Clinician-Guided Continual Learning for Plasma Cell Detection in Bone Marrow Whole Slide Images

Anonymized Authors

Anonymized Affiliations
email@anonymized.com

Abstract. Accurate plasma-cell quantification in bone marrow smears (BMS) is central to diagnosis of plasma-cell disorders such as multiple myeloma. However, deployed detection and classification models must remain reliable with new patient smears. We present a clinician-guided continual learning (CL) study for plasma-cell detection in BMS whole slide images. Using 11 whole slide images containing 47,895 annotated plasma cells, we train and update two detector families, YOLO11n and RT-DETR-L, and integrate them into a QuPath extension that supports hematologist review that drive model redeployment. We compare six CL strategies: static deployment, naive fine-tuning, bounded replay, cumulative training, elastic weight consolidation (EWC), and synaptic intelligence (SI). Performance is assessed under three complementary regimes: recall on non-exhaustively annotated held-out WSIs, precision-recall evaluation on exhaustively reviewed regions of interest on held-out WSIs, and external evaluation on an external dataset. Naive updating achieved high precision but reduced recall. Bounded replay provided the strongest final exhaustive-ROI performance for RT-DETR-L, improving F1 from 0.689 to 0.741 and AP50 from 0.752 to 0.809 over the frozen baseline, while YOLO11n showed limited improvement over its strong static baseline. Regularization-based methods, EWC and SI, were inconsistent and sometimes degraded performance. These results suggest that safe CL in computational pathology requires not only model updates, but also bounded data retention, expert auditing, and evaluation protocols capable of detecting clinically relevant forgetting.

Keywords: Continual Learning · Clinician-in-the-loop · Plasma cell detection

1 Introduction

Clinical assessment of plasma-cell disorders relies on bone marrow smear (BMS) examination, where hematologists identify and count plasma cells for diagnosis [12, 1]. This process is laborious and subject to inter-observer variation. Although object detectors could support more consistent and scalable assessment, their clinical usefulness depends not only on initial accuracy but also on maintaining

performance as new patient whole slide images (WSIs) become available after deployment.

Continual learning (CL) addresses settings in which data arrive sequentially and models must acquire new information while limiting forgetting of previously acquired knowledge [10, 4]. Recent work has highlighted CL as a practical requirement for medical-imaging systems exposed to evolving data streams, distribution shift, scanner variation, and post-deployment updates [11]. In computational pathology, CL has begun to be studied for WSI-level multiple-instance learning and adaptive pathology classification [8, 7, 9]. However, these studies primarily address slide-level classification, weakly supervised localization, or general pathology classification, while cell-level detection and classification remain comparatively underexplored. This distinction is important for plasma-cell quantification, where forgetting may appear not only as an incorrect slide-level prediction but as missed cells during disease-burden estimation.

In this work, we therefore study post-deployment CL for plasma-cell detection and classification under a clinician-in-the-loop workflow using YOLO11n [5] and RT-DETR-L [14]. We compare six CL strategies with different assumptions about historical-data access, including static deployment, naive fine-tuning, bounded replay, cumulative training, and parameter regularization (elastic weight consolidation (EWC) [6] and synaptic intelligence (SI) [13]), and evaluate them on recall against non-exhaustively annotated held-out WSI set, precision and recall on exhaustively annotated regions of interest (ROIs) on held-out WSI set, and generalization on the external dataset, PCMMD [2]. We additionally implement a QuPath [3] extension that supports tiled model inference, confidence-ranked expert review, correction export, and subsequent model redeployment.

The contributions of this work can be summarized as follows:

- We formulate plasma-cell detection in BMS as a post-deployment CL problem, where models are sequentially updated from newly reviewed patient WSIs rather than treated as static detectors/classifiers.
- We introduce a QuPath-integrated clinician-in-the-loop workflow for plasma-cell detection, supporting tiled WSI inference, confidence-ranked hematologist review, correction export, and enabling subsequent model redeployment.
- We provide a controlled benchmark of six CL strategies across two detector families, YOLO11n and RT-DETR-L, showing that update performance depends on both historical-data access and model architecture, with bounded replay offering a practical alternative when full cumulative retraining is impractical.

2 Materials and Methods

2.1 Dataset

A curated dataset from a hospital partner (anonymized), consisting of 11 WSIs and GeoJSON files containing 47,895 plasma-cell annotations was used for experimentation. As this dataset is not yet published pending required approval

procedures, we additionally evaluate our models on a public dataset [2]. Each BMS is tiled on a deterministic 640×640 px grid at stride 512 ($\sim$20% overlap) (Figure 1). Tiles containing at least one annotation centroid were considered annotation-positive. Annotation boxes crossing tile boundaries were clipped and retained when at least 50% of their original area remained, then converted to YOLO format. Conservative background tiles were sampled at a ratio of 0.5 per positive tile from regions at least 640 px from any known annotation. Because the WSI annotations were not assumed exhaustive, these represent background relative to known labels rather than confirmed plasma-cell-free regions.

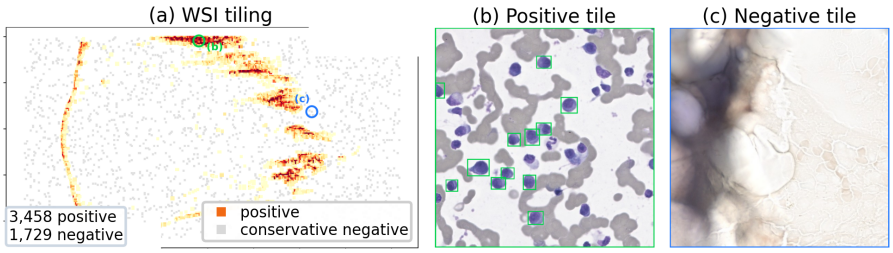

Fig. 1: WSI tiling example. (a) Distribution of positive and conservatively sampled negative tiles. (b) Positive tile with plasma-cell annotations. (c) Negative background tile.

2.2 Detector training and evaluation

The experimental setup is summarized in Figure 2. We evaluated two object-detection architectures, YOLO11n and RT-DETR-L. All detectors were initialized from COCO-pretrained weights. Initial training was performed for up to 150 epochs, followed by up to 75 epochs at each sequential update, with early-stopping patience of 25 epochs for YOLO11n and 35 epochs for RT-DETR-L on a fixed validation slide. Training used Ultralytics automatic optimizer and learning-rate selection, a nominal weight decay of 5×10^{-4}, 640×640-pixel tiles, and batch sizes of 32 and 8 for YOLO11n and RT-DETR-L, respectively. Replay retained 512 labelled tiles from each previously observed slide, while SI used $\lambda = 1000$. Experiments used deterministic training with random seed 0 and were repeated over two predefined sequential case streams. Statistical uncertainty was assessed by WSI-level bootstrap resampling. Five targeted strategy contrasts were evaluated using Wilcoxon signed-rank tests on stream-averaged per-WSI F1, with Holm correction for multiple comparisons. Supporting training, evaluation, and deployment code is available[1].

[1] anonymized repository: https://anonymous.4open.science/r/reviewartifact7F3C9AD2/

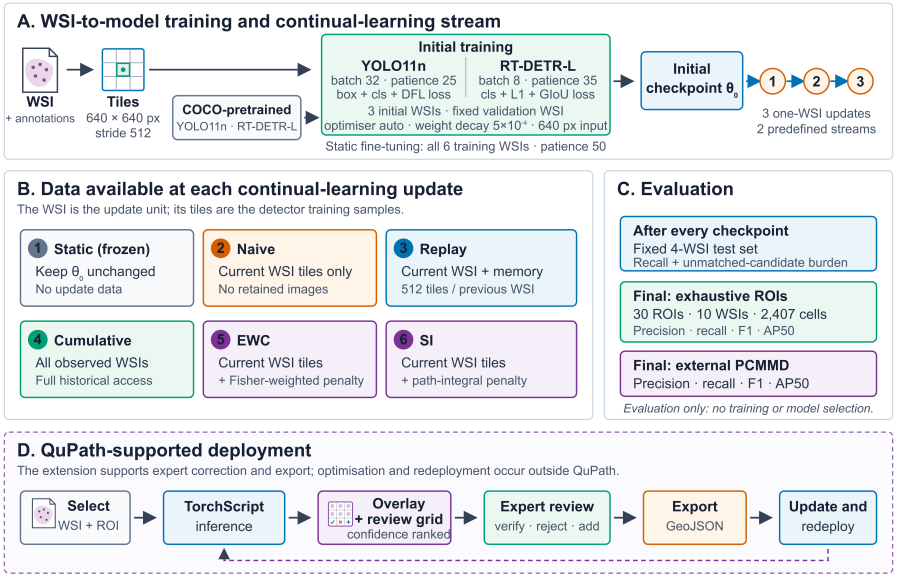

Fig. 2: Flowchart illustrating model training and CL of two detectors using six strategies evaluated on held-out WSI, exhaustive-ROI and external dataset, then deployed through QuPath-supported expert-review loop.

Data-access strategies. We evaluated four strategies with different assumptions about access to historical training data.

Static. The initial three-WSI checkpoint was used unchanged at every update step. No retraining was performed after deployment. This frozen model provides a lower-bound reference for performance without adaptation.

Naive. At each update step, the model was fine-tuned exclusively on the incoming slide. No previous slides or replay tiles were retained. This setting requires no long-term storage of historical patient images but is susceptible to catastrophic forgetting.

Replay. At each update step, the model was trained on the current incoming slide together with a bounded replay buffer containing 512 labelled tiles from each previously seen training slide. Replay tiles were selected deterministically using a fixed random seed. Thus, at update step k, the training set consisted of the current slide and replay buffers from slides $1, \ldots, k-1$. This strategy requires bounded storage that grows linearly with the number of previous slides.

Cumulative. At each update step, the model was fine-tuned on all slides observed so far. This setting assumes unrestricted access to all previous training slides and provides an upper-reference condition among the sequential update strategies.

Regularization-based strategies. These methods use only the current incoming slide during each update step and do not store previous slide images or annotations. However, they retain method-specific state, such as previous model parameters, parameter-importance estimates, or the previous model checkpoint.

Elastic Weight Consolidation penalizes changes to parameters estimated to be important for previous training steps. We approximated parameter importance using the empirical diagonal Fisher, computed from squared task-loss gradients over valid optimisation batches from the previous training phase:

$$F_i = \frac{1}{N_{\text{valid}}} \sum_t g_{i,t}^2, \tag{1}$$

where $g_{i,t}$ is the gradient of parameter θ_i at optimisation step t. The EWC objective was:

$$L_{\text{EWC}} = L_{\text{det}} + \frac{\lambda}{2} \sum_i F_i \left(\theta_i - \theta_i^*\right)^2, \tag{2}$$

where θ_i^* denotes the parameter value saved after the previous training phase. We used $\lambda = 1000$ in both EWC and SI.

Synaptic Intelligence estimates parameter importance online by accumulating each parameter's contribution along the optimisation trajectory. For update step k, the path-integral importance was estimated as:

$$\omega_i^{(k)} = \text{ReLU}\left(\frac{\sum_t g_{i,t} \Delta\theta_{i,t}}{\sum_t \Delta\theta_{i,t}^2 + \varepsilon}\right), \tag{3}$$

with $\varepsilon = 10^{-3}$. Importance estimates were accumulated across completed update steps and used to penalise deviations from the previous reference parameters:

$$L_{\text{SI}} = L_{\text{det}} + \frac{\lambda}{2} \sum_i \omega_i^{\text{cumulative}} \left(\theta_i - \theta_i^*\right)^2. \tag{4}$$

QuPath Integration. To support model-assisted review, we implemented a Java extension for QuPath. The extension loads an exported TorchScript detector together with a sidecar manifest specifying the model family, output encoding, input size, confidence threshold, and non-maximum-suppression threshold. Within each user-selected annotation, the WSI is partitioned into model-sized patches 640 px and processed through the Deep Java Library PyTorch engine, using a GPU when available and otherwise falling back to the CPU. Tile-local detections are transformed to slide coordinates, deduplicated using cross-tile non-maximum suppression, and added to the QuPath hierarchy as rectangular annotations carrying confidence and model metadata. Confidence tiers are encoded by annotation colour to support rapid visual triage.

For candidate review, detections are sorted in ascending confidence and presented in pages of up to 25 image patches. Each page is anchored by the lowest-confidence unreviewed candidate and its spatially nearest neighbours, prioritizing uncertain predictions while preserving local context. Hematologists

mark candidates as *verified plasma cell* or *not plasma cell* while missed cells added manually through the standard QuPath interface can be retained as verified annotations. Reviewed objects are exported as a tab-separated table with slide coordinates, prediction confidence, model identity, review label, and session metadata, accompanied by a GeoJSON manifest summarizing the review. These exports provide the input to the downstream dataset-building and retraining pipeline; model optimization and redeployment are performed outside QuPath.

3 Results

Across the fixed held-out WSI set, zero-shot recall was 0.0% for YOLO11n and 0.7% for RT-DETR-L, increasing to 85.4% and 90.2% after fine-tuning. Additionally, RT-DETR-L recovered more available annotations but also produced substantially more unmatched candidates than YOLO11n (5,940 versus 1,955) as shown in Table. 1.

Table 1: Zero-shot versus fine-tuned performance on held-out WSIs.

Model	**Recall**		**Unmatched candidates**	
	Zero-shot	**Fine-tuned**	**Zero-shot**	**Fine-tuned**
YOLO11n	0.0%	85.4%	293	1,955
RT-DETR-L	0.7%	90.2%	10,968	5,940

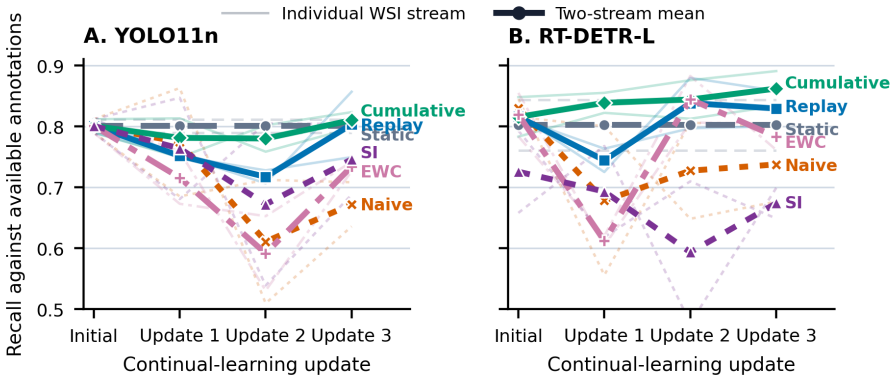

Fig. 3: Recall trajectories during CL on the WSI test set after initial training and each of three WSI updates.

Recall on the fixed held-out WSI set varied substantially across update strategies and WSI streams as shown in Fig. 3. Replay and cumulative training generally maintained or recovered sensitivity, whereas naive, EWC, and SI exhibited

Table 2: Final-step exhaustive-ROI performance. Values are means over two predefined update streams. F1 confidence intervals were obtained using 5,000 cluster-bootstrap replicates over the 10 WSIs.

Model	**Strategy**	**Prec.**	**Recall**	**F1**	**95% CI**	**AP$_{50}$**
YOLO11n	Static	0.815	**0.658**	**0.728**	[0.637, 0.767]	0.798
	Naive	**0.870**	0.555	0.677	[0.625, 0.736]	0.782
	Replay	0.785	0.648	0.708	[0.630, 0.752]	0.779
	Cumulative	0.820	0.651	0.726	[0.653, 0.761]	**0.807**
	EWC	0.846	0.548	0.665	[0.594, 0.745]	0.766
	SI	0.862	0.612	0.716	[0.659, 0.759]	0.802
RT-DETR-L	Static	0.721	0.669	0.689	[0.595, 0.753]	0.752
	Naive	**0.797**	0.640	0.709	[0.630, 0.742]	0.778
	Replay	0.764	**0.720**	**0.741**	[0.642, 0.786]	**0.809**
	Cumulative	0.731	0.716	0.720	[0.624, 0.758]	0.748
	EWC	0.714	0.690	0.698	[0.587, 0.743]	0.780
	SI	0.777	0.516	0.613	[0.550, 0.676]	0.715

larger strategy- and architecture-dependent declines. Because these WSI annotations are non-exhaustive, the trajectories measure recall against available labels but not precision.

Table 3: External dataset performance after the final CL step. AP$_{50}$ use the pooled-fold split (512 images; 1,615 boxes), whereas Pat. AP$_{50}$ uses the patient split (344 images; 459 boxes).

Model	**Strategy**	**Prec.**	**Recall**	**F1**	**AP50**	**Pat. AP50**
YOLO11n	Static	0.640	0.672	0.642	0.709	0.881
	Naive	0.652	0.672	0.651	0.702	0.790
	Replay	0.760	0.708	0.731	0.770	**0.935**
	Cumulative	**0.795**	**0.739**	**0.766**	**0.818**	0.928
	EWC	0.752	0.633	0.687	0.733	0.817
	SI	0.775	0.581	0.664	0.719	0.928
RT-DETR-L	Static	0.596	0.509	0.547	0.584	0.758
	Naive	0.468	0.484	0.475	0.431	0.819
	Replay	0.568	**0.609**	0.588	0.569	**0.826**
	Cumulative	**0.779**	0.607	**0.681**	**0.659**	0.787
	EWC	0.490	0.500	0.494	0.479	0.810
	SI	0.401	0.348	0.371	0.309	0.688
Published on PCMMD paper under original protocol [2]						
YOLOv8	5-fold cross-validation	–	–	0.780	–	–
	Full-data tuned	–	–	0.840	0.935	–

On the exhaustive-ROI held-out WSI, replay achieved the highest RT-DETR-L F1 and AP$_{50}$, whereas YOLO11n showed little improvement over its strong static baseline. Paired analysis supported RT-DETR-L replay over EWC (mean ΔF1 = 0.046, 95% CI 0.024–0.072; Holm-adjusted $p = 0.029$), but evidence against the static model was weaker after correction (ΔF1 = 0.044; $p = 0.078$). YOLO11n cumulative and static performance did not differ (ΔF1 = 0.010; $p =$

0.322). Naive achieved high precision but lower recall, indicating under-detection rather than improved specificity (Table 2).

On PCMMD, cumulative training achieved the strongest pooled performance for both architectures, reaching F1/AP_{50} values of 0.766/0.818 for YOLO11n and 0.681/0.659 for RT-DETR-L (Table 3). Replay achieved the highest patient-level AP_{50} for YOLO11n (0.935) and RT-DETR-L (0.826).

4 Discussion

The effect of CL depended on both architecture and access to historical data. Exhaustive ROI review materially changed the interpretation of model behaviour: high precision under naive updating was accompanied by reduced recall, a distinction that could not be established from non-exhaustive WSI annotations alone.

No single CL strategy dominated all evaluation regimes. Replay provided the strongest final exhaustive-ROI performance for RT-DETR-L and the highest patient-level PCMMD AP_{50} for both architectures, suggesting that it may provide a reduced-retention alternative when complete historical access is unavailable. YOLO11n showed limited improvement over its strong frozen baseline. EWC and SI avoided retaining previous images but were less consistent, illustrating a past-data-access versus performance trade-off.

External PCMMD evaluation supported retaining prior data: replay performed best on the patient-level split, while cumulative training led on the pooled split. Despite receiving no PCMMD training images, the strongest YOLO11n achieved a pooled F1 of 0.766 and patient-split AP50 of 0.935, compared with the PCMMD authors' in-domain results of 0.780 and 0.935, respectively. These descriptive comparisons suggest promising cross-domain transfer, but not parity, as the original benchmark used an in-domain, two-class YOLOv8 detector whereas our evaluation used a transferred single-class plasma-cell detector.

The QuPath extension provides a practical interface for inference, expert review, and correction export, but prospective CL driven by routine clinical use remains to be evaluated. A general limitation is the small institutional cohort used. Future work should include larger multi-institutional cohorts.

5 Conclusion

CL in computational pathology is not solely a problem of updating model parameters; it also requires decisions about what historical information may be retained, how forgetting is detected, and when expert review should intervene. Our results identify bounded replay combined with periodic exhaustive auditing as a promising approach to this broader problem, while showing that its benefit remains architecture-dependent. Establishing safe CL in pathology will ultimately require adaptive models to be evaluated as part of governed clinician-in-the-loop systems rather than as isolated algorithms.

References

1. Abdulrahman, A.A., et al.: Is a 500-cell count necessary for bone marrow differentials? American Journal of Clinical Pathology **150**(1), 84–91 (2018)
2. Andrade, C.L.B., Ferreira, M.V., Alencar, B.M., Filho, J.L.S.B., Guimaraes, M.A., Moraes, I.P.C., Lopes, T.J.S., dos Santos, A.S., dos Santos, M.M., Silva, M.I.C.S.e., et al.: PCMMD: A novel dataset of plasma cells to support the diagnosis of multiple myeloma. Scientific Data **12**(1), 161 (2025). https://doi.org/10.1038/s41597-025-04459-1
3. Bankhead, P., Loughrey, M.B., Fernández, J.A., Dombrowski, Y., McArt, D.G., Dunne, P.D., McQuaid, S., Gray, R.T., Murray, L.J., Coleman, H.G., James, J.A., Salto-Tellez, M., Hamilton, P.W.: QuPath: Open source software for digital pathology image analysis. Scientific Reports **7**(1), 16878 (2017). https://doi.org/10.1038/s41598-017-17204-5
4. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3366–3385 (2021). https://doi.org/10.1109/TPAMI.2021.3057446
5. Jocher, G., Qiu, J.: Ultralytics YOLO11. https://github.com/ultralytics/ultralytics (2024), version 11.0.0
6. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017). https://doi.org/10.1073/pnas.1611835114
7. Lee, B.H., Jeong, W., Han, W., Lee, K., Chun, S.Y.: Continual multiple instance learning with enhanced localization for histopathological whole slide image analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025). https://doi.org/10.1109/ICCV51701.2025.02157
8. Li, X., Cui, Y., Li, J., Chan, A.B.: Advancing multiple instance learning with continual learning for whole slide imaging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20800–20809 (2025)
9. Nguyen, A.T., Byeon, K., Kim, K., Song, B., Chae, S.W., Kwak, J.T.: Camp: Continuous and adaptive learning model in pathology. npj Artificial Intelligence (2026). https://doi.org/10.1038/s44387-025-00051-5
10. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019). https://doi.org/10.1016/j.neunet.2019.01.012
11. Qazi, M.A., Hashmi, A.U.R., Sanjeev, S., Almakky, I., Saeed, N., Gonzalez, C., Yaqub, M.: Continual learning in medical imaging: A survey and practical analysis. ACM Computing Surveys **58**(8), 197:1–197:25 (2026). https://doi.org/10.1145/3785663
12. Rajkumar, S.V.: Multiple myeloma: 2024 update on diagnosis, risk-stratification, and management. American Journal of Hematology (2024). https://doi.org/10.1002/ajh.27422
13. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3987–3995. PMLR (2017)
14. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: DETRs beat YOLOs on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16965–16974 (2024)