0.572±0.08	0.204±0.01
CSMIL	0.595±0.06	0.262±0.05	0.516±0.06	0.193±0.01	0.570±0.11	0.203±0.01
MSPN	0.668±0.06	0.258±0.04	0.527±0.05	0.187±0.01	0.569±0.15	0.203±0.01
MARBLE	0.654±0.08	0.267±0.03	0.572±0.07	0.195±0.01	0.582±0.09	0.211±0.02
Patch-JEPA	**0.694**±0.07	0.330±0.03	**0.596**±0.03	**0.237**±0.02	**0.676**±0.06	0.209±0.01

Table 12: Survival and stage prediction metrics across MIL models with a ViT-B/16 backbone (stage reported as F1).

Method	RCC		BRCA		UCEC	
	Surv.	Stage	Surv.	Stage	Surv.	Stage
ABMIL	0.674±0.07	0.365±0.04	0.602±0.05	0.227±0.02	0.446±0.11	0.248±0.03
CLAM	0.677±0.07	0.363±0.05	0.578±0.04	0.219±0.02	0.380±0.08	0.251±0.06
TransMIL	0.638±0.07	0.298±0.06	0.555±0.05	0.233±0.03	0.348±0.08	0.254±0.03
CSMIL	0.663±0.09	0.300±0.09	0.598±0.03	0.200±0.03	0.659±0.05	0.203±0.01
DSMIL	0.689±0.08	0.313±0.04	0.617±0.03	0.216±0.04	0.680±0.09	0.203±0.01
MRAN	0.689±0.06	0.304±0.07	0.619±0.05	0.193±0.01	**0.688**±0.08	0.212±0.02
MSPN	0.681±0.07	0.258±0.05	0.619±0.05	0.200±0.02	0.642±0.09	0.198±0.01
MARBLE	**0.697**±0.06	0.311±0.07	0.620±0.05	0.202±0.02	0.687±0.10	0.225±0.05
Patch-JEPA	0.694±0.03	**0.392**±0.02	**0.622**±0.05	**0.292**±0.02	0.642±0.07	**0.263**±0.02

Table 13: Subtype and metastasis classification metrics across MIL models with a ViT-B/16 backbone.

Method	RCC Subtype			BRCA Subtype			UCEC Subtype			Camelyon16 Metastasis		
	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1
ABMIL	0.893±0.02	0.975±0.01	0.869±0.03	0.883±0.03	0.903±0.05	0.771±0.07	0.780±0.03	0.748±0.04	0.601±0.04	0.692±0.05	0.732±0.09	0.639±0.06
CLAM	0.896±0.03	0.977±0.01	0.876±0.03	**0.889**±0.05	0.905±0.05	**0.783**±0.05	0.774±0.03	0.781±0.04	0.600±0.04	0.677±0.05	0.738±0.08	0.636±0.06
TransMIL	0.889±0.03	0.970±0.02	0.856±0.04	0.864±0.02	0.860±0.05	0.719±0.06	0.782±0.04	0.778±0.07	0.600±0.10	0.644±0.07	0.708±0.09	0.587±0.13
CSMIL	0.894±0.02	0.975±0.01	0.878±0.03	0.862±0.03	0.888±0.05	0.727±0.05	0.766±0.05	0.760±0.05	0.619±0.10	0.694±0.07	0.732±0.08	0.646±0.07
DSMIL	0.898±0.02	0.980±0.01	0.878±0.02	0.857±0.04	0.887±0.05	0.727±0.05	**0.800**±0.05	0.771±0.06	0.657±0.09	0.694±0.11	0.761±0.10	0.632±0.11
MRAN	0.889±0.01	0.978±0.01	0.867±0.02	0.877±0.03	0.894±0.05	0.778±0.04	0.780±0.04	0.794±0.04	**0.678**±0.08	**0.734**±0.09	0.771±0.10	**0.711**±0.11
MSPN	0.888±0.01	0.976±0.01	0.867±0.02	0.855±0.03	0.891±0.06	0.716±0.07	0.758±0.04	0.761±0.05	0.556±0.10	0.694±0.04	0.718±0.03	0.646±0.05
MARBLE	0.901±0.02	0.976±0.01	0.882±0.02	0.876±0.04	0.894±0.04	0.754±0.05	0.780±0.04	0.769±0.04	0.615±0.07	0.677±0.02	0.688±0.07	0.581±0.03
Patch-JEPA (ViT-B/16)	**0.933**±0.01	**0.988**±0.02	**0.934**±0.05	0.888±0.02	**0.906**±0.04	0.776±0.03	0.788±0.05	**0.798**±0.07	**0.678**±0.09	0.711±0.03	**0.773**±0.06	0.676±0.05

In-domain MIL comparison. Tables 15 and 14 rerun the appendix MIL evaluation with in-domain pretrained encoders throughout instead of generic patch representations. Besides Patch-JEPA, we include SimCLR, MAE, MPCS, and Cross-MAE, each pretrained in-domain and then evaluated under the same MIL heads trained on frozen representations with both random and ImageNet-1K initialization. The ImageNet-1K-initialized rows match the main-text RQ2 baseline regime, while the randomly initialized rows serve as controls, so these tables isolate initialization effects without changing the underlying in-domain pathology pretraining setting.

Table 15: Subtype and metastasis classification metrics for the in-domain MIL comparison across SSL objectives and downstream MIL heads. Values are mean ± standard deviation over five folds. Each row group corresponds to one SSL objective/initialization setting, and the shaded cells mark the best and second-best values in each column.

MIL head	RCC			BRCA			UCEC			Camelyon16		
	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1
MAE (He et al., 2022) + random init.												
ABMIL	0.606±0.02	0.717±0.04	0.432±0.04	0.804±0.03	0.659±0.06	0.431±0.01	0.761±0.03	0.534±0.06	0.455±0.03	0.612±0.04	0.612±0.07	0.480±0.06
CLAM	0.609±0.02	0.713±0.04	0.433±0.03	0.805±0.03	0.654±0.07	0.432±0.01	0.759±0.03	0.542±0.06	0.446±0.02	0.601±0.05	0.605±0.07	0.473±0.07
TransMIL	0.604±0.04	0.707±0.05	0.417±0.04	0.805±0.03	0.657±0.05	0.432±0.01	0.759±0.03	0.518±0.06	0.431±0.01	0.609±0.03	0.614±0.03	0.479±0.05
DSMIL	0.885±0.02	0.973±0.01	0.858±0.02	0.831±0.03	0.807±0.05	0.644±0.05	0.740±0.05	0.682±0.09	0.535±0.09	0.624±0.05	0.623±0.03	0.568±0.07
MRAN	0.862±0.01	0.970±0.01	0.832±0.02	0.831±0.03	0.795±0.06	0.636±0.07	0.746±0.03	0.643±0.04	0.481±0.06	0.584±0.06	0.601±0.08	0.553±0.08
CSMIL	0.886±0.02	0.973±0.01	0.863±0.02	0.820±0.03	0.801±0.05	0.636±0.05	0.740±0.04	0.634±0.06	0.464±0.05	0.604±0.06	0.625±0.09	0.539±0.10
MSPN	0.872±0.04	0.966±0.01	0.852±0.03	0.830±0.02	0.808±0.07	0.657±0.03	0.742±0.04	0.654±0.05	0.502±0.07	0.614±0.03	0.615±0.10	0.537±0.10
MARBLE	0.872±0.02	0.966±0.01	0.845±0.02	0.832±0.03	0.794±0.09	0.665±0.06	0.751±0.05	0.632±0.10	0.493±0.04	0.609±0.04	0.646±0.07	0.525±0.06
SimCLR (Chen et al., 2020) + random init.												
ABMIL	0.617±0.03	0.678±0.06	0.416±0.02	0.808±0.03	0.620±0.05	0.461±0.05	0.761±0.02	0.416±0.04	0.432±0.01	0.614±0.05	0.577±0.07	0.496±0.09
CLAM	0.626±0.03	0.676±0.06	0.413±0.03	0.809±0.04	0.625±0.05	0.476±0.07	0.761±0.02	0.449±0.04	0.432±0.01	0.616±0.04	0.575±0.04	0.514±0.05
TransMIL	0.645±0.04	0.715±0.04	0.448±0.03	0.799±0.03	0.606±0.06	0.451±0.03	0.761±0.02	0.439±0.02	0.432±0.01	0.616±0.05	0.572±0.04	0.542±0.05
DSMIL	0.879±0.03	0.970±0.01	0.850±0.04	0.822±0.04	0.783±0.06	0.597±0.06	0.747±0.03	0.643±0.08	0.525±0.07	0.621±0.04	0.637±0.07	0.564±0.05
MRAN	0.875±0.02	0.969±0.01	0.855±0.03	0.824±0.02	0.783±0.06	0.639±0.05	0.740±0.05	0.641±0.08	0.482±0.07	0.586±0.07	0.599±0.12	0.540±0.06
CSMIL	0.876±0.02	0.970±0.01	0.851±0.03	0.835±0.03	0.799±0.06	0.653±0.03	0.736±0.05	0.607±0.07	0.457±0.04	0.621±0.07	0.628±0.09	0.578±0.08
MSPN	0.871±0.01	0.965±0.01	0.839±0.02	0.825±0.03	0.775±0.05	0.602±0.03	0.734±0.06	0.637±0.07	0.487±0.05	0.616±0.04	0.616±0.09	0.508±0.09
MARBLE	0.853±0.01	0.964±0.01	0.817±0.03	0.819±0.05	0.744±0.06	0.567±0.11	0.757±0.02	0.606±0.11	0.449±0.04	0.611±0.05	0.606±0.07	0.522±0.07
Cross-MAE + random init.												
ABMIL	0.614±0.03	0.721±0.05	0.461±0.03	0.804±0.03	0.652±0.04	0.443±0.02	0.761±0.02	0.518±0.05	0.438±0.01	0.621±0.03	0.640±0.05	0.533±0.04
CLAM	0.617±0.02	0.714±0.04	0.437±0.06	0.799±0.04	0.665±0.05	0.435±0.01	0.761±0.02	0.526±0.06	0.438±0.01	0.604±0.04	0.629±0.07	0.524±0.04
TransMIL	0.608±0.03	0.715±0.04	0.388±0.03	0.805±0.03	0.647±0.07	0.432±0.01	0.761±0.02	0.483±0.08	0.432±0.01	0.626±0.05	0.631±0.06	0.542±0.07
DSMIL	0.896±0.02	0.977±0.01	0.866±0.02	0.832±0.03	0.801±0.07	0.640±0.04	0.758±0.02	0.666±0.08	0.497±0.07	0.636±0.05	0.656±0.09	0.565±0.05
MRAN	0.885±0.03	0.970±0.01	0.857±0.04	0.818±0.04	0.789±0.06	0.600±0.11	0.734±0.05	0.666±0.06	0.514±0.06	0.629±0.04	0.642±0.04	0.553±0.03
CSMIL	0.898±0.02	0.974±0.01	0.872±0.03	0.829±0.04	0.803±0.06	0.651±0.08	0.733±0.05	0.610±0.06	0.458±0.03	0.619±0.04	0.628±0.08	0.560±0.05
MSPN	0.891±0.02	0.973±0.01	0.865±0.03	0.828±0.04	0.785±0.07	0.666±0.07	0.730±0.05	0.602±0.06	0.445±0.02	0.599±0.05	0.615±0.06	0.551±0.03
MARBLE	0.880±0.02	0.966±0.01	0.856±0.03	0.830±0.04	0.793±0.08	0.632±0.06	0.761±0.02	0.628±0.07	0.473±0.03	0.607±0.05	0.622±0.06	0.550±0.06
MPCS (Chhipa et al., 2023) + random init.												
ABMIL	0.664±0.02	0.779±0.05	0.567±0.06	0.804±0.03	0.655±0.07	0.431±0.01	0.749±0.05	0.515±0.03	0.428±0.02	0.637±0.03	0.601±0.07	0.541±0.06
CLAM	0.666±0.02	0.788±0.05	0.571±0.05	0.805±0.03	0.656±0.07	0.432±0.01	0.751±0.04	0.518±0.04	0.429±0.01	0.637±0.04	0.579±0.07	0.532±0.06
TransMIL	0.657±0.02	0.781±0.03	0.556±0.04	0.805±0.03	0.653±0.06	0.432±0.01	0.761±0.02	0.511±0.06	0.432±0.01	0.631±0.04	0.579±0.03	0.537±0.04
DSMIL	0.881±0.02	0.975±0.01	0.856±0.02	0.830±0.03	0.791±0.07	0.636±0.03	0.766±0.03	0.663±0.09	0.523±0.08	0.621±0.05	0.659±0.06	0.557±0.05
MRAN	0.888±0.02	0.971±0.01	0.866±0.03	0.829±0.03	0.798±0.06	0.653±0.04	0.744±0.04	0.639±0.05	0.505±0.06	0.654±0.05	0.661±0.08	0.579±0.06
CSMIL	0.882±0.02	0.972±0.01	0.857±0.01	0.819±0.03	0.801±0.06	0.641±0.05	0.731±0.05	0.625±0.07	0.473±0.05	0.649±0.06	0.631±0.07	0.577±0.08
MSPN	0.886±0.01	0.972±0.01	0.858±0.02	0.818±0.04	0.780±0.08	0.619±0.09	0.741±0.03	0.621±0.10	0.489±0.07	0.639±0.06	0.643±0.06	0.577±0.05
MARBLE	0.870±0.01	0.967±0.01	0.845±0.02	0.835±0.04	0.807±0.06	0.652±0.04	0.751±0.05	0.623±0.09	0.494±0.07	0.637±0.02	0.636±0.05	0.540±0.07
SimCLR (Chen et al., 2020) + ImageNet-1K init.												
ABMIL	0.606±0.03	0.715±0.05	0.399±0.03	0.814±0.03	0.769±0.06	0.627±0.07	0.715±0.03	0.585±0.06	0.442±0.06	0.614±0.02	0.625±0.06	0.494±0.10
CLAM	0.610±0.02	0.733±0.04	0.395±0.05	0.821±0.03	0.787±0.05	0.644±0.08	0.713±0.03	0.586±0.06	0.434±0.06	0.619±0.03	0.621±0.08	0.456±0.10
TransMIL	0.645±0.03	0.753±0.05	0.504±0.09	0.803±0.02	0.760±0.05	0.565±0.07	0.723±0.02	0.560±0.06	0.420±0.02	0.647±0.05	0.655±0.08	0.555±0.14
DSMIL	0.891±0.03	0.976±0.01	0.867±0.03	0.822±0.04	0.846±0.04	0.699±0.09	0.728±0.05	0.718±0.04	0.585±0.09	0.707±0.03	0.756±0.07	0.685±0.04
MRAN	0.887±0.01	0.970±0.01	0.858±0.01	0.819±0.03	0.845±0.04	0.680±0.07	0.734±0.02	0.734±0.04	0.558±0.06	0.719±0.04	0.777±0.07	0.695±0.05
CSMIL	0.875±0.02	0.969±0.01	0.847±0.03	0.826±0.03	0.827±0.05	0.667±0.07	0.730±0.03	0.695±0.08	0.535±0.11	0.654±0.07	0.679±0.08	0.612±0.07
MSPN	0.882±0.01	0.970±0.01	0.853±0.02	0.815±0.03	0.827±0.04	0.608±0.11	0.745±0.04	0.639±0.08	0.508±0.10	0.624±0.05	0.632±0.09	0.557±0.12
MARBLE	0.855±0.02	0.959±0.01	0.816±0.02	0.822±0.03	0.811±0.05	0.648±0.09	0.718±0.07	0.639±0.09	0.536±0.07	0.626±0.04	0.607±0.08	0.546±0.10
MAE (He et al., 2022) + ImageNet-1K init.												
ABMIL	0.616±0.03	0.731±0.05	0.452±0.05	0.789±0.04	0.656±0.05	0.425±0.01	0.733±0.02	0.452±0.07	0.409±0.01	0.619±0.02	0.616±0.04	0.464±0.05
CLAM	0.613±0.02	0.731±0.05	0.440±0.04	0.789±0.04	0.654±0.05	0.425±0.01	0.731±0.02	0.428±0.08	0.408±0.01	0.619±0.01	0.614±0.06	0.478±0.01
TransMIL	0.607±0.02	0.709±0.05	0.399±0.02	0.795±0.03	0.665±0.07	0.422±0.01	0.731±0.02	0.488±0.09	0.402±0.01	0.609±0.02	0.652±0.05	0.431±0.11
DSMIL	0.894±0.02	0.976±0.01	0.869±0.02	0.822±0.03	0.844±0.05	0.704±0.06	0.737±0.07	0.757±0.07	0.606±0.07	0.752±0.04	0.770±0.08	0.724±0.05
MRAN	0.879±0.02	0.973±0.01	0.849±0.03	0.824±0.04	0.847±0.04	0.690±0.06	0.760±0.05	**0.790**±0.04	0.630±0.07	0.729±0.06	0.782±0.07	0.706±0.06
CSMIL	0.880±0.03	0.971±0.01	0.853±0.03	0.831±0.03	0.858±0.04	0.715±0.03	0.760±0.06	0.756±0.06	0.647±0.10	0.712±0.06	0.733±0.10	0.666±0.06
MSPN	0.874±0.02	0.968±0.01	0.842±0.02	0.821±0.03	0.850±0.05	0.651±0.11	0.717±0.05	0.723±0.07	0.592±0.11	0.694±0.05	0.746±0.05	0.654±0.06
Cross-MAE + ImageNet-1K init.												
ABMIL	0.680±0.02	0.807±0.03	0.552±0.04	0.819±0.03	0.662±0.04	0.447±0.02	0.729±0.03	0.462±0.09	0.401±0.01	0.602±0.02	0.525±0.04	0.420±0.05
CLAM	0.666±0.03	0.809±0.03	0.540±0.05	0.814±0.04	0.676±0.05	0.439±0.01	0.729±0.03	0.455±0.09	0.401±0.01	0.607±0.01	0.521±0.04	0.426±0.05
TransMIL	0.686±0.03	0.828±0.02	0.587±0.08	0.820±0.03	0.657±0.07	0.436±0.01	0.727±0.03	0.465±0.07	0.401±0.01	0.609±0.05	0.659±0.08	0.509±0.13
DSMIL	0.888±0.01	0.973±0.01	0.862±0.01	0.847±0.03	0.816±0.07	0.650±0.04	0.753±0.03	0.778±0.05	0.657±0.05	0.752±0.05	0.782±0.08	0.715±0.08
MRAN	0.874±0.02	0.970±0.01	0.846±0.03	0.833±0.04	0.803±0.06	0.609±0.11	0.741±0.04	0.776±0.05	0.604±0.08	0.742±0.05	0.789±0.05	0.709±0.05
CSMIL	0.885±0.03	0.971±0.01	0.861±0.03	0.844±0.04	0.818±0.06	0.662±0.08	0.734±0.06	0.748±0.06	0.615±0.08	0.704±0.05	0.727±0.09	0.655±0.05
MSPN	0.878±0.02	0.970±0.01	0.853±0.02	0.843±0.04	0.799±0.07	0.677±0.07	0.728±0.07	0.734±0.08	0.573±0.12	0.727±0.06	0.729±0.10	0.684±0.07
MARBLE	0.884±0.01	0.970±0.01	0.862±0.02	0.845±0.04	0.807±0.08	0.642±0.06	0.749±0.06	0.744±0.07	0.561±0.11	0.687±0.05	0.729±0.05	0.640±0.06
MPCS (Chhipa et al., 2023) + ImageNet-1K init.												
ABMIL	0.788±0.02	0.906±0.02	0.747±0.03	0.819±0.03	0.665±0.07	0.435±0.01	0.740±0.02	0.568±0.10	0.453±0.04	0.714±0.06	0.716±0.09	0.681±0.08
CLAM	0.794±0.02	0.909±0.02	0.759±0.03	0.820±0.03	0.666±0.07	0.436±0.01	0.713±0.04	0.592±0.09	0.438±0.04	0.694±0.06	0.715±0.08	0.658±0.07
TransMIL	0.755±0.02	0.883±0.01	0.724±0.02	0.820±0.03	0.663±0.06	0.436±0.01	0.731±0.02	0.557±0.06	0.409±0.02	0.682±0.08	0.718±0.08	0.650±0.08
DSMIL	0.903±0.01	0.981±0.01	0.879±0.02	0.845±0.03	0.805±0.07	0.646±0.03	0.743±0.07	0.745±0.08	0.600±0.14	**0.775**±0.06	0.812±0.08	0.738±0.10
MRAN	0.898±0.03	0.976±0.01	0.876±0.04	0.844±0.03	0.813±0.06	0.663±0.04	0.760±0.04	0.770±0.07	**0.662**±0.07	0.762±0.05	**0.825**±0.06	**0.742**±0.06
CSMIL	0.885±0.02	0.973±0.01	0.862±0.03	0.834±0.03	0.816±0.06	0.651±0.05	0.721±0.07	0.740±0.07	0.584±0.10	0.727±0.04	0.737±0.06	0.681±0.06
MSPN	0.869±0.03	0.969±0.01	0.845±0.03	0.833±0.04	0.794±0.08	0.628±0.09	0.745±0.04	0.754±0.07	0.608±0.06	0.709±0.03	0.733±0.06	0.653±0.05
MARBLE	0.874±0.02	0.971±0.01	0.850±0.02	0.851±0.04	0.822±0.06	0.662±0.04	0.747±0.05	0.735±0.07	0.556±0.10	0.684±0.07	0.715±0.12	0.631±0.09
Patch-JEPA	**0.918**±0.01	**0.983**±0.01	**0.906**±0.02	**0.862**±0.02	**0.872**±0.04	**0.736**±0.03	**0.782**±0.05	0.775±0.07	0.592±0.09	0.719±0.03	0.743±0.06	0.683±0.05

C.3 RQ2. OBJECTIVE COMPARISON

C.3.1 DETAILED SETUP

RQ2 tests objective families under a matched in-house re-pretraining protocol. Unless otherwise noted, all newly trained RQ2 pretraining runs use Adam, whereas downstream supervised probing follows the shared AdamW protocol in Appendix C.1. The shared-hyperparameter table refers only to settings held fixed when newly pretraining the compared RQ2 methods, not to the separate Patch-JEPA implementation table in Appendix D. Table 16 records the shared trainable-backbone settings, and Table 17 records the objective-dependent view construction and target-branch design across the compared SSL baselines. The main-text RQ2 comparison uses ImageNet-1K-initialized trainable backbones for the SSL baselines, matching the initialization regime used when Patch-JEPA keeps its fine-scale target encoder frozen; Appendix C.3.3 additionally reports random-initialization controls. The trainable encoder and coarse-only downstream probe are matched across RQ2 methods, while target-side mechanisms remain objective-dependent; Patch-JEPA uses a frozen target branch for latent prediction rather than a distillation objective.

Table 16: Shared hyperparameters for the RQ2 re-pretraining comparison.

Category	**Setting**
Shared Input and Backbone Settings	
Pretraining corpus	same in-domain pathology pretraining corpus across all RQ2 methods
Encoder family	shared ViT-S/16 backbone family across the compared objectives
Trainable encoder initialization	ImageNet-1K initialization for the main-text RQ2 SSL comparison; appendix control tables additionally report random initialization
Crop size per view	224×224
ViT patch size per view	16×16
Input token grid	14×14 tokens per view
Shared Optimization Settings	
Batch size	256
Epochs	100
Optimizer	Adam
Optimizer learning rate	1×10^{-4}
Weight decay	1×10^{-6}

Table 17: Objective-specific settings for the RQ2 pretraining comparison.

Setting	**SimCLR**	**DINOv2**	**MAE**	**MPCS**	**X-MAG**	**Cross-MAE**
SSL family	align.	distill.	recon.	x-scale align.	x-scale distill.	x-scale recon.
Pretraining views	same-view crop	same-view crop	same-view crop	aligned coarse and fine	aligned coarse and fine	aligned coarse and fine
Cross-scale correspondence	none	none	none	aligned pair	aligned pair	aligned pair
Primary supervision	aug. agreement	EMA target representation	masked same-view content	cross-scale positives	cross-mag target representation	masked opposite-view content
Target-side branch	none	same-run EMA target	none	none	cross-mag target	none
External pretrained target init	–	none	–	–	none	–
Target-side update	–	EMA	–	–	objective-native	–

C.3.2 BASELINE GROUPS

Below we briefly summarize the RQ2 baselines at the model level.

- **SimCLR** (Chen et al., 2020) is a standard contrastive self-supervised baseline that learns view-invariant representations by maximizing agreement between two augmented views of the same image, making it a natural representative of alignment-based SSL.
- **DINOv2** (Oquab et al., 2023) is a self-distillation baseline in which an online network matches EMA target representations without labels or explicit negatives, representing a strong modern distillation-based SSL objective.
- **MAE** (He et al., 2022) is a masked autoencoding baseline that reconstructs missing image content from visible patches, representing the reconstruction-based SSL family in the comparison.

- **MPCS** (Chhipa et al., 2023) is a pathology-specific multi-magnification contrastive baseline that treats aligned regions across magnifications as positives, encouraging shared representations across scales through contrastive agreement.
- **X-MAG** (Su et al., 2025) is a cross-magnification distillation baseline designed for pathology, where information learned at one magnification is transferred to another by matching target representations across magnifications.
- **Cross-MAE** is a cross-scale MAE-style reconstruction baseline that uses aligned information from another magnification to help predict masked content, serving as a reconstruction-based multi-scale comparison point.

C.3.3 Additional Experiments

RQ2 summary across datasets and endpoints. Tables 18 and 19 provide the most compact appendix view of classification, survival, and stage performance across all four benchmarks before the head-specific breakdowns below. Throughout these RQ2 appendix tables, each SSL objective is shown with both random and ImageNet-1K initialization; the main-text RQ2 comparison corresponds to the ImageNet-1K setting, which matches the initialization regime used for the frozen Patch-JEPA fine-scale target encoder when applicable. The column-wise shading highlights which objective families remain strong across datasets and endpoint types, and Table 20 then opens that comparison into the first detailed classification breakdown.

Table 18: RQ2 classification metrics with random and ImageNet-1K initialization. Values are mean ± standard deviation over five folds. The shaded cells mark the best and second-best values in each column.

Method	RCC			BRCA			UCEC			Camelyon16		
	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1
Random initialization												
SimCLR (Chen et al., 2020)	0.648±0.02	0.743±0.05	0.541±0.03	0.826±0.03	0.492±0.10	0.471±0.05	0.761±0.02	0.452±0.05	0.432±0.01	0.621±0.03	0.624±0.06	0.490±0.05
DINOv2 (Oquab et al., 2023)	0.623±0.03	0.720±0.05	0.453±0.01	0.800±0.03	0.671±0.04	0.555±0.07	0.759±0.03	0.499±0.07	0.431±0.01	0.629±0.02	0.658±0.07	0.541±0.05
MAE (He et al., 2022)	0.578±0.02	0.646±0.05	0.354±0.06	0.815±0.03	0.591±0.03	0.461±0.02	0.755±0.03	0.521±0.07	0.430±0.01	0.602±0.01	0.597±0.06	0.401±0.04
MPCS (Chhipa et al., 2023)	0.647±0.01	0.752±0.04	0.501±0.03	0.816±0.03	0.664±0.04	0.512±0.04	0.761±0.02	0.519±0.07	0.432±0.01	0.644±0.02	0.609±0.10	0.528±0.04
X-MAG (Su et al., 2025)	0.821±0.01	0.898±0.01	0.810±0.01	0.780±0.03	0.811±0.02	0.693±0.02	0.660±0.01	0.713±0.01	0.510±0.01	0.587±0.01	0.622±0.01	0.565±0.01
Cross-MAE	0.587±0.02	0.661±0.04	0.365±0.03	0.805±0.04	0.643±0.05	0.504±0.05	0.761±0.02	0.473±0.09	0.430±0.01	0.619±0.05	0.633±0.05	0.534±0.09
ImageNet-1K initialization												
SimCLR (Chen et al., 2020)	0.630±0.02	0.755±0.04	0.411±0.03	0.841±0.03	0.801±0.04	0.640±0.03	0.749±0.02	0.617±0.06	0.472±0.07	0.616±0.04	0.634±0.07	0.554±0.04
DINOv2 (Oquab et al., 2023)	0.825±0.01	0.938±0.01	0.779±0.02	0.814±0.02	0.832±0.06	0.679±0.03	0.753±0.03	0.706±0.05	0.533±0.07	0.654±0.05	0.671±0.09	0.613±0.07
MAE (He et al., 2022)	0.589±0.02	0.688±0.05	0.367±0.02	0.822±0.03	0.656±0.03	0.469±0.04	0.757±0.02	0.497±0.09	0.431±0.01	0.602±0.01	0.609±0.06	0.381±0.01
MPCS (Chhipa et al., 2023)	0.713±0.02	0.836±0.02	0.654±0.04	0.816±0.03	0.664±0.04	0.512±0.04	0.757±0.03	0.540±0.11	0.431±0.01	0.664±0.05	0.681±0.08	0.583±0.07
X-MAG (Su et al., 2025)	0.856±0.01	0.937±0.01	0.845±0.01	0.845±0.03	0.846±0.02	0.723±0.02	0.689±0.01	0.744±0.01	0.536±0.01	0.613±0.01	0.648±0.01	0.590±0.01
Cross-MAE	0.605±0.03	0.710±0.05	0.389±0.03	0.805±0.04	0.643±0.05	0.504±0.05	0.761±0.02	0.417±0.04	0.432±0.01	0.607±0.01	0.599±0.06	0.394±0.03
Patch-JEPA	**0.918**±0.01	0.983±0.01	**0.906**±0.02	**0.862**±0.02	**0.872**±0.04	**0.736**±0.03	**0.782**±0.05	**0.775**±0.07	**0.592**±0.09	**0.719**±0.03	**0.743**±0.06	**0.683**±0.05

Table 19: RQ2 survival and stage metrics with random and ImageNet-1K initialization. Values are mean ± standard deviation over five folds; entries with unavailable fold standard deviation are shown as mean ± n/a. Stage uses F1 and survival uses C-index. The shaded cells mark the best and second-best values in each column.

Random initialization

Method	RCC		BRCA		UCEC	
	Stage	Surv.	Stage	Surv.	Stage	Surv.
Single-scale objectives						
SimCLR (Chen et al., 2020)	0.176±0.01	0.560±0.06	0.187±0.01	0.490±0.09	0.198±0.01	0.543±0.05
DINOv2 (Oquab et al., 2023)	0.197±0.03	0.539±0.09	0.183±0.01	0.532±0.07	0.198±0.01	0.544±0.06
MAE (He et al., 2022)	0.175±0.01	0.555±0.07	0.189±0.01	0.552±0.08	0.198±0.01	0.557±0.08
Cross-scale objectives						
MPCS (Chhipa et al., 2023)	0.189±0.01	0.505±0.06	0.180±0.01	0.515±0.03	0.198±0.01	0.612±0.08
X-MAG (Su et al., 2025)	0.225±0.07	0.542±0.06	0.197±0.02	0.547±0.03	0.202±0.02	0.601±0.05
Cross-MAE	0.175±0.01	0.569±0.07	0.180±0.01	0.555±0.06	0.198±0.01	0.604±0.08
Patch-JEPA	**0.327**±0.02	**0.623**±0.06	**0.249**±0.01	**0.561**±0.03	**0.212**±0.02	**0.616**±0.02

ImageNet-1K initialization

Method	RCC		BRCA		UCEC	
	Stage	Surv.	Stage	Surv.	Stage	Surv.
Single-scale objectives						
SimCLR (Chen et al., 2020)	0.175±0.01	0.529±0.02	0.192±0.01	0.530±0.07	0.221±0.01	0.527±0.04
DINOv2 (Oquab et al., 2023)	0.309±0.02	0.596±0.08	0.188±0.01	0.541±0.05	**0.240**±0.05	0.517±0.04
MAE (He et al., 2022)	0.176±0.01	0.532±0.06	0.194±0.01	0.537±0.05	0.198±0.01	0.559±0.06
Cross-scale objectives						
MPCS (Chhipa et al., 2023)	0.207±0.04	0.450±0.09	0.185±0.01	0.528±0.03	0.198±0.01	0.612±0.08
X-MAG (Su et al., 2025)	0.225±0.07	0.542±0.06	0.202±0.02	0.561±0.03	0.202±0.02	0.601±0.05
Cross-MAE	0.175±0.01	0.531±0.05	0.185±0.01	0.569±0.06	0.198±0.01	0.475±0.05
Patch-JEPA	**0.330**±0.03	**0.694**±0.07	**0.237**±0.02	**0.596**±0.03	0.209±0.01	**0.676**±0.06

ABMIL classification breakdown. Table 20 expands the ABMIL classification results into full per-dataset ACC, BAcc, AUROC, and F1 values across RCC, BRCA, UCEC, and Camelyon16. Relative to the compact summary tables above, this view makes it easier to see where gains concentrate within each cohort. The column-wise shading shows Patch-JEPA strongest across most columns, with X-MAG and DINOv2 remaining the most competitive non-JEPA baselines in several settings.

Table 20: RQ2 classification metrics with ABMIL. Values are mean ± standard deviation over five folds. The shaded cells mark the best and second-best values in each column.

Method	RCC				BRCA				UCEC				Camelyon16			
	ACC	BAcc	AUROC	F1	ACC	BAcc	AUROC	F1	ACC	BAcc	AUROC	F1	ACC	BAcc	AUROC	F1
Single-scale objectives																
SimCLR (Chen et al., 2020)	0.617±0.03	0.438±0.02	0.678±0.06	0.416±0.02	0.828±0.03	0.515±0.03	0.640±0.05	0.481±0.05	0.761±0.02	0.500±0.01	0.416±0.04	0.432±0.01	0.614±0.05	0.542±0.05	0.577±0.07	0.496±0.09
DINOv2 (Oquab et al., 2023)	0.608±0.02	0.471±0.05	0.722±0.05	0.479±0.06	0.825±0.03	0.500±0.01	0.665±0.06	0.452±0.01	0.763±0.02	0.504±0.01	0.491±0.07	0.440±0.02	0.621±0.04	0.577±0.05	0.645±0.07	0.566±0.07
MAE (He et al., 2022)	0.606±0.01	0.438±0.03	0.717±0.04	0.432±0.04	0.824±0.03	0.499±0.01	0.679±0.06	0.451±0.01	0.761±0.03	0.509±0.01	0.534±0.06	0.455±0.03	0.612±0.04	0.533±0.04	0.612±0.07	0.480±0.06
Cross-scale objectives																
MPCS (Chhipa et al., 2023)	0.664±0.02	0.549±0.06	0.779±0.05	0.567±0.06	0.824±0.03	0.499±0.01	0.675±0.07	0.451±0.01	0.749±0.04	0.492±0.02	0.515±0.03	0.428±0.01	0.637±0.03	0.570±0.04	0.601±0.07	0.541±0.06
X-MAG (Su et al., 2025)	0.782±0.02	0.769±0.03	0.860±0.01	0.766±0.03	0.818±0.03	0.596±0.04	0.780±0.03	0.624±0.04	0.753±0.06	0.552±0.08	0.711±0.06	0.547±0.11	0.632±0.03	0.574±0.06	0.641±0.06	0.561±0.08
Cross-MAE	0.614±0.03	0.456±0.03	0.721±0.05	0.461±0.03	0.824±0.03	0.504±0.01	0.672±0.04	0.463±0.02	0.761±0.02	0.502±0.01	0.518±0.05	0.438±0.01	0.621±0.03	0.558±0.03	0.640±0.05	0.533±0.04
Patch-JEPA	**0.919**±0.02	**0.899**±0.01	**0.983**±0.01	**0.902**±0.02	**0.871**±0.02	**0.733**±0.06	**0.893**±0.03	**0.748**±0.04	**0.793**±0.03	**0.671**±0.04	**0.833**±0.03	**0.682**±0.04	**0.727**±0.08	**0.704**±0.10	**0.772**±0.08	**0.703**±0.10

CLAM classification breakdown. Table 21 repeats the same per-dataset classification breakdown with CLAM in place of ABMIL. Read together with Table 20, it separates representation quality from downstream aggregator choice and tests whether the ordering of objectives stays stable across MIL heads. The column-wise shading again shows Patch-JEPA strongest overall, with X-MAG serving as the most consistent comparison point.

Table 21: RQ2 classification metrics with CLAM. Values are mean ± standard deviation over five folds.

Method	RCC				BRCA				UCEC				Camelyon16			
	ACC	BAcc	AUROC	F1	ACC	BAcc	AUROC	F1	ACC	BAcc	AUROC	F1	ACC	BAcc	AUROC	F1
Single-scale objectives																
SimCLR (Chen et al., 2020)	0.626±0.03	0.437±0.03	0.676±0.06	0.413±0.03	0.829±0.04	0.522±0.04	0.645±0.05	0.496±0.07	0.761±0.02	0.500±0.01	0.449±0.04	0.432±0.01	0.616±0.04	0.550±0.03	0.575±0.04	0.514±0.05
DINOv2 (Oquab et al., 2023)	0.631±0.03	0.501±0.06	0.732±0.04	0.512±0.07	0.825±0.03	0.506±0.01	0.664±0.06	0.465±0.04	0.763±0.02	0.504±0.01	0.509±0.06	0.440±0.02	0.619±0.04	0.575±0.06	0.650±0.06	0.562±0.07
MAE (He et al., 2022)	0.609±0.02	0.442±0.02	0.713±0.04	0.433±0.03	0.825±0.03	0.500±0.01	0.674±0.07	0.452±0.01	0.759±0.03	0.504±0.01	0.542±0.06	0.446±0.02	0.601±0.05	0.525±0.05	0.605±0.07	0.473±0.07
Cross-scale objectives																
MPCS (Chhipa et al., 2023)	0.666±0.02	0.552±0.05	0.788±0.05	0.571±0.05	0.825±0.03	0.500±0.01	0.676±0.07	0.452±0.01	0.751±0.04	0.493±0.01	0.518±0.04	0.429±0.01	0.637±0.04	0.567±0.04	0.579±0.07	0.532±0.07
X-MAG (Su et al., 2025)	0.787±0.03	0.772±0.03	0.858±0.01	0.777±0.04	0.816±0.03	0.608±0.07	0.778±0.03	0.625±0.05	0.757±0.04	0.545±0.07	0.697±0.06	0.536±0.10	0.632±0.03	0.569±0.06	0.643±0.06	0.555±0.08
Cross-MAE	0.617±0.03	0.444±0.04	0.714±0.04	0.437±0.06	0.819±0.04	0.499±0.01	0.685±0.06	0.455±0.01	0.761±0.02	0.502±0.01	0.526±0.06	0.438±0.01	0.604±0.04	0.546±0.03	0.629±0.07	0.524±0.04
Patch-JEPA	**0.918**±0.02	**0.899**±0.02	**0.983**±0.01	**0.904**±0.02	**0.862**±0.03	**0.711**±0.07	**0.893**±0.04	**0.725**±0.05	**0.811**±0.04	**0.727**±0.05	**0.839**±0.04	**0.732**±0.05	**0.744**±0.08	**0.719**±0.09	**0.777**±0.08	**0.720**±0.09

Survival and stage results for ABMIL and CLAM. Table 22 places the ABMIL and CLAM survival and stage results side by side so endpoint behavior can be compared across downstream heads without switching tables. This view makes it easier to judge whether gains beyond subtype classification transfer across MIL formulations. Across these endpoints, Patch-JEPA and X-MAG capture most of the top values, while the remaining baselines are less consistently strong across datasets.

Table 22: RQ2 survival and stage metrics with ABMIL and CLAM. Values are mean ± standard deviation over five folds.

ABMIL downstream

Method	RCC		BRCA		UCEC	
	Stage	Surv.	Stage	Surv.	Stage	Surv.
Single-scale objectives						
SimCLR (Chen et al., 2020)	0.178±0.01	0.539±0.04	0.187±0.01	0.466±0.06	0.198±0.01	0.539±0.05
DINOv2 (Oquab et al., 2023)	0.193±0.02	0.470±0.05	0.186±0.01	0.526±0.08	0.198±0.01	0.557±0.04
MAE (He et al., 2022)	0.179±0.01	0.574±0.09	0.192±0.01	0.564±0.07	0.198±0.01	0.555±0.06
Cross-scale objectives						
MPCS (Chhipa et al., 2023)	0.191±0.01	0.511±0.06	0.185±0.01	0.500±0.06	0.198±0.01	0.604±0.07
X-MAG (Su et al., 2025)	0.246±0.03	0.558±0.01	0.193±0.01	0.561±0.04	0.208±0.03	0.595±0.06
Cross-MAE	0.175±0.01	0.569±0.08	0.185±0.01	0.574±0.07	0.198±0.01	0.576±0.08
Patch-JEPA	**0.361**±0.04	**0.674**±0.06	**0.242**±0.04	**0.599**±0.07	**0.228**±0.02	**0.655**±0.06

CLAM downstream

Method	RCC		BRCA		UCEC	
	Stage	Surv.	Stage	Surv.	Stage	Surv.
Single-scale objectives						
SimCLR (Chen et al., 2020)	0.176±0.01	0.560±0.06	0.192±0.01	0.503±0.09	0.198±0.01	0.543±0.05
DINOv2 (Oquab et al., 2023)	0.197±0.03	0.539±0.09	0.188±0.01	0.546±0.07	0.198±0.01	0.544±0.06
MAE (He et al., 2022)	0.175±0.01	0.555±0.07	0.194±0.01	0.566±0.08	0.198±0.01	0.557±0.08
Cross-scale objectives						
MPCS (Chhipa et al., 2023)	0.189±0.01	0.505±0.06	0.185±0.01	0.528±0.03	0.198±0.01	0.612±0.08
X-MAG (Su et al., 2025)	0.225±0.07	0.542±0.06	0.202±0.02	0.561±0.03	0.202±0.02	0.601±0.05
Cross-MAE	0.175±0.01	0.569±0.07	0.185±0.01	0.569±0.06	0.198±0.01	0.604±0.08
Patch-JEPA	**0.373**±0.02	**0.683**±0.06	**0.244**±0.04	**0.596**±0.06	**0.212**±0.02	**0.683**±0.08

Patch-level foundation model baselines under low- and high-magnification evaluation. Tables 23 and 24 summarize patch-level foundation model performance under separate low- and high-magnification evaluation. Each foundation model is evaluated at the specified input magnification. Classification and survival and stage are split so Camelyon16 appears only for metastasis classification, while the TCGA cohorts retain their available clinical endpoints.

Table 23: Classification metrics under low and high magnification. Values are mean ± standard deviation over five folds.

Method	Params	RCC			BRCA			UCEC			Camelyon16		
		ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1
Coarse-scale													
ViT-S/16	21.7M	0.616±0.04	0.650±0.06	0.550±0.05	0.846±0.03	0.771±0.04	0.603±0.03	0.756±0.03	0.684±0.09	0.504±0.06	0.614±0.04	0.636±0.06	0.498±0.09
ViT-B/16	85.8M	0.735±0.04	0.790±0.04	0.690±0.05	0.858±0.03	0.855±0.05	0.684±0.05	0.766±0.02	0.719±0.03	0.556±0.06	0.614±0.03	0.626±0.05	0.549±0.06
Phikon-v2	307M	0.848±0.03	0.900±0.03	0.805±0.04	0.571±0.05	0.640±0.06	0.525±0.05	0.765±0.04	0.715±0.05	0.471±0.06	0.629±0.02	0.650±0.07	0.511±0.07
CONCH	371M	0.893±0.03	0.935±0.03	0.860±0.04	0.874±0.02	0.902±0.04	0.759±0.02	0.782±0.04	0.755±0.06	0.623±0.07	0.624±0.03	0.639±0.03	0.522±0.07
Virchow2	632M	0.910±0.02	0.950±0.02	0.880±0.03	0.875±0.04	0.887±0.04	0.740±0.07	0.780±0.03	0.778±0.08	0.532±0.05	0.632±0.04	0.632±0.07	0.522±0.04
UNI2-H	681M	0.926±0.02	0.960±0.02	0.895±0.03	0.891±0.04	0.914±0.03	0.787±0.07	0.799±0.04	0.821±0.05	0.647±0.07	0.662±0.06	0.710±0.07	0.599±0.07
Fine-scale													
ViT-S/16	21.7M	0.847±0.03	0.895±0.03	0.805±0.04	0.854±0.03	0.849±0.06	0.653±0.06	0.762±0.03	0.735±0.09	0.517±0.04	0.639±0.04	0.646±0.05	0.527±0.08
ViT-B/16	85.8M	0.880±0.03	0.920±0.03	0.840±0.04	0.874±0.03	0.888±0.05	0.717±0.05	0.774±0.05	0.776±0.04	0.591±0.10	0.654±0.05	0.670±0.06	0.594±0.06
Phikon-v2	307M	0.912±0.03	0.950±0.02	0.880±0.03	0.593±0.05	0.670±0.06	0.555±0.05	0.751±0.05	0.731±0.05	0.437±0.03	**0.792**±0.19	0.685±0.09	**0.724**±0.26
CONCH	371M	0.916±0.02	0.955±0.02	0.890±0.03	0.896±0.03	0.935±0.02	0.802±0.05	0.784±0.07	0.785±0.08	0.605±0.12	0.644±0.03	0.663±0.07	0.564±0.05
Virchow2	632M	0.928±0.02	0.965±0.02	0.905±0.03	0.890±0.02	0.916±0.03	0.784±0.05	0.794±0.04	0.836±0.07	0.617±0.08	0.679±0.03	0.673±0.08	0.598±0.04
UNI2-H	681M	**0.933**±0.02	0.970±0.02	0.903±0.02	**0.905**±0.02	**0.940**±0.01	**0.813**±0.05	**0.837**±0.06	**0.873**±0.06	**0.739**±0.08	0.722±0.07	0.770±0.08	0.658±0.09
Patch-JEPA (ViT-S/16)	21.7M	0.918±0.01	0.983±0.01	0.906±0.02	0.862±0.02	0.872±0.04	0.736±0.03	0.782±0.05	0.775±0.07	0.592±0.09	0.719±0.03	0.743±0.06	0.683±0.05
Patch-JEPA (ViT-B/16)	85.8M	**0.933**±0.01	**0.988**±0.02	**0.934**±0.05	0.888±0.02	0.906±0.04	0.776±0.03	0.788±0.05	0.798±0.07	0.678±0.09	0.711±0.03	**0.773**±0.06	0.676±0.05

Table 24: Stage and survival metrics under low and high magnification. Foundation-model evaluation magnifications may differ from their native pretraining magnifications; this is not a magnification-matched comparison. Values are mean ± standard deviation over five folds.

Coarse-scale

Method	RCC		BRCA		UCEC	
	Stage	Surv.	Stage	Surv.	Stage	Surv.
ViT-S/16	0.281±0.05	0.384±0.16	0.203±0.02	0.589±0.10	0.198±0.01	0.506±0.14
ViT-B/16	0.300±0.03	0.469±0.20	0.257±0.03	0.515±0.12	0.271±0.04	0.604±0.06
Phikon-v2	0.202±0.04	0.585±0.19	0.219±0.04	0.557±0.09	0.198±0.01	0.554±0.08
CONCH	0.307±0.04	0.588±0.16	0.285±0.06	0.566±0.06	**0.281**±0.05	0.562±0.07
Virchow2	0.316±0.04	0.594±0.16	0.293±0.04	0.570±0.08	0.250±0.02	0.566±0.11
UNI2-H	0.323±0.04	0.602±0.15	**0.302**±0.04	0.609±0.07	0.264±0.05	0.674±0.07
Patch-JEPA (ViT-S/16)	0.330±0.03	**0.694**±0.07	0.237±0.02	0.596±0.03	0.209±0.01	**0.676**±0.06
Patch-JEPA (ViT-B/16)	**0.392**±0.02	**0.694**±0.03	0.292±0.02	**0.622**±0.05	0.263±0.02	0.642±0.07

Fine-scale

Method	RCC		BRCA		UCEC	
	Stage	Surv.	Stage	Surv.	Stage	Surv.
ViT-S/16	0.273±0.04	0.520±0.23	0.221±0.06	0.513±0.09	0.198±0.01	0.545±0.10
ViT-B/16	0.351±0.02	0.507±0.22	0.276±0.04	0.569±0.11	0.260±0.05	0.633±0.09
Phikon-v2	0.343±0.03	0.585±0.20	0.238±0.04	0.558±0.09	0.198±0.01	0.472±0.09
CONCH	0.352±0.04	0.592±0.16	0.307±0.05	0.561±0.06	0.250±0.01	0.595±0.10
Virchow2	0.358±0.04	0.601±0.15	0.297±0.05	0.584±0.08	0.240±0.05	0.519±0.06
UNI2-H	0.367±0.04	0.607±0.15	**0.317**±0.03	0.593±0.09	**0.287**±0.04	0.663±0.12
Patch-JEPA (ViT-S/16)	0.330±0.03	**0.694**±0.07	0.237±0.02	0.596±0.03	0.209±0.01	**0.676**±0.06
Patch-JEPA (ViT-B/16)	**0.392**±0.02	**0.694**±0.03	0.292±0.02	**0.622**±0.05	0.263±0.02	0.642±0.07

C.4 RQ3. Effect of Cross-Scale Supervision

C.4.1 Detailed Setup

RQ3 isolates whether the gains come from simply making JEPA multi-scale during pretraining or from preserving correct coarse–fine correspondence. All variants use the same in-domain pathology pretraining corpus, ViT-S/16 encoder family, reported pretraining schedule, and shared downstream linear-probing protocol from Appendix C.1. Unless otherwise noted, coarse and fine refer to the main reported $1.25\times$ and $5\times$ views. Every RQ3 variant is evaluated from a coarse-scale slide bag only, including Joint and I-JEPA (fine). Table 25 specifies the pretraining contexts, targets, and retained inference inputs.

Table 25: RQ3 variant protocol. All variants use only coarse-scale slide inputs at downstream inference; fine views are used only as indicated during pretraining.

Method	Training context	Prediction target / objective	Inference input
I-JEPA (coarse)	Coarse view	Masked tokens from the same coarse view; same-view I-JEPA	Coarse slide bag
I-JEPA (fine)	Fine view	Masked tokens from the same fine view; same-view I-JEPA	Coarse slide bag
Joint	Aligned coarse and fine views	Joint representation learned from aligned coarse–fine views	Coarse slide bag
Unaligned	Coarse view with a fine-patch field from a different site	Ordered fine-scale latent field from a different site; $\mathcal{L}_c + 0.1\mathcal{L}_f$	Coarse slide bag
Patch-JEPA	Coarse view	Spatially aligned ordered fine-scale latent field; $\mathcal{L}_c + 0.1\mathcal{L}_f$	Coarse slide bag

C.4.2 BASELINE GROUPS

Below we summarize the RQ3 baseline construction. Under the shared coarse-only evaluation setting, the comparison isolates how the pretraining views and cross-scale correspondence affect transfer.

- **I-JEPA (coarse)** is pretrained using only coarse patches with the standard same-view I-JEPA objective. This baseline tests whether coarse contextual pretraining alone is sufficient for downstream transfer under the shared coarse-only evaluation setting.
- **I-JEPA (fine)** is pretrained using only fine patches with the same-view I-JEPA objective. This variant isolates the effect of fine-scale morphology pretraining when the downstream slide-level representation is still evaluated from coarse-scale inputs.
- **Joint** is pretrained with paired coarse and fine views sampled from the same spatial hierarchy, preserving the true coarse–fine correspondence between views. Its downstream representation is read out from coarse-scale slide inputs only. This baseline tests whether giving JEPA aligned multi-scale views improves transfer beyond single-scale pretraining under the same coarse-only inference condition.
- **Unaligned** uses the same coarse and fine view composition as Joint, but intentionally breaks the coarse–fine correspondence between paired views. This control tests whether improvements come from true cross-scale alignment rather than from simply turning JEPA into a multi-scale pretraining setup.

C.5 RQ4. PREDICTING FINE-SCALE STRUCTURE FROM COARSE REPRESENTATIONS

This section presents the TCGA-RCC aligned latent-field probe, image-derived targets, and prototype-histogram analyses and controls.

C.5.1 MAIN RQ4 PROBE

Probe protocol. RQ4 tests whether frozen coarse representations support prediction of the corresponding ordered fine-scale latent field. Each frozen encoder receives a coarse patch, and a ridge probe trained under the same protocol for each method maps its representation to the aligned $5\times$ latent field. Fine-scale data defines the target; the evaluated encoder receives only the coarse patch.

We evaluate TCGA-RCC latent-field prediction using cosine similarity and R^2, and retrieve held-out ordered fields from a common candidate pool using Recall@1 (R@1) and Recall@5 (R@5).

Baseline groups. We compare single-scale SSL baselines (SimCLR and MAE), cross-scale SSL baselines (MPCS and X-MAG), Cross-MAE, and Patch-JEPA. Every encoder is frozen and evaluated from the same coarse-scale input, so the comparison isolates the fine-scale information predictable from each learned coarse representation.

Result summary. In the main TCGA-RCC comparison (Table 4), Patch-JEPA has the highest cosine similarity (0.830), R^2 (0.450), R@1 (18.0%), and R@5 (60.4%). The improvement over Cross-MAE is largest in R^2 (0.231 to 0.450), indicating that the frozen coarse representation supports prediction of the aligned fine-scale latent field. The additional analyses below examine image-derived properties, spatial order, and prototype distributions under coarse-only evaluation.

C.5.2 ADDITIONAL FINE-SCALE TARGET ANALYSES

Beyond the aligned latent field, we evaluate image-derived properties of the $5\times$ region and their within-site spatial variation. Both analyses predict targets defined from aligned fine-scale data using ridge probes on frozen $1.25\times$ coarse representations.

Image-derived measurements. We evaluate whether coarse representations predict image-derived measurements computed from the raw aligned $5\times$ region.

Table 26: Patient-macro Pearson correlation on TCGA-RCC for raw $5\times$ image-derived measurements predicted from frozen $1.25\times$ coarse representations. Higher values are better.

Raw $5\times$ measurement	**Patch-JEPA**	**Cross-MAE**	**MAE**	**MPCS**	**SimCLR**	**X-MAG**
Hematoxylin-object density	**0.729**	0.664	0.646	0.638	0.617	0.593
Hematoxylin OD standard deviation	**0.735**	0.685	0.677	0.643	0.622	0.536
Intensity entropy	**0.844**	0.838	0.820	0.792	0.787	0.762
Hematoxylin-object area CV	**0.459**	0.458	0.442	0.424	0.442	0.306

Table 27: Prediction of within-site spatial variation on TCGA-RCC from frozen $1.25\times$ coarse representations. Values are Pearson correlations; higher is better. The shuffled-location control permutes target locations within each site.

	Patch-JEPA	**Cross-MAE**	**MAE**	**MPCS**	**SimCLR**	**X-MAG**	**Shuffled locations**
Within-site location (r)	**0.675**	0.660	0.657	0.627	0.628	0.579	0.003
Neighbor-difference (r)	**0.648**	0.617	0.614	0.577	0.589	0.531	0.009

Within-site spatial variation. We also test whether the frozen coarse representation predicts spatial variation within the aligned fine-scale field. The shuffled-location control permutes the target locations within each site while preserving the set of local values.

Result summary. Patch-JEPA gives the highest correlations for all four image-derived measurements in Table 26. It also leads both spatial targets in Table 27, with correlations of 0.675 for within-site location and 0.648 for neighboring differences. The shuffled-location control falls to 0.003 and 0.009, respectively. Together, these results show that the coarse representation predicts measured properties of the aligned fine-scale region and retains information about their spatial arrangement.

C.5.3 Prototype-Histogram Analysis

Target construction and metrics. As an additional latent diagnostic, we construct a prototype-histogram target from the aligned fine-scale region. We use TCGA-RCC with 40,000 training coarse sites, 10,000 held-out coarse sites, seed 7, and a training-set-fitted $K = 64$ fine-token prototype codebook. Aligned fine tokens are assigned to the fixed codebook and summarized as a normalized prototype histogram.

Kullback–Leibler divergence (KL) and cross-entropy (CE) measure distributional prediction error. Top-1 and Top-5 measure dominant-prototype recovery, and Pearson denotes correlation between predicted and target histograms. Retrieval uses cosine similarity between the predicted histogram and candidate target histograms, with the aligned prototype histogram as the positive.

Baseline groups. This analysis compares SimCLR, DINOv2, I-JEPA (fine), X-MAG, Cross-MAE, Patch-JEPA, and a Patch-JEPA variant without the fine-scale prediction branch.

Table 28: Prototype-histogram prediction and retrieval metrics on TCGA-RCC.

Method	**KL ↓**	**CE ↓**	**Top-1 ↑**	**Top-5 ↑**	**Pearson ↑**	**R@1 ↑**	**R@5 ↑**	**MRR ↑**
Single-scale SSL baselines								
SimCLR	2.614	3.956	0.130	0.385	0.253	0.032	0.137	0.103
DINOv2	2.685	4.026	0.126	0.381	0.246	0.032	0.136	0.102
I-JEPA (fine)	2.389	3.731	0.171	0.455	0.306	0.060	0.201	0.146
Cross-scale SSL baselines								
X-MAG	1.837	3.178	0.320	0.713	0.496	0.012	0.041	0.036
Cross-MAE	2.164	3.805	0.148	0.414	0.372	0.043	0.166	0.120
Patch-JEPA variants								
w/o fine-scale prediction branch	1.789	3.131	0.349	0.742	0.522	0.202	0.530	0.355
Patch-JEPA	**1.735**	**3.076**	**0.364**	**0.759**	**0.537**	**0.235**	**0.573**	**0.391**

Result summary. Patch-JEPA gives the strongest distributional, dominant-prototype, correlation, and retrieval results in the prototype-histogram analysis. The variant without the fine-scale prediction branch remains competitive, indicating that the coarse-scale prediction branch already provides substantial supervision for prototype-distribution prediction, while the fine-scale prediction branch improves the reported metrics.

C.5.4 Prototype-Histogram Controls

Tables 29 and 30 test retrieval after excluding same-slide or same-patient candidate negatives. Table 31 tests whether the order of fine-patch targets matters.

Same-slide exclusion retrieval. Same-slide exclusion removes candidate negatives from the query slide while retaining its aligned prototype histogram as the positive.

Table 29: Same-slide exclusion retrieval on TCGA-RCC. Lower KL/CE and higher Top-1/Top-5/Pearson/R@1/R@5/MRR are better. The **best** and second-best values are marked.

Method	Prediction					No exclusion			Same-slide exclusion		
	KL	CE	Top-1	Top-5	Pearson	R@1	R@5	MRR	R@1	R@5	MRR
SimCLR	2.614	3.955	0.130	0.386	0.253	0.0009	0.0048	0.0059	0.0009	0.0049	0.0061
DINOv2	2.685	4.026	0.126	0.381	0.246	0.0010	0.0048	0.0059	0.0011	0.0050	0.0061
I-JEPA (fine)	2.648	3.990	0.104	0.317	0.208	0.0005	0.0035	0.0043	0.0005	0.0040	0.0044
X-MAG	1.837	3.178	0.320	0.713	0.496	0.0118	0.0409	0.0361	0.0136	0.0445	0.0389
Cross-MAE	2.273	3.615	0.179	0.488	0.324	0.0026	0.0106	0.0115	0.0027	0.0110	0.0119
Patch-JEPA	**1.754**	**3.095**	**0.350**	**0.749**	**0.525**	**0.0149**	**0.0604**	**0.0471**	**0.0198**	**0.0690**	**0.0537**

Result summary. Patch-JEPA has the strongest prediction metrics and the highest retrieval scores in Table 29. With same-slide candidate negatives excluded, its R@1 rises from 0.0149 to 0.0198 and R@5 from 0.0604 to 0.0690; the method ranking is preserved. The control shows that Patch-JEPA's lead persists under this restricted candidate pool.

Same-patient exclusion retrieval. Same-patient exclusion removes candidate negatives from the query patient while retaining its aligned prototype histogram as the positive.

Table 30: Same-patient exclusion retrieval on TCGA-RCC. Lower KL/CE and higher Top-1/Top-5/Pearson/R@1/R@5/MRR are better. The **best** and second-best values are marked.

Method	Prediction					No exclusion			Same-patient exclusion		
	KL	CE	Top-1	Top-5	Pearson	R@1	R@5	MRR	R@1	R@5	MRR
SimCLR	2.614	3.955	0.130	0.386	0.253	0.0009	0.0048	0.0059	0.0009	0.0049	0.0061
DINOv2	2.685	4.026	0.126	0.381	0.246	0.0010	0.0048	0.0059	0.0011	0.0050	0.0061
I-JEPA (fine)	2.648	3.990	0.104	0.317	0.208	0.0005	0.0035	0.0043	0.0005	0.0040	0.0044
X-MAG	1.837	3.178	0.320	0.713	0.496	0.0118	0.0409	0.0361	0.0139	0.0445	0.0391
Cross-MAE	2.273	3.615	0.179	0.488	0.324	0.0026	0.0106	0.0115	0.0027	0.0110	0.0119
Patch-JEPA	**1.754**	**3.095**	**0.350**	**0.749**	**0.525**	**0.0149**	**0.0604**	**0.0471**	**0.0198**	**0.0692**	**0.0539**

Result summary. Patch-JEPA also leads the prediction and retrieval columns in Table 30. After same-patient candidate negatives are excluded, its R@1 is 0.0198, R@5 is 0.0692, and MRR is 0.0539. The ordering across methods is similar to the same-slide control, so the reported ranking holds under both candidate-pool restrictions.

Shuffled fine-patch-grid order ablation. This ablation compares the ordered row-major 4×4 field of fine-patch prototype distributions with a shuffled target that preserves the same 16 local

Table 32: Compute and timing protocol for the efficiency analysis.

Item	Protocol
Hardware	Single NVIDIA RTX PRO 6000 Blackwell GPU, 94.97 GiB VRAM
CPU / RAM	2× Intel Xeon Silver 4510 CPUs, 251 GiB RAM
Software	PyTorch 2.11.0+cu130, CUDA 13.0
Precision	Automatic mixed precision for GPU forward passes
Pretraining unit	Minutes per epoch
Inference unit	Model-side milliseconds per fixed bag
Pretraining batch / epoch definition	Same as the reported pretraining configuration
Pretraining scope	Training epoch time; validation and downstream evaluation excluded
Inference scope	Encoder forward pass + lightweight slide-level head
Excluded operations	WSI loading, patch extraction, cache construction, disk I/O
Timing method	CUDA events with explicit synchronization
Timing statistic	Mean ± standard deviation over 7 timed repeats after 2 warm-up runs

prototype distributions but permutes their positions. It tests whether the aligned fine-patch-grid order contributes predictive signal beyond unordered local fine appearance.

Table 31: Shuffled fine-patch-grid order ablation on TCGA-RCC. Lower KL and higher Pearson/Top-1/Top-5/R@1/R@5/MRR are better. The **best** and second-best values are marked within each ordered/shuffled block.

Method	Ordered							Shuffled						
	KL	Pearson	Top-1	Top-5	R@1	R@5	MRR	KL	Pearson	Top-1	Top-5	R@1	R@5	MRR
SimCLR	4.216	0.144	0.097	0.317	0.0050	0.0101	0.0116	4.316	0.134	0.089	0.298	0.0032	0.0055	0.0077
DINOv2	4.404	0.139	0.098	0.312	0.0054	0.0123	0.0129	4.511	0.128	0.089	0.290	0.0031	0.0064	0.0081
I-JEPA (fine)	4.297	0.118	0.081	0.271	0.0028	0.0096	0.0095	4.392	0.105	0.074	0.243	0.0029	0.0044	0.0063
X-MAG	3.472	0.278	0.227	0.556	0.0075	0.0177	0.0216	3.493	0.277	0.224	0.553	0.0081	0.0172	0.0214
Cross-MAE	3.779	0.186	0.137	0.399	0.0077	0.0171	0.0181	3.859	0.176	0.128	0.379	0.0053	0.0092	0.0121
Patch-JEPA	**3.286**	**0.302**	**0.247**	**0.593**	**0.0093**	**0.0234**	**0.0266**	**3.326**	**0.297**	**0.242**	**0.583**	**0.0090**	**0.0199**	**0.0241**

Result summary. Patch-JEPA scores higher with ordered targets than with shuffled targets across the reported distributional and retrieval metrics in Table 31. For example, ordered targets reduce KL from 3.326 to 3.286 and increase R@5 from 0.0199 to 0.0234. Because shuffling retains the same local prototype distributions while changing their positions, the difference indicates that aligned fine-patch-grid order contributes predictive signal beyond unordered local content.

RQ4 synthesis. The RCC latent-field probe, image-derived targets, and prototype analyses converge on a spatially specific result. Frozen coarse representations support prediction of aligned fine-scale properties, and the ordered-grid control shows that target location matters. The exclusion experiments preserve the method ranking under different retrieval candidate pools. Together, these observations support the interpretation that cross-scale pretraining shapes a coarse representation that retains information about the aligned fine-scale region.

C.6 Efficiency Analysis

Figure 4 reports pretraining time and model-side inference time under a fixed setup. Pretraining cost is measured in minutes per epoch using the same batch and epoch definitions as the reported pretraining configuration. Inference cost is measured in model-side milliseconds per fixed bag, including encoder forward computation and a lightweight slide-level head. Table 32 summarizes the compute environment and timing protocol used for this comparison.

Compute accounting. All compute accounting uses the single-GPU worker reported in Table 32. That worker is one NVIDIA RTX PRO 6000 Blackwell GPU with 94.97 GiB VRAM. Since each reported run uses one GPU, wall-clock hours equal GPU-hours.

For one 100-epoch pretraining run, measured TCGA-RCC epoch timings give the following approximate costs. Patch-JEPA 4.17 GPU-hours, DINOv2 3.46, SimCLR 3.00, MAE 1.69, Cross-

MAE 5.83, X-MAG 10.17, and MPCS 6.67. The matched RQ2 objective comparison retrains six SSL baselines across four cohorts, yielding 24 baseline pretraining runs and approximately 123.3 single-GPU hours in total. The four Patch-JEPA cohort pretraining runs add approximately 16.7 single-GPU hours.

Downstream retraining is substantially lighter because all linear probes and MIL heads freeze the patch encoder and train only the downstream head or aggregation module for 20 epochs on each of five folds. The RQ1 MIL baseline grid contains 8 MIL models $\times$ 4 classification cohorts $\times$ 5 folds, giving 160 classification fits, plus 8 MIL models $\times$ 3 TCGA cohorts $\times$ 2 clinical endpoints $\times$ 5 folds, giving 240 survival and stage fits.

Slide-level foundation model baselines are not re-pretrained. Their additional cost is a patch-encoding pass with a frozen encoder for each required model, cohort, and magnification, followed by the same five-fold downstream head training. For the three released slide-level foundation model baselines in RQ1, HIPT, CHIEF, and GigaPath, the main four-cohort comparison requires 12 patch-encoding jobs with frozen encoders. In the paired $1.25\times/5\times$ cache, one retained coarse site corresponds to 16 fine patches, so paired coarse and fine patch encoding costs roughly 17 patch-forward units per coarse site versus 1 unit for $1.25\times$-only extraction. For 384-dimensional FP32 cached representations, this corresponds to about 0.29 MB/slide for $1.25\times$-only representations and 4.88 MB/slide for paired $1.25\times+5\times$ representations, before container and filesystem overhead.

Under this protocol, the efficiency comparison isolates when fine-scale computation is incurred. Methods that retain explicit cross-scale computation at inference must process both coarse and fine inputs in the timed forward path, whereas Patch-JEPA incurs the fine-scale cost during pretraining and keeps only the coarse-scale context encoder at inference. Figure 4 reports the model-side compute trade-off under this matched setup.

D Patch-JEPA Implementation Details

This appendix section gives implementation details specific to Patch-JEPA. These details are kept separate from the downstream evaluation and baseline-comparison protocols in Appendix C.1–C.5. Unless otherwise noted, all settings correspond to the reported coarse-to-fine pretraining configuration, $1.25\times \rightarrow 5\times$, and pretraining optimization uses Adam; downstream supervised probing uses AdamW as described in Appendix C.1.

Algorithm 1 gives the complete pretraining flow, and Table 33 records the reported configuration. The implementation notes below focus on the tensor-grid mapping, projector-axis convention, and predictor complexity that are not visible in the representation-level formulation. Objective-dependent target-branch settings for the RQ2 baselines are listed separately in Table 17.

Section 2 uses representation-level symbols such as S_i^c, $\widetilde{S}_i^c$, S_i^f, $\widetilde{S}_i^f$, $\widetilde{T}_i^f$, $\widetilde{T}_i^c$, and $\widehat{T}_i^f$, while the implementation uses explicit token grids. In the reported ViT-S/16 setting, the coarse-scale context encoder output for S_i^c is stored as a tensor of shape $1 \times 196 \times d$, reshaped as a $14 \times 14 \times d$ grid, with $d = 384$. The fine-scale target encoder outputs $16 \times 196 \times d$ tokens per coarse site, one $196 \times d$ token set for each aligned fine patch. These are reshaped to sixteen $14 \times 14 \times d$ grids and arranged row-major into the $56 \times 56 \times d$ grid corresponding to the ordered fine-scale latent field S_i^f and to the stop-gradient fine-scale target $\widetilde{S}_i^f$. The fine-scale projector expands the masked coarse representation $\widetilde{S}_i^c$ to a $56 \times 56 \times d$ grid for $\widetilde{T}_i^f$ by stride-4 transposed convolution, and the coarse-scale projector aggregates the complete fine-token grid $\widetilde{S}_i^f$ back to a $14\times14\times d$ stop-gradient coarse-scale target $\widetilde{T}_i^c$ by the EMA-tied convolution. For the EMA-tied coarse-scale projector, if the transposed-convolution kernel ω is indexed as $(C_{\text{in}}, C_{\text{out}}, K, K)$, then $\mathcal{P}(\omega)$ swaps only the first two axes to $(C_{\text{out}}, C_{\text{in}}, K, K)$. We do not apply an additional spatial-axis reversal when defining this channel-permuted transpose. The reported predictors match this asymmetry. The coarse-scale predictor is a 2-block, 6-head ViT applied only to the 14×14 coarse-token grid, while the fine-scale predictor is a token-wise linear layer applied after expansion. As a result, predictor-side self-attention is incurred only on 196 coarse tokens, and the fine-scale prediction branch scales linearly with the 3,136 expanded fine tokens because it does not perform self-attention over that grid.

Algorithm 1 Step-by-step Patch-JEPA pretraining flow with a frozen fine-scale target space, an auxiliary fine-scale prediction branch, and a main coarse-scale prediction branch

Sample coarse sites and their aligned fine-patch fields $X = \{(x_i^c, X_i^f)\}_{i=1}^N$.
Sample masked coarse-site set Ω_X, initialize $\mathcal{L} \leftarrow 0$, and keep the fine-scale target encoder frozen.
for $i = 1$ to N **do**
  for $j = 1$ to 16 **do**
    Encode fine representation $S_{i,j}^f = f_{\theta_T}^f(x_{i,j}^f)$.
  end for
  Arrange the 16 fine representations in fixed row-major order to form the ordered fine-scale latent field S_i^f.
  Corrupt coarse input $\widetilde{x}_i^c = \mathcal{M}_c(x_i^c; \mathbf{1}[i \in \Omega_X])$.
  Expand the coarse token/block mask to the aligned fine-token mask $m_i^f = \mathcal{U}_4(m_i^c)$.
  Encode coarse representation $\widetilde{S}_i^c = f_\theta^c(\widetilde{x}_i^c)$.
  if $i \in \Omega_X$ **then**
    Expand to the aligned fine-token-grid geometry $\widetilde{T}_i^f = p_\omega^f(\widetilde{S}_i^c)$.
    Compute the predicted fine-scale latent field $\widehat{T}_i^f = g_\eta^f(\widetilde{T}_i^f)$.
    Define stop-gradient fine-scale target $\widetilde{S}_i^f = \mathcal{M}_f(S_i^f; \mathbf{1}[i \in \Omega_X])$.
    Aggregate stop-gradient coarse-scale target $\widetilde{T}_i^c = p_\psi^c(\widetilde{S}_i^f)$.
    Compute the coarse-scale prediction $\widehat{T}_i^c = g_\phi^c(\widetilde{S}_i^c)$.
    Accumulate site loss $\mathcal{L} \leftarrow \mathcal{L} + \ell_c(\widehat{T}_i^c, \mathrm{sg}(\widetilde{T}_i^c))$.
    Accumulate site loss $\mathcal{L} \leftarrow \mathcal{L} + \lambda\, \ell_f(\widehat{T}_i^f, \mathrm{sg}(\widetilde{S}_i^f); m_i^f)$.
  end if
end for
Normalize loss by masked-site count: $\mathcal{L} \leftarrow \mathcal{L}/|\Omega_X|$.
Update $\theta, \phi, \eta, \omega$ by gradient descent.
Update the coarse-scale projector by EMA: $\psi \leftarrow \alpha\psi + (1-\alpha)\mathcal{P}(\omega)$.

Table 33: Patch-JEPA pretraining hyperparameters and view definitions, including the target-space controls used in RQ2.

Category	Setting
Input and Scale Configuration	
Views and scale	Coarse patch and its aligned fine-patch field at $1.25\times \rightarrow 5\times$
Crops and patch size	224×224 coarse patch with 16×16 ViT patches in both branches
Fine-patch field and token grids	Fixed row-major 4×4 field of aligned 224×224 fine patches per coarse patch (896×896 before partition), with 14×14 coarse and 56×56 assembled fine token grids
Encoders and Objective	
Encoders	ViT-S/16 coarse-scale context encoder and ViT-S/16 fine-scale target encoder
Target space	Initialized with `vit_small_patch16_224.augreg_in1k` and frozen in the reported setting. RQ2 uses either a random frozen target or a same-run EMA target with all other settings unchanged
Losses and predictors	Mean absolute error for the coarse-scale prediction loss and the masked-subset fine-scale prediction loss with $\lambda = 0.1$. The fine-scale predictor is token-wise linear
Masking	
Sampling and source	Uniform sampling with mix 0.0 and a $1.25\times$ tissue mask as the context-mask source
Tissue filtering	White threshold 240 and minimum foreground fraction 0.05
Spatial mask	Token/block mask scale $[0.15, 0.15]$ within selected coarse sites and aspect ratio $[0.75, 1.5]$
Optimization	
Training schedule	Batch size 256, 100 epochs, and Adam optimizer
Optimizer settings	Learning rate 1×10^{-4} and weight decay 1×10^{-6}
Coarse-scale projector EMA	Momentum schedule from 0.999 to 0.9995