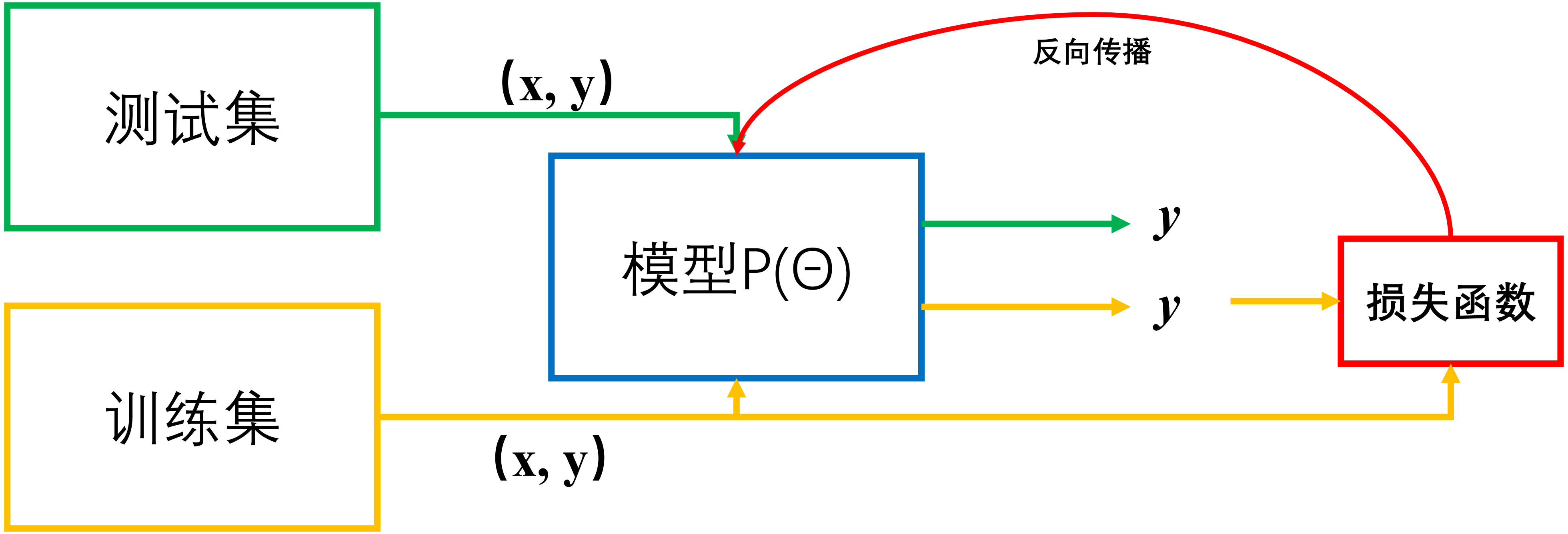
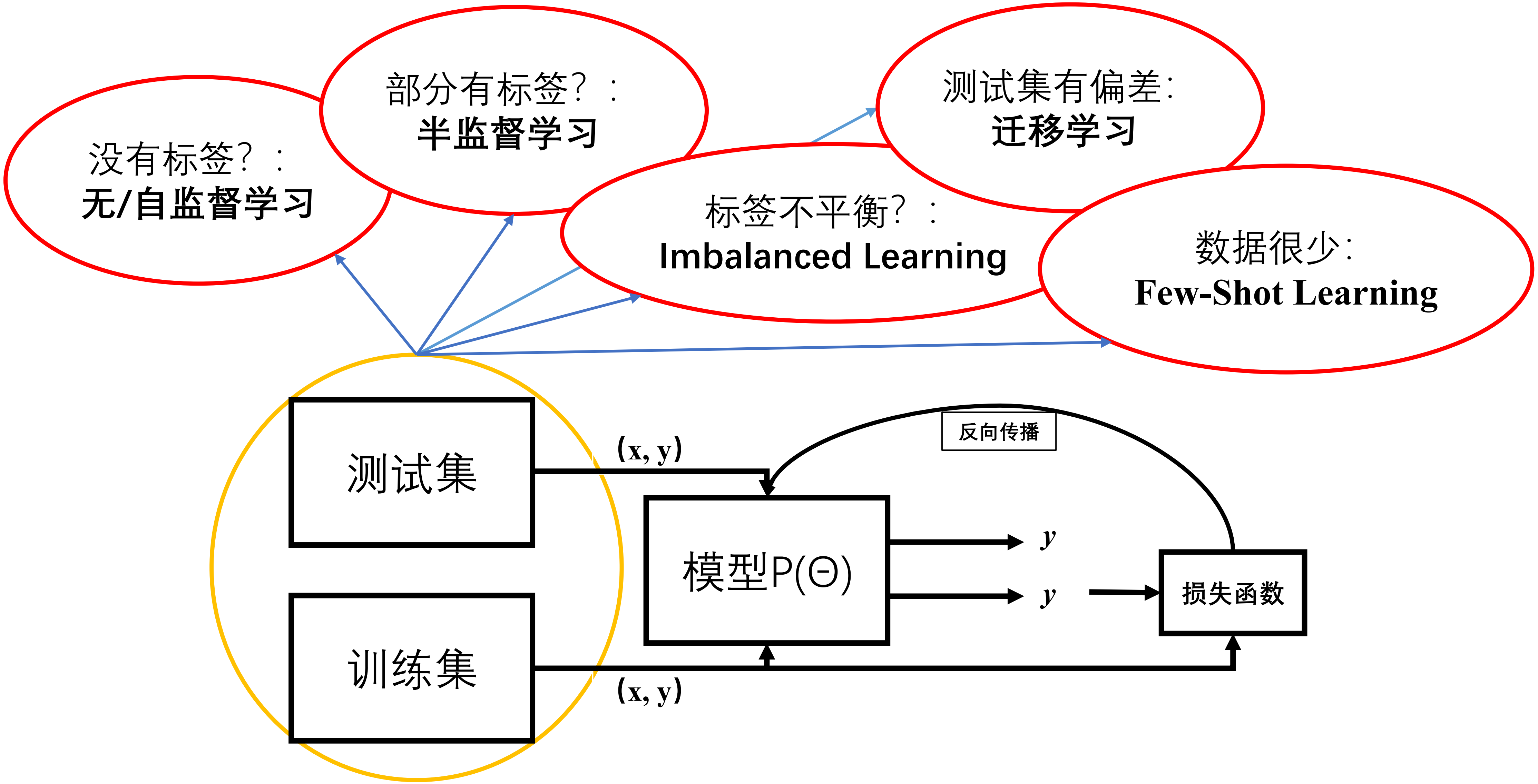
Few-Shot Learning

1. “标准”机器学习



2. “不标准”机器学习


没有标签？：
无/自监督学习
部分有标签？：
半监督学习
标签不平衡？：
Imbalanced Learning
测试集有偏差:
迁移学习
数据很少:
Few-Shot Learning
测试集
训练集
(x, y)
(x, y)
模型P(Θ)
反向传播
y
y
损失函数

3. 问题定义

1.**Machine Learning:** A computer program is said to learn from experience **E** with respect to some classes of task **T** and performance measure **P** if its performance can improve with **E** on **T** measured by **P**.

2.**Few-Shot Learning:** FSL is a type of machine learning problems (Specified by **E**,**T** and **P**) where **E** contains little supervised information for the target **T**.

4. FSL的典型应用场景

1.**Few-Shot due to rare cases:** 由于各种各样的原因没有数据

2.**Few-Shot to reduce data gathering effort and computation cost:** 可以帮助减轻收集数据的负担

3.**Test bed for human-like learning:** 解决小样本问题是迈向真正的人工智能的重要一步

T	E		P
	supervised information	prior knowledge	
character generation [62]	a few examples of new character	pre-learned knowledge of parts and relations	pass rate of visual Turing test
image classification [56]	supervised few labeled images for each class of the target T	raw images of other classes, or pre-trained models.	classification accuracy
drug toxicity discovery [3]	new molecule's limited assay	similar molecules' assays	classification accuracy

5. 符号定义

1. 数据集$D = \{Dtrain, Dtrain\}$，其中$D^{train} = \{(x^{(i)}, y^{(i)})\}$， $D^{\text{test}} = \{xtest\}$;

2. 设$P(x, y)$是输入x和输出y的联合分布;

3. o是x到y全部映射的集合， $o*$是o中对目标函数的最优映射;

4. H是模型决定的假设空间，并使用$h(\cdot;\theta)$表示参数为θ的假设（模型）;

5. 使用符号$\hat{y}$表示模型的输出;

6. 使用符号$l(\hat{y}, y)$表示定义的损失函数

6. Error Decomposition

1. expected risk R on o：

$$R(o) = \int l(o(x), y)\,\mathrm{d}p\,(x, y) = E[l(o(x), y)]$$

2. expected risk R on $h \in H$: $R(h)$

3. empirical risk $R_I(h)$ is used to estimate the expected risk $R(h)$, defined as the average of the sample losses over the training data set D^{train}:

$$R_I(h) = \frac{1}{n} \sum_{i=1}^{I} l(h(x^{(i)}), y^{(i)})$$

and learning is done by **empirical risk minimization.**

6. Error Decomposition

4. $o^* = \arg\min_f R(o)$, where R attains its minima;
$h^* = \arg\min_{h\in\mathcal{H}} R(h)$, where R is minimized with respect to $h \in \mathcal{H}$;
$h_I = \arg\min_{h\in\mathcal{H}} R_I(h)$, where R_I is minimized with respect to $h \in \mathcal{H}$;

5. The ***total error*** of learning taken with respect to the random choice of training set can be decomposed into:

$$\mathbb{E}[R(\tilde{h}_I)] = \underbrace{\mathbb{E}[R(h^*) - o^*]}_{\mathcal{E}_{\text{app}}(\mathcal{H})} + \underbrace{\mathbb{E}[R(h_I) - R(h^*)]}_{\mathcal{E}_{\text{est}}(\mathcal{H},I)}.$$

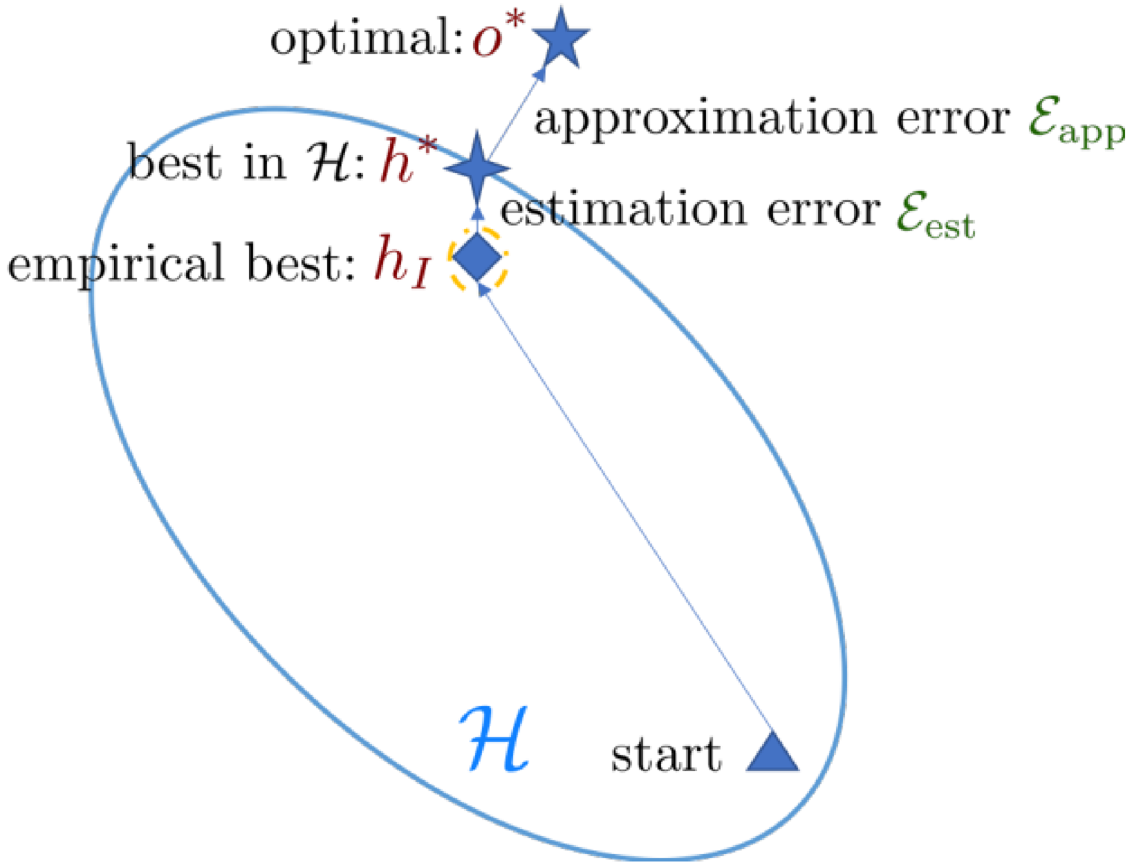
评估所选模型的容量

评估模型可达到的效果

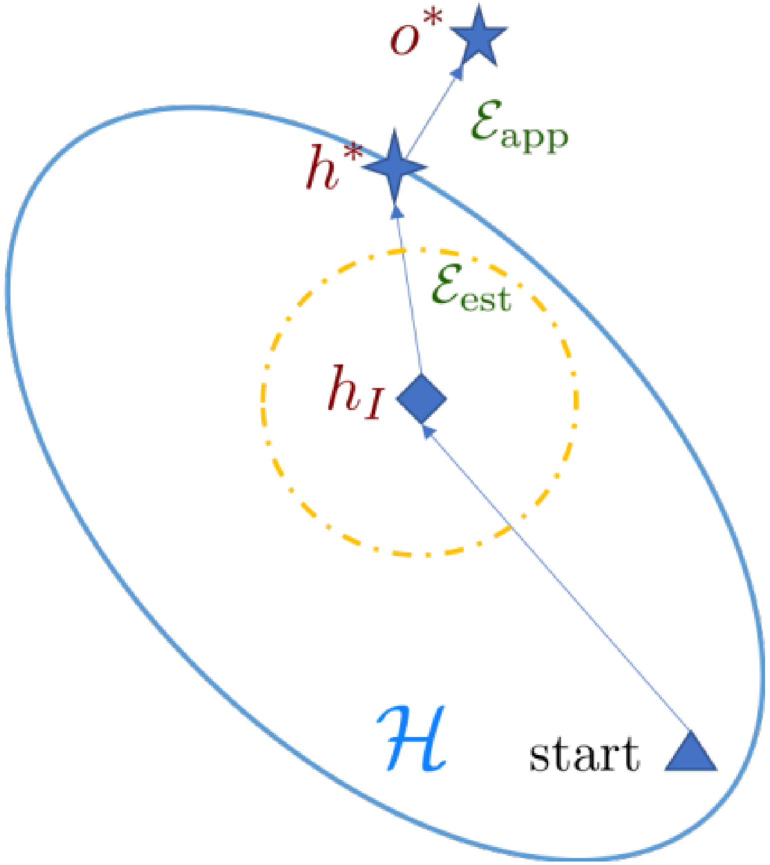
7. Unreliable ERM

1. **sample complexity:** refers to the number of training samples needed to guarantee the effort of ERM.
2. sample complexity increases with **more complicated H** chosen by the model, **higher probability that learned h_I is approximately correct**, and **higher demand of optimization accuracy of algorithm**.
3. $$\mathcal{E}_{\text{est}}(\mathcal{H}, \infty) = \lim_{I \to \infty} \mathbb{E}[R(h_I) - R(h^*)] = 0, \qquad \lim_{I \to \infty} \text{Var}[R(h_I)] = 0,$$
4. Under FSL, the number of available example is smaller than the required sample complexity, leading the ERM is **no longer reliable.**

7. Unreliable ERM



(a) Large I.

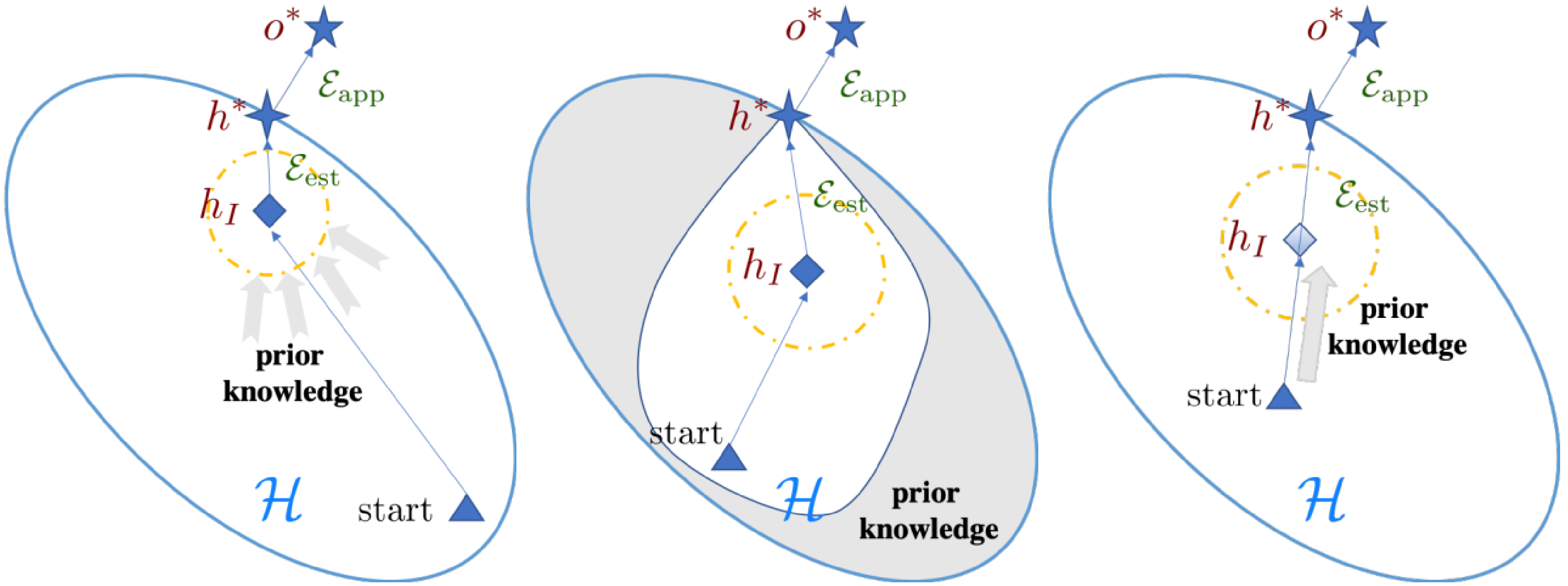


(b) Small I.

8. FSL Taxonomy

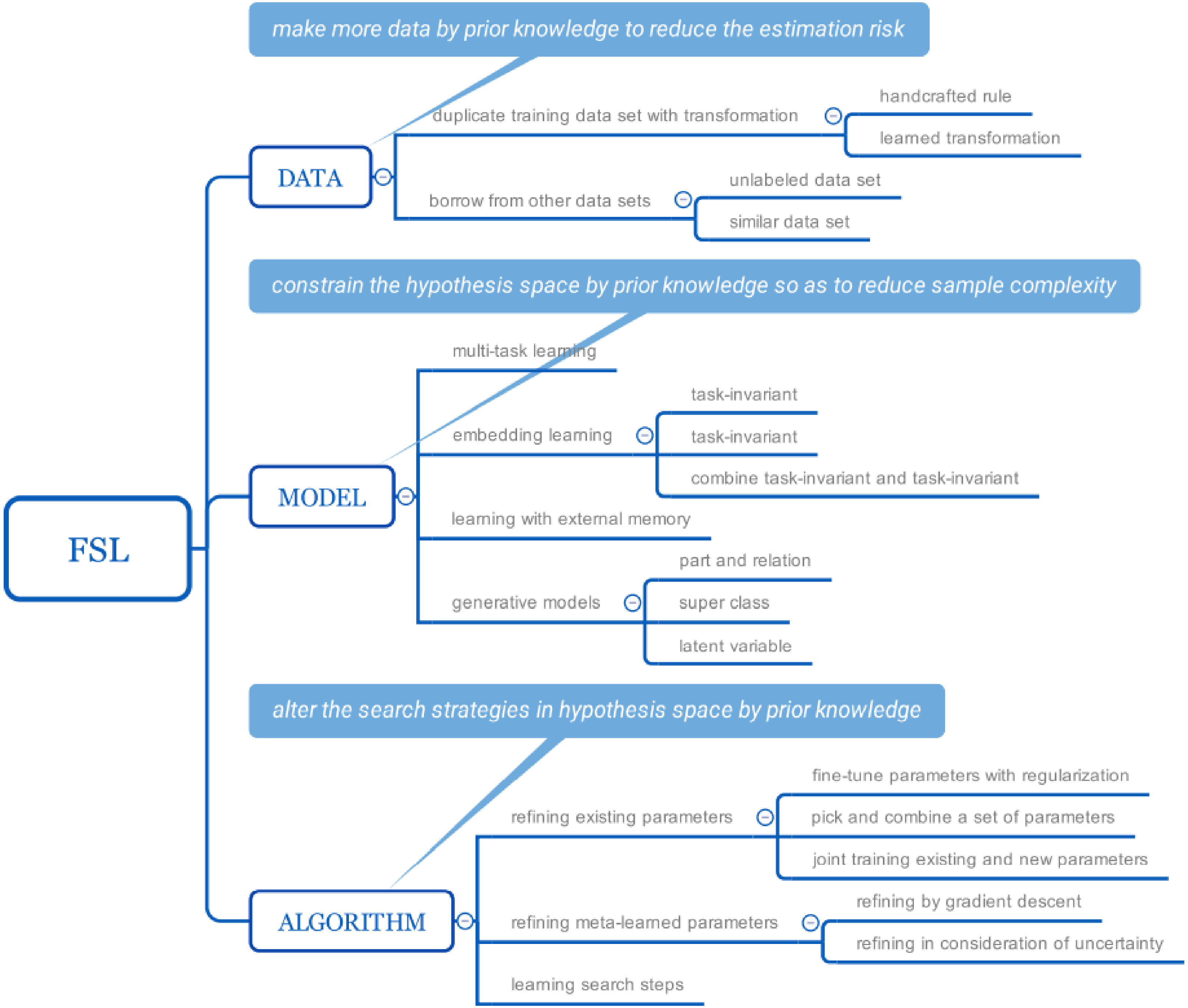
1. **Data:** use ***prior knowledge*** to augment D^{train} so as to provide an accurate $R_I(h)$ of smaller variance and to meet the sample complexity.
2. **Model:** design H based on ***prior knowledge*** in experience E to constrain the complexity of H and reduce its sample complexity.
3. **Algorithm:** take advantage of ***prior knowledge*** to search for the θ which parameterizes the best $h \in H$.

8. FSL Taxonomy



(a) Data. (b) Model. (c) Algorithm.

Fig. 3. How FSL methods solve few-shot problem from the perspectives from data (left), model (middle) and algorithm (right).


make more data by prior knowledge to reduce the estimation risk
DATA
duplicate training data set with transformation
handcrafted rule
learned transformation
borrow from other data sets
unlabeled data set
similar data set
constrain the hypothesis space by prior knowledge so as to reduce sample complexity
MODEL
multi-task learning
embedding learning
task-invariant
task-invariant
combine task-invariant and task-invariant
learning with external memory
generative models
part and relation
super class
latent variable
FSL
alter the search strategies in hypothesis space by prior knowledge
ALGORITHM
refining existing parameters
fine-tune parameters with regularization
pick and combine a set of parameters
joint training existing and new parameters
refining meta-learned parameters
refining by gradient descent
refining in consideration of uncertainty
learning search steps

9. Algorithm

1. **refining existing parameters θ:** an initial θ learned from other tasks
2. **refining meta-learned θ:** a *meta-learner* is learned from a set of tasks drawn from the same task distribution as the few-shot task to output a general θ
3. **learning search steps**

strategy	prior knowledge	how to search θ in $\mathcal{H}$
refining existing parameters θ^0	learned θ^0 as initialization	refine θ^0 by D^{train}
refining meta-learned θ	meta-learner learned from a task distribution	refine the meta-learned θ by D^{train}
learning search steps	meta-learner learned from a task distribution	search steps provided by meta-learner

10. Few-Shot Learning VS Meta-Learning

1. **Few-Shot Learning:** tackle the problem that lacks the ability of learning from limited exemplars and fast generalizing to new tasks.
2. **Meta-Learning:** learning-to-learn, the goal of the trained model is to quickly *learn a new task from a small amount of new data*, and **the model is trained by the meta-learner to be able to learn on a large number of different tasks.**

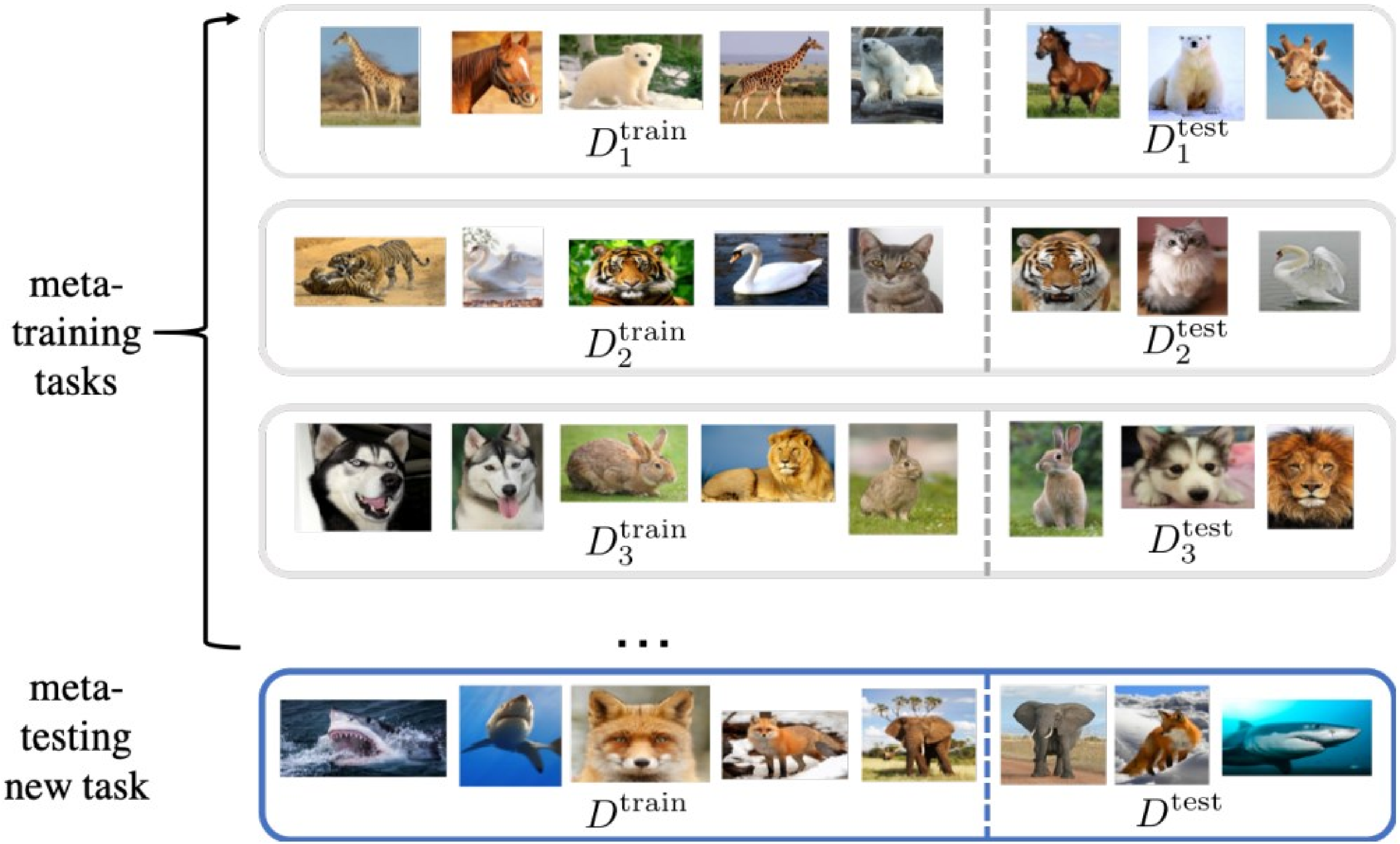
11. Meta-learning

- In meta-learning, it learns from **a set of tasks $\boldsymbol{T_s \sim p(T)}$.** Each task Ts operates on data set D_{T_s} of N classes where $D_{T_s} = \{D_{T_s}^{train}, D_{T_s}^{test}\}$. Each learner learns from $D_{T_s}^{train}$ and measures test error on $D_{T_s}^{test}$. The parameter of meta-learner learns to minimize the error across all learners by

$$\theta = arg \min_{\theta} E_{T_s \sim p(T)}\, l_{\theta}(D_{T_s})$$

- Then in meta-testing, **another disjoint set of tasks $\boldsymbol{T_t \sim p(T)}$** is used to test the generalization ability of meta-learner. Each T_t works on D_{T_t}. Finally, learner learns from $D_{T_t}^{train}$ and test on $D_{T_t}^{test}$ to obtain the meta-learning testing error.

11. Meta-learning



12. MAML

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3:     Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4:     **for all** $\mathcal{T}_i$ **do**
5:         Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6:         Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7:         Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8:         Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9:     **end for**
10:     Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

谢谢