Squeezing performance from pathology foundation models with chained hyperparameter searches

Joseph Cappadona[1], **Ken Gary Zeng**[1], **Carlos Fernandez-Granda**[2],
Jan Witowski[1], **Yann LeCun**[2,3], **Krzysztof J. Geras**[1,2]
[1]Ataraxis AI
[2]New York University
[3]Meta AI
k.j.geras@nyu.edu

Abstract

Self-supervised learning (SSL) is perfectly suited for applications in digital pathology due to the scarcity of labeled data. Over the past years, many academic and industrial labs have published pathology foundation models, claiming 'state-of-the-art' performance due to improvements in architecture, methodology, and/or training data. In this paper, we demonstrate that simply tuning the hyperparameters of popular SSL method DINOv2, using a relatively small dataset, leads to similar or superior performance. Specifically, we conduct three successive hyperparameter searches, iteratively increasing either dataset or model size while narrowing the hyperparameter search space and carrying over promising hyperparameters. Overall, this preliminary study demonstrates the importance of hyperparameter tuning in this domain and proposes straight-forward strategies to improve foundation models with additional compute and data.

1 Introduction

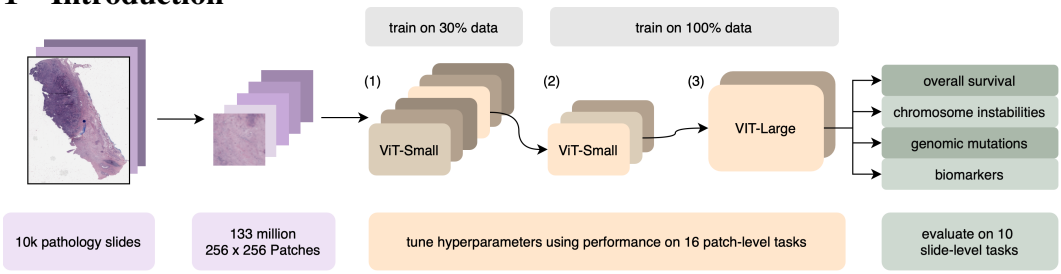

Figure 1: Successive hyperparameter searches are used to tune DINOv2 hyperparameters by optimizing performance 16 patch-level tasks. (1) ViT-Small (ViT-S) is tuned on 30% of the training dataset. (2) Hyperparameter ranges are narrowed and the 5 best samples from (1) are used as initial samples for tuning ViT-S using 100% of the dataset. (3) Hyperparameter ranges are again narrowed and the 5 best samples from (2) are used as initial samples for tuning ViT-Large (ViT-L) using 100% of the data. The best models from each search are then evaluated on a suite of 10 slide-level tasks.

Self-supervised learning (SSL) is perfectly suited for applications in digital pathology due to the scarcity of labeled data. Although millions of tissue slides are scanned every year, collecting labels for most tasks of clinical interest requires multi-year follow-up or annotation by experts, so large labeled datasets are rare. For example, the largest cohort of patients with shared characteristics in The Cancer Genome Atlas (TCGA) [1] contains just 1133 slides, with less than 400 slides on average across the 32 cohorts. And the largest cohort in the Clinical Proteomic Tumor Analysis Consortium (CPTAC) [2] contains only 1137 slides, with less than 600 slides on average across the 11 cohorts. To account for the prohibitively small amount of labeled data, the following paradigm has become commonplace. First, train a foundation model via SSL on a large number of unlabeled pathology slides. Then, use representations extracted from this foundation model to train a classification/regression model for a downstream task of interest, such as disease diagnosis or prognosis [3, 4, 5, 6].

Preprint. Under review.

Over the past years, several academic and industrial labs have published foundation models for digital pathology, claiming superiority due to improvements in architecture [7, 8], methodology [6, 9, 10, 11, 12], and/or training data [13, 14, 15]. In this study we consider an alternative route: carefully tuning the hyperparameters of a baseline SSL model. Tuning the hyperparameters of SSL models is challenging. First, the lack of a meaningful validation loss makes it difficult to predict downstream performance. This problem can be alleviated by constructing auxiliary validation tasks and using them to evaluate models throughout training. Second, training modern foundation models is slow due to the large size of the models and the training sets. Training a single model can take hundreds or thousands of GPU-hours. Therefore, researchers typically focus on scaling model size and/or dataset size/diversity, subsequently training models with fixed hyperparameters, either using the values reported by the SSL method authors or some lightly modified version thereof.

In this paper we demonstrate that carefully tuning the hyperparameters of DINOv2 on a suite of 16 patch-level tasks leads to matching or even surpassing the slide-level task performance of some digital pathology foundation models considered 'state-of-the-art', even when using a training set of modest size. Specifically, we first tune a ViT-S on 30% of our training dataset. We then take the best hyperparameter settings from this search and use them as the initial samples for a hyperparameter search tuning a ViT-S on 100% of the training dataset. Finally, we use the best hyperparameter settings from this search and use them as the initial samples for a hyperparameter search tuning a ViT-L on 100% of the training dataset. In each successive search, we narrow the hyperparameter ranges, helping the hyperparameter optimizer hone in on the most promising regions of the hyperparameter space. We present an overview of this procedure in Figure 1. We conclude by showing a strong correlation between patch-level and slide-level performance, validating our approach.

2 Experimental setup

Model We train two sizes of Vision Transformer (ViT) [16], ViT-Small (ViT-S) (21M parameters) and ViT-Large (ViT-L) (303M parameters), using DINOv2 with iBOT disabled[1]. We tune DINOv2 hyperparameters related to the learning rate schedule, weight decay schedule, teacher temperature and momentum schedules, and augmentation policy. ViT architectural specifications are in Appendix A.3, and the full list of the tuned hyperparameters, along with their ranges, is in Appendix A.4.

Training data All foundation models we train in this study are trained on 10,073 formalin-fixed paraffin-embedded hematoxylin-and-eosin-stained whole slide images (WSIs) from TCGA [17]. We use Otsu's method [18] to distinguish background from foreground and subsequently patchify the foreground into non-overlapping 256×256 patches at $20\times$ magnification, yielding 133M patches (approximately 13,000 patches per slide on average), representing tissue from 31 cancer types spanning 27 organs. A breakdown of slides and patches per cancer type is provided in Appendix A.5. Models are trained for 1 epoch (i.e., each patch is seen exactly once) with a batch size of 4,480.

Patch-level and slide-level tasks Patch-level tasks, constructed by patchifying WSIs and then annotating at the patch level, are designed to benchmark a model's ability to learn micro-scale properties of cancerous tissue (such as tumor cellularity, cancer grade, or histology) and typically contain a few thousand to a few hundred-thousand patches. Slide-level tasks, on the other hand, are constructed by annotating full WSIs, and are designed to benchmark a model's ability to learn macro-scale properties of cancerous tissue (such as major histological subtypes) or properties of the associated patient (such as genomic characteristics or risk of cancer recurrence). Because slide-level tasks are the most clinically actionable, we treat them as the downstream tasks of interest. However, instead of tuning directly on slide-level tasks, which would bias the model unfairly in their favor, we tune SSL hyperparameters by evaluating on 16 patch-level tasks. After hyperparameter tuning is completed, we evaluate the best models on 10 slide-level tasks and compare to public pathology foundation models (see Section 4.2 and Figure 2). Subsequently, we show a strong correlation between patch-level and slide-level performance (see Section 4.3 and Figure 4). Overviews of patch- and slide-level tasks are provided in Appendixes A.1 and A.2, respectively.

3 Hyperparameter tuning strategy

To account for variation in the difficulty of different tasks and the scale of their corresponding evaluation metrics, we tune hyperparameters with respect to a weighted summary score (`WSS`) computed based on performance relative to the performances of a set of baseline models. That is,

[1]We found in our preliminary experiments that iBOT leads only to a marginal improvement in performance while making learning less stable.

$$\texttt{WSS}_m = \frac{1}{|T|} \sum_{t \in T} \log n_t \cdot \frac{\mu_m^t - \mu_M^t}{\sigma_M^t},$$

where m is the model being evaluated, T is the set of tasks, M is a set of baseline models, n_t is the number of patches in task $t \in T$, μ_m^t is the average performance of model m on task t across all seeds and cross-validation splits, and μ_M^t and σ_M^t are the average and standard deviation, respectively, of performance across M for task t. μ_M^t and σ_M^t were computed by first collecting several 'baseline' models consisting of a variety of pathology and non-pathology foundation models. Then, we evaluated all baseline models across all patch-level tasks and computed the mean and the standard deviation of their performances. We compute a model's `WSS` every 10% through training and early stop after two consecutive epochs where a new best `WSS` has not been attained.

3.1 Chaining hyperparameter searches

At a high level, we perform hyperparameter tuning via the following protocol:

0. Design initial hyperparameter search space based on DINOv2 defaults.
1. Run Bayesian hyperparameter search using `MODEL_SIZE` and `DATA_FRACTION` for some number of samples.
2. Look at the best hyperparameter settings (based on `WSS`) and narrow the hyperparameter search space.
3. If possible, increase either `MODEL_SIZE` or `DATA_FRACTION` and return to Step 1, using the top hyperparameter settings from the previous search as the initial samples for the next search. Otherwise, continue to Step 4.
4. Evaluate the top models from the final search on the slide-level evaluation suite.

3.2 Downstream models

For patch-level tasks, in order to evaluate the learned representation rather than the downstream model, we use simple models: k-NN ($k = 20$) for classification, linear regression for regression, and softmax regression for multi-class classification with probability outputs. We use 5-fold cross-validation and repeat it 10 times with randomly drawn cross-validation splits and, where applicable, model initializations and average the results.

	TP53	PIK3CA	CDH1	MSI	ER	HER2	ILC	IDC	HRD	OS	avg
Prov-GigaPath	0.685	0.565	0.723	0.682	0.759	0.596	0.780	0.761	0.620	0.592	0.676
Kaiko-L/14	0.651	0.566	0.673	0.651	0.734	0.569	0.807	0.784	0.608	0.594	0.664
ViT-L Ensemble	0.614	0.549	0.650	0.624	0.672	0.575	0.753	0.747	0.577	0.600	0.636
ViT-L #2	0.639	0.559	0.639	0.618	0.656	0.577	0.745	0.735	0.571	0.598	0.634
ViT-L #1	0.612	0.560	0.637	0.618	0.668	0.562	0.739	0.735	0.571	0.594	0.630
ViT-S #1	0.608	0.552	0.628	0.621	0.659	0.580	0.736	0.730	0.561	0.611	0.629
Hibou-L	0.616	0.545	0.669	0.598	0.680	0.549	0.747	0.715	0.565	0.593	0.628
Virchow	0.616	0.521	0.627	0.634	0.685	0.570	0.719	0.707	0.572	0.602	0.625
ViT-L Default	0.608	0.554	0.622	0.648	0.650	0.565	0.741	0.722	0.547	0.595	0.625
ViT-S 30% #1	0.581	0.540	0.618	0.596	0.634	0.568	0.712	0.713	0.547	0.593	0.610

Figure 2: Performances broken down by model (y-axis) and slide-level task (x-axis). For overall survival (OS) the metric reported is the C-index [19], and for all other tasks the metric is AUC. For tasks with multiple datasets, performances across datasets are averaged.

Since pathology foundation models extract representations at the patch level, some aggregation is needed to turn the thousands of patch-level representations into a single slide-level representation. Learnable pooling, such as a AttMIL [20], could be used, but this would complicate the evaluation protocol by introducing more hyperparameters and the need for gradient descent-based optimization. Instead, we opt for parameterless mean-pooling following Wolfein et al. [21] and Chen et al. [22]. Accordingly, we fit the following models on top of PCA-reduced (number of components $C = 50$) mean-pooled patch embeddings: k-NN with $k = 5$ for classification and regression, and Cox Proportional Hazards model (with regularization coefficient $\alpha = 0.01$) for time-to-event prediction. PCA is critical due to the large dimensionality of SSL representations and relatively small size of slide-level datasets. For each task, we use 3-fold cross-validation and repeat it 10 times with randomly drawn cross-validation splits and average the results. For all patch- and slide-level tasks, neither k nor α nor any other hyperparameters were tuned at any point beyond ensuring in preliminary experiments that the chosen values enabled learning.

4 Results

4.1 Hyperparameter searches

We started with a hyperparameter search training a ViT-S on 30% of the full training dataset. We drew 58 samples in this search. The top 5 models, by peak `WSS` at any point during training, had an average `WSS` of -7.1. We then narrowed the hyperparameter search space to exclude parts of the

space not represented among the best samples and used the top 5 samples as the initial samples for the next hyperparameter search: ViT-S trained on 100% of the training dataset. Adjustments were made to carried-over hyperparameters as needed to cast them into the new search space, and warm-up schedules were scaled so their absolute lengths remained unchanged.

For the ViT-S 100% search, we drew 61 hyperparameter samples in addition to the initial 5 samples. The top 5 samples in this search had an average peak WSS of 0.1. The top samples from the ViT-S 30% search perform well in the ViT-S 100% search. Notably, the second best sample from the 30% search achieved the third highest WSS in the 100% search. We again narrowed the hyperparameter search space and used the top 5 samples as the initial samples for the final hyperparameter search: ViT-L trained on 100% of the training dataset. For this final search, we sampled 7 sets of hyperparameters in addition to the initial 5 samples. The top 5 samples yielded an average peak WSS of 1.4, outperforming a baseline ViT-L trained with the default DINOv2 hyperparameters, which achieved a WSS of -0.8.

4.2 Comparison to strong publicly available models

We take the best two models from the ViT-L 100% search (ViT-L #1 and ViT-L #2), and evaluate them on the slide-level task suite. We compare against four models: Virchow [13], a ViT-H (631M parameters); Hibou-L [14], a ViT-L (303M parameters); Kaiko-L/14 [9], a ViT-L; and Prov-GigaPath's tile encoder [8], a ViT-G (1.1B parameters). We find that our models outperform both Virchow and Hibou-L, considered 'state-of-the-art' at the time of their publication. Full slide-level task results are in Figure 2. Furthermore, we can trivially improve downstream performance by taking the trained downstream models for ViT-L #1 and #2 and simply averaging their predictions. However, Kaiko-L/14 and Prov-GigaPath still outperform our models by a significant margin, demonstrating that increases in training data yields significantly stronger models when used correctly. As visualized in Figure 3, ViT-L #1, ViT-L #2, and the ViT-L Ensemble all perform extremely well relative to models Virchow and Hibou-L despite seeing a fraction of the data.

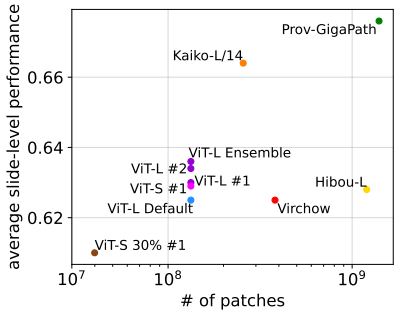

Figure 3: Average slide-level performance as a function of the number of patches in the training dataset. Performance of models trained in this study increases monotonically with dataset and model size. The impact of the dataset size appears to be stronger than the impact of the model size.

4.3 Correlation between patch-level and slide-level performance

Figure 4 demonstrates the relationship between WSS and slide-level performance. Using all points from the ViT-L #1, ViT-S #1, and DINOv2 baseline training runs, we find a correlation of 0.929 (95% CI: [0.785, 0.978]) between WSS and average slide-level performance (see Figure 5).

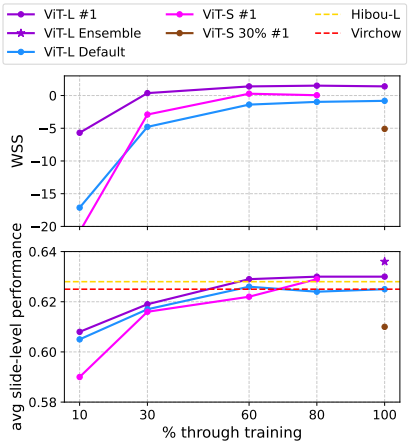

Figure 4: Patch-level (top) and slide-level (bottom) performance for the best ViT-L, ViT-S 100%, and ViT-S 30% models found during the hyperparameter searches. We compare to a ViT-L trained with DINOv2 default hyperparameters, as well as Hibou-L and Virchow. Kaiko-L/14 and Prov-GigaPath are omitted.

5 Discussion

Despite the impressive empirical progress in applying SSL to digital pathology, the field is still in its infancy. Even the largest models and datasets are small in comparison to analogous efforts in other domains, e.g., language modeling. Our work highlights the burning need for development of better methods for selecting SSL hyperparameters. Inspiration can be taken from language modeling, where significantly more resources have been put into understanding data and model scaling laws and how to accordingly tune hyperparameters. For example, some attention in the language modeling community has been given to the Maximal Update Parameterization (μP) [23, 24] which aims to provide theoretical foundations for hyperparameter transfer across model size. While it is unclear how to extend μP to vision-based SSL methods such as DINOv2, this research direction is critical.

References

[1] Jianfang Liu, Tara Lichtenberg, Katherine A Hoadley, Laila M Poisson, Alexander J Lazar, Andrew D Cherniack, Albert J Kovatich, Christopher C Benz, Douglas A Levine, Adrian V Lee, et al. An Integrated TCGA Pan-Cancer Clinical Data Resource to Drive High-Quality Survival Outcome Analytics. *Cell*, 2018.

[2] Nathan J Edwards, Mauricio Oberti, Ratna R Thangudu, Shuang Cai, Peter B McGarvey, Shine Jacob, Subha Madhavan, and Karen A Ketchum. The CPTAC Data Portal: A Resource for Cancer Proteomics Research. *Journal of Proteome Research*, 14(6), 2015.

[3] Christian Abbet, Inti Zlobec, Behzad Bozorgtabar, and Jean-Philippe Thiran. Divide-and-Rule: Self-Supervised Learning for Survival Analysis in Colorectal Cancer. In *MICCAI*, 2020.

[4] Charlie Saillard, Olivier Dehaene, Tanguy Marchand, Olivier Moindrot, Aurélie Kamoun, Benoit Schmauch, and Simon Jegou. Self-supervised learning improves dMMR/MSI detection from histology slides across multiple cancers. In *MICCAI*, volume 156, 2021.

[5] Andre Esteva, Jean Feng, Douwe van der Wal, Shih-Cheng Huang, Jeffry P Simko, Sandy DeVries, Emmalyn Chen, Edward M Schaeffer, Todd M Morgan, Yilun Sun, et al. Prostate cancer therapy personalization via multi-modal deep learning on randomized phase III clinical trials. *npj Digital Medicine*, 5(1), 2022.

[6] Xiyue Wang, Junhan Zhao, Eliana Marostica, Wei Yuan, Jietian Jin, Jiayu Zhang, Ruijiang Li, Hongping Tang, Kanran Wang, Yu Li, Fang Wang, Yulong Peng, Junyou Zhu, Jing Zhang, Christopher R. Jackson, Jun Zhang, Deborah Dillon, Nancy U. Lin, Lynette Sholl, Thomas Denize, David Meredith, Keith L. Ligon, Sabina Signoretti, Shuji Ogino, Jeffrey A. Golden, MacLean P. Nasrallah, Xiao Han, Sen Yang, and Kun-Hsing Yu. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, 2024.

[7] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical Image Analysis*, 81, 2022.

[8] Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, Yanbo Xu, Mu Wei, Wenhui Wang, Shuming Ma, Furu Wei, Jianwei Yang, Chunyuan Li, Jianfeng Gao, Jaylen Rosemon, Tucker Bower, Soohee Lee, Roshanthi Weerasinghe, Bill J Wright, Ari Robicsek, Brian Piening, Carlo Bifulco, Sheng Wang, and Hoifung Poon. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 2024.

[9] Nanne Aben, Edwin D de Jong, Ioannis Gatopoulos, Nicolas Känzig, Mikhail Karasikov, Axel Lagré, Roman Moser, Joost van Doorn, and Fei Tang. Towards large-scale training of pathology foundation models. *arXiv:2404.15217*, 2024.

[10] Alexandre Filiot, Ridouane Ghermi, Antoine Olivier, Paul Jacob, Lucas Fidon, Alice Mac Kain, Charlie Saillard, and Jean-Baptiste Schiratti. Scaling Self-Supervised Learning for Histopathology with Masked Image Modeling. *medRxiv*, 2023.

[11] Dinkar Juyal, Harshith Padigela, Chintan Shah, Daniel Shenker, Natalia Harguindeguy, Yi Liu, Blake Martin, Yibo Zhang, Michael Nercessian, Miles Markey, Isaac Finberg, Kelsey Luu, Daniel Borders, Syed Ashar Javed, Emma L Krause, Raymond Biju, Aashish Sood, Allen Ma, Jackson Nyman, John Shamshoian, Guillaume Chhor, Darpan Sanghavi, Marc Thibault, Limin Yu, Fedaa Najdawi, Jennifer A. Hipp, Darren Fahy, Benjamin Glass, Eric Walk, John Abel, Harsha Vardhan pokkalla, Andrew H. Beck, and Sean Grullon. PLUTO: Pathology-Universal Transformer. In *ICML*, 2024.

[12] Benedikt Roth, Valentin Koch, Sophia J Wagner, Julia A Schnabel, Carsten Marr, and Tingying Peng. Low-Resource Finetuning of Foundation Models Beats State-of-the-Art in Histopathology. In *IEEE International Symposium on Biomedical Imaging*, 2024.

[13] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, Nicolo Fusi, Ellen Yang, Philippe Mathieu, Alexander van Eck, Donghun Lee, Julian Viret, Eric Robert, Yi Kan Wang, Jeremy D. Kunz, Matthew C. H. Lee, Jan H. Bernhard, Ran A. Godrich, Gerard Oakley, Ewan Millar, Matthew Hanna, Hannah Wen, Juan A. Retamero, William A. Moye, Razik Yousfi, Christopher Kanan, David S. Klimstra, Brandon Rothrock, Siqi Liu, and Thomas J. Fuchs. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature Medicine*, 2024.

[14] Dmitry Nechaev, Alexey Pchelnikov, and Ekaterina Ivanova. Hibou: A Family of Foundational Vision Transformers for Pathology. *arXiv:2406.05074*, 2024.

[15] 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.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

[17] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, 45(10), 2013.

[18] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 1979.

[19] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the Yield of Medical Tests. *JAMA*, 247(18), 1982.

[20] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based Deep Multiple Instance Learning. In *ICML*, 2018.

[21] Georg Wölflein, Dyke Ferber, Asier Rabasco Meneghetti, Omar SM El Nahhas, Daniel Truhn, Zunamys I Carrero, David J Harrison, Ognjen Arandjelović, and Jakob N Kather. Benchmarking Pathology Feature Extractors for Whole Slide Image Classification. *arXiv:2311.11772*, 2024.

[22] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling Vision Transformers to Gigapixel Images via Hierarchical Self-Supervised Learning. In *CVPR*, 2022.

[23] Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer. In *NeurIPS*, 2021.

[24] Lucas Lingle. A Large-Scale Exploration of μ-Transfer. *arXiv:2404.05728*, 2024.

[25] Nicholas Petrick, Shazia Akbar, Kenny H Cha, Sharon Nofech-Mozes, Berkman Sahiner, Marios A Gavrielides, Jayashree Kalpathy-Cramer, Karen Drukker, Anne L Martel, et al. SPIE-AAPM-NCI BreastPathQ challenge: an image analysis challenge for quantitative tumor cellularity assessment in breast cancer histology images following neoadjuvant treatment. *Journal of Medical Imaging*, 8(3), 2021.

[26] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Naofumi Tomita, Lorenzo Torresani, Jason Wei, and Saeed Hassanpour. A Petri Dish for Histopathology Image Analysis. In *AIME*, 2021.

[27] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, 2018.

[28] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation Equivariant CNNs for Digital Pathology. In *MICCAI*, 2018.

[29] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22), 2017.

[30] Gang Xu, Zhigang Song, Zhuo Sun, Calvin Ku, Zhe Yang, Cancheng Liu, Shuhao Wang, Jianpeng Ma, and Wei Xu. CAMEL: A Weakly Supervised Learning Framework for Histopathology Image Segmentation. In *ICCV*, 2019.

[31] Chuang Zhu, Wenkai Chen, Ting Peng, Ying Wang, and Mulan Jin. Hard Sample Aware Noise Robust Learning for Histopathology Image Classification. *IEEE Transactions on Medical Imaging*, 41(4), 2021.

[32] Eirini Arvaniti, Kim S Fricker, Michael Moret, Niels Rupp, Thomas Hermanns, Christian Fankhauser, Norbert Wey, Peter J Wild, Jan H Rueschoff, and Manfred Claassen. Automated Gleason grading of prostate cancer tissue microarrays via deep learning. *Scientific Reports*, 8(1), 2018.

[33] Christian Matek, Sebastian Krappe, Christian Münzenmayer, Torsten Haferlach, and Carsten Marr. An Expert-Annotated Dataset of Bone Marrow Cytology in Hematologic Malignancies, 2021. [Data set].

[34] Andrew Janowczyk and Anant Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*, 7(1), 2016.

[35] Patrick Leavey, Anita Sengupta, Dinesh Rakheja, Ovidiu Daescu, Harish B Arunachalam, and Rashika Mishra. Osteosarcoma data from UT Southwestern/UT Dallas for Viable and Necrotic Tumor Assessment (Osteosarcoma Tumor Assessment), 2019. [Data set].

[36] Simon Graham, Quoc Dang Vu, Mostafa Jahanifar, Martin Weigert, Uwe Schmidt, Wenhua Zhang, Jun Zhang, Sen Yang, Jinxi Xiang, Xiyue Wang, Josef Lorenz Rumberger, Elias Baumann, Peter Hirsch, Lihao Liu, Chenyang Hong, Angelica I. Aviles-Rivero, Ayushi Jain, Heeyoung Ahn, Yiyu Hong, Hussam Azzuni, Min Xu, Mohammad Yaqub, Marie-Claire Blache, Benoît Piégu, Bertrand Vernay, Tim Scherr, Moritz Böhland, Katharina Löffler, Jiachen Li, Weiqin Ying, Chixin Wang, David Snead, Shan E. Ahmed Raza, Fayyaz Minhas, Nasir M. Rajpoot, et al. CoNIC Challenge: Pushing the frontiers of nuclear detection, segmentation, classification and counting. *Medical Image Analysis*, 92, 2024.

[37] Luca Bertero, Carlo Alberto Barbano, Daniele Perlo, Enzo Tartaglione, Paola Cassoni, Marco Grangetto, Attilio Fiandrotti, Alessandro Gambella, and Luca Cavallo. UNITOPATHO, 2021.

[38] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transactions on Biomedical Engineering*, 63(7), 2015.

[39] Hamidreza Bolhasani, Elham Amjadi, Maryam Tabatabaeian, and Somayyeh Jafarali Jassbi. A histopathological image dataset for grading breast invasive ductal carcinomas. *Informatics in Medicine Unlocked*, 19, 2020.

[40] Tan N N Doan, Boram Song, Trinh T L Vuong, Kyungeun Kim, and Jin T Kwak. SONNET: A Self-Guided Ordinal Regression Neural Network for Segmentation and Classification of Nuclei in Large-Scale Multi-Tissue Histology Images. *IEEE Journal of Biomedical and Health Informatics*, 26(7), 2022.

[41] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, Maria Gabrani, Florinda Feroce, and Maria Frucci. BRACS: A Dataset for BReAst Carcinoma Subtyping in H&E Histology Images. *Database*, 2022, 2022.

[42] Yiqing Shen, Yulin Luo, Dinggang Shen, and Jing Ke. RandStainNA: Learning Stain-Agnostic Features from Histology Slides by Bridging Stain Augmentation and Normalization. In *MICCAI*, 2022.

A Appendix

A.1 Patch-level tasks

Table 1: Patch-level tasks used for hyperparameter tuning. For classification tasks, the number in parentheses represents the number of classes.

Task name	# patches	Task type	Organ	Description	Metric
BreastPathQ [25]	2,579	regression	breast	tumor cellularity	Kendall's tau
MHIST [26]	3,152	classification (2)	colorectal	histology	AUC
CRC-100K [27]	107,000	classification (9)	colorectal	tissue types	balanced accuracy
PatchCamelyon [28, 29]	327,680	classification (2)	lymph node	metastases	balanced accuracy
CAMEL [30]	15,403	classification (2)	colorectal	adenomas	AUC
Chaoyang [31]	6,160	classification (4)	colorectal	adenomas	macro AUC
Gleason [32]	21,496	classification (4)	prostate	Gleason score	macro AUC
Cytomorphology [33]	171,373	classification (14)	bone marrow	cell types	macro AUC
IDC [34]	277,524	classification (2)	breast	invasive ductal carcinoma	AUC
Osteosarcoma [35]	1,144	classification (3)	bone	necrosis	balanced accuracy
CoNIC [36]	4,831	classification (6)	colon	cell types	1 - L1 loss
UniToPatho [37]	8,669	classification (6)	colon	lesion histology	macro AUC
BreakHis [38]	7,909	classification (8)	breast	lesion histology	macro AUC
IDC Grading [39]	906	classification (3)	breast	grade of invasive ductal carcinoma	macro AUC
GLySAC [40]	14,315	classification (7)	stomach	cell nuclei	macro AUC
BRACS [41]	4,539	classification (7)	breast	lesion histology	macro AUC

A.2 Slide-level tasks

Table 2: Slide-level tasks used for evaluation. Data for all TCGA cohorts are available publicly. For overall survival, the metric used is the C-index [19], and for all other tasks the metric is AUC.

Task name	Datasets	Targets
Mutations	TCGA-BRCA (938)	TP53, PIK3CA, CDH1
MSI	TCGA-COAD (332), TCGA-STAD (339)	
HRD	TCGA-BRCA (1022)	
Biomarkers	TCGA-BRCA (1062)	ER, HER2, IDC, ILC
Survival	TCGA-COAD (449), TCGA-STAD (388)	Overall survival

Slide-level tasks focus on predicting tumor molecular and histological characteristics and predicting patient survival. The molecular characteristics included predicting mutation status in several clinically meaningful genes often associated with targeted therapies. For example, patients with a PIK3CA mutation are candidates for alpesilib, patients with homologous recombination deficiency (HRD) are candidates for PARP inhibitors such as olaparib, and patients with microsatellite instability (MSI) are candidates for immune checkpoint inhibitors such as pembrolizumab. Testing for these mutations from a digitized pathology slide may enable avoiding time-consuming and expensive NGS panels, which are not available to all patients. Additionally, predicting patient survival typically relies on either basic clinicopathological characteristics or genomic signatures, which share the time and cost constraints of testing for genomic alterations and only have modest accuracy. Tasks focused on other molecular and histological biomarkers, such as identifying ductal (IDC) and lobular (ILC) carcinomas, are usually done by pathologists, and an automated tool could speed up diagnostic workflows.

A.3 Vision transformer details

Table 3: Architectural parameters of Vision Transformer model variants. Since various ViT-Giant configurations exist, we report the one used for Prov-GigaPath [8].

Model	Layers	Hidden size	MLP size	Heads	Params
ViT-Small	12	384	1536	6	21M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M
ViT-Giant	40	1536	8192	24	1.1B

A.4 Tuned hyperparameters

Table 4: A summary of hyperparameters tuned in this study and their initial ranges for the first (ViT-S 30% data) hyperparameter search.

Hyperparameter	Initial Range	Equation
LR	[0.0002, 0.0024] (log)	
LR final value factor	[0.01, 0.5] (log)	
LR warmup epochs	[0.02, 0.15]	
patch-embed LR multiplier	[0.1, 0.4]	
weight decay init	[0.01, 0.1] (log)	
weight decay final value factor	[1, 8] (log)	final value = init $\times$ factor
teacher temp init	[0.02, 0.06]	
teacher temp final value factor	[1.0, 3.0]	final value = init $\times$ factor
teacher temp warmup epochs	[0.25, 0.35]	
momentum init	[0.990, 0.996]	
momentum final value factor	[0.5, 1.0]	final value = init + (1-init) $\times$ factor
DINO local crop max	[0.3, 0.6]	global crop scale = (max, 1.0)
DINO local crop min factor	[0.1, 0.5]	local crop scale = (max $\times$ factor, max)
normal color jitter probability	[0.0, 1.0]	
stain color jitter intensity	[0.03, 0.10]	
stain color jitter probability	[0.6, 0.9]	
sample super-patch frequency	[0.0, 0.5]	

The final three hyperparameters of Table 4 are hyperparameters that we introduce based on recent papers showing potential for mixed-magnification training [11, 9] and stain augmentation [14]. In particular, 'sample super-patch frequency' represents the probability that the sampled 256×256 patch is combined with neigboring 256×256 to form a single 512×512 patch. 'stain color jitter intensity' and 'stain color jitter probability' determine the strength and frequency of the RandStainNA stain augmentation [42].

A.5 Training Data

Table 5: A breakdown of the TCGA cohorts used for training and the respective number of slides used and average patches per slide (PPS).

Cancer Subtype	Abbrev.	# Slides	Avg PPS
Bladder Urothelial Carcinoma	BLCA	107	14,991
Breast Invasive Carcinoma	BRCA	20	13,179
Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma	CESC	279	10,541
Cholangiocarcinoma	CHOL	39	19,847
Colon Adenocarcinoma	COAD	459	12,297
Diffuse Large B-Cell Lymphoma	DLBC	44	10,363
Esophageal Carcinoma	ESCA	158	12,181
Glioblastoma Multiforme	GBM	860	10,102
Head and Neck Squamous Cell Carcinoma	HNSC	472	11,941
Kidney Chromophobe	KICH	121	14,534
Kidney Renal Clear Cell Carcinoma	KIRC	519	14,333
Kidney Renal Papillary Cell Carcinoma	KIRP	298	13,738
Brain Lower Grade Glioma	LGG	844	10,787
Liver Hepatocellular Carcinoma	LIHC	379	14,261
Lung Adenocarcinoma	LUAD	541	12,824
Lung Squamous Cell Carcinoma	LUSC	512	12,756
Mesothelioma	MESO	87	10,393
Ovarian Serous Cystadenocarcinoma	OV	107	15,086
Pancreatic Adenocarcinoma	PAAD	209	12,936
Pheochromocytoma and Paraganglioma	PCPG	196	15,208
Prostate Adenocarcinoma	PRAD	449	13,223
Rectum Adenocarcinoma	READ	165	10,828
Sarcoma	SARC	600	17,227
Skin Cutaneous Melanoma	SKCM	475	13,360
Stomach Adenocarcinoma	STAD	442	11,640
Testicular Germ Cell Tumors	TGCT	254	15,797
Thyroid Carcinoma	THCA	519	14,131
Thymoma	THYM	181	15,335
Uterine Corpus Endometrial Carcinoma	UCEC	566	17,496
Uterine Carcinosarcoma	UCS	91	17,892
Uveal Melanoma	UVM	80	9,469

A.6 Correlation between WSS and slide-level performance

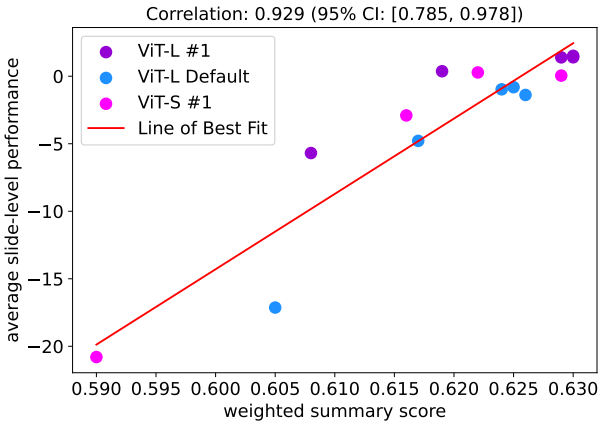

Figure 5: Relationship between weighted summary score and average slide-level performance. The strong correlation confirms that optimizing performance of the model on patch-level tasks is an effective way to tune hyperparameters for slide-level performance without tuning directly on the slide-level tasks. Different points for the same model represent different stages of training.