Concept-level Debugging of Part-Prototype Networks

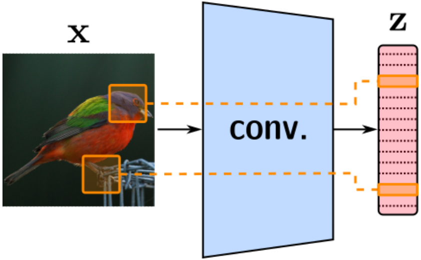
Andrea Bontempelli[1], Stefano Teso[1], Katya Tentori[1], Fausto Giunchiglia[1,2], Andrea Passerini[1]

ICLR 2023

[1] University of Trento, Italy
[2] Jilin University, China

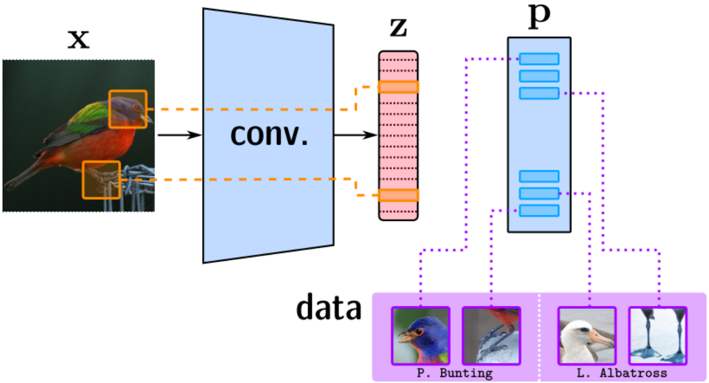
Part-Prototype Networks (aka ProtoPNets)

Embedding stage



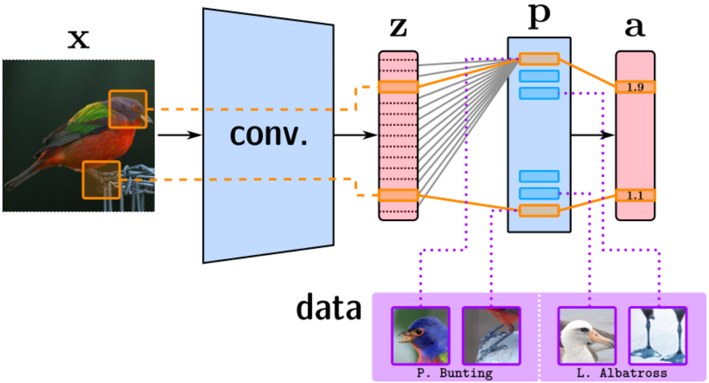
Part-Prototype Networks (aka ProtoPNets)

Part-Prototype stage



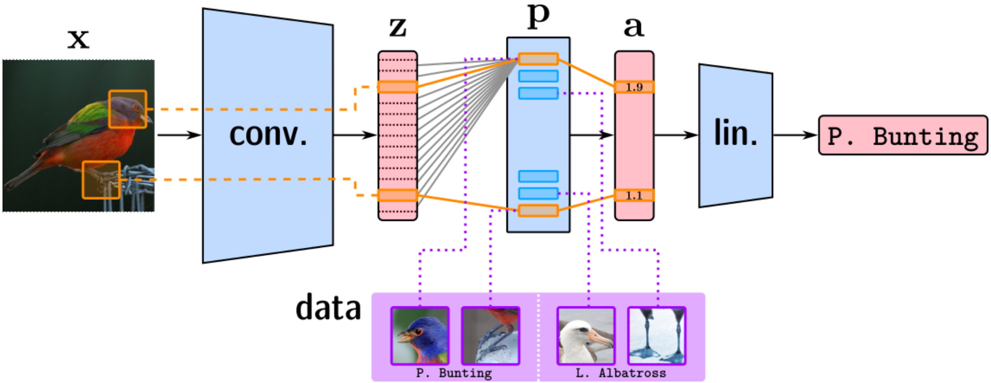
Part-Prototype Networks (aka ProtoPNets)

Part-Prototype stage



Part-Prototype Networks (aka ProtoPNets)

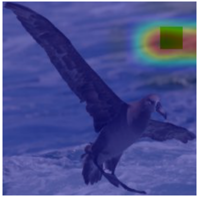
Aggregation stage

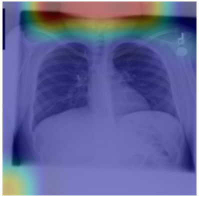


Confounding in ProtoPNets

Explanations expose confounds picked up from training data as part-prototypes.

Models exploit confounds to **maximize** training set performance.

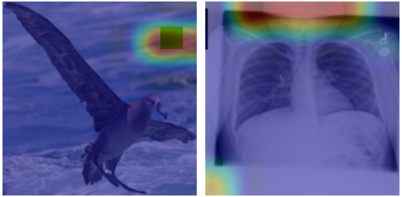




Confounding in ProtoPNets

Explanations expose confounds picked up from training data as part-prototypes.

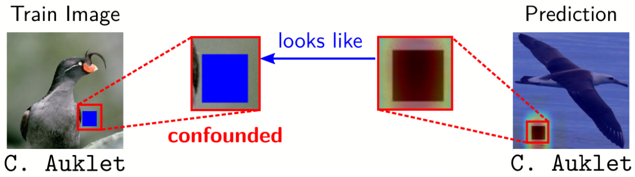
Models exploit confounds to **maximize** training set performance.



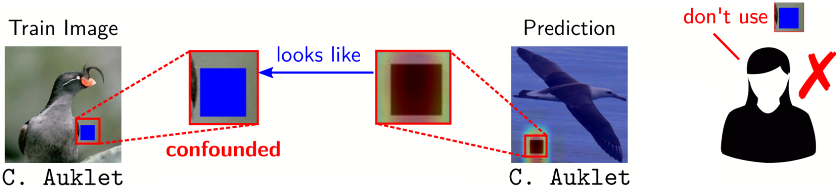
Issue: they impact **generalization** and **out-of-distribution** performance, also trustworthiness! [Lapuschkin et al., 2019].

How to **dissuade** the model from acquiring confounds?

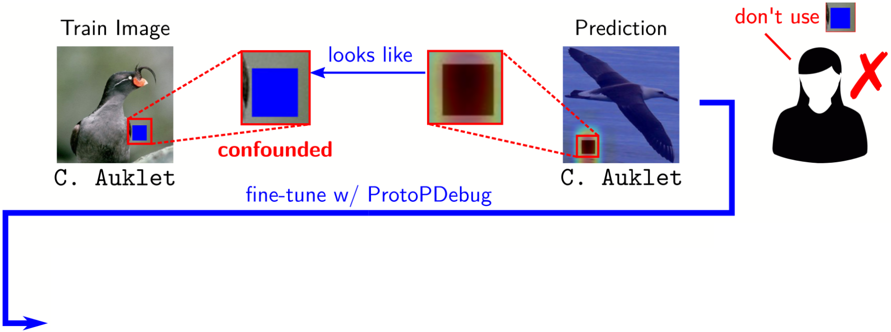
Concept-level debugging with ProtoPDebug



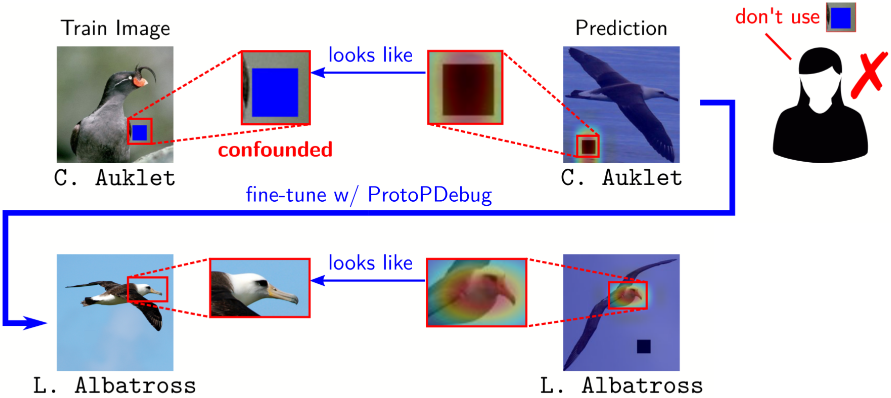
Concept-level debugging with ProtoPDebug



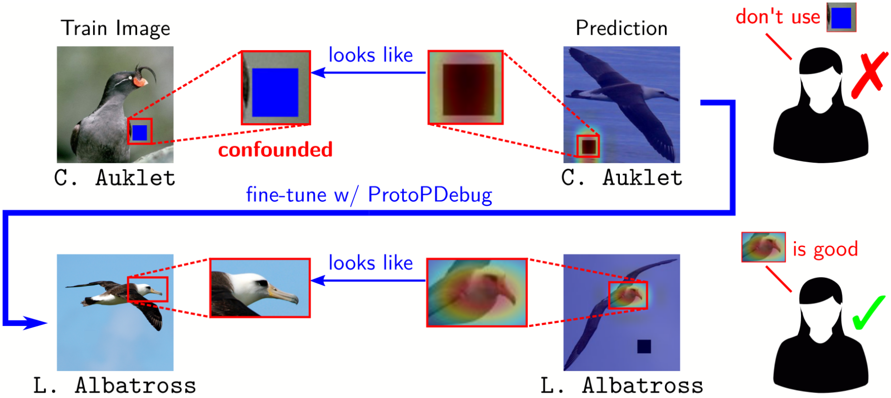
Concept-level debugging with ProtoPDebug



Concept-level debugging with ProtoPDebug



Concept-level debugging with ProtoPDebug



Empirical Analysis

Exp. 1 Concept-level debugging is useful …

Exp. 2 … even for natural confounds …

Exp. 3 … and in high-stakes applications.

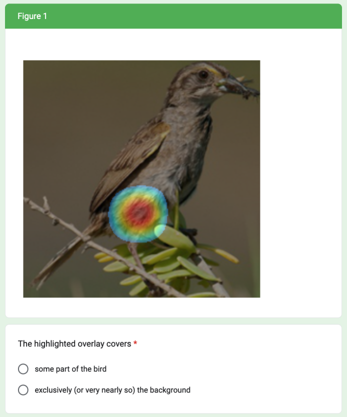


Figure 1

The highlighted overlay covers *

○ some part of the bird

○ exclusively (or very nearly so) the background

Take-aways

- ProtoPNets, like other models, pick up **confounds** from the data
- ProtoPDebug is an effective **concept-level debugger** for ProtoPNets
- human supervisor provides **click-based feedback** to forget or to keep part-prototypes
- leads to **better models**, speed up convergence and avoid relapse

Thank You!

andrea.bontempelli@unitn.it

🗎 https://arxiv.org/abs/2205.15769

</> https://github.com/abonte/protopdebug

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019).
Unmasking clever hans predictors and assessing what machines really learn.
Nature communications.