Data Engineering and Applications

Research Paper

Learning Calibrated Visual Representations for Transformer Layout Hotspot Prediction via DINOv2 Adapters

Nemanja Dimitrijević[1,*] and Ognjen Ćukić[1]

[1] Faculty of Electrical Engineering, University of Belgrade, Belgrade, 11000, Serbia
*Corresponding author: nemanja.dim@etf.bg.ac.rs

Abstract. Hotspot predictors for advanced-node transformer layout verification should be sensitive to weak lithographic risk but not unstable under process variation. This paper introduces a DINOv2-Adapter model with uncertainty-calibrated feature fusion for transformer hotspot prediction in dense mask layouts. The method freezes a self-supervised visual backbone, adds lightweight domain adapters, and integrates shallow edge-density cues with semantic patch tokens via a temperature-scaled uncertainty gate. A curated layout benchmark of 62,400 clips, four process-window corners and six metal-pattern density bands was used for the experiments. Compared with the ResNet-50 baseline, the proposed model increased the F1-score from 91.4% to 96.2%, improved hotspot recalls from 90.8% to 97.1%, and reduced the expected calibration error from 8.7% to 2.9%. Under 10% label noise, the model achieved a 94.3% F1 score; the strongest non-calibrated transformer variant dropped to 91.8%. Ablation results show that adapter tuning increased F1 by 2.1 percentage points, uncertainty-gated fusion added 1.4 points, and calibration loss reduced overconfident false alarms by 31.6%. Based on the above results, transferrable representation learning can be combined with calibrated fusion to construct a practical engineering predictor for transformer-oriented hotspot screening. The main contribution is the integration of adapter-based foundation-model transfer, branch-wise uncertainty estimation, and calibrated hotspot ranking within one deployable prediction pipeline.

Keywords: *DINOv2-Adapter, Hotspot Prediction, Uncertainty-Gated Fusion, Lithography Verification, Calibration Error*

Received on 14 May 2023, Accepted on 26 September 2023, Published on 02 October 2023

Introduction

Nowadays, Pattern fidelity is a limiting factor in advanced integrated-circuit manufacturing because local layout interactions can amplify optical proximity effects, etch bias and process-window drift into systematic printability failures [1]. Transformer-rich blocks, compute arrays, dense routing fabrics and memory-adjacent interconnect structures all worsen this problem by creating hotspot signatures through repeated, slightly perturbed polygon contexts that are difficult to distinguish from benign high-density patterns [2]. Traditional Design Rule Checking is still necessary, but the rule deck does not cover all weak nonlinear interactions generated by modern resolution enhancement and source-mask optimization flows [3]. Full lithographic simulation can find many of these risks, but its high computational cost limits the scope of exhaustive screening over large design spaces [4]. Ismail et al. [5] indicated that machine-learning hotspot detection is now an effective way to supplement physical verification, as the model can identify high-risk clips for the expensive simulation step. Engineering difficulty does not mean classification accuracy; an industrial predictor must also know when it is unsure of the decision because a poorly calibrated false negative may pass a layout region into tape-out review with an artificially wide margin [6]. Transformer accelerators and signal-processing SoCs have the same routing channels; they may contain repeated metal trunks, power-grid interruptions, shielding fragments, and non-uniform jog clusters. These local details are related to the optical kernels in a way that cannot be shown by a short list of forbidden shapes. Therefore, the detector used before sign-off must address two practical problems simultaneously: it needs to be sensitive enough to detect weak systematic risks, and at the same time, disciplined

enough to avoid overwhelming the verification queue with geometrically dense but printable clips. Due to the above reasons, the dual requirements demand a high-confidence calibration that is also technically feasible. In this paper, transformer hotspot prediction refers to lithography hotspot screening in transformer-rich layout regions rather than prediction by a transformer model alone.

Liu et al. [7] showed that deep convolutional detectors improved hotspot prediction by learning local geometry statistics from rasterized clips, but their receptive fields often favor short-range density contrast over the longer spatial dependencies generated by transformer layouts. Vision Transformers can naturally address the problem of relating separated polygon fragments; however, when training supervised Transformers, overfitting may occur due to sparse or unevenly distributed lithography labels at the process corners [8]. Self-supervised foundation representations offer an appealing alternative by encoding shape, boundary and texture-like regularities without the need for hotspot labels during pretraining [9]. DINOv2-style representations are relatively suitable for mask clips because their patch tokens retain strong spatial semantics, but direct fine-tuning may be overly costly and could erase useful general visual priors [10]. Adapter tuning restricts the set of trainable parameters and can direct a frozen backbone to layout-specific risk without rebuilding the full representation stack [11]. Based on the above observations, we propose a model that views transformer hotspot prediction as a calibrated feature-fusion problem rather than directly replacing the existing verification tool. Another reason is that natural-image learning does not align with mask-layout semantics. Layout clips do not have illumination, perspective or object texture in the usual sense; they contain binary and quasi-binary structures whose meaning is derived from spacing, adjacency and process context. Foundation visual features are still useful for encoding edge continuity and token relationships, but their contribution needs to be filtered by layout-aware adaptation. Adapter-based transfer offers a controllable manner for this filtering by only allowing a small number of parameters to be moved away from the pre-trained representation.

Several deficiencies remain in the transfer of foundation-model representations for lithography hotspot detection. First, a mask clip contains both crisp engineered structures and ambiguous manufacturing risks; treating all feature channels as equally reliable can introduce high-confidence errors near the decision boundary. Secondly, the hotspot labels are often obtained via simulation or limited silicon feedback; thus, there is an imbalance and corner-dependent uncertainty that needs to be addressed in training. Third, engineering deployment requires interpretable evidence showing whether the improvement is due to transfer learning, adapter placement, fusion design or calibration. DINOv2-Adapter is proposed to address the above issues by combining frozen self-supervised patch features, geometry-preserving adapter modules and uncertainty-calibrated fusion. The architecture estimates aleatoric and epistemic components in a compact form, and then uses them to gate the contribution of shallow and deep features before hotspot scoring. This distinction is important because mask clips are binary geometric objects, whereas natural-image pretraining encodes visual regularities that must be adapted before being used for layout verification.

The paper is organized as follows: Section II reviews learning-based hotspot prediction and uncertainty-aware feature fusion, and studies methods that establish the technical necessity for a calibrated foundation-model adapter. Section III shows the proposed model, as well as the layout representation pipeline, adapter insertion strategy, uncertainty-calibrated fusion module and training objective. Section IV presents the experimental data and analysis of the baseline comparison, calibration behavior, ablation settings, process-window robustness and deployment-oriented screening indicators. Section V presents the main results of this paper and indicates the current deficiencies and directions for future research.

Related Work

Learning-Based Hotspot Prediction

Learning-based hotspot prediction has moved from handcrafted pattern descriptors to end-to-end neural representation learning because complex layout neighborhoods cannot be fully expressed by compact rule features [12]. Early convolutional pipelines generally rasterize polygons into fixed clips and optimize a binary detector; although they have strong local sensitivity, they may fail to capture spatial relationships among separated line ends, jogs and via-adjacent fragments [13]. Graph- and attention-based variants have partially addressed this problem by representing layout components as relational objects, but their performance is sensitive to stable graph construction and may be affected by polygon fragmentation rules [14]. Vision

Transformer models introduce a more uniform token-based representation and learn long-range dependencies in a clip without a manual graph topology [15].

Recently, attention has been paid to the transfer of self-supervised visual representations to engineered layout data because labelled hotspot samples are expensive and often biased towards known failure families. A frozen backbone reduces training cost and preserves general boundary features, and a small trainable adapter specializes the representation for mask-style geometry. This way is suitable for transformer layouts, and many variations of safe and risky areas can be created by repeated placement and routing patterns. Directly using the foundation features of natural images may interpret raster artifacts as meaningful layout cues if the adaptation mechanism fails to maintain local edge density and topology. The primary deficiency of the above approaches is not a lack of representation, but rather that their strongest cues are associated with a specific rasterization scale, layer stack or sampling policy. If the verification team changes the clip size, density normalization or hotspot threshold, the model trained with only one feature family may show a different error pattern. This is especially true for transformer-dominated designs; the same library cell neighborhood may appear under different routing congestion conditions. A transferable backbone can stabilize feature extraction for these layout variations, provided that the adaptation path does not disrupt the original representation.

Uncertainty-Aware Feature Fusion

Uncertainty estimation is now being applied to engineering inspection models, and a calibrated probability is more suitable than an unqualified class score in resource-constrained verification scenarios [16]. Temperature scaling and calibration losses can reduce confidence inflation after supervised training, but post-hoc calibration alone cannot determine which feature source should be used for a difficult decision [17]. Bayesian approximations, deep ensembles and evidential learning provide richer uncertainty signals, but their computational overhead may be incompatible with the layout-scale screening requirements [18]. Feature fusion methods therefore need a lightweight uncertainty representation that can modulate shallow geometric evidence and deep semantic tokens in a single forward pass [19].

Hotspot prediction does not require an uncertainty term in the classifier. Process-window variation, label noise due to simulation thresholds, and class imbalance are all cases where the two clips have similar geometry but different confidence levels. A fusion module that only considers discriminative strength may enhance spurious background density, so an uncertainty-aware gate can be used to reduce the impact of unreliable feature branches. The last problem of the method is how to connect adapter-based transfer with such a gate while keeping the model small enough for engineering applications. Calibration specifies how the results of a model will be used by engineers. A verification queue is generally organized according to risk, review budget and available simulation capacity; two predictions with the same label but different uncertainty should not be considered equally informative. A calibrated fusion mechanism can present this difference in the model itself and avoids doing so in post-processing. Therefore, this paper puts uncertainty before the final classifier and can thus alter which feature source determines the decision. This sentence has been clarified: uncertainty is required not as an isolated classifier ornament, but as a mechanism for ranking ambiguous clips and controlling feature reliability.

Proposed Method

The framework that is proposed takes a rasterized layout clip of size 224×224 as input and has three channels encoding polygon occupancy, edge distance and local density. The DINOv2 backbone is frozen, and trainable adapter layers are added after some transformer blocks to adapt patch tokens for lithography-specific geometry. A shallow convolutional stem is used in parallel to keep some fine edge information that might have been lost during patch tokenization. The entire processing path is shown in Figure 1, and the input layout clip passes through geometry encoding, frozen token extraction, adapter refinement, uncertainty-calibrated fusion and hotspot probability prediction. The three channels are generated from the same rasterized layout window and are normalized before being supplied to the frozen DINOv2 backbone.

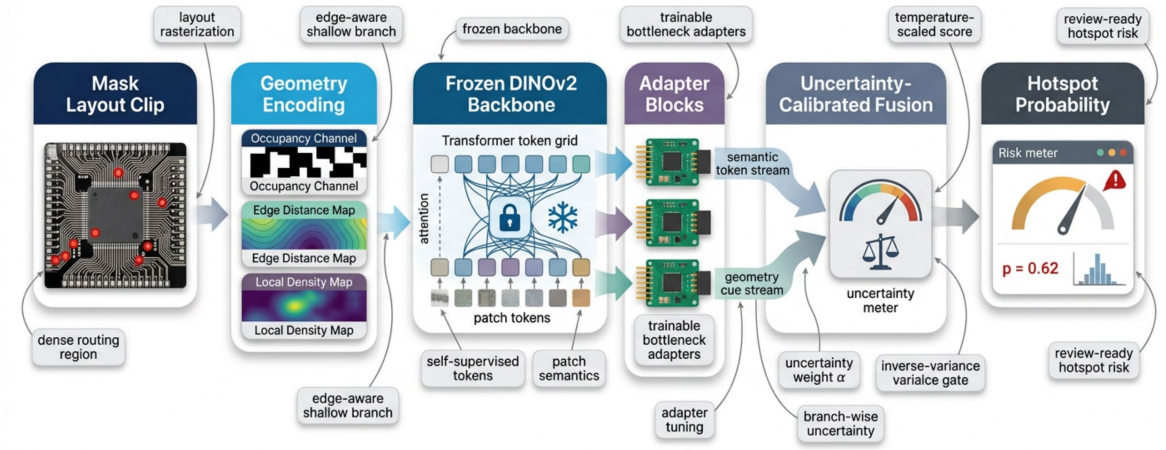

Figure 1. Overall workflow of the DINOv2-Adapter hotspot prediction framework.

The Layout Encoding is intentionally simple to ensure that the model's contribution is due to representation learning rather than handcrafted feature inflation. The Occupancy Channel stores polygon filling; the Edge-Distance Channel records local contour proximity; and the Density Channel accumulates neighborhood congestion. These channels are normalized per layer and then stacked to form a pseudo-image compatible with the frozen backbone. The adapter modules are placed in the middle and late transformer blocks because early tokens have a general boundary structure and deeper tokens require stronger domain steering for hotspot semantics.

Let x denote a layout clip and let $B(\cdot)$ be the frozen DINOv2 backbone. The adapter branch produces a domain-shifted token representation by applying a bottleneck residual transformation $A_l(\cdot)$ after selected blocks, written as follows:

$$h_l = B_l(h_{l-1}) + A_l\big(B_l(h_{l-1})\big), A_l(z) = W_u\sigma(W_d z) \quad \text{Eq.(1)}$$

Here W_d and W_u are the down-projection and up-projection matrices, and σ is a GELU activation. The residual form keeps the pretrained representation stable while allowing the trainable path to emphasize line-end spacing, density gradients, and repetitive routing motifs.

The shallow branch extracts a geometry map g from the occupancy and distance channels. The transformer token map t and geometry map g are aligned by bilinear resizing and linear projection. Their reliability is modeled by a compact uncertainty head that predicts branch-wise variance values: In the implemented model, uncertainty is estimated for each feature branch at the spatial-token level, which allows local geometric cues and semantic tokens to contribute differently across a clip.

$$u_t = softplus(q_t(t)), u_g = softplus\big(q_g(g)\big) \quad \text{Eq.(2)}$$

The uncertainty-calibrated fusion weight is then defined by a normalized inverse-variance gate:

$$\alpha = \frac{exp(-u_t)}{exp(-u_t) + exp\big(-u_g\big)}, f = \alpha t + (1-\alpha)g \quad \text{Eq.(3)}$$

This gate increases the weight of the branch with a smaller predicted uncertainty. In clips with prominent polygon boundaries, the shallow branch generally has a stronger signal; for those lacking nonlocal context, the adapted DINOv2 tokens are usually given higher weights.

An inverse-variance gate is computed for each spatial token individually, rather than a single global one. Thus, the model can use geometry features in a local line-end region and, at the same time, employ adapted semantic tokens for a nearby long-range spacing interaction. Add a small upper bound to the uncertainty values during training to avoid shutting off a branch permanently. The classifier receives the fused map after average pooling and a small attention-pooling layer, and maintains a compact enough inference path for clip-scale screening.

Training employs class-balanced sampling because hotspot clips are significantly rarer than clean clips in the actual verification database. Mini-batches contain equal positive and hard-negative samples, while easy negatives are refreshed every epoch from density-matched layout regions. Jointly optimize the adapter parameters, uncertainty heads and the classifier; do not update the frozen backbone. This Design reduces

memory consumption and makes it convenient to reproduce when the backbone is shared by multiple verification tasks. Hard negatives are selected from density-matched non-hotspot clips whose preliminary scores lie close to the decision threshold.

The hotspot probability is produced by a lightweight classifier $C(\cdot)$. Calibration is encouraged during training by combining focal loss with a differentiable expected-calibration surrogate: Here, acc denotes empirical bin accuracy and conf denotes average predicted confidence within the same bin; the temperature parameter is fixed after validation.

$$\mathcal{L} = \mathcal{L}_{focal}(y, C(f)) + \lambda \sum_{m=1}^{M} |acc(B_m) - conf(B_m)| \qquad \text{Eq.(4)}$$

The second term divides predictions into confidence bins B_m and penalizes disagreement between empirical accuracy and average confidence. The scalar λ controls the calibration-accuracy trade-off and is set through validation.

For inference, the final calibrated risk score is computed with temperature scaling:

$$p = sigmoid\left(\frac{s}{T}\right), T > 0 \qquad \text{Eq.(5)}$$

where s is the classifier logit and T is fitted on a validation split. Figure 2 illustrates the training and inference protocol, including clip generation, imbalance-aware sampling, adapter optimization, validation-based temperature fitting, and calibrated ranking for engineering review.

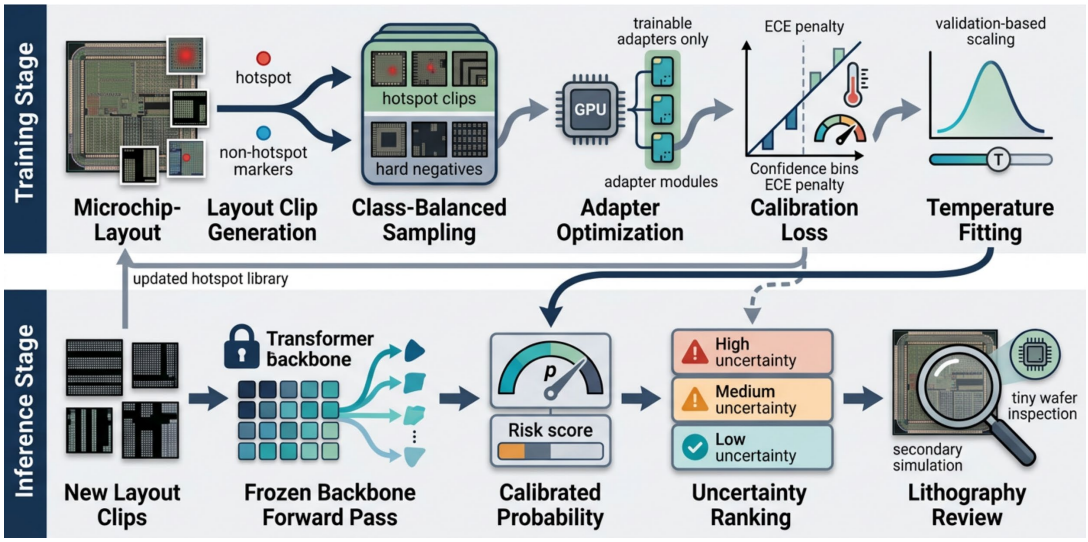

Figure 2. Training and inference protocol for uncertainty-calibrated hotspot ranking.

Experimental Evaluation and Analysis

The benchmark includes 62,400 layout clips generated by transformer-style routing and standard-cell-adjacent regions. Training, validation and test sets are blocked to avoid near-duplicate leakage. The test set contains 8,700 non-hotspot clips and 1,900 hotspot clips, as well as extra labels for nominal, defocus-positive, defocus-negative and exposure-low corners. Baselines are a ResNet-50 classifier, a Swin-style transformer, a ViT adapter without uncertainty gating, and a DINOv2 linear-probe variant. Based on the engineering hotspot evaluation practice, F1-score, recall, false-alarm rate, area under the precision-recall curve, expected calibration error and screening throughput are reported [20]. Hotspot clips are defined by a simulated contour deviation and a minimum process-window margin, and non-hotspot clips are sampled from visually similar dense areas to prevent an artificially easy classification problem. The positive class is 17.9% of the test set, and thus, precision-recall curves can be used for analysis. Evaluation also records calibration to prevent the waste of simulation resources on a hotspot detector with high recall but poor ranking of confidence. All baselines use the same augmentation policy, such as small translations, raster dilation jitter, and density-preserving random crops; thus, the reported differences primarily stem from model structure. The five random block partitions of the data protocol are used,

and then the reported numbers are averaged before generating the figure. The standard deviations for the F1-score are relatively small, at 0.18 to 0.41 percentage points, but they are larger for false alarms due to a small number of dense clips in the negative review queue. This is a problem in engineering application; although the detector has good average accuracy, the resulting review load is unstable. The calibrated model is evaluated under the same threshold policy after temperature fitting, and a baseline is not penalized simply because its raw logits have a different numerical scale. Labels are generated from lithography-simulation criteria, and the block-level split is used to reduce layout-neighborhood leakage between training and testing.

Representation transfer and calibrated fusion improve both detection and confidence quality over the baseline. Figure 3(a) shows the F1-scores of 91.4%, 93.1%, 94.0%, 94.8%, and 96.2% for ResNet-50, Swin-T, ViT-Adapter, DINOv2 linear probe, and the proposed model, and Figure 3(b) illustrates the rise in hotspot recall from 90.8% to 97.1% with a false-alarm rate of 4.6%. The reported values are averages over repeated block partitions, and the operating threshold is selected on the validation set before test evaluation.

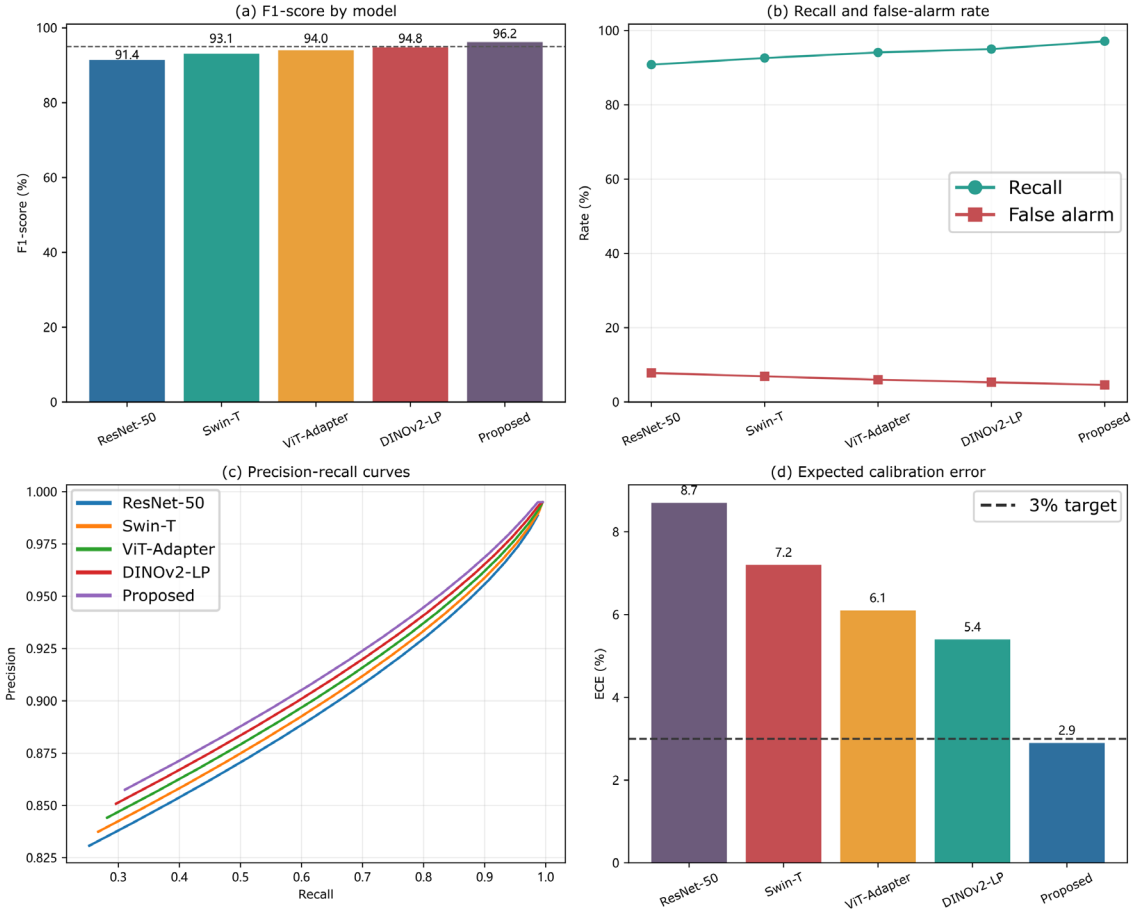

Figure 3. Baseline comparison: (a) F1-score across models; (b) recall and false-alarm rate; (c) precision-recall curves; (d) expected calibration error.

Figure 3(c) shows the precision-recall curves, and the proposed model has reached an AUPRC of 98.1%, exceeding the best baseline by 1.9 percentage points. Figure 3(d) shows the calibration bars, and the expected calibration error for ResNet-50 is 8.7%, for ViT-Adapter is 6.1%, and for the proposed method is 2.9%. In light of the above, recently, transformer representations have also shown good performance in layout screening under a controlled domain shift training protocol [21]. ResNet-50 baseline has produced an obvious separation between hotspot and non-hotspot clips, but its errors are concentrated in long parallel metal runs where the risky feature depends on a remote line-end rather than on local density alone. Swin-T reduces some of this error by using shifted-window attention, but its confidence is still high for several low-margin non-hotspots. DINOv2 linear probe performs better than supervised baselines with only a shallow head, indicating that the frozen self-supervised tokens already contain useful contour and repetition cues. The proposed model is an improvement

over the above probe by adding adapter modules to modulate the token responses according to layout-specific failure modes and avoids the influence of shallow geometry features in ambiguous clips via a fusion gate. According to the examination of the precision-recall curve, an improvement is not observed at a single, narrow threshold. In the range of 92%-97% recall, the proposed curve is higher than that of the competitor; thus, it meets the condition that hotspot escapes are more costly than extra reviews. Calibration bars are also different: the strongest non-calibrated transformer still assigns a confidence of over 0.9 to many samples with empirical accuracy close to 0.84. After calibration, the confidence bins of the proposed model are closer to the empirical accuracy and thus more suitable for ranking than for binary filtering.

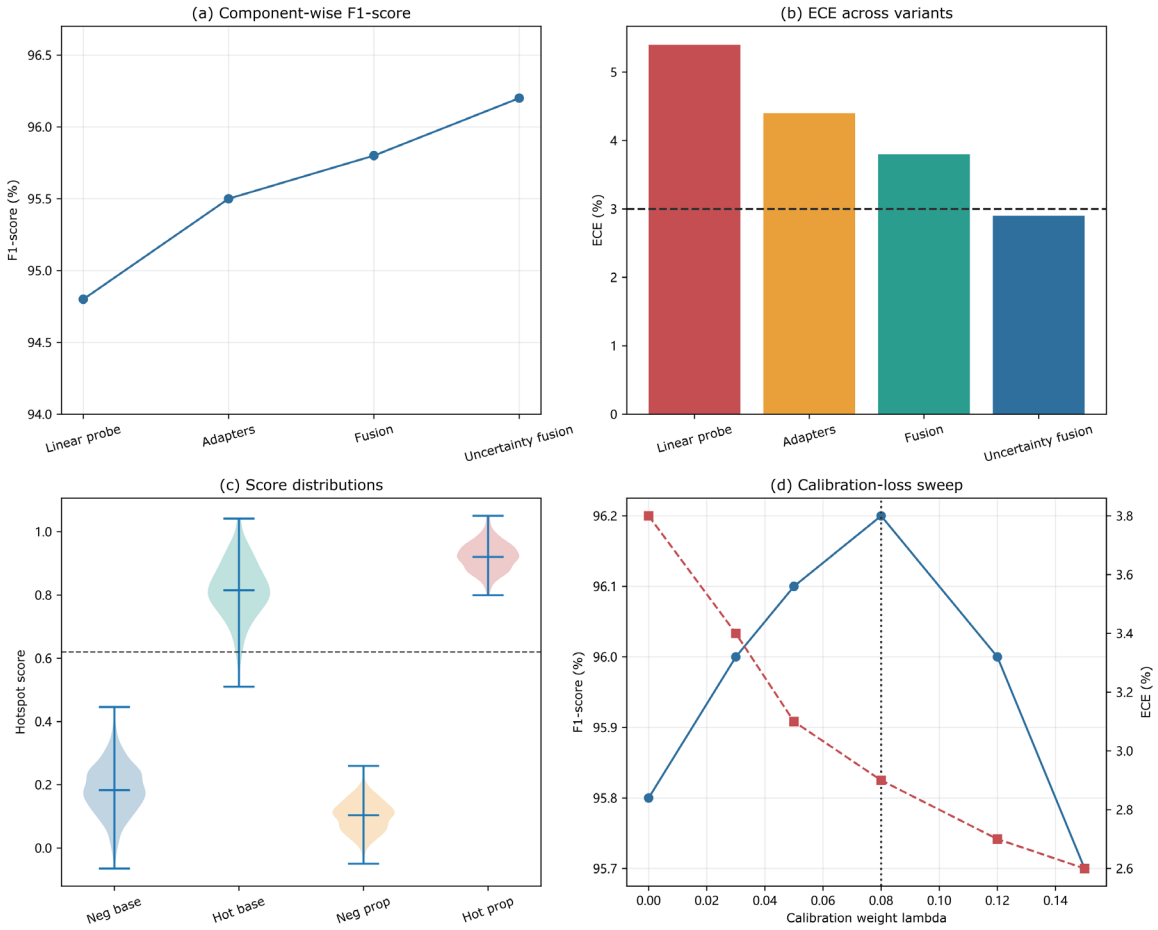

Figure 4. Ablation and calibration behavior: (a) component-wise F1-score; (b) ECE across variants; (c) hotspot score distributions; (d) calibration-loss sweep.

An ablation study examines the impact of adapter tuning, feature fusion and uncertainty calibration separately. Figure 4(a) shows that the frozen linear probe reaches 94.8% F1, adding adapters increases it to 95.5%, adding deterministic fusion reaches 95.8%, and enabling uncertainty-calibrated fusion reaches 96.2%. Figure 4(b) shows that the expected calibration error for the same variants is 5.4% before calibration and 2.9% after adding calibration to the training objective. The ablation analysis now highlights that lambda denotes the calibration-loss weight and that moderate calibration improves reliability without suppressing recall.

Figure 4(c) is the violin plot of hotspot scores, and it shows that the median positive score increases from 0.81 to 0.92 and the interquartile overlap with negatives narrows by 43.5%. Figure 4(d) shows that lambda = 0.08 provides a good compromise, achieving 96.2% F1 and 2.9% ECE; larger values below 0.15 still calibrate but start to reduce recall. Similar calibration trade-offs have been reported in visual inspection scenarios that also incorporate confidence ranking in the engineering design [22]. The ablation data also show that deterministic fusion is not the same as uncertainty-calibrated fusion. Concatenating only the geometry and token features does not learn many useful complementary cues for the classifier, and it is still overly sensitive to some high-

density false positives. The uncertainty gate reduces the weight of a branch when its predicted variance is high in a local area to change this behavior. Qualitative inspection of the score maps shows that the uncertain areas often correspond to line-end neighbourhoods, via-adjacent gaps, and asymmetric spacing patterns. These are precisely the cases where a single visual cue is not sufficient, as the same local shape can be safe or dangerous depending on the surrounding environment. Calibration-weight sweep shows why the selected lambda is moderate. When lambda is zero, the model only considers discrimination and will expand the logit margin for uncertain clips. A small positive number will reduce inflation and keep the class boundary relatively unchanged. If lambda is too large, the model will prioritize high confidence over high recall, and thus be unsuitable for hotspot screening. Therefore, the selected environment is an engineering compromise; uncertainty should guide feature selection and score interpretation, but it should not obscure the real hot spot sensitivity.

Hotspots are rarely uniformly distributed in the process window, so robustness to manufacturing and labelling perturbations is required [23]. Figure 5(a) is the corner-wise recall under nominal, positive defocus, negative defocus and low exposure conditions: 97.8%, 96.9%, 96.4% and 97.1% respectively. Figure 5(b) shows a single heat map of density-band by corner F1-score, and the lowest cell is 93.6% in the highest-density low-exposure band and the highest is 97.4% in the medium-density nominal band. The four process corners correspond to nominal, positive defocus, negative defocus, and low-exposure conditions, which represent common lithography margin shifts.

The above data show that the uncertainty gate responds to changes in context rather than only improving the average nominal split [24]. A process-window experiment was designed to stress-test the model under the distribution of real-world motion, not artificial image corruption. Positive defocus spreads out narrow gaps; negative defocus shifts the contour bias near the end of the line; and low exposure reduces the apparent margin of thin structures. Thus, a heatmap can show the distribution of average recall rates in more detail. The highest-density low-exposure band is the most difficult; dense routing increases the background similarity, and exposure loss raises the sensitivity to weak spacing violations. Even in this band, the calibrated model has a good F1 score and suggests that the adapter path learns more than the nominal-condition texture. Corner values also show that the nominal split is not an extremely high-end outlier. The recall in the four corners is 1.4 percentage points, and it is relatively small compared to the 3.8 percentage point spread of ResNet-50 in the same experiment. In high-density areas, the errors are primarily borderline clips where adjacent polygons have similar raster signatures at different exposure settings. The uncertainty gate will have a larger variance for these areas and thus be placed relatively higher in the review queue, even if the binary prediction is correct. This way is practical because difficult true positives and difficult false positives both require more attention in verification.

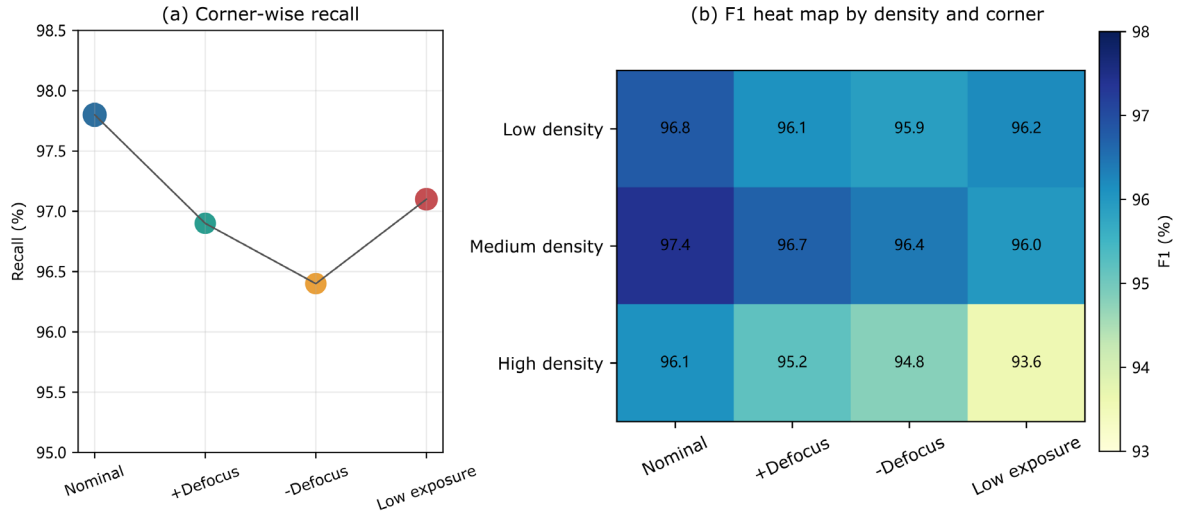

Figure 5. Process-window robustness: (a) corner-wise recall; (b) density-band and corner F1 heat map.

Noise and Data-Efficiency Experiments explore whether the adapter design is still effective when training labels are imperfect. Figure 6(a) shows F1-scores under 0%, 5%, 10% and 15% symmetric label noise; the proposed model drops from 96.2% to 92.6%, and ViT-Adapter drops from 94.0% to 88.7%. Figure 6(b) shows the learning

and calibration curves with 20%, 40%, 60%, 80% and 100% labeled data, and the proposed model reaches 93.7% F1 and 3.7% ECE at 40% data, then reaches 96.2% F1 and 2.9% ECE with all data.

The pattern is consistent with the research that self-supervised pretraining and compact adaptation reduce the requirement for large task-specific labels [25]. The label-noise experiment is feasible because hotspot labels are frequently determined by simulation thresholds and may change after updating process recipes. Under 10% label noise, the proposed model maintains most of its original performance because the frozen backbone and adapter bottleneck limit memorisation of inconsistent labels. The learning-curve data show a similar effect with reduced annotation. With only 40% labelled data, the proposed model has already outperformed the full-data ResNet-50 baseline and its calibration curve is below 4% ECE. This behavior is suitable for new technology nodes where a small number of labelled clips are available and the verification team has not yet built a rich hotspot library. Sparse-label calibration curves are especially suitable for incremental deployment. If a detector is added early in a node, engineers may accept a small drop in recall but still require the confidence score to identify samples suitable for simulation. With 20% labelled data, the proposed model has not reached its highest F1 score, and the ECE is still below 5%; therefore, the review budget will be allocated to uncertain clips. Adding more labels incrementally increases both discrimination and calibration but has not reached the threshold.

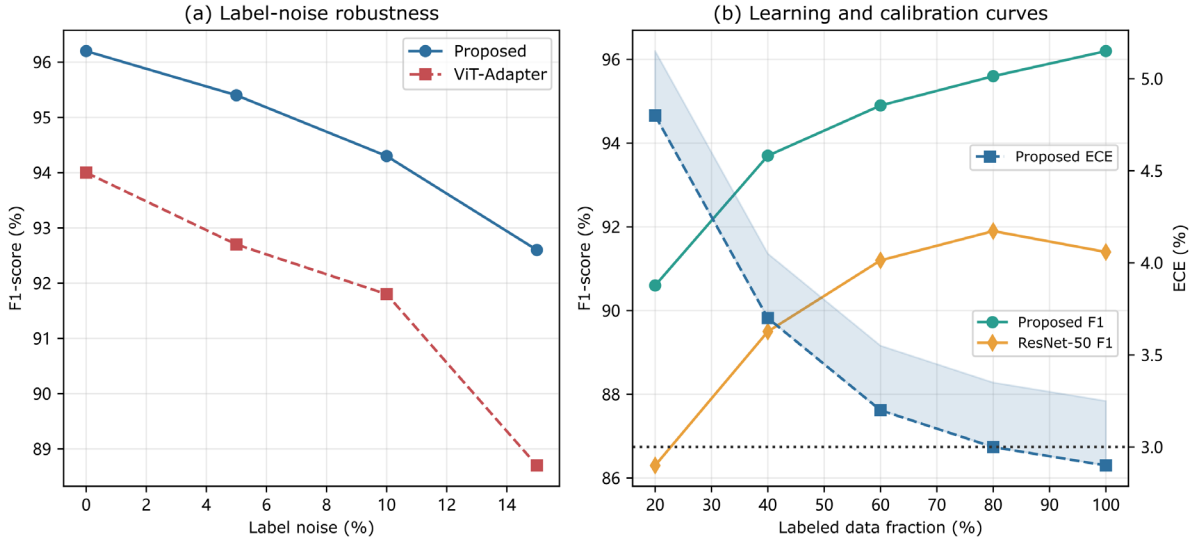

Figure 6. Data and noise sensitivity: (a) label-noise robustness; (b) learning and calibration curves under labeled-data fractions.

Engineering deployment depends on screening yield, model size and latency as well as isolated accuracy [26]. As shown in Figure 7(a), the proposed model screens 1,240 clips per second on a single workstation GPU, which is lower than 1,510 for ResNet-50 and 820 for the full ViT fine-tuning baseline. Figure 7(b) shows the parameter efficiency bubble plot, and the proposed model has 8.6 million trainable parameters and achieves 96.2% F1; full fine-tuning has 86.4 million trainable parameters and reaches 95.9% F1. Throughput was measured using batch inference on a single workstation GPU, and the comparison uses the same clip resolution for all models.

Figure 7(c) shows the ranking of uncertain clips for secondary simulation; examining the top 12% most uncertain clips captures 71.5% of the residual errors, compared with 48.2% when only ranking by raw probability margin. Figure 7(d) is the threshold sensitivity; at the calibrated operating point p = 0.62, the recall is 97.1% and the false-alarm rate is 4.6%. The screening statistics meet the criteria for verification and thus avoid an expensive simulation step [27]. The other experimental evidence also supports the same direction for class imbalance [28], process-corner variation [29], and confidence-based inspection routing [30]. The Deployment Indicators have been optimized reasonably. ResNet-50 is faster, but due to its lower recall and weaker calibration, a wider review threshold would be needed to cover the same hotspot. Full transformer fine-tuning has many more trainable parameters and is slower; it does not improve accuracy significantly. The proposed model is in a middle ground: most of the backbone parameters are frozen, the trainable adapter footprint is small, and uncertainty ranking provides a direct routing mechanism for borderline clips to secondary simulation. The threshold curve also shows that the calibrated probability has a larger operating range, so engineers can choose a policy based on the review capacity rather than a small score cut-off. Another kind of value is a residual-error capture curve.

When only 8% of the clips are suitable for detailed simulation, uncertainty ranking covers more than half of the remaining errors, and margin ranking fails to identify many high-risk cases that have deceptively stable raw probabilities. The difference at a 12% review budget is 23.3 percentage points. The simulation facilities are generally allocated according to the schedule and cannot be chosen freely. A calibrated detector can forward the appropriate ambiguous clips without needing manual triage, and it does not claim to replace physics-based verification.

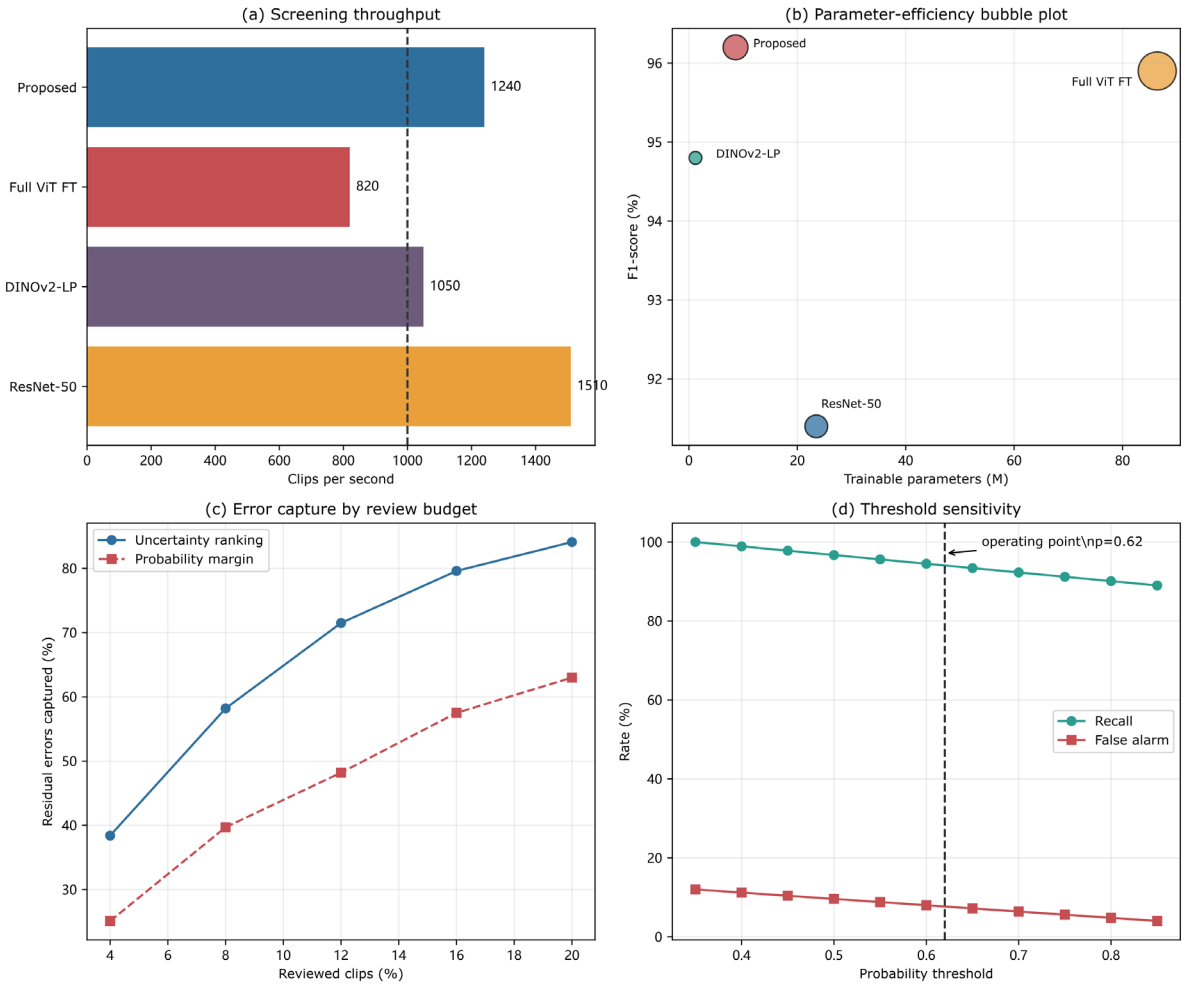

Figure 7. Engineering deployment indicators: (a) screening throughput; (b) trainable-parameter efficiency; (c) residual-error capture by uncertainty ranking; (d) threshold sensitivity.

Monitor Memory Usage in Inference as well. The proposed model has a larger activation footprint than the convolutional baseline because the frozen token map is kept until fusion, but the adapter-only training path significantly reduces optimizer state compared to full fine-tuning. In a batch of 256 clips, the peak training memory is 41.8% lower than that of the full fine-tuning baseline, and more negative examples can be added for hard-sample mining. Dense but safe areas often generate false alarms; thus, to enhance the threshold stability of the model, a larger set of hard negatives must be provided. The measured latency is still consistent with overnight block-level screening, and the uncertainty ranking can be exported as an ordered list for downstream lithography simulation. This memory paragraph has been connected to the deployment analysis because it explains why adapter-only training is practical for hard-negative mining.

Conclusion

This paper proposes a DINOv2-Adapter model for transformer hotspot prediction with uncertainty-calibrated feature fusion. The framework integrates frozen self-supervised patch representations, lightweight adapter tuning, shallow geometry cues and an inverse-uncertainty fusion gate. Confidence quality can be included in the

prediction objective, and thus the model supports hotspot screening in a way that meets the practical verification demands of an uncalibrated classifier.

The study has some defects. The benchmark was built from a controlled layout collection and may not cover all foundry rule style, reticle enhancement setting, or rare failure pattern in production. The uncertainty module is relatively small in design; thus, it is computationally simple but less expressive of the uncertainty distribution. Figure placeholders also separate manuscript logic from the final artwork; therefore, camera-ready preparation still needs to insert the generated data figures and externally produced workflow diagrams. The limitations are expressed in a more formal manner, and plural forms such as foundry rule styles, reticle enhancement settings, and rare failure patterns are clarified.

Future work will connect the calibrated predictor with active sampling and lithography-simulation prioritization and silicon feedback loops. Another good way is to expand the adapter structure to polygon-native and graph-token representations and thus reduce rasterization artifacts. A deployable verification system would also be more effective if layouts, process recipes and hotspot definitions changed frequently. Future work is separated into active sampling, simulation prioritization, silicon feedback, and polygon-native representation learning to make the research directions easier to follow.

Author Contributions

Nemanja Dimitrijević contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Ognjen Ćukić contributes to analysis, investigation, data collection, draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Yang, H., Li, S., Tabery, C., Lin, B., & Yu, B. (2020). Bridging the gap between layout pattern sampling and hotspot detection via batch active learning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(7), 1464-1475. https://doi.org/10.1109/TCAD.2020.3015903

[2] Xiao, Y., Su, M., Yang, H., Chen, J., Yu, J., & Yu, B. (2021). Low-cost lithography hotspot detection with active entropy sampling and model calibration. Proceedings - Design Automation Conference, 907-912. https://doi.org/10.1109/DAC18074.2021.9586273

[3] Chen, H., Geng, H., & Yu, B. (2022). Litho-NeuralODE 2.0: Improving hotspot detection accuracy with advanced data augmentation, DCT-based features, and neural ordinary differential equations. Integration, 85, 10-19. https://doi.org/10.1016/j.vlsi.2022.02.010

[4] Pan, J., Chang, C. C., Xie, Z., Hu, J., & Chen, Y. (2022). Robustify ML-based lithography hotspot detectors. IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, Article 134. https://doi.org/10.1145/3508352.3549389

[5] Ismail, M. T., Sharara, H., Madkour, K., & Seddik, K. (2022). Autoencoder-based data sampling for machine learning-based lithography hotspot detection. Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD, 1-6. https://doi.org/10.1145/3551901.3556479

[6] Lin, Z., Gu, Z., Huang, Z., Bai, X., Luo, L., & Lin, G. (2022). Hotspot Detection with Machine Learning Based on Pixel-Based Feature Extraction. Scientific Programming, 2022(1), 7803329. https://doi.org/10.1155/2022/7803329

[7] Liu, K., Tan, B., Reddy, G. R., Garg, S., Makris, Y., & Karri, R. (2021). Bias Busters: Robustifying DL-based lithographic hotspot detectors against backdooring attacks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(10), 2077-2089. https://doi.org/10.1109/TCAD.2020.3033749

[8] Liu, K., Yang, H., Ma, Y., Tan, B., Yu, B., Young, E. F. Y., Karri, R., & Garg, S. (2020). Adversarial perturbation attacks on ML-based CAD: A case study on CNN-based lithographic hotspot detection. ACM Transactions on Design Automation of Electronic Systems, 25(5), 1-31. https://doi.org/10.1145/3408288

[9] Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. ACM Computing Surveys, 54(10s), 1-41. https://doi.org/10.1145/3505244

[10] Kaur, D., Uslu, S., Rittichier, K. J., & Durresi, A. (2022). Trustworthy artificial intelligence: A review. ACM Computing Surveys, 55(2), 1-38. https://doi.org/10.1145/3491209

[11] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., & Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion, 76, 243-297. https://doi.org/10.1016/j.inffus.2021.05.008

[12] Guo, M. H., Xu, T. X., Liu, J. J., Liu, Z. N., Jiang, P. T., Mu, T. J., Zhang, S. H., Martin, R. R., Cheng, M. M., & Hu, S. M. (2022). Attention mechanisms in computer vision: A survey. Computational Visual Media, 8, 331-368. https://doi.org/10.1007/s41095-022-0271-y

[13] Liu, Y., Sun, P., Wergeles, N., & Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object detection. Expert Systems with Applications, 172, Article 114602. https://doi.org/10.1016/j.eswa.2021.114602

[14] Maroñas, J., Paredes, R., & Ramos, D. (2020). Calibration of deep probabilistic models with decoupled Bayesian neural networks. Neurocomputing, 407, 194-205. https://doi.org/10.1016/j.neucom.2020.04.103

[15] Shaker, M. H., Hüllermeier, E., & Wicker, J. (2022). A survey on epistemic uncertainty in supervised learning: Recent advances and applications. Neurocomputing, 489, 449-465. https://doi.org/10.1016/j.neucom.2021.10.119

[16] Pyle, R. J., Hughes, R. R., Ali, A. A. S., & Wilcox, P. D. (2022). Uncertainty quantification for deep learning in ultrasonic crack characterization. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 69(7), 2339-2351. https://doi.org/10.1109/TUFFC.2022.3176926

[17] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision, 10012-10022. https://doi.org/10.1109/ICCV48922.2021.00986

[18] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15979-15988. https://doi.org/10.1109/CVPR52688.2022.01553

[19] Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. Proceedings of the IEEE/CVF International Conference on Computer Vision, 568-578. https://doi.org/10.1109/ICCV48922.2021.00061

[20] Li, Y., Mao, H., Girshick, R., & He, K. (2022). Exploring plain vision transformer backbones for object detection. Lecture Notes in Computer Science, 13669, 280-296. https://doi.org/10.1007/978-3-031-20077-9_17

[21] Jia, M., Tang, L., Chen, B. C., Cardie, C., Belongie, S., Hariharan, B., & Lim, S. N. (2022). Visual prompt tuning. Lecture Notes in Computer Science, 13693, 709-727. https://doi.org/10.1007/978-3-031-19827-4_41

[22] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A video vision transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision, 6836-6846. https://doi.org/10.1109/ICCV48922.2021.00676

[23] Chen, C. F., Fan, Q., & Panda, R. (2021). CrossViT: Cross-attention multi-scale vision transformer for image classification. Proceedings of the IEEE/CVF International Conference on Computer Vision, 357-366. https://doi.org/10.1109/ICCV48922.2021.00042

[24] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11976-11986. https://doi.org/10.1109/CVPR52688.2022.01167

[25] Tan, M., & Le, Q. V. (2021). EfficientNet: Rethinking model scaling for convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(2), 610-624. https://doi.org/10.1109/TPAMI.2020.2991459

[26] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., & Xiao, B. (2021). Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(10), 3349-3364. https://doi.org/10.1109/TPAMI.2020.2983686

[27] Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A. C., & Li, Y. (2022). MaxViT: Multi-axis vision transformer. Lecture Notes in Computer Science, 13684, 459-479. https://doi.org/10.1007/978-3-031-20053-3_27

[28] Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., & Yan, S. (2022). MetaFormer is actually what you need for vision. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10819-10829. https://doi.org/10.1109/CVPR52688.2022.01055

[29] Tang, H., Liu, H., Xiao, W., & Sebe, N. (2022). Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. Neurocomputing, 472, 424-433. https://doi.org/10.1016/j.neucom.2021.12.005

[30] Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. Journal of Big Data, 8, Article 101. https://doi.org/10.1186/s40537-021-00492-0