WHERE TO ADAPT MATTERS: ONLINE DEPTH-WISE ROUTING FOR TEST-TIME ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-Time Adaptation (TTA) adapts a deployed model during online inference to mitigate the impact of domain shift. Existing methods concentrate on designing *what* to optimize and *how* to regularize the update, yet *where* to update is fixed in advance: they adapt a parameter subset chosen before deployment. We show that no single fixed choice covers diverse test conditions well: the preferred adaptation depth varies across settings, for example with the test batch size, where normalization groups at different depths take turns being optimal. Based on this insight, we propose Adaptive Group-wise Online Experts (AGOE), which turns *where* to update into an online decision. AGOE partitions normalization parameters into depth-wise experts and reallocates a fixed update budget among them at each step via per-depth learning-rate scaling, guided by a label-free reliability signal and a Hedge-style online rule. Routing preserves the base TTA objective, adds no extra backward pass, and acts as a plug-in complement to existing adapters. Experiments show that AGOE consistently improves its base adapters, especially in regimes where fixed-depth adaptation degrades severely, such as small-batch adaptation.

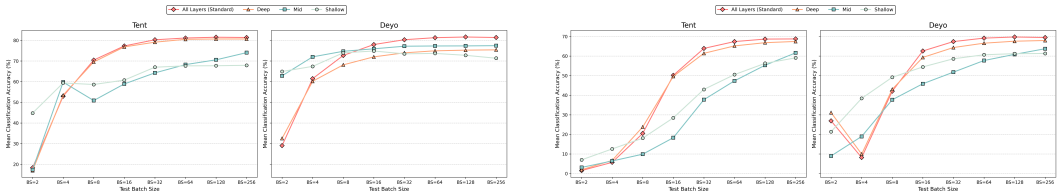
1 INTRODUCTION

Deep neural networks often degrade under distribution shifts between training and deployment (He et al., 2016; Hendrycks & Dietterich, 2019). Test-time adaptation (TTA) addresses this problem by updating a deployed model from unlabeled test batches arriving online, without source data or labels. This setting is doubly constrained. The model receives no supervision at test time, so every update must rely on self-derived signals that may themselves be wrong. Meanwhile, batches arrive one at a time and are often small, so the statistics behind each update are noisy and never settle into the large-sample regime that supervised training enjoys. Even when *what* to follow and *how* to stabilize are fixed, every update must still decide *where* to spend the limited update budget, since no depth can be trusted unconditionally: *what* signal to follow when no ground truth exists, *how* to keep the update stable when batches are small and noisy, and *where* to allocate updates across depths.

Existing methods focus primarily on the first two. *What* to optimize is defined by self-supervised proxies: entropy minimization (Wang et al., 2021), auxiliary test-time-training losses (Sun et al., 2020; Liu et al., 2021), and contrastive or prediction-diversity objectives (Chen et al., 2022; Zhang et al., 2022; Zhao et al., 2023). *How* to update stably is addressed by filtering or reweighting unreliable samples (Niu et al., 2022; 2023), teacher–student smoothing and model resetting (Wang et al., 2022; Lim et al., 2023), and memory-economic or sharpness-aware updates (Hong et al., 2023; Gong et al., 2023). Unlike *what* and *how*, *where* admits no direct self-supervised signal, since the loss does not reveal which depth should move. The closest line is layer selection, notably GALA (Sahoo et al., 2026), which scores layers by update–trajectory alignment but commits to a fixed subset once; it does not revisit the decision as the stream evolves.

Freezing the update location is a deliberate safeguard. Updating every parameter is unstable on small, noisy batches, so efficient TTA restricts adaptation to normalization layers and keeps that choice static (Ioffe & Szegedy, 2015).The cost is that adaptation depth becomes a static hyperparameter, even though the preferred location varies with test conditions. Depths behave differently: shallow layers retain more generic features while deeper layers are more task-specific (Yosinski et al., 2014; Neyshabur et al., 2020), so the reliable depth shifts as the test regime changes. As test

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) CIFAR-10-C (b) CIFAR-100-C

Figure 1: **Static adaptation depths show batch-size-dependent cross-over behavior.** Different normalization-depth choices achieve the best accuracy under different test batch sizes, indicating that no single static depth is consistently optimal.

conditions change, so should the allocation of updates across depths. We show that a single frozen choice cannot cover diverse test conditions well: normalization groups at different depths take turns being optimal across settings, for example as the test batch size changes (Figure 1 and App. Figs. 9–10), so no static depth is uniformly reliable. The next section quantifies this cross-over and traces its mechanism.

We therefore propose *Adaptive Group-wise Online Experts* (AGOE), which turns *where* to update into an online decision. AGOE partitions normalization parameters into depth-wise experts, allocates a fixed update budget among them at each step, and refines the allocation with a Hedge-style rule driven by update–trajectory alignment, a label-free measure of update consistency. Unlike one-time selection, AGOE revisits the allocation at every step, which matters when the reliable depth itself drifts. Routing is realized as per-depth learning-rate scaling: the optimizer holds one parameter group per expert, and the routing distribution scales each group's learning rate. This requires only one backward pass per test batch, preserves the base TTA objective, and plugs into existing adapters such as Tent (Wang et al., 2021), DeYO (Lee et al., 2024), and ZeroSiam (Chen et al., 2026).

We validate AGOE on CIFAR-10-C and ImageNet-C across batch-size regimes and online shift scenarios. For example, at BS=2 the fixed-depth Tent baseline collapses to 18.34%, while AGOE restores it to 57.71%. Routing visualizations further show that allocation changes across corruption types and batch sizes rather than collapsing to a single expert, supporting depth-wise routing as a complement to objective design and sample filtering. The routing rule further enjoys a static-regret guarantee against the best fixed depth in hindsight, which justifies it as a stable allocator when the reliable depth is unknown in advance.

Contributions. 1) We identify fixing *where* to update as an overlooked decision in current TTA, as evidenced by batch-size-dependent cross-over among static adaptation depths. 2) We propose AGOE, an online routing mechanism that allocates a fixed update budget across depth-wise normalization experts, with a label-free alignment signal, a Hedge-style update, and a static-regret guarantee for the routing rule. 3) We show AGOE is a lightweight plug-in for multiple TTA objectives requiring one backward pass per batch, with gains concentrated where fixed-depth adaptation degrades while remaining at parity when it is already stable.

2 Where to Update is a Stream-Dependent Decision

Although contemporary TTA methods have improved *what* to optimize and *how* to stabilize updates, they still treat *where* to update as an overlooked decision fixed before deployment. This motivates our central question: does a single depth provide a uniformly reliable place to update, or is the best depth determined by the stream itself?

Observation 1: No static depth wins everywhere. To answer this, we compare four static policies that differ only in update location: updating only the shallow, middle, or deep normalization groups, or all of them uniformly. The three groups are obtained by splitting normalization layers in depth order into three contiguous chunks of roughly equal size; three is the coarsest resolution that still separates early, intermediate, and late stages while keeping enough parameters per group for stable statistics. We evaluate on CIFAR-10-C and CIFAR-100-C across batch sizes and average over 15 corruptions, resetting to the source model for each corruption to isolate the effect of depth.

As shown in Figure 1 (and App. Figs. 9–10 for per-dataset views), a consistent batch-size-dependent cross-over emerges across adapters. (i) *Small batches:* shallow updates dominate while deep and middle policies collapse (on CIFAR-10-C at BS=2, shallow-only 44.79% vs. deep-only 16.99% and all-layer 18.34%). (ii) *Large batches:* deep and all-layer updates overtake shallow ones. (iii) *No safe middle ground:* the all-layer ranking tracks the deep group across the sweep, not the compact shallow group. The ordering is consistent across both datasets. Hence *where* to update is a stream-dependent decision, and any fixed choice pays a price at the opposite end of the batch-size axis.

Observation 2: Noisy depths reveal themselves through inconsistent directions. Why does the preferred depth flip? Small-batch gradients spread widely in deep and all-layer groups with many outliers, while shallow gradients stay compact; the dispersion gap shrinks as batches grow (Figure 2). This is heterogeneous noise across depths (i.e., noise levels differ across depths): when evidence is scarce, deep layers amplify it, while shallow layers remain stable; when evidence is abundant, all depths stabilize. This matches observations that shallow layers encode more generic features while deeper layers specialize (Yosinski et al., 2014). At small batches, updating only the shallow group avoids the noisy depths, so shallow wins; as batches grow, all groups concentrate and noise ceases to be the bottleneck, so the expressiveness of deep and all-layer updates takes over.

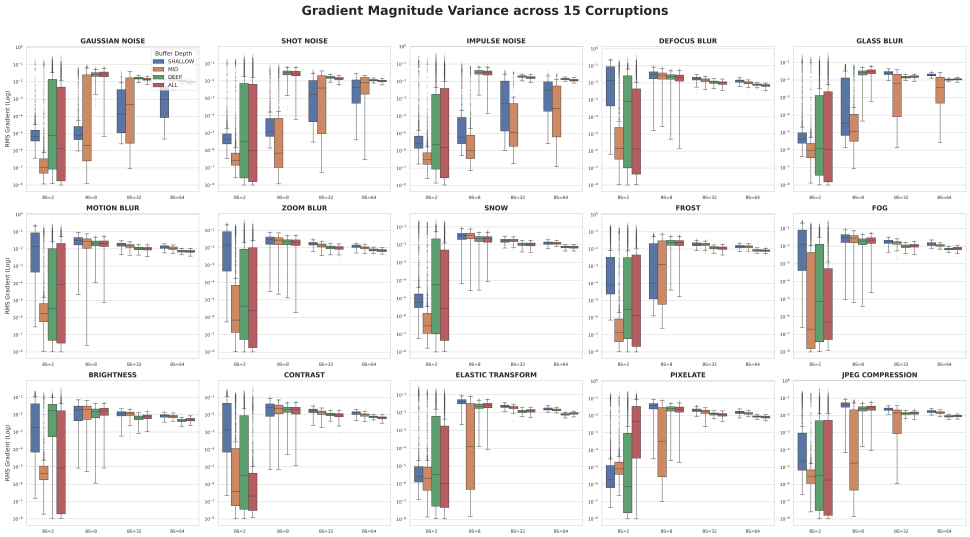


Figure 2: **RMS gradient magnitude distributions across corruption types and batch sizes.** Under small batches, deep and all-layer updates exhibit larger dispersion and more outliers than shallow updates. The all-layer distribution often follows the behavior of the deep group, indicating that updating all layers can inherit deep-layer instability rather than acting as a stable compromise.

The two observations point to adaptive group-wise allocation: the allocator must act per depth, redecide as the stream evolves, and work without labels. Moreover, since uniform updating resembles the noisiest depth rather than averaging across depths, the problem is closer to isolating the unreliable depths at each step than to tuning a global step size.

3 METHODOLOGY

The preceding observations call for per-step isolation of unreliable depths, but two difficulties stand in the way. First, the reliability of each depth must be estimated at every step without labels and from small, noisy batches. Second, the allocation must track the stream without oscillating wildly, and without a second backward pass. AGOE addresses both with three coupled choices: an update–trajectory alignment signal that is label-free and available from quantities already in the optimizer, a Hedge-style rule that revises the allocation smoothly, and per-depth learning-rate scaling that real-

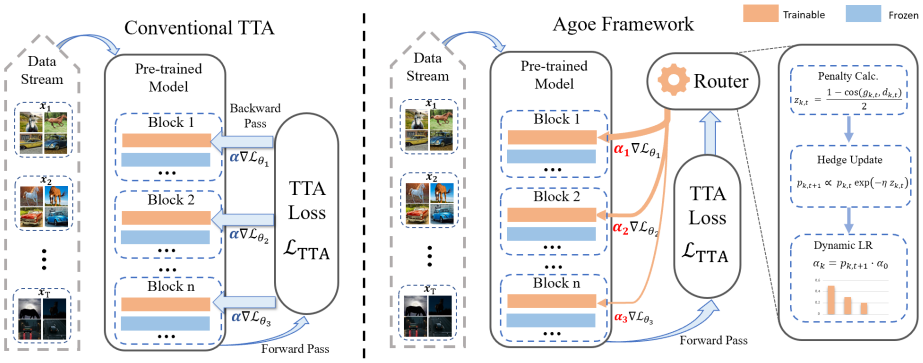


Figure 3: **Overview of AGOE.** For each unlabeled test batch, AGOE uses the current routing distribution to set per-depth learning rates for depth-wise normalization experts, computes the base TTA loss with one backward pass, measures depth-wise reliability via update–trajectory alignment, and updates the routing weights with a Hedge rule for the next step.

izes routing with a single backward pass per batch. Figure 3 summarizes the pipeline; Algorithm 1 gives the exact step order.

AGOE operates on top of a frozen pretrained model $f_{\boldsymbol{\theta}}$ with an unsupervised base loss $\mathcal{L}_t$ on test batch t. At step t it uses the current routing distribution $\mathbf{p}_t$ to set per-expert learning rates, computes $\mathcal{L}_t$ with one backward pass to obtain both the model update and the routing signal, applies the routed optimizer step, and revises routing to $\mathbf{p}_{t+1}$ for the next batch. AGOE complements base methods as a plug-in router over depth: it preserves the base loss, sample filtering, and predictor updates. Routing never changes the base objective; since $\sum_k \alpha_{k,t} = \alpha_0$ always holds, it only changes how much of the fixed budget each depth receives, i.e., a zero-sum reallocation rather than extra updates.

3.1 Depth Experts and Update Budget

Let Θ_{norm} be the trainable affine parameters of normalization layers. AGOE sorts normalization layers by depth and splits them uniformly into K contiguous, disjoint groups of roughly equal size, each acting as a depth-wise expert: the first group covers early layers, the last covers late layers, and no layer belongs to two groups. With $K = 1$ this reduces to uniform all-layer updating; larger K gives finer depth resolution. Each normalization layer keeps its own affine parameters: experts share only the learning-rate multiplier, not parameters.

AGOE maintains a routing distribution $\mathbf{p}_t = (p_{1,t}, \ldots, p_{K,t}) \in \Delta_K$, where $p_{k,t}$ is the fraction of the update budget assigned to expert k at step t. Routing is implemented by scaling per-expert learning rates,

$$\alpha_{k,t} = \alpha_0 p_{k,t}, \tag{1}$$

with one optimizer parameter group per expert, so routing only rescales step sizes. with a fixed total budget α_0. K sets depth resolution; finer partitioning fragments the fixed budget and makes per-expert reliability estimates noisier, so the optimal granularity depends on gradient reliability rather than increasing monotonically with K. We initialize $\mathbf{p}_1$ uniformly so each group starts from the same per-group rate as the corresponding fixed-depth baseline.

3.2 Alignment Signal

Gradient magnitude alone cannot drive routing: a large step may carry a strong adaptation signal or pure batch noise. The distinguishing feature is directional. After computing the base loss $\mathcal{L}_t$ and one backward pass, let $\mathbf{g}_{k,t} = \nabla_{\boldsymbol{\theta}_{k,t}} \mathcal{L}_t$ and let the proposed direction be $\mathbf{u}_{k,t} = -\mathbf{g}_{k,t}$ under SGD. Let $\mathbf{d}_{k,t} = \boldsymbol{\theta}_{k,t} - \bar{\boldsymbol{\theta}}_k$ be the displacement from the stream anchor $\bar{\boldsymbol{\theta}}_k$, maintained as an EMA of $\boldsymbol{\theta}_k$ (decay 0.9). Both the anchor and $\mathbf{p}_t$ are reset at each corruption. AGOE scores each expert by

$$z_{k,t} = \frac{1 - \cos(\mathbf{u}_{k,t}, \mathbf{d}_{k,t})}{2}, \tag{2}$$

Algorithm 1: Adaptive Group-wise Online Experts (AGOE)

Input: Model $f_{\boldsymbol{\theta}}$; test stream $\{x_t\}_{t=1}^{T}$; base loss $\mathcal{L}_t$; total budget α_0; Hedge rate η; depth groups $\{G_k\}_{k=1}^{K}$

Freeze non-normalization parameters and partition Θ_{norm} into depth experts;
Initialize $p_{k,1} = 1/K$ and anchor $\bar{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_{k,1}$ for each expert k;
for $t = 1$ **to** T **do**
    Set expert learning rates by Eq. 1;
    Compute $\mathcal{L}_t(\boldsymbol{\theta}_t; x_t)$ and backpropagate once to obtain $\mathbf{g}_{k,t}$;
    Compute penalties $z_{k,t}$ by Eq. 2 from the current gradients;
    Update model parameters with the current expert learning rates;
    Update routing weights by Eq. 3 for the next step;
end

with $z_{k,t} = 0.5$ when either vector has zero norm. Small $z_{k,t}$ means the proposed update continues the trajectory the expert has been following; large $z_{k,t}$ flags an abrupt directional break. The same alignment idea underlies recent layer-selection work (Sahoo et al., 2026). GALA uses alignment as a one-time score to freeze a subset, assuming the reliable depth stays put. AGOE uses the same signal as per-step feedback because Obs. 2 shows the noisy depth drifts: selection answers which depth, allocation answers how much at each step. The signal is label-free, available at every step, per-depth, and costs no extra backward pass.

3.3 Hedge routing

Given $\mathbf{z}_t$, AGOE revises routing with the Hedge rule for prediction with expert advice (Freund & Schapire, 1997; Cesa-Bianchi & Lugosi, 2006):

$$p_{k,t+1} = \frac{p_{k,t} \exp(-\eta z_{k,t})}{\sum_{j=1}^{K} p_{j,t} \exp(-\eta z_{j,t})}, \tag{3}$$

where η is the Hedge rate. Poorly aligned experts are multiplicatively down-weighted while stable ones gain relative share, and no expert is ever permanently fixed as the sole update location.

Proposition 1 (Static regret of AGOE)**.** *Assume $z_{k,t} \in [0, 1]$ for all k and t. Updating $\mathbf{p}_t$ by Eq. 3 gives, for any $T \geq 1$,*

$$\sum_{t=1}^{T} \langle \mathbf{p}_t, \mathbf{z}_t \rangle - \min_k \sum_{t=1}^{T} z_{k,t} \leq \frac{\log K}{\eta} + \frac{\eta T}{8}, \tag{4}$$

so $\eta = \sqrt{8 \log K / T}$ yields $\mathcal{O}(\sqrt{T \log K})$ regret.

This bounds cumulative directional inconsistency against the best fixed depth in hindsight: no matter how the per-step scores fluctuate, average routing is almost as consistent as the single best depth chosen with full knowledge of the stream. It does not imply accuracy improvement; it justifies Hedge as a stable allocator when the reliable depth is unknown in advance. The bound applies whenever scores lie in $[0, 1]$, which holds for cosine-based penalties by construction, and its proof is given in App. C. Accuracy gains are established empirically in Sec. 4.2 and Sec. 4.4.

Each batch costs one base-model backward pass plus K vector cosine evaluations and K scalar weight updates, negligible next to the model pass. Routing is soft: the full budget is shared across experts according to $\mathbf{p}_t$. AGOE applies to TTA objectives that (i) update a partitionable normalization subset, (ii) provide a batch-level loss or equivalent update direction, and (iii) admit per-group learning rates; Tent, DeYO, and ZeroSiam satisfy this interface. Full-network teacher–student schemes and objectives with tightly coupled cross-layer statistics need additional adaptation and are outside the present scope.

4 Experiments

Questions addressed by the experiments. Our experiments answer three questions: (i) whether AGOE resolves the static-depth trade-off across batch-size regimes, (ii) whether the benefit persists

under harder online shifts, and (iii) whether the mechanism and compute profile support the practical value of the method.More results are in the appendix.

4.1 Experimental setup

Benchmarks, models, and baselines. We evaluate AGOE on CIFAR-10-C, CIFAR-100-C, and ImageNet-C (Hendrycks & Dietterich, 2019): 15 corruption types in noise/blur/weather/digital groups, severity 5 unless stated, top-1 accuracy (%). WRN28 for CIFAR, AugMix ResNeXt for CIFAR-100-C, ResNet-50-GN for ImageNet-C. Baselines: source, Tent (Wang et al., 2021), DeYO (Lee et al., 2024), ZeroSiam (Chen et al., 2026), EATA (Niu et al., 2022), SAR (Niu et al., 2023), CoTTA (Wang et al., 2022), GALA (Sahoo et al., 2026); AGOE instantiated with Tent, DeYO, ZeroSiam. All results averaged over 3 runs with different seeds.

Online adaptation protocol and implementation fairness. All methods share the backbone, benchmark, corruption order, reset policy, batch size, and seeds within each setting; severity 5 unless stated. In the standard setting adaptation resets per corruption; label-shift uses BS=32 with imbalance ratio ∞; continuous streams use no resets.

Depth partition and learning-rate budget. Normalization layers are sorted by depth and split uniformly into K contiguous groups with approximately equal numbers of layers (default $K = 3$: shallow/middle/deep); only trainable affine parameters are updated. WRN28-10 is used for CIFAR-10-C and split by its three blocks; the AugMix ResNeXt backbone is used for CIFAR-100-C and split by its three stages; ResNet-50-GN is used for ImageNet-C. Baselines apply learning rate 0.0025 to every group, so with K groups the summed budget is $K \times 0.0025$. AGOE sets total budget $\alpha_0 = K \times 0.0025$ with $\alpha_{k,t} = \alpha_0 p_{k,t}$; since the routing weights sum to one, the K group rates always sum to the same $K \times 0.0025$ as the baseline, i.e., the $K\times$ factor only matches the K groups and leaves the summed budget unchanged.

4.2 Main effectiveness

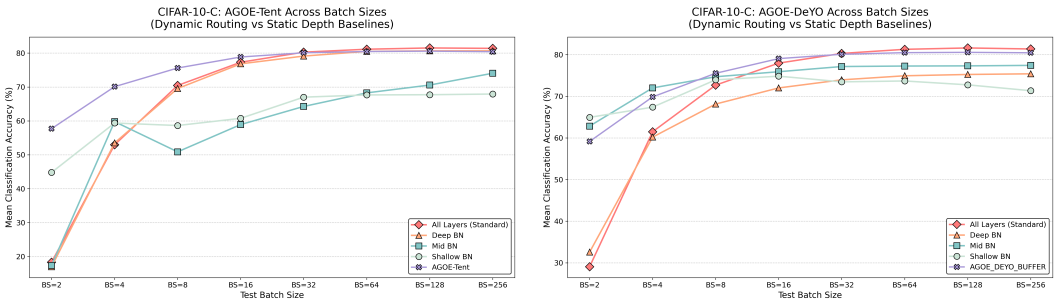
Main result: resolving static-depth trade-off across batch sizes. We first test whether online update-budget allocation resolves the static-depth cross-over in Figure 1. Figure 4 reports mean accuracy across test batch sizes, averaged over 15 corruptions with a separate reset for each corruption. In Figure 4a, "AGOE" specifically denotes AGOE instantiated with Tent (AGOE-Tent). AGOE-Tent performs competitively or better than fixed-depth Tent baselines across batch sizes, and the same routing principle also improves DeYO, showing that depth-wise routing can complement sample-selection and sample-weighting based TTA. Table 1 summarizes the cross-batch behavior on CIFAR-10-C (mean over 15 corruptions). No static depth wins across batch sizes: shallow leads at BS2 while all/deep lead at large batches. AGOE follows the favorable envelope in both families, achieving the highest average (Tent 73.80, DeYO 74.06) and the strongest worst-case among its family, which supports using one routing configuration across batch regimes.

Table 1: Cross-batch summary on CIFAR-10-C (mean acc. %).

BS	2	4	8	16	32	64	Avg	Worst
• All-Tent	18.34	52.95	70.45	77.29	**80.27**	**81.13**	63.41±0.31	18.34
• Deep-Tent	16.99	53.50	69.54	76.82	79.11	80.49	62.74±0.28	16.99
• Mid-Tent	17.39	59.78	50.87	58.91	64.26	68.26	53.25±0.35	17.39
• Shallow-Tent	44.79	59.37	58.64	60.75	67.00	67.66	59.70±0.26	44.79
• AGOE-Tent	**57.71**	**70.09**	**75.59**	**78.84**	80.08	80.46	**73.80±0.22**	**57.71**
• All-DeYo	29.05	61.48	72.69	77.97	**80.30**	**81.29**	67.13±0.24	29.05
• Deep-DeYo	32.56	60.19	68.14	72.01	73.96	74.93	63.63±0.27	32.56
• Mid-DeYo	62.84	**72.02**	74.70	75.91	77.16	77.27	73.32±0.21	62.84
• Shallow-DeYo	**64.94**	67.40	73.96	74.84	73.48	73.68	71.38±0.23	**64.94**
• AGOE-DeYo	59.16	69.83	**75.42**	**79.17**	80.22	80.54	**74.06±0.20**	59.16

Representative gains under small-batch adaptation (BS=2). We next evaluate AGOE on ImageNet-C at severity level 5 with ResNet-50-GN under BS=2. Table 2 shows that AGOE improves Tent, DeYO, and ZeroSiam while leaving their base objectives unchanged, isolating the effect of depth-wise update allocation. AGOE-DeYO improves DeYO from 36.07% to 38.08%, and AGOE-ZEROSIAM improves ZEROSIAM from 43.40% to 45.96%, achieving the best average accuracy.

We also evaluate AGOE on CIFAR-10-C at severity level 5 under BS=2. Table 3 shows that AGOE improves multiple base adapters by changing where the update budget is applied, not the under-



(a) AGOE-Tent (b) AGOE-DeYO

Figure 4: **Adaptive expert routing improves robustness across test batch sizes.** (a) The curve labeled "AGOE" denotes AGOE-Tent. (b) The same routing principle complements DeYO.

Table 2: Comparisons with baselines on ImageNet-C at severity level 5 under the small-batch setting (BS=2). Results are reported as accuracy (%), and the **bold** value signifies the top-performing result.

	Noise			Blur				Weather				Digital				
Batch Size 2	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet-50-GN (Source)	22.28	22.96	22.18	19.50	11.58	21.80	25.38	40.16	46.60	33.10	68.68	36.14	18.24	29.88	52.16	31.38
• EATA	28.16	33.30	33.32	16.78	19.70	29.66	29.86	43.68	42.18	44.72	70.86	45.68	27.68	49.50	55.72	38.05
• SAR	30.60	33.38	31.00	18.48	15.20	29.70	33.20	45.44	43.98	17.52	72.22	46.00	9.42	49.92	**56.14**	35.48
• CoTTA	29.94	35.04	31.52	16.40	16.64	24.52	27.98	42.16	47.20	49.20	68.26	16.56	20.38	51.20	55.94	35.53
• GALA	26.86	28.34	27.96	15.42	15.74	22.82	25.82	35.56	38.24	11.42	69.64	40.20	16.36	43.00	53.90	31.42
• Tent	4.48	5.62	9.08	10.58	13.72	8.74	15.98	11.54	8.60	1.36	69.22	38.14	4.90	42.02	51.38	19.69
• AGOE-Tent	31.72	36.62	33.84	14.98	20.12	20.34	26.02	16.76	19.80	2.14	71.66	44.48	6.70	51.80	55.34	30.15 (+10.46)
• DeYO	37.66	38.80	36.26	8.58	15.86	37.84	19.82	49.98	32.86	2.96	72.34	49.54	23.52	58.26	56.78	36.07
• AGOE-DeYO	38.30	39.92	37.86	12.12	18.24	36.56	20.00	50.86	39.24	2.18	72.14	48.02	43.42	57.06	55.22	38.08 (+2.01)
• ZEROSIAM	36.32	38.70	37.44	24.46	27.62	34.30	38.32	47.90	44.62	54.44	71.72	47.46	38.02	54.26	55.44	43.40
• AGOE-ZEROSIAM	**39.12**	**41.86**	**39.44**	**29.44**	**28.90**	**37.34**	**41.32**	**51.62**	**48.14**	**55.48**	**72.76**	**48.76**	**43.14**	**56.02**	**56.08**	**45.96** (+2.56)

lying loss. AGOE-Tent improves Tent from 18.35% to 57.71%, AGOE-DeYO improves DeYO from 29.18% to 59.15%, and AGOE-ZEROSIAM achieves the best average accuracy of 64.87%. AGOE-Tent (57.71%) also surpasses one-time selection GALA (51.82%) by 5.89%, showing per-step reallocation helps beyond a frozen subset.

Table 3: Comparisons with baselines on CIFAR-10-C at severity level 5 under the small-batch setting (BS=2). Results are reported as accuracy (%), and the **bold** value signifies the top-performing result.

	Noise			Blur				Weather				Digital				
Batch Size 2	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
WRN28 (Source)	27.67	34.27	27.08	53.05	45.67	65.24	57.99	74.93	58.70	73.98	90.69	53.29	73.42	41.55	69.70	56.48
• EATA	48.75	48.60	40.39	63.47	35.59	62.09	63.81	60.25	60.17	64.92	66.29	70.66	40.54	54.75	47.09	55.56
• SAR	52.03	53.78	43.38	68.10	39.36	68.24	67.40	65.39	63.93	68.39	74.36	74.22	51.40	59.50	50.88	60.42
• CoTTA	53.37	55.22	46.73	68.20	44.97	68.47	67.61	63.97	65.05	67.77	74.29	73.82	55.50	60.46	53.09	61.23
• GALA	47.13	51.86	39.54	65.23	31.46	64.34	65.72	37.05	52.62	65.05	65.84	56.02	39.82	50.43	45.14	51.82
• Tent	16.08	17.30	17.45	20.69	15.59	19.51	18.41	18.88	16.89	23.54	22.34	16.20	17.75	18.70	15.71	18.35
• AGOE-Tent	52.74	52.17	44.03	67.66	35.71	66.88	67.54	58.29	63.24	67.08	75.19	63.21	45.04	58.11	48.82	57.71 (+39.36)
• DeYO	16.87	18.79	16.09	35.22	16.80	29.76	28.30	22.92	33.34	36.52	53.22	49.43	22.76	29.38	26.29	29.18
• AGOE-DeYO	52.99	55.51	32.72	69.64	28.05	68.89	68.57	65.22	61.47	71.21	76.07	76.18	45.08	62.25	53.34	59.15 (+29.97)
• ZEROSIAM	56.78	59.19	49.87	70.72	48.06	70.45	69.93	66.64	66.90	70.54	76.04	75.99	57.71	63.54	56.94	63.95
• AGOE-ZEROSIAM	**58.44**	**59.81**	**51.11**	**70.92**	**48.86**	**70.91**	**70.69**	**67.60**	**67.89**	**71.38**	**76.41**	**76.63**	**58.03**	**65.14**	**59.27**	**64.87** (+0.92)

4.3 Robustness under online shifts

Robustness under online imbalanced label shifts. We evaluate online imbalanced label shifts on ImageNet-C at severity level 5 and BS=32 using the SAR protocol with imbalance ratio ∞. This setting tests whether update allocation remains reliable under an imbalanced class prior. Table 4 reports $T = 10000$ online adaptation steps. For label shift, we follow SAR with imbalance ratio ∞ and report $T = 10000$ in the main text, with longer streams deferred to App. D.2. AGOE-DeYO improves DeYO from 32.89% to 34.43%, and AGOE-ZEROSIAM improves ZEROSIAM from 33.20% to 33.86%, indicating that routing can help adapters sensitive to online drift.

Robustness in continuously changing environments (CTTA). We also evaluate continuously changing environments on ImageNet-C at severity level 5. This setting removes corruption-wise resets, allowing adaptation errors to accumulate across domains. AGOE shows substantially larger gains here than in reset-based settings (Tent: 3.29%→10.17%, DeYO: 11.10%→15.61%, ZeroSiam: 12.72%→37.83%), suggesting that online budget allocation is especially beneficial when long-horizon errors compound; for continuous adaptation, detailed ImageNet-C results are in App. D.3.

Table 4: Comparisons with baselines on ImageNet-C at severity level 5 under online imbalanced label shifts (imbalance ratio = ∞). $T = 10000$ online adaptation steps.

	Noise			Blur				Weather				Digital				
Label Shift	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet-50-GN (Source)	22.59	23.27	22.89	16.54	10.86	18.35	24.60	41.51	**50.41**	34.67	**69.88**	38.52	17.96	29.15	53.65	31.66
• EATA	23.15	25.97	26.23	13.75	12.74	20.48	26.70	44.44	41.56	34.39	69.35	40.49	25.55	31.06	55.70	32.77
• SAR	23.16	25.12	25.24	**16.37**	11.76	18.42	25.92	41.30	47.16	33.58	68.52	39.96	20.40	32.65	54.65	32.28
• CoTTA	22.08	23.60	23.19	16.45	10.10	17.60	24.22	41.31	50.62	31.00	69.54	39.15	17.97	30.29	55.16	31.49
• Tent	23.13	24.87	23.66	13.42	12.06	18.51	24.73	40.70	45.15	9.88	69.11	40.52	19.91	39.44	55.13	30.68
• AGOE-Tent	22.81	25.05	24.73	13.92	12.61	18.87	25.06	41.30	45.13	11.98	68.90	41.19	21.27	41.05	54.72	31.24 (+0.56)
• DeYO	22.24	25.55	**34.90**	14.40	12.25	**23.11**	**27.31**	43.93	42.32	9.76	68.81	**42.11**	23.59	**46.33**	56.77	32.89
• AGOE-DeYO	22.65	25.76	34.29	**16.24**	14.04	22.60	25.87	**44.66**	42.42	29.26	69.57	41.20	**26.81**	44.20	**56.95**	**34.43** (+1.54)
• ZEROSIAM	24.96	25.95	27.28	15.32	12.26	20.00	25.89	42.02	45.65	33.13	69.11	40.93	21.03	39.28	55.13	33.20
• AGOE-ZEROSIAM	**25.07**	**27.30**	28.03	15.54	**14.20**	21.01	25.96	41.92	45.45	**34.69**	69.35	41.46	22.35	40.74	54.86	33.86 (+0.66)

Table 5: Comparisons on ImageNet-C at severity level 5 under continuously changing environments.

t ⟶

	Noise			Blur				Weather				Digital				
CTTA (continuous environment)	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
• SAR	31.88	42.32	44.11	16.01	19.49	10.51	2.53	0.97	33.44	**50.75**	**70.37**	**46.73**	15.34	13.08	6.66	26.95
• CoTTA	22.26	23.35	22.27	**18.30**	16.33	22.47	26.99	30.02	**41.73**	27.72	59.43	33.56	22.44	**42.80**	**48.84**	30.57
• Tent	24.09	5.03	0.69	5.74	1.28	0.72	0.98	1.55	0.36	0.40	4.60	0.94	0.11	0.27	2.54	3.29
• AGOE-Tent	30.77	38.93	39.67	11.61	15.07	10.72	3.77	0.87	0.19	0.12	0.40	0.12	0.12	0.12	0.12	10.17 (+6.88)
• DeYO	40.83	48.52	47.93	7.15	9.29	6.62	3.08	1.00	0.47	0.36	0.45	0.22	0.18	0.18	0.20	11.10
• AGOE-DeYO	**41.81**	**49.50**	**49.30**	15.08	**20.61**	**26.08**	29.20	1.68	0.11	0.11	0.15	0.13	0.13	0.09	0.09	15.61 (+4.51)
• ZEROSIAM	32.96	38.79	36.04	14.26	12.75	16.19	19.34	19.70	0.10	0.10	0.10	0.10	0.10	0.10	0.10	12.72
• AGOE-ZEROSIAM	34.15	43.72	44.80	14.78	19.81	24.22	**30.09**	**39.72**	41.14	49.91	69.30	42.34	**23.15**	40.49	49.80	**37.83** (+25.11)

4.4 Mechanism and Practicality

Hyperparameter sensitivity of dynamic budget allocation. We analyze sensitivity to the base learning-rate budget α_0 and routing scale η on ImageNet-C (severity 5), as shown in Figure 5. AGOE remains competitive across a broad region rather than a single isolated setting, indicating that performance is not fragile to small hyperparameter perturbations. Online routing redistributes that budget across depth groups by reliability. As a result, increasing α_0 does not uniformly amplify unstable updates; reliable experts can take larger effective steps, while unstable experts are down-weighted. This selective amplification explains why AGOE can benefit from more aggressive base learning rates than static uniform-depth updates.

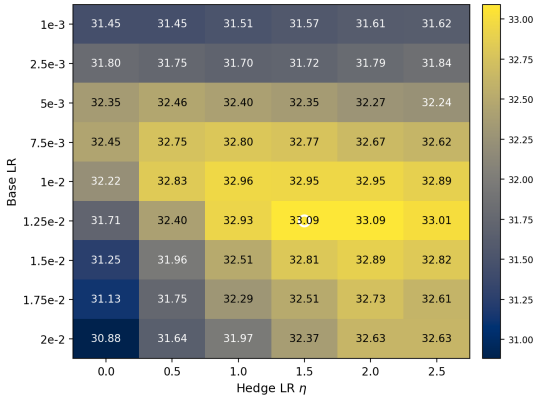


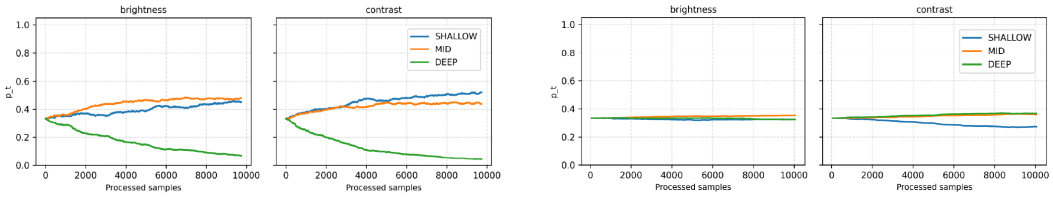
Figure 5: **AGOE-Tent sensitivity to base LR and η.**

Table 6: Three-level routing ablation on CIFAR-10-C (severity 5), accuracy (%).

Method (Tent-based)	CIFAR-10-C BS=2	CIFAR-10-C BS=64
(i-1) Tent (all)	18.34	81.13
(i-2) GALA	51.42	79.30
(ii-1) Shallow-only	44.79	67.66
(ii-2) Mid-only	17.39	68.26
(ii-3) Deep-only	16.99	80.49
(iii-1) AGOE (cosine)	57.71	81.76
(iii-2) AGOE (gradnorm)	22.71	80.29
(iii-3) AGOE (random)	26.77	80.36

Ablation: gains come from adaptive allocation, not a fixed depth choice. Table 6 evaluates whether gains come from adaptive routing itself rather than a fixed depth choice. Fixed-depth baselines exhibit opposite strengths across regimes: shallow-only is strong at BS2 but weak at BS64, while deep-only shows the reverse pattern. AGOE (cosine) improves both ends (57.71% at BS2 and 81.76% at BS64), indicating that online allocation outperforms committing to one static depth. Moreover, weaker routing controls underperform: AGOE (random) and AGOE (gradnorm) are substantially worse at BS2 (26.77% and 22.71%), showing that gains come from the proposed reliability signal and allocation strategy, not from adding a controller alone. GALA (51.42%/79.30%) trails AGOE at both ends: scoring layers once cannot track the reliable depth as it drifts across batch regimes, while per-step allocation can.

Routing dynamics across batch-size regimes. To diagnose AGOE-DeYO adaptation, we visualize depth-wise routing weights for each corruption type. Figure 6 compares CIFAR-10-C under BS=2 and BS=64. The distributions shift across both corruption types and batch regimes, and AGOE-DeYO does not collapse to a fixed expert. At BS=2, routing moves substantially from initialization: on brightness, shallow and middle experts rise to 0.4–0.5 while deep falls toward 0.1; on contrast, shallow climbs steadily and deep collapses, consistent with isolating noisy deep updates under small batches. At BS=64, all three experts stay near 1/3 with only mild drift, consistent with low noise where no isolation is needed. Instead, AGOE reallocates depth-wise updates online according to reliability, consistent with the static-depth cross-over motivation. Additional full per-corruption routing grids are provided in Appendix D.4.



(a) CIFAR-10-C, BS=2

(b) CIFAR-10-C, BS=64

Figure 6: **AGOE-DeYO changes its routing behavior with the test batch size.**

Expert granularity under a fixed budget. K controls depth resolution, not capacity: $K{=}1$ is uniform all-layer updating, $K{=}\infty$ is per-layer routing (25 layers on CIFAR-10-C, 31 on CIFAR-100-C). All use alignment routing with EMA anchors; BS64 uses total budget $\alpha_0 = K \times 0.0025$, BS2 uses fixed $\alpha_0 = 0.001$. At BS2 the default $K{=}3$ reaches 57.71%, while finer partitions degrade (13.36% at $K{=}6$, 13.19% per-layer); at BS64 accuracy peaks at moderate K and declines for fine partitions. Hence $K{=}3$ is a robust default, not a tuned optimum.

Table 7: Expert-granularity sensitivity with AGOE-Tent, accuracy (%).

	C10 B64	C10 B2	C100 B64
Tent-all ($K{=}1$)	81.13	18.34	54.13
AGOE $K{=}3$	81.76	57.71	54.61
AGOE $K{=}6$	78.43	34.73	54.25
AGOE $K{=}\infty$	77.93	13.19	52.82

Computational overhead remains minimal. Table 8 reports runtime under the continual setting (severity 5, BS=64) on a single NVIDIA RTX 4090 (24GB); all methods run in the same environment with identical batch size and stream order, and time is measured from log timestamps covering the full 15-corruption stream. AGOE remains in the same runtime regime as its base adapters: AGOE-Tent matches Tent in this setup, while AGOE-DeYO and AGOE-ZEROSIAM stay close to their non-AGOE counterparts in minute-level runtime. Overall, the robustness gains come without a meaningful increase in practical compute cost.

Table 8: Computational cost on a single RTX 4090 (continual, BS=64, mean over 3 runs).

Method	Total time (min)	Time/domain (s)
Tent	60.12	240.5
DeYO	60.18	240.7
ZEROSIAM	53.58	214.3
AGOE-Tent	61.05	244.2
AGOE-DeYO	61.20	244.8
AGOE-ZEROSIAM	54.52	218.1

5 Conclusion

This work revisit where test-time updates should be applied: static depths show a batch-size-dependent cross-over with no uniformly reliable choice. AGOE partitions normalization parameters into depth-wise experts and reallocates a fixed budget via Hedge routing driven by update–trajectory alignment, using per-depth learning rates with one backward pass. Experiments show it improves base adapters where fixed depths degrade, complementing objective design and sample filtering.

USE OF AI ASSISTANTS

Large language models were used only as writing aids, and no part of the technical content, method design, proofs, experimental design, or analysis depended on AI-generated material. Specifically, they were used for (i) polishing and editing author-written text (grammar, conciseness, and readability), and (ii) literature retrieval and discovery, such as locating related work and candidate citations. All retrieved references were checked against the original papers by the authors, who verified every citation's relevance and correctness; all scientific claims, results, and figures remain the responsibility of the authors.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Guohao Chen, Shuaicheng Niu, Deyu Chen, Jiahao Yang, Zitian Zhang, Mingkui Tan, Pengcheng Wu, and Zhiqi Shen. Zerosiam: An efficient asymmetry for test-time entropy optimization without collapse. 2026.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36:14070–14093, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

Junyuan Hong, Lingjuan Lyu, Jiayu Zhou, and Michael Spranger. MECTA: Memory-economic continual test-time model adaptation. In *International Conference on Learning Representations*, 2023.

Xuefeng Hu, Gokhan Uzunbas, Sirius Chen, Rui Wang, Ashish Shah, Ram Nevatia, and Ser-Nam Lim. Mixnorm: Test-time adaptation through online normalization estimation. *arXiv preprint arXiv:2110.11478*, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Minsoo Kim et al. Buffer layers for test-time adaptation. *arXiv preprint arXiv:2601.00000*, 2026.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9w3iw8wDuE.

Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. Ttn: A domain-shift aware batch normalization in test-time adaptation. In *The Eleventh International Conference on Learning Representations*, 2023.

Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 512–523, 2020.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *Internetional Conference on Learning Representations*, 2023.

Junyoung Park, Jin Kim, Hyeongjun Kwon, Ilhoon Yoon, and Kwanghoon Sohn. Layer-wise auto-weighting for non-stationary test-time adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1414–1423, 2024.

Sabyasachi Sahoo, Mostafa ElAraby, Jonas Ngnawé, Yann Pequignot, Frédéric Precioso, and Christian Gagné. A layer selection approach to test time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2026.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11539–11551, 2020.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Jin Shin and Hyun Kim. L-tta: Lightweight test-time adaptation using a versatile stem layer. *Advances in Neural Information Processing Systems*, 37:39325–39349, 2024.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Pedro Vianna, Muawiz Sajjad Chaudhary, An Tang, Guy Cloutier, Guy Wolf, Michael Eickenberg, and Eugene Belilovsky. Channel selection for test-time adaptation under distribution shift. *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=uXl3bZLkr3c`.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022.

Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision*, pp. 3–19, 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/532a2f85b6977104bc93f8580abbb330-Paper.pdf.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022.

Bowen Zhao, Chen Chen, and Shu-Tao Xia. DELTA: Degradation-free fully test-time adaptation. In *International Conference on Learning Representations*, 2023.

A Related Work

Test-time adaptation. TTA adapts a deployed model to unlabeled test streams under distribution shift, without source data or target labels (Wang et al., 2021). Most progress has been made on the first two questions of our framework. *What* to optimize is typically defined by self-supervised proxies such as entropy, auxiliary test-time losses, and collapse-resistant variants (Wang et al., 2021; Chen et al., 2026), while *how* to update stably is addressed by sample selection and reweighting (e.g., EATA, SAR, DeYO), teacher–student smoothing, and memory-economic updates (Niu et al., 2022; 2023; Wang et al., 2022; Hong et al., 2023). These advances, however, all presuppose a third decision: *where* the update is applied. Except for continual variants that update the full network (Wang et al., 2022), the parameter subset is fixed before deployment, usually to all normalization affine parameters. In our framing, TTA methods have matured on *what* and *how* while leaving *where* as a static hyper-parameter.

Normalization and parameter-efficient adaptation. Adapting normalization affine parameters is the dominant recipe for efficient TTA, because it is cheap, stable, and architecture-agnostic (Ioffe & Szegedy, 2015; Wu & He, 2018; Ba et al., 2016; Schneider et al., 2020; Nado et al., 2020; Hu et al., 2021; Wang et al., 2021). Yet this practice freezes the depth at which adaptation occurs, and mounting evidence suggests the choice matters: different layers respond differently to distribution shift (Vianna et al., 2023; Park et al., 2024), some methods restrict updates to shallow stem layers to reduce cost (Shin & Kim, 2024), others select layer subsets by update–trajectory alignment, notably GALA (Sahoo et al., 2026), which scores layers by alignment but commits to a fixed subset once and does not revisit the decision as the stream evolves, and buffer-layer schemes re-introduce trainable layers at fixed depths (Kim et al., 2026). These results indicate that *where* to adapt is a non-trivial, data-dependent decision rather than a design constant. Unlike one-time selection, AGOE revisits the allocation at every step, which matters when the reliable depth itself drifts. AGOE differs by treating depth-wise normalization groups as online experts: the routing distribution is revised at every step with a label-free consistency signal, turning the layer-selection problem into an online allocation problem.

Online learning and expert aggregation. Expert aggregation maintains a distribution over experts and updates it multiplicatively from observed feedback, with regret guarantees against the best fixed expert in hindsight (Shalev-Shwartz et al., 2012; Hoi et al., 2021). We adapt this machinery to TTA with two departures from the classical formulation. First, the "experts" are not prediction models but parameter groups: AGOE aggregates *update budgets* across depth-wise normalization experts rather than their outputs. Second, the feedback is not a loss on labeled data but a label-free measure of update consistency—the alignment between each group's update direction and its displacement from a stream anchor—which is available at every online step without supervision. This yields a lightweight controller that composes with any normalization-based TTA objective and requires a single backward pass per batch.

B Discussion and Limitations

Relation to sample-level methods. Sample-level methods (EATA, SAR, DeYO) decide what evidence to trust: which samples or losses should drive the update. AGOE decides where the resulting update budget is spent: which depths receive it. The two are complementary rather than competing. In our experiments, routing improves adapters that already filter samples (DeYO) as well as plain entropy minimization (Tent), with the largest margins where the base update is unstable. When the base objective is already stable, routing adds little, which is expected: there is no unreliable depth to isolate.

When to use AGOE. Use AGOE under small test batches, online label shifts, and long continuously changing streams, where fixed-depth updates overreact to noisy batch statistics. Under large stable batches with a strong base adapter, AGOE stays near parity and can be left on or off without much difference. The routing visualizations give a practical diagnostic: substantial movement of routing weights indicates depths disagree and routing matters; weights staying near uniform indicates all depths agree and routing is optional.

Limitations. First, AGOE only reallocates a fixed budget and cannot create new budget: if every depth is unreliable (e.g., collapsed baselines in late continuous domains), routing cannot rescue accuracy. Second, the alignment score measures directional consistency rather than downstream accuracy, so a stable but unhelpful direction can still attract budget; combining alignment with uncertainty or loss-based signals is future work. Third, the current study uses coarse depth groups and per-corruption resets in the standard protocol; finer granularity fragments the budget and fully continual operation without resets remains less explored.

C Proof of the routing guarantee

In this part, we provide the detailed proof for Prop. 1.

Proof. Prop. 1 is the standard Hedge bound for prediction with expert advice (Freund & Schapire, 1997; Cesa-Bianchi & Lugosi, 2006), restated for routing scores. Let $w_{k,1} = 1$ and $w_{k,t+1} = w_{k,t} \exp(-\eta z_{k,t})$, and write $W_t = \sum_k w_{k,t}$, so that $\mathbf{p}_t = \mathbf{w}_t / W_t$. Since $z_{k,t} \in [0,1]$, Hoeffding's lemma (Cesa-Bianchi & Lugosi, 2006, Lemma 2.2) applied to the distribution $\mathbf{p}_t$ gives

$$W_{t+1}/W_t = \sum_k p_{k,t} \exp(-\eta z_{k,t}) \le \exp(-\eta \langle \mathbf{p}_t, \mathbf{z}_t \rangle + \eta^2/8). \tag{5}$$

Taking logs,

$$\log W_{T+1} - \log W_1 \le -\eta \sum_t \langle \mathbf{p}_t, \mathbf{z}_t \rangle + T\eta^2/8. \tag{6}$$

Telescoping $\log(W_{T+1}/W_1)$ against $\log w_{k^*,T+1} = -\eta \sum_t z_{k^*,t}$ for the best fixed expert k^* yields

$$\sum_{t=1}^{T} \langle \mathbf{p}_t, \mathbf{z}_t \rangle - \min_k \sum_{t=1}^{T} z_{k,t} \le \frac{\log K}{\eta} + \frac{\eta T}{8}. \tag{7}$$

With $\eta = \sqrt{8 \log K / T}$ the right-hand side is $\sqrt{(T/2) \log K} = \mathcal{O}(\sqrt{T \log K})$. The guarantee is on scores, not accuracy: routing cannot be much more inconsistent than the best fixed depth, which is exactly the stability property needed when the noisy depth drifts.

D Additional Experimental Results

This appendix provides implementation details, long-horizon online-shift results, full-scale continuously changing-stream evaluations, and fine-grained routing and static-depth analyses. Together, these results test whether the main conclusions remain stable as stream length and evaluation granularity increase.

D.1 Normalization-parameter partition

For each backbone, we collect only the trainable affine parameters of normalization layers; convolutional, linear, and other non-normalization parameters remain frozen. We order the normalization layers by their module depth and divide the ordered layers into K contiguous groups containing approximately equal numbers of layers. The trainable affine parameters belonging to each layer inherit its group assignment. If N normalization layers are present, group k contains the layers with indices $\lfloor (k-1)N/K \rfloor + 1$ through $\lfloor kN/K \rfloor$. Thus, the groups are disjoint and cover every trainable normalization parameter exactly once.

For WRN28, the partition follows its three backbone blocks, so the default $K = 3$ groups correspond to the shallow, middle, and deep blocks. For ResNet-50-GN, we first order all trainable GN affine parameter groups according to their occurrence from the input to the output of the network and then apply the same uniform contiguous partition; the convolutional weights and all other parameters are not included. Changing K changes only these boundaries and the routing resolution, not the set of trainable parameters or the total learning-rate budget. AGOE partitions trainable normalization affine parameters into shallow, middle, and deep groups. Routing weights are initialized uniformly at the beginning of each stream and updated by the Hedge rule described in the main paper. The base learning rate follows the corresponding base-adapter protocol, and AGOE scales each depth group's effective learning rate by its current routing weight.

Hyperparameters are tuned per method under the same search budget. Non-AGOE baselines use an SGD learning rate of 0.0025 for each trainable normalization group. For AGOE with $K = 3$ disjoint depth experts, we use the same total learning-rate budget, $0.0025 \times 3 = 0.0075$, denoted by α_0, and allocate it as $\alpha_{k,t} = \alpha_0 p_{k,t}$. Since $p_{k,1} = 1/3$, each group starts from $\alpha_{k,1} = 0.0025$, matching the baseline group-wise learning rate. Thus, AGOE does not use a larger initial per-group step size; it reallocates the same total update budget online.

Small-batch (BS=2), online label-shift, and continuously changing-stream results are treated as stress tests. Within every setting, all methods use the same backbone, stream construction, corruption order, batch size, reset policy, and evaluation protocol.

D.2 Long online imbalanced label-shift stream

To complement the main-text results at $T = 10000$, we evaluate a substantially longer stream with $T = 100000$ under persistent prior drift. Here, T denotes the number of online adaptation steps per corruption stream. This setting is stricter because update errors can accumulate over a much longer horizon.

Table 9 reports the results. The benefit becomes adapter-dependent over this extended horizon: AGOE preserves clear gains for Tent and ZeroSiam, while AGOE-DeYO remains close to its already strong base adapter. This heterogeneity indicates that long-stream robustness depends jointly on routing quality and the base adapter's tolerance to sustained prior mismatch.

Table 9: Comparisons with baselines on ImageNet-C at severity level 5 under online imbalanced label shifts (imbalance ratio = ∞). $T = 100000$ online adaptation steps.

	Noise			Blur				Weather				Digital				
Label Shift	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet-50-GN (Source)	22.15	23.12	22.09	19.91	11.38	21.38	25.05	40.38	47.03	34.01	68.80	36.19	18.66	29.25	52.78	31.48
• EATA	38.90	41.09	39.51	**27.99**	**27.39**	36.49	**39.56**	51.40	50.22	56.62	71.47	51.14	41.74	57.38	58.60	45.97
• SAR	36.56	38.31	37.52	19.01	21.96	33.82	34.64	36.79	45.50	3.52	71.19	46.84	7.04	52.36	56.56	36.11
• CoTTA	22.37	23.56	22.46	19.45	11.11	22.77	25.78	40.77	46.20	29.91	69.08	38.77	18.06	31.62	53.78	31.71
• Tent	14.14	19.20	14.36	13.77	4.00	18.69	18.82	16.22	21.82	1.94	69.61	42.44	6.09	49.22	53.82	24.28
• AGOE-Tent	32.24	37.20	35.29	16.99	18.47	27.03	28.10	14.05	25.40	2.21	71.96	45.97	6.25	53.04	56.21	31.36 (+7.08)
• DeYO	43.48	45.14	44.46	23.12	25.29	**41.18**	37.76	53.00	**51.47**	**59.25**	72.91	**52.70**	**48.26**	**59.66**	**59.54**	**47.81**
• AGOE-DeYO	**44.12**	**46.32**	**45.52**	23.85	25.44	40.58	38.29	**53.88**	50.59	59.21	**73.24**	51.38	46.32	59.12	57.71	47.70 (−0.11)
• ZEROSIAM	35.08	36.94	36.41	20.61	22.69	31.26	32.88	44.76	42.59	49.97	70.00	45.57	32.62	51.26	55.02	40.51
• AGOE-ZEROSIAM	37.58	40.23	39.12	20.37	20.95	32.46	34.52	48.50	46.01	51.80	71.78	46.83	34.16	53.04	55.82	42.21 (+1.70)

D.3 ImageNet-C 5k continuous stream

This section extends the CTTA analysis with a longer continuously changing stream on ImageNet-C. Relative to reset-based protocols, this setting is more challenging because no corruption-wise reset is applied, so adaptation decisions interact over time.

Table 10 reports the continuously changing-stream results. The overall pattern remains aligned with the main text: methods that are more vulnerable to long-horizon instability benefit more from adaptive budget allocation, while gains can be smaller for stronger or more stable adapters in this particular stream. These results support the claim that AGOE is most valuable when cumulative online mismatch is a primary failure mode.

Table 10: Comparisons with baselines on ImageNet-C at severity level 5 under continuously changing environments.

	Noise			Blur				Weather				Digital				
Continuous Environment	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet-50-GN (Source)	22.30	22.94	22.18	19.52	11.56	21.80	25.38	40.18	46.58	33.10	68.68	36.14	18.24	29.88	52.16	31.38
• EATA	25.82	35.20	39.32	22.40	21.68	27.96	33.42	40.74	44.90	48.32	66.14	44.40	26.60	45.00	55.74	38.51
• SAR	23.34	27.44	29.86	20.18	15.66	25.08	27.92	41.20	42.38	43.06	68.04	41.04	21.20	41.54	54.54	34.83
• CoTTA	22.20	23.18	22.30	19.76	12.10	21.94	25.42	39.74	46.10	32.80	68.62	37.26	18.20	35.42	53.88	31.93
• Tent	22.78	25.72	26.34	19.32	15.18	22.12	23.64	38.38	38.64	36.86	63.80	36.24	15.32	35.70	48.46	31.23
• AGOE-Tent	23.12	26.82	28.86	18.20	16.40	22.38	24.78	39.90	41.80	40.48	63.44	36.74	17.88	39.54	52.48	32.85 (+1.62)
• DeYO	26.80	37.26	40.34	17.80	19.82	24.00	29.38	39.00	42.20	46.34	68.24	44.74	24.20	44.80	56.28	37.41
• AGOE-DeYO	28.12	39.28	41.86	16.62	19.58	22.54	28.64	38.64	41.76	46.70	69.62	43.44	21.74	42.22	52.96	36.91 (−0.50)
• ZEROSIAM	23.12	27.22	31.50	19.28	15.96	24.18	25.96	39.74	41.16	40.02	65.82	38.92	18.02	38.80	50.46	33.34
• AGOE-ZEROSIAM	23.40	28.90	33.76	17.76	16.72	23.42	26.30	40.52	42.92	41.60	64.36	37.94	19.22	41.04	52.62	34.03 (+0.69)

D.4 Additional per-corruption routing visualizations

The aggregate main-text plots show that routing changes with batch size. We additionally report the complete per-corruption maps to verify that this behavior is not driven by only a small subset of corruptions. Figures 7 and 8 show all 15 CIFAR-10-C corruptions for AGOE-DeYO under BS=2 and BS=64, each panel tracing shallow/middle/deep routing weights over processed samples from uniform initialization. At BS=2, most corruptions show large departures: deep weights fall steadily while shallow or middle weights rise (e.g., glass blur, contrast, elastic transform), indicating per-step isolation of noisy deep updates; a few corruptions move less, showing the allocation is corruption-dependent rather than hard-coded. At BS=64, all 15 panels stay near 1/3 with only mild drift, indicating low noise where no isolation is needed. The contrast between the two maps confirms that AGOE does not collapse to a fixed expert but shifts budget with the batch regime.

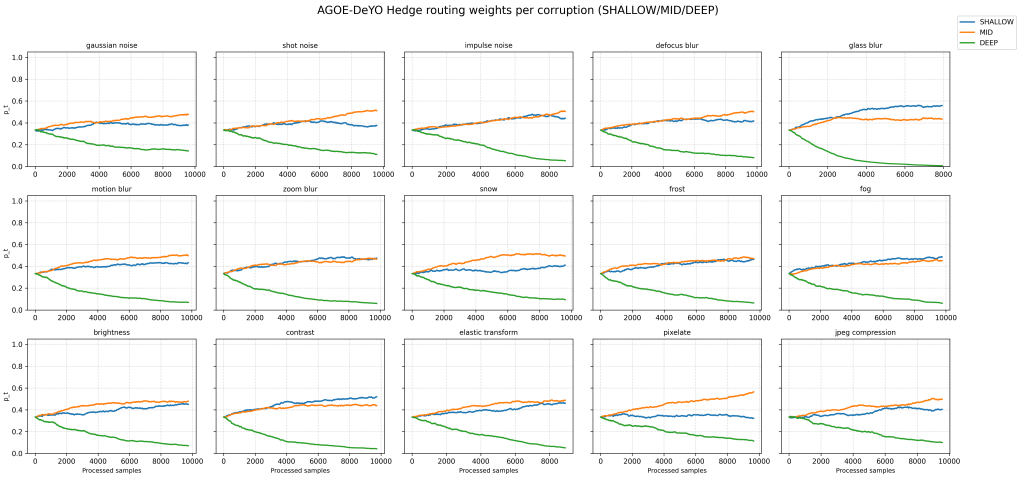


Figure 7: **Full per-corruption routing map for AGOE-DeYO at BS=2.** Small batches more frequently favor shallow or middle experts, consistent with noisier batch statistics.

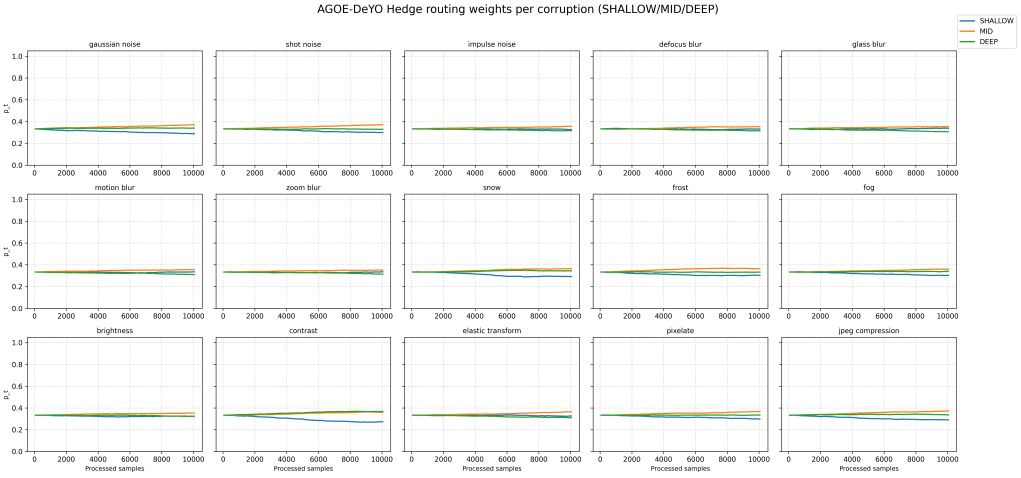


Figure 8: **Full per-corruption routing map for AGOE-DeYO at BS=64.** Larger batches produce a different and generally more balanced allocation across depth-wise experts.

D.5 Additional static-depth cross-over visualizations

Figures 9 and 10 provide enlarged versions of the static-depth comparisons used to motivate AGOE. Across both datasets, the preferred update depth changes with test batch size. Thus, the cross-over is not specific to one dataset and no single fixed depth remains uniformly optimal.

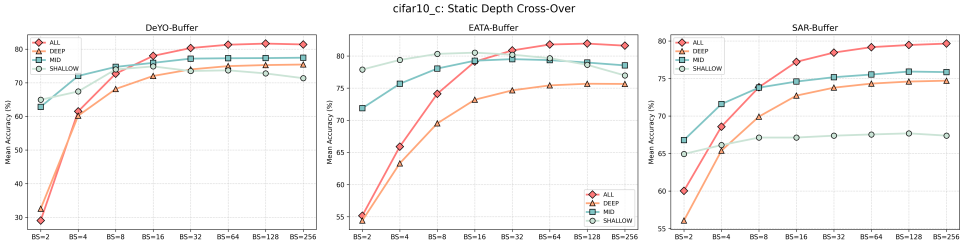


Figure 9: **Static-depth cross-over on CIFAR-10-C.** Different normalization-depth choices dominate in different batch-size regimes.

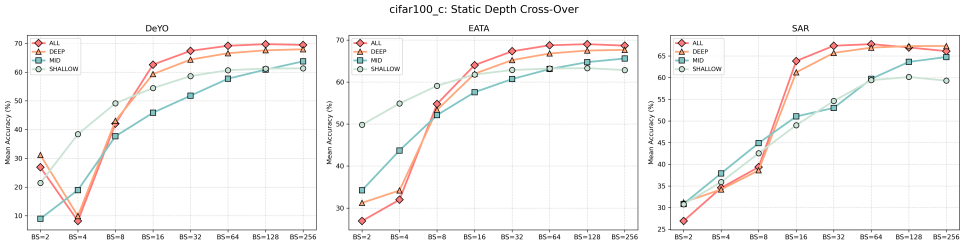


Figure 10: **Static-depth cross-over on CIFAR-100-C.** The preferred adaptation depth again varies with batch size.