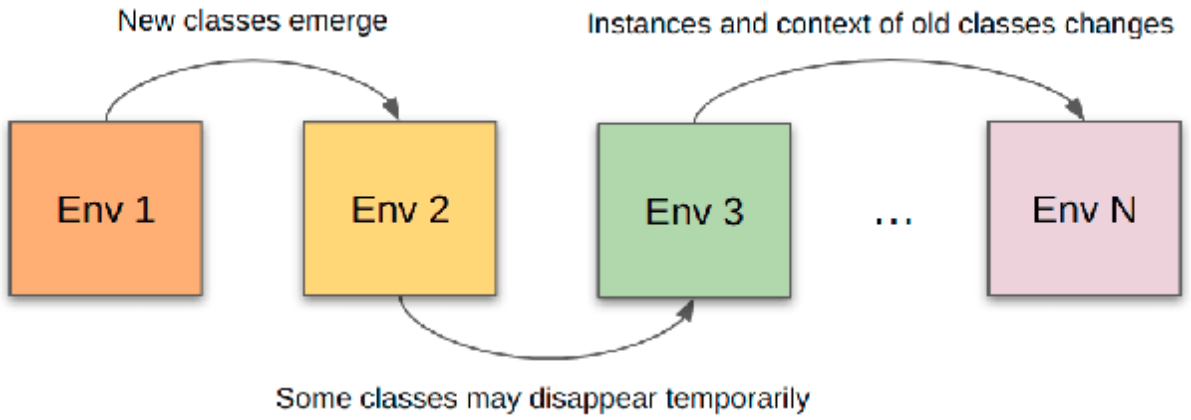
PAC-Bayes bounds for cumulative loss in Continual Learning

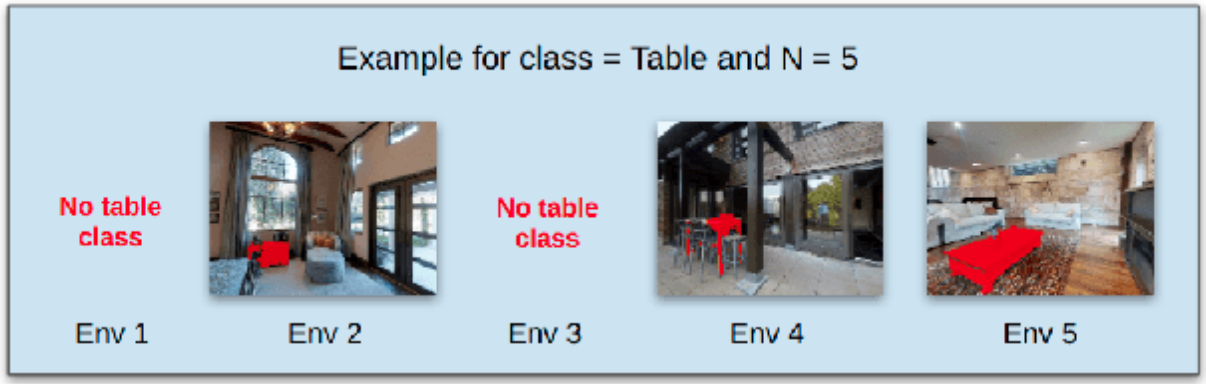
Lior Friedman, Prof. Ron Meir

Continual Learning

- Learn sequence of tasks, one by one
  - Limited memory for past data/models
  - E.g. learning new skills, continual deployment



- Trade-off: **learning plasticity** Vs. memory stability
  - Easy adaptation to new data => hard to remember old tasks

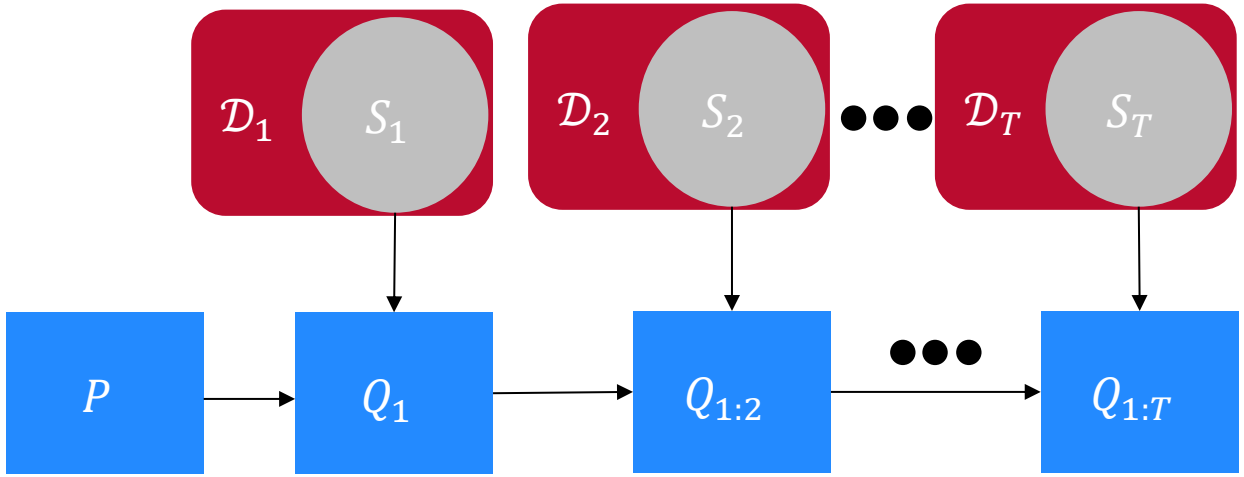


Existing results

- Many focus on forgetting (usually on training data)
  - Usually specific problems/methods – e.g. linear regression w/ SGD optimizer
  - Useful for specific behavior and insights
  - 2025 work - PAC-Bayes bounds on forgetting

- Haddouche and Guedj 2022: PAC-Bayes bounds for online learning
  - Close to what we want
  - Pessimistic: each sample can be adversarial

PAC-Bayes continual learning



- Data-free prior P, T tasks, each task has dataset $S_i \sim \mathcal{D}_i^m$
  - Each task has i.i.d. data
  - Tasks may be related (positive or negative)
  - Cannot access all historic data
- Learn posterior distributions $Q_{1:i}$ in a sequence

PAC-Bayes continual learning – cumulative loss

- $CuL\left((Q_{1:i})_{i=1}^{T}, (\mathcal{D}_i)_{i=1}^{T}\right) = \sum_{i=1}^{T} \mathcal{L}(Q_{1:i}, \mathcal{D}_i | \mathcal{F}_{i-1})$
- $\frac{1}{T} CuL\left((Q_{1:i})_{i=1}^{T}\right) \leq \frac{1}{T} \sum_{i=1}^{T} \left[\hat{\mathcal{L}}(Q_{1:i}, S_i) + \frac{1}{\lambda} D_{\mathrm{KL}}(Q_{1:i} || Q_{1:i-1})\right] + C(\lambda, m, T)$
- Some insights:
  - $m \gg 1$ implies easy problem
  - If $D_{\mathrm{KL}}(Q_{1:i} || Q_{1:i-1})$ small, similar models -> less data
    - Model capacity can help
    - Task relatedness can help/hinder
  - Can get bounds that converge as $T \to \infty$

PAC-Bayes continual learning – cumulative loss

- Oracle bounds for asymptotic case $m, T \to \infty$
  - $h^*_{1:i-1} = \min_{h \in \mathcal{H}} \sum_{t=1}^{i-1} \mathcal{L}(h, \mathcal{D}_t)$
  - $\lim_{m,T\to\infty} \frac{1}{T} CuL\left((Q_{1:i})_{i=1}^{T}\right) \leq \lim_{T\to\infty} \frac{1}{T} \sum_{i=2}^{T} \min\left(\mathcal{L}(h^*_{1:i-1}, \mathcal{D}_i), \mathcal{L}(h^*_{i-1}, \mathcal{D}_i)\right)$
- Some insights:
  - Switching tasks: if **enough capacity**, easy, otherwise depends on **task relatedness**
  - Slow change rate: proportional to rate r – task order **matters** here