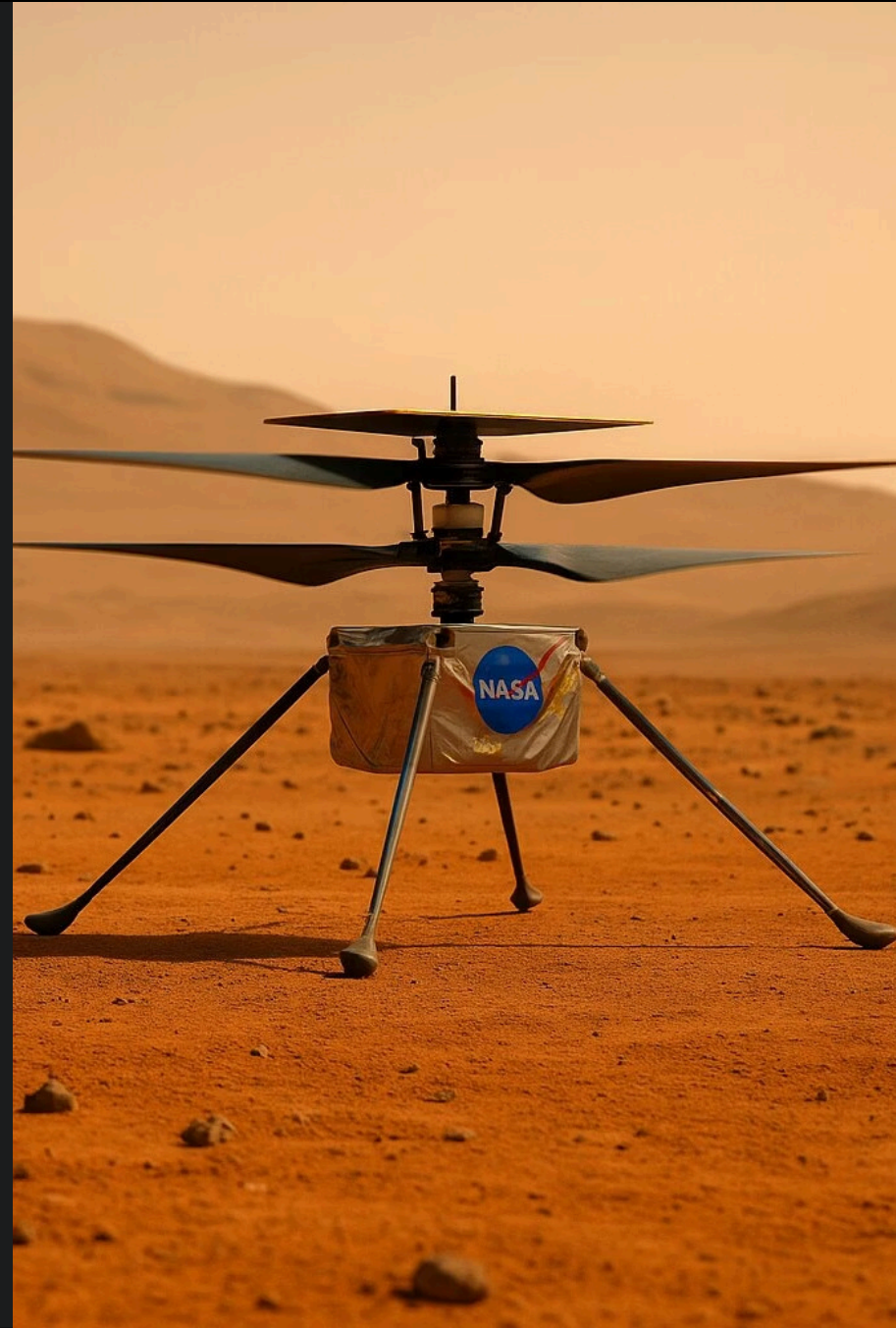
Memory-Robust Few-Shot **Test-Time Adaptation** *for* Small Vision Models

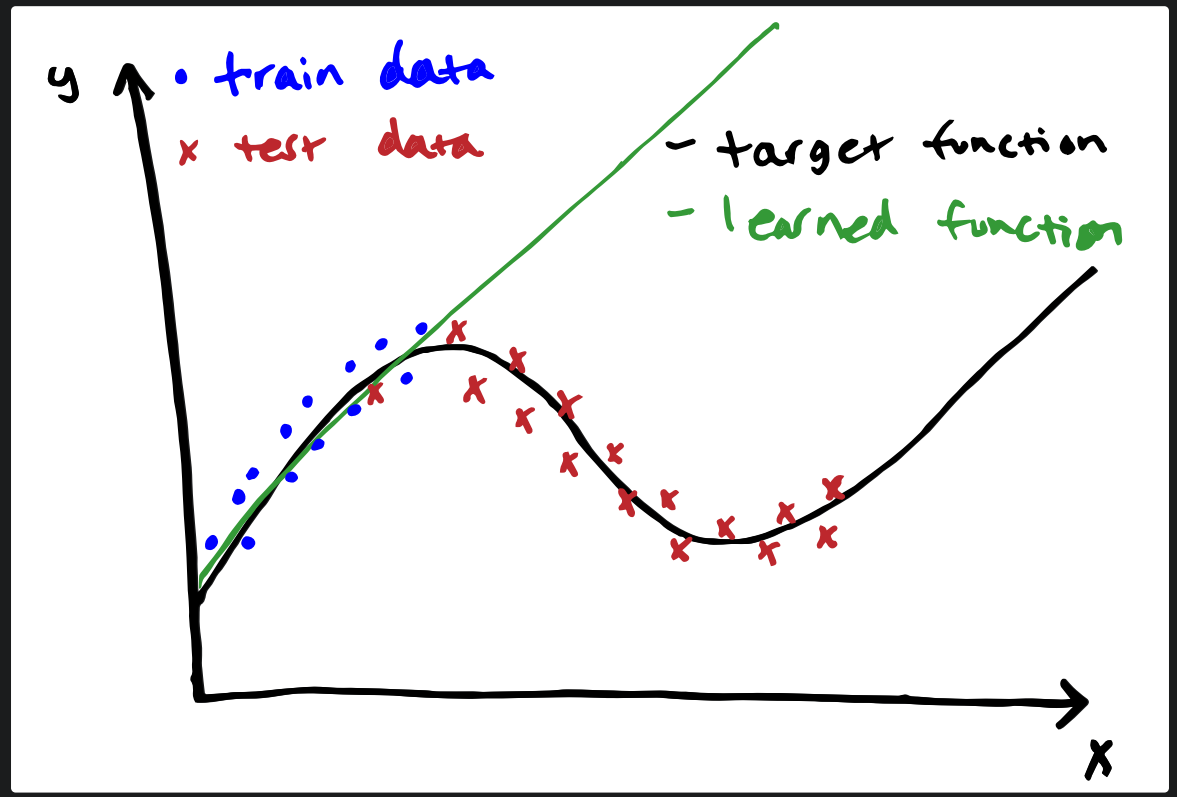
Image:

Ingenuity, NASA's pioneering Mars helicopter, stands on the Martian surface with its twin counter-rotating rotor blades poised above the dusty red terrain—an engineering milestone demonstrating the first powered, controlled flight on another planet.

Image source: flyingglass.com.au



Distribution Shift



When …

- **p (x) changes**
  - Input's probability distribution changes
  - *What the world looks like*
- **p (y | x) does not**
  - Input-label relationship does not
  - *What the world means*

Image:

Co-variate distribution shift—the most common kind of distribution shift—can be solved by Test-time Adaptation.

Image Source: ***dcai.csail.mit.edu***

Test-Time Adaptation: Promises and Problems

A Solution

Allows models to 'fix themselves' *during deployment* instead of waiting for costly retraining.

Thus making systems more resilient to real-world changes such as: new lighting; sensor noise; new weather.

Traditional 'Offline Problems'

Some Has Changed

'Yesterday's World' had worse connectivity & hardware —> fully offline adaptation using unlabeled test data only.

'Today's World' has better of that —> some labeled samples feasible, thus the door to few-shot, better-anchored adaptation opens.

Some Problems

Modern TTA struggles with:

- ☐ Unstable tiny-batch updates
- ☐ Noisy or malicious inputs
- ☐ Predicting drift w.o. ground truth
- ☐ working w.o. large or clean data

Test-Time Adaptation: Promises and Problems

A Solution

Allows models to 'fix themselves' *during deployment* instead of waiting for costly retraining.

Thus making systems more resilient to real-world changes such as: new lighting; sensor noise; new weather.

Traditional 'Offline Problems'

Some Has Changed

'Yesterday's World' had worse connectivity & hardware —> fully offline adaptation using unlabeled test data only.

'Today's World' has better of that —> some labeled samples feasible, thus the door to few-shot, better-anchored adaptation opens.

Some Problems

Modern TTA struggles with:

- [x] ~~Unstable tiny-batch updates~~
- [x] ~~Noisy or malicious inputs~~
- [x] ~~Predicting drift w.o. ground truth~~
- [x] ~~working w.o. large or clean data~~

Building on State-of-the-Art

1

MemBN (2024)

By using a running statistics memory in batch normalization layers, stabilizes adaptation on tiny batches with varying sizes.

2

MedBN (2024)

Replaces mean with median estimator in batch normalization to resist malicious test samples that could skew adaptation.

3

FS-TTA (2024)

Two-stage approach leveraging few labeled support examples from target domain to guide self-training and reduce blind drift.

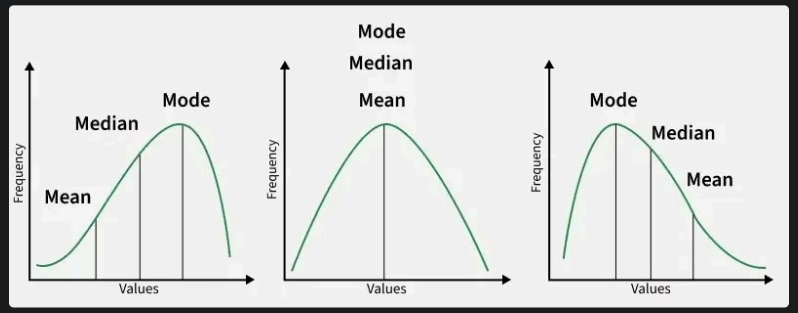


Image:

An illustration of the representativeness of the mean, median, and mode.

Image source: geeksforgeeks.orgl

The Research Gap *(Silos)*

Recent literature solves failure modes in isolation:

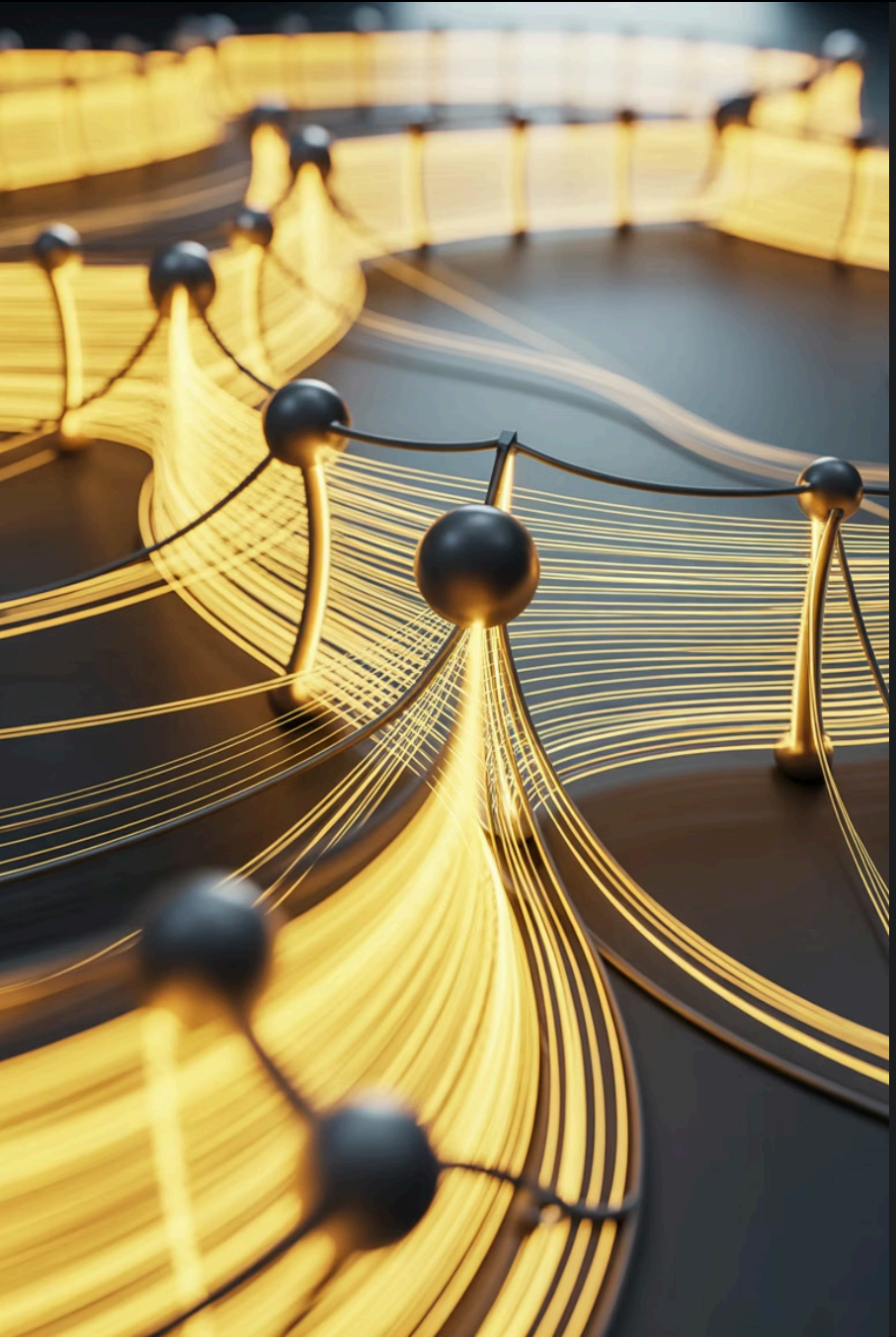
1. **MemBN** **stabilizes tiny batches** but is **outlier-prone**
2. **MedBN** is **outlier-resistant** but is **tiny-batch-prone**
3. **FS-TTA** **anchors adaptation** but **can't handle sequential drift or tiny batches**

When: **These methods aren't Incompatible, They're Complementary**

Our **unified framework** is designed to simultaneously achieve:

1. Stability under tiny batches
2. Tolerance to outliers
3. minimal supervision for large shifts
4. Drift detection without error accumulation

Image: A silo. | ***Image Source:*** ***gettyimages.com***



Q-MemBN+

Our Unified Framework

Q-MemBN+?

Robust Statistics

Quantile-based **Median and IQR** replace mean and variance in batch normalization.

—> outlier resistance

—> accuracy maintenence

Statistics Memory Queue

FIFO queue at each Quantile BatchNorm (Q-BN) layer stores recent batch medians and IQRs.

—> stable tiny batch normalization

ResNet-18

Strong but small model that is easy to train and stabilize—a vision paper benchmark.

—> efficiency

—> comparable results

+?

Prototype Memory Bank

Class-wise feature prototype EMA updated with high-confidence examples.

—> prevent drift

Drift Detection

Detects accumulating changes in the model's BN statistics and confidence.

—> identify non-stationary stream

Drift Resetting

Reset BN statistics to when stable when confidence harm outweighs adaptation loss harm from a reset.

—> correct drift

Our Objectives

01

Accuracy Under Shift

Improve our adapting model's accuracy when the environment changes—by 5% over our original, non-adapting model.

02

Close the Oracle Gap

Reduce the difference between our non-adapting model and a version that was fully trained on the new environment by 50% using our adapting model.

03

Robustness to Anomalies

Limit the accuracy drop to 5% even if 20% of the incoming images are noisy, strange, or intentionally confusing.

04

Efficiency

Update quickly—under 30 ms per image—and use a small model—under 50 M parameters—so that it can easily run on small vision devices.

Experimental Setup

Training Dataset

CIFAR-10: *The* benchmark training dataset for test-time adaptation for small vision models.

Testing Dataset

CIFAR-10-C: CIFAR-10 corrupted through 19 methods at 5 severity levels each. One of the benchmark testing datasets.

Model Architecture

Backbone: ResNet-18 with custom Q-MemBN layers

Parameters: 11.17 M

Optimization: SGD with LR $\leq 10^{-3}$ for adaptation

Evaluation Pipeline

Four branches testing static, oracle, clean adaptation, and naive adaptation scenarios, plus poisoned stream tests.

Experimental Setup Cont.

CIFAR-10

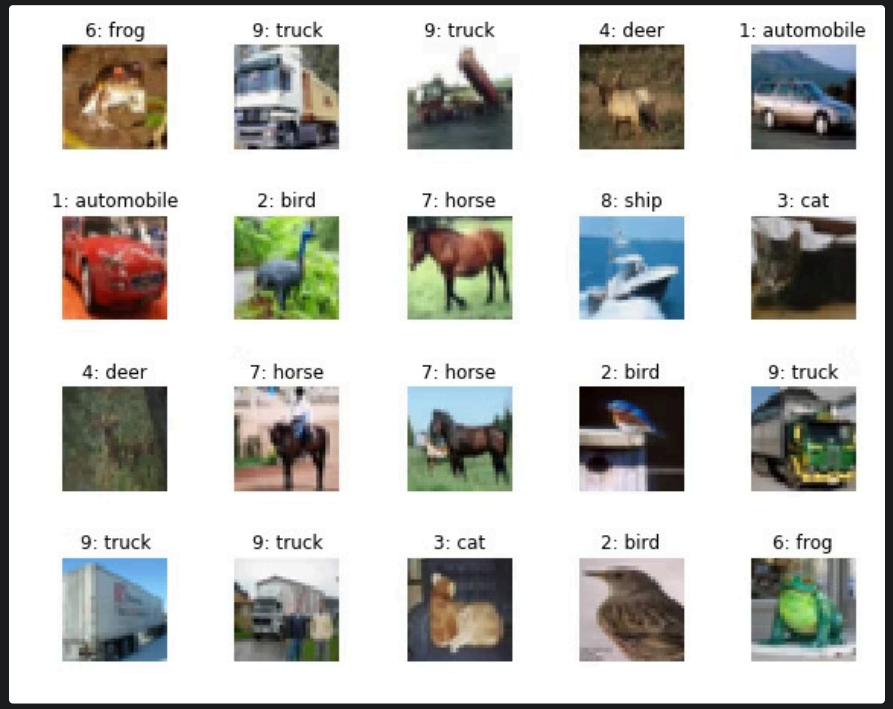


Image Source: corochann.com

CIFAR-10-C

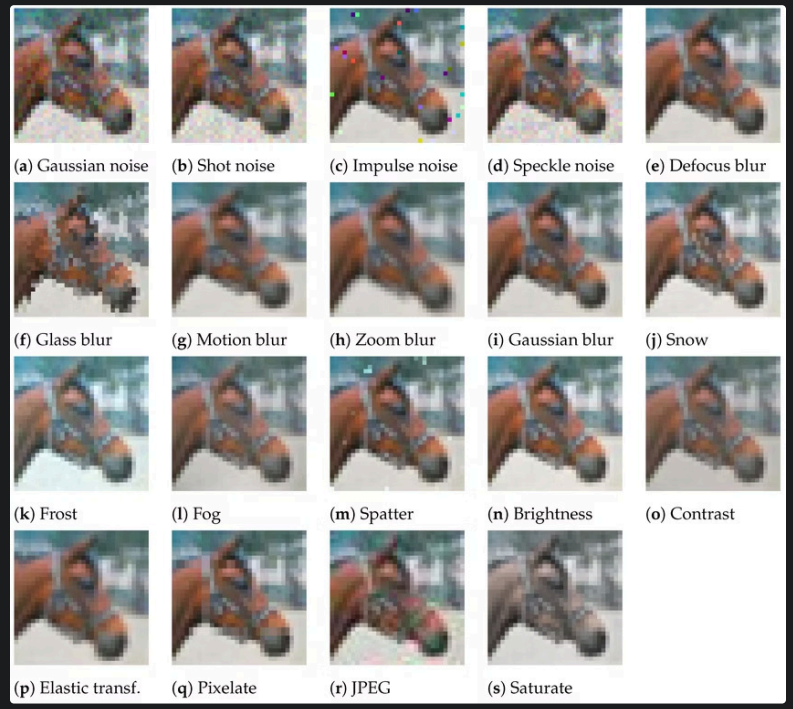
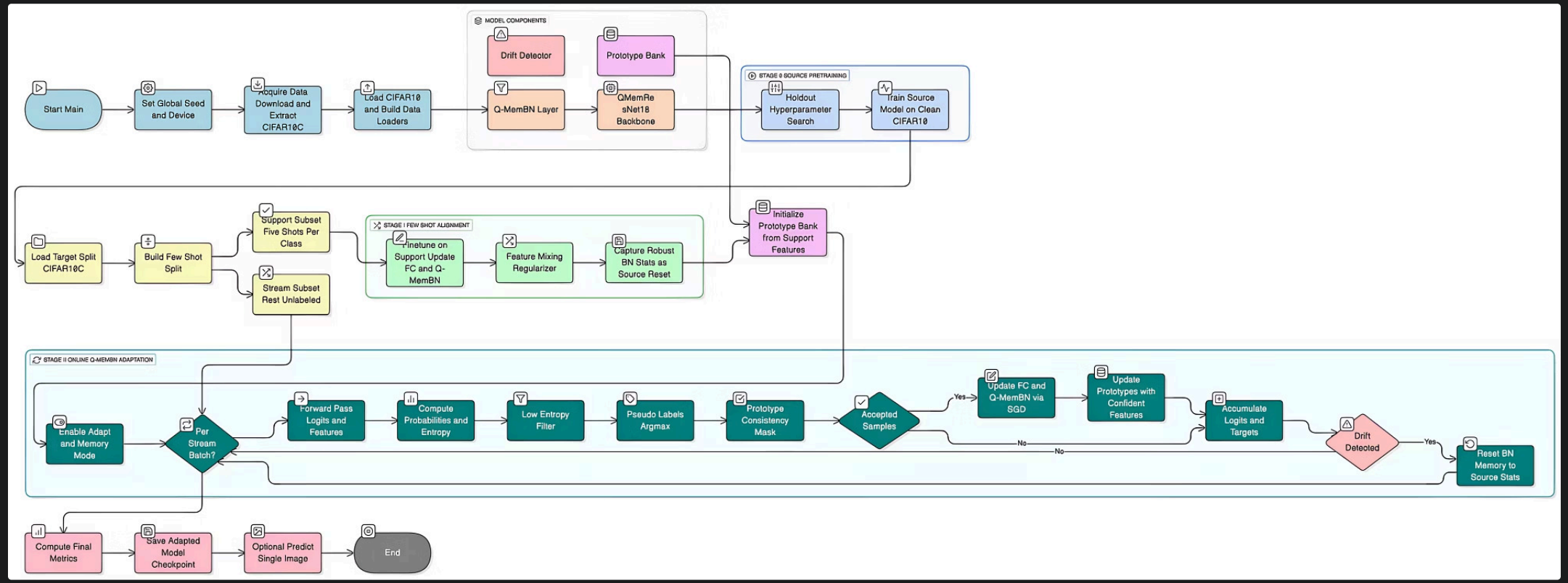


Image Source: www.researchgate.net

Model Pipeline

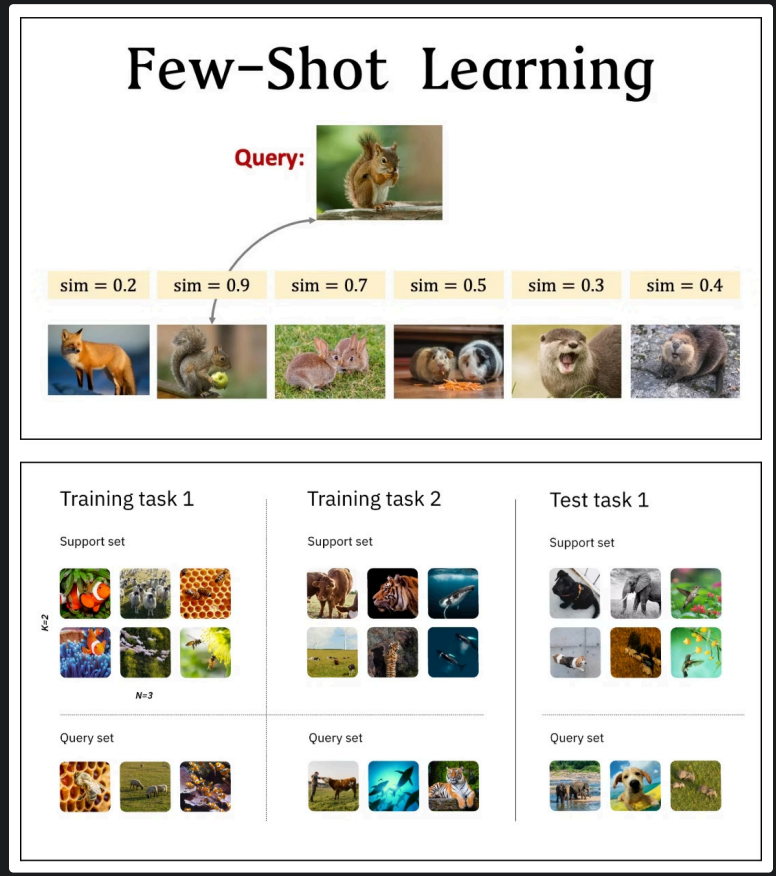


Stage I: Support Fine-Tuning

The Initialization Process

1. **Integrate the Q-BN layer with the median and IQR** instead of the mean and variance.
2. **Initialize the Q-BN statistics from the support set** to prevent improperly-scaled normalization.
3. Apply some Feature Diversity Augmentation to avoid overfitting.

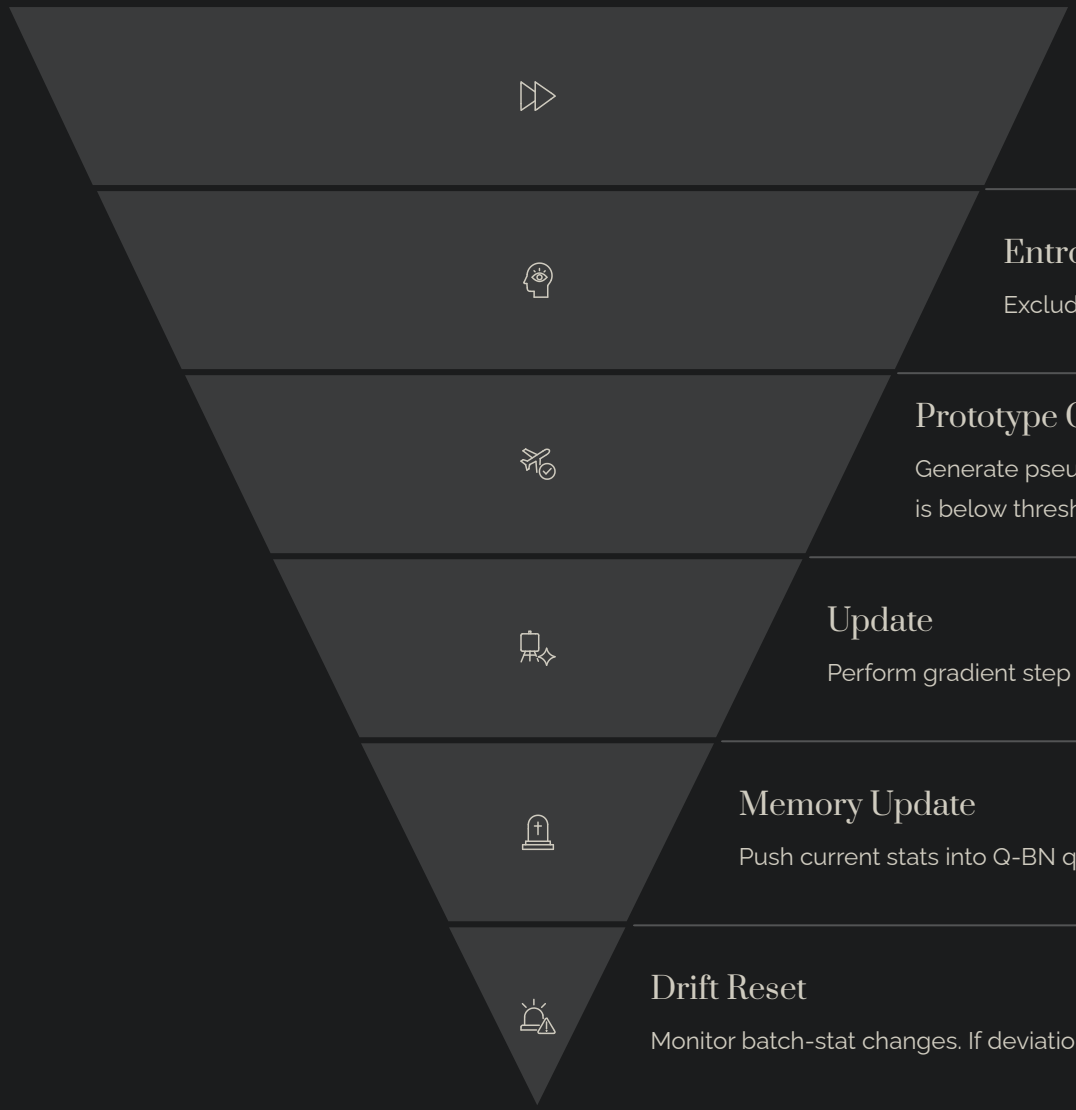
Train—5 epochs, supervised loss—on the support set—5-shot—with AdamW—LR ≈ 10^{-4}.



Images: *Illustrations of the process (up) and implementation (down) of few-shot learning.*

Image Sources: ***Shusen Wang*** *(up) &* ***ibm.com*** *(down)*

Stage II: Online Adaptation



Forward Pass

Process batch through Q-BN layers using memory queue stats. Obtain predictions.

Entropy Filter

Exclude high-entropy samples (above threshold τ) from adaptation while still yielding outputs.

Prototype Check

Generate pseudo-labels, compare features to nearest class prototype. Accept if class matches and distance is below threshold.

Update

Perform gradient step on BN parameters for confident samples. Update prototypes via feature averaging.

Memory Update

Push current stats into Q-BN queue (FIFO). Update class prototypes using exponential moving average.

Drift Reset

Monitor batch-stat changes. If deviation exceeds threshold and confidence is low, flag shift and reset BN memory.

Results Overview *(acc. ppt)*

77.0%

Our Source Model

The branching point.
Trained on **CIFAR-10.**

50 epochs

65.9%

Our Static Model *(Not Adapted)*

A pipeline branch. No adaptation on **CIFAR-10-C.**

66.8%

Q-MemBN+ *(Clean Adapted)*

Our full pipeline. Clean adaptation on **CIFAR-10-C.**

66.6%

Q-MemBN+ *(Naive Adapted)*

A pipeline branch. Naive adaptation on **CIFAR-10-C.**

No *prototype bank, drift detection & resetting, etc.*

70.0%

Our Oracle *(Upper Bound)*

A pipeline branch. Target trained on **CIFAR-10-C.**

50 epochs on CIFAR-10
5 epochs on CIFAR-10-C

Complete Results

*Image Source: **Self.***

```
=== Training source model on CIFAR-10 ===
Best hyperparameters for source training: {'lr': 0.1, 'weight_decay': 0.0001}
...
[Source] Epoch 50/50 Train loss 0.6470 acc 0.7747 Val loss 0.7164 acc 0.7537
=== Preparing CIFAR-10-C target domain ===
```

```
Static (No Adaptation) metrics on the CIFAR-10-C stream: {'accuracy': 0.6590954773869346, 'precision_macro': 0.6696963906288147,
'recall_macro': 0.6590954661369324, 'f1_macro': 0.6576816439628601, 'rmse': 0.21556910872459412}
...
Oracle (target-supervised) metrics on CIFAR-10-C stream: {'accuracy': 0.7, 'precision_macro': 0.6992928981781006, 'recall_macro':
0.699999988079071, 'f1_macro': 0.6992831230163574, 'rmse': 0.20176437497138977}
[Deploy] Saved model to ./checkpoints/qmembn_oracle_cifar10c.pth
...
[Stage II] Final metrics on adapted stream: {'accuracy': 0.6678391959798995, 'precision_macro': 0.6743624806404114, 'recall_macro':
0.6678391695022583, 'f1_macro': 0.666246771812439, 'rmse': 0.2124655842781067}
[Deploy] Saved model to ./checkpoints/qmembn_adapt_clean_cifar10c.pth
...
[Stage II] Final metrics on adapted stream: {'accuracy': 0.6662311557788945, 'precision_macro': 0.671379566192627, 'recall_macro':
0.6662311553955078, 'f1_macro': 0.6635282635688782, 'rmse': 0.21326902508735657}
[Deploy] Saved model to ./checkpoints/qmembn_naive_clean_cifar10c.pth
```

```
...
[Stage II] Final metrics on adapted stream: {'accuracy': 0.5516582914572864, 'precision_macro': 0.6218993663787842, 'recall_macro':
0.5516583323478699, 'f1_macro': 0.564264714717865, 'rmse': 0.26018691062927246}
...
[Stage II] Final metrics on adapted stream: {'accuracy': 0.5458291457286432, 'precision_macro': 0.6163973808288574, 'recall_macro':
0.545829176902771, 'f1_macro': 0.5571845173835754, 'rmse': 0.25905847549438477}
...
=== Objective 1: Static vs Q-MemBN+ on CIFAR-10-C ===
Static accuracy:      0.6591
Adapted accuracy:     0.6678
Absolute gain:        0.87 percentage points

=== Objective 2: Gap to oracle (CIFAR-10-C) ===
Oracle accuracy:      70.00 %
Gap (oracle - static):  4.09 pp
Gap (oracle - adapted): 3.22 pp
Gap reduction:          21.35 %

=== Objective 3: Robustness to 20% poisoned stream ===
Naive adaptation drop:   12.20 percentage points
Q-MemBN+ drop:           11.46 percentage points

=== Objective 4: Efficiency ===
Model size:              11.17M parameters
Stage-II latency (B=1):  28.08 ms per batch
```

For Reference ...

94.74%

Somebody's ResNet-18 on **CIFAR-10 after 150 epochs**

77.47%

Somebody's ResNet-18 on **CIFAR-10-C after 150 epochs**

17.27%

Their transition's **Accuracy Drop**

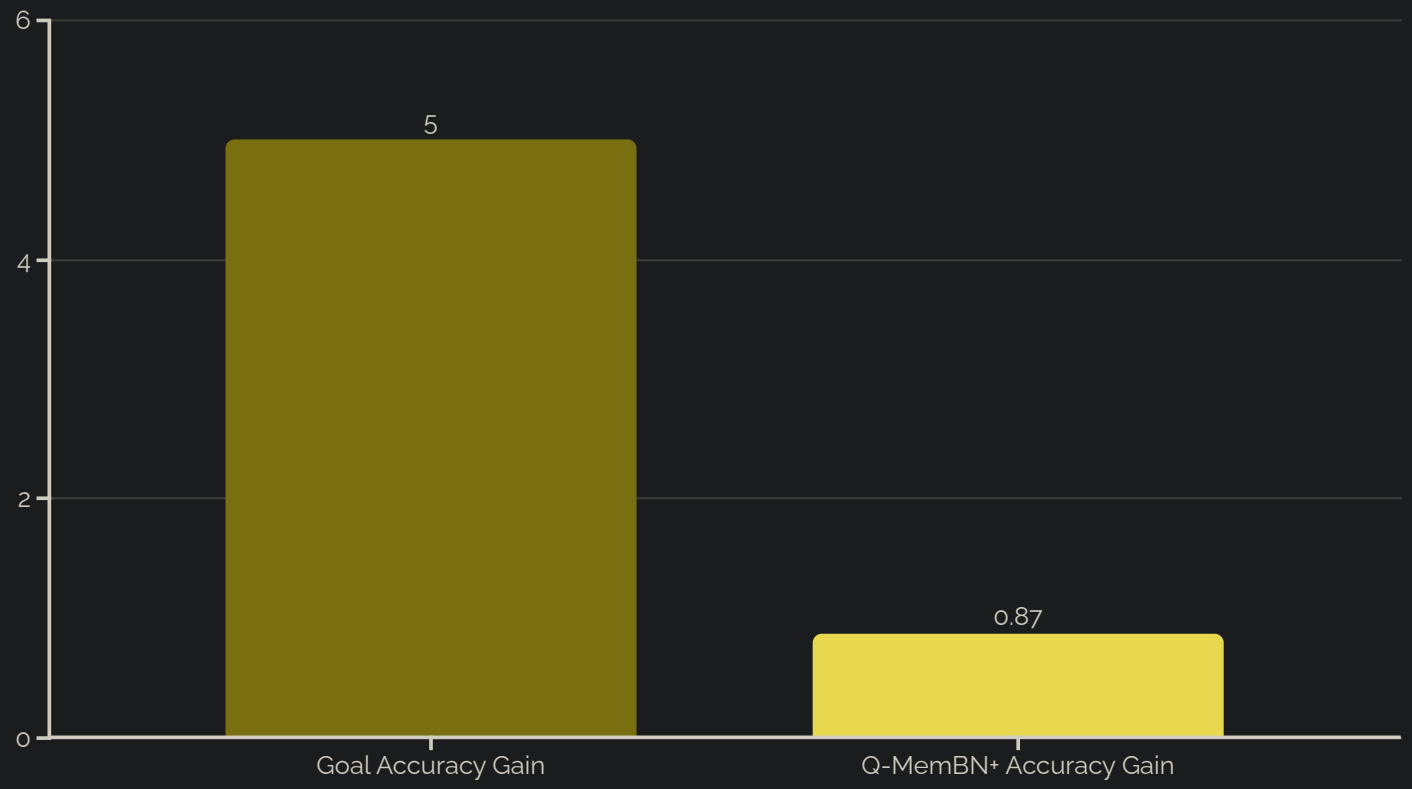
10.22%

Our transition's **Accuracy Drop**

Data Source:

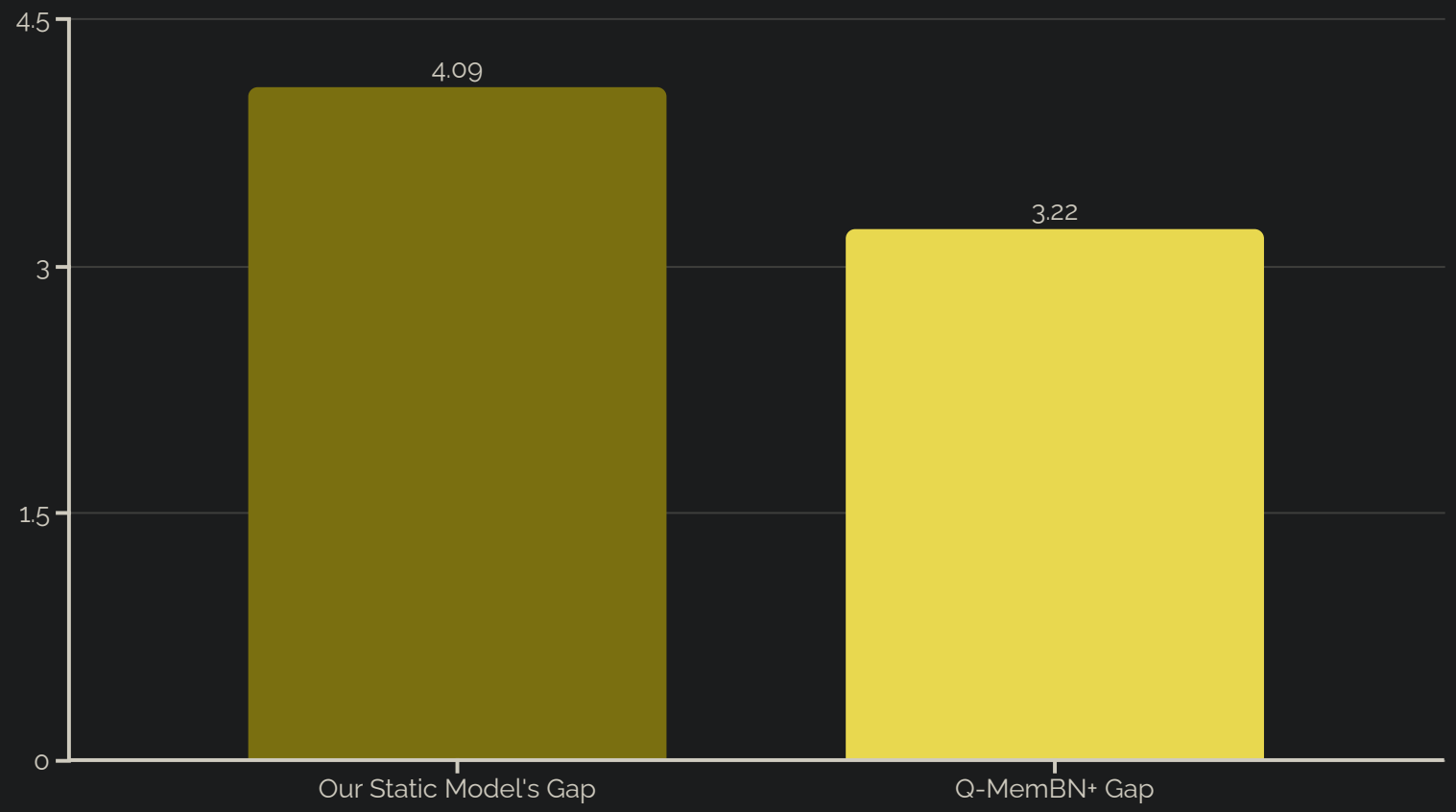
https://www.researchgate.net/publication/350647157_Misclassification-Aware_Gaussian_Smoothing_improves_Robustness_against_Domain_Shifts

Objective 1 — Accuracy Under Shift *(ppt)*



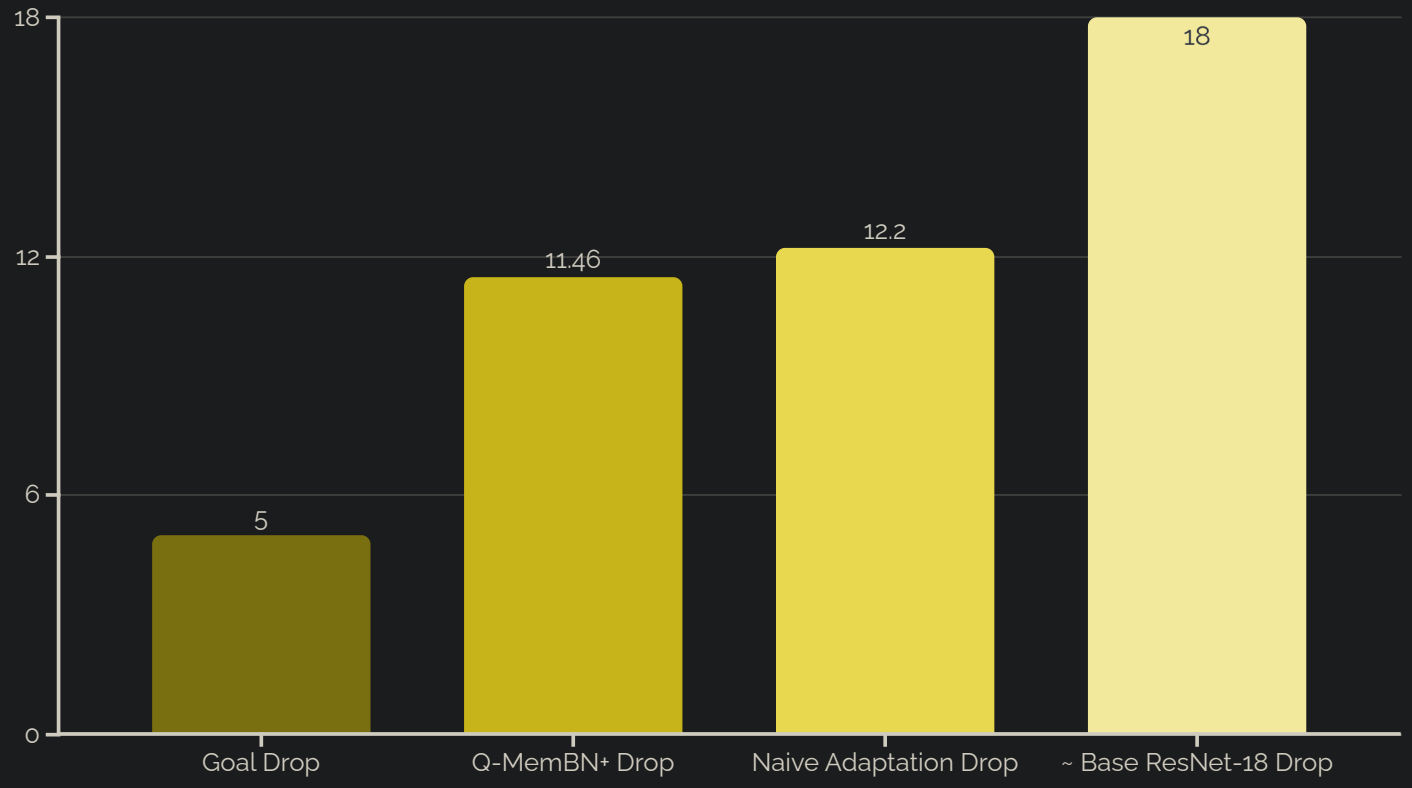
- ☐ Got an accuracy gain of 0.87%, 4.13% below target
- ☐ A statistically significant gain on the CIFAR-10 to CIFAR-10-C transition *(50 epochs; macro-precision, recall, F1-score, RMSE consistent)*
- ☐ *'Me vs. Me'*

Objective 2 — Oracle Gap *(ppt)*



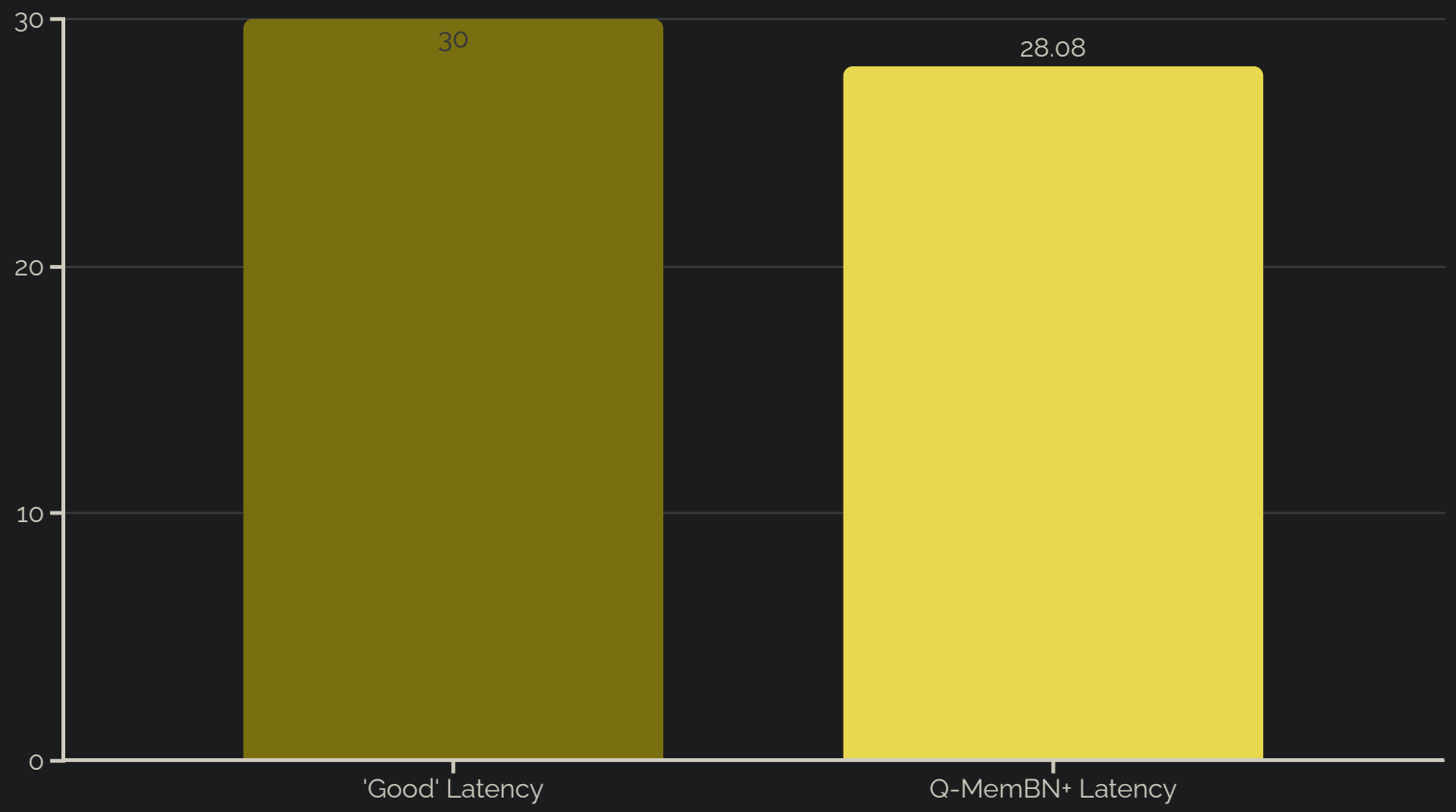
- ☐ Got a gap reduction of 21.35%, 28.65% below target
- ☐ Validates our few-shot anchoring & quantile-memory updates (our unified approach can bridge the source-/target-trained gap)

Objective 3 — Robustness to 20% Poisoned Inputs *(ppt)*



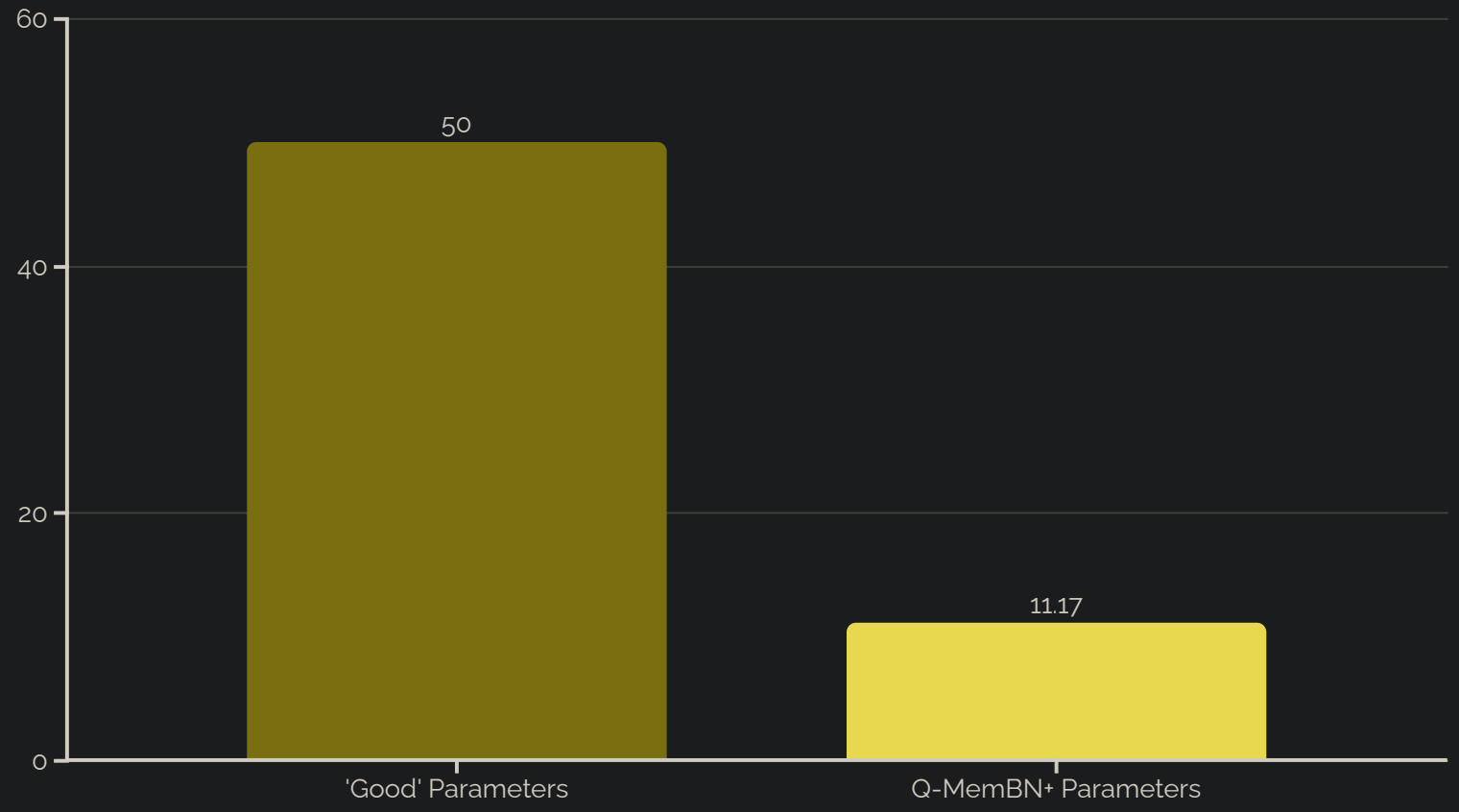
- ☐ **Got an accuracy drop of 11.46%, 6.46% above target**
- ☐ **This is 0.74% above Naive Adaptation**
- ☐ **A Potential Problem:** Drift detection mis-identified sequential drift **—>** unnecessary reset **==>** unnecessary adaptation loss
- ☐ **Poisoned Stream Process:** Gaussian noise of $N(0, 2.5^2)$ added per-pixel on 20% of images *(highly perturbed)*

Objective 4 — Latency *(ms)* *{b = 1}*

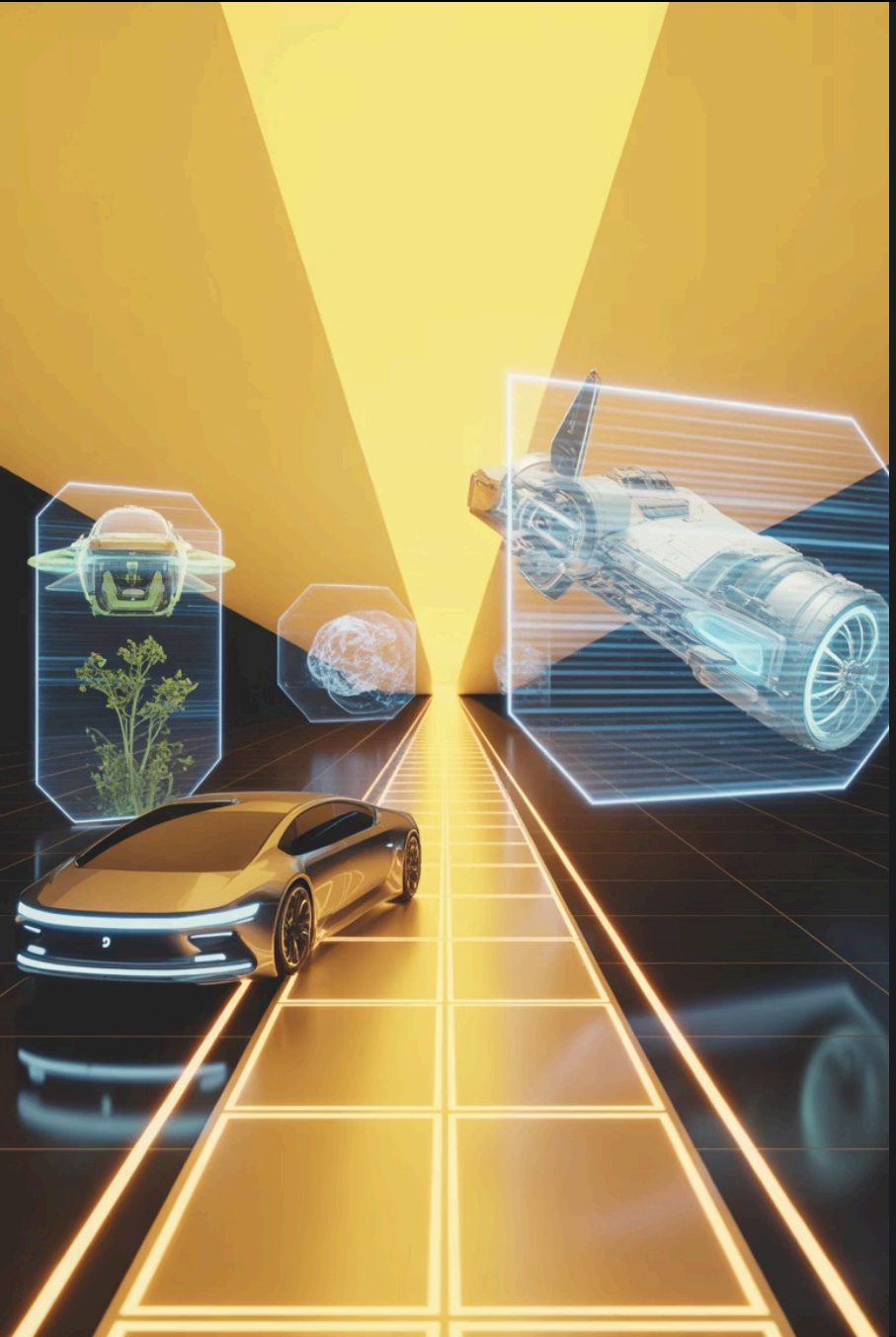


- [ ] Got a latency of 28.08 ms, 1.92 ms below target
- [ ] Quick enough for deployment on edge small-vision devices *such as* UAVs & UGVs.

Objective 4 — Parameters *(M)*



- [ ] Employed **11.17 M** parameters, **38.83 M below target**
- [ ] Cheap enough for **deployment on edge small-vision devices** ***such as*** **Assistive Devices and Wearables.**



Lessons Learned & a To Do

Ambitious Objectives

Our initial targets were too optimistic. Research progress is incremental—we stand on the shoulders of giants and reach just a little further into the unknown.

Meaningful Progress

We, through Q-MemBN+, demonstrated that a unified approach can simultaneously deliver outlier robustness, statistical stability, and guided adaptation.

Future Work

1. Refine our drift detection logic to reduce false positives.
2. Explore the feasibility of multi-prototype memory per class.
3. Test on real-world edge deployments.

Conclusion

> Our Q-MemBN+ offers a novel, long-overdue pathway to test-time robustness in small vision models. By balancing theoretical innovation with real-world development needs, we provide an alluring pathway to a solution for competent and reliable edge AI deployment.

Open Source

Our code, documentation, and pipeline is available on GitHub for learning and research.

Practical Impact

This is ready for deployment on drones, warehouse robots, security cameras, and the such right now.

A Novel Research Avenue

We justified the pursuance of unified frameworks in TTA research.

PRACTICAL IMPACT

NOVEL RESEARCH AVENUE

References

1. Wang, Dequan, et al. (2021). Tent: Fully Test-Time Adaptation by Entropy Minimization. International Conference on Learning Representations (ICLR). OpenReview ID: rUTXo-x4DX[3][61]
2. Kang, Juwon, et al. (2024). MemBN: Robust Test-Time Adaptation via Batch Norm with Statistics Memory. European Conference on Computer Vision (ECCV), pp. 467–483. DOI: 10.1007/978-3-031-25068-7_27 [7][24]
3. Park, Hyejin, et al. (2024). MedBN: Robust Test-Time Adaptation against Malicious Test Samples. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:2403.19326 [cs.LG][8][6]
4. Luo, Siqi, et al. (2025). Enhancing Test Time Adaptation with Few-shot Guidance. arXiv preprint arXiv:2409.01341v2 [cs.CV]. (Accepted to TBD conference). DOI: 10.48550/arXiv.2409.01341 [28][10]
5. Ma, Jing. (2024). Improved Self-Training for Test-Time Adaptation. CVPR. OpenAccess PDF: CVPR2024_Ma_TestTimeAdaptation.pdf[62]
6. Zhang, Zhen-Yu, et al. (2024). Test-time Adaptation in Non-stationary Environments via Adaptive Representation Alignment. Advances in Neural Information Processing Systems (NeurIPS). (Poster Presentation)[47][48]
7. Wang, Shuo, et al. (2023). Feature Alignment and Uniformity for Test Time Adaptation. CVPR. DOI: 10.1109/CVPR46375.2023.00968 (Referred to as TSD in text)[63]
8. Boudiaf, Malik, et al. (2022). Parameter-Free Online Test-Time Adaptation. CVPR. DOI: 10.1109/CVPR52688.2022.00207 (Introduced LAME consistency regularization)[58][64]
9. Niu, Yulei, et al. (2022). Efficient Test-Time Model Adaptation Without Forgetting. International Conference on Machine Learning (ICML). PMLR. (Introduced EATA early stopping)[65][64]
10. Schneider, Steffen, et al. (2020). Improving Robustness Against Common Corruptions by Covariate Shift Adaptation. NeurIPS. arXiv:2006.16971 (Test-Time Normalization baseline)[66]

Thank you for your time and attention. :)

Your Questions?