A Layer Selection Approach to Test Time Adaptation

Sabyasachi Sahoo[1,2], Mostafa ElAraby[2,3], Jonas Ngnawe[1,2], Yann Batiste Pequignot[1], Frédéric Precioso[4], Christian Gagné[1,2,5]

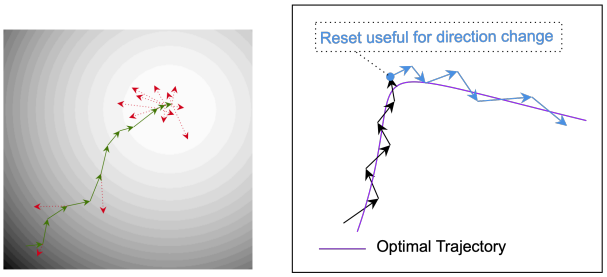
[1]IID, Université Laval
[2]Mila
[3]Université de Montréal
[4]Université Cote d'Azur, CNRS, INRIA, I3S, Maasai
[5]Canada CIFAR AI Chair
sabyasachi.sahoo.1@ulaval.ca

Abstract

Test Time Adaptation (TTA) addresses the problem of distribution shift by adapting a pretrained model to a new domain during inference. When faced with challenging shifts, most methods collapse and perform worse than the original pretrained model. In this paper, we find that not all layers are equally receptive to the adaptation, and the layers with the most misaligned gradients often cause performance degradation. To address this, we propose GALA, a novel layer selection criterion to identify the most beneficial updates to perform during test time adaptation. This criterion can also filter out unreliable samples with noisy gradients. Its simplicity allows seamless integration with existing TTA loss functions, thereby preventing degradation and focusing adaptation on the most trainable layers. This approach also helps to regularize adaptation to preserve the pretrained features, which are crucial for handling unseen domains. Through extensive experiments, we demonstrate that the proposed layer selection framework improves the performance of existing TTA approaches across multiple datasets, domain shifts, model architectures, and TTA losses.

1 Introduction

Distribution shifts (Gulrajani and Lopez-Paz 2021) present significant challenges when deploying deep learning models in real-world scenarios. Test Time Adaptation (TTA) (Liang, He, and Tan 2023) has emerged as a promising approach for adapting pretrained models to novel domains during inference. However, these methods often falter when confronted with severe or diverse distributional changes. To mitigate potential performance degradation, various regularization strategies have been proposed (Niu et al. 2022; Shin et al. 2024). Nevertheless, these strategies might not effectively address all types of shifts or TTA losses (Burns and Steinhardt 2021; Zhao et al. 2023a). Moreover, the selection of layers in the existing TTA approaches typically remains unchanged across different shifts (Wang et al. 2024), which may not be optimal. In contrast, layer selection has demonstrated substantial improvements in related fields such as domain generalization (Chattopadhyay, Balaji, and Hoffman 2020), fine-tuning (Lee et al. 2023), multi-task learning (Wallingford et al. 2022), and continual learning (Zhao et al. 2023b), underscoring the importance and broad potential of layer selection. Still, the question of optimal layer selection remains largely unexplored in the context of TTA.



(a) Gradient alignment (b) Helpful reset

Figure 1: Intuition for proposed approaches: **(a)** As the model reaches closer to minima, the individual sample gradients start to be misaligned with gradients of previous samples (Mahsereci et al. 2017; Forouzesh and Thiran 2021; Agarwal, D'souza, and Hooker 2022). We leverage this misalignment to identify trainable layers. **(b)** While effective in moving in the direction of most aligned gradients, the introduced criterion based on angular deviation could prevent adaptation when a direction change is needed, even if the following updates (or gradients) are aligned. A reset of the past horizon (i.e., gradients of previous samples) considered in the alignment condition can help resolve such situations.

In this paper, we study layer selection for TTA and show that not all layers of a given model are equally receptive to adaptation. Our findings suggest that adapting the right layer can lead to meaningful improvement, while adapting the wrong layer can cause significant performance degradation in TTA approaches. Specifically, we find that while adapting a certain layer may benefit one shift, it may be detrimental to another. Additionally, we find that on a given shift, the effect of adapting a certain layer also depends on the loss used. Therefore, while we observe an important potential in selecting the right layer to adapt in each situation, identifying these layers at test time can be challenging.

To address the challenges of layer selection, we propose Gradient-Aligned Layer Adaptation, GALA, a novel criterion to identify good layers for adaptation at test time.

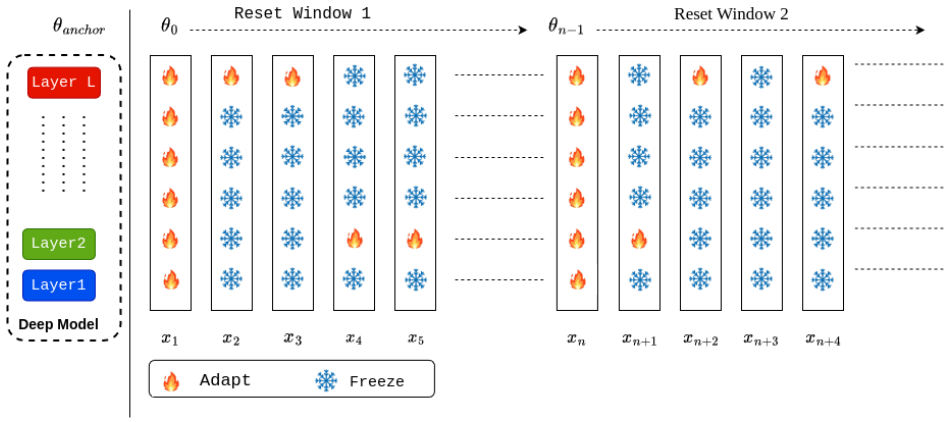


Figure 2: Gradient-Aligned Layer Adaptation or GALA framework adapts the most gradient-aligned layer per sample. It adapts all the layers for the first sample in a reset window (e.g., $x_1, x_n, \dots$). For all the other samples, it adapts the most gradient-aligned layer per sample. It can also skip the adaptation on a given sample if all the layers are misaligned. We use a reset window to periodically reset the anchor parameters to allow for a change in direction.

GALA ranks all the layers of a model based on the gradient alignment of the current adaptation step. As the model approaches the optimization minima, the variance in gradient updates increases (Mahsereci et al. 2017; Forouzesh and Thiran 2021; Agarwal, D'souza, and Hooker 2022), leading to potential overfitting and performance degradation. Building on this insight, for each layer, we propose to measure the angle deviation of the proposed gradient update from the average of all gradient updates performed so far (including the proposed one). This measure can also be expressed as the cosine between the proposed update and the (anticipated) total displacement of the parameters from their pretrained values. This allows us to compare the updates for each layer on a common scale and only perform the update of the layer with the smallest angle.

Our extensive experiments on Domainbed (Gulrajani and Lopez-Paz 2021) and Continual TTA benchmark (Wang et al. 2022) demonstrate that GALA consistently surpasses *all layers* and *ERM* (no adaptation) baselines and other existing layer selection baselines across various datasets, various neural network backbones, and various losses. Further analysis reveals that GALA can identify the good layers, which exhibit significant displacement in a single direction and higher gradient alignment. This layer selection strategy enhances the model's ability to adapt to novel domains by mitigating performance degradation and potentially serves as a regularization mechanism, reducing catastrophic forgetting of source domain knowledge. Ablation studies reveal that GALA's performance is robust to hyperparameter choices.

The contributions of our paper are summarized as follows:

1. We study the problem of layer selection for TTA and find that while adapting specific layers can enhance performance, the optimal set of layers for adaptation is not universal but rather contingent upon the particular distribution shift encountered and the TTA loss function employed during inference.
2. We introduce GALA, a novel layer selection criterion to identify good layers to adapt per sample that can be applied across various distribution shifts and TTA loss functions at test time.
3. Through extensive experiments across different backbones, datasets, and TTA losses, we show that GALA outperforms standard *ERM* (no adaptation), *all layers* baselines, and other layer selection baselines (i.e., AutoRGN and AutoSNR (Lee et al. 2023)) for TTA.

2 Proposed Approach

In the following, we describe the Gradient-Aligned Layer Adaptation (GALA) framework for Test Time Adaptation (TTA). We first introduce our layer selection framework for TTA (Sec. 2.1), before describing the cosine distance criterion proposed to identify the most trainable layers (Sec. 2.2), and then present the reset window strategy used to improve performances with the proposed cosine criterion (Sec. 2.3).

2.1 Layer selection framework for TTA

Let $f_{\theta_{\text{src}}}$ denote the model parameterized by parameters θ_{src} trained beforehand on the source domain $\mathcal{D}_{\text{src}}$. Let us also assume that target domain samples $\{x_i\}_{i=1}^{n}$ are coming in an online fashion at test time. For some sample x_i at test time, TTA adapts the model to obtain θ_i before performing inference (Sun et al. 2020; Liang, He, and Tan 2023). We set $\theta_0 = \theta_{\text{src}}$ and, at each step, θ_i is obtained by updating θ_{i-1} using the following equation:

$$\theta_i = \theta_{i-1} + \mathbf{u}_i, \tag{1}$$

where $\mathbf{u}_i$ is a parameter update specific to the TTA algorithm. Typically, if SGD optimizer is used with learning rate η, this update takes the form $\mathbf{u}_i = -\eta \nabla \mathcal{L}(x_i; \theta_{i-1})$, where $\mathcal{L}$ is the unsupervised loss specific to the TTA method.

In this section, we consider single-step TTA performed online on a single input sample using an SGD optimizer for

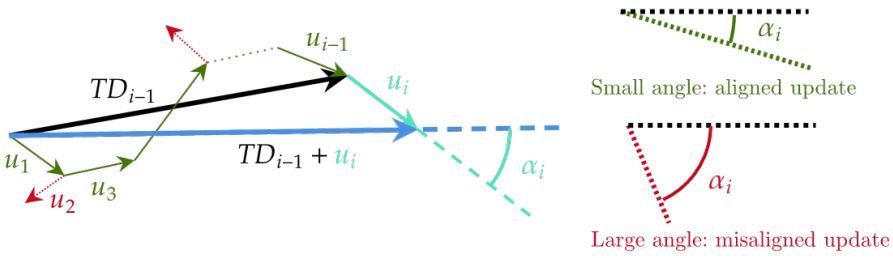


Figure 3: Illustration of proposed criterion based on angular deviation. Different layers can be ranked based on their alignments with previous gradient updates. In the figure, updates drawn in red are discarded, while green updates are applied, adding up to $\mathbf{TD}_{i-1}$. The update under scrutiny $\mathbf{u_i}$ is drawn in cyan, and its sum with $\mathbf{TD}_{i-1}$ is drawn in blue. Application of update $\mathbf{u_i}$ or not is based on the angle α_i.

notation simplicity. Throughout, we assume the deep learning model is written as a certain composition of functions, which we simply refer to as layers, though any granularity would do. This allows us to write the model at step i as $f_{\theta_i} = f_{\theta_{i,L}} \circ \cdots \circ f_{\theta_{i,1}}$, where $\theta_{i,l}$ denote the parameters of layer l at step i. The update equation at step i can be written for each layer as:

$$\theta_{i,l} = \theta_{i-1,l} + \mathbf{u}_{i,l}. \tag{2}$$

To perform layer selection, we modify this update equation by introducing a mask:

$$\theta_{i,l} = \theta_{i-1,l} + m_{i,l}\, \mathbf{u}_{i,l}, \tag{3}$$

where $m_{i,l} \in \{0, 1\}$ is the value of the binary mask applied to the update $\mathbf{u}_{i,l}$.

2.2 Cosine distance criterion

Existing works have shown that gradient descent happens in a tiny subspace (Gur-Ari, Roberts, and Dyer 2018). Moreover, as the model reaches closer to the minima, the gradients across the samples get noisy (Mahsereci et al. 2017; Forouzesh and Thiran 2021; Agarwal, D'souza, and Hooker 2022). We aim to identify the layers with the most beneficial gradient updates to the model for adapting to the new domain. Let us assume that the total displacement of parameters of layer l at the start of the i^{th} step is given by:

$$\mathbf{TD}_{i-1,l} = \sum_{j=1}^{i-1} m_{j,l}\mathbf{u}_{j,l} = \theta_{i-1,l} - \theta_{0,l}. \tag{4}$$

Our proposed criterion relies on the angular deviation of the update $\mathbf{u}_{i,l}$ from the direction of the total displacement that would result from making this update:

$$\cos(\alpha_{i,l}) = \frac{\mathbf{u}_{i,l} \cdot (\mathbf{u}_{i,l} + \mathbf{TD}_{i-1,l})}{\|\mathbf{u}_{i,l}\|_2 \, \|\mathbf{u}_{i,l} + \mathbf{TD}_{i-1,l}\|_2}. \tag{5}$$

This angle can be interpreted as the deviation of the update under consideration from the anticipated average update, which has the same direction as the anticipated total displacement $\mathbf{u}_{i,l} + \mathbf{TD}_{i-1,l}$ – see this illustrated in Fig. 3.

Comparing our criterion across layers allows us to define which update is performed by defining the mask:

$$m_{i,l} = \begin{cases} 1 & \text{if } \cos(\alpha_{i,l}) > \lambda \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where λ is the selection threshold. The fact that the cosine metric lies in the $[-1, 1]$ domain allows us to compare the alignment of updates for layers with different sizes of parameters. We set a single $\lambda > 0$ for thresholding over all layers, which prevents the adaptation of updates that are misaligned with the updates applied in the past. A λ close to 1 will only allow adaptation of updates aligned with past updates, while a lower λ would be less restrictive.

2.3 Cosine distance with reset

While the cosine distance can stop adaptation for noisy gradients, our criterion may fail, especially when the gradient update trajectory needs to change direction after a certain point. If the gradient updates meet an inflection point in the loss landscape, cosine distance will prevent further adaptation, and the model will remain stuck at this point even if the gradient update is informative. To solve such cases, we propose to use resets for the computation of the total displacement of a layer. We use a fixed window scheme for resetting the initial parameter point, which we will call the *anchor point*. This corresponds to:

$$\mathbf{TD}_{i,l} = \theta_{i,l} - \theta_{r,l}, \tag{7}$$

where $\theta_{r,l}$ is the parameter at last reset step $r = \lfloor \frac{i-1}{s} \rfloor$, and s is the size of the reset window. The anchor point changes only when the reset window changes, as illustrated in Fig. 2.

3 Experiments

This section compares our proposed approaches with existing baselines on Domainbed (Gulrajani and Lopez-Paz 2021), a popular benchmark with large single distribution shifts, and Continual TTA, a popular benchmark with multiple distribution shifts.

TTA losses Two popular TTA losses are considered: Pseudo-Labeling (PL) (Lee et al. 2013) and SHOT (Liang, Hu, and Feng 2020). We perform hyperparameter selection based on Zhao et al. (2023a), where we report the performance for the best hyperparameter set found by sweeping over a range of values.

Baselines We compare the TTA performance obtained by adapting *All layers* vs. the layers proposed by our approach. We also report the *ERM* (*no adaptation*) performance of the

	TTA	Method	PACS ↑	VLCS ↑	Terra ↑	Office ↑	Mean ↑
ResNet-18	ERM	–	80.99 (±0.9)	75.14 (±1.2)	40.80 (±0.2)	62.18 (±0.4)	64.78
	PL	All layers	81.79 (±0.7)	65.69 (±1.5)	35.40 (±9.7)	60.20 (±1.4)	60.77
		AutoRGN	82.82 (±0.6)	72.63 (±1.3)	38.18 (±6.1)	62.38 (±0.2)	64.00
		AutoSNR	80.58 (±1.2)	65.72 (±1.8)	35.01 (±10.4)	59.82 (±0.9)	60.28
		GALA	**83.56 (±0.6)**	**75.48 (±1.2)**	**44.19 (±1.1)**	**62.67 (±0.2)**	**66.47**
	SHOT	All layers	83.48 (±0.3)	66.23 (±2.8)	33.81 (±1.3)	63.03 (±0.4)	61.64
		AutoRGN	**84.10 (±0.5)**	69.78 (±1.3)	37.37 (±0.7)	63.09 (±0.2)	63.59
		AutoSNR	83.43 (±0.3)	66.26 (±2.7)	33.75 (±1.2)	63.02 (±0.4)	61.62
		GALA	83.92 (±0.8)	**76.23 (±1.1)**	**42.13 (±1.4)**	**63.32 (±0.3)**	**66.40**
ResNet-50	ERM	–	82.84 (±0.5)	75.83 (±0.9)	46.14 (±2.3)	66.93 (±0.3)	67.93
	PL	All layers	82.36 (±2.8)	69.22 (±1.4)	42.28 (±3.2)	61.54 (±3.3)	63.85
		AutoRGN	**85.03 (±1.9)**	75.35 (±1.4)	48.44 (±2.4)	66.93 (±0.3)	68.94
		AutoSNR	83.41 (±3.4)	70.14 (±4.6)	44.08 (±3.4)	61.95 (±3.0)	64.90
		GALA	84.87 (±0.8)	**76.88 (±1.6)**	**50.10 (±2.5)**	**67.34 (±0.3)**	**69.80**
	SHOT	All layers	85.15 (±1.1)	64.25 (±1.1)	35.33 (±3.1)	67.37 (±0.3)	63.03
		AutoRGN	**86.34 (±1.1)**	70.2 (±0.9)	40.59 (±1.3)	68.10 (±0.4)	66.31
		AutoSNR	85.51 (±0.5)	64.26 (±1.3)	34.97 (±3.2)	67.33 (±0.2)	63.02
		GALA	86.13 (±0.8)	**76.48 (±1.0)**	**45.94 (±1.6)**	**68.13 (±0.3)**	**69.17**

Table 1: Accuracy (%) of various layer selection methods on Domainbed benchmark (setup described in Sec. 3.1). The best method for a given TTA loss and backbone is in bold.

pretrained model. In Domainbed, we also compare against AutoRGN and AutoSNR (Lee et al. 2023), two popular baselines proposed to identify optimal layers in fine-tuning setup.

Implementational details We report results for GALA with *window size* of 20 and *selection threshold* of 0.75 with single-layer granularity. It appears that GALA is not overly sensitive to hyperparameters, and those values work well overall – see Sec. 5 for more discussion on hyperparameter values and the design choices. We also scale the updates for a few initial samples in the reset window to reduce their impact on incorrect layer selection.

3.1 Domainbed results

For the experiments on Domainbed, we follow the evaluation protocol as described in Iwasawa and Matsuo (2021), including dataset splits for the following four datasets: PACS (Li et al. 2017), VLCS (Fang, Xu, and Rockmore 2013), Terra Incognita (Beery, Van Horn, and Perona 2018), and Office-Home (Venkateswara et al. 2017). Results are reported on two backbones (i.e., ResNet-18 and ResNet-50) with batch normalization layers, while the pretrained models are made using default hyperparameters described in Gulrajani and Lopez-Paz (2021). Mean and standard deviation are reported over three repetitions with different random seeds. See Appendix A.1 for further details.

Key takeaways from results are reported in Tab. 1:

- GALA outperforms ERM (no adaptation) by 2% overall and *All layers* TTA baselines by more than 5% overall across all losses, backbones, and datasets.
- Existing layer selection baselines like AutoRGN or AutoSNR can improve performance compared to all layers TTA in most setups, especially AutoRGN, but fail to improve against no adaptation baselines for some datasets like VLCS or TerraIncognita or some TTA losses like SHOT. GALA consistently demonstrates equivalent or superior performance across all datasets and TTA losses, achieving an overall improvement of about 2%.

TTA	Method	CIFAR10C ↓	CIFAR100C ↓
ERM		43.50 (±18.7)	46.40 (±15.7)
PL	All layers	88.72 (±1.2)	98.63 (±1.5)
	GALA	**28.68 (±6.6)**	**33.69 (±5.7)**
SHOT	All layers	89.33 (±2.3)	97.32 (±4.8)
	GALA	**20.46 (±7.7)**	**32.87 (±5.6)**

Table 2: Accuracy (%) of layer selection methods on Continual TTA benchmark (with the setup described in Sec. 3.2). The best method for a given TTA loss is in bold.

- GALA improves over Domainbed large shift datasets (i.e., PACS, OfficeHome) similar to AutoRGN and AutoSNR while comfortably outperforming the ERM baseline. On small shift datasets (i.e., VLCS, TerraIncognita), existing baselines struggle to outperform the *no adaptation* baseline while GALA appears to prevent degradation caused by over-adaptation, thereby enhancing performance over the ERM baseline and safeguard against further degradation.

3.2 Continual TTA results

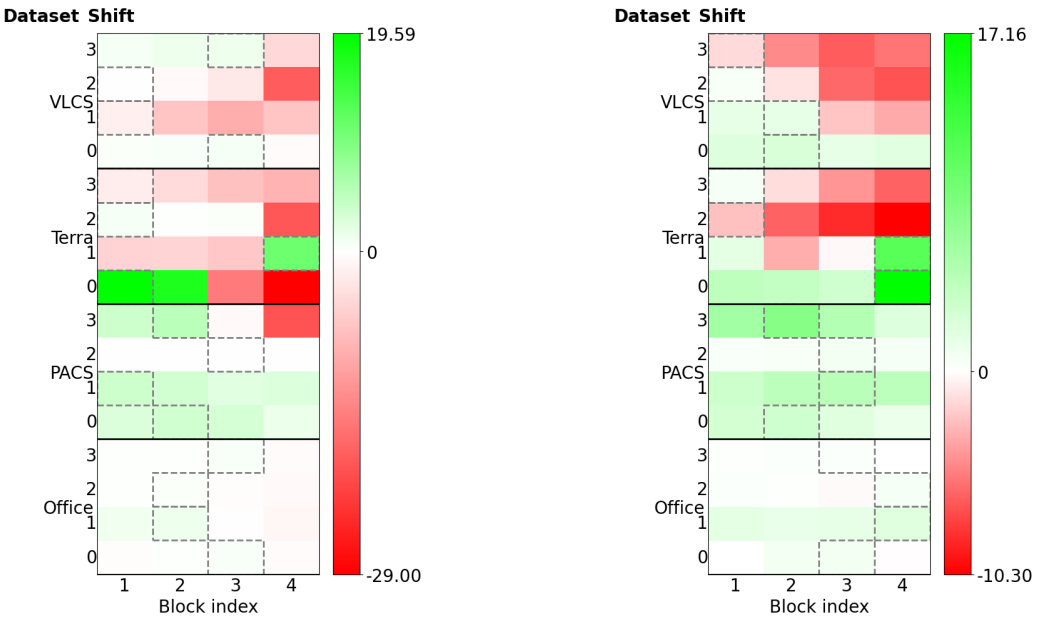
We follow the evaluation protocol as described in Wang et al. (2022), evaluating performance on two datasets-backbones: 1) CIFAR10C (Hendrycks and Dietterich 2019) with WideResNet-28 (Zagoruyko and Komodakis 2016) and CIFAR100C (Hendrycks and Dietterich 2019) with ResNeXt-29 (Xie et al. 2017). The pretrained models are trained as described in Robustbench (Croce et al. 2021). Mean and standard deviation are reported across the 15 corruption types. Further details are given in Appendix A.2.

The key takeaways based on the results from Tab. 2 are:

- Performance degradation by training all layers is worse in the Continual TTA benchmark containing multi-domain shifts than degradation in the Domainbed benchmark containing single-domain shifts. Moreover, more severe degradations are observed in CIFAR100C, which has 100 classes, compared to CIFAR10, which includes 10 classes, despite similar ERM performance on both datasets.
- GALA consistently outperforms ERM by about 15% and all layers TTA baseline by about 65%, despite severe degradation.

4 Layer Selection Study

In this section, we evaluate the importance of layer selection for test time adaptation on the Domainbed benchmark and provide some analysis and motivation for GALA. We use the Domainbed benchmark with the ResNet-18 backbone, which contains four blocks of layers. We study the effect of choosing one block over another by performing adaptation on a single block while freezing all the other blocks of the model. We refer to blocks and layers interchangeably in this section. We report the difference between TTA and ERM accuracy over all blocks for each loss and dataset shift setting. Otherwise, we rely on the same setup and evaluation protocol described in Sec. 3.1.



(a) PL loss-based TTA

(b) SHOT loss-based TTA

Figure 4: Heatmap of Performance improvement (%) per-block on Domainbed benchmark. Performance improvement is the difference between the TTA accuracy of a given block/layer and ERM accuracy for the same shift. Positive performance improvements are shown in green, and negative performance improvements (or degradation) are in red. Using the bounding box, we highlight the best block per loss and dataset shift. Further details in Sec. 4.

4.1 Layer selection matters

In Fig. 4, we observe that not all layers are equally receptive to adaptation. We refer to a layer as good or bad based on the accuracy improvement of selecting a given layer w.r.t. performance of a pretrained model on the same shift. We compare against Empirical Risk Minimization (ERM) or the frozen pretrained model's performances, as we are interested in measuring the performance improvement or degradation brought by individual layers during adaptation. Also, the ERM model performs better on average than *all layer* TTA, as seen in Sec. 3, and it becomes a natural baseline that can help contrast different layers.

The selection of layers in existing TTA approaches typically remains unchanged in all adaptation settings. We find it can be a suboptimal strategy and one of the major causes of degradation in existing TTA approaches – no single layer adaptation is suitable for all settings. Therefore, layer selection is essential for TTA, and we propose GALA to improve the performance of existing TTA approaches in various settings.

4.2 What affects the adaptability of a layer?

Using the same setup and evaluation protocol (cf., Sec. 3.1), we are making the following observations on Fig. 4 about the factors affecting the adaptability of good layers:

- Location of good layers in a model can change across shifts of a given dataset, despite using pretrained models trained on the same class labels. Similar observations have also been made in fine-tuning setups(Lee et al. 2023). There is a need for a good layer selection criterion that depends on target samples observed by the model at test time.
- We also find that good layers in a model can change with different TTA loss functions, even for the same shift and dataset. Hence, a good layer selection criterion must also depend on the TTA loss function used to adapt the model at inference.

Since gradients depend on the shift and TTA loss function used, GALA uses layerwise gradients to identify the adaptability of each layer in the model.

4.3 How do good layers differ from bad layers?

To perform a detailed per-layer analysis, we created the Tiny-Domainbed benchmark, which was made as a smaller version of Domainbed. It consists of all the critical shifts with the brightest red/green layers (displayed in Fig. 4), whose good layers can also change with the TTA method. We follow the benchmark and evaluation protocol described in Sec. 3.1, with further details given in Appendix A.3. Based on Tab. 3, the following are the differences between good and bad layers:

- Adaptation with *Worst Block* results in poor TTA accuracy, poorer generalization to the target domain, and higher forgetting. Since training all layers involves training the worst layer, this could explain why training all layers results in poorer TTA accuracy. On the other hand, *Best Block* results in better generalization to the target domain. This implies that TTA with good layers can poten-

Method	TTA Acc. ↑	Gen. ↑	Forget. ↓	Rank corr. ↑
All Blocks	53.6	46.5	31.3	N/A
Worst Block (oracle)	43.5	38.7	39.9	-1
Best Block (oracle)	**64.1**	**63.9**	28.7	**1**
Random Block	53.1	49.1	<u>13.1</u>	0
GALA	<u>59.4</u>	<u>58.0</u>	**9.3**	<u>0.76</u>

Table 3: Effect of various layer selection methods on TTA Accuracy (%), Generalization (%), Forgetting (%) and Spearman correlation with Best Block ($\in [-1, 1]$) averaged over different shifts on Tiny-Domainbed benchmark (with the setup described in Sec. 4.3). *TTA Acc* is the accuracy of testing samples from the target domain seen during adaptation. *Generalization* is the accuracy of the held-out split of the target domain after adaptation. *Forgetting* is the drop in accuracy on the held-out split of source domains after adaptation. *Rank correlation* is the Spearman correlation of layer selection rank between the oracle and the method. Bold and underlined denote best and second-best, respectively.

tially learn target domain features better than TTA with bad layers.

- We observe that *Best Block* results in reduced source forgetting compared to *Worst Block*. This implies that TTA with good layers strikes an improved balance between learning new features on the target domain while retaining useful pretrained features from the source domain.

Therefore, we propose GALA to identify good layers for adaptation, which can help balance adaptation to the new domain while reducing source forgetting.

4.4 How does GALA compare to oracle strategies?

To analyze GALA's layer selection behavior, we compare it to the oracle strategies given by Best block and Worst block on the Tiny-Domainbed benchmark (Tab. 3).

GALA well approximates the oracle layer selection GALA substantially improves over *All Blocks*, *Worst Block*, and *Random Block* method. In some sense, *Best Block* method acts as an empirical upper-bound performance if we have access to a target domain with labels while incurring the high computational cost of brute forcing over individual layers of the model. GALA comes close to this upper bound performance without requiring any target labels using a cheap layer selection criterion. As a result, GALA also effectively balances computational cost with performance.

GALA is more conservative than the oracle GALA selects the layers for adaptation with the most aligned gradients. It can stop adaptation if the gradients are noisy or no longer aligned to prevent further degradation. In Tab. 3, we see that it may have aggressively stopped a few useful updates compared to the *Best Block*, our empirical upper bound. As a result, it gets much better at avoiding forgetting but is a bit lower on TTA accuracy and generalization.

GALA tends to select more often the blocks with better accuracy Oracle TTA performance, as measured in Fig. 4, ranks the four blocks for each configuration. Similarly, GALA chooses to update each layer with a particular frequency during TTA, leading to a ranking of the four blocks. We assess the relationship between these two different ways of ranking blocks using Spearman rank correlation and find $\rho = 0.76$ (cf. Tab. 3), which seems to indicate that the selection strategy used by GALA is a good proxy for the oracle TTA performance achieved when adapting always the same layer.

Setting	Condition	Accuracy ↑
Partitioning	Single block	67.64
	Single layer	68.57
	Multiple layers	66.48
Window Size	5	68.46
	20	68.57
	∞	68.37
Threshold	0.5	68.57
	0.75	68.57
	0.99	68.6
Batch Size = 1	All Layers	33.47
	GALA	67.28
Continual TTA	No Reset	69.9
	With Reset	71.1

Table 4: Accuracy (%) under different experimental conditions. The values are averaged on Domainbed for the first four settings and Continual TTA for the last.

5 Analysis of GALA

In this section, we evaluate the impact of different design choices and hyperparameters of GALA in Tab. 4, supporting choices presented in Tab. 1. For the partitioning setting, *Single block* means a single block of many layers is updated at each iteration, *Single layer* corresponds to the best layer selected for the update, and *Multiple layers* corresponds to individually best layers selected for the update based on the cosine distance and the threshold. Also, a window size of ∞ implies no reset. Some important observations stemming from Tab. 4:

- Layer granularity performs better than block granularity. At layer granularity, GALA has better fine-grained control over choosing the layers to adapt, improving performances in all cases tested.
- Adaptation with the best single layer is much better than with the best multiple layers. Cosine distance can correctly identify the single best layer to train, although it may still struggle to determine the best set of multiple layers to update.
- Optimal reset-window size can improve performance. We see that a reset window size of 20 works reasonably well across the backbones and the TTA losses tested on Domainbed.
- The choice of selection threshold is not very sensitive. A threshold of 0.75 seems to work across the board without being too restrictive.

In the following section, we briefly analyze some aspects of the proposed approach.

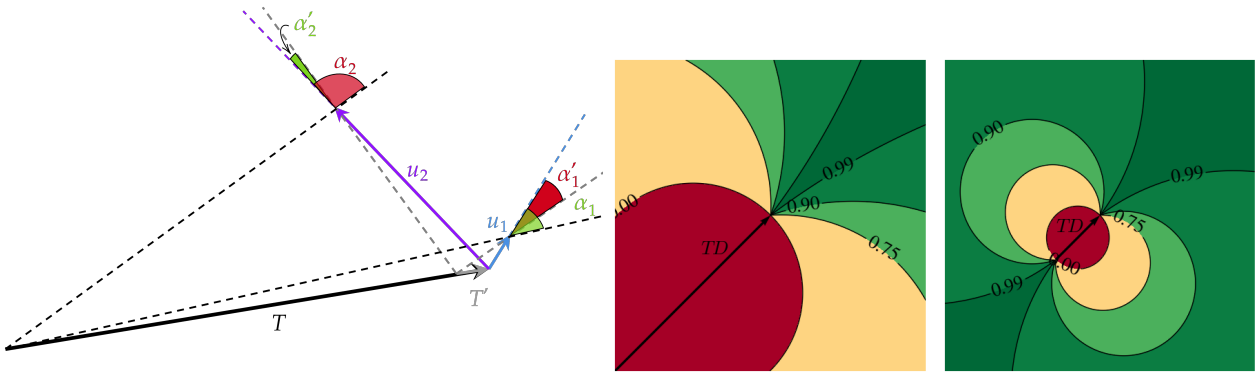


Figure 5: Effect of magnitude of u on cosine distance criterion. **Left:** Consider two vectors such that u_1 is smaller than u_2 but is better aligned with its displacement. For large displacements (T), alignment becomes crucial and GALA selects u_2. For small displacements ($T^{'}$), the update's magnitude can dominate the criterion, and GALA selects u_1. **Middle and Right:** Plot of cosine metric values with level curves. Alignment prevails for small updates compared to the total displacement (Middle). But, for updates with large magnitude compared to total displacement (Right), large cosine values can be obtained even for misaligned updates.

Proposed cosine distance criterion effectively balances gradient magnitude and direction. Let us first rewrite the GALA criterion in Eq. 5 for a given layer l in terms of $T = \|\mathbf{TD}_{i-1,l}\|$, $u = \|\mathbf{u}_{i,l}\|$ and the angle β between $\mathbf{TD}_{i-1,l}$ and $\mathbf{u}_{i,l}$. Using the Pythagorean theorem, we obtain:

$$\cos(\alpha) = \frac{T\cos(\beta) + u}{\sqrt{(T + u\cos(\beta))^2 + (u\sin(\beta))^2}}. \quad (8)$$

We observe that our criterion depends on the norm T of the total displacement, the norm u of the update, and their alignment, given by the angle β between these vectors. Fig. 5 shows the cosine metric plots. We see that while alignment is crucial for large displacements, the update's magnitude can also dominate for small displacements. For example, consider two layers with the same norm T but different updates $\mathbf{u}_1$ and $\mathbf{u}_2$. If $\|\mathbf{u}_1\|$ is smaller than $\|\mathbf{u}_2\|$ but $\mathbf{u}_1$ is more aligned with its displacement, two scenarios arise:

1. For larger T, GALA selects layer 1, favoring the alignment and exploiting the learned direction. This scenario would seem more common during TTA.
2. For small T, GALA selects layer 2, favoring the magnitude, and can explore over different directions. This can occur for initial samples.

Consequently, GALA effectively balances the gradient magnitude and the direction of gradients for selecting the best layer. More discussion is in Appendix A.4.

Proposed layer selection framework offers a more flexible adaptation strategy for TTA. The selection of layers in existing TTA approaches typically remains unchanged across different shifts. On the other hand, sample selection-based TTA (Niu et al. 2022) approaches aim to improve performance by skipping the adaptation of all layers on a few unreliable samples. Based on Eq. 3 , we can see that GALA is more flexible and general than the existing layer selection and sample selection strategies in TTA for performing layerwise adaptation.

Reset mechanism seems beneficial in multi-domain shift settings. Comparing GALA with and without reset on Tab. 4, we see that while reset yields only marginal improvement on Domainbed, a single-domain shift benchmark, its benefits are more evident on a multi-shift benchmark like Continual TTA. This indicates that the reset mechanism's ability to facilitate slight adjustments in the overall gradient update direction may be advantageous in a continuously changing testing domain.

GALA is quite robust on single sample adaptation. In Tab. 4, we show that in the adverse setting of batch size of 1, while existing TTA approaches witness severe performance degradation, GALA improves on *all layers* baseline on Domainbed.

6 Conclusion

In this paper, we introduce Gradient Aligned Layer Adaptation (GALA), a novel layer selection framework explicitly designed for Test Time Adaptation (TTA). Our comprehensive study reveals that layers in neural networks exhibit varying receptiveness to adaptation, and the optimal set of layers for adaptation depends on both the specific distribution shift and the loss function employed during inference. Building on these insights, we propose GALA, a dynamic layer selection criterion that ranks layers based on gradient alignment, effectively mitigating overfitting and performance degradation. Extensive experiments across diverse datasets, model architectures, and TTA losses demonstrate GALA's superior performance compared to existing methods, including standard *ERM*, *all-layers* adaptation, and other layer selection baselines.

Acknowledgments

This work is supported by the DEEL Project CRDPJ 537462-18 funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Consortium for Research and Innovation in Aerospace in Québec (CRIAQ), together with its industrial partners Thales Canada inc, Bell Textron Canada Limited, CAE inc and Bombardier inc. Computations were made on the cedar, and beluga supercomputers, managed by Calcul Québec and the Digital Research Alliance of Canada (Alliance). We extend our gratitude to the members of the #lunch-at-mila and #deel_ood for their valuable input, with special thanks to Vineetha Kondameedi for her essential feedback in enhancing the quality of this paper.

References

Agarwal, C.; D'souza, D.; and Hooker, S. 2022. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10368–10378.

Beery, S.; Van Horn, G.; and Perona, P. 2018. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, 456–473.

Burns, C.; and Steinhardt, J. 2021. Limitations of post-hoc feature alignment for robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2525–2533.

Chattopadhyay, P.; Balaji, Y.; and Hoffman, J. 2020. Learning to Balance Specificity and Invariance for In and Out of Domain Generalization. In *European Conference in Computer Vision (ECCV)*.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2021. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Fang, C.; Xu, Y.; and Rockmore, D. N. 2013. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, 1657–1664.

Forouzesh, M.; and Thiran, P. 2021. Disparity between batches as a signal for early stopping. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*, 217–232. Springer.

Gulrajani, I.; and Lopez-Paz, D. 2021. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*.

Gur-Ari, G.; Roberts, D. A.; and Dyer, E. 2018. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*.

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

Iwasawa, Y.; and Matsuo, Y. 2021. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896. Atlanta.

Lee, Y.; Chen, A. S.; Tajwar, F.; Kumar, A.; Yao, H.; Liang, P.; and Finn, C. 2023. Surgical Fine-Tuning Improves Adaptation to Distribution Shifts. In *The Eleventh International Conference on Learning Representations*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Liang, J.; He, R.; and Tan, T.-P. 2023. A Comprehensive Survey on Test-Time Adaptation under Distribution Shifts. *International Journal of Computer Vision*.

Liang, J.; Hu, D.; and Feng, J. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, 6028–6039. PMLR.

Mahsereci, M.; Balles, L.; Lassner, C.; and Hennig, P. 2017. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*.

Niu, S.; Wu, J.; Zhang, Y.; Chen, Y.; Zheng, S.; Zhao, P.; and Tan, M. 2022. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, 16888–16905. PMLR.

Shin, J.; Lee, J.; Lee, S.; Park, M.; Lee, D.; Hwang, U.; and Yoon, S. 2024. Gradient Alignment with Prototype Feature for Fully Test-time Adaptation. *arXiv preprint arXiv:2402.09004*.

Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A.; and Hardt, M. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, 9229–9248. PMLR.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5018–5027.

Wallingford, M.; Li, H.; Achille, A.; Ravichandran, A.; Fowlkes, C.; Bhotika, R.; and Soatto, S. 2022. Task adaptive parameter sharing for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7561–7570.

Wang, Q.; Fink, O.; Van Gool, L.; and Dai, D. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7201–7211.

Wang, Z.; Luo, Y.; Zheng, L.; Chen, Z.; Wang, S.; and Huang, Z. 2024. In Search of Lost Online Test-time Adaptation: A Survey. *International Journal of Computer Vision (IJCV)*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *British Machine Vision Conference*. British Machine Vision Association.

Zhao, H.; Liu, Y.; Alahi, A.; and Lin, T. 2023a. On Pitfalls of Test-Time Adaptation. In *International conference on machine learning*. PMLR.

Zhao, H.; Zhou, T.; Long, G.; Jiang, J.; and Zhang, C. 2023b. Does continual learning equally forget all parameters? In *International Conference on Machine Learning*, 42280–42303. PMLR.