Safeguarding AI

Dr Varun Ojha

School of Computing Newcastle University

UKRI National Edge AI Hub

Centre for AI Safety

Varun.Ojha@Newcastle.ac.uk

The 9th Euro-China Conference on Intelligent Data Analysis and Applications

VSB TECHNICAL UNIVERSITY OF OSTRAVA

(ECC 2025) July 21-23,2025 Ostrava, Czech Republic

Safeguarding AI

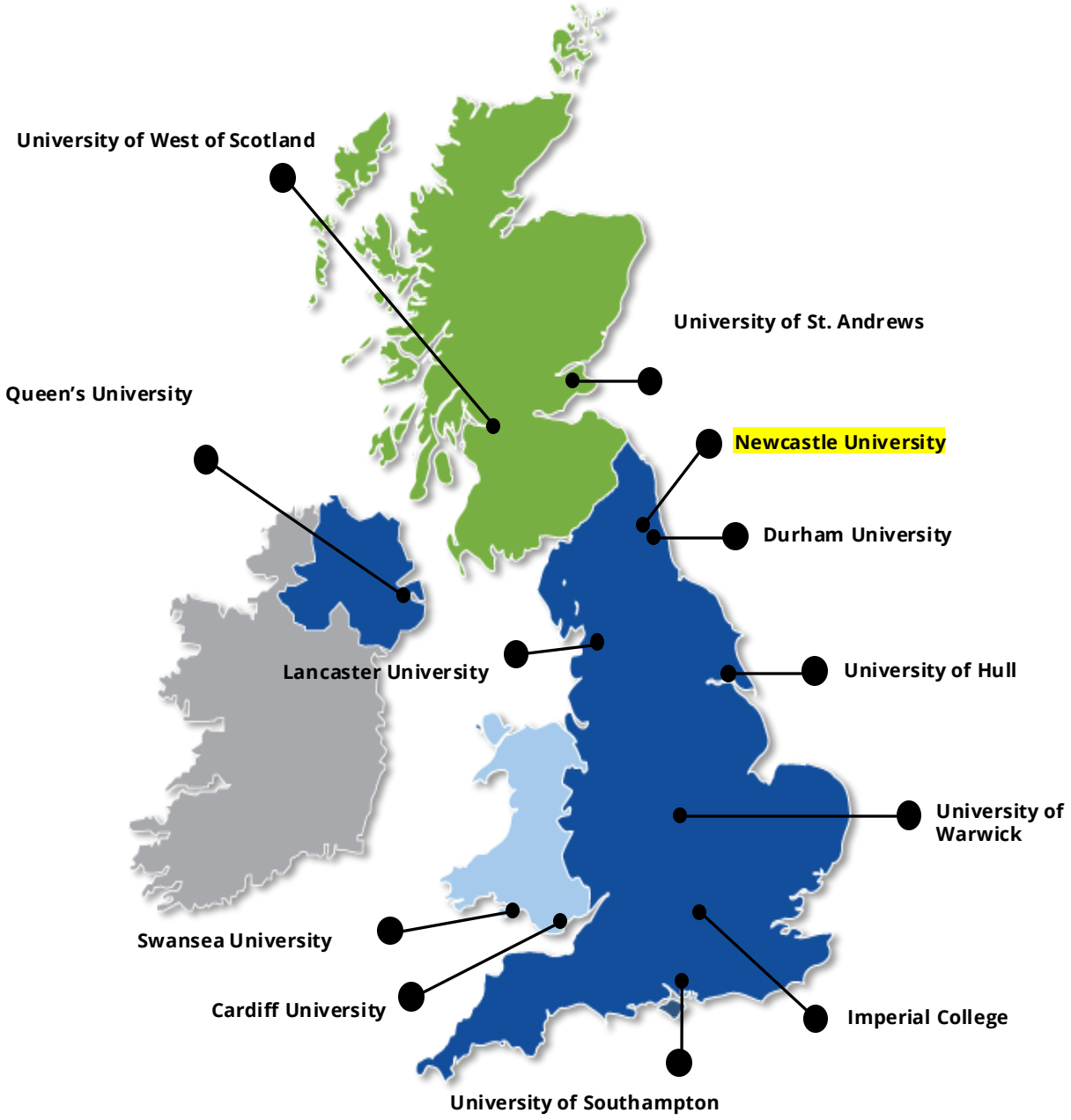
Dr Varun Ojha | The 9th Euro-China Conference 2025
July 21 – 23 2025, Ostrava, Czech Republic

National Edge AI Hub

Engineering and Physical Sciences Research Council

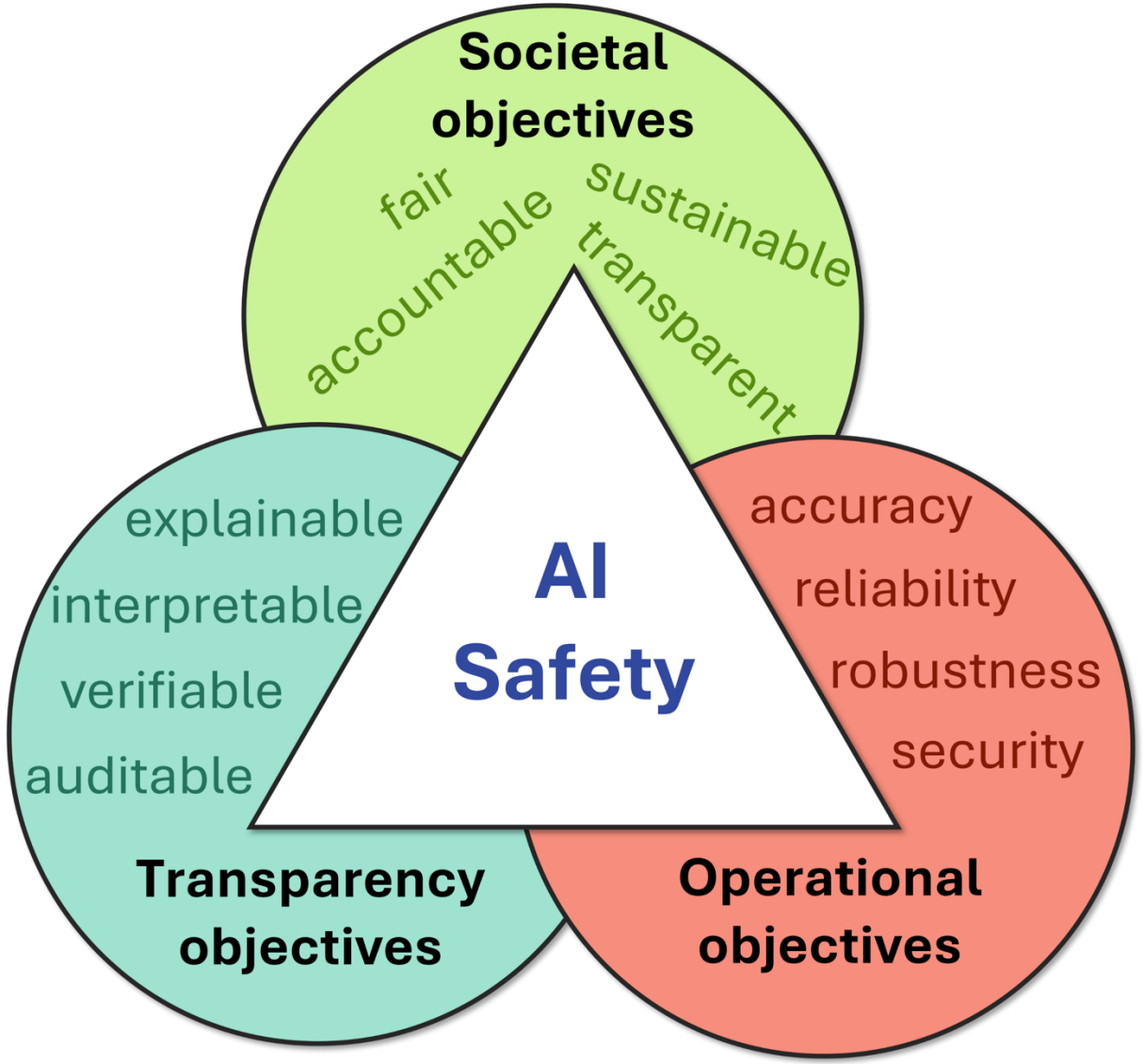
Newcastle University

edgeaihub.co.uk



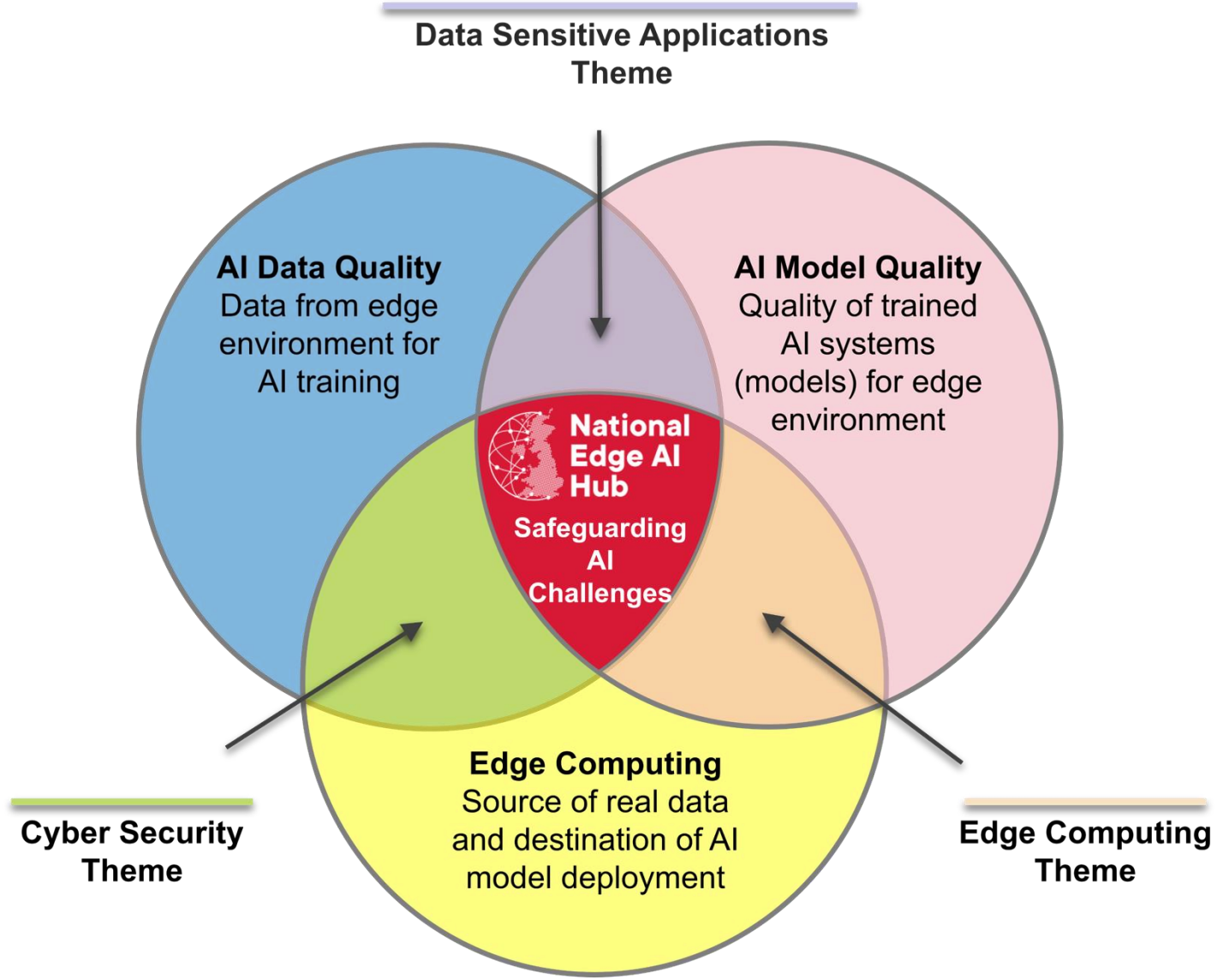
Safeguarding AI at the National Edge AI Hub

Varun.Ojha@ncl.ac.uk | edgeaihub.co.uk | ojhavk.github.io



Safeguarding AI at the National Edge AI Hub

Varun.Ojha@ncl.ac.uk | edgeaihub.co.uk | ojhavk.github.io



Safeguarding AI challenges

- **Monitoring of Data/Model Quality**

  How to monitor cyber-disturbances impact on the quality of data, AI algorithms learning and the overall application resilience?

- **Recovery of Data/Model Quality**

  How to recover data and AI model quality that are impacted by cyber-disturbances and ensure suitability for AI model deployment on devices at Tiers 1, 2 of EC architectures ?

- **Assurance of Continuity of Data Quality and Model Quality**

  How to assure AI algorithms continually adapt to EC environments where unknown cyber-disturbances that were not present in the original training dataset?

Part 1
Safeguarding AI: Model Robustness

Adversarial attacks

Calculated using Deep Neural Networks (DNNs) weights (white-box attack)

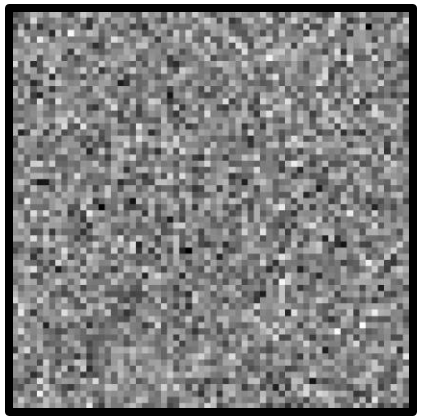
x

Input example
Predicted as '***Horse***'

Perturbation magnitude

$+ \quad \varepsilon \quad *$

δ



Adversarial perturbation
('***Plane***' class)

$=$

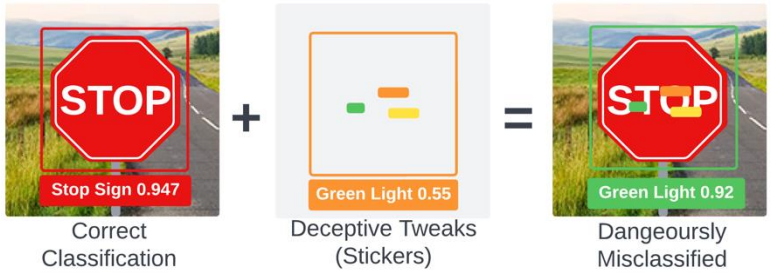
x_ε

Adversarial example
Predicted as '***Plane***'

The general premise of a robustness analysis is to subject DNNs to the '**worst case**' conditions and evaluate the ***ability for a DNN to remain invariant*** under such settings.

Adversarial attack types

Evasion Attacks

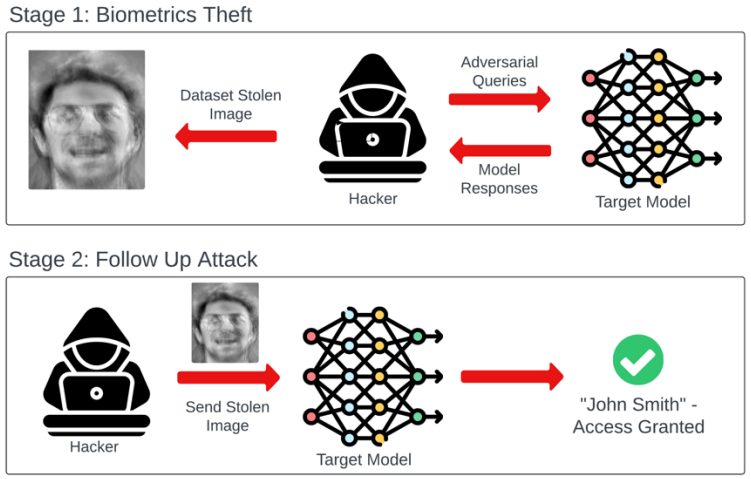


Attacks are designed to subtly alter inputs to mislead AI models during inference, causing them to misclassify specific inputs

Poison Attacks

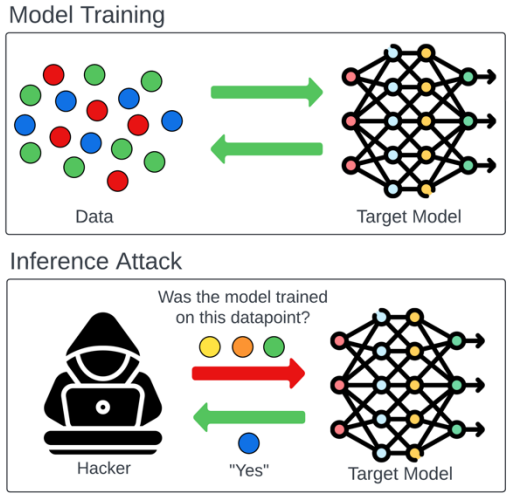
Attacks are designed to subtly alter the labels of training examples or inject anomalous data points, thus, attackers can manipulate the model to favour certain outcomes or fail under specific conditions

Inversion Attacks



Attackers can deduce characteristics or even reconstruct portions of the original training dataset

Inference Attacks

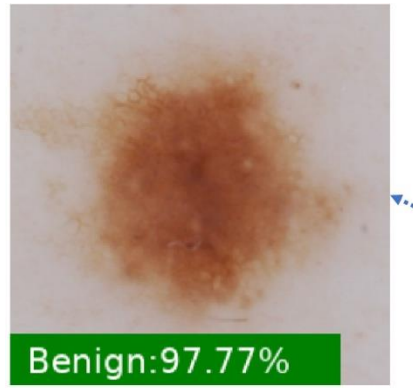


Adversary's attempts to deduce sensitive information from an AI model by examining its outputs and behaviours

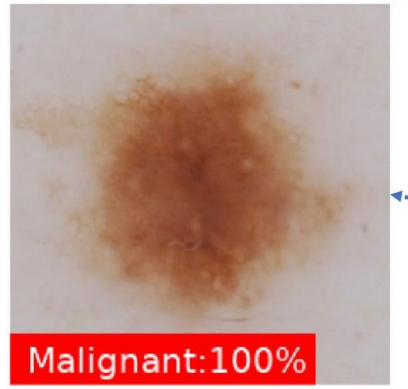
Image source: https://mindgard.ai/blog/ai-under-attack-six-key-adversarial-attacks-and-their-consequences

White box attack: Gradient based attacks

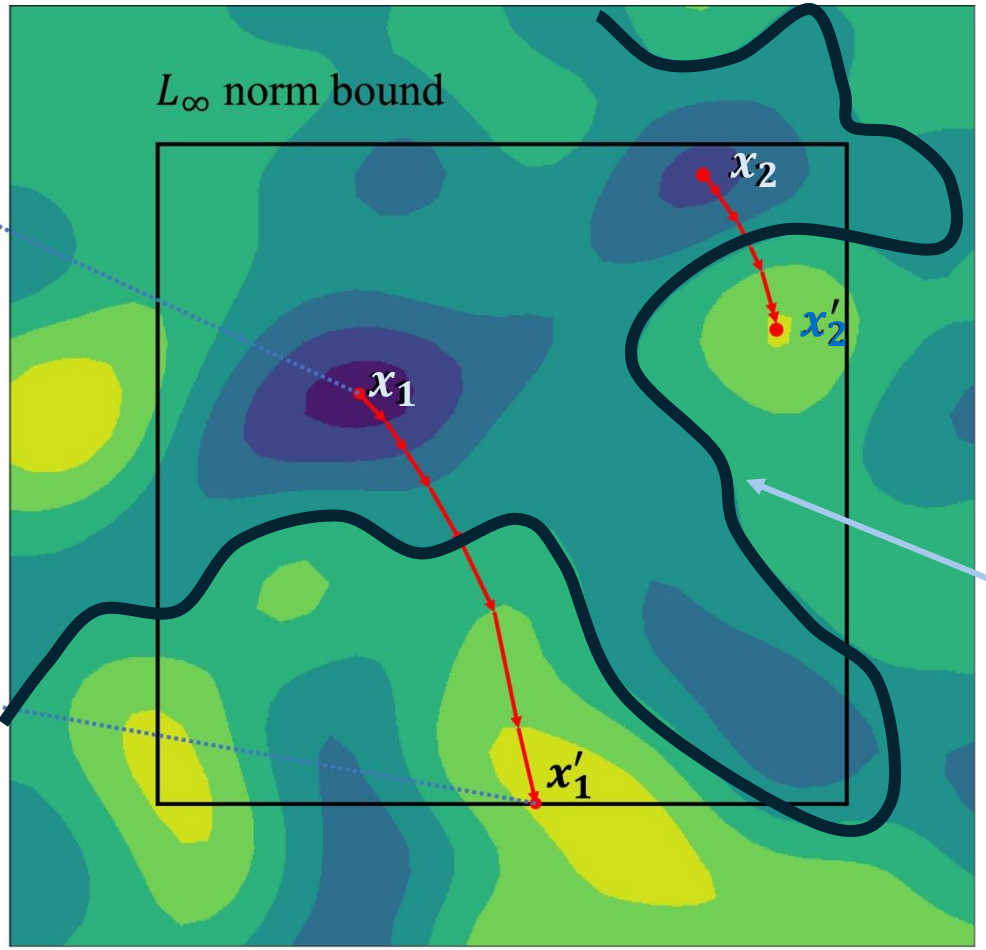
Attacks known the model (gradient/parameters) and carefully craft an attack on the model



x_1: original image



x_1': adversarial image



High Loss

Low Loss

x_1 and x_2 are original and **clean images** with low loss values

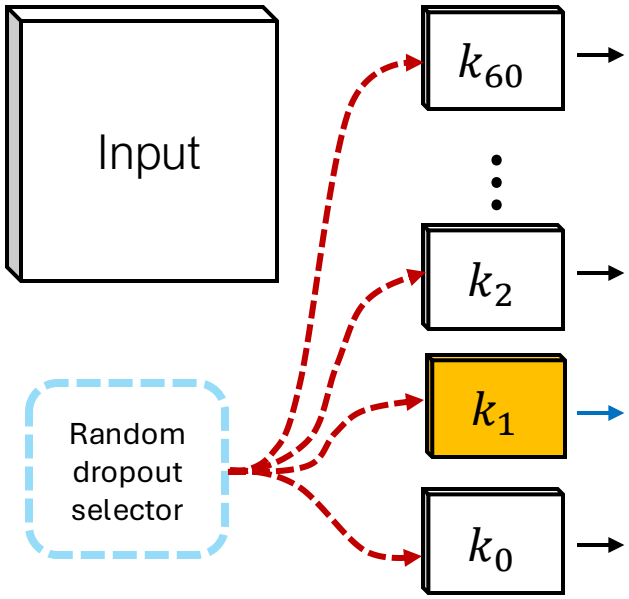
x'_1 and x'_2 represent their corresponding **adversarial images** with high loss values

Classification boundary

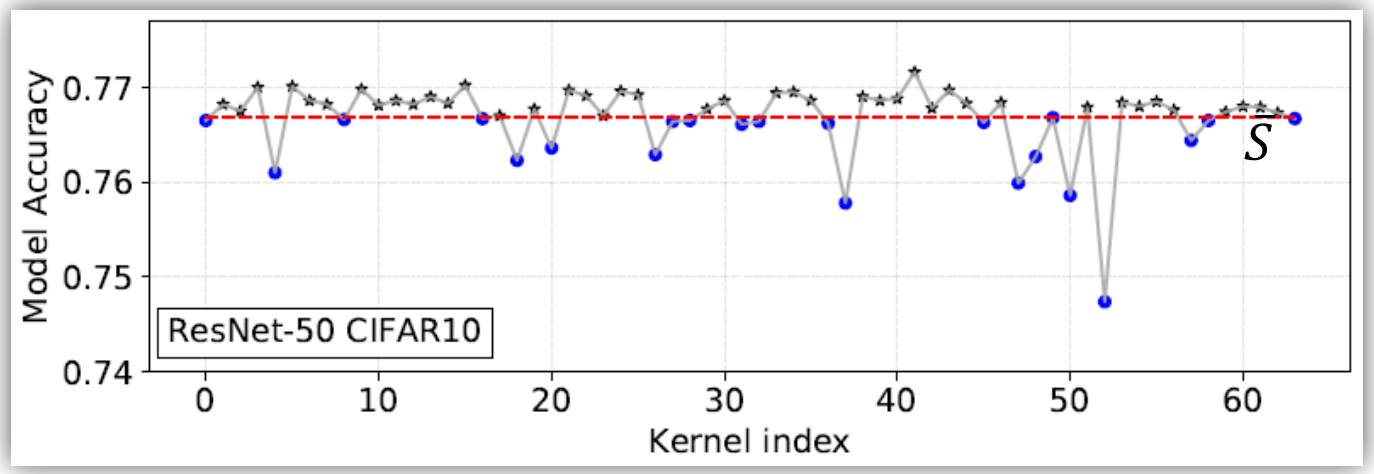
Image source: Xu et al 2021, *Medical Image Analysis*

Attacks on fragile neurons

We remove kernel from the first convolutional layer and define fragile nodes to be all nodes that reduce the model performance on the test set to be below the mean dropout performance.



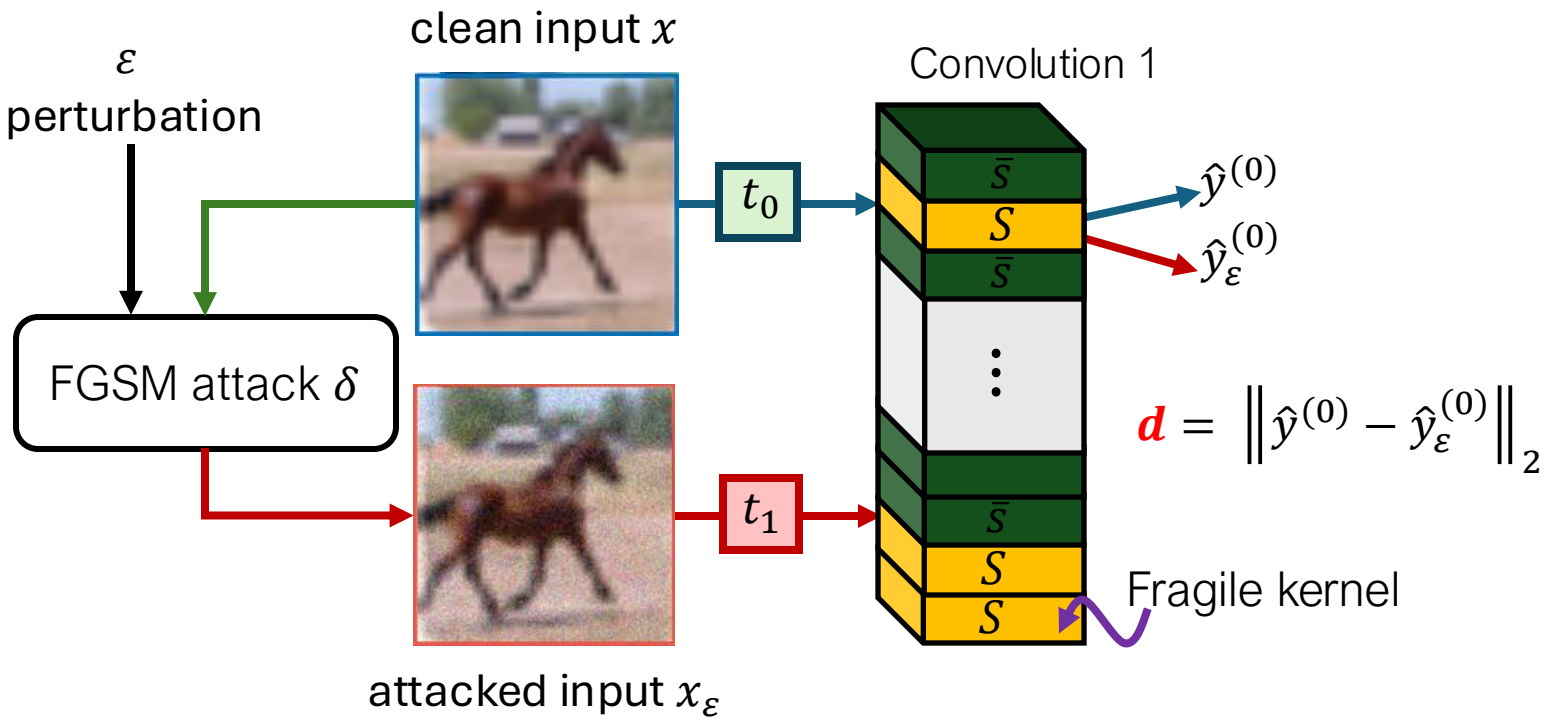
Nodal Dropouts

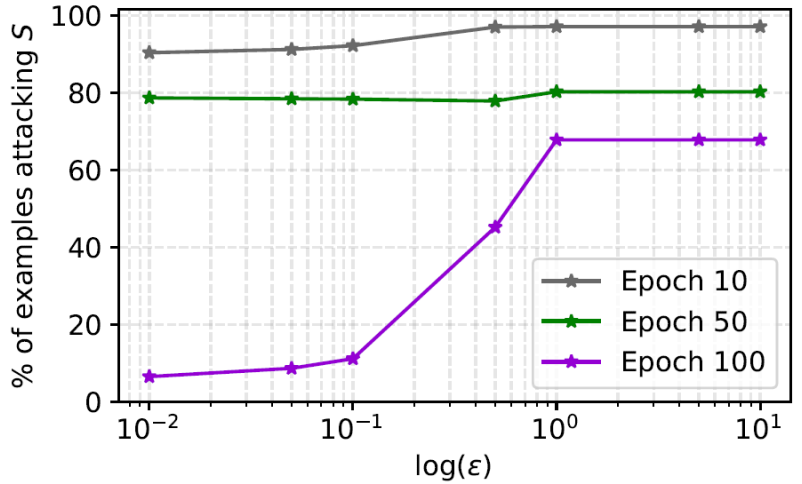


Fragile kernels (nodes) shown in blue (•) below mean/baseline DNN performance line in red and **null kernels** are shown in black star (★) above mean line in red

Adversarial targeting algorithm

We measure the **average magnitude difference d at the output** of the first convolutional layer, between fragile and non-fragile neurons, on both clean and adversarial inputs.

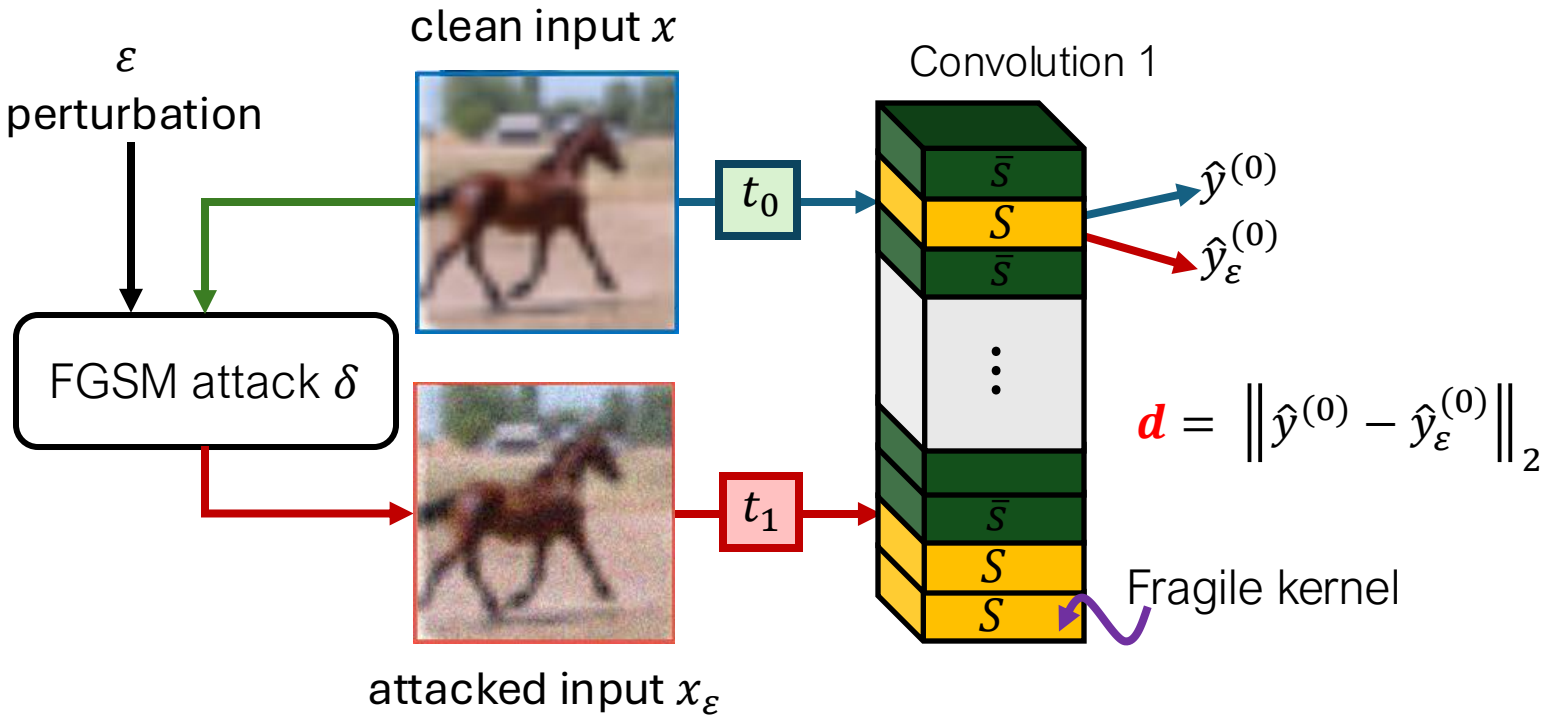


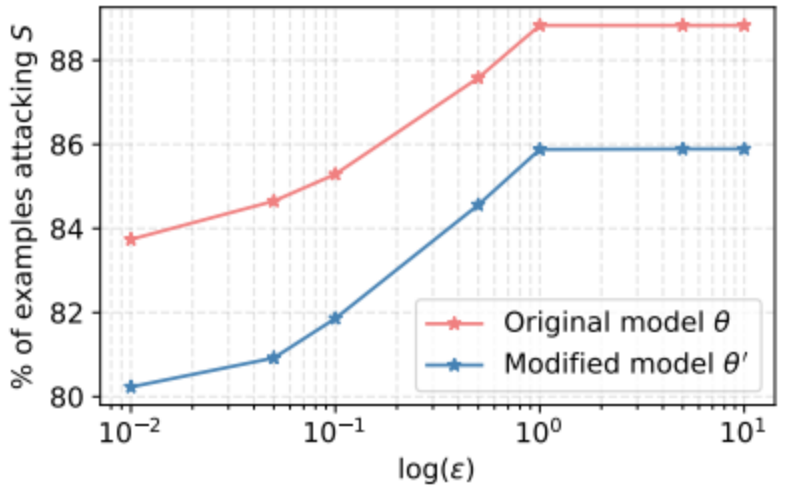


if avg. distance of fragile kernels S
greater than
avg. distance of non-fragile kernels $\bar{s}$
then
x_ε attacks fragile kernels

Adversarial targeting algorithm

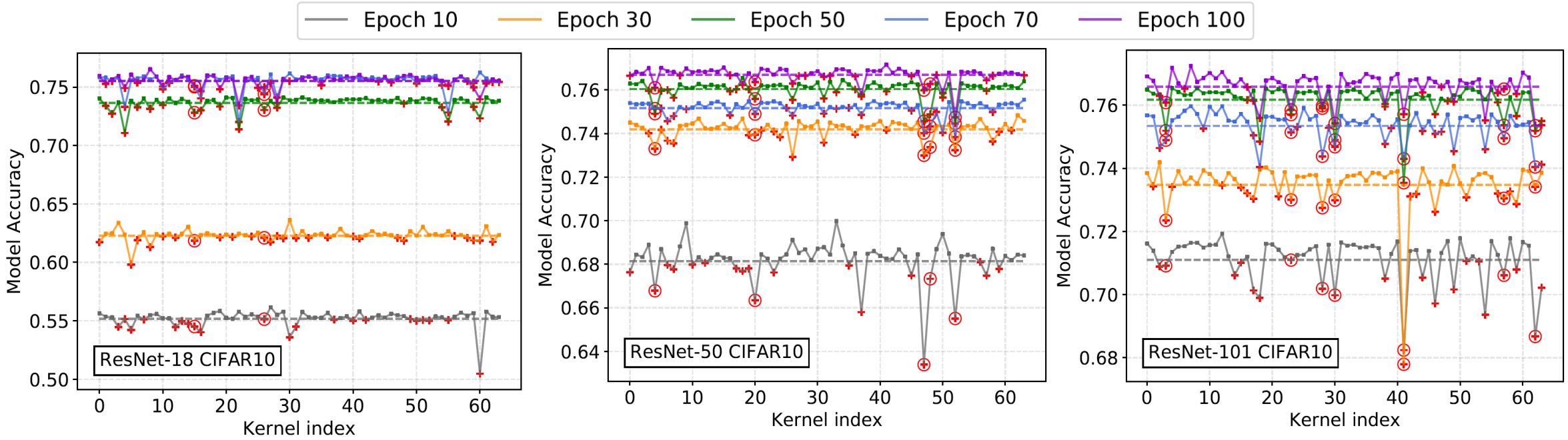
We measure the **average magnitude difference d at the output** of the first convolutional layer, between fragile and non-fragile neurons, on both clean and adversarial inputs.





if avg. distance of fragile kernels S
greater than
avg. distance of non-fragile kernels $\bar{s}$
then
x_ε attacks fragile kernels

Fragile kernels / neurons



Red crosses (+) represent fragile kernels and red circles around red crosses (⊕) represent kernels that have shown to be consistently fragile throughout the training phase for each model.

Challenges for DNN robustness

- **DNNs are susceptible to adversarial attacks** and thus any DNN prediction can be unreliable and vulnerable to an adversary.

- **How each component of a DNN behaves due to an adversarial attack is a lesser-known area of research**.

- Adversarial attacks on DNNs has been well studies on state-of-the-art datasets, however, **adversarial attacks on DNNs and their remedies has rarely been studied extensively**.

What can we promise for DNN robustness?

- We can use adversarial attacks to **identify the strengths and weaknesses** of DNN architectures.

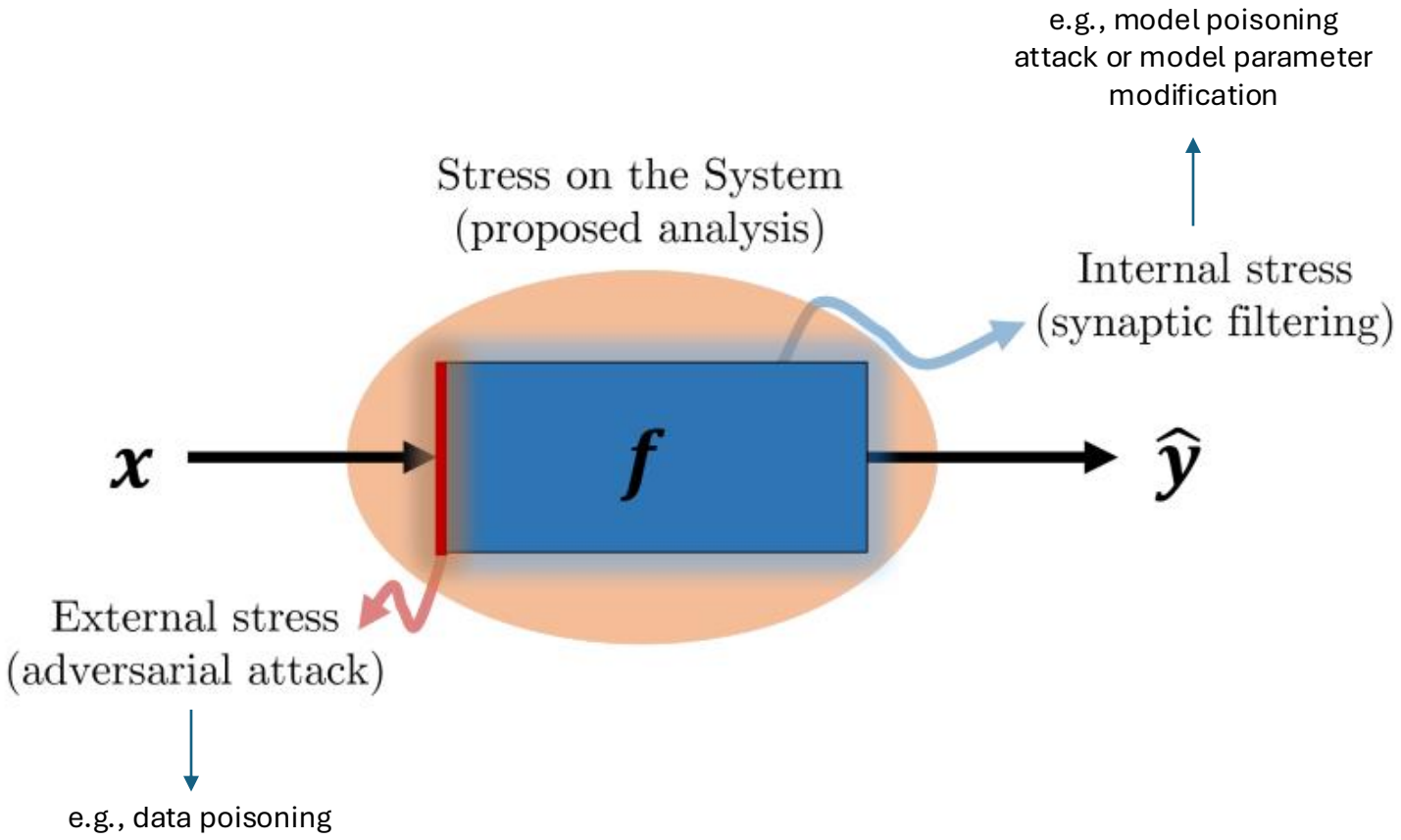
- Upon identifying the strengths and weaknesses of DNN architectures **we can improve the performance of DNNs against both adversarial attacks** and the clean dataset.

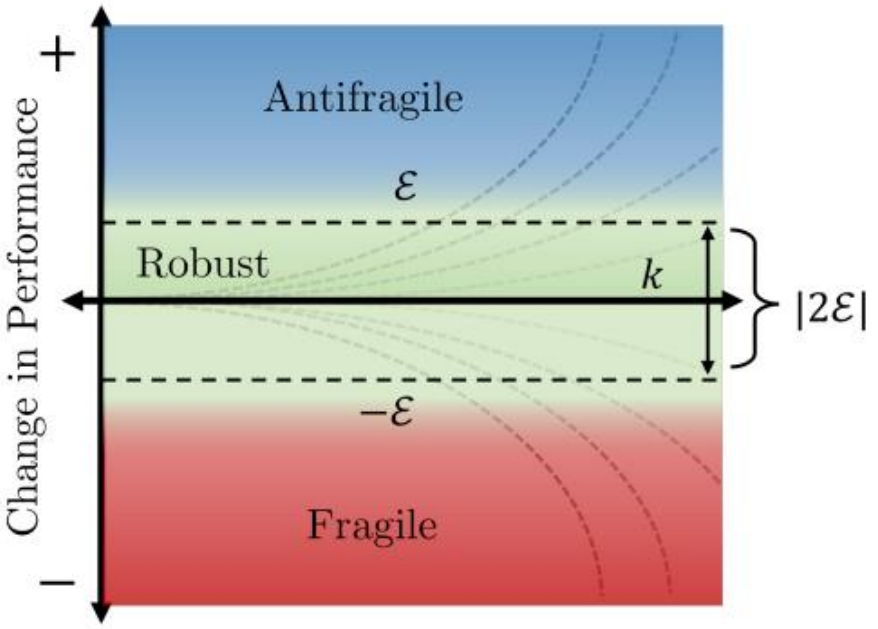
- DNNs **robustness analysis can develop stronger networks** that are capable of performing under sub-optimal conditions.

How can we ensure DNN robustness?

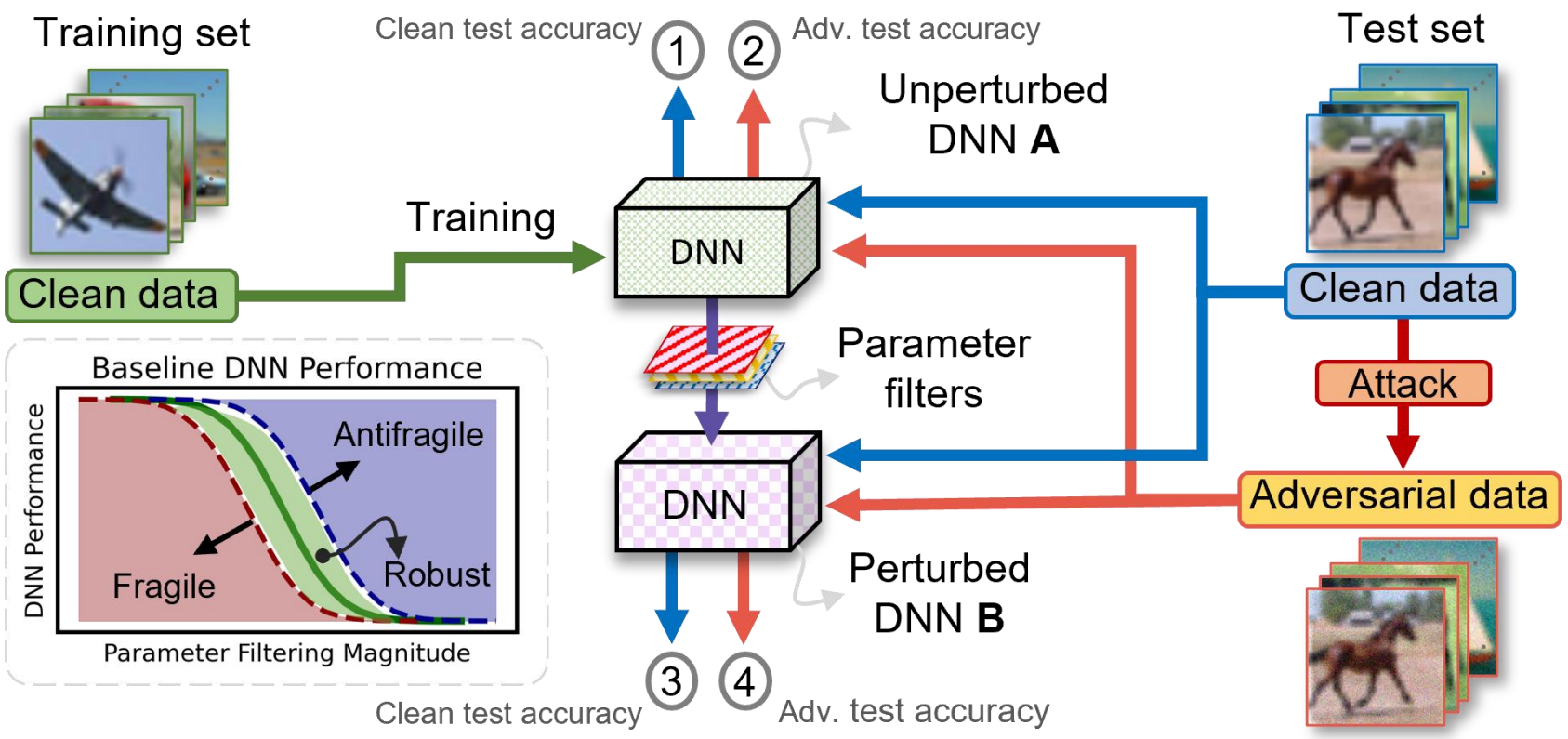
- Establish the **relationship between DNN parameters and adversarial attacks** to identify parameters that are targeted by the adversary.
- Formalise the notions of DNN parameter perturbations and adversarial attacks as **internal and external stressors on DNNs**.
- Define **fragility**, **robustness**, and **antifragility** in DNN to encapsulate parameter characterisations and
- Evaluate the effects of **only re-training parameters characterised as robust and antifragile (selective backpropagation).**

Deep learning and systems





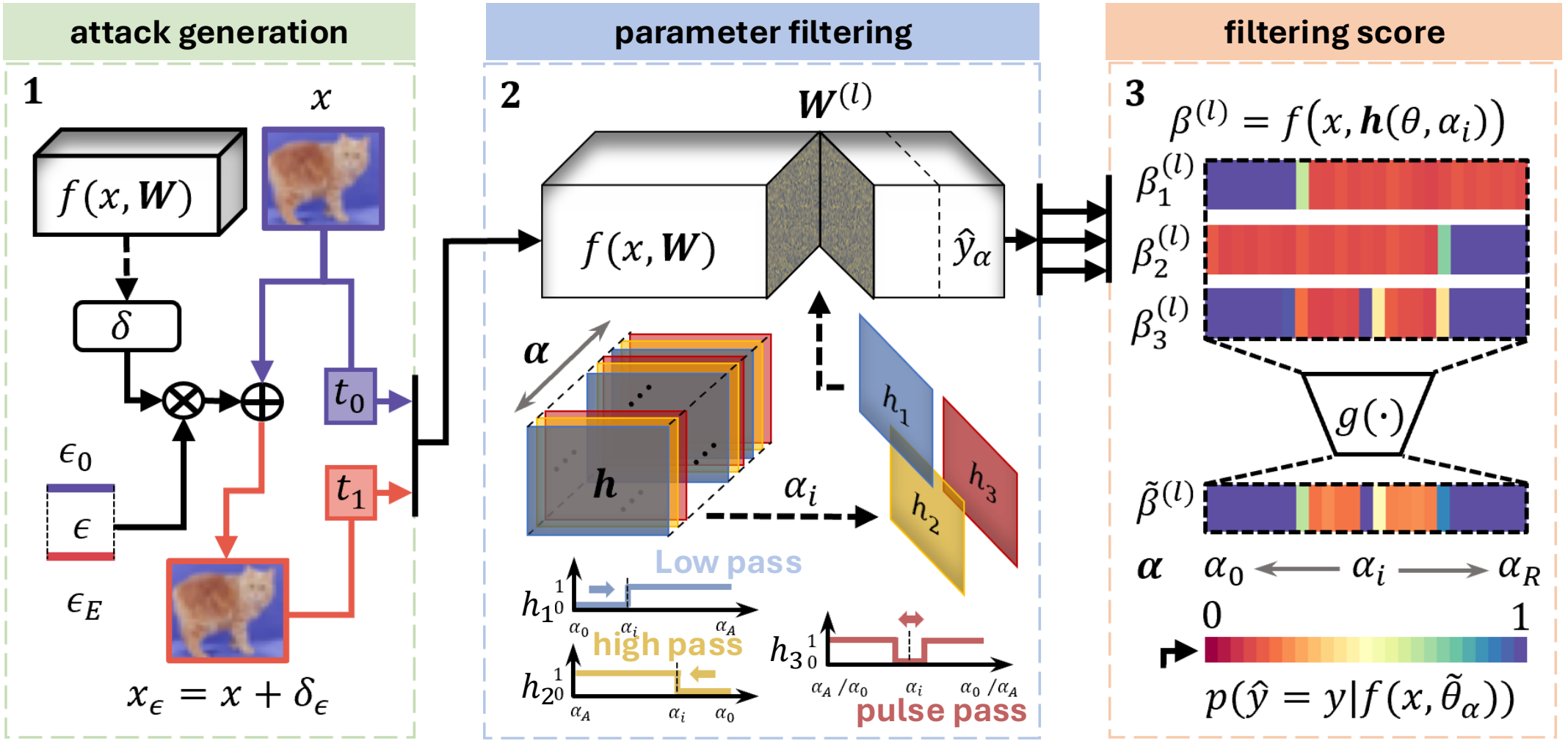
Fragility, robustness and antifragility



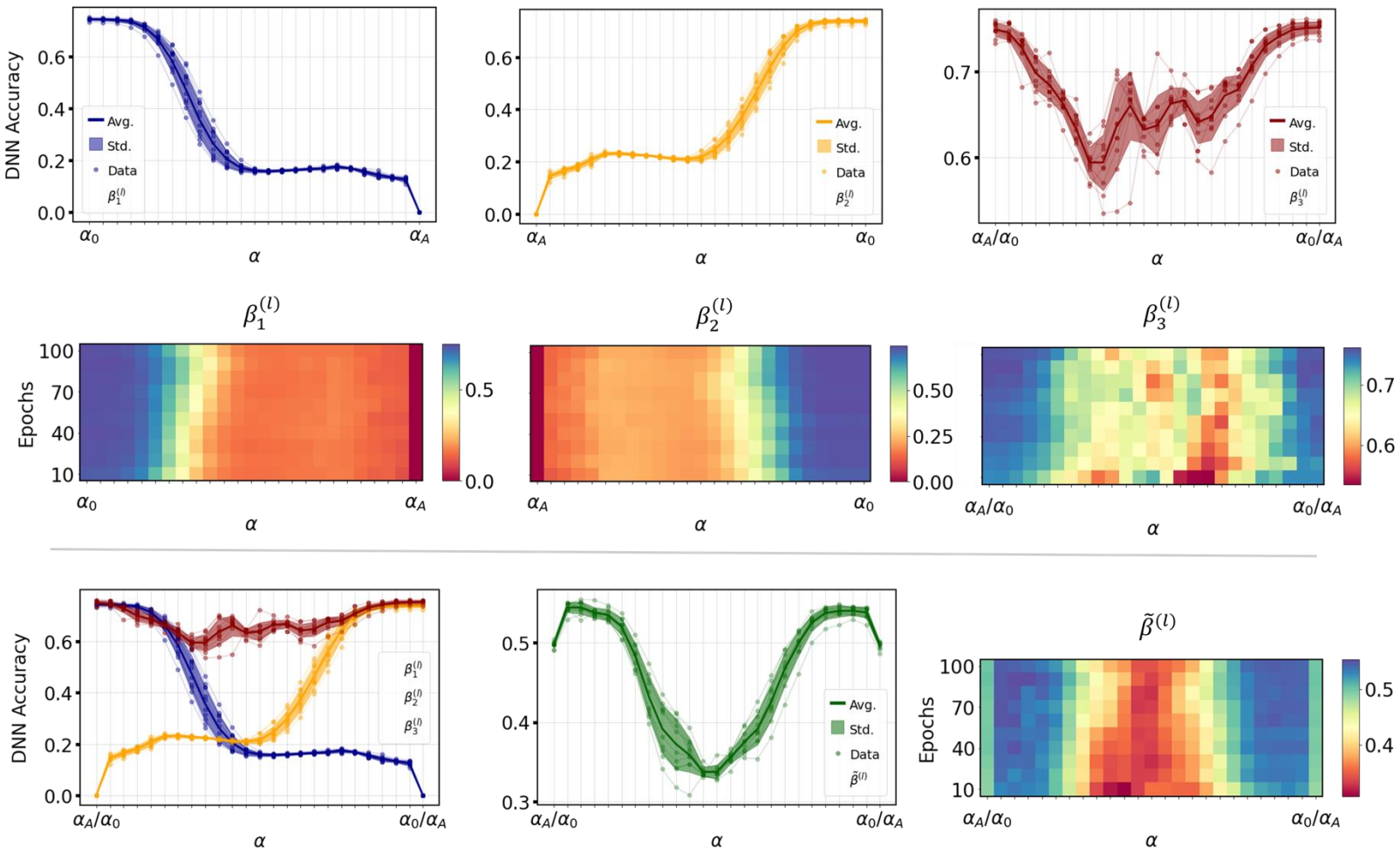
- a new method of parameter filtering (**synaptic filtering**)
- **synaptic filtering of all layers and parameters** of a DNN architecture.
- **compare clean and adversarial performance** of a regular DNN and perturbed DNN.
- **characterise** parameters as fragile, robust, and antifragile

Synaptic filtering algorithm

$$h_1(\theta, \alpha_i) = \begin{cases} 0 & \text{if } \theta \leq \alpha_i, \\ 1 & \text{otherwise} \end{cases}$$



Learning landscape (performance vs epoch vs filtering strength)



The influence of parameters varies as the network is trained and learns more dataset features.

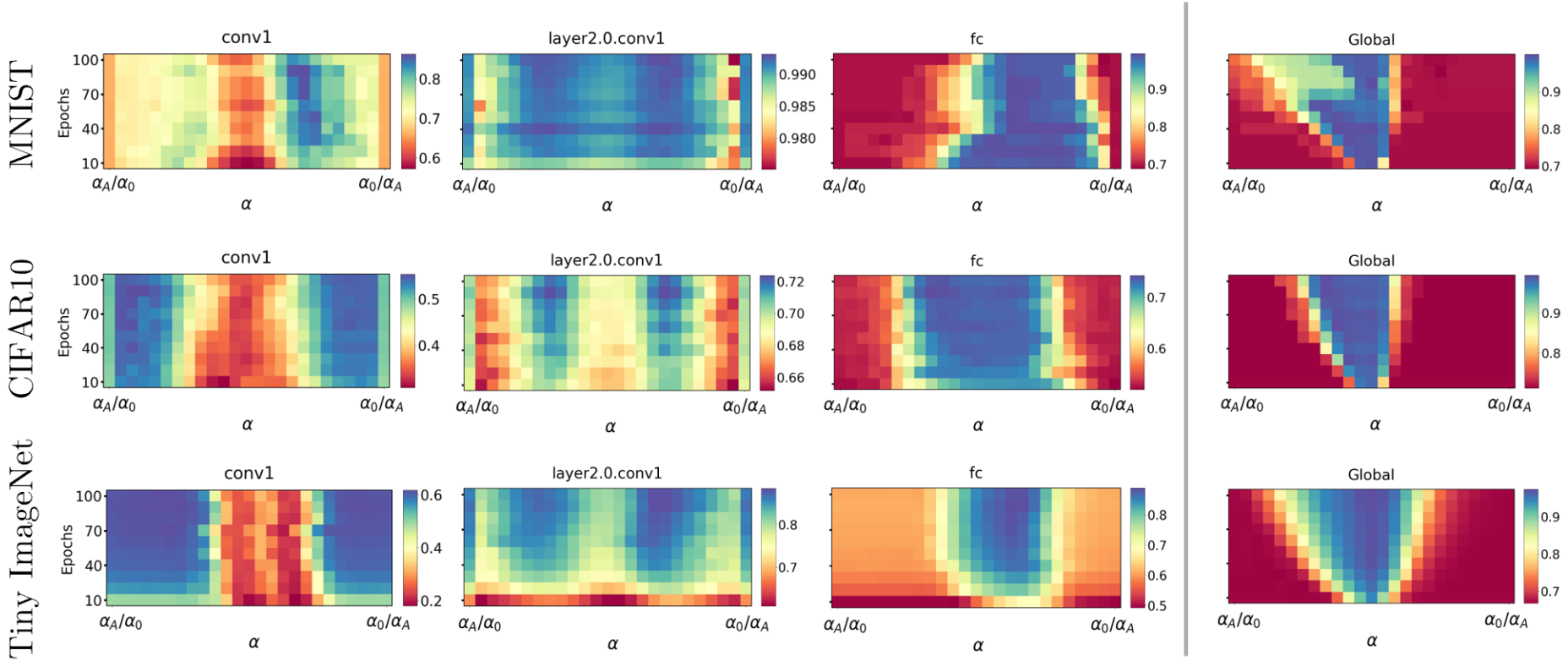
The three different filters h_1, h_2, and h_3 highlight different parameters as **influential** (■) and **non influential** (■) to DNN performance.

The combined performance highlights the parameters that are **most influential** (■) using all the three different filters.

Learning landscape (performance vs epoch vs filtering strength)

We show that the **same layer of a DNN has similar learning landscapes** for different datasets.

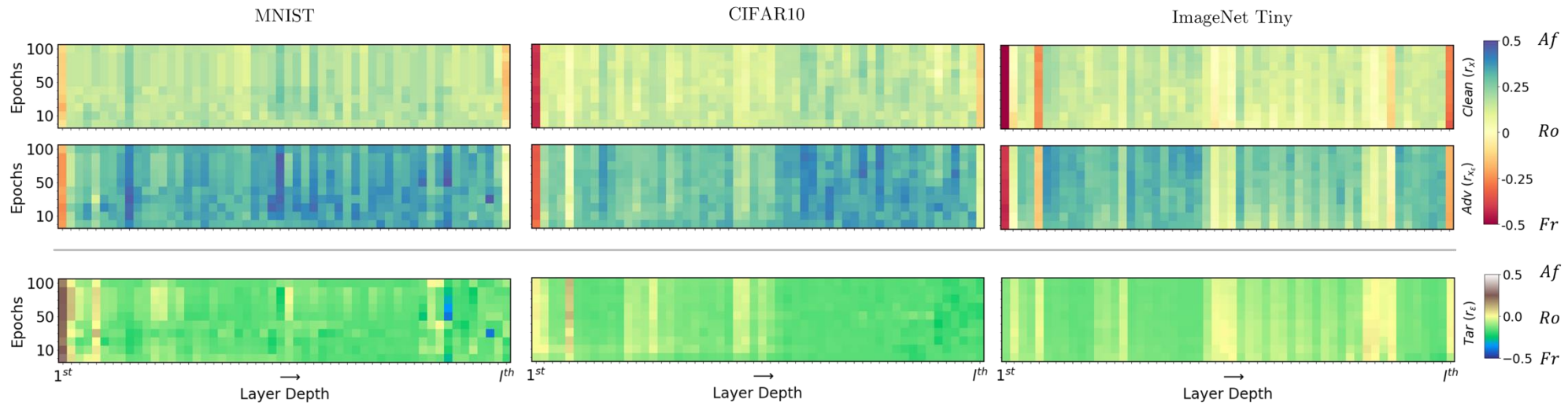
This shows that there are **invariant characteristics of DNN architectures**, even when applied to different datasets.



Different layers in the network show to have different characteristics when subjected to the parameter filters (internal stressor). The results are the combined responses using filters h_1, h_2, and h_3.

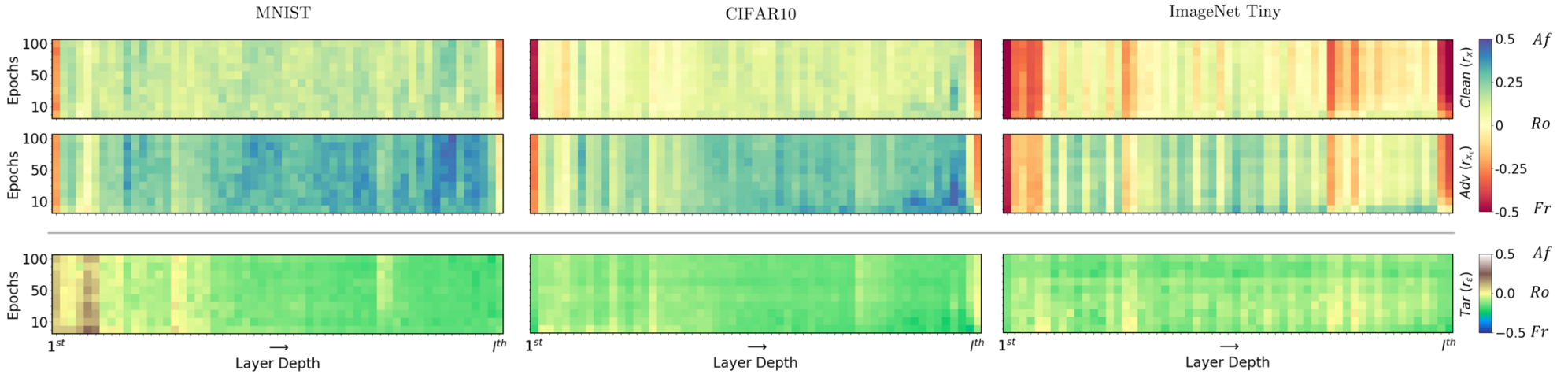
Parameter scores (layer-wise and epoch wise)

ResNet-50



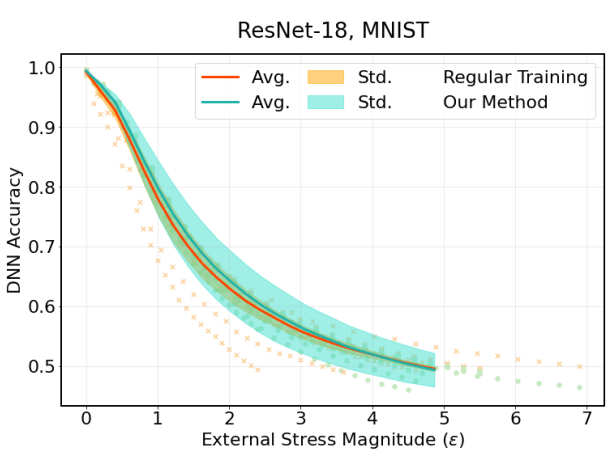
Periodic parameter characterisation shown for some networks.

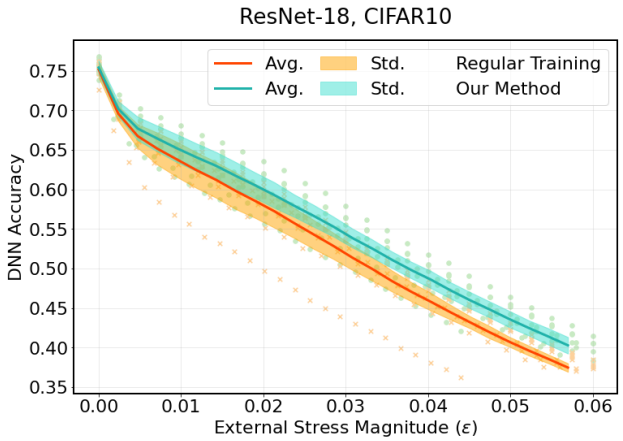
ShuffleNet V2 x1.0

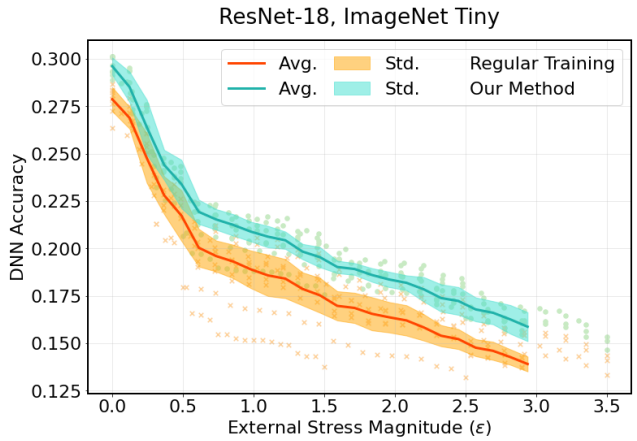


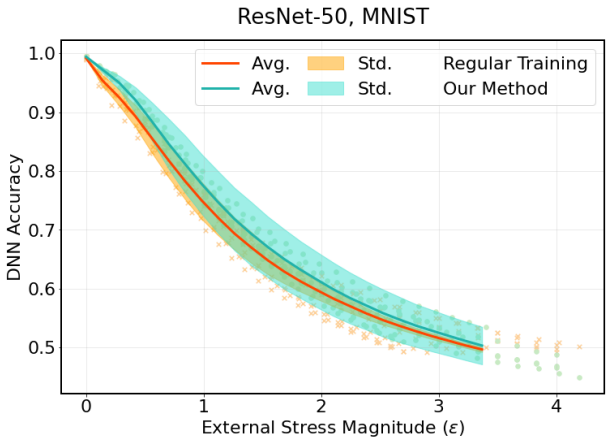
We say that fragile parameters are important to network performance.

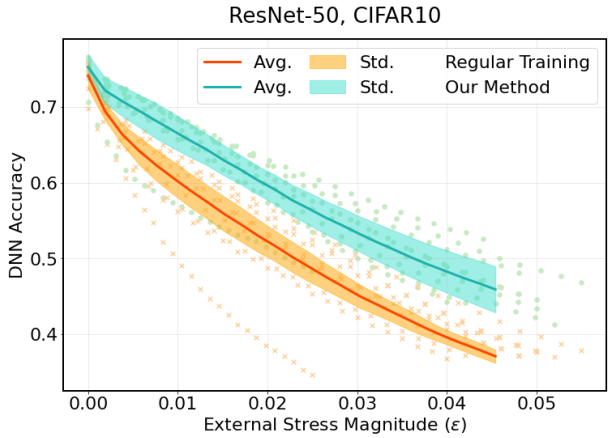
Selective backpropagation for DNN robustness

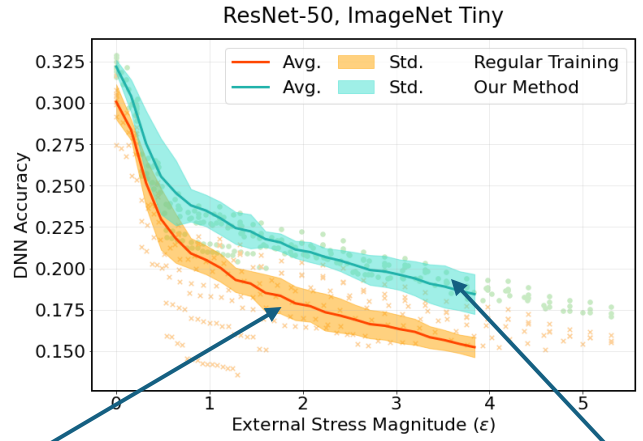










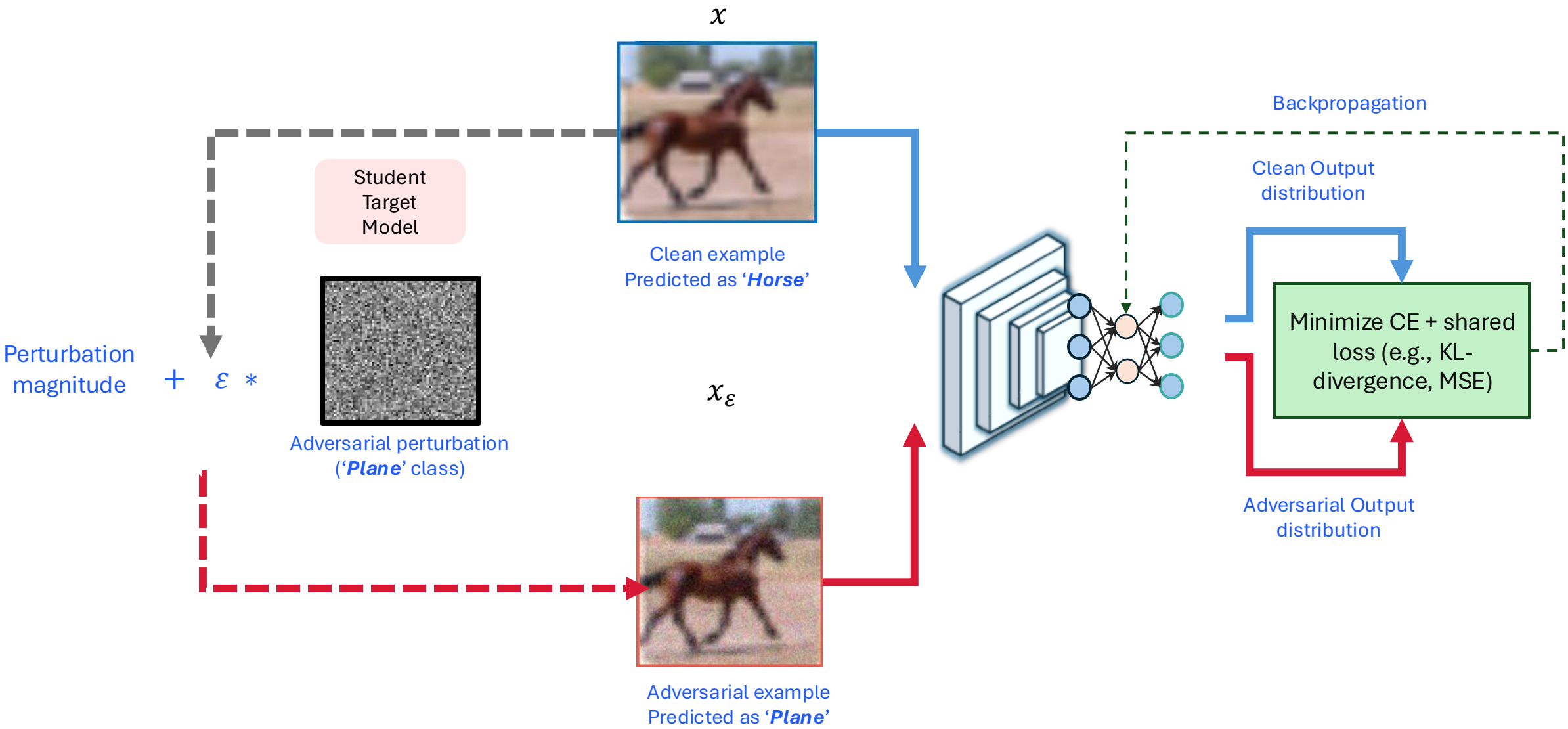


When we **retrain** networks at periodic intervals using only the characterised ***robust* and *antifragile*** layer parameters (selective backpropagation), we observe an **increase in adversarial performance**, and clean performance for some networks and datasets.

Regular training

Selective backpropagation

Adversarial training for DNN robustness

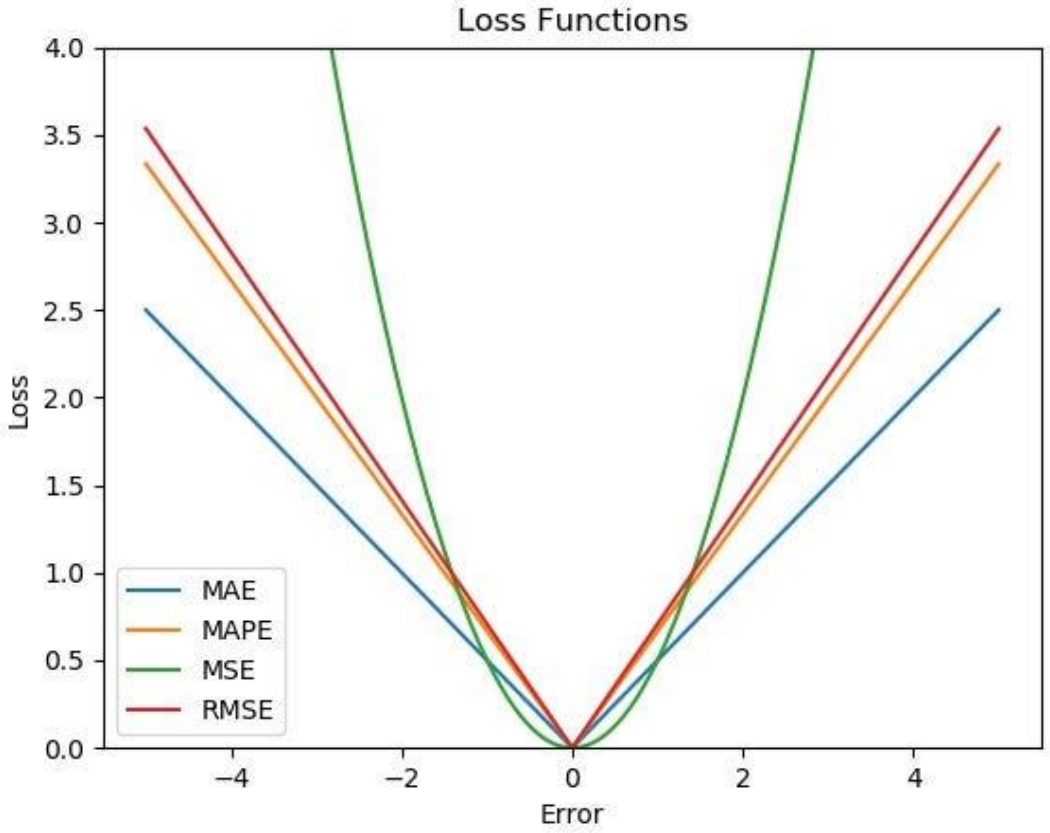


Loss functions

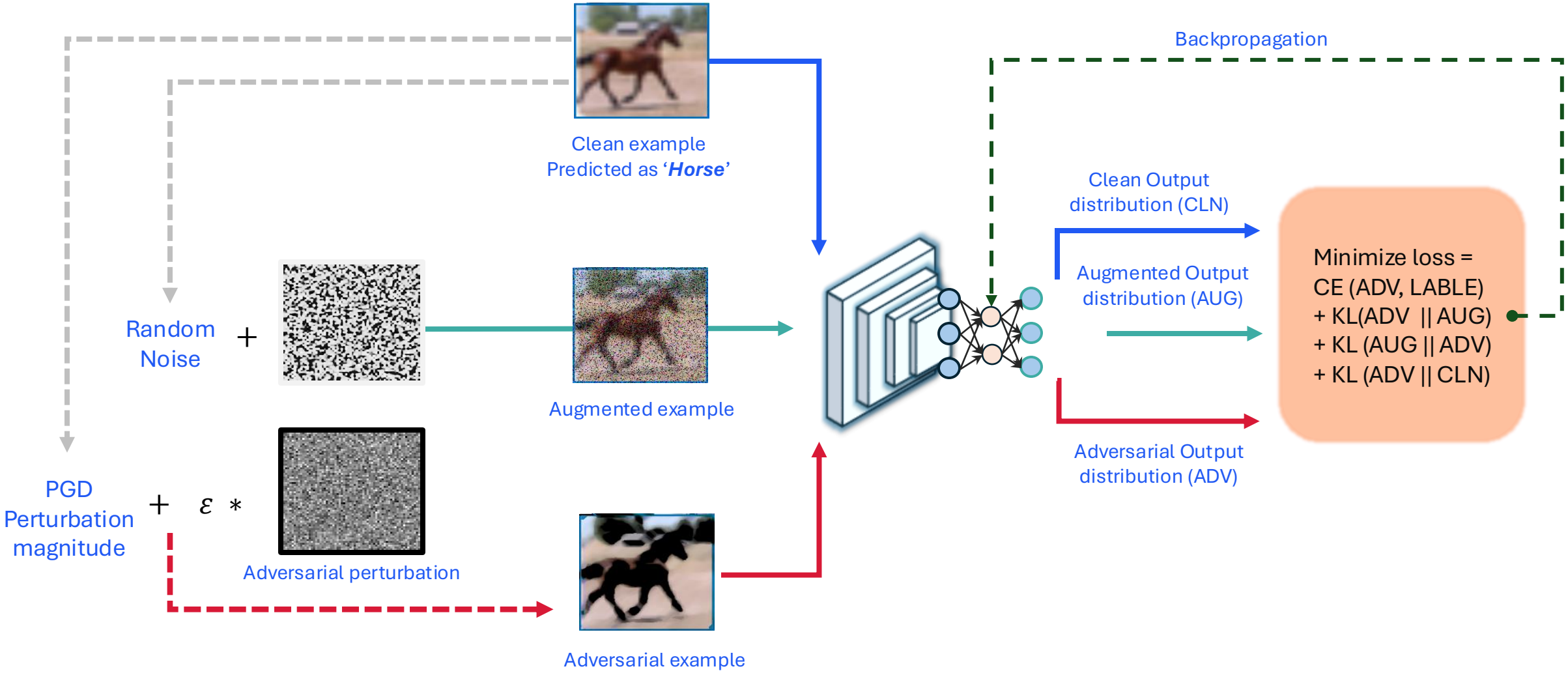
Mean Absolute Error (MAE) :
MAE calculates loss by considering all the errors on the same scale. Therefore, network will not be able to distinguish between them just based on MAE, and so, it's hard to alter weights during backpropagation.

Mean Squared Error (MSE) :
MSE helps converge to the minima efficiently, as the gradient reduces gradually. Ate the same time, extremely large loss may lead to a drastic jump during backpropagation, which is not desirable. MSE is also sensitive to outliers.

Root Mean Squared Error (RMSE) :
Less extreme losses even for larger values, however, near minima, the gradient change is abrupt

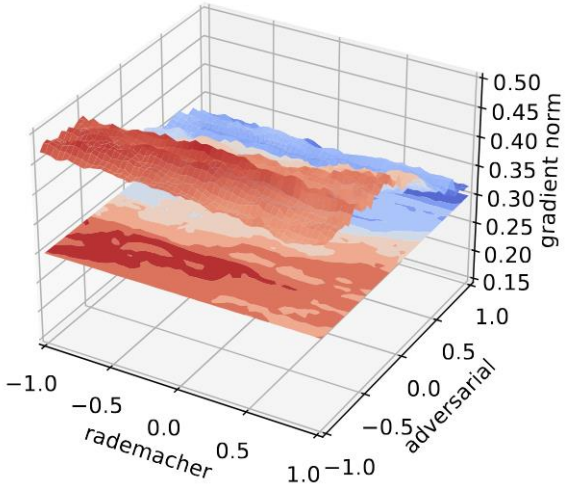


RegMix: Adversarial mutual and generalization regularization

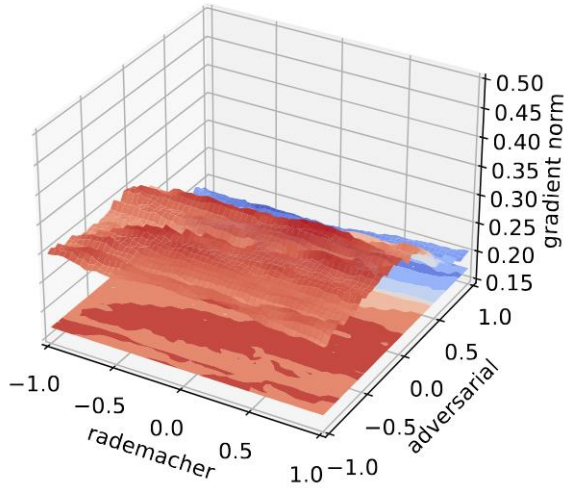


Loss landscape comparison

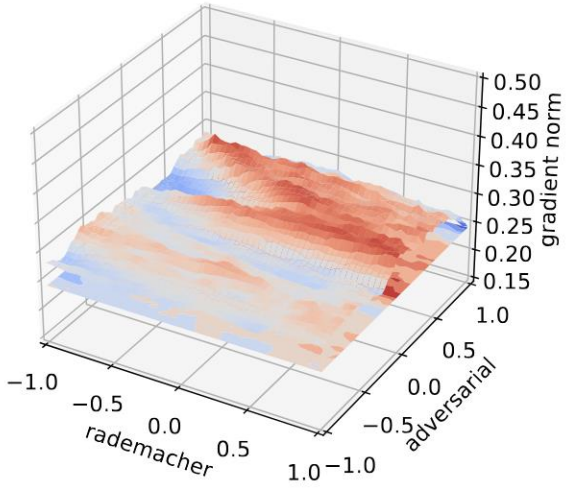
RegMix: Adversarial mutual and generalization regularization



(a) FGSM-PGI

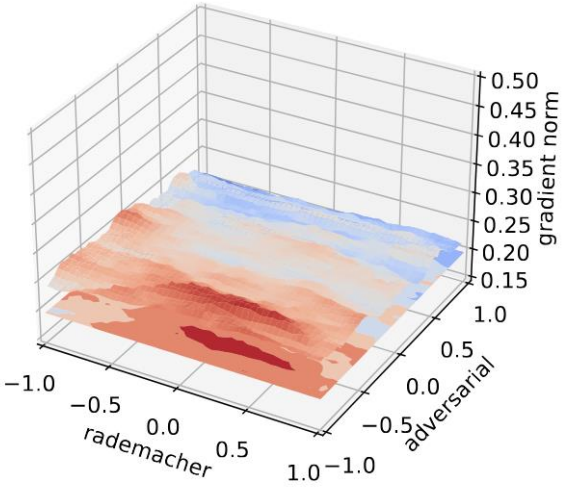


(b) FGSM-PGK



(c) FGSM-AMR (Ours)

Adversarial
Mutual
Regularization



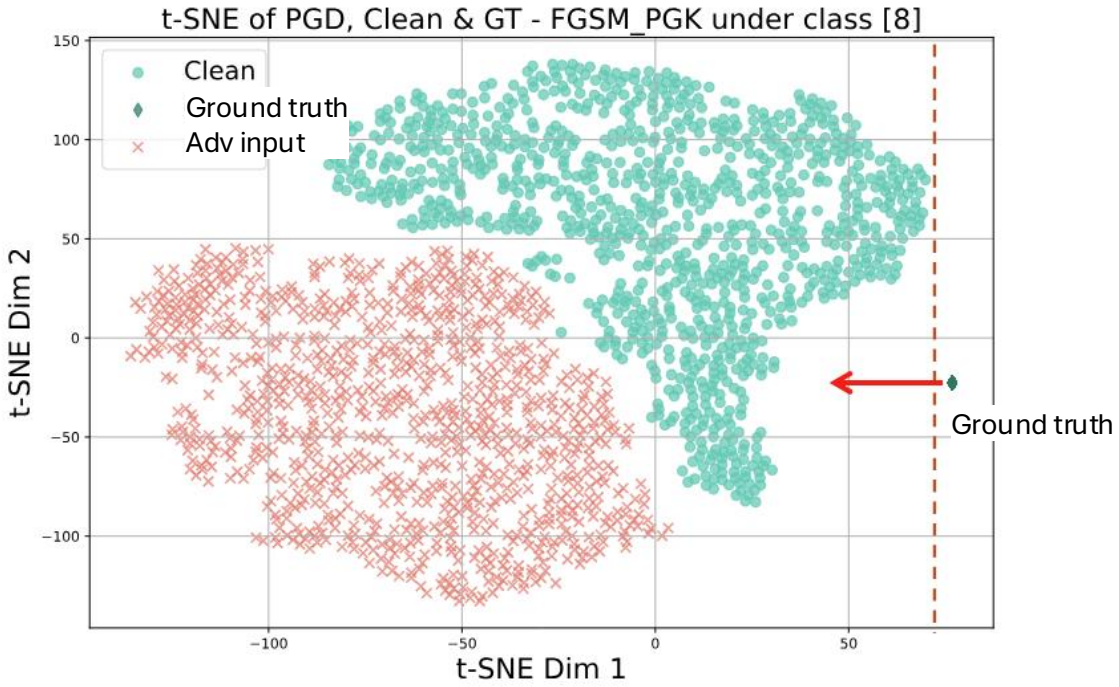
(d) FGSM-AGR (Ours)

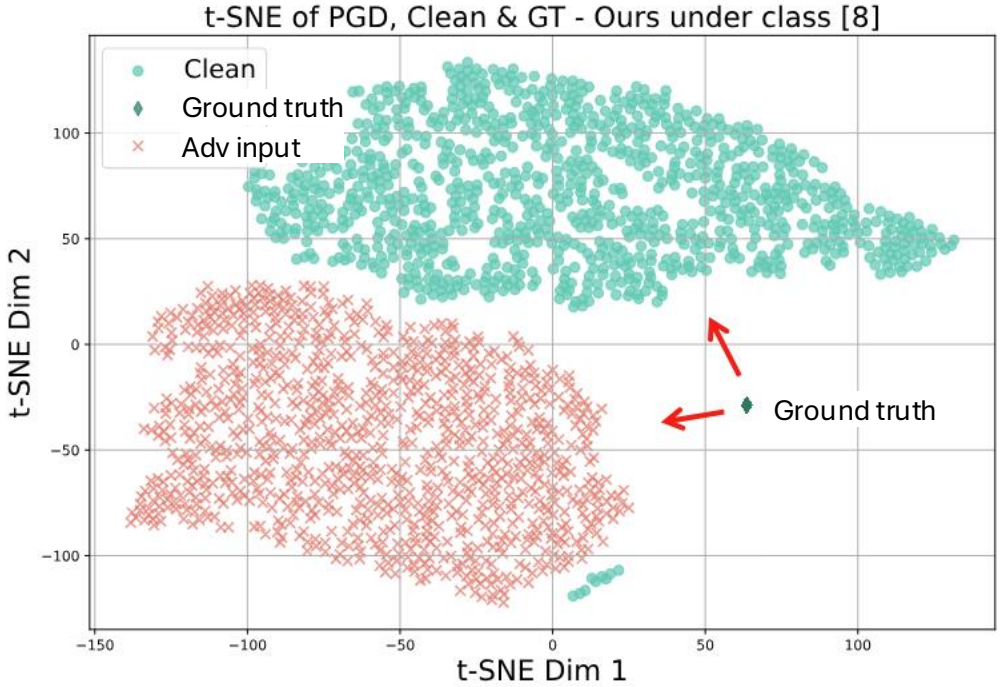
Adversarial
Generalization
Regularization

Classification Visualisation

RegMix: Adversarial mutual and generalization regularization

Plot: Predicted adversarial and clean probability distribution





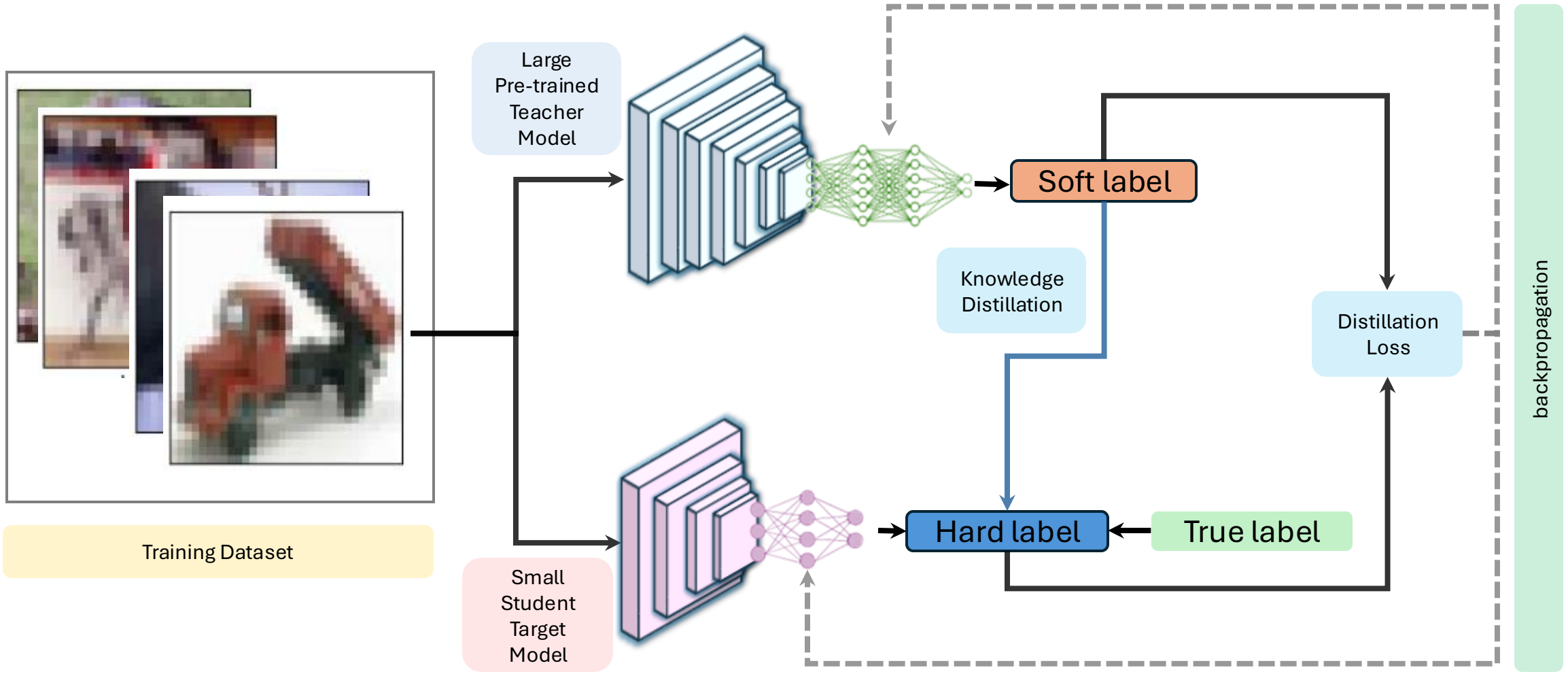
Performance

RegMix: Adversarial mutual and generalization regularization

WideResNet-34-10 on CIFAR-100 dataset

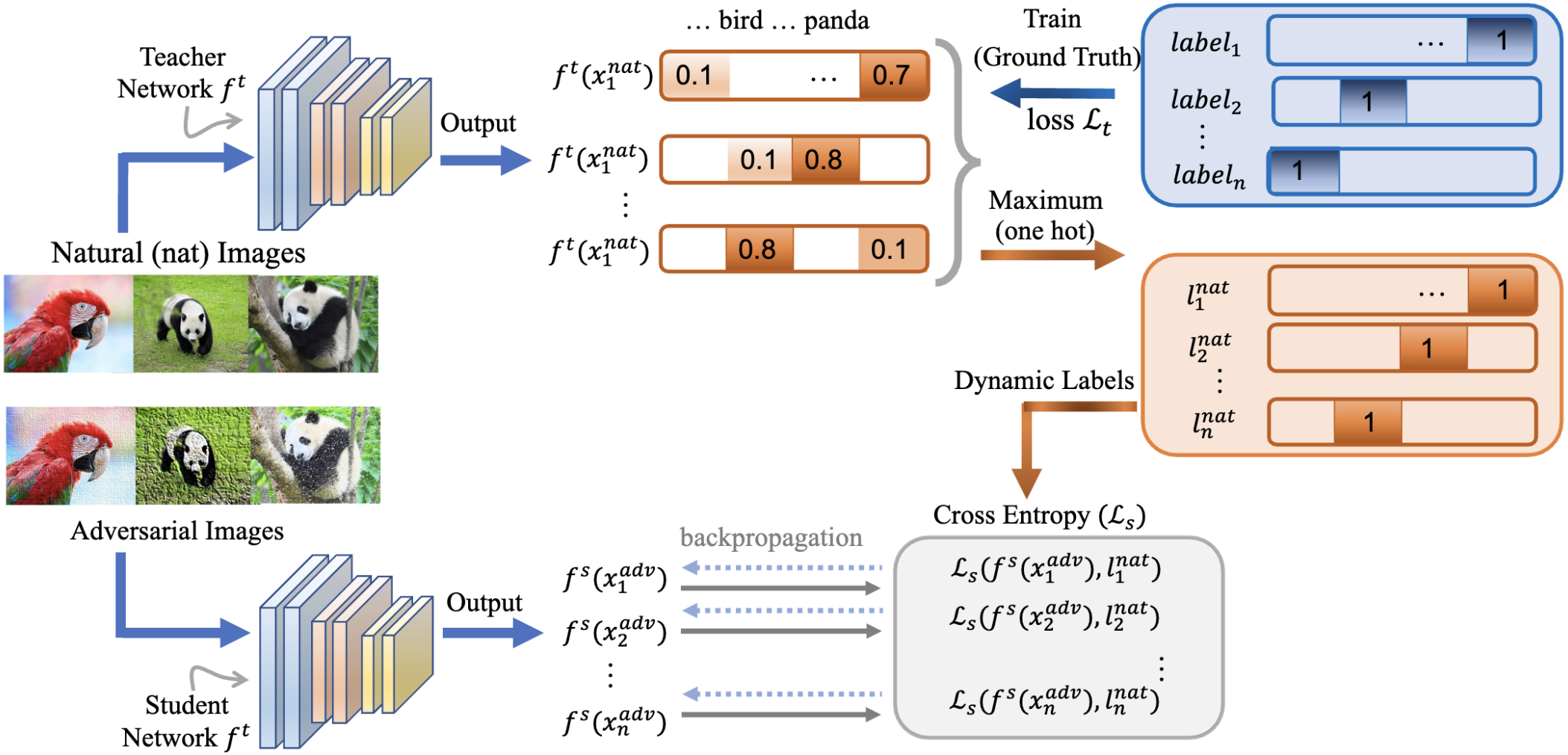
Method	Clean Best/Last	PGD-10 Best/Last	PGD-20 Best/Last	PGD-50 Best/Last	C&W Best/Last	AA Best/Last
PGD-AT [56]	57.52/57.50	29.60/29.54	28.99/29.00	28.87/28.90	28.85/27.60	25.48/25.58
FGSM-RS [68]	49.85/60.55	22.47/0.45	22.01/0.25	21.82/0.19	20.55/0.25	18.29/0.00
FGSM-CKPT [35]	**60.93/60.93**	16.58/16.69	15.47/15.61	15.19/15.24	16.40/16.60	14.17/14.34
FGSM-SDI [33]	60.67/60.82	31.50/30.87	30.89/30.34	30.60/30.08	27.15/27.30	25.23/25.19
NuAT [63]	59.71/59.62	27.54/27.07	23.02/22.72	20.18/20.09	22.07/21.59	11.32/11.55
GAT [62]	57.01/56.07	24.55/23.92	23.80/23.18	23.55/23.00	22.02/21.93	19.60/19.51
FGSM-GA [2]	54.35/55.10	22.93/20.04	22.36/19.13	22.20/18.84	21.20/18.96	18.88/16.45
Free-AT (m=8) [59]	52.49/52.63	24.07/22.86	23.52/22.32	23.36/22.16	21.66/20.68	19.47/18.57
FGSM-PGI [30]	58.78/58.81	31.78/31.60	31.26/31.06	31.14/30.88	28.06/27.72	25.67/25.42
FGSM-PGK [31]	56.27/58.13	33.15/32.38	32.85/31.90	32.83/31.87	28.39/27.95	26.86/26.35
FGSM-SAR (ours)	56.08/55.71	33.26/33.06	32.93/32.86	32.84/32.68	28.64/28.89	27.27/27.22
FGSM-AGR (ours)	53.57/53.57	**33.29/33.29**	**33.02/33.02**	**32.95/32.95**	**28.91/28.91**	**27.42/27.42**

Knowledge distillation for DNN robustness



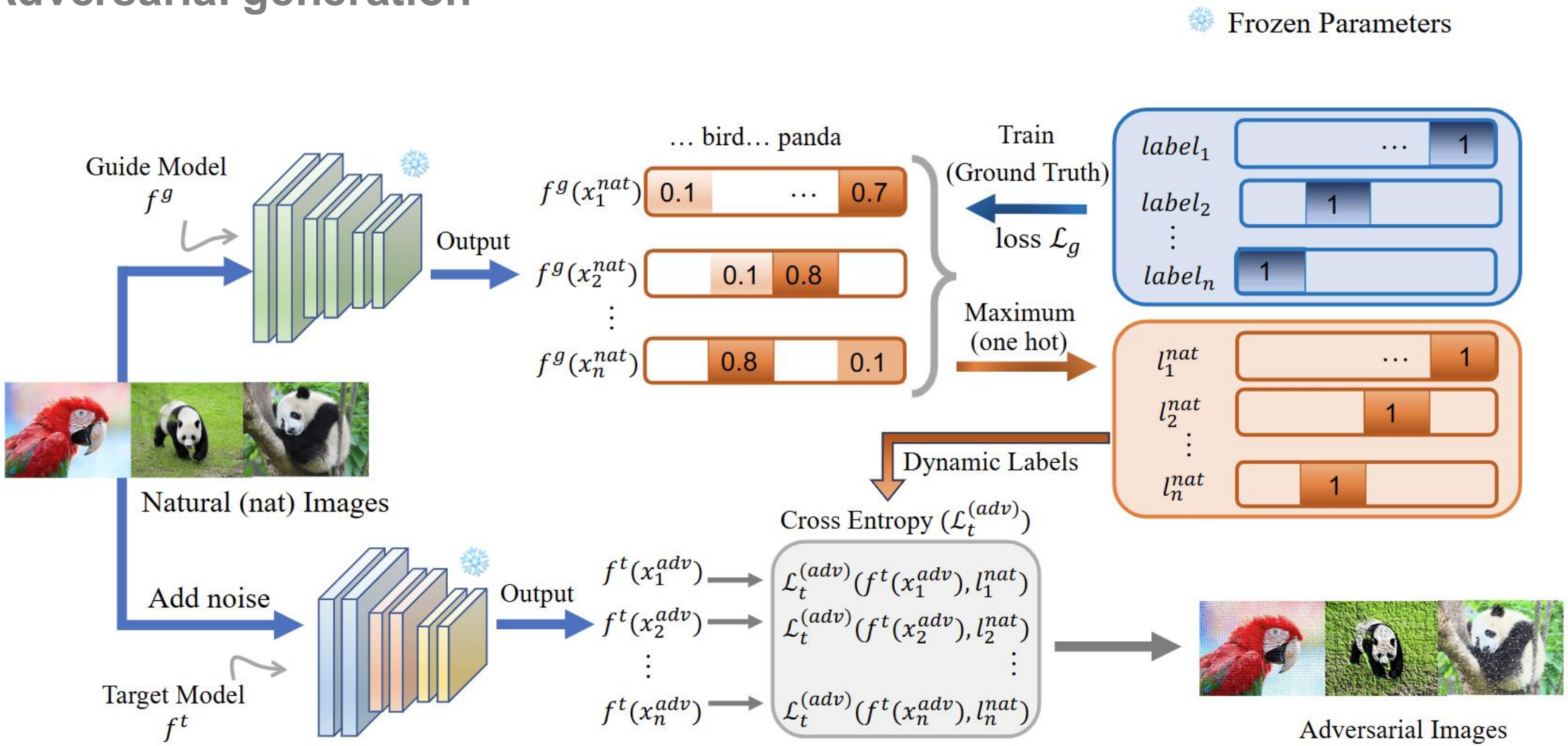
DynAT: Dynamic Label Adversarial Training

Knowledge distillation framework



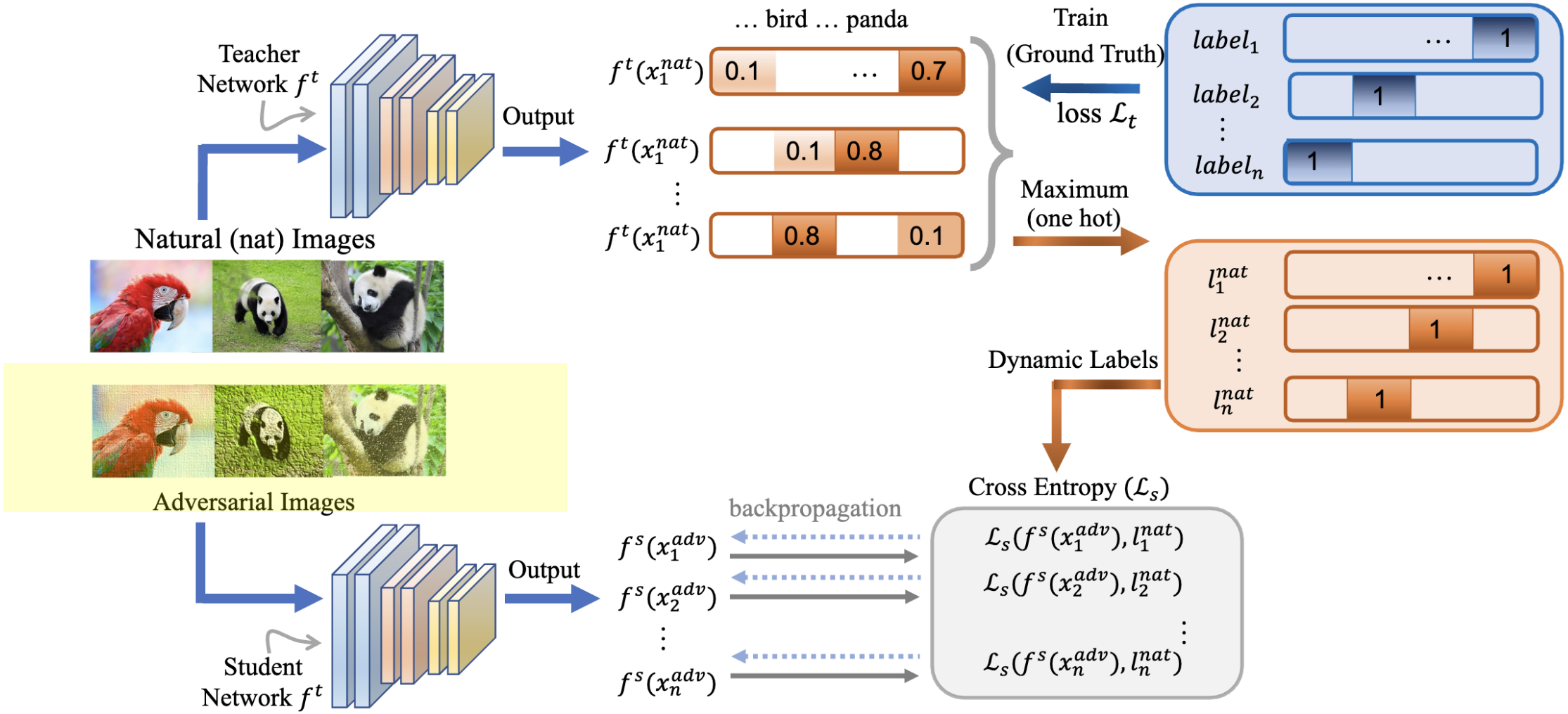
DynAT: Dynamic Label Adversarial Training

Adversarial generation



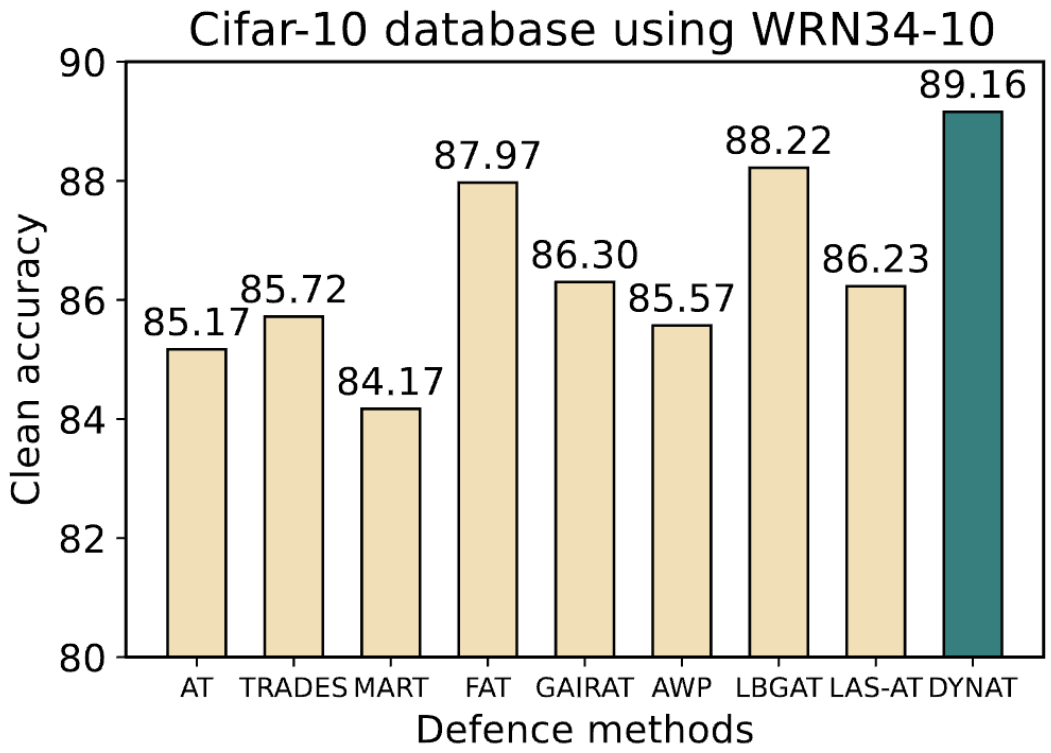
DynAT: Dynamic Label Adversarial Training

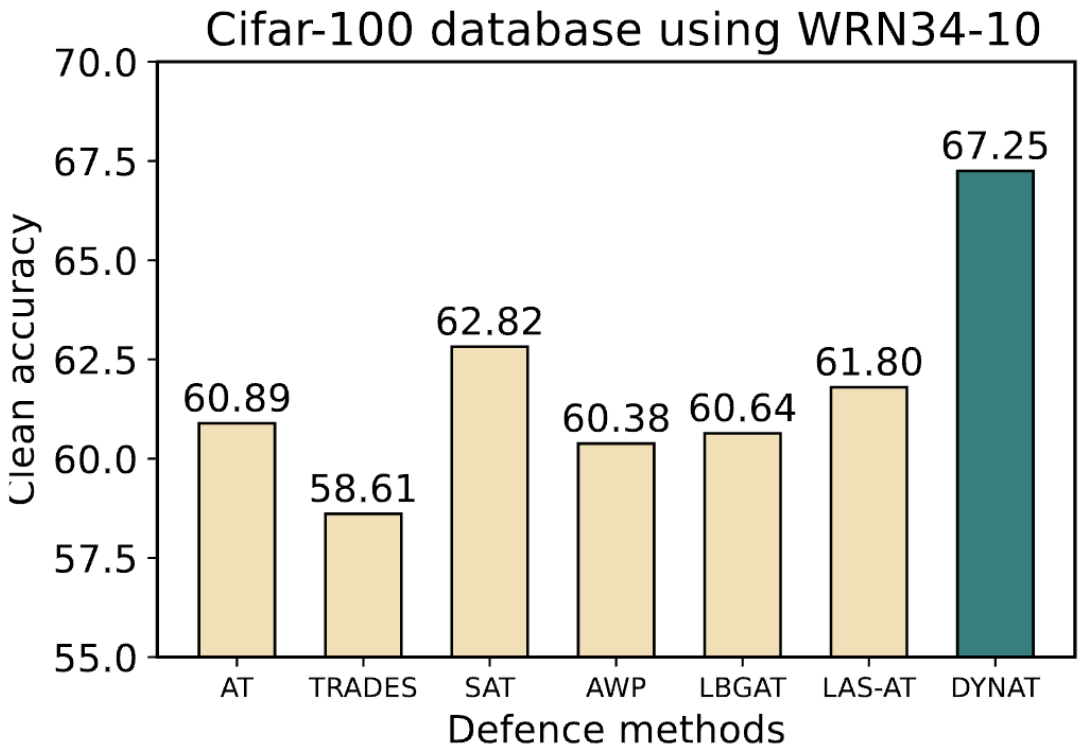
Knowledge distillation framework



Performance

Comparison with other typical defense methods





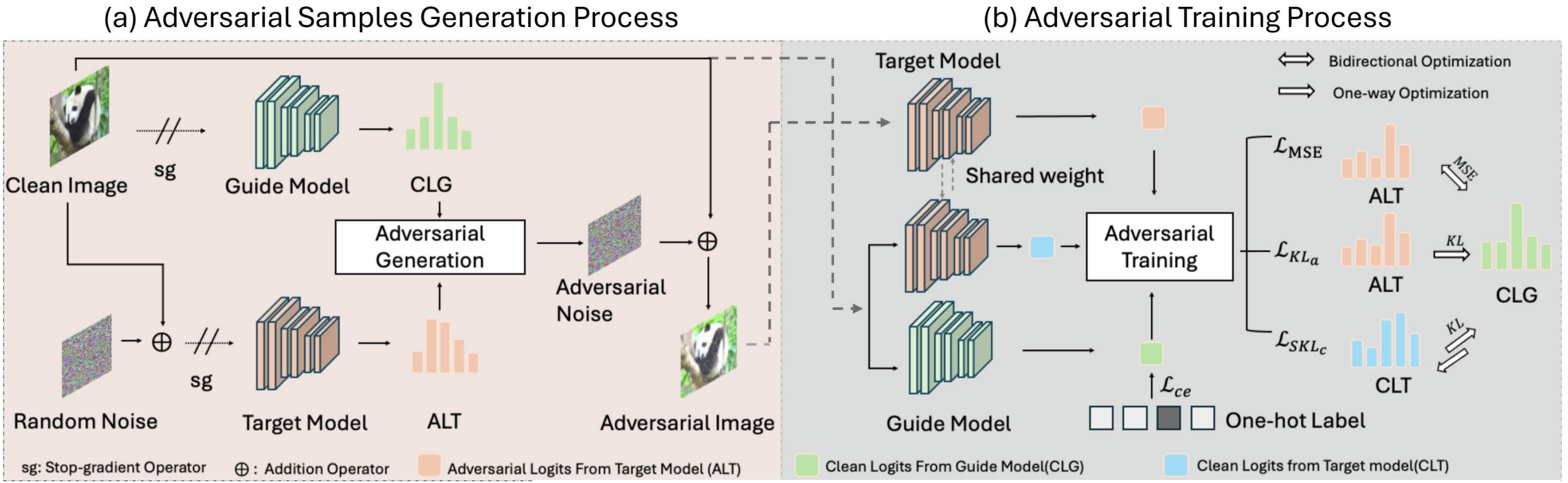
Performance

Comparison with other defense methods

WideResNet-34-10 on CIFAR-10 dataset

	Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
Others	PGD-AT	60.89	32.19	31.69	31.45	30.1	27.86
	TRADES	58.61	29.20	28.66	28.56	27.05	25.94
	SAT	<u>62.82</u>	28.1	27.17	26.76	27.32	24.57
	AWP	60.38	34.13	33.86	33.65	31.12	28.86
	LBGAT	60.64	35.13	<u>34.75</u>	34.62	30.65	29.33
Ours	DYNAT	**67.25**	28.03	26.97	26.81	26.62	24.10
	DYNAT-AWP ($\alpha = 1$)	62.29	<u>35.45</u>	**35.09**	<u>34.92</u>	<u>31.50</u>	**30.20**
	DYNAT-Inner-AWP ($\alpha = 1$)	58.87	**35.61**	**35.09**	**35.05**	**32.10**	<u>29.70</u>

D^2R: Dual regularization loss with adversarial generation



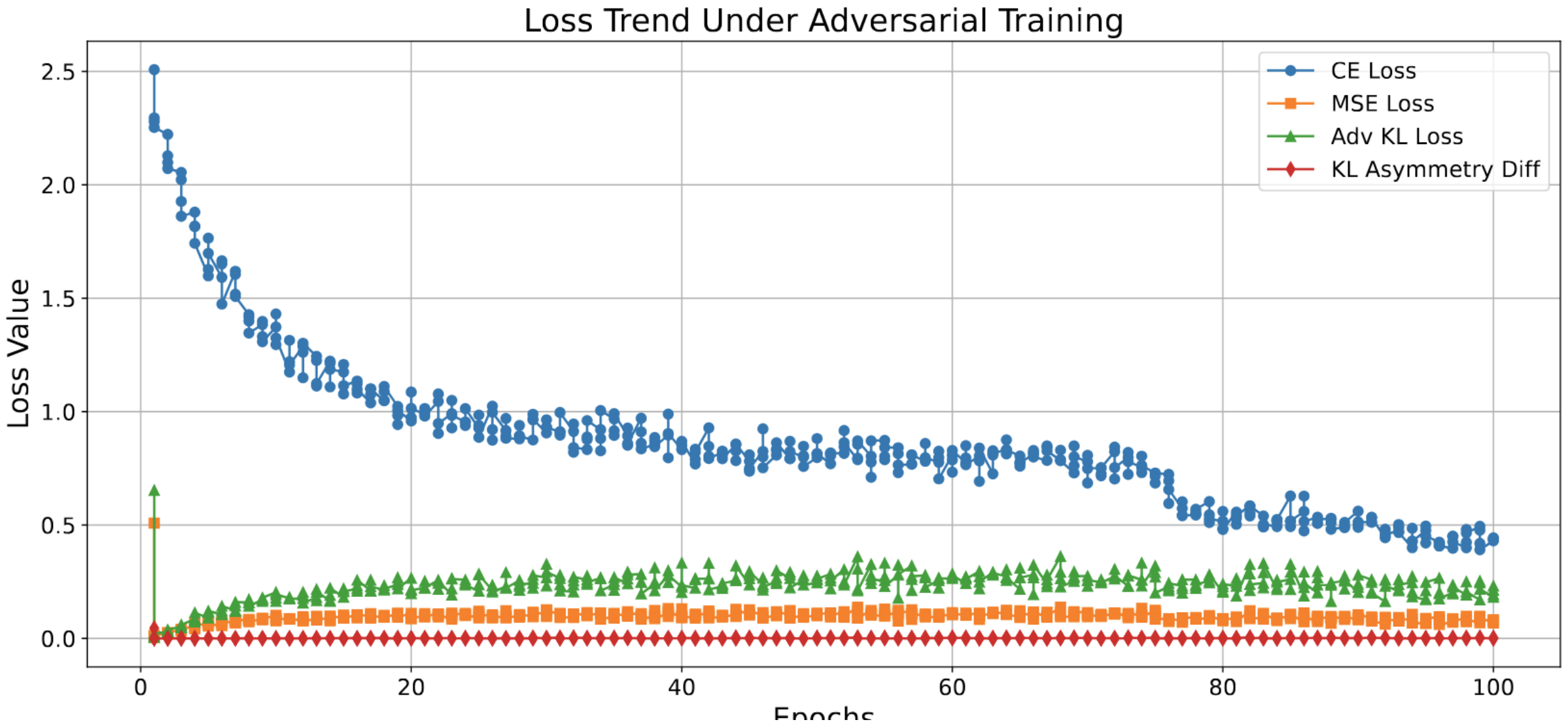
Guide Model (Clean output)

Target Model (Adversarial output)

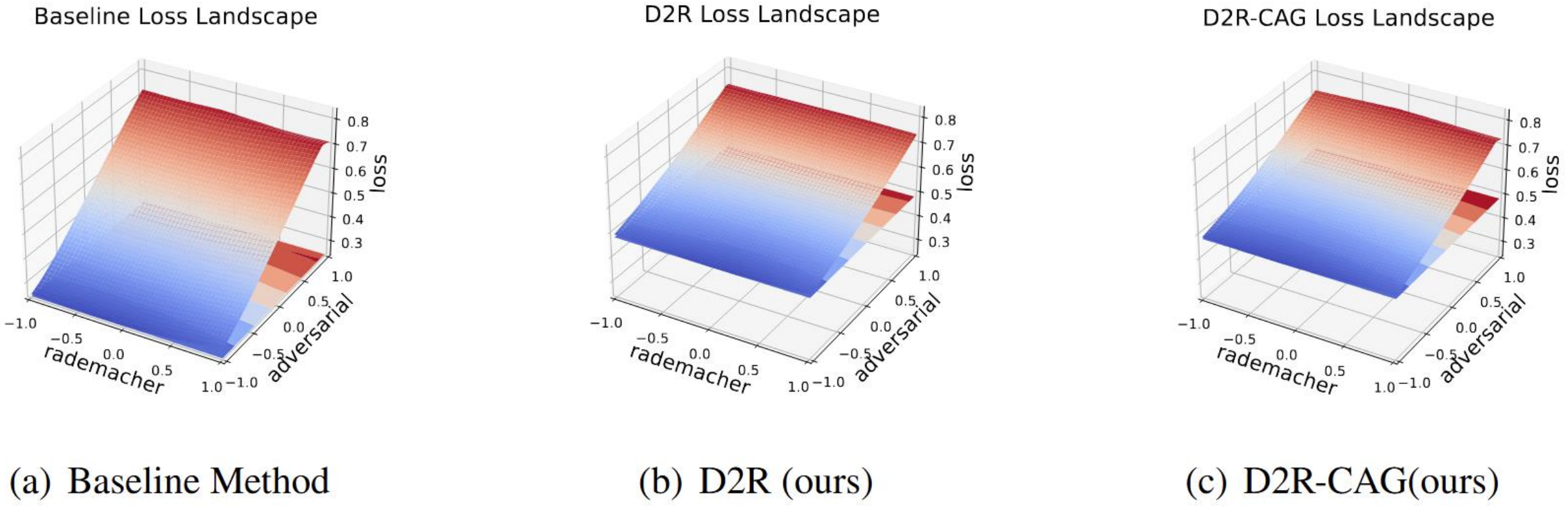
Target Model (Clean output)

$$\mathcal{L}_{\mathrm{D2R}}(x, y) = \min_{\theta_g, \theta_t} \mathbb{E}_{(x,y)\in D}\Big\{\lambda \mathcal{L}_{\mathrm{CE}}(\theta_g, x, y) + \mathcal{L}_{\mathrm{MSE}}(f_g(x), f_t(x')) + \alpha \mathcal{L}_{\mathrm{KL}}(f_g(x) \,\|\, f_t(x')) + \beta \left| \mathcal{L}_{\mathrm{KL}}(f_t(x) \,\|\, f_g(x)) - \mathcal{L}_{\mathrm{KL}}(f_g(x) \,\|\, f_t(x)) \right| \Big\},$$

D²R: Dual regularization loss with adversarial generation



D²R: Dual regularization loss with adversarial generation



(a) Baseline Method (b) D2R (ours) (c) D2R-CAG(ours)

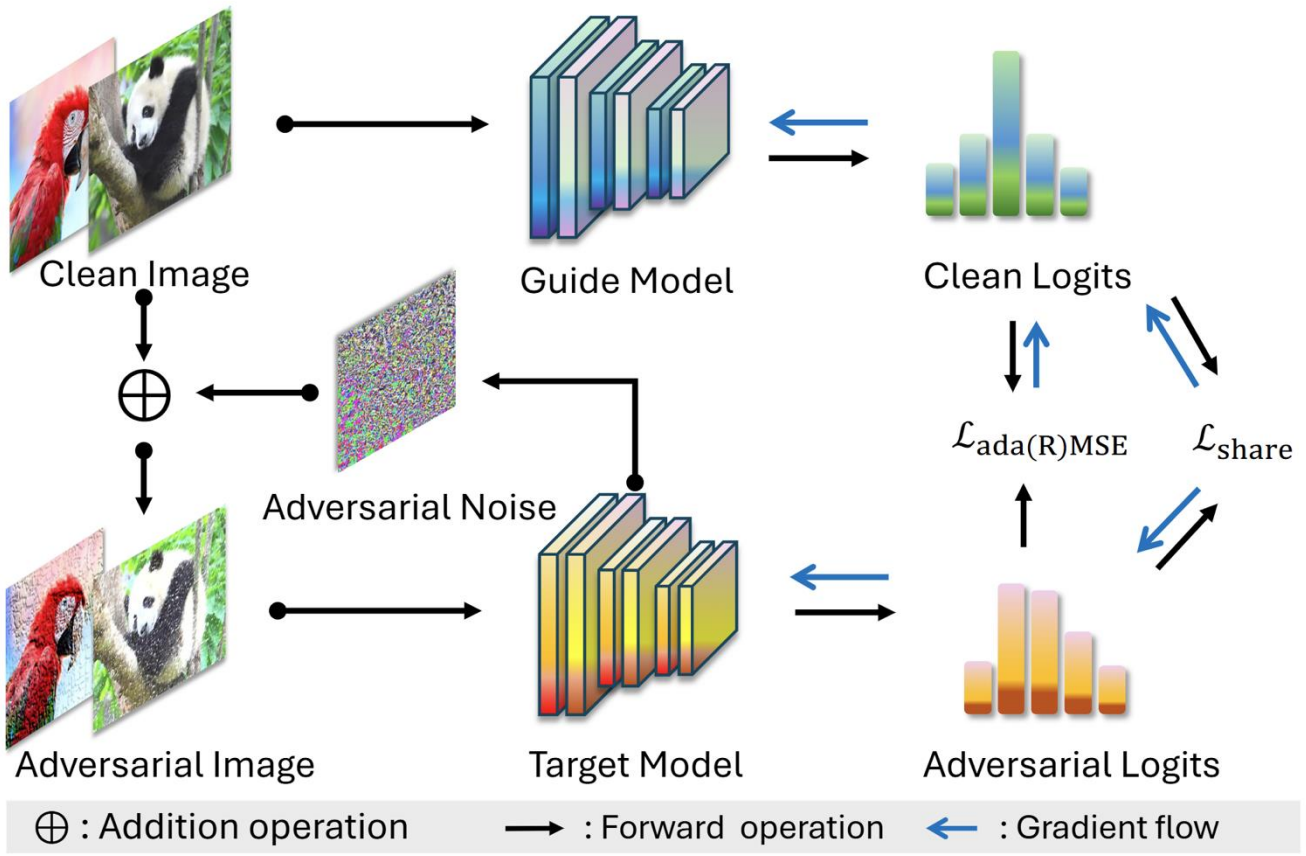
A noticeably flatter loss profile can be observed in our methods, indicating improved robustness against adversarial perturbations

D^2R: Dual regularization loss with adversarial generation

WideResNet-34-10 on CIFAR-10 dataset

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
PGD-AT	85.17	56.07	55.08	54.88	53.91	51.69
TRADES	85.72	56.75	56.1	55.9	53.87	53.40
MART	84.17	58.98	58.56	58.06	54.58	51.10
FAT	87.97	50.31	49.86	48.79	48.65	47.48
GAIRAT	86.30	60.64	59.54	58.74	45.57	40.30
AWP	85.57	58.92	58.13	57.92	56.03	53.90
LBGAT (baseline)	88.22	56.25	54.66	54.30	54.29	52.23
LAS-AT	86.23	57.64	56.49	56.12	55.73	53.58
RAT(TRADES)	85.98	-	58.47	-	56.13	54.20
D2R(ours)	86.00	58.17	56.88	56.60	55.69	54.04
D2R-CAG(ours)	85.68	58.50	57.22	56.73	**56.66**	**54.65**

AdaGAT: Adaptive guidance for adversarial training

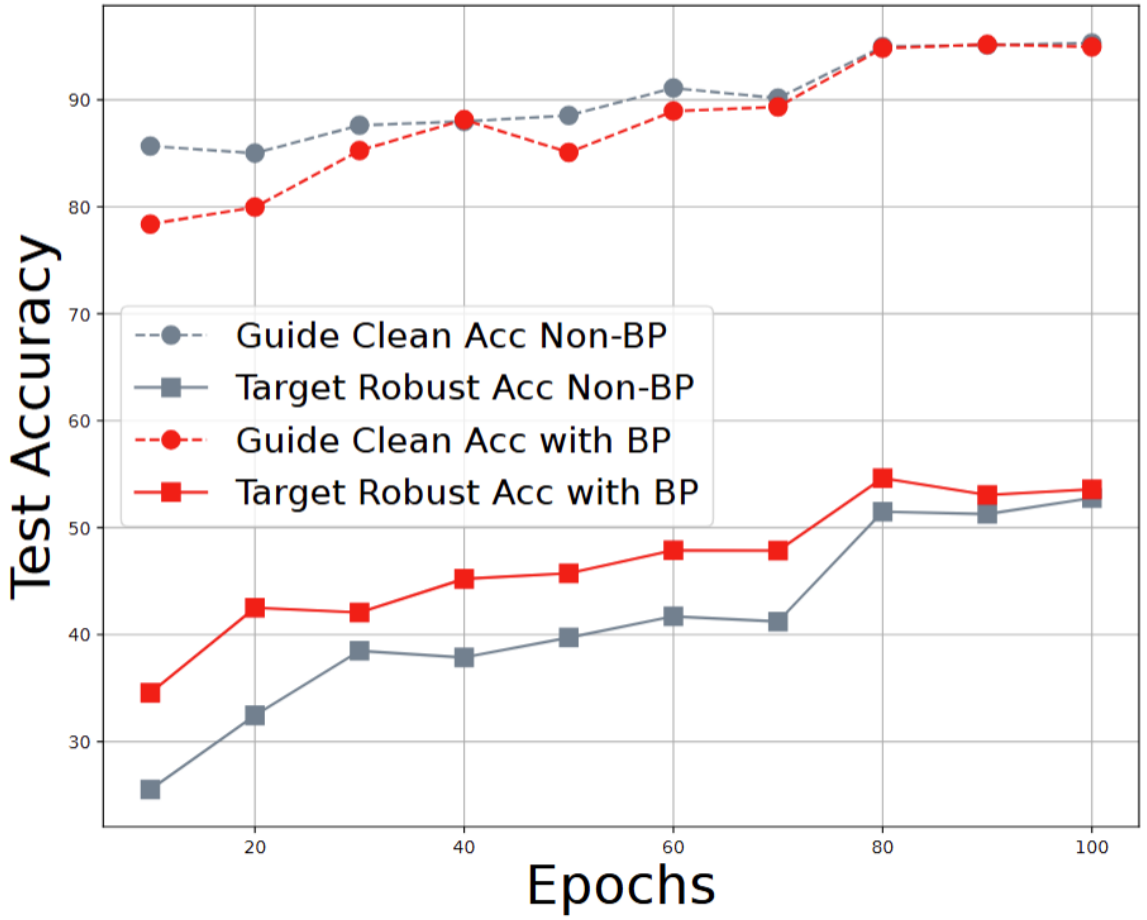


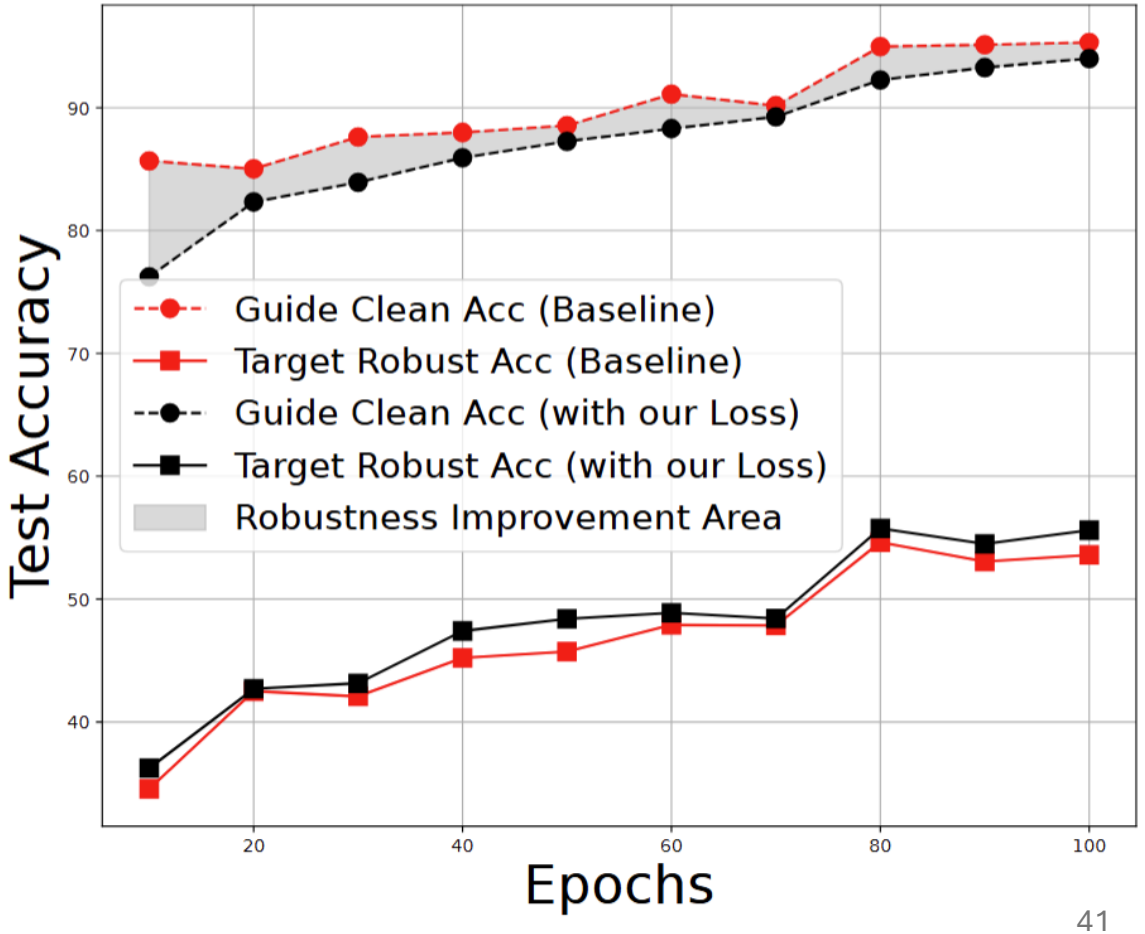
$$\mathcal{L}_{\text{AdaGAT-MSE}} = \min_{\theta_g}\Big\{ \mathcal{L}_{\text{CE}}\left(f_{\theta_g}(x), y\right) + \mathcal{L}_{share} + \lambda\, \mathcal{L}_{\text{adaMSE}}\left(\overline{f_{\theta_t}(x+\delta)}, f_{\theta_g}(x)\right) \Big\}$$

$$\mathcal{L}_{\text{AdaGAT-RMSE}} = \min_{\theta_g}\Big\{ \mathcal{L}_{\text{CE}}\left(f_{\theta_g}(x), y\right) + \mathcal{L}_{share} + \lambda\, \mathcal{L}_{\text{adaRMSE}}\left(\overline{f_{\theta_t}(x+\delta)}, f_{\theta_g}(x)\right) \Big\}$$

Guide Model Target Model

AdaGAT: Comparison of the guiding model's performance with and without backpropagation





Performance

AdaGAT: Adaptive guidance for adversarial training

WideResNet-34-10 on CIFAR-10 dataset

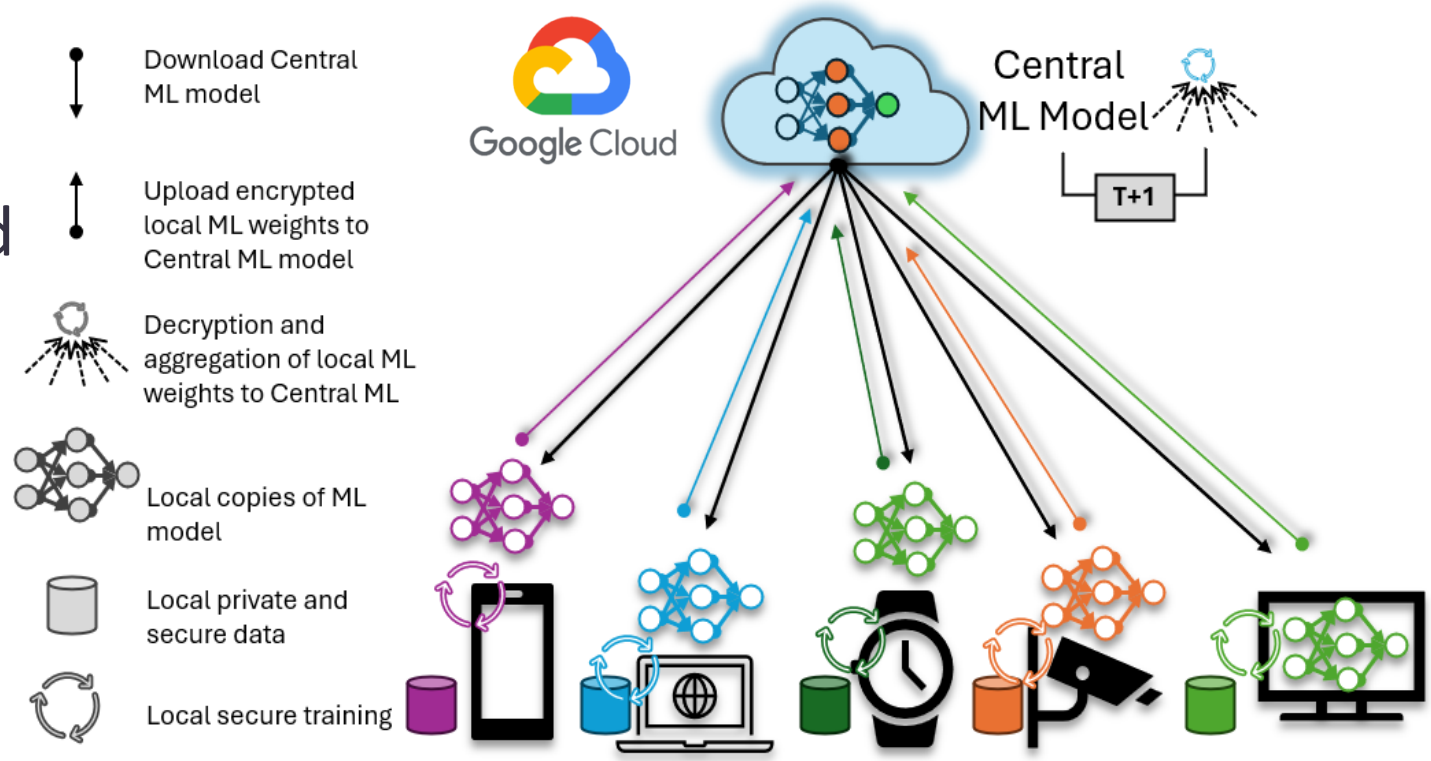
Method	PGD-10	PGD-20	PGD-50	C&W	AA
TRADES	29.20	28.66	28.56	27.05	25.94
SAT	28.10	27.17	26.76	27.32	24.57
LBGAT (baseline)	32.05	30.77	30.42	28.72	27.16
AdaGAT-MSE (ours)	32.50	31.59	31.31	29.24	27.69
AdaGAT-RMSE (ours)	**32.63**	**31.63**	**31.35**	**29.37**	**27.79**

Part 2
Safeguarding AI: Security and Privacy

Data privacy and Security

Model-centric federated learning

Data is generated locally and remains de-centralised. Each client stores its own data and cannot read the data of other clients. Data is not independently or identically distributed (non-IID*)

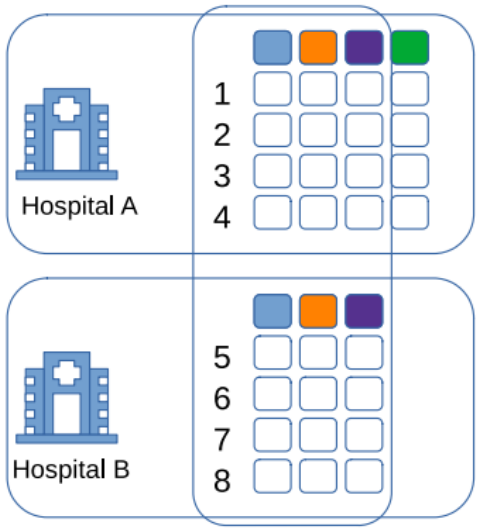


*Non-IID (non-independent and identically distributed) data refers to datasets where samples are not drawn from the same underlying distribution or are not independent of each other. This means that the data exhibits skewness or heterogeneity across different clients or data points

Horizontal federated learning

(Sample-based/Homogenous) federated learning

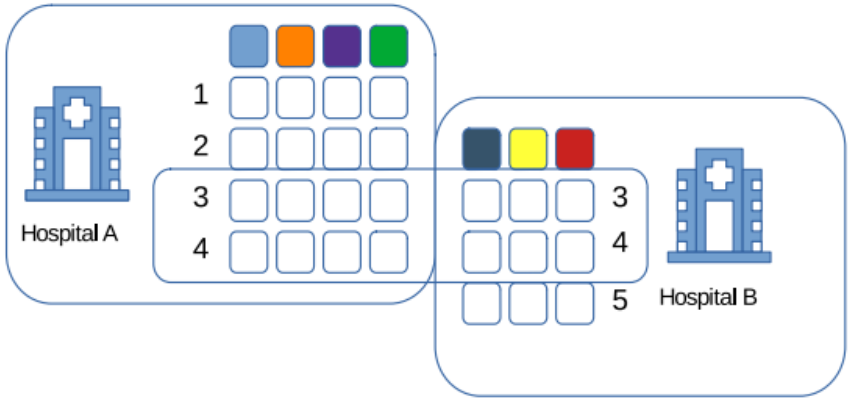
Multiple hospitals can collaboratively train a disease analysis model without sharing customer information. Hospitals A and B have the same feature but samples of different patients



Vertical federated learning

(Feature-based/**Heterogeneous**) federated learning

Two Hospitals/Institutions jointly train a model, with the one providing users' medical image data and other providing medical records. Hospital A has information about Patient A related to heart issues' treatment history, and Hospital B has data about patient A's monthly routine checkup history



Sachin et al . (2024). Federated learning for digital healthcare: concepts, applications, frameworks, and challenges. *Computing*, *106*(9), 3113-3150.

Hierarchical federated learning (HFL)

Wearable devices may transmit data to a hospital's local server, which trains a preliminary model, and then shares it with a central research institution for further refinement

- **Intermediate aggregation**

  Local devices aggregate updates before sending them to a central node.

- **Reduced communication overhead**

  Fewer direct transmissions to ground stations, conserving bandwidth.

- **Scalable**

  Handles large number of clients with minimal latency.

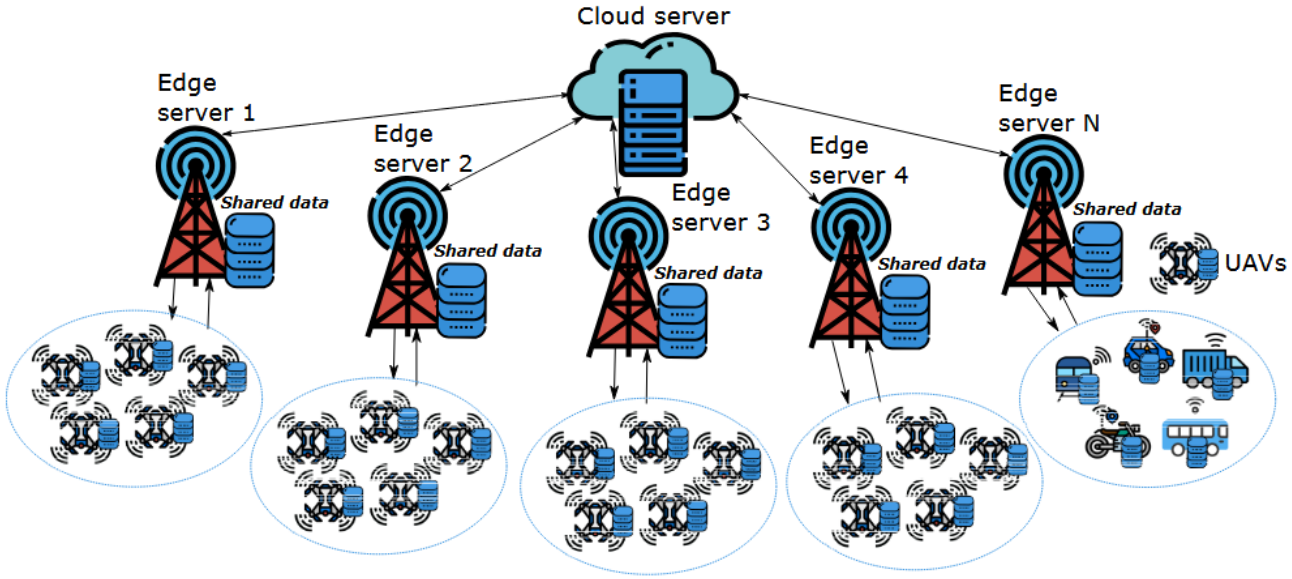
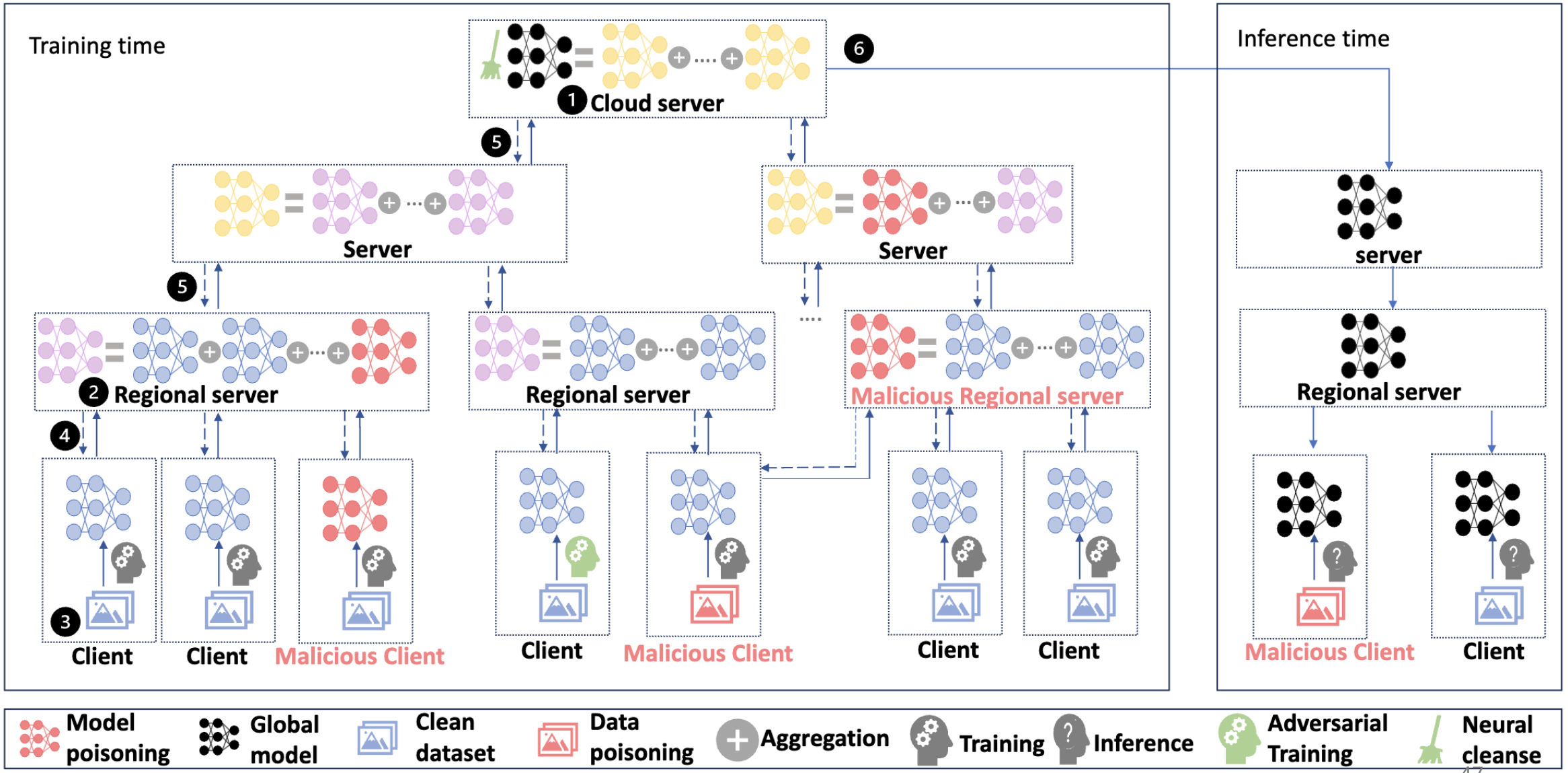


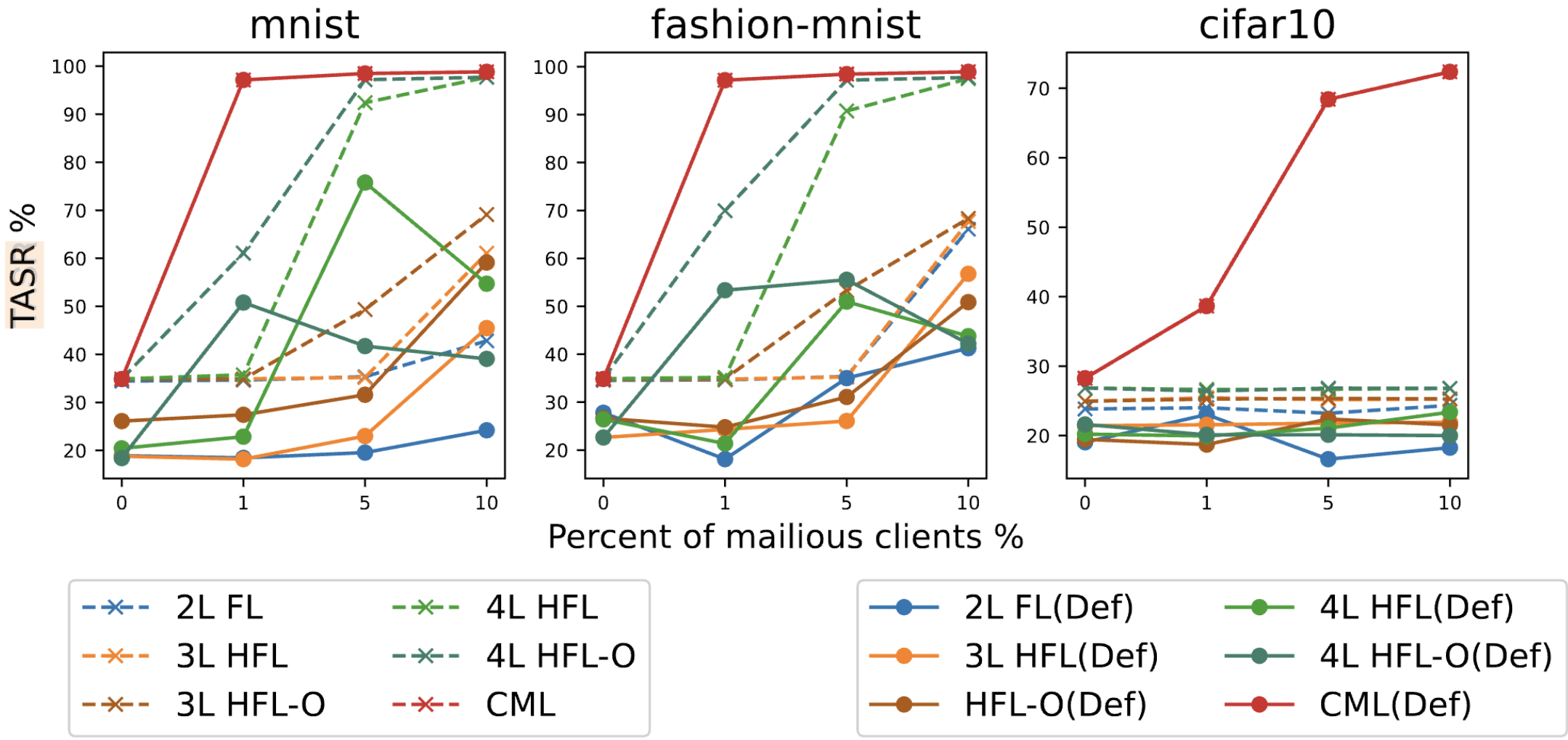
Image source: Tursunboev et al. Hierarchical Federated Learning for Edge-Aided Unmanned Aerial Vehicle Networks. Appl. Sci. 2022

Adversarial attacks on federated learning



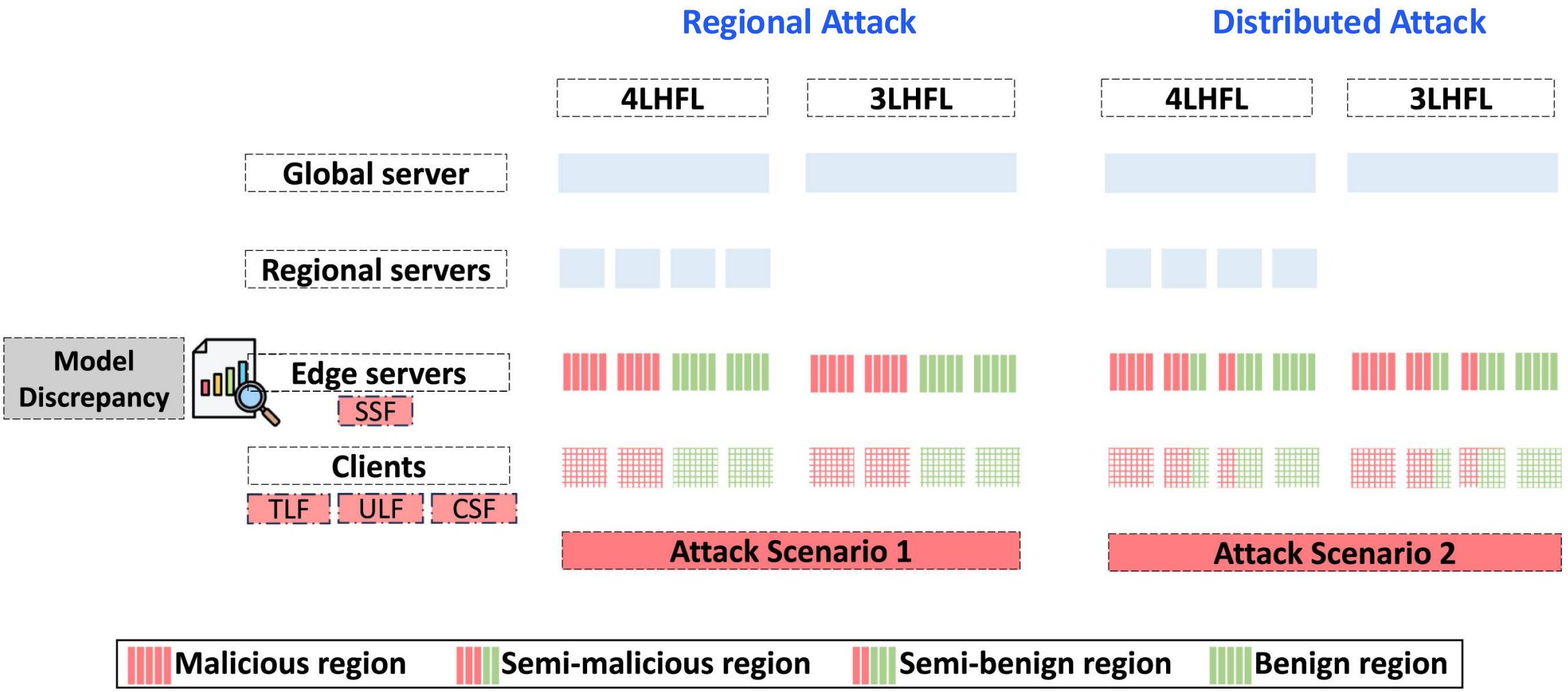
Targeted attack success rate

Targeted Attack/Defense Success Rate of Backdoor Attacks: attack (dashed line) and after defence (solid line)



Targeted Attack Success rate (TASR) Central Machine Learning (CML) Hierarchical Federated Leaning (HFL)

Attack/defense on hierarchical federated learning



Targeted Label Flipping (TLF), Untargeted Label Flipping (ULF), Client-Side Sign Flipping (CSF), and Server-Side Sign Flipping (SSF). For both scenarios, 50% of clients or edge servers were malicious.

Defense on hierarchical federated learning

Model Discrepancy Score (MDS)

$$MDS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\text{Normalized Metric}_i\right)^2}$$

where N represents the number of metrics

Dissimilarity (Cosine similarity).

Dissimilarity quantifies the angular deviation between two model weight vectors

Distance (Euclidean distance).

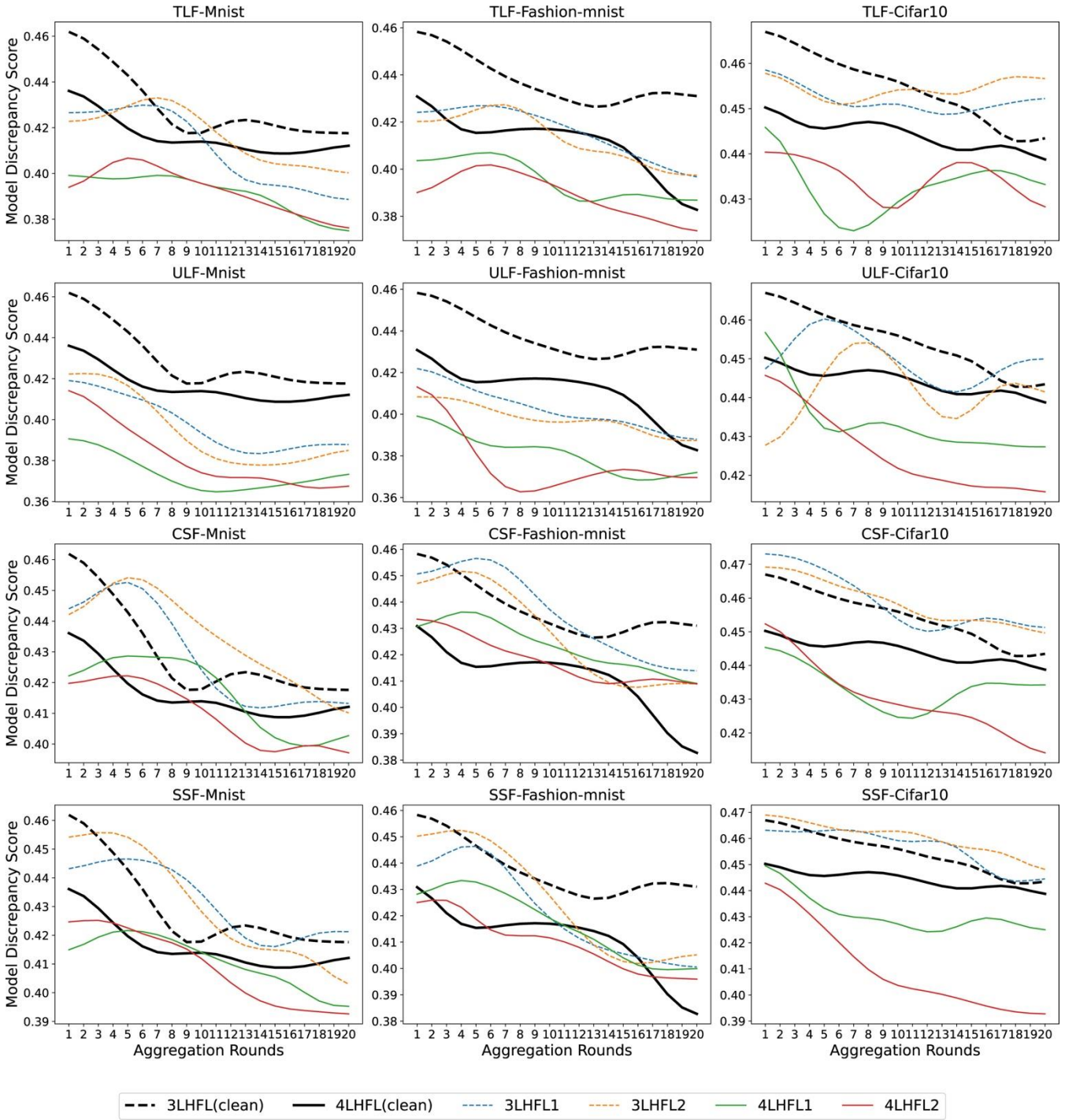
Euclidean Distance measures the magnitude of deviation between two model updates

Uncorrelation (Pearson correlation).

Uncorrelation assesses the linear dependency between updates

Divergence (Jensen–Shannon divergence).

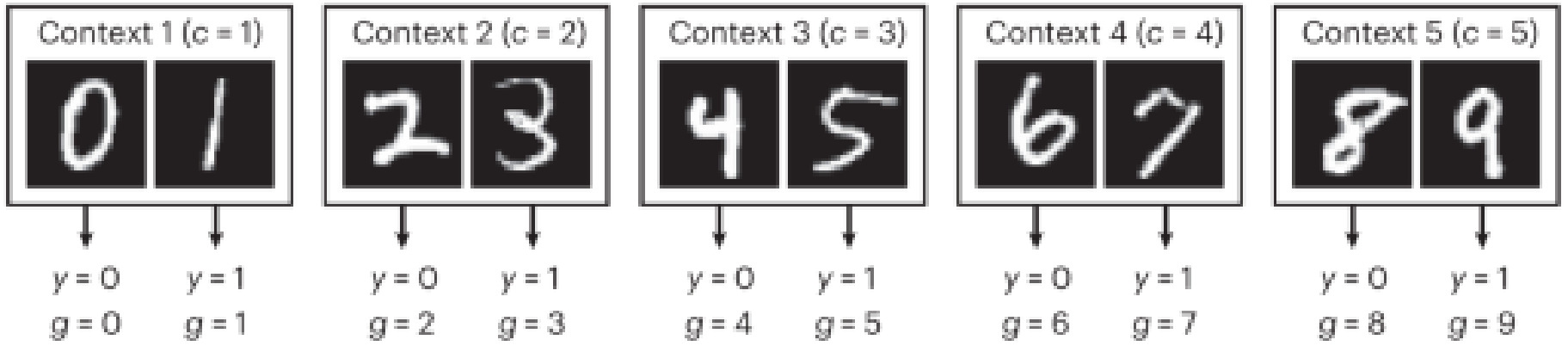
Jensen–Shannon Divergence (JSD) captures probabilistic shifts in weight distributions



Part 3
Safeguarding AI: Continuity of Learning

Type of Incremental learning

a

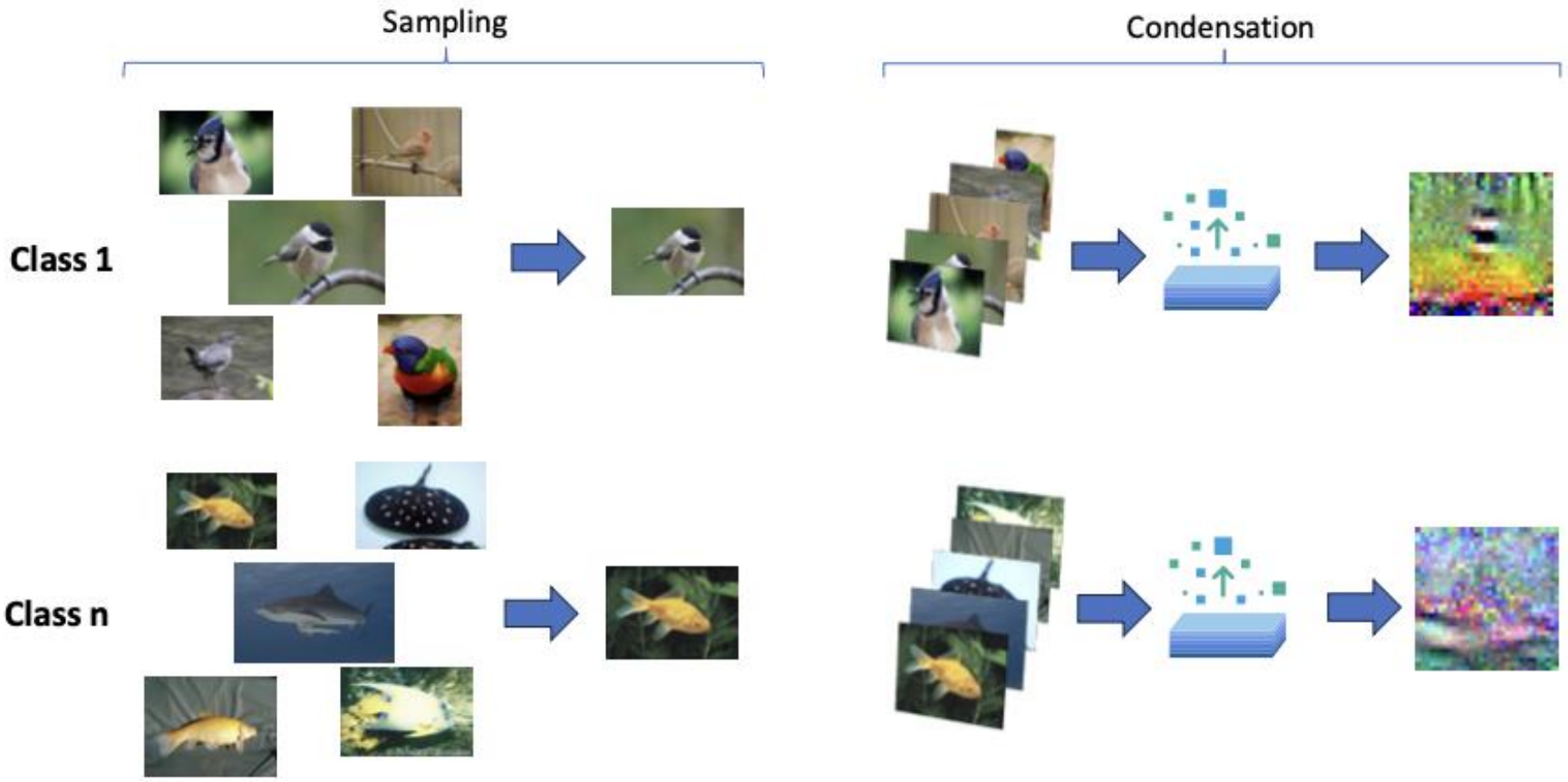


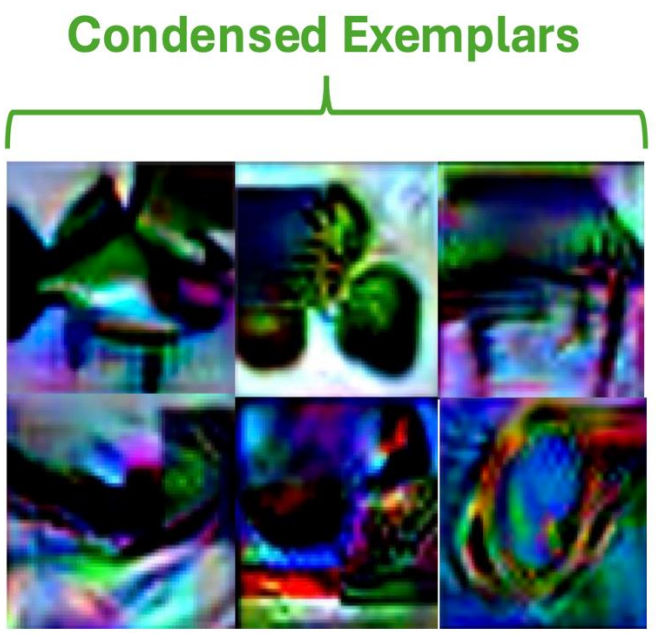
b

	Input (at test time)	Expected output	Intuitive description
Task-incremental learning	Image + context label	Within-context label[a]	Choice between two digits of same context (e.g. 0 or 1)
Domain-incremental learning	Image	Within-context label	Is the digit odd or even?
Class-incremental learning	Image	Global label	Choice between all ten digits

van de Ven, G.M., Tuytelaars, T. & Tolias, A.S. Three types of incremental learning. *Nat Mach Intell* **4**, 1185–1197 (2022). https://doi.org/10.1038/s42256-022-00568-3

Class incremental federated learning

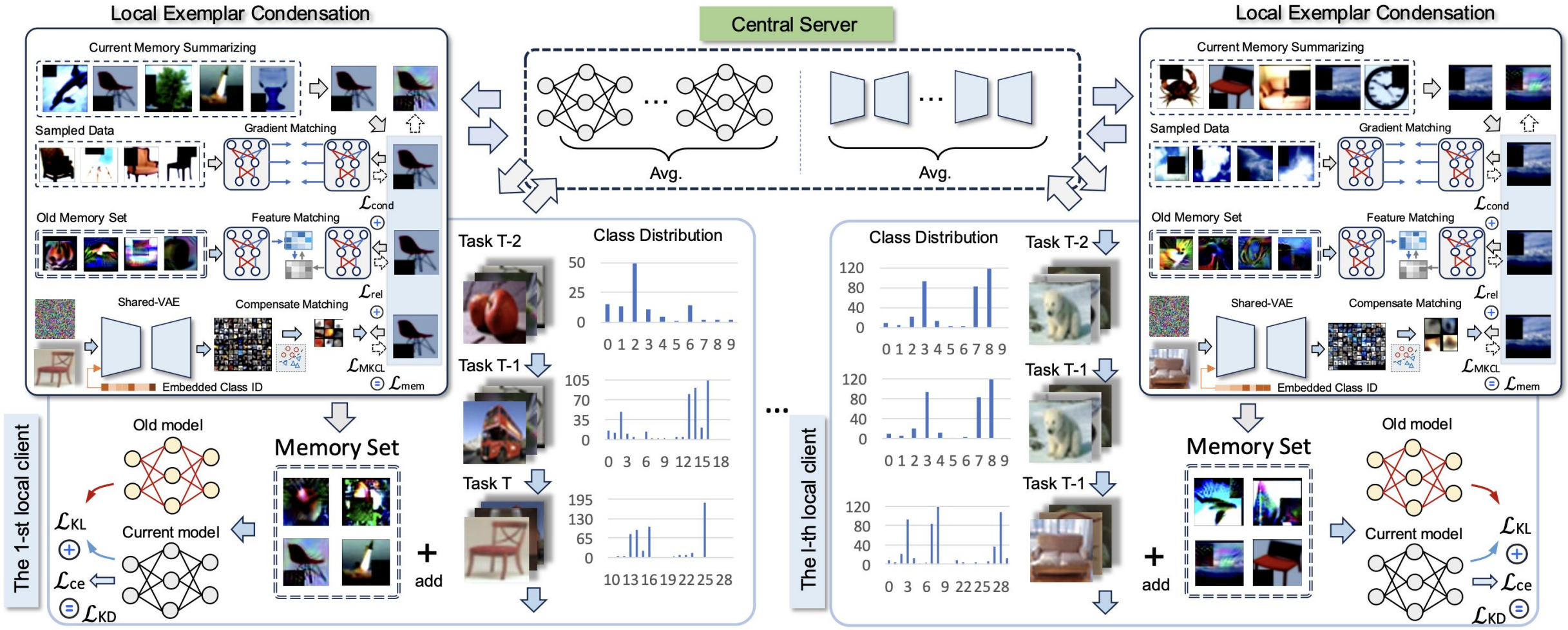




Condensed feature refers to a transformed or derived feature that represents a subset of the original features or a combination of them

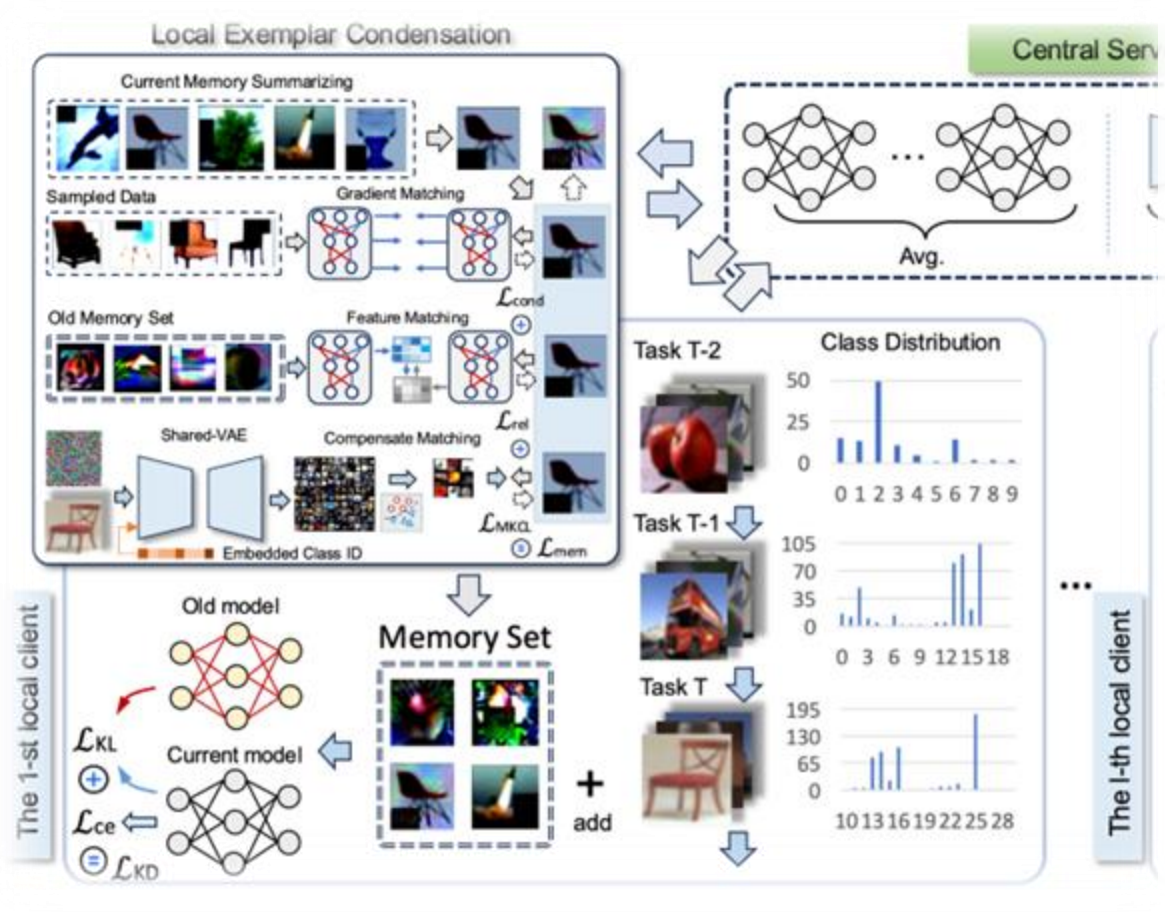
ExReplay eliminates the limitations of exemplar selection in replay-based approaches for mitigating catastrophic forgetting in federated continual learning

Class incremental federated learning



Ex Replay: Clients continuously learn from new class data sequences using a dual-distillation structure to mitigate catastrophic forgetting.

Class incremental federated learning



Ex Replay: Clients continuously learn from new class data sequences using a dual-distillation structure to mitigate catastrophic forgetting.

The exemplar condensation process involves three key components:

a gradient matching loss (L_{cond}) for meta-information distillation,

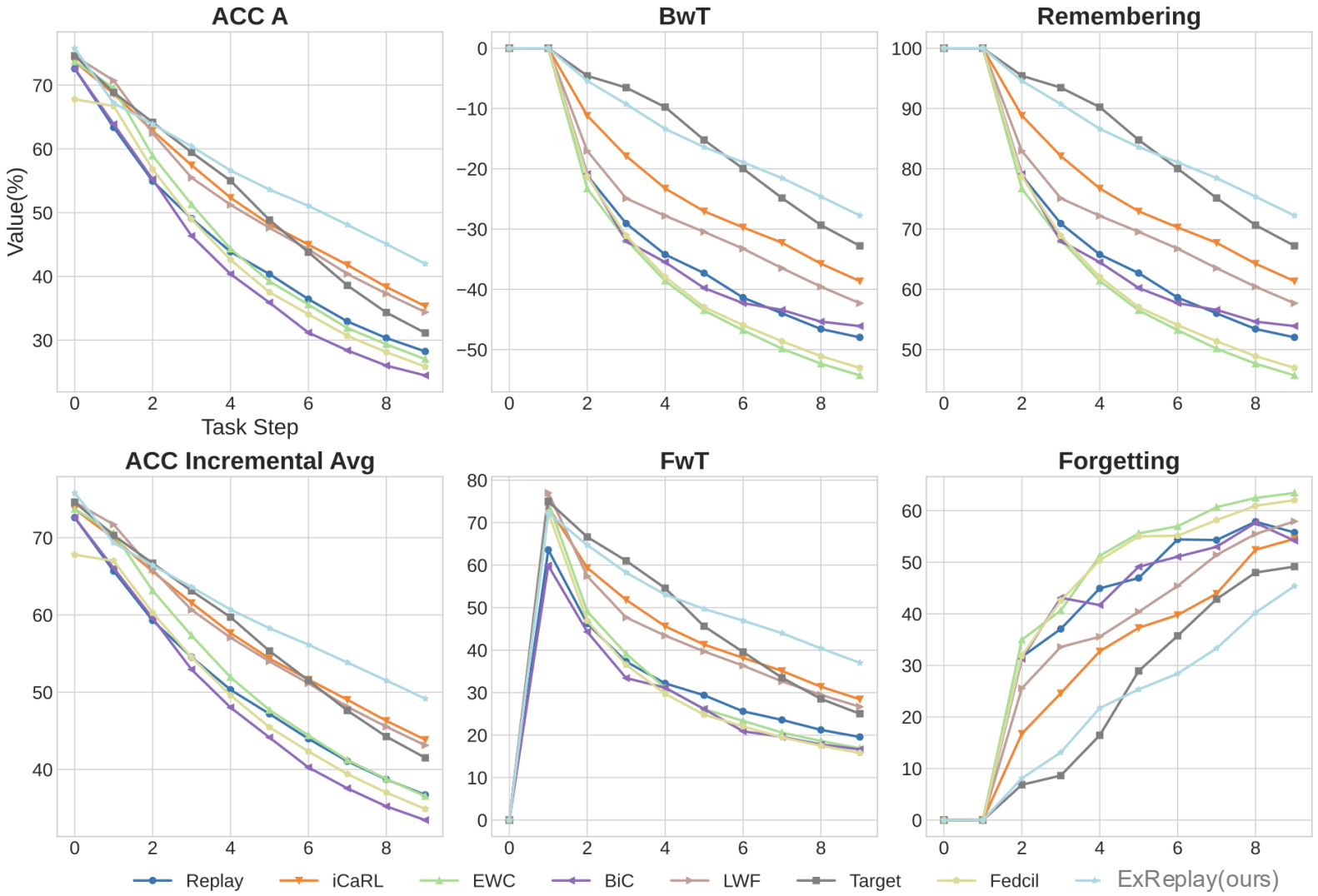
a feature matching loss (L_{rel}) for consistency between condensed samples and real images

a compensation loss (L_{MKCL}) to address meta-information heterogeneity using disentangled features

A knowledge distillation loss (L_{KD}) helps retain prior knowledge.

Class incremental federated learning

Evaluation of multiple metrics (%) on CIFAR100 under a Non-IID setting



Backward Transfer (BwT): measures the influence that learning a new task has on the performance of previously learned tasks.

Forward Transfer (FwT):assesses the influence that learning a new task has on the performance of future tasks.

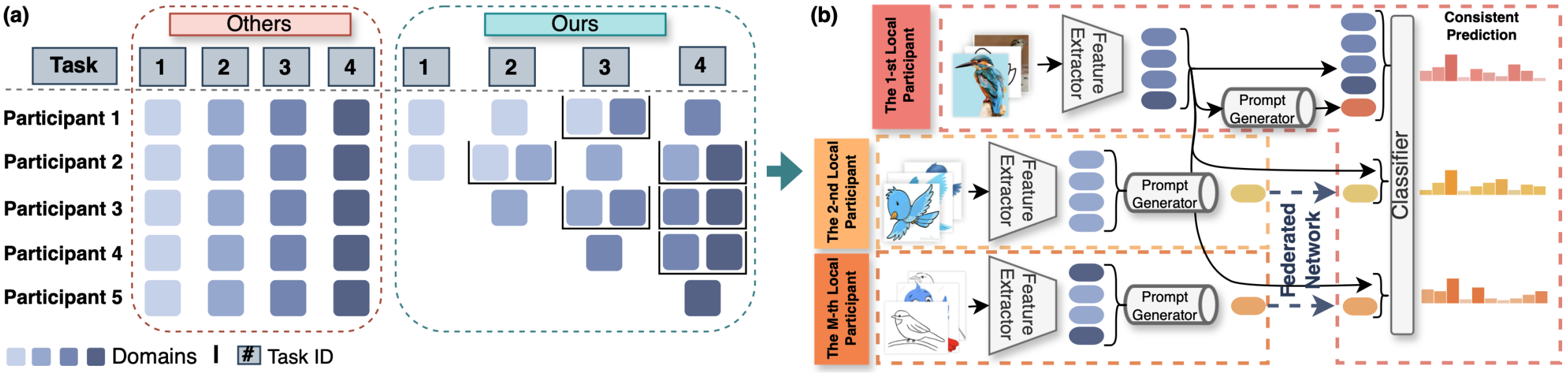
Remembering: calculates the degree of retention for previous tasks as part of the backward transfer process.

Forgetting: measures the average amount of forgetting across all tasks, helping to quantify how much information is lost as new tasks are learned

Domain incremental federated learning

RefFiL: Rehearsal free federated domain-incremental learning framework

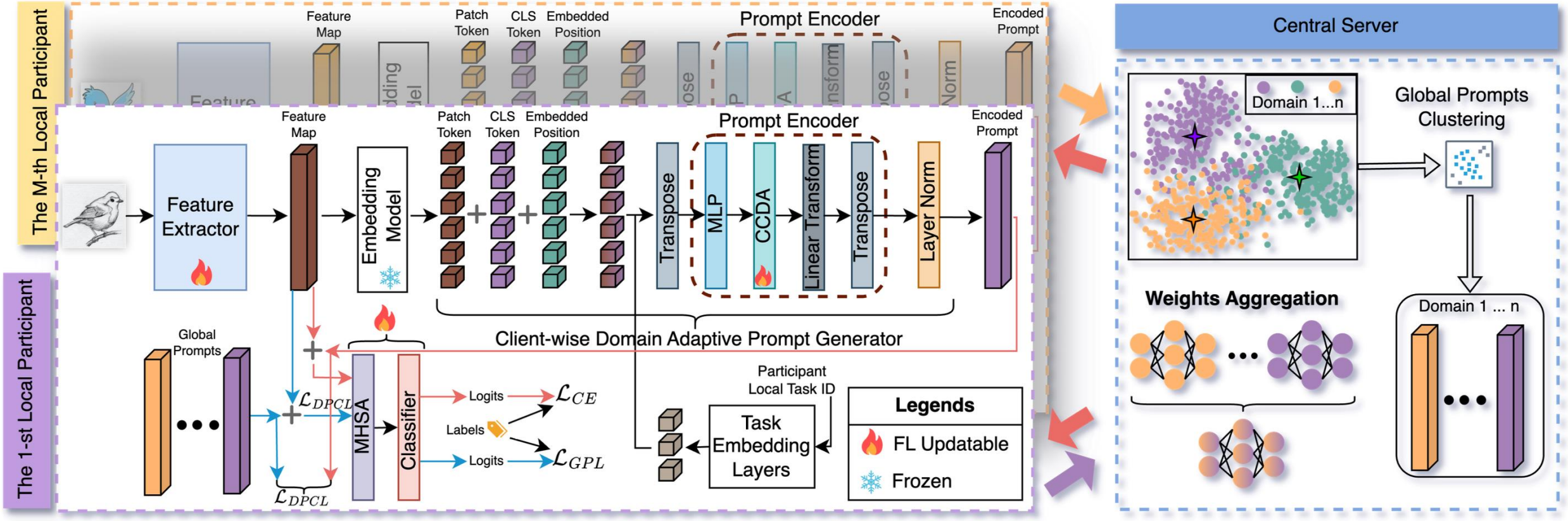
unseen domains are continually learned domain-incremental learning



Key steps: the 1st participant processes new domain data using global prompts from the 2nd to m-th participants and local prompts, enhancing robustness by aligning the model's predictions across diverse domain prompts as inputs.

Domain incremental federated learning

RefFiL: Rehearsal free federated domain-incremental learning framework



Each participant first encodes local prompts using the tokenized feature map and task ID embedding. These local prompts are then concatenated with the feature map to compute the loss L_{CE} . Simultaneously, the feature map is combined with global prompts to calculate the loss L_{GPL}, and the loss L_{DPCL} is determined between global and local prompts. Subsequently, all local prompts, along with the updated local models, are transmitted to the central server.

Domain incremental federated learning

Comparison of RefFiL's performance with five baseline methods on four widely used datasets, showcasing average accuracy (Avg %) and accuracy for each domain task (%)

	Task 1 → 5 on Digit-Five							Task 1 → 4 on OfficeCaltech10				
Methods	MNIST	MNIST-M	USPS	SVHN	SYN	–	Avg	Amazon	Caltech	Webcam	DSLR	Avg
Finetune	99.68	97.75	63.87	75.84	49.80	–	77.39	76.56	57.79	24.58	19.29	44.56
FedLwF	99.68	92.80	69.16	69.39	56.86	–	77.58	76.56	53.24	28.57	28.74	46.78
FedEWC	99.68	97.48	74.63	73.32	45.89	–	78.20	76.56	56.59	29.83	15.55	44.38
FedL2P	99.66	98.06	80.01	81.89	57.65	–	83.45	76.56	51.80	31.09	26.57	46.51
FedL2P†	99.64	97.65	85.18	81.65	60.17	–	84.86	71.35	55.88	29.20	25.20	45.41
FedDualPrompt	99.67	97.96	86.88	81.95	59.30	–	85.15	74.48	50.36	31.93	23.82	45.15
FedDualPrompt†	99.65	97.90	84.68	81.40	58.34	–	84.39	75.90	53.96	33.82	27.76	47.86
RefFiL	**99.68**	**98.25**	**90.96**	**83.70**	**62.11**	–	**86.94**	**78.65**	**61.15**	**40.76**	**33.66**	**53.56**

	Task 1 → 6 on FedDomainNet							Task 1 → 4 on PACS				
Methods	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg	Photo	Cartoon	Sketch	Art Painting	Avg
Finetune	**51.48**	15.89	28.05	**27.84**	29.45	18.07	28.46	61.68	47.45	36.12	30.82	40.18
FedLwF	**51.48**	18.10	26.71	25.98	27.47	17.96	27.95	61.68	47.07	25.11	26.61	40.12
FedEWC	50.76	15.46	22.66	21.87	27.45	18.37	26.10	63.17	47.70	23.66	27.36	40.27
FedL2P	40.55	13.19	21.09	28.15	30.13	18.42	25.26	64.97	48.32	**50.09**	35.32	49.68
FedL2P†	37.63	9.29	16.79	27.09	26.68	15.59	22.18	65.57	54.67	45.25	34.52	50.00
FedDualPrompt	51.17	19.48	28.74	22.68	29.40	18.05	28.25	73.65	56.54	44.93	41.07	54.05
FedDualPrompt†	51.14	20.20	28.91	23.09	30.07	17.76	28.53	**75.75**	54.55	43.23	37.62	52.79
RefFiL	51.27	**20.91**	**29.23**	22.57	**30.62**	**18.98**	**28.93**	73.95	**59.90**	43.17	**44.27**	**55.32**

Varun.Ojha@ncl.ac.uk | edgeaihub.co.uk | ojhavk.github.io

References

- Fragility, Robustness and Antifragility in Deep Learning
  Artificial Intelligence, Elsevier. (2024)
  Pravin C, Martino I, Nicosia G, Ojha V
- Security Assessment of Hierarchical Federated Deep Learning
  33rd International Conference on Artificial Neural Networks (ICANN). (2024)
  Alqattan D, Sun R, Liang H, Nicosia G, Snasel V, Ranjan R, and Ojha V
- Adversarial robustness in deep learning: Attacks on fragile neurons
  30th Int. Conf. on Artificial Neural Net., ICANN (pp 16-28), Springer (2021)
  Pravin C, Martino I, Nicosia G, Ojha V
- Rehearsal-free federated domain-incremental learning
  45th IEEE International Conference on Distributed Computing Systems (IEEE ICDCS 2025)
  R Sun, H Duan, J Dong, V Ojha, T Shah, R Ranjan
- D2R: dual regularization loss with collaborative adversarial generation for model robustness
  34th International Conference on Artificial Neural Networks (ICANN 2025)
  Z Liu, H Liang, R Ranjan, Z Zhu, V Snasel, V Ojha
- Analysis of deep learning under adversarial attacks in Hierarchical Federated Learning
  High-Confidence Computing, Elsevier. (2025)
  Alqattan DS, Snasel V, Ranjan R, Ojha V
- Dynamic Label Adversarial Training for Deep Learning Robustness Against Adversarial Attacks
  31st International Conference on Neural Information Processing (ICONIP). (2024)
  Liu Z, Duan H, Liang H, Long Y, Snasel V, Nicosia G, Ranjan R and Ojha V