A Simple Baseline for Cross-domain Few-shot Text Classification

Chen Zhang, Dawei Song

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

Outline

- Preliminary
  - Few-shot Text Classification
  - Cross-domain Text Classification
- Cross-domain Few-shot Text Classification (XFew) Fails Few-shot Classifiers
- A Simple Baseline Performs Considerably
- Experiments
- Conclusion

Text Classification

- Given a training dataset $D^{tr} = \{(x_i, y_i)\}_i$, and a test dataset $D^{ts} = \{(x_i, y_i)\}_i$.
- Learning a classifier $f_\theta : x \rightarrow y$ on D^{tr} so that it can perform perfectly on D^{ts}.
- Typically, $x \sim P(X)$ and $y \in Y$ hold across D^{tr} and D^{ts}; That is, no domain shift or concept shift.
- E.g., intent detection
    - “how is the weather today?” => “weather”.

Few-shot Text Classification

- Text classification under concept shift; That is, $Y^{tr} \cap Y^{ts} = \emptyset$.
- E.g., new intents may emerge now and then
  - These new intents usually come with only a few examples, <100 per class.
  - Similar to cold start phenomenon in recommender systems.
- Conventional classifiers can not generalize.

Cross-domain Text Classification

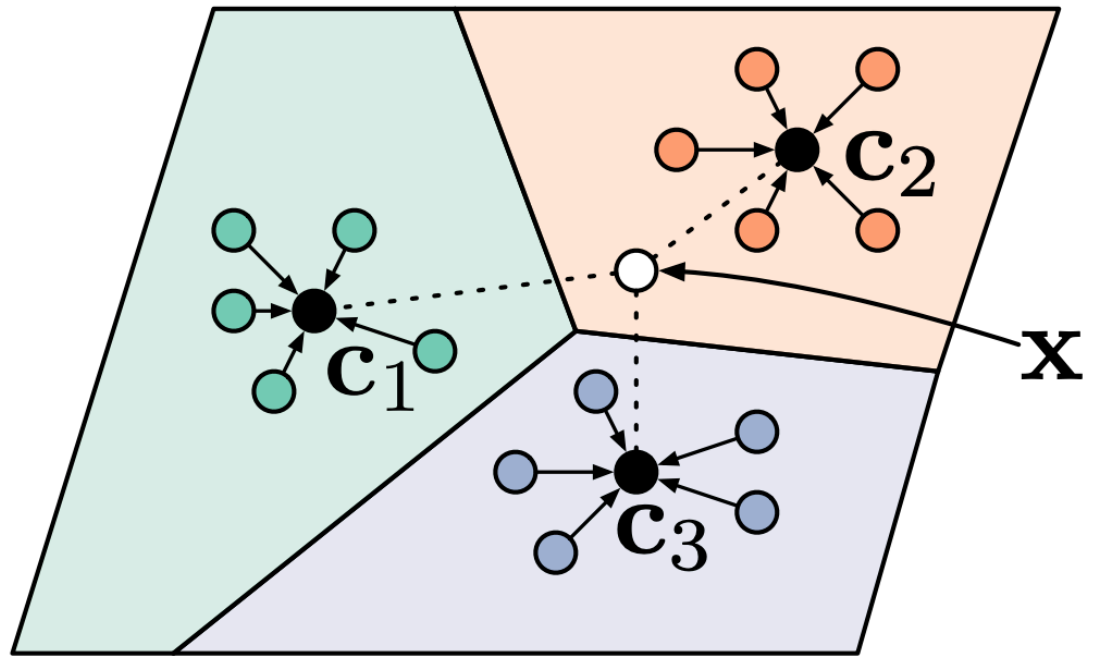
- Text classification under domain shift; That is, $P^{tr}(X) \neq P^{ts}(X)$.
- E.g., we have a plenty of sentiment annotations from the restaurant domain but we want to perform sentiment classification on the electronics domain
  - “The sushi here is great.” => “positive”.
  - “The resolution of the display is terrific.” => “?”.
- Conventional classifiers get degraded performance.

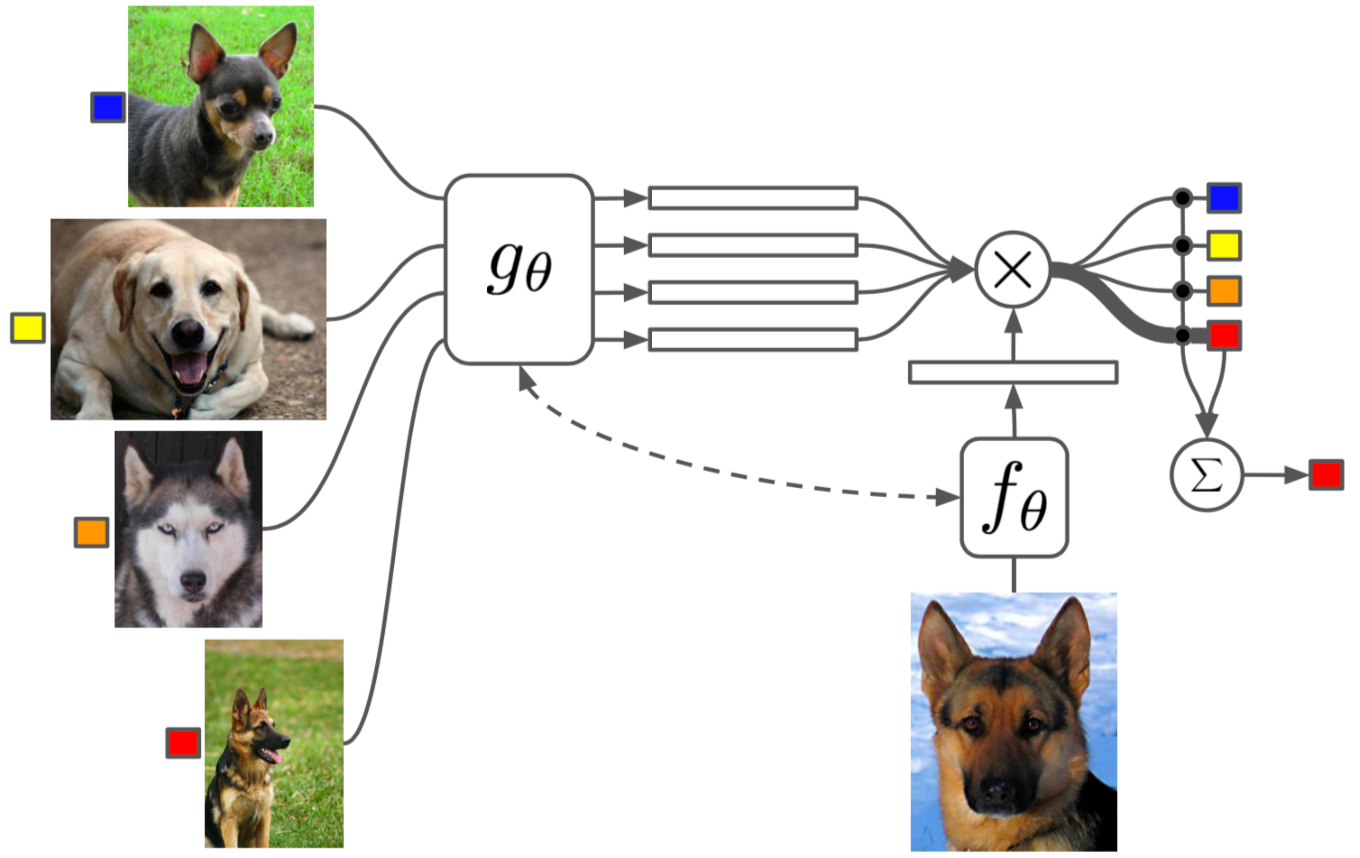
Cross-domain Few-shot Text Classification (XFew)

- Sometimes a new domain comes with a new label set.
- E.g.,
  - “How is the weather today?” => “weather”. (“siri” domain)
  - “How to cancel the credit card?” => “cancellation of credit card.” (banking domain)
- A small yet important step towards achieving lifelong text classification.

Few-shot Classifiers

- Few-shot classifiers
  - Metric-based





  - Optimization-based

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
  end for Note: the meta-update is using different set of data.
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
end while

Algorithm 2 Reptile, batched version

Initialize θ
for iteration $= 1, 2, \ldots$ **do**
  Sample tasks $\tau_1, \tau_2, \ldots, \tau_n$
  for $i = 1, 2, \ldots, n$ **do**
    Compute $W_i = \mathrm{SGD}(L_{\tau_i}, \theta, k)$
  end for
  Update $\theta \leftarrow \theta + \beta \frac{1}{n} \sum_{i=1}^{n} (W_i - \theta)$
end for

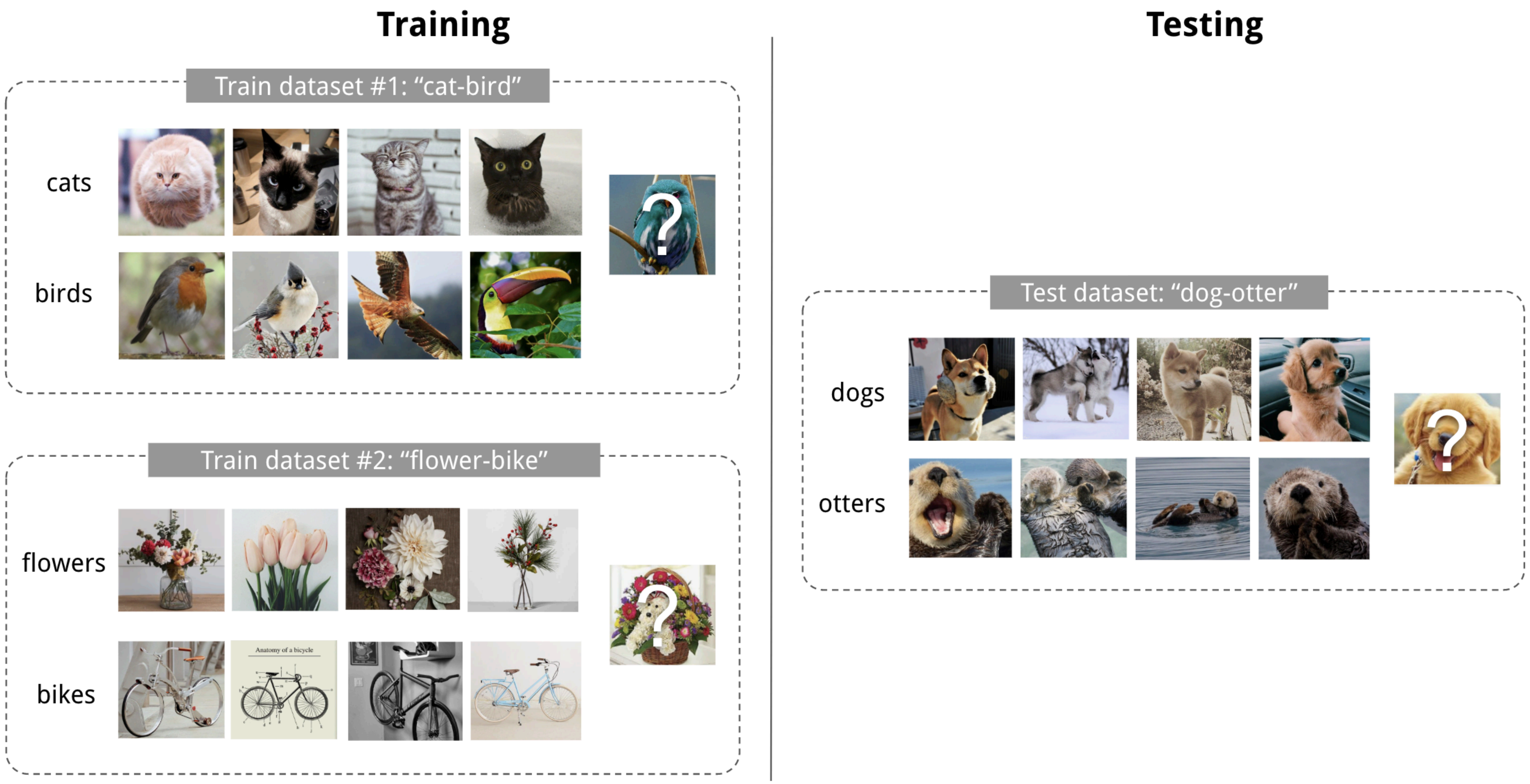
  - Etc.

N-way K-shot Setting

- N-way K-shot setting is arranged for evaluation of few-shot classifiers
    - N-way stands for N classes, and K-shot stands for K examples per class.
    - An episode contains N-way K-shot support examples, and N-way Q-shot query examples.
    - A few-shot classifier should adapt to the offered support examples and be evaluated on the query examples during an episode.
    - An evaluation result is obtained by averaging results of multiple episodes.
- In order to align training and evaluation, previous few-shot classifiers conduct episode-based arrangement also at training time.

N-way K-shot Setting

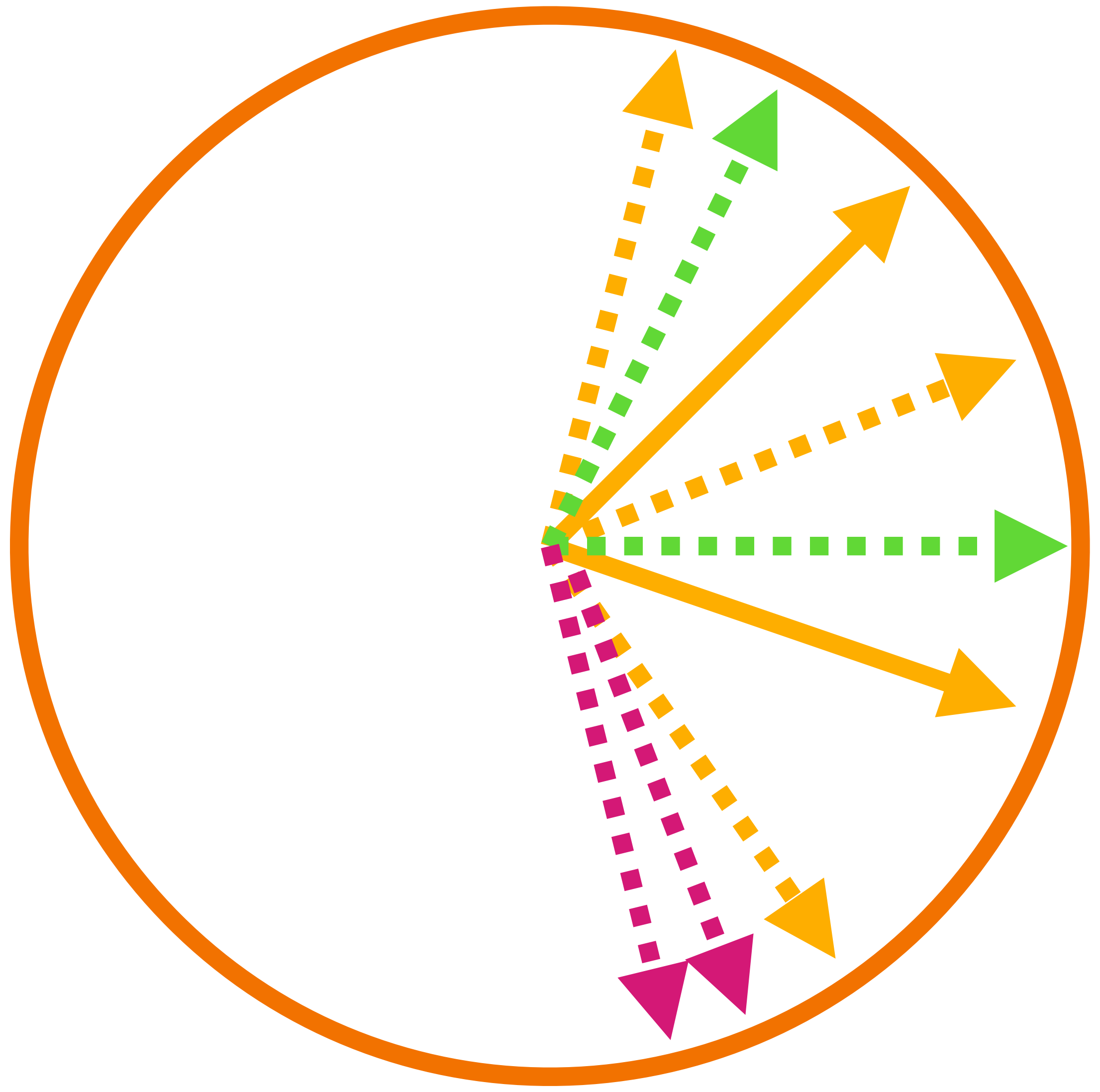
- An illustration with the image scenario (here, 2-way 4-shot).



Few-shot Classifiers Can Fail

- Given the limited number of classes seen at each episode, we hypothesize the classifier can not gain a good insight of class manifolds, or a good class capacity.

- N-way sampling shall be good for in-domain scenarios, but can fail few-shot classifiers for cross-domain scenarios.



Solid lines - sampled classes
Dashed lines - other classes
Yellow - seen classes
Green - in-domain unseen classes
Purple - cross-domain unseen classes

A few-shot classifier (w/ small class capacity) can hardly extrapolate.

A Simple Baseline Performs Considerably

- N-way K-shot setting seems to be not suitable for cross-domain scenario, we are driven by this thus propose a simple baseline, named PtNet.
- Train conventionally (larger class capacity compared with training schemes constrained by the N-way K-shot setting).
- Induct classifier weights instantly (able to adapt in a few-shot manner required by the N-way K-shot setting).

$$\mathbf{w}_j = \sum_{x_i \in \mathcal{S}_j} f_\theta(x_i)/k \qquad \mathbf{y}_{i,j} = \mathrm{softmax}(\alpha \cdot \mathbf{w}_j^\top f_\theta(x_i)/\|\mathbf{w}_j\|\|f_\theta(x_i)\|), \quad x_i \in \mathcal{Q}$$

Experimental Setup

- Two intent detection datasets, one is from home domain and the other is from banking domain.
- Constructing two in-domain datasets and two cross-domain datasets from above two.
- 5-way {1, 5, 10}-shot setting.

	Home	Banking	Home2Banking	Banking2Home
# (base) classes for training	39	49	56	70
# (base) classes for validation	6	7	7	7
# (novel) classes for test	18	21	77	63

In-domain Evaluation

- Our PtNet can nice results on in-domain evaluation.

Table 2. In-domain comparison results (%) under 5-way 1-shot, 5-shot, and 10-shot settings. Results in **bold** are the best performing ones under each setting.

Model	Home			Banking		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
InductNet	63.19±0.41	71.67±0.31	74.90±0.29	76.72±0.38	85.00±0.27	85.41±0.25
RelationNet	63.38±0.41	74.81±0.33	73.19±0.34	81.31±0.35	88.00±0.26	89.57±0.23
MAML	58.58±0.38	68.44±0.35	71.01±0.32	69.51±0.39	81.58±0.29	84.03±0.26
ProtoNet	**67.91**±0.39	82.92±0.26	86.15±0.22	**82.59**±0.31	**92.20**±0.17	**93.44**±0.14
PtNet	63.82±0.38	**83.32**±0.23	**86.63**±0.20	75.83±0.34	89.80±0.20	92.08±0.16

Cross-domain Evaluation

- Our PtNet is better than baselines on cross-domain evaluation.

Table 3. Cross-domain comparison results (%) under 5-way 1-shot, 5-shot, and 10-shot settings. Results in **bold** are the best performing ones under each setting.

Model	Home2Banking			Banking2Home		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
InductNet	46.15±0.36	54.64±0.33	54.52±0.31	44.54±0.34	57.78±0.32	64.98±0.31
RelationNet	43.55±0.35	56.89±0.32	55.26±0.31	42.10±0.34	55.00±0.32	57.19±0.30
MAML	44.42±0.34	52.07±0.36	35.90±0.31	37.53±0.30	46.31±0.31	44.51±0.33
ProtoNet	**56.90**±0.34	**79.23**±0.28	82.90±0.24	54.62±0.35	78.32±0.27	82.02±0.24
PtNet	50.78±0.34	77.14±0.27	**84.35**±0.21	**58.87**±0.36	**81.34**±0.26	**84.95**±0.22

Conclusion

- Few-shot classifiers perform less promisingly on XFew.
- PtNet can perform considerably better than previous methods on XFew.
- XFew is still challenging.