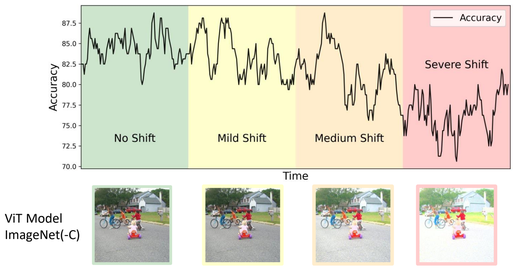
Protected Test-Time Adaptation via Online Entropy Matching: A Betting Approach

Yarin Bar* Shalev Shaer* Yaniv Romano

Technion – Israel Institute of Technology

ML deployment in the wild is a wild challenge

- ML models demonstrate unexpected degradation when facing unfamiliar environments
- Want to enhance robustness to test-time drifting data in an online manner



- **Challenges**
  (1) Lack of up-to-date labeled data
  (2) Test instances are not available a-priori (sudden/continual shift)

Our solution: from monitoring to adaptation

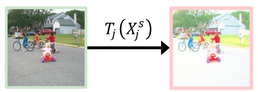
This work
(1) How can we **alert** that the model "behaves" differently?
(2) How to use this info. to **correct** the model's "behavior"?
→ without annotating new samples

Problem setup: test-time adaptation

- **Source domain**: in-dist. labeled data $(X^s, Y^s) \sim P^s_{XY}$
  $X^s \in \mathcal{X}$: covariates (e.g., an image)
  $Y^s \in \{0,1,\dots,K\}$: labels (e.g., tricycle)
- **Target domain**: a stream of unlabeled test points $\left(X_j^t, ?\right)$, $j = 1,2,\dots$
  $X_j^t \sim P_X^t$ is obtained by applying an unknown "shifting" function T_j to a fresh X_j^s
- T_j can vary over time

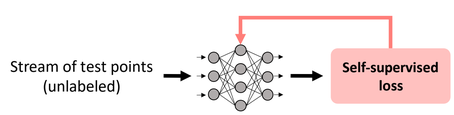
Assumptions (invariance)
Despite the shift
- The label remains the same
- The prediction difficulty (uncertainty) remains the same

Source (ex. ImageNet) **Target** (ex. ImageNet-C)



Background: self-training at test time

Input: a pre-trained classifier $\hat{f}_{\hat{\theta}}$ fitted to labeled source data

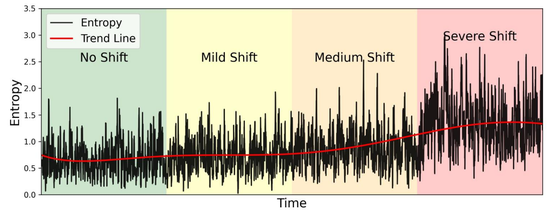


- State-of-the-art loss: **entropy minimization**
- Main issue: overconfident predictions
  → bad calibration, even if applied to in-distribution data

Our Approach: Build Invariance via
Online Entropy Matching

Key principle in domain adaptation
drive invariance to shifts
through distributional matching

Key observation
pre-trained classifier's entropy
reflects dist. shifts: lack of invariance



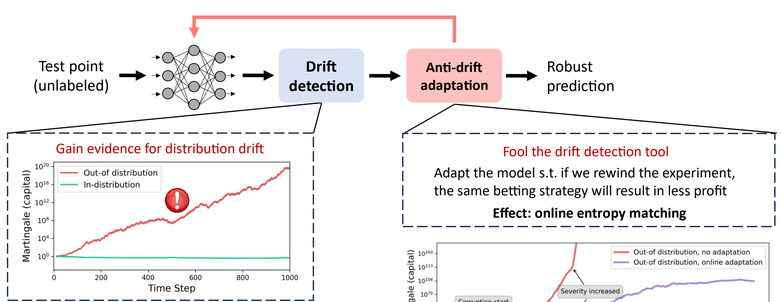
Source entropy $Z^s = \ell^{\text{ent}}(\hat{f}_{\hat{\theta}}(X^s))$, evaluated on in-distribution data
$\hat{f}_{\hat{\theta}}$: a pre-trained model

Target entropy $Z_j^t = \ell^{\text{ent}}(f_\theta(X_j^t))$, evaluated online
f_θ : self-trained model

Adapt the model by matching the distribution of Z^s and Z_j^t online for all $j = 1,2,\dots$

Online Entropy Matching

How can we match the entropy distributions online (dist. is changing over time)?



Testing by betting

Bet money on how much the model's test entropies deviate from source entropy dist.

Proposition: **It's a fair game** (*non-negative martingale*)
- No shift? capital will not grow in expectation
- Significant capital growth? strong evidence for shift

Theorem: **entropy matching** (*optimal transport*)
If the bet is ideal = true likelihood ratio, the anti-drift correction is the optimal transport map from target to source ents. (w.r.t.Wass. distance)

POEM: algorithm

Let's play a game. Start with an initial "toy money", $S_0 = 1$

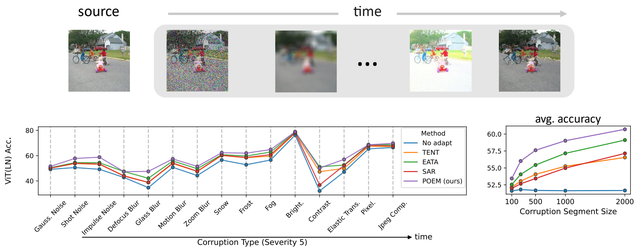
- For each $t = 1, 2, \dots$

Monitoring
1) **Place a bet** on how much the coming entropy will deviate from the source ent. dist.
2) **Collect** a test point X_j^t and compute its entropy $Z_j^t = \ell^{\text{ent}}(f_\theta(X_j^t))$
3) **Reveal the truth** (how?)
4) **You win the bet?** your wealth S_t is increased; otherwise, it's decreased

Adaptation
5) **Anti-drift correction**: derive an adapted-entropy $\tilde{Z}_j$ s.t.:
if we rewind the experiment, the same betting strategy will result in less profit
6) **Self-training**: update f_θ by minimizing $\ell^{\text{match}}\left(Z_j^t(\theta), \tilde{Z}_j\right) = \frac{1}{2}\left(Z_j^t(\theta) - \tilde{Z}_j\right)^2$

Continual shifts: varying corruptions

- Segments of 1,000 examples from each corruption, severity 5 (highest)

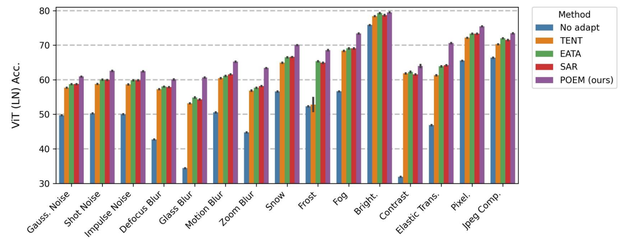


ViT Model
ImageNet(-C)

- Superior accuracy
- Faster adaptation (powerful betting)

Single shift: severity level 5 (highest)

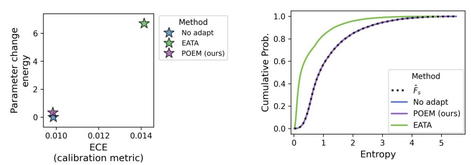
- Apply adaptation on the whole ImageNet-C test set, for each corruption



Superior performance across the board

In-distribution test data: comparing to state -of-the-art

Accuracy is ≈ 84.5 for all methods



"No-harm" effect
- Same accuracy and calibration as the pre-trained model
- Minimal change in model parameters

GitHub

Paper

References

- Vladimir Vovk (2021). Protected probabilistic regression. In Technical report
- Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell (2020). Tent: Fully test-time adaptation by entropy minimization. In International Conference on Learning Representations
- Glenn Shafer and Vladimir Vovk (2019). Game-theoretic foundations for probability and finance. In Wiley Series in Probability and Statistics
- Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference (2021). In Statistical Science, 38(4):576–601
- Aaditya Ramdas and Ruodu Wang. Hypothesis testing with e-values (2024). In arXiv Preprint
- Peter Grünwald, Rianne de Heide, and Wouter M Koolen. Safe testing (2020). In IEEE Information Theory and Applications Workshop (ITA), pages 1–54
- Francesco Orabona and Dávid Pál. Scale-free online learning (2018). In Theoretical Computer Science,716:50–69

* Equal Contribution