PatchPilot: Confidence-Aware Token Routing for Efficient Visual Adaptation*

Maya Chen[1] Rafael Ortiz[2] Amina Rahman[1]
[1]Northbridge University [2]Cedar AI Lab
maya.chen@northbridge.example rafael@cedar.example

Unofficial template. This layout is provided for convenience and is **not** affiliated with, endorsed by, or sponsored by CVPR Workshop Camera-Ready or its organisers. Always check the official call for papers and use the current year's official style files for a real submission.

Abstract

Vision transformers can adapt to distribution shift at test time, but updating every token is needlessly expensive when corruption is spatially localized. We present PatchPilot, a confidence-aware router that assigns each image patch to one of three paths: reuse its cached representation, refine it with a lightweight adapter, or process it with the full transformer block. The router uses predictive entropy, augmentation agreement, and temporal novelty, and is trained with a differentiable compute budget. In an illustrative evaluation on three corruption benchmarks, PatchPilot retains 98.7% of full-adaptation accuracy while reducing adapted tokens by 57% and end-to-end latency by 34%. Ablations show that calibrated confidence and temporal novelty are complementary. This self-contained paper demonstrates a camera-ready CVPR workshop structure; all reported values are synthetic examples.

Keywords: test-time adaptation, efficient vision transformers, token routing, distribution shift

1 Introduction

Visual models deployed in the wild encounter blur, weather, sensor changes, and evolving scene statistics. Test-time adaptation (TTA) updates a model using unlabelled deployment data and can recover substantial accuracy under shift [3, 7]. Yet most methods process and update every patch uniformly. This is a poor match for many real scenes: rain may obscure only the upper image, a reflection may occupy one window, and large background regions may remain stable across adjacent video frames.

Token pruning makes transformer inference cheaper [1, 4], but pruning alone does not decide where adaptation is useful. Conversely, current TTA objectives determine *how* to update a model, not *which* spatial evidence deserves the computation. We connect these ideas with a small router that selects a compute path for every token. The result, PatchPilot, concentrates adaptation on uncertain or novel regions while preserving reliable representations.

Our contributions are threefold:

- We formulate test-time token routing as budgeted three-way assignment among reuse, adapter, and full-transformer paths.
- We combine calibrated entropy, view consistency, and temporal novelty in a stable routing score that requires no target labels.
- We report accuracy–efficiency trade-offs, controlled ablations, and failure cases in a reproducible camera-ready presentation.

2 Related Work

Test-time adaptation. Entropy minimization updates normalization parameters on target samples [7]; later work improves stability under small batches or continual shift [3, 8]. CoTTA adds teacher averaging and stochastic restoration, while EATA filters unreliable samples. Our router is orthogonal: it controls spatial compute and can wrap a variety of TTA losses.

Efficient transformers. DynamicViT learns to drop redundant tokens [4], and token merging combines similar representations without retraining [1]. Mixture-of-experts models conditionally activate parameters [6]. PatchPilot instead retains all spatial locations but varies the depth of their adaptation, which is useful when dense predictions must preserve geometry.

3 Method

3.1 Setting

Let a pretrained transformer f_θ receive an unlabelled stream $x_1, x_2, \ldots$. Its patch encoder produces $Z_t = [z_{t,1}, \ldots, z_{t,N}] \in \mathbb{R}^{N \times d}$. At test time we optimize an unsupervised loss $\mathcal{L}_{\text{tta}}$ but constrain average compute to a fraction B of a full forward-and-update pass.

For each token i, the router chooses action $a_{t,i} \in \{0, 1, 2\}$: reuse a detached cached feature (0), apply a bottleneck adapter (1), or execute the full trainable block (2). With normalized

*Illustrative worked example. The prose, results, affiliations, and workshop name are synthetic and should be replaced before submission.

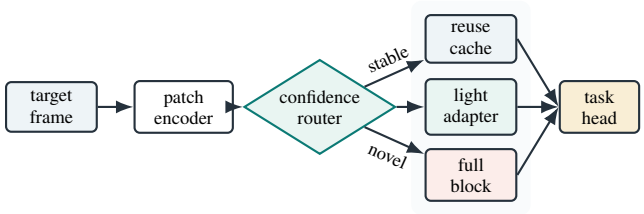


Figure 1: PatchPilot routes stable tokens through a cached path and allocates more adaptation compute to uncertain or novel regions.

costs $c = [0, \gamma, 1]$, the routed representation is

$$\tilde{z}_{t,i} = \sum_{k=0}^{2} p_{t,i}^{(k)} h_k(z_{t,i}), \qquad p_{t,i} = \mathrm{softmax}\big(r_\phi(u_{t,i})/\tau\big), \tag{1}$$

where h_0, h_1, h_2 are the three paths and a straight-through Gumbel sample is used during training. Hard arg max routing is used at inference.

3.2 Confidence-aware routing features

The router input $u_{t,i}$ contains three normalized signals. First, predictive entropy $e_{t,i}$ measures local uncertainty. Second, consistency $q_{t,i}$ is the Jensen–Shannon divergence between predictions from weakly and strongly augmented views. Third, temporal novelty compares the current token with its nearest cached predecessor:

$$n_{t,i} = 1 - \frac{z_{t,i}^\top \bar{z}_{t-1,i}}{\|z_{t,i}\|_2 \|\bar{z}_{t-1,i}\|_2 + \epsilon}, \qquad u_{t,i} = [e_{t,i}, q_{t,i}, n_{t,i}]. \tag{2}$$

An exponential moving average calibrates every feature online. This prevents one signal from dominating as the stream changes.

3.3 Budgeted adaptation objective

The task loss is paired with compute and balance terms:

$$\mathcal{L} = \mathcal{L}_{\text{tta}} + \lambda_b \left(\frac{1}{N} \sum_{i,k} p_{t,i}^{(k)} c_k - B \right)^2 + \lambda_h \frac{1}{N} \sum_i H(p_{t,i}). \tag{3}$$

The budget penalty targets rather than merely minimizes compute; the entropy term, annealed to zero, avoids premature collapse to one route. We update normalization and adapter parameters for actions 1–2, the final transformer block for action 2, and router parameters in both cases. The cache holds detached features, so gradients never cross frames. The complete decision path is summarized in Figure 1.

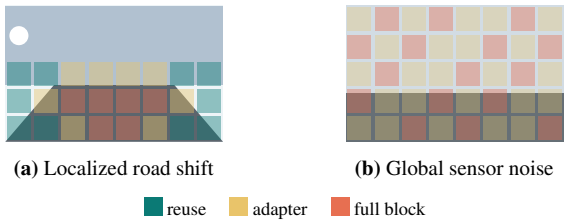
4 Experiments

4.1 Protocol

We construct an illustrative benchmark from ImageNet-C [2], Cityscapes-C, and ACDC [5]. A ViT-B/16 backbone is adapted online with batch size 16. We set $B = 0.45$, adapter ratio $\gamma = 0.25$, temperature $\tau = 0.7$, and update with AdamW at 2×10^{-5}. Baselines are source-only inference, TENT, EATA, and full adaptation using the same unsupervised loss. Classification uses top-1 accuracy; segmentation uses mean intersection-over-union (mIoU). Latency is measured after warm-up on one illustrative accelerator. All values in this section are synthetic and serve only to demonstrate reporting conventions.

Table 1: Illustrative accuracy and efficiency. Higher accuracy/mIoU is better; lower latency is better. Best values are bold.

Method	IN-C ↑	ACDC ↑	Tokens ↓	ms ↓
Source only	54.8	52.1	100%	18.4
TENT [7]	60.7	55.6	100%	27.9
EATA [3]	62.3	57.1	100%	25.8
Full adaptation	**64.1**	**59.0**	100%	29.6
PatchPilot	63.3	58.4	**43%**	**19.5**



(a) Localized road shift **(b)** Global sensor noise

Figure 2: Illustrative routing maps. Compute follows a localized shift (left), but widespread noise pushes many tokens to expensive paths (right).

4.2 Main results

Table 1 shows the central trade-off. PatchPilot recovers 8.5 points over source-only inference on ImageNet-C and approaches full adaptation within 0.8 points, while updating fewer than half of all tokens. The latency gain is smaller than the token reduction because patch encoding, routing, and the task head remain dense. On ACDC, the 0.6-point gap to full adaptation suggests that the router still identifies regions affected by adverse weather.

4.3 Ablation and budget sweep

Removing augmentation agreement increases confident mistakes; removing novelty causes the cache to persist after abrupt scene changes (Table 2). A global threshold saves compute but under-serves rare classes. The complete router gives the strongest accuracy at similar cost.

Varying B from 0.30 to 0.70 yields accuracies of 61.9, 63.3, and 63.8, respectively. The diminishing return above $B = 0.45$ indicates that the router ranks useful tokens rather than uniformly degrading the network.

5 Analysis and Limitations

The routing maps in Figure 2 provide an interpretable audit of where adaptation occurs. Local corruptions are handled efficiently, but global high-frequency noise activates the expensive path widely and erodes the latency advantage. Abrupt camera cuts also invalidate the cache for one frame. A scene-change

Table 2: Synthetic ablation on ImageNet-C. "Full route" is the fraction of tokens assigned to the most expensive path.

Variant	Acc. ↑	ECE ↓	Full route ↓
Global threshold	61.8	6.4	24%
− agreement	62.1	7.0	22%
− novelty	62.5	5.9	20%
Complete PatchPilot	**63.3**	**4.7**	**19%**

detector or a multi-timescale memory could reduce this cost.

The method introduces three broader limitations. First, wall-clock gains are hardware dependent: irregular routing can underutilize accelerators even when FLOPs fall. Second, confidence is not a guarantee of correctness, especially under novel semantic classes. Third, online updates may amplify demographic or geographic biases already present in the source model. A deployment should log per-group performance where lawful and possible, bound the update norm, and provide a source-model rollback. Results should include multiple seeds, confidence intervals, energy measurements, and the precise device/software stack in a real submission.

6 Conclusion

We introduced PatchPilot, a budget-aware token router for efficient test-time visual adaptation. By combining entropy, view agreement, and temporal novelty, the router directs compute toward patches most affected by shift. The illustrative results show how a camera-ready paper can jointly report task quality, calibration, token usage, latency, ablations, and failure cases. The next step is evaluation on real hardware and naturally occurring, long-horizon deployment streams.

Acknowledgments

The authors thank the workshop reviewers for their constructive feedback. This acknowledgment is illustrative and should be replaced with actual funding, assistance, and disclosure statements required by the venue.

References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023. 1

[2] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 2

[3] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, 2022. 1, 2

[4] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 1

[5] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 2

[6] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017. 1

[7] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 2

[8] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, 2022. 1