Towards Scalable Context-Aware Single-Cell Spatial Transcriptomics Prediction from Histology Images

Zijun Gao[1,2] **Chunbin Gu**[1] **Jinxi Xiang**[2] **Xiangde Luo**[2]✉ **Pheng-Ann Heng**[1]

[1]The Chinese University of Hong Kong [2]Stanford University School of Medicine

Code: github.com/zjgao02/CELLO
Model: hf.co/gaozijun/CELLO
Data: hf.co/datasets/gaozijun/cello_data

Abstract

Predicting gene expression from H&E-stained histology images offers a scalable alternative to costly spatial transcriptomics, yet most existing methods operate at the spot level, where signals from multiple cells are aggregated and critical cellular heterogeneity is obscured. Extending this paradigm to single-cell resolution is non-trivial. Naively applying pathology foundation models faces a scale mismatch: their patch-level representations mix multiple cells, whereas per-cell cropping or resizing distorts morphology and removes local context. Conversely, segmentation-based models without strong pretrained visual encoders often lack the morphological representation capacity needed for accurate molecular prediction and inherit errors from imperfect cell boundary masks. Here, we present CELLO, an efficient end-to-end framework that performs a single pathology foundation model forward pass per image and uses grid sampling to extract location-specific features for all cells simultaneously. We further introduce a distance-decay cross-attention module that refines each cell representation using spatially biased local morphological context. Using 52 public Xenium–H&E pairs from HEST-1k that span 12 organs and approximately 10 million cells, CELLO improves the average predictive accuracy over the evaluated baselines while reducing the mean whole-slide inference time compared to DeepSpot2Cell, a 14.0× speed-up on average that excludes upstream cell segmentation. Our work establishes a scalable foundation for single-cell gene expression prediction from H&E images. Code is available at https://github.com/zjgao02/CELLO.

1 Introduction

Hematoxylin and Eosin (H&E)-stained histology imaging has long been a cornerstone of clinical diagnosis and biomedical research, providing rich morphological details of tissue composition [1]. However, it lacks the molecular depth required to fully understand complex disease mechanisms. Spatial transcriptomics (ST) bridges this gap by mapping gene expression profiles to specific spatial locations, revealing the biological heterogeneity of the tumor microenvironment [2, 3]. Despite its immense value, ST remains far less accessible than routine H&E due to substantial costs and complex experimental protocols [4]. To democratize spatial molecular profiling, numerous computational methods have been developed to directly infer ST profiles from ubiquitous H&E images [5, 6, 7, 8, 9, 10, 11, 12]. These pioneering works primarily focus on spot-level predictions, matching the resolution of widely used platforms like 10x Visium, where each micrometer-sized spot captures a mixture of cells [13].

While spot-level prediction provides valuable regional insights, it fundamentally limits our understanding of cellular heterogeneity. A single Visium spot typically encompasses dozens of cells,

✉Corresponding author: luoxd96@stanford.edu

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

resulting in an averaged gene expression profile that masks distinct cell types, rare cellular states, and intricate cell-cell interactions. As the field advances, emerging imaging-based platforms like 10x Xenium have pushed spatial resolution to the single-cell level. To truly unlock the potential of digital pathology, there is a pressing need to shift the paradigm of H&E-to-ST prediction from spot level directly to single-cell level. Achieving this would enable the extraction of highly granular, cell-specific molecular landscapes from standard histology slides without the need for expensive specialized assays.

Recent research has attempted to push beyond spot-level resolution [14, 15]. Super-resolution approaches, such as iStar [16] and scstGCN [17], generate spatially refined expression maps at the superpixel level, but they still require ST data as input and do not yield true single-cell profiles. More recent cell-level methods differ mainly in how they construct cell representations. DeepSpot2Cell [18] leverages the strong visual representation of pathology foundation models (PFMs) by cropping each segmented cell and encoding it independently, then models spots as bags of cells under weak spot-level supervision. However, this strategy is a naive adaptation of patch-level PFMs to irregular cell-level targets: per-cell cropping and resizing can distort cellular morphology, remove local microenvironmental context, and fail to align naturally with the patch-level representations learned during pretraining. It also requires an independent PFM forward pass for every cell, which is computationally prohibitive for WSIs containing hundreds of thousands to millions of cells. Another line of work, exemplified by GHIST [2], constructs more cell-aware representations from segmentation masks, which can better preserve cell-level spatial structure. However, such models do not fully benefit from the strong morphology representations learned by modern PFMs, and their predictions remain tightly coupled to the quality of upstream cell segmentation, as illustrated in Fig.2.

These limitations highlight the central challenge of single-cell H&E-to-ST prediction: obtaining accurate, scalable, and context-aware cell embeddings. To address this, we propose **CELLO** (**CEL**l-**L**evel gene expression prediction from hist**O**logy), an efficient end-to-end framework that predicts gene expression at cell locations from H&E images. CELLO performs a single pathology foundation model forward pass per image and uses grid sampling to extract location-specific features for all cells simultaneously, avoiding per-cell cropping, boundary-mask dependence, and the ill-posed disentanglement of cell profiles from aggregated spot-level supervision. A distance-decay cross-attention module further refines each cell representation with spatially biased local morphological context, as illustrated in Fig.2D.

In summary, our main contributions are as follows:

- We present a scalable end-to-end framework for single-cell H&E-to-ST prediction that directly trains on paired single-cell ST and H&E images, rather than relying on weakly supervised spot-level expression.
- We drastically reduce computational cost and mitigate the cascading effects of segmentation inaccuracies using grid sampling, which preserves a strong visual representation of PFM.
- We explore multiple context fusion strategies and demonstrate that distance-decay cross attention most effectively incorporates spatially biased local morphological context for strong cell representation.
- Extensive experiments on 52 paired Xenium–WSI samples across 12 organs from HEST-1k show that CELLO achieves state-of-the-art predictive performance and substantially improved computational efficiency that excludes upstream cell segmentation.

2 Related Work

2.1 Spatial Transcriptomics Prediction From H&E Images

Early regression-based methods extract patch-level visual features and map them to expression vectors via transfer learning [5, 19, 7], with later improvements incorporating multi-resolution context [20, 21] or retrieval-based strategies [6, 10]. Beyond regression, recent work has explored contrastive alignment [6], denoising diffusion [12], and flow matching [22] to better capture the stochastic and multimodal nature of gene expression. Despite these methodological advances, nearly all existing approaches operate at the **spot level**, predicting a single expression vector that aggregates signal from hundreds of cells. Several recent efforts attempt to push beyond spot-level granularity, broadly following two strategies: **(1) Super-resolution-based deconvolution [17, 16, 23].** They use hierarchical ViT or graph convolutional networks to produce high-resolution expression maps.

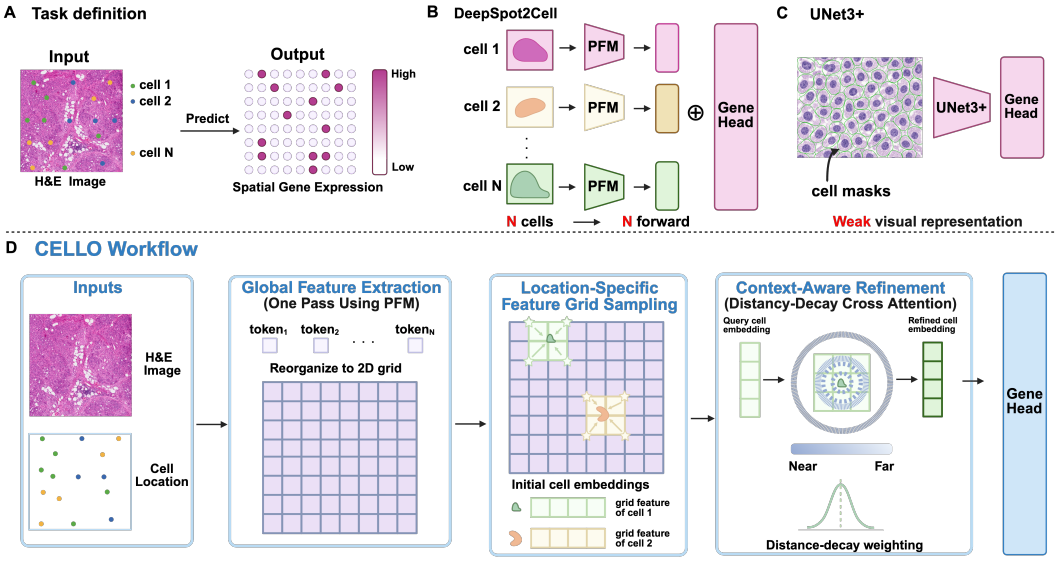


Figure 1: **(A)** Task definition: predicting single-cell spatial gene expression from H&E images with cell locations. **(B)** PFM-based naive cropping methods such as DeepSpot2Cell require N per-cell forward passes and distort local morphology. **(C)** Cell mask based UNet3+ lacks strong pretrained visual representations. **(D)** CELLO performs a single PFM forward pass and uses grid sampling to extract initial feature for all cells simultaneously. A context-aware cross-attention module with 2D RoPE and Gaussian distance-decay bias captures local cellular context and cell–cell interactions.

However, their outputs are high-dimensional superpixel expression maps rather than precise cell transcriptomic profiles, and they still require custom post-processing to approximate. **(2) Spot-level weakly supervised approaches** such as DeepSpot2Cell [18] distributes spot-level expression to individual cells detected within each spot. These methods cause prohibitive inference costs that prevent scaling to whole-slide images containing millions of cells. GHIST [2] builds single cell-aware representations from segmentation masks, better preserving cell-level spatial structure. However, it does not fully exploit the rich morphological features learned by modern PFMs. **CELLO** addresses these limitations by directly training on true pairs of single-cell ST and H&E images and employing an end-to-end, scalable architecture, enabling efficient scaling to whole-slide-level (WSI) analysis.

2.2 Computational Pathology Foundation Models

Foundation models in computational pathology have made great progress in generating patch-level embeddings that transfer across diverse downstream tasks [24]. They are trained on millions of histology images via self-supervised learning objectives including DINO [25], image-text contrastive learning [26], and masked image modeling and have demonstrated that rich morphological and tissue-level information can be captured in compact feature representations [27]. Notable examples include UNI [28], Virchow2 [29], H-Optimus-0 [30]. The latter two are scaled to larger ViT architectures and trained on broader data. In this work, we treat the foundation models as a feature extractor and systematically evaluate them to assess how encoder choice affects cell-level expression prediction.

3 Method

3.1 Problem Formulation

Let $\mathbf{V} \in \mathbb{R}^{L\times L\times 3}$ denote an image extracted from a H&E-stained whole-slide image, where L is the patch size in pixels. Within each patch, we observe a variable number of cells $\{c_i\}_{i=1}^{N}$, each associated with a spatial coordinate $\mathbf{s}_i = (x_i, y_i) \in \mathbb{R}^2$ obtained via spatial transcriptomics or cell segmentation, and a ground-truth gene expression vector $\mathbf{x}_i \in \mathbb{R}^{G}_{\geq 0}$ over a union gene vocabulary of size G.

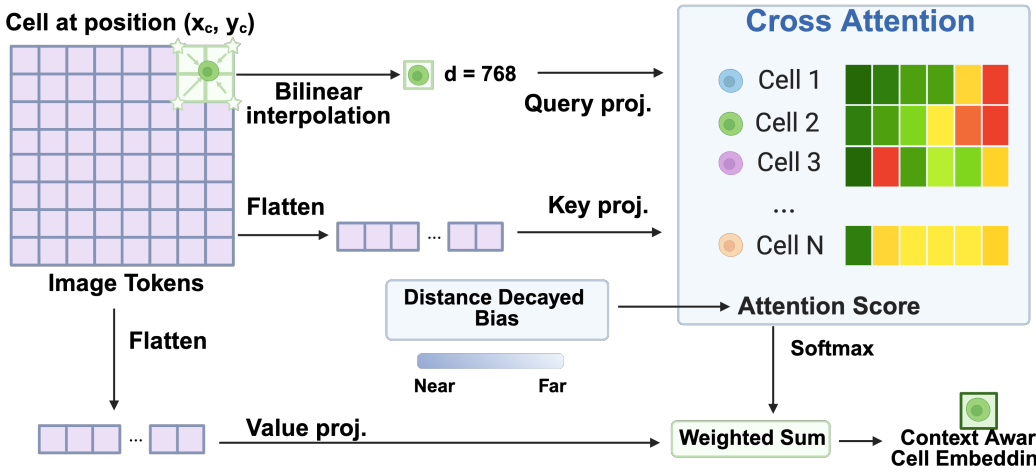


Figure 2: Given the 2D visual token map from a single PFM forward pass, CELLO first queries a location-specific cell feature by bilinear interpolation. The cell query is then refined by distance-decay cross-attention over patch tokens, where spatially nearby tokens receive a larger prior. This produces a context-aware cell embedding that preserves precise cell-location information while incorporating local morphological context.

Prior work typically predicts one expression vector per spot, i.e., $f : \mathbf{V} \mapsto \mathbf{x}$, which cannot resolve cell-level heterogeneity within the patch. We instead formulate H&E-to-ST prediction at single-cell resolution. Given an image tile $\mathbf{V}$ and cell locations $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^{N}$, we learn

$$f_\theta : (\mathbf{V}, \mathbf{s}_i) \longmapsto \hat{\mathbf{x}}_i \in \mathbb{R}^G_{\geq 0}, \quad i = 1, \ldots, N, \tag{1}$$

where $\hat{\mathbf{x}}_i$ denotes the predicted gene expression vector for cell i. This formulation requires extracting spatially precise visual features at each cell location while preserving local morphological context, which motivates our position-aware querying and context-fusion design.

3.2 Visual Feature Extraction

We use a pretrained pathology foundation model f_{enc} as the visual backbone. Given an H&E image patch $\mathbf{V}$, the encoder produces a CLS token $\mathbf{z}_{\texttt{cls}} \in \mathbb{R}^d$ and M spatial tokens $[\mathbf{t}_1, \ldots, \mathbf{t}_M] \in \mathbb{R}^{M \times d}$ from non-overlapping $p \times p$ sub-patches. Because these tokens correspond to a regular image grid, we reshape them into a 2D feature map $\mathbf{T} \in \mathbb{R}^{L_g \times L_g \times d}$, where $L_g = L/p$ and $M = L_g^2$. This preserves the geometric correspondence between visual tokens and image regions, enabling continuous cell-location querying.

3.3 Position-Aware Cell Query Construction

A central design goal is to construct a cell-level representation that is anchored to the cell's precise spatial location within the patch, rather than being tied to a discrete token grid. We achieve this through a two-stage process.

Bilinear Query Initialization. Each cell's spatial coordinate $\mathbf{s}_i$ is recorded in the global whole-slide coordinate system. We first convert it to a patch-local coordinate by subtracting the top-left corner $\mathbf{o} \in \mathbb{R}^2$ of the patch: $\mathbf{s}_i^{\text{loc}} = \mathbf{s}_i - \mathbf{o}$, yielding a position in $[0, L)^2$. We then normalize to the $[-1, 1]$ range required by bilinear sampling:

$$\tilde{\mathbf{s}}_i = \frac{2\,\mathbf{s}_i^{\text{loc}}}{L} - 1. \tag{2}$$

The initial cell query is extracted from the spatial feature map $\mathbf{T}$ via differentiable bilinear interpolation [31]:

$$\mathbf{q}_i^{(0)} = \texttt{GridSample}\big(\mathbf{T},\ \tilde{\mathbf{s}}_i\big) \in \mathbb{R}^d, \tag{3}$$

which produces a continuous, sub-patch feature by smoothly interpolating information from the four nearest spatial tokens surrounding the cell.

Cross-Attention with 2D RoPE and Distance Decay. The initial query $\mathbf{q}_i^{(0)}$ is then refined by attending over all spatial tokens via multi-head cross-attention. To inject explicit spatial structure, we augment the attention mechanism with two complementary positional inductive biases.

First, we apply 2D RoPE [32] to queries and keys. It encodes relative spatial relationships between the cell query and each image token directly into the attention logits without additional parameters. For cell queries, we encode positions in the same coordinate system as visual tokens. Specifically, cell pixel coordinates are divided by the visual token stride to obtain continuous token-grid coordinates, while visual keys use token-center grid coordinates. This avoids feeding raw pixel coordinates into RoPE and aligns the positional scale between cell queries and image tokens.

Second, we add a *distance-decay bias* to the pre-softmax attention scores:

$$b_{i,m} = -\frac{\|\mathbf{s}_i^{\text{loc}} - \mathbf{c}_m\|^2}{2p^2}, \tag{4}$$

where $\mathbf{c}_m \in \mathbb{R}^2$ is the pixel-space center of the m-th spatial token within the patch, computed as $\mathbf{c}_m = (p_m^x \cdot p + p/2,\ p_m^y \cdot p + p/2)$ with (p_m^x, p_m^y) being the grid index of token m. Both $\mathbf{s}_i^{\text{loc}}$ and $\mathbf{c}_m$ are in the same patch-local pixel coordinate system, and the normalization by p^2 makes the decay scale relative to the token receptive field size. Finally, the output is layer-normalized to yield the cell embedding $\mathbf{z}_i \in \mathbb{R}^d$

3.4 Gene Expression Prediction and Training Objectives

The final cell representation $\mathbf{z}_i$ is decoded into a predicted gene expression vector through a two-layer MLP with ReLU activation: $\hat{\mathbf{x}}_i = h_{\text{cell}}(\mathbf{z}_i') \in \mathbb{R}^G_{\geq 0}$. The model is trained with a weighted combination of cell-level and patch-level Huber losses [33]:

$$\mathcal{L} = \mathcal{L}_{\text{cell}} + \lambda\, \mathcal{L}_{\text{patch}}, \tag{5}$$

where $\mathcal{L}_{\text{cell}}$ averages per-gene Huber loss over all cells and active genes. The auxiliary patch-level loss $\mathcal{L}_{\text{patch}}$ is obtained by predicting the aggregate expression within the patch, with the patch-level target defined as the sum of all cell-level expression vectors in that patch, i.e., $\mathbf{x}_{\text{patch}} = \sum_{i=1}^{N} \mathbf{x}_i$. We use Huber loss because CELLO is formulated as supervised regression on normalized gene expression values and is evaluated by Pearson correlation, rather than as generative modeling of raw transcript counts. Compared with mean squared error, Huber loss is less sensitive to heavy-tailed expression values and occasional outliers.

4 Experiments

In this section, we comprehensively evaluate our proposed method. Specifically, we present: (1) benchmark results on test datasets across 12 different organs from human; (2) ablation studies analyzing the impact of different pretrained encoders and context integration; (3) a computational efficiency comparison demonstrating the superior scalability of our approach over baseline methods; and (4) visualizations through representative case studies.

Evaluation Metrics We train a unified gene prediction head over a union vocabulary of 1,915 genes; each sample's panel contains 280–480 genes (Appendix B). Since individual Xenium panels may contain different measured genes, the training loss and evaluation metrics are computed only over the genes observed in the corresponding sample. Our evaluation metric is the mean Pearson Correlation Coefficient (PCC). Following widely used gene selection protocols, we selected three gene sets: top-k most predictive genes (denoted as MPG) [9], top-k most predictive genes from the highly variant genes ordered in mean (denoted as HVG) and spatially variable genes (denoted as SVG) [5]. In the main text, we will only show HVG; the other two gene subsets are shown in the Appendix D. The MPG, HVG, and SVG top-k sets are used only as evaluation subsets of the predicted gene vocabulary and do not affect model training. We also report performance over the full gene set in Appendix D to provide a more comprehensive assessment beyond top-k evaluation subsets.

4.1 10X-Xenium-52 Benchmark

Dataset and Preprocessing We evaluate CELLO and baseline methods on a dataset comprising 52 paired Whole Slide Images (WSIs) and Spatial Transcriptomics (ST) samples sequenced via the Xenium platform from HEST-1k [24]. The dataset spans 12 distinct organs across three health conditions (cancer, diseased, and healthy), with significant variations ranging from 16,429 to 882,692 cells per sample. During preprocessing, WSIs are cropped into 224×224 patches, yielding between 3,465 and 93,515 patches per sample. The total number of cells is approximately 10M. We log-transformed the gene expression following [12].

To rigorously assess the model's generalization capability without data leakage, we form a 4-sample validation set by sampling from the most abundant organs (lung, breast, bowel). The test set is carefully designed to evaluate model performance under two distinct settings:

- **In-Distribution (ID) Setting:** Evaluates performance on six organ types (Lung, Breast, Bowel, Skin, Pancreas, and Lymphoid). For these samples, both the organ types and their associated health conditions were observed during training, assessing the model's capacity to exploit statistical regularities within well-represented categories.
- **Out-of-Distribution (OOD) Setting:** provides case studies of transfer to organs or conditions absent from training. This includes four **completely unseen** organs (Brain, Bone, Heart, and Ovary), each represented by a single WSI reserved exclusively for testing. The other two organs (Kidney and Liver) were seen during training, but their specific health conditions present at test time were never encountered.

All remaining WSIs are allocated to the training set. Dataset details are provided in Appendix B

Baseline Methods Single-cell H&E-to-ST prediction remains under-explored, and only a few recent methods explicitly study how to construct cell embeddings from histology. The first strategy uses segmentation-aware CNN features, as in GHIST [2], where UNet3+ is used to extract cell representations from cell masks. This design preserves mask-defined cell structure but lacks strong visual representation without leveraging modern PFMs. The second strategy uses PFM features from per-cell crops, as in DeepSpot2Cell [18]. While this benefits from strong pretrained visual representations, it requires cell cropping and separate feature extraction for each cell, which is computationally expensive and can distort cell morphology or remove local context. Finally, to isolate the contribution of contextual fusion, we include a grid-sampling baseline, denoted as *Grid*, which queries cell features from the shared patch-level feature map using cell locations but does not apply distance-decay cross-attention. Training details are provided in Appendix C.

Results As shown in Table 1, CELLO substantially outperforms UNet3+ and DeepSpot2Cell in both ID and OOD settings, demonstrating the effectiveness of constructing cell embeddings through location-specific querying over shared PFM features. In the ID setting, CELLO improves over the strongest baseline, DeepSpot2Cell, by 43.8% on HVG PCC-50. This advantage becomes even larger in the OOD setting, where CELLO achieves a 62.4% relative improvement on HVG PCC-50, indicating stronger robustness under unseen tissue and disease conditions. In addition, the comparison with *Grid*, an ablated version of CELLO without the distance-decay cross-attention module, further validates the importance of contextual refinement. CELLO improves over Grid by 7.8% and 10.1% on HVG PCC-50 in the ID and OOD settings, respectively, showing that spatially biased cross-attention provides complementary local morphological context beyond location-based feature querying alone.

4.2 Ablation Study

In this section, we conduct comprehensive ablation studies to validate the key design choices of CELLO. Specifically, we first investigate the impact of selecting different pathology foundation models as our vision encoder. Furthermore, we explore a different context for fusion to identify which context truly matters in this task.

Choice of Foundation Models To ensure a fair comparison among different pre-trained pathology foundation models and to prevent the introduction of potential confounding factors, we abstain from using any data augmentation techniques in these experiments. Specifically, we evaluate three representative models: UNI [28], Virchow2 [29], H-Optimus-0 [30]. These models vary in their

Table 1: **HVG results on the 10X-Xenium-52 benchmark.** ID tissues are shown on the left and OOD tissues on the right. Higher PCC values are better. The best results within each tissue are highlighted in **bold**.

In-distribution (ID)					Out-of-distribution (OOD)				
Tissue	Model	HVG-10	HVG-50	HVG-200	Tissue	Model	HVG-10	HVG-50	HVG-200
Lung	UNet3+	0.4293	0.2177	0.0824	Kidney	UNet3+	0.1828	0.1189	0.0367
	DeepSpot2Cell	0.4082	0.2264	0.0763		DeepSpot2Cell	0.3303	0.1803	0.0610
	Grid	0.5481	0.3730	0.1455		Grid	0.4728	0.3147	0.1235
	CELLO	**0.5537**	**0.3827**	**0.1492**		**CELLO**	**0.5625**	**0.3468**	**0.1369**
Breast	UNet3+	0.3078	0.1910	0.0546	Heart	UNet3+	0.2141	0.1209	0.0430
	DeepSpot2Cell	0.5228	0.3583	0.1340		DeepSpot2Cell	0.2018	0.1106	0.0349
	Grid	0.6325	0.4932	0.2318		Grid	0.3490	0.1868	0.0672
	CELLO	**0.6736**	**0.5243**	**0.2541**		**CELLO**	**0.4015**	**0.2318**	**0.0842**
Bowel	UNet3+	0.3868	0.2615	0.1058	Liver	UNet3+	0.3957	0.2616	0.0839
	DeepSpot2Cell	0.6358	0.4163	0.1494		DeepSpot2Cell	0.4622	0.3410	0.1218
	Grid	0.6742	0.4821	0.2125		Grid	0.5217	0.3932	0.1570
	CELLO	**0.6946**	**0.5170**	**0.2340**		**CELLO**	**0.5483**	**0.4305**	**0.1826**
Skin	UNet3+	0.3806	0.1985	0.0671	Bone	UNet3+	0.1042	0.0553	0.0122
	DeepSpot2Cell	0.5219	0.2689	0.0854		DeepSpot2Cell	0.1351	0.0638	0.0164
	Grid	0.6444	0.3520	0.1232		Grid	0.1626	0.0837	0.0244
	CELLO	**0.6765**	**0.3879**	**0.1358**		**CELLO**	**0.1897**	**0.0960**	**0.0282**
Pancreas	UNet3+	0.2761	0.1709	0.0589	Brain	UNet3+	0.2021	0.0875	0.0256
	DeepSpot2Cell	0.3123	0.1977	0.0580		DeepSpot2Cell	0.2054	0.1011	0.0275
	Grid	0.3382	0.2197	0.0864		Grid	0.4254	0.1921	0.0596
	CELLO	**0.4101**	**0.2790**	**0.1144**		**CELLO**	**0.4920**	**0.2123**	**0.0636**
Lymphoid	UNet3+	0.3606	0.1870	0.0579	Ovary	UNet3+	0.4089	0.1929	0.0572
	DeepSpot2Cell	0.3451	0.1959	0.0520		DeepSpot2Cell	0.3957	0.1840	0.0526
	Grid	**0.5029**	0.3001	0.0851		Grid	0.5184	**0.2773**	0.0857
	CELLO	0.4848	**0.3019**	**0.0899**		**CELLO**	**0.5219**	0.2760	**0.0876**
Average	UNet3+	0.3569	0.2044	0.0711	Average	UNet3+	0.2513	0.1395	0.0431
	DeepSpot2Cell	0.4577	0.2773	0.0925		DeepSpot2Cell	0.2884	0.1635	0.0524
	Grid	0.5567	0.3700	0.1474		Grid	0.4083	0.2413	0.0862
	CELLO	**0.5822**	**0.3988**	**0.1629**		**CELLO**	**0.4526**	**0.2656**	**0.0972**

architectural specifications: UNI utilizes a patch size of 16×16 with a feature dimension of 1024, whereas both Virchow2 and H-optimus-0 employ a 14×14 patch size, yielding feature dimensions of 1280 and 768, respectively.

As illustrated in Fig. 3A, across MPG, HVG, and SVG evaluation sets, Virchow2 achieves the best average performance among the three evaluated encoders. On PCC-50, Virchow2 obtains average scores of 0.4512, 0.3988, and 0.4492 for MPG, HVG, and SVG, respectively, outperforming H0 by 5.9%, 4.2%, and 6.1%, and UNI by 9.8%, 9.3%, and 9.8%. H0 also provides modest but consistent gains over UNI across all three gene sets. These results suggest that stronger pathology foundation representations improve the quality of cell embeddings, and motivate our use of Virchow2 as the default visual encoder in CELLO.

What Context Really Matters A natural question arising from our architecture design is what type of contextual information truly matters for single-cell gene expression prediction. Our baseline model primarily captures context through local cell-cell interactions within the spatial neighborhood of each target patch, reflecting the biological intuition that a cell's transcriptomic state is shaped by its immediate microenvironment. However, it remains unclear whether supplementing this local context with broader, slide-level information could further enhance prediction. To investigate this, we conduct experiments that augment final cell features with the CLS token from the vision encoder, which encodes a global summary of the entire histology image. Specifically, for each cell embedding $\mathbf{z}_i$ and the corresponding patch CLS token $\mathbf{z}_{\texttt{cls}}$, we learn a scalar gate to adaptively fuse local cell-level and global patch-level context:

$$g_i = \sigma(\phi_g([\mathbf{z}_i; \mathbf{z}_{\texttt{cls}}])), \qquad \tilde{\mathbf{z}}_i = g_i \mathbf{z}_i + (1 - g_i)\mathbf{z}_{\texttt{cls}},$$

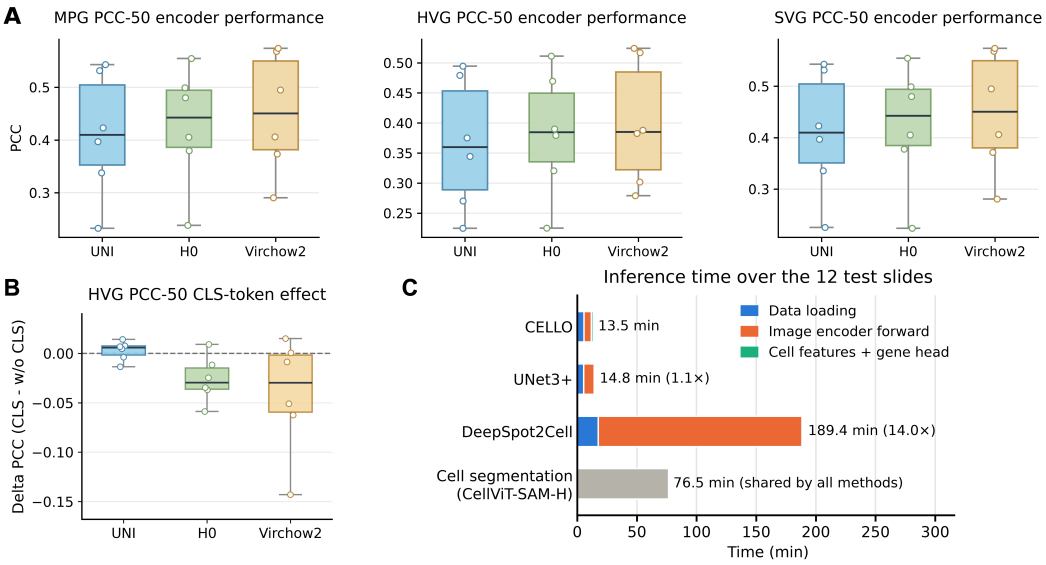


Figure 3: **Ablation and efficiency analysis of CELLO. (A)** Performance comparison of different vision encoders on ID settings. Columns show, from left to right: MPG-10, HVG-10, SVG-10. **(B)** Effect of CLS-token fusion. Mean ΔPCC is computed as PCC (with cls) - PCC (w/o cls), averaged across different vision encoders. **(C)** Total inference time over the 12 test slides, split into data loading, image-encoder forward pass, and cell features with gene head. Cell segmentation (CellViT-SAM-H), required by all methods, is shown separately. CELLO is 14.0$\times$ faster than DeepSpot2Cell.

where $[\cdot;\cdot]$ denotes concatenation, ϕ_g is a two-layer MLP with GELU activation, and $g_i \in (0,1)$ controls the contribution of local versus global context for cell i. Then $\tilde{\mathbf{z}}_i$ will be served as final cell embedding instead of being used directly for calculating $\mathcal{L}_{\text{patch}}$.

If high-level contextual information were beneficial, one would expect consistent improvements from this augmentation. Instead, as we show in Fig. 3B, incorporating the CLS token consistently degrades performance, providing strong empirical evidence that single-cell gene expression prediction is predominantly driven by fine-grained, local morphological context rather than abstract global semantics. We provide detailed ID and OOD results in Appendix E.

4.3 Computational Efficiency

A critical consideration for practical deployment on whole-slide images (WSIs) is inference speed. Existing methods suffer from inherent scalability bottlenecks. DeepSpot2Cell requires a separate forward pass through a frozen pathology foundation model (PFM) for each individual cell tile, in addition to the spot tile and neighboring spot tiles. Since a single WSI can contain hundreds of thousands to millions of cells, this per-cell PFM inference introduces substantial computational overhead that scales linearly with cell count, making it prohibitively expensive for large-scale slides.

Given cell locations, CELLO and UNet3+ take comparable time on the 12 test slides: CELLO needs 67.5 ± 48.8 s per slide and UNet3+ 73.9 ± 52.1 s. Across slides, the UNet3+/CELLO time ratio ranges from 0.99 to 1.21 (median 1.12$\times$), and it is smallest on the largest slides. Most of CELLO's cost is the single Virchow2 forward pass per patch, which takes 81.8% of its GPU time; grid sampling, cross-attention, CLS fusion and the gene head together add less than 20%. When cell segmentation with CellViT-SAM-H is included (382.5 s per slide on average), the two pipelines are on par (median 1.02$\times$). Compared with dense mask-based models such as UNet3+, the advantage of CELLO is therefore higher accuracy at a similar cost rather than speed; the large speed-up applies to methods that run the foundation model once per cell, as discussed next.

We also compare CELLO with DeepSpot2Cell, which runs the foundation model once per cell (Fig. 3C). On the 12 test slides, DeepSpot2Cell needs 947.2 ± 1068.3 s per slide, 14.0$\times$ the time of CELLO on average (median 11.7$\times$, range 3.5–26.3$\times$). The speed-up grows with cell density

(Spearman $\rho = 0.97$ with the mean number of cells per patch), because DeepSpot2Cell runs one forward pass per cell whereas CELLO runs one per patch: 114,429 patches versus 2,996,926 cell crops in total. Including cell segmentation, CELLO remains 2.5× faster (median over slides).

These results demonstrate that CELLO not only exceeds the prediction accuracy of existing approaches but does so with substantially greater computational efficiency, making it a practical solution for large-scale spatial transcriptomics inference from H&E images.

4.4 Case Study

Figure 4 visualizes H&E images, ground-truth expression, and model predictions across six representative test tissues using genes KRT8, SCGB1A1, COL17A1, DES, AQP3, and MYH11 [34, 35, 36, 37, 38, 39]. Across these heterogeneous morphologies, CELLO closely recovers the spatial localization of measured gene expression.

Compared with UNet3+, CELLO produces sharper and more spatially faithful expression maps, preserving both high-expression domains and low-expression background regions. Quantitatively, CELLO achieves substantially higher Pearson correlations with ground truth for all six markers (0.746, 0.735, 0.886, 0.901, 0.829, and 0.646), whereas UNet3+ shows weaker localization and lower correlations, particularly for KRT8, DES, and MYH11. These qualitative results demonstrate that CELLO can infer cell-level, spatially resolved transcriptional programs directly from H&E morphology and better capture local tissue context than mask-based baseline prediction.

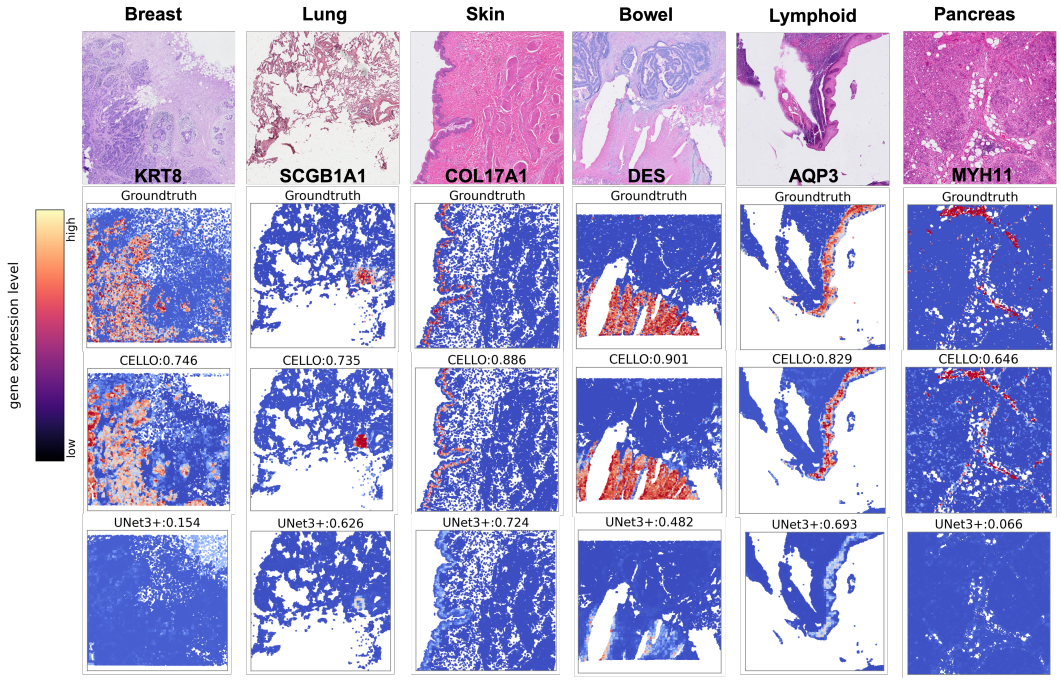


Figure 4: **Visualization of spatial gene expression predicted from H&E.** Rows show, from top to bottom: H&E image, measured gene expression, CELLO predictions, and UNet3+ predictions. Compared with UNet3+, CELLO produced substantially higher correlations with ground-truth gene expression.

5 Conclusion

In this work, we propose CELLO, a scalable end-to-end framework that predicts single-cell spatial transcriptomics from H&E-stained histology images. By performing a single pathology foundation model forward pass per image and leveraging grid sampling to simultaneously extract features for all cells, CELLO eliminates the per-cell inference bottleneck. The distance-decay cross-attention module encodes biologically motivated spatial priors, enabling each cell to attend preferentially

to its local morphological neighborhood while still capturing broader tissue context. Extensive experiments on 52 Xenium whole-slide images spanning 12 organs demonstrate that CELLO achieves state-of-the-art prediction accuracy in both ID and OOD settings while being 14.0× faster on average than DeepSpot2Cell at inference, excluding cell segmentation. We hope CELLO establishes a strong foundation for scalable, single-cell resolution morphological-to-molecular translation.

Acknowledgments and Disclosure of Funding

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project T45-401/22-N. The authors declare no competing interests.

References

[1] Andrew H Song, Guillaume Jaume, Drew FK Williamson, Ming Y Lu, Anurag Vaidya, Tiffany R Miller, and Faisal Mahmood. Artificial intelligence for digital and computational pathology. *Nature Reviews Bioengineering*, 1(12):930–949, 2023.

[2] Xiaohang Fu, Yue Cao, Beilei Bian, Chuhan Wang, Dinny Graham, Nirmala Pathmanathan, Ellis Patrick, Jinman Kim, and Jean Yee Hwa Yang. Spatial gene expression at single-cell resolution from histology using deep learning with ghist. *Nature methods*, 22(9):1900–1910, 2025.

[3] Sabrina M Lewis, Marie-Liesse Asselin-Labat, Quan Nguyen, Jean Berthelet, Xiao Tan, Verena C Wimmer, Delphine Merino, Kelly L Rogers, and Shalin H Naik. Spatial omics and multiplexed imaging to explore cancer biology. *Nature methods*, 18(9):997–1012, 2021.

[4] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.

[5] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Åke Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature biomedical engineering*, 4(8):827–834, 2020.

[6] Ronald Xie, Kuan Pang, Sai Chung, Catia Perciani, Sonya MacParland, Bo Wang, and Gary Bader. Spatially resolved gene expression prediction from histology images via bi-modal contrastive learning. *Advances in Neural Information Processing Systems*, 36:70626–70637, 2023.

[7] Yuansong Zeng, Zhuoyi Wei, Weijiang Yu, Rui Yin, Yuchen Yuan, Bingling Li, Zhonghui Tang, Yutong Lu, and Yuedong Yang. Spatial transcriptomics prediction from histology jointly through transformer and graph neural networks. *Briefings in Bioinformatics*, 23(5):bbac297, 2022.

[8] Yuran Jia, Junliang Liu, Li Chen, Tianyi Zhao, and Yadong Wang. Thitogene: a deep learning method for predicting spatial transcriptomics from histological images. *Briefings in Bioinformatics*, 25(1):bbad464, 2024.

[9] Kalin Nonchev, Sebastian Dawo, Karina Silina, Holger Moch, Sonali Andani, Tumor Profiler Consortium, Viktor H Koelzer, and Gunnar Rätsch. Deepspot: Leveraging spatial context for enhanced spatial transcriptomics prediction from h&e images. *MedRxiv*, pages 2025–02, 2025.

[10] Yan Yang, Md Zakir Hossain, Eric A Stone, and Shafin Rahman. Exemplar guided deep neural network for spatial transcriptomics analysis of gene expression prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5039–5048, 2023.

[11] Wenwen Min, Zhiceng Shi, Jun Zhang, Jun Wan, and Changmiao Wang. Multimodal contrastive learning for spatial gene expression prediction using histology images. *Briefings in Bioinformatics*, 25(6):bbae551, 2024.

[12] Sichen Zhu, Yuchen Zhu, Molei Tao, and Peng Qiu. Diffusion generative modeling for spatially resolved gene expression inference from histology images. *arXiv preprint arXiv:2501.15598*, 2025.

[13] Lambda Moses and Lior Pachter. Museum of spatial transcriptomics. *Nature methods*, 19(5):534–546, 2022.

[14] Ludvig Bergenstråhle, Bryan He, Joseph Bergenstråhle, Xesús Abalo, Reza Mirzazadeh, Kim Thrane, Andrew L Ji, Alma Andersson, Ludvig Larsson, Nathalie Stakenborg, et al. Super-resolved spatial transcriptomics by deep data fusion. *Nature biotechnology*, 40(4):476–479, 2022.

[15] Chao-Hui Huang, Yoson Park, Jincheng Pang, and Jadwiga R Bienkowska. Single-cell gene expression prediction using h&e images based on spatial transcriptomics. In *Medical Imaging 2023: Digital and Computational Pathology*, volume 12471, pages 17–25. SPIE, 2023.

[16] Daiwei Zhang, Amelia Schroeder, Hanying Yan, Haochen Yang, Jian Hu, Michelle YY Lee, Kyung S Cho, Katalin Susztak, George X Xu, Michael D Feldman, et al. Inferring super-resolution tissue architecture by integrating spatial transcriptomics with histology. *Nature biotechnology*, 42(9):1372–1377, 2024.

[17] Shuailin Xue, Fangfang Zhu, Jinyu Chen, and Wenwen Min. Inferring single-cell resolution spatial gene expression via fusing spot-based spatial transcriptomics, location, and histology using gcn. *Briefings in Bioinformatics*, 26(1):bbae630, 2025.

[18] Kalin Nonchev, Glib Manaiev, Viktor H Koelzer, and Gunnar Rätsch. Deepspot2cell: Predicting virtual single-cell spatial transcriptomics from h&e images using spot-level supervision. *bioRxiv*, pages 2025–09, 2025.

[19] Minxing Pang, Kenong Su, and Mingyao Li. Leveraging information in spatial transcriptomics to predict super-resolution gene expression from histology images in tumors. *BioRxiv*, pages 2021–11, 2021.

[20] Youngmin Chung, Ji Hun Ha, Kyeong Chan Im, and Joo Sang Lee. Accurate spatial gene expression prediction by integrating multi-resolution features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11591–11600, 2024.

[21] Hongyi Wang, Xiuju Du, Jing Liu, Shuyi Ouyang, Yen-Wei Chen, and Lanfen Lin. M2ort: many-to-one regression transformer for spatial transcriptomics prediction from histopathology images. *arXiv preprint arXiv:2401.10608*, 2024.

[22] Tinglin Huang, Tianyu Liu, Mehrtash Babadi, Wengong Jin, and Rex Ying. Scalable generation of spatial transcriptomics from histology images via whole-slide flow matching. *arXiv preprint arXiv:2506.05361*, 2025.

[23] Edward Zhao, Matthew R Stone, Xing Ren, Thomas Pulliam, Paul Nghiem, Jason H Bielas, and Raphael Gottardo. Bayesspace enables the robust characterization of spatial gene expression architecture in tissue sections at increased resolution. *bioRxiv*, pages 2020–09, 2020.

[24] Guillaume Jaume, Paul Doucet, Andrew H Song, Ming Y Lu, Cristina Almagro-Pérez, Sophia J Wagner, Anurag J Vaidya, Richard J Chen, Drew F Williamson, Ahrong Kim, et al. Hest-1k: A dataset for spatial transcriptomics and histology image analysis. *Advances in Neural Information Processing Systems*, 37:53798–53833, 2024.

[25] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[27] Weiqing Chen, Pengzhi Zhang, Tu N Tran, Yiwei Xiao, Shengyu Li, Vrutant V Shah, Hao Cheng, Kristopher W Brannan, Keith Youker, Li Lai, et al. A visual–omics foundation model to bridge histopathology with spatial transcriptomics. *Nature Methods*, 22(7):1568–1582, 2025.

[28] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature medicine*, 30(3):850–862, 2024.

[29] Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, David Klimstra, Razik Yousfi, et al. Virchow2: Scaling self-supervised mixed magnification models in pathology. *arXiv preprint arXiv:2408.00738*, 2024.

[30] Charlie Saillard, Rodolphe Jenatton, Felipe Llinares-López, Zelda Mariet, David Cahané, Eric Durand, and Jean-Philippe Vert. H-optimus-0 (2024). *URL https://github.com/bioptimus/releases/tree/main/models/h-optimus/v0*, 2(6):7, 2024.

[31] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[32] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[33] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[34] Roland Moll, Werner W Franke, Dorothea L Schiller, Benjamin Geiger, and Reinhard Krepler. The catalog of human cytokeratins: patterns of expression in normal epithelia, tumors and cultured cells. *Cell*, 31(1):11–24, 1982.

[35] Emma L Rawlins, Tadashi Okubo, Yan Xue, David M Brass, Richard L Auten, Hiroshi Hasegawa, Fan Wang, and Brigid LM Hogan. The role of scgb1a1+ clara cells in the long-term maintenance and repair of lung airway, but not alveolar, epithelium. *Cell stem cell*, 4(6):525–534, 2009.

[36] Hiroyuki Matsumura, Yasuaki Mohri, Nguyen Thanh Binh, Hironobu Morinaga, Makoto Fukuda, Mayumi Ito, Sotaro Kurata, Jan Hoeijmakers, and Emi K Nishimura. Hair follicle aging is driven by transepidermal elimination of stem cells via col17a1 proteolysis. *Science*, 351(6273):aad4395, 2016.

[37] Mariko Hara-Chikuma, Shunsuke Chikuma, Yoshinori Sugiyama, Kenji Kabashima, Alan S Verkman, Shintaro Inoue, and Yoshiki Miyachi. Chemokine-dependent t cell migration requires aquaporin-3–mediated hydrogen peroxide uptake. *Journal of Experimental Medicine*, 209(10):1743–1752, 2012.

[38] Denise Paulin and Zhenlin Li. Desmin: a major intermediate filament protein essential for the structural integrity and function of muscle. *Experimental cell research*, 301(1):1–7, 2004.

[39] Gary K Owens, Meena S Kumar, and Brian R Wamhoff. Molecular regulation of vascular smooth muscle cell differentiation in development and disease. *Physiological reviews*, 84(3):767–801, 2004.

[40] Michelle M Li, Yepeng Huang, Marissa Sumathipala, Man Qing Liang, Alberto Valdeolivas, Ashwin N Ananthakrishnan, Katherine Liao, Daniel Marbach, and Marinka Zitnik. Contextual ai models for single-cell protein biology. *Nature Methods*, 21(8):1546–1557, 2024.

[41] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[42] Suyuan Zhao, Yizhen Luo, Ganbo Yang, Yan Zhong, Hao Zhou, and Zaiqing Nie. Stofm: a multi-scale foundation model for spatial transcriptomics. *arXiv preprint arXiv:2507.11588*, 2025.

[43] Zixuan Cang, Yanxiang Zhao, Axel A Almet, Adam Stabell, Raul Ramos, Maksim V Plikus, Scott X Atwood, and Qing Nie. Screening cell–cell communication in spatial transcriptomics via collective optimal transport. *Nature methods*, 20(2):218–228, 2023.

A Additional Related Work

A.1 Context-Aware Modeling in Computational Biology

Biological processes are inherently context-dependent: a protein's function varies across tissue microenvironments, a gene's expression is shaped by neighboring cells, and a molecule's binding affinity shifts with its interaction partners. Explicitly encoding such contextual priors into model design has proven effective across a wide range of biological tasks. PINNACLE [40] introduces a geometric deep learning framework that generates context-aware protein representations by modeling protein-protein interactions conditioned on diverse biological contexts, substantially improving embedding quality. AlphaFold [41] leverages multiple sequence alignments (MSAs) to retrieve evolutionary co-variation signals from homologous sequences, providing rich phylogenetic context that is critical to its breakthrough accuracy in protein structure prediction. In spatial transcriptomics, SToFM [42] constructs multi-scale contextual representations, from subcellular to tissue levels, to pretrain a more expressive foundation model. COMMOT [43] explicitly models the spatial context of receptor-ligand co-localization to infer intercellular communication patterns. A common thread across these advances is that incorporating biologically motivated inductive biases, rather than relying solely on increased model parameters, can yield substantial performance gains. This perspective suggests that *context* constitutes a complementary and largely under-explored scaling axis: while most recent progress in biological AI has been driven by scaling model parameters and training data, enriching the contextual information available to a model may offer an equally impactful path toward improved performance. CELLO embraces this philosophy by equipping each cell query with explicit spatial position context and surrounding tissue morphology.

B Dataset Details

Data source, registration and preprocessing. All 52 H&E–Xenium pairs are public samples from HEST-1k [24]; we generated no new tissue or sequencing data. For every Xenium sample, the H&E image and the transcript readout come from the same tissue section: the section is stained with H&E and re-imaged after in situ decoding, and 10x Genomics provides an affine transform between the two images. HEST-1k uses this alignment, re-estimates the pixel size of each slide and converts all samples to one coordinate convention. We use the HEST-1k registration and quality control as provided and perform no additional alignment. Each cell is represented by its segmentation centroid in whole-slide pixel coordinates. Each WSI is tiled into non-overlapping 224×224 patches on a grid anchored at the slide origin, and a patch is kept if it lies entirely inside the slide and contains at least one cell centroid, giving 520,402 patches and 9.53M cells. Expression is $\log(1 + x)$ of counts normalized to 100 per cell. The final split contains 36 training samples, 4 validation samples, and 12 test samples, with the detailed organ, health-condition, and sample-ID composition reported in Table 2.

The curated dataset spans 12 organ types with substantial heterogeneity in both tissue origin and sample scale. As shown in Fig. 5A, Lung is the most heavily represented organ with 22 samples, followed by Breast and Bowel (6 each), Skin (4), Lymphoid and Pancreas (3 each), Liver and Kidney (2 each). Brain, Ovary, Bone and Heart each contribute only 1 sample, yielding a long-tailed distribution across 52 total samples. This imbalance creates a realistic yet challenging benchmark: well-represented organs support in-distribution learning, while single-sample organs provide a stringent test of cross-organ generalization.

Importantly, total cell and patch counts are not simply proportional to sample numbers (Fig. 5B). Although Lung has the most samples, Breast contributes the largest aggregate cell count, followed by Bowel and Lung. Lymphoid and Kidney also contain substantial cell populations despite having only 3 and 2 samples each, indicating pronounced variation in tissue density and slide coverage.

At the single-sample level (Fig. 5C), cell counts span more than one order of magnitude—from roughly 20–40 K (Heart, Bone) to 1M (individual Lymphoid outliers). Lung samples are numerous but relatively moderate in per-slide size, whereas Breast, Bowel and Lymphoid include considerably larger individual slides. Brain is especially notable: a single sample containing about 1M cells, making it an important out-of-distribution stress case that combines high cell count with minimal training support.

Table 2: Dataset split composition of the 10X-Xenium-52. Sample IDs are grouped by data split, organ type, and health condition.

Split	Organ	Health condition	# samples	Sample IDs
Train	Bowel	Cancer	3	TENX111, TENX139, TENX147
Train	Bowel	Healthy	1	TENX114
Train	Breast	Cancer	4	NCBI783, TENX96, TENX97, TENX98
Train	Kidney	Cancer	1	TENX105
Train	Liver	Healthy	1	TENX121
Train	Lung	Cancer	1	TENX141
Train	Lung	Diseased	15	NCBI856, NCBI857, NCBI858, NCBI859, NCBI860, NCBI861, NCBI864, NCBI865, NCBI866, NCBI867, NCBI870, NCBI873, NCBI880, NCBI881, NCBI882
Train	Lung	Healthy	3	NCBI875, NCBI876, NCBI884
Train	Lymphoid	Cancer	1	TENX134
Train	Lymphoid	Healthy	1	TENX133
Train	Pancreas	Cancer	2	TENX126, TENX140
Train	Skin	Cancer	2	TENX115, TENX117
Train	Skin	Healthy	1	TENX123
Val	Bowel	Cancer	1	TENX149
Val	Breast	Cancer	1	NCBI785
Val	Lung	Cancer	1	TENX118
Val	Lung	Healthy	1	NCBI883
Test	Bone	Healthy	1	TENX132
Test	Bowel	Cancer	1	TENX148
Test	Brain	Cancer	1	TENX138
Test	Breast	Cancer	1	NCBI784
Test	Heart	Healthy	1	TENX119
Test	Kidney	Healthy	1	TENX106
Test	Liver	Cancer	1	TENX120
Test	Lung	Healthy	1	NCBI879
Test	Lymphoid	Diseased	1	TENX124
Test	Ovary	Cancer	1	TENX142
Test	Pancreas	Cancer	1	TENX116
Test	Skin	Healthy	1	TENX122

Finally, the disease-state composition is also heavily imbalanced (Fig. 5D). Lung and Lymphoid contain all three categories. Bowel, Skin, Liver and Kidney mix cancer and healthy tissue while Bone and Heart are exclusively healthy, and Pancreas and Ovary are exclusively cancer. Together, these distributions underscore that the benchmark evaluates not only prediction accuracy but also robustness to organ imbalance, slide-scale variation and heterogeneous disease contexts.

Gene panels. Xenium panels differ between samples. The model predicts a union vocabulary of 1,915 features, and for each sample the loss and all metrics use only the genes measured in that sample. Excluding Xenium negative-control and antisense probes, a panel contains 280–480 genes (Table 3). Table 4 lists every sample.

Table 3: Measured genes per sample, excluding Xenium control probes.

Split	Samples	Mean	Min	Median	Max
Train	36	365	280	343	480
Val	4	364	313	360	422
Test	12	414	313	400	480

Table 4: Per-sample statistics of the 10X-Xenium-52 (genes exclude Xenium control probes).

Split	Organ	Condition	Sample	Genes	Cells	Patches
Train	Bowel	Cancer	TENX111	425	587,016	13,816
Train	Bowel	Cancer	TENX139	480	388,015	8,858
Train	Bowel	Cancer	TENX147	422	275,875	9,997
Train	Bowel	Healthy	TENX114	325	270,788	37,902
Train	Breast	Cancer	NCBI783	280	141,829	8,177
Train	Breast	Cancer	TENX96	380	365,592	24,657
Train	Breast	Cancer	TENX97	380	574,426	33,795
Train	Breast	Cancer	TENX98	280	882,692	85,887
Train	Kidney	Cancer	TENX105	377	56,504	3,461
Train	Liver	Healthy	TENX121	377	239,197	15,332
Train	Lung	Cancer	TENX141	480	160,448	8,073
Train	Lung	Diseased	NCBI856	343	46,277	4,134
Train	Lung	Diseased	NCBI857	343	59,609	5,018
Train	Lung	Diseased	NCBI858	343	16,429	2,594
Train	Lung	Diseased	NCBI859	342	26,389	3,193
Train	Lung	Diseased	NCBI860	343	34,807	2,811
Train	Lung	Diseased	NCBI861	343	45,650	3,083
Train	Lung	Diseased	NCBI864	343	107,973	8,581
Train	Lung	Diseased	NCBI865	343	133,193	9,568
Train	Lung	Diseased	NCBI866	343	72,153	4,950
Train	Lung	Diseased	NCBI867	343	140,660	10,983
Train	Lung	Diseased	NCBI870	343	24,643	2,342
Train	Lung	Diseased	NCBI873	343	27,971	3,597
Train	Lung	Diseased	NCBI880	343	38,285	3,242
Train	Lung	Diseased	NCBI881	343	40,775	3,488
Train	Lung	Diseased	NCBI882	343	52,160	4,417
Train	Lung	Healthy	NCBI875	343	31,189	2,365
Train	Lung	Healthy	NCBI876	343	12,915	1,421
Train	Lung	Healthy	NCBI884	343	73,783	5,151
Train	Lymphoid	Cancer	TENX134	477	225,339	4,392
Train	Lymphoid	Healthy	TENX133	477	78,904	3,796
Train	Pancreas	Cancer	TENX126	377	140,194	5,099
Train	Pancreas	Cancer	TENX140	380	234,859	9,211
Train	Skin	Cancer	TENX115	282	106,791	4,297
Train	Skin	Cancer	TENX117	382	87,499	9,457
Train	Skin	Healthy	TENX123	377	62,469	14,630
Val	Bowel	Cancer	TENX149	422	307,385	9,420
Val	Breast	Cancer	NCBI785	313	166,364	6,280
Val	Lung	Cancer	TENX118	377	161,894	4,883
Val	Lung	Healthy	NCBI883	343	34,674	2,829
Test	Bone	Healthy	TENX132	477	33,005	8,096
Test	Bowel	Cancer	TENX148	422	339,683	9,421
Test	Brain	Cancer	TENX138	480	816,574	23,927
Test	Breast	Cancer	NCBI784	313	118,691	4,652
Test	Heart	Healthy	TENX119	377	26,328	4,039
Test	Kidney	Healthy	TENX106	377	97,512	5,063
Test	Liver	Cancer	TENX120	474	162,626	7,051
Test	Lung	Healthy	NCBI879	343	23,632	2,292
Test	Lymphoid	Diseased	TENX124	377	862,385	15,572
Test	Ovary	Cancer	TENX142	480	247,454	7,468
Test	Pancreas	Cancer	TENX116	474	190,961	19,584
Test	Skin	Healthy	TENX122	377	86,381	8,050

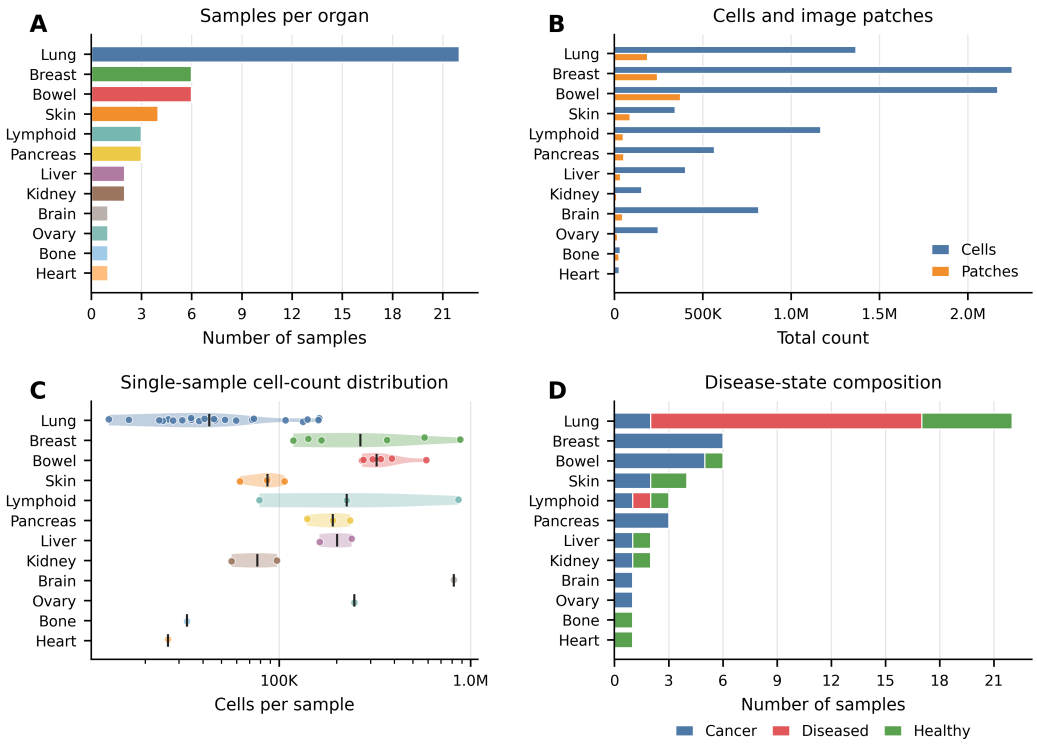


Figure 5: **Distribution of the 10X-Xenium-52.** **(A)** Number of samples per organ. **(B)** Total number of cells and image patches across organs. **(C)** Per-sample cell-count distribution on a log scale, with points denoting individual samples and vertical bars denoting organ-level medians. **(D)** Disease-state composition across organs, including cancer, diseased, and healthy samples.

C Training and Inference Parameter Details

C.1 Implementation Details of CELLO

We train our model with 4 NVIDIA H100 GPUs using PyTorch's DistributedDataParallel (DDP) framework. The per-GPU batch size is set to 128, yielding an effective global batch size of 512. Mixed-precision training (FP16) is enabled to accelerate computation and reduce memory consumption, with gradient scaling applied for numerical stability. Gradients are clipped to a maximum norm of 1.0 to prevent training instability.

The model is optimized using AdamW with a cosine annealing learning rate schedule ($\eta_{\min} = 10^{-8}$) over 200 epochs. We adopt a two-group parameter strategy to balance the pretrained backbone and the newly initialized task-specific heads: the backbone uses a base learning rate of 5×10^{-6} with a weight decay of 0.01, while the heads use a learning rate of 2×10^{-4} with a weight decay of 0.05. To preserve the pretrained representations during early training, we employ a progressive unfreezing schedule: it is fully frozen for the first 10% of epochs, gradually unfrozen with a linearly increasing learning rate (up to 2×10^{-5}) over the next 30% of epochs, and fully fine-tuned thereafter.

The overall training objective combines a cell-level gene regression loss and a patch-level gene prediction loss, weighted by 1.0 and 0.5, respectively. Validation is performed every 5 epochs using whole-slide-image (WSI) sliding-window inference with a stride equal to the patch size (224), and the model achieving the highest gene-level Pearson correlation coefficient (PCC) on the validation set is retained as the best checkpoint. Random seeds are fixed (seed $= 42$) for reproducibility.

For WSI-level evaluation, we used non-overlapping 224×224 patches with stride 224, matching the input resolution used during training. Each cell was assigned to the patch containing its spatial coordinate, and predictions were generated using only the visual tokens within that patch.

C.2 Implementation Details of DeepSpot2Cell

For DeepSpot2Cell, we follow the original design of the method and keep the pathology foundation model frozen. This choice is mainly practical: DeepSpot2Cell extracts PFM features independently for each segmented cell crop, and fine-tuning the PFM would require backpropagating through a separate large-encoder forward pass for every cell, resulting in prohibitive training cost on million-cell WSIs. In contrast, CELLO amortizes the PFM computation at the patch level and therefore can be trained end-to-end with progressive backbone unfreezing. We note that this difference may partly contribute to the performance gap, since CELLO benefits from task-specific representation adaptation. However, it also reflects an inherent scalability limitation of per-cell-crop PFM baselines: making them end-to-end trainable is substantially more expensive than patch-level amortized training.

D Additional Gene Panel Performance

In the main experiments, we report performance on most commonly used and clinically significant gene subsets: highly variable genes (denoted as HVG). Here we report another two selected subsets: most predictive genes (MPG) and spatially variable genes (SVG). As shown in Table 5, CELLO also substantially outperforms UNet3+ and DeepSpot2Cell in both ID and OOD settings. In the ID setting, CELLO improves over the strongest baseline, DeepSpot2Cell, by 40.1% on MPG PCC-50 and 39.8% on SVG PCC-50. This advantage becomes even larger in the OOD setting, where CELLO achieves a 62.9% relative improvement on MPG PCC-50 and 62.7% on SVG PCC-50, indicating stronger robustness under unseen tissue and disease conditions. In addition, the comparison with *Grid*, an ablated version of CELLO without the distance-decay cross-attention module, further validates the importance of contextual refinement.

Moreover, practical single-cell spatial transcriptomics prediction ultimately requires recovering the broader measured gene panel, including genes with weaker morphology-associated signals and higher sparsity. Therefore, we additionally evaluate all methods using the full Xenium gene panel to assess whether the performance gains of CELLO remain robust beyond curated gene subsets.

As shown in Fig 6, CELLO consistently achieves the highest gene-level PCC across both in-distribution (ID) and out-of-distribution (OOD) settings. In the ID setting, CELLO substantially outperforms DeepSpot2Cell and UNet3+ across all six tissues, with particularly large gains in Breast and Bowel. This indicates that CELLO's patch-level feature extraction and context-aware cell representation remain effective even when evaluated over the full, noisier gene panel. In the OOD setting, overall correlations are lower, reflecting the increased difficulty of generalizing to unseen tissues and disease contexts. Nevertheless, CELLO remains the highest on every organ, especially in Kidney and Liver, demonstrating stronger cross-tissue robustness. These results further support that CELLO improves not only selected high-signal genes but also broader panel-wide gene expression prediction.

E Additional Ablation Study Results

In this section, we show additional ablation results of encoder choices and cls token context on both ID and OOD setting. The encoder ablation further identified Virchow2 as the most effective pathology foundation encoder. As shown in Table 6, in the ID setting, Virchow2 achieved the best average among the three encoders on all nine metrics. Relative to the strongest other encoder H0, Virchow2 improved average MPG PCC-10/50/200 by +4.3%, +5.9%, and +6.4%, HVG PCC-10/50/200 by +3.1%, +4.2%, and +6.1%, and SVG PCC-10/50/200 by +4.4%, +6.1%, and +6.9%. The same holds in the OOD ablation, where Virchow2 achieved the best average among the three encoders on all nine metrics. As shown in Table 8, compared with H0, Virchow2 improved OOD MPG PCC-10/50/200 by +8.0%, +6.0%, and +4.6%, HVG PCC-10/50/200 by +6.1%, +5.2%, and +5.9%, and SVG PCC-10/50/200 by +8.0%, +6.0%, and +3.4%. The advantage is not uniform across tissues: UNI or H0 is higher on the Skin, Lymphoid and Ovary slides. These results suggest that Virchow2 provides more transferable morphological representations for spatial gene expression prediction.

In contrast, incorporating the CLS token did not provide consistent benefits. For the strongest encoder, Virchow2, adding the CLS token reduced the simple average over the nine averaged metrics from 0.4793 to 0.4143 in the ID setting, corresponding to a 13.6% relative decrease. A similar trend was observed in the OOD setting, where the average dropped from 0.3298 to 0.2867, a 13.1% relative decrease. Although a few isolated entries improved, such as OOD HVG PCC-200, these gains were

Table 5: **MPG and SVG results on the 10X-Xenium-52 benchmark.** ID tissues are shown on the left and OOD tissues on the right. Higher PCC values are better. The best results within each tissue are highlighted in **bold**.

In-distribution (ID)								Out-of-distribution (OOD)							
Tissue	Model	M-10	M-50	M-200	S-10	S-50	S-200	Tissue	Model	M-10	M-50	M-200	S-10	S-50	S-200
Lung	UNet3+	0.4530	0.2470	0.1322	0.4530	0.2470	0.1225	Kidney	UNet3+	0.2065	0.1356	0.0697	0.2065	0.1348	0.0479
	DeepSpot2Cell	0.4232	0.2580	0.1387	0.4232	0.2580	0.1279		DeepSpot2Cell	0.3617	0.1975	0.0871	0.3617	0.1970	0.0717
	Grid	0.5492	0.3956	0.2390	0.5492	0.3956	0.2234		Grid	0.4879	0.3431	0.1730	0.4879	0.3431	0.1483
	CELLO	**0.5538**	**0.4060**	**0.2450**	**0.5538**	**0.4060**	**0.2286**		**CELLO**	**0.6180**	**0.3934**	**0.1908**	**0.6180**	**0.3934**	**0.1633**
Breast	UNet3+	0.3126	0.2138	0.1048	0.3126	0.2138	0.0778	Heart	UNet3+	0.2166	0.1255	0.0600	0.2166	0.1255	0.0451
	DeepSpot2Cell	0.5353	0.3772	0.1793	0.5353	0.3772	0.1656		DeepSpot2Cell	0.2115	0.1154	0.0482	0.2115	0.1153	0.0382
	Grid	0.6616	0.5407	0.3229	0.6616	0.5407	0.2995		Grid	0.3490	0.2030	0.0971	0.3490	0.2029	0.0843
	CELLO	**0.6995**	**0.5681**	**0.3473**	**0.6995**	**0.5681**	**0.3217**		**CELLO**	**0.4015**	**0.2437**	**0.1144**	**0.4015**	**0.2437**	**0.1007**
Bowel	UNet3+	0.4071	0.2904	0.1739	0.4071	0.2902	0.1245	Liver	UNet3+	0.3957	0.2719	0.1362	0.3957	0.2712	0.0938
	DeepSpot2Cell	0.6592	0.5041	0.2973	0.6592	0.5041	0.2608		DeepSpot2Cell	0.4640	0.3534	0.1773	0.4640	0.3534	0.1316
	Grid	0.6911	0.5463	0.3592	0.6911	0.5463	0.3273		Grid	0.5301	0.4064	0.2292	0.5301	0.4064	0.1818
	CELLO	**0.7116**	**0.5739**	**0.3843**	**0.7116**	**0.5739**	**0.3531**		**CELLO**	**0.5540**	**0.4416**	**0.2484**	**0.5540**	**0.4416**	**0.1919**
Skin	UNet3+	0.4365	0.2639	0.1185	0.4365	0.2639	0.1121	Bone	UNet3+	0.1109	0.0705	0.0349	0.1109	0.0667	0.0134
	DeepSpot2Cell	0.5781	0.3310	0.1306	0.5781	0.3310	0.1253		DeepSpot2Cell	0.1435	0.0818	0.0314	0.1435	0.0792	0.0206
	Grid	0.6613	0.4530	0.2096	0.6613	0.4530	0.2019		Grid	0.1838	0.1112	0.0498	0.1838	0.1093	0.0360
	CELLO	**0.6888**	**0.4949**	**0.2346**	**0.6888**	**0.4949**	**0.2259**		**CELLO**	**0.2127**	**0.1204**	**0.0553**	**0.2127**	**0.1194**	**0.0403**
Pancreas	UNet3+	0.2761	0.1760	0.0903	0.2761	0.1755	0.0586	Brain	UNet3+	0.2056	0.0940	0.0404	0.2056	0.0933	0.0267
	DeepSpot2Cell	0.3190	0.2125	0.0848	0.3190	0.2086	0.0598		DeepSpot2Cell	0.2054	0.1076	0.0388	0.2054	0.1070	0.0273
	Grid	0.3388	0.2272	0.1183	0.3379	0.2164	0.0764		Grid	0.4054	0.2046	0.0837	0.4254	0.2036	0.0677
	CELLO	**0.4169**	**0.2906**	**0.1573**	**0.4169**	**0.2807**	**0.1121**		**CELLO**	**0.4920**	**0.2338**	**0.0897**	**0.4920**	**0.2274**	**0.0717**
Lymphoid	UNet3+	0.3948	0.2567	0.1181	0.3948	0.2567	0.1111	Ovary	UNet3+	0.4089	0.2358	0.1096	0.4089	0.2358	0.0952
	DeepSpot2Cell	0.3806	0.2493	0.1034	0.3806	0.2493	0.0943		DeepSpot2Cell	0.3981	0.2117	0.0847	0.3981	0.2106	0.0666
	Grid	0.5010	**0.3757**	0.1795	**0.5110**	0.3708	0.1649		Grid	0.5184	**0.3189**	**0.1490**	0.5184	**0.3172**	**0.1289**
	CELLO	**0.5083**	0.3734	**0.1814**	0.5083	**0.3715**	**0.1650**		**CELLO**	**0.5219**	0.3058	0.1394	**0.5219**	0.3040	0.1174
Average	UNet3+	0.3800	0.2413	0.1230	0.3800	0.2412	0.1011	Average	UNet3+	0.2574	0.1556	0.0751	0.2574	0.1546	0.0537
	DeepSpot2Cell	0.4826	0.3220	0.1557	0.4826	0.3214	0.1390		DeepSpot2Cell	0.2974	0.1779	0.0779	0.2974	0.1771	0.0593
	Grid	0.5672	0.4231	0.2381	0.5687	0.4205	0.2156		Grid	0.4158	0.2645	0.1303	0.4158	0.2638	0.1078
	CELLO	**0.5965**	**0.4512**	**0.2583**	**0.5965**	**0.4492**	**0.2344**		**CELLO**	**0.4667**	**0.2898**	**0.1397**	**0.4667**	**0.2882**	**0.1142**

M and S denote MPG and SVG, respectively.

not systematic and were outweighed by broad degradation across MPG and SVG metrics. This indicates that the global CLS representation contributes limited useful information for this task and may even dilute the spatially localized morphological signals required for accurate gene expression prediction.

Loss Weight and Distance-Decay Bias We trained CELLO with $\lambda \in \{0, 0.1, 0.25, 0.5, 1.0\}$ and, at $\lambda = 0.5$, without the distance-decay bias. All six runs use the same data, one GPU with 16 patches per batch and 50 epochs, so they are comparable with each other but not with the 200-epoch model of the main tables. Table 7 reports HVG PCC averaged over the ID and OOD test slides. Within this budget the settings differ by less than 0.03, $\lambda = 0.5$ scores lowest in both settings, and removing the distance-decay bias does not lower performance relative to $\lambda = 0.5$. These single-seed short runs therefore do not show a clear effect of either component.

F Discussion About Context Information

Spatial transcriptomics prediction fundamentally relies on establishing precise mappings between local tissue morphology and site-specific gene expression, requiring the model to capture fine-grained features such as cellular composition, tissue architecture, and microenvironment characteristics within each patch and its spatial neighborhood. The CLS token, as a global summary of the entire image, encodes highly abstract and compressed slide-level semantics. Such overly high-level information not only struggles to provide complementary cues for local prediction, but may instead introduce noise signals that are irrelevant to the target patch, thereby interfering with the model's ability to learn accurate morphology-to-expression correspondences. This effect is especially pronounced for more powerful encoders such as Virchow2, whose patch tokens already contain sufficiently rich local semantics; appending the CLS token effectively dilutes the informative signal within the feature space, leading to marked degradation in generalization. These findings indicate that, for tasks demanding precise local reasoning such as spatial transcriptomics prediction, directly leveraging fine-grained

Table 6: **Ablation results on the 10X-Xenium-52 ID setting evaluated by MPG, HVG, and SVG metrics.** We compare different pathology foundation models as vision encoders and investigate whether incorporating the **CLS** token to provide additional global contextual information improves performance. Higher values of PCC-10, PCC-50, and PCC-200 indicate better results. The best results are highlighted in **bold**.

Model	MPG			HVG			SVG		
	PCC-10↑	PCC-50↑	PCC-200↑	PCC-10↑	PCC-50↑	PCC-200↑	PCC-10↑	PCC-50↑	PCC-200↑
Lung									
UNI	0.5451	0.3969	0.2410	0.5451	0.3753	0.1492	0.5451	0.3969	0.2253
UNI_cls	0.5441	0.3931	0.2368	0.5441	0.3714	0.1437	0.5441	0.3931	0.2204
H0	0.5533	0.4053	0.2457	0.5533	0.3795	0.1485	0.5533	0.4053	0.2292
H0_cls	0.4892	0.3734	0.2213	0.4880	0.3427	0.1249	0.4892	0.3734	0.1960
Virchow2	**0.5538**	0.4060	0.2450	0.5537	0.3827	0.1492	**0.5538**	0.4060	0.2286
Virchow2_cls	0.5528	**0.4095**	**0.2497**	**0.5538**	**0.3832**	**0.1535**	0.5528	**0.4095**	**0.2325**
Breast									
UNI	0.6417	0.5315	0.3174	0.6342	0.4946	0.2316	0.6417	0.5315	0.2972
UNI_cls	0.6515	0.5316	0.3243	0.6439	0.5023	0.2407	0.6515	0.5316	0.3013
H0	0.6065	0.4800	0.2962	0.6065	0.4696	0.2308	0.6065	0.4800	0.2744
H0_cls	0.5986	0.4619	0.2527	0.5986	0.4581	0.1980	0.5986	0.4619	0.2207
Virchow2	0.6995	**0.5681**	**0.3473**	**0.6736**	**0.5243**	**0.2541**	0.6995	**0.5681**	**0.3217**
Virchow2_cls	**0.7008**	0.5643	0.3421	0.6717	0.5157	0.2476	**0.7008**	0.5643	0.3193
Bowel									
UNI	0.6974	0.5428	0.3612	0.6784	0.4793	0.2148	0.6974	0.5428	0.3341
UNI_cls	0.7010	0.5544	0.3680	0.6809	0.4935	0.2195	0.7010	0.5544	0.3383
H0	0.7088	0.5545	0.3669	**0.6989**	0.5113	0.2244	0.7088	0.5545	0.3334
H0_cls	0.6969	0.5295	0.3344	0.6700	0.4767	0.1820	0.6969	0.5295	0.2761
Virchow2	**0.7116**	**0.5739**	**0.3843**	0.6946	**0.5170**	**0.2340**	**0.7116**	**0.5739**	**0.3531**
Virchow2_cls	**0.7116**	**0.5739**	**0.3843**	0.6946	**0.5170**	**0.2340**	**0.7116**	**0.5739**	**0.3531**
Skin									
UNI	0.6500	0.4230	0.1974	0.6141	0.3444	0.1214	0.6500	0.4230	0.1901
UNI_cls	0.6562	0.4082	0.1828	0.6373	0.3309	0.1142	0.6562	0.4082	0.1768
H0	**0.7040**	**0.4990**	**0.2359**	**0.6864**	**0.3901**	**0.1367**	**0.7040**	**0.4990**	**0.2265**
H0_cls	0.6712	0.4326	0.1895	0.6413	0.3313	0.1092	0.6712	0.4326	0.1792
Virchow2	0.6888	0.4949	0.2346	0.6765	0.3879	0.1358	0.6888	0.4949	0.2259
Virchow2_cls	0.6888	0.4949	0.2346	0.6765	0.3879	0.1358	0.6888	0.4949	0.2259
Pancreas									
UNI	0.3372	0.2327	0.1215	0.3247	0.2248	0.0857	0.3372	0.2253	0.0797
UNI_cls	0.3382	0.2377	0.1274	0.3260	0.2296	0.0960	0.3364	0.2284	0.0868
H0	0.3510	0.2383	0.1237	0.3469	0.2252	0.0864	0.3484	0.2237	0.0797
H0_cls	0.3640	0.2544	0.1289	0.3553	0.2343	0.0828	0.3637	0.2402	0.0886
Virchow2	**0.4169**	**0.2906**	**0.1573**	**0.4101**	**0.2790**	**0.1144**	**0.4169**	**0.2807**	**0.1121**
Virchow2_cls	**0.4169**	**0.2906**	**0.1573**	**0.4101**	**0.2790**	**0.1144**	**0.4169**	**0.2807**	**0.1121**
Lymphoid									
UNI	0.4747	0.3378	0.1589	0.4570	0.2703	0.0803	0.4747	0.3356	0.1460
UNI_cls	0.4657	0.3376	0.1645	0.4469	0.2771	0.0825	0.4657	0.3333	0.1479
H0	0.5063	**0.3798**	**0.1886**	**0.4965**	**0.3206**	**0.0943**	0.5063	**0.3778**	**0.1725**
H0_cls	**0.5101**	0.3782	0.1765	0.4953	0.2960	0.0816	**0.5101**	0.3758	0.1610
Virchow2	0.5083	0.3734	0.1814	0.4848	0.3019	0.0899	0.5083	0.3715	0.1650
Virchow2_cls	0.5083	0.3734	0.1814	0.4848	0.3019	0.0899	0.5083	0.3715	0.1650
Average									
UNI	0.5577	0.4108	0.2329	0.5423	0.3648	0.1472	0.5577	0.4092	0.2121
UNI_cls	0.5595	0.4104	0.2340	0.5465	0.3675	0.1494	0.5592	0.4082	0.2119
H0	0.5717	0.4262	0.2428	0.5648	0.3827	0.1535	0.5712	0.4234	0.2193
H0_cls	0.5550	0.4050	0.2172	0.5414	0.3565	0.1298	0.5550	0.4022	0.1869
Virchow2	**0.5965**	**0.4512**	**0.2583**	**0.5822**	**0.3988**	**0.1629**	0.5965	**0.4492**	0.2344
Virchow2_cls	**0.5965**	0.4511	0.2582	0.5819	0.3975	0.1625	**0.5967**	0.4491	**0.2347**

Table 7: Loss weight λ and distance-decay bias (50-epoch runs, HVG PCC).

	ID			OOD		
Setting	10	50	200	10	50	200
$\lambda = 0$	0.5598	0.3882	0.1752	0.4314	0.2574	0.0975
$\lambda = 0.1$	0.5565	0.3866	0.1732	0.4178	0.2473	0.0933
$\lambda = 0.25$	0.5533	0.3823	0.1683	0.4147	0.2484	0.0922
$\lambda = 0.5$	0.5447	0.3750	0.1665	0.4018	0.2349	0.0850
$\lambda = 1.0$	0.5614	0.3808	0.1692	0.4298	0.2553	0.0950
$\lambda = 0.5$, no distance bias	0.5472	0.3822	0.1685	0.4232	0.2468	0.0908

Table 8: **Ablation results on the 10X-Xenium-52 OOD setting evaluated by MPG, HVG, and SVG metrics.** We compare different pathology foundation models as vision encoders and investigate whether incorporating the **CLS** token to provide additional global contextual information improves performance. Higher values of PCC-10, PCC-50, and PCC-200 indicate better results. The best results are highlighted in **bold**.

Model	MPG			HVG			SVG		
	PCC-10↑	PCC-50↑	PCC-200↑	PCC-10↑	PCC-50↑	PCC-200↑	PCC-10↑	PCC-50↑	PCC-200↑
Kidney									
UNI	0.5638	0.3600	0.1783	0.5220	0.3229	0.1269	0.5638	0.3600	0.1545
UNI_cls	0.5155	0.3419	0.1737	0.4850	0.3070	0.1249	0.5155	0.3419	0.1515
H0	0.5041	0.3605	0.1843	0.5041	0.3404	0.1368	0.5041	0.3604	0.1592
H0_cls	0.4927	0.3275	0.1628	0.4750	0.3045	0.1127	0.4927	0.3275	0.1378
Virchow2	**0.6180**	**0.3934**	**0.1908**	**0.5625**	**0.3468**	**0.1369**	**0.6180**	**0.3934**	**0.1633**
Virchow2_cls	**0.6180**	**0.3934**	**0.1908**	**0.5625**	**0.3468**	**0.1369**	**0.6180**	**0.3934**	**0.1633**
Heart									
UNI	0.3452	0.2059	0.0981	0.3431	0.1876	0.0680	0.3452	0.2059	0.0876
UNI_cls	0.3522	0.2064	0.0969	0.3480	0.1925	0.0705	0.3522	0.2053	0.0856
H0	0.3774	0.2185	0.0998	0.3740	0.2009	0.0691	0.3774	0.2185	0.0882
H0_cls	0.3837	0.2062	0.0951	0.3805	0.1955	0.0694	0.3837	0.2062	0.0812
Virchow2	**0.4015**	**0.2437**	**0.1144**	**0.4015**	**0.2318**	**0.0842**	**0.4015**	**0.2437**	**0.1007**
Virchow2_cls	**0.4015**	**0.2437**	**0.1144**	**0.4015**	**0.2318**	**0.0842**	**0.4015**	**0.2437**	**0.1007**
Liver									
UNI	0.5084	0.3952	0.2190	0.5046	0.3853	0.1531	0.5084	0.3952	0.1658
UNI_cls	0.5192	0.3933	0.2140	0.5116	0.3808	0.1435	0.5192	0.3933	0.1627
H0	0.5535	0.4169	0.2359	0.5460	0.4062	0.1684	0.5535	0.4169	0.1875
H0_cls	0.5163	0.4063	0.2129	0.5163	0.4005	0.1424	0.5163	0.4063	0.1590
Virchow2	**0.5540**	**0.4416**	**0.2484**	**0.5483**	**0.4305**	**0.1826**	**0.5540**	**0.4416**	**0.1919**
Virchow2_cls	**0.5540**	**0.4416**	**0.2484**	**0.5483**	**0.4305**	**0.1826**	**0.5540**	**0.4416**	**0.1919**
Bone									
UNI	0.1870	0.1114	0.0487	0.1679	0.0854	0.0242	0.1870	0.1108	0.0355
UNI_cls	0.1764	0.1066	0.0462	0.1553	0.0798	0.0219	0.1764	0.1058	0.0330
H0	0.1941	0.1213	0.0538	0.1705	0.0893	0.0262	0.1941	0.1187	0.0386
H0_cls	0.2036	**0.1232**	**0.0587**	0.1851	0.0937	0.0271	0.2036	**0.1213**	**0.0419**
Virchow2	**0.2127**	0.1204	0.0553	**0.1897**	**0.0960**	**0.0282**	**0.2127**	0.1194	0.0403
Virchow2_cls	**0.2127**	0.1204	0.0553	**0.1897**	**0.0960**	**0.0282**	**0.2127**	0.1194	0.0403
Brain									
UNI	0.4148	0.2012	0.0768	0.4148	0.1861	0.0558	0.4148	0.1978	0.0593
UNI_cls	0.3969	0.1755	0.0667	0.3929	0.1626	0.0491	0.3969	0.1719	0.0515
H0	0.4419	0.2113	0.0787	0.4419	0.1953	0.0580	0.4419	0.2077	0.0608
H0_cls	0.4284	0.1909	0.0707	0.4284	0.1772	0.0520	0.4284	0.1898	0.0585
Virchow2	**0.4920**	**0.2338**	**0.0897**	**0.4920**	**0.2123**	**0.0636**	**0.4920**	**0.2274**	**0.0717**
Virchow2_cls	**0.4920**	**0.2338**	**0.0897**	**0.4920**	**0.2123**	**0.0636**	**0.4920**	**0.2274**	**0.0717**
Ovary									
UNI	**0.5296**	**0.3233**	**0.1559**	**0.5295**	**0.2828**	0.0907	**0.5296**	**0.3221**	**0.1348**
UNI_cls	0.5252	0.3201	0.1521	0.5252	0.2757	0.0879	0.5252	0.3189	0.1317
H0	0.5217	0.3116	0.1489	0.5217	0.2821	**0.0924**	0.5217	0.3098	0.1280
H0_cls	0.5272	0.3115	0.1393	0.5272	0.2776	0.0820	0.5272	0.3095	0.1194
Virchow2	0.5219	0.3058	0.1394	0.5219	0.2760	0.0876	0.5219	0.3040	0.1174
Virchow2_cls	0.5219	0.3058	0.1394	0.5219	0.2760	0.0876	0.5219	0.3040	0.1175
Average									
UNI	0.4248	0.2662	0.1295	0.4136	0.2417	0.0864	0.4248	0.2653	0.1062
UNI_cls	0.4142	0.2573	0.1249	0.4030	0.2331	0.0830	0.4142	0.2562	0.1027
H0	0.4321	0.2733	0.1336	0.4264	0.2524	0.0918	0.4321	0.2720	0.1104
H0_cls	0.4253	0.2609	0.1232	0.4187	0.2415	0.0809	0.4253	0.2601	0.0996
Virchow2	**0.4667**	**0.2898**	**0.1397**	**0.4526**	**0.2656**	**0.0972**	**0.4667**	**0.2882**	**0.1142**
Virchow2_cls	**0.4667**	**0.2898**	**0.1397**	**0.4526**	**0.2656**	**0.0972**	**0.4667**	**0.2882**	**0.1142**

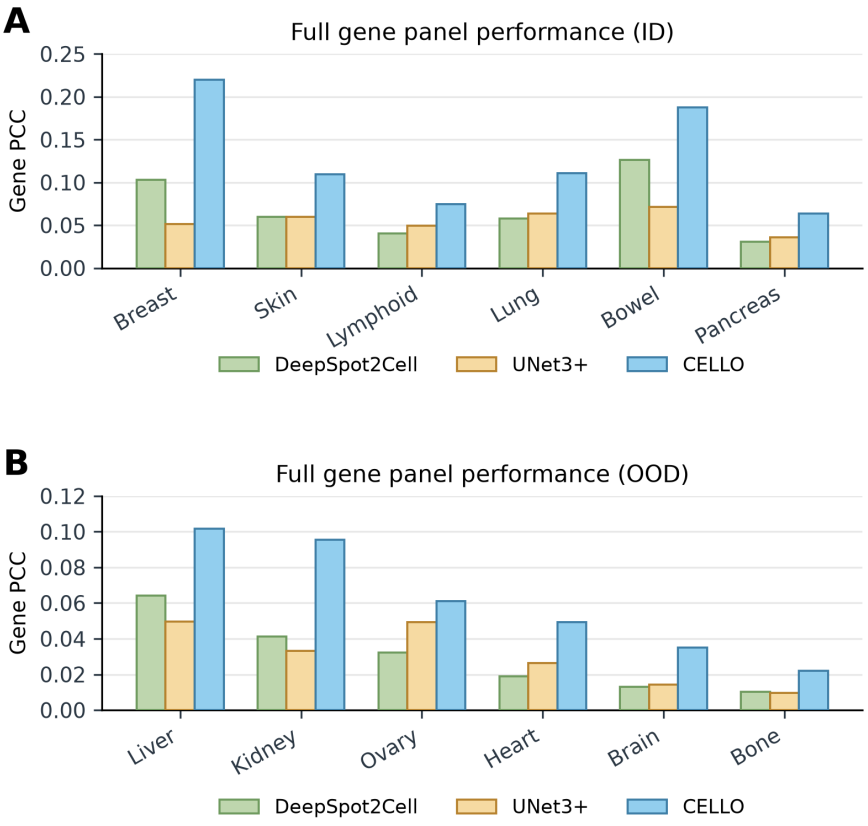


Figure 6: **Full gene panel performance comparison between DeepSpot2Cell, UNet3+ and CELLO. (A)** Full gene panel performance of ID setting. **(B)** Full gene panel performance of OOD setting.

patch-level features constitutes a more principled design choice than attempting to incorporate global information through the CLS token.

G Per-Cell Crop Baseline

To isolate the effect of querying a shared token map, we trained a baseline that uses the same Virchow2 backbone, training split and gene head, but represents each cell by the CLS embedding of a 224×224 crop centred on the cell, computed with the frozen backbone. Table 9 compares it with CELLO under the same single-cell evaluation. CELLO is higher at every k in both settings. Because CELLO also fine-tunes its backbone, the gap reflects both shared-token querying and representation adaptation.

Table 9: Per-cell crop baseline vs. CELLO (HVG PCC, single-cell evaluation).

	ID			OOD		
Method	10	50	200	10	50	200
Per-cell crop	0.4981	0.3126	0.1258	0.3184	0.1691	0.0535
CELLO	0.5822	0.3988	0.1629	0.4526	0.2656	0.0972

H External Validation

We applied CELLO without adaptation to five Xenium samples from HEST-1k that are not part of the 10X-Xenium-52 (Table 10), using the same evaluation protocol.

Table 10: External validation on five additional HEST-1k Xenium samples (PCC at k = 10 / 50 / 200).

Sample	Organ (subtype)	Cells	MPG	HVG	SVG
TENX94	Breast (ILC)	356,730	0.700 / 0.580 / 0.358	0.700 / 0.575 / 0.331	0.700 / 0.580 / 0.350
TENX190	Lung (LUAD)	278,349	0.580 / 0.436 / 0.234	0.578 / 0.421 / 0.198	0.580 / 0.436 / 0.220
TENX199	Breast (IDC)	149,995	0.454 / 0.314 / 0.125	0.454 / 0.314 / 0.125	0.454 / 0.314 / 0.125
TENX189	Lung (LUAD)	276,951	0.518 / 0.358 / 0.219	0.485 / 0.285 / 0.089	0.516 / 0.350 / 0.140
TENX158	Skin (SKCM)	111,193	0.604 / 0.401 / 0.199	0.470 / 0.196 / 0.051	0.604 / 0.390 / 0.119

I Sensitivity to Cell Localization

CELLO needs cell centroids at inference. To measure how localization errors affect it, we perturbed the query coordinates of a fixed random subset of 3,000 test patches without changing the ground truth, either by adding zero-mean Gaussian noise to each centroid or by randomly removing a fraction of cells. Table 11 reports the mean per-patch PCC, computed over all cells and measured genes of a patch. Performance degrades gradually and drops by less than 1% at a 10-pixel error.

Table 11: Sensitivity of CELLO to cell localization errors on 3,000 test patches.

Perturbation	Level	Cells	Mean PCC	Change
None	–	43,572	0.1348	–
Centroid jitter (std)	5 px	43,572	0.1346	-0.2%
	10 px	43,572	0.1336	-0.9%
	20 px	43,572	0.1294	-4.1%
	40 px	43,572	0.1206	-10.6%
	80 px	43,572	0.1114	-17.4%
Cell removal	10%	39,196	0.1336	-0.9%
	20%	34,817	0.1324	-1.8%
	40%	26,177	0.1286	-4.6%

J Computational Efficiency Comparison Details

All methods were timed on one NVIDIA L40S GPU (48 GB) used exclusively, with fp16 autocast, non-overlapping 224×224 patches and 16 images per forward pass. Each WSI was copied to local NVMe and its page cache pre-warmed, warm-up used the same tensor shapes as the timed runs, every timed stage was bracketed by `torch.cuda.synchronize()`, and each measurement was repeated five times. For each of the 12 test slides we timed 200 randomly sampled patches and scaled the result to the whole slide by its number of patches. Inference time excludes model loading and cell segmentation; segmentation is reported separately.

Per 100 patches (mean over four slides), CELLO spends 0.253 s on reading the image region, 0.327 s on the Virchow2 forward pass (81.8% of GPU time), 0.005 s on grid sampling, 0.053 s on cross-attention, 0.014 s on CLS fusion and 0.001 s on the gene head.

Projected whole-slide inference time is 67.5 ± 48.8 s for CELLO (39.3 ± 29.3 s on the GPU), 73.9 ± 52.1 s for UNet3+ and 947.2 ± 1068.3 s for DeepSpot2Cell. CELLO is $14.0\times$ faster than DeepSpot2Cell on average (median over slides $11.7\times$, range 3.5–$26.3\times$; $21.9\times$ in GPU time). The gap follows the number of backbone forward passes, 114,429 for CELLO versus 2,996,926 for DeepSpot2Cell. Against UNet3+, the times are comparable (median $1.12\times$). Cell segmentation with CellViT-SAM-H takes 382.5 ± 334.1 s per slide (74–1,067 s). Including it, CELLO remains $2.5\times$ faster than DeepSpot2Cell (median) and on par with UNet3+ ($1.02\times$).

K Limitations

Although our benchmark covers 52 paired Xenium–WSI samples across 12 organs, several OOD organs are represented by a single slide, which limits the precision with which cross-organ generalization can be estimated. In the future work, collecting more pair data covering a wider range of tissues and organs will improve the model's generalization ability. Apart from this, in actual deployment, although we avoid masking the precise shape of cells, obtaining cell locations still relies on cell segmentation. Improving the accuracy of cell segmentation can reduce the probability of bad predictions.

CELLO requires cell locations at inference, so its predictions depend on an upstream cell detector. The OOD test set contains a single slide per unseen organ, and Kidney and Liver appear in training under other health conditions, so these results are case studies rather than estimates of organ-level generalization.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have included the main contribution of this paper in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have discussed the limitations of our work in Appendix K

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide implementation details for reproducing in Appendix C

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code is available at `https://github.com/zjgao02/CELLO`, and the trained model and preprocessed data at `https://huggingface.co/gaozijun/CELLO` and `https://huggingface.co/datasets/gaozijun/cello_data`, with instructions to reproduce the evaluation.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide training, test details in Appendix C and dataset splits in Appendix B.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: We followed the standard evaluation protocol of DeepSpot2Cell, GHIST, STEM.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We described the compute resource in Appendix C

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We confirm the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work in Section 1 and 5

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite all datasets and models utilized in our experiments.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The released code, model and preprocessed data are documented in the GitHub README and the Hugging Face model and dataset cards, including their licenses.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: This research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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