From IID to Independent Mechanisms in Continual Learning

Oleksiy Ostapenko[1,2,3], Pau Rodríguez López[3], Alexandre Lacoste[3], Laurent Charlin[1,4]

1-Mila, 2-University of Montreal, 3 - ServiceNow AI, 4 - HEC Montreal

Continual Learning = learning from non-iid stream of (locally iid) tasks

Continual Learning = learning from non-iid stream of (locally iid) tasks

- **CL emerged as a problem in ML before tools from causality became popular**
  → we used techniques that were available back then, i.e. replay that simply simulates iid

Continual Learning = learning from non-iid stream of (locally iid) tasks

- **CL emerged as a problem in ML before tools from causality became popular**
  → we used techniques that were available back then, i.e. replay that simply simulates iid

- **Causality has emerged as an independent field in ML that gives us some new tools**
  → how can these tools help ML to go beyond the iid assumption

Continual Learning = learning from non-iid stream of (locally iid) tasks

Desiderata:

(1) Knowledge retention (Catastrophic Forgetting)

(2) Forward Transfer

(3) Backward Transfer

(4) Automatic task inference?

…

Continual Learning = learning from non-iid stream of (locally iid) tasks

Desiderata:

(1) Knowledge retention (Catastrophic Forgetting)

(2) **Forward Transfer**

(3) **Backward Transfer**

(4) Automatic task inference?

…

We already assume that knowledge is shared across distributions/tasks

SCM & Independent Mechanisms (IM) assumption[1]

SCM & Independent Mechanisms (IM) assumption[1]

The training data is sampled from the joint: $P(Y, X, K) = P(Y|X, K)P(K)P(X)$, that is induced by **e.g.**:

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

SCM & Independent Mechanisms (IM) assumption[1]

The training data is sampled from the joint: $P(Y, X, K) = P(Y|X, K)P(K)P(X)$, that is induced by **e.g.**:

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= X_2 - X_1, \\
M_5 &= 2X_1 + cos(X_2) \\
&\dots \\
M_N &= X_1 * X_1
\end{aligned}
$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward

2: traditionally, in ML the causal direction is Y→ X

SCM & Independent Mechanisms (IM) assumption[1]

The training data is sampled from the joint: $P(Y, X, K) = P(Y|X, K)P(K)P(X)$, that is induced by **e.g.**:

$$X_1 \sim U(-1, 1)$$
$$X_2 \sim U(-1, 1)$$
$$K \in \{1, \ldots, N\}$$

Task descriptor

$$\begin{aligned} M_1 &= X_1 + X_2, \\ M_2 &= X_1 * X_2, \\ M_3 &= X_1 - X_2, \\ M_4 &= X_2 - X_1, \\ M_5 &= 2X_1 + cos(X_2) \\ &\ldots \\ M_N &= X_1 * X_1 \end{aligned}$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

SCM & Independent Mechanisms (IM) assumption[1]

The training data is sampled from the joint: $P(Y, X, K) = P(Y|X, K)P(K)P(X)$, that is induced by **e.g.**:

$$X_1 \sim U(-1, 1)$$
$$X_2 \sim U(-1, 1)$$
$$K \in \{1, \ldots, N\}$$

Task descriptor

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= X_2 - X_1, \\
M_5 &= 2X_1 + cos(X_2) \\
&\ldots \\
M_N &= X_1 * X_1
\end{aligned}
$$

$$Y = \sum_N 1_{\{n=K\}} M_n$$

Corresponds to SCM: $\mathbb{M} = \langle \mathbf{Y} = \{Y, M_1, \ldots M_N, X_1, X_2, X_3\}, \mathbf{U} = \{U_1, U_2, U_3\}, \mathcal{M}, P(U_1, U_2, U_3) \rangle$

1: it’s still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

SCM & Independent Mechanisms (IM) assumption[1]

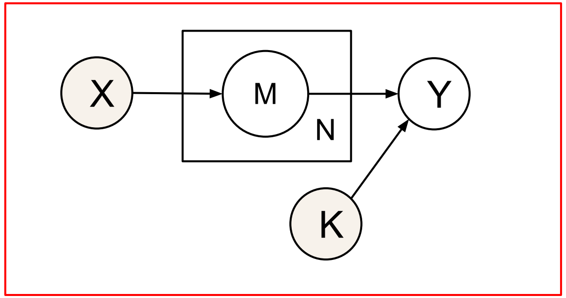
The training data is sampled from the joint: $P(Y, X, K) = P(Y|X, K)P(K)P(X)$, that is induced by **e.g.**:

$$X_1 \sim U(-1,1)$$
$$X_2 \sim U(-1,1)$$
$$K \in \{1, \ldots, N\}$$

Task descriptor

$$
\begin{aligned}
M_1 &= X_1 + X_2,\\
M_2 &= X_1 * X_2,\\
M_3 &= X_1 - X_2,\\
M_4 &= X_2 - X_1,\\
M_5 &= 2X_1 + cos(X_2)\\
&\ldots\\
M_N &= X_1 * X_1
\end{aligned}
$$

$$Y = \sum_N 1_{\{n=K\}} M_n$$



Corresponds to SCM: $\mathbb{M} = \langle \mathbf{Y} = \{Y, M_1, \ldots M_N, X_1, X_2, X_3\}, \mathbf{U} = \{U_1, U_2, U_3\}, \mathcal{M}, P(U_1, U_2, U_3) \rangle$

1: it’s still an assumption → there are problems where it doesn't hold, but it may bring us forward

2: traditionally, in ML the causal direction is Y→ X

Learning the mechanisms M_1 … M_N

To model

$$P(Y|X,K)$$

We learn

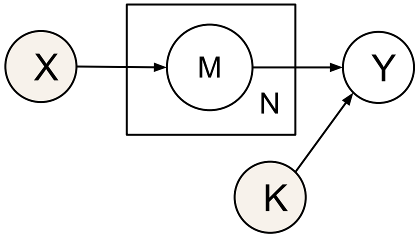
$$M_1 = X_1 + X_2,$$
$$M_2 = X_1 * X_2,$$
$$M_3 = X_1 - X_2,$$
$$M_4 = X_2 - X_1,$$
$$M_5 = 2X_1 + cos(X_2)$$
$$\dots$$
$$M_N = X_1 * X_1$$

Induce Y given task descriptor T

$$Y = \sum_N 1_{\{n=K\}} M_n$$

Learn **an expert per mechanism** → **Modularity**

Learning the mechanisms in CL

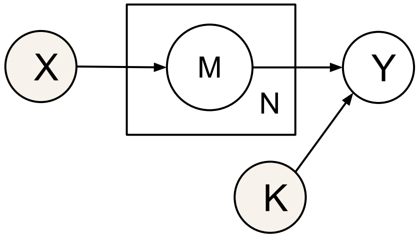


$$P(Y, X, K) = \underbrace{P(Y|X, K)}_{\text{Stationary}}\underbrace{P(X, K)}_{\text{Non-stationary}}{}^{*,**}$$

* for simplicity X,T are independent

** Yoshua Bengio's talk at CoLLAs https://www.youtube.com/watch?v=2G-__dde8SI&ab_channel=ConferenceonLifelongLearningAgents%28CoLLAs%29

Learning the mechanisms in CL



$$P(Y,X,K) = \underbrace{P(Y|X,K)}_{\text{Stationary}}\underbrace{P(X,K)}_{\text{Non-stationary}}{}^{*,**}$$

* for simplicity X,T are independent

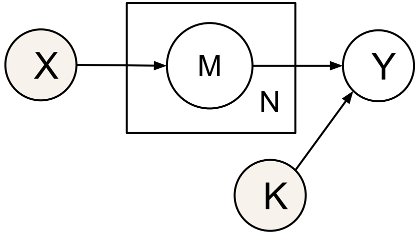
Distribution shifts:

(1) **Domain shift – shift in P(X), i.e. different values are assigned to input variables**

- **Modular system resilient** the better the learned modules can approximate the true mechanisms (i.e. no CF)

** Yoshua Bengio's talk at CoLLAs https://www.youtube.com/watch?v=2G-__dde8SI&ab_channel=ConferenceonLifelongLearningAgents%28CoLLAs%29

Learning the mechanisms in CL



$$P(Y,X,K) = \underbrace{P(Y|X,K)}_{\text{Stationary}}\underbrace{P(X,K)}_{\text{Non-stationary}}{}^{*}$$

* for simplicity X,T are independent

Distribution shifts:

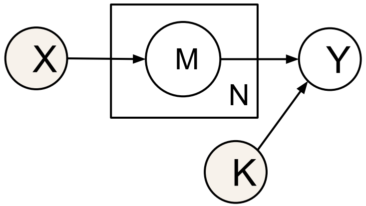
(1) **Domain shift – shift in P(X), i.e. different values are assigned to input variables**

- **Modular system resilient** the better the learned modules can approximate the true mechanisms (i.e. no CF)

(2) **New mechanism shift – shift in P(K)** → new mechanisms are introduced

- **Modular system resilient** given correct **routing** (i.e. deal with CF in routing mechanism)
- Modular system profit from **high transfer** (assuming mechanisms are shared)

Learning the mechanisms in CL



$$P(Y,X,K) = \sum_Z \underbrace{P(Y|X,K,Z)}_{\text{Stationary}} \underbrace{P(X,K,Z)}_{\text{Non-stationary}} \overset{*}{=} P(Y|X,K)P(X,K)$$

* for simplicity X,T and Z are independent

Distribution shifts:

(1) **Domain shift – shift in P(X), i.e. different values are assigned to input variables**

- **Modular system resilient** the better the learned modules can approximate the true mechanisms (i.e. no CF)

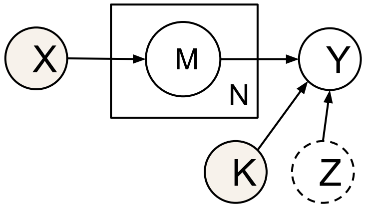
(2) **New task shift – shift in P(K)** → new mechanisms are introduced

- **Modular system resilient** given correct **routing** (i.e. deal with CF in routing mechanism)
- Modular system profit from **high transfer** (assuming mechanisms are shared)

(3) **Hidden shift – shift in P(Z)** → apparent shift in existing mechanism

- **Modular system resilient** e.g. if existing modules are not frozen → CF prevented solely through routing
- **Replay & Regularization underperform**

Learning the mechanisms in CL



$$P(Y,X,K)=\sum_Z \underbrace{P(Y|X,K,Z)}_{\text{Stationary}}\underbrace{P(X,K,Z)}_{\text{Non-stationary}}{}^{*}=P(Y|X,K)P(X,K)$$

* for simplicity X,T and Z are independent

Distribution shifts:

(1) **Domain shift – shift in P(X), i.e. different values are assigned to input variables**

- **Modular system resilient** the better the learned modules can approximate the true mechanisms (i.e. no CF)

(2) **New task shift – shift in P(K)** → new mechanisms are introduced

- **Modular system resilient** given correct **routing** (i.e. deal with CF in routing mechanism)
- Modular system profit from **high transfer** (assuming mechanisms are shared)

(3) **Hidden shift – shift in P(Z)** → apparent shift in existing mechanism

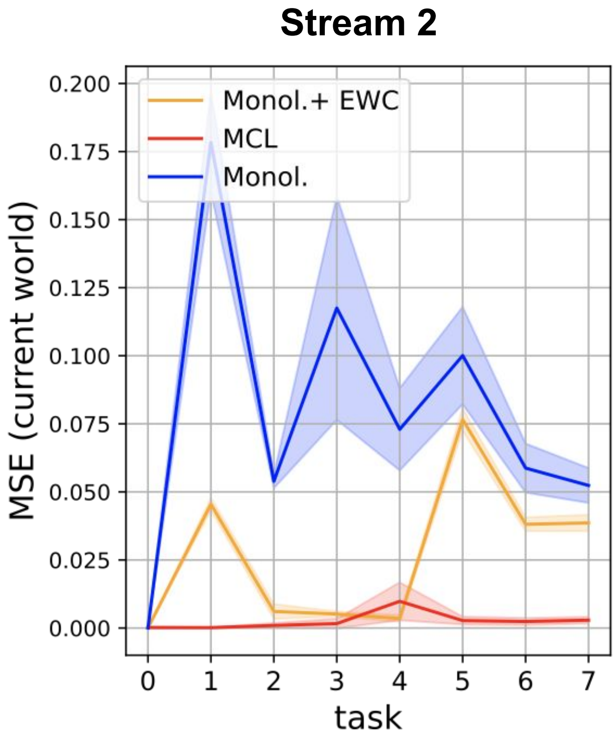
- **Modular system resilient** e.g. if existing modules are not frozen → CF prevented solely through routing
- **Replay & Regularization underperform**

(4) **Data amount shift (see Veniat et al., 2021)**

(5) **Spurious shift (see Lesort et al., 2022)**

Simple model with attention based routing (MoE)

Inspired by Neural Production Systems (Goyal et al., 2021) and LMC (Ostapenko et al., 202)



Stream 1 shift in V_2 (the operation):

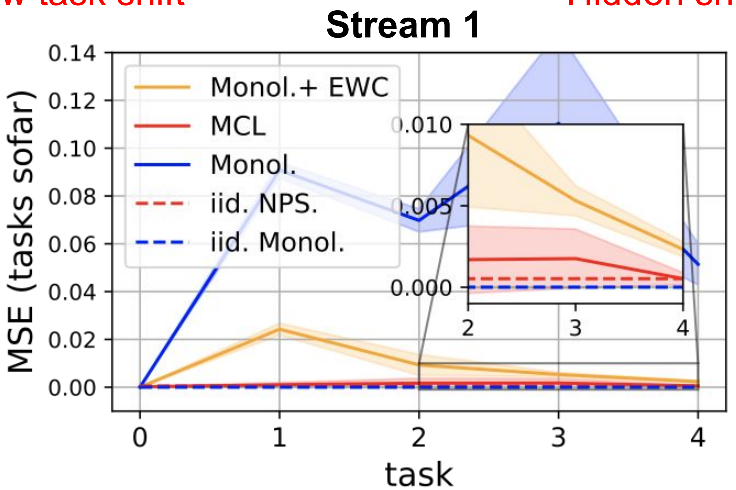
- T_0: operations $x_1 + x_2$ and $x_1 - x_2$;
- T_1: $x_1 + x_2$ and $(x_1 + x_2) * x_2$;
- T_2: $x_1 - x_2$ and $x_1 * x_2$, ;
- T_3: $x_1 + x_2$ and x_1^2 ;
- T_4: $x_1 + x_2$ and $x_1 * x_2$.

New task shift

Stream 2 shift in V_2 and V_4:

- T_0: operations $x_1 + x_2$ and $x_1 - x_2$;
- $T_1 - T_4$: see Stream 1 ;
- T_5: $(x_1 + x_2)/5$ and $(x_1 - x_2)/5$;
- T_6: $(x_1 - x_2)/5$ and $(x_1 * x_2)/5$;
- T_7: $(x_1 * x_2)/5$, and $(x_1 + x_2)/5$.

Hidden shift



(MoEs have many limitations)

Challenges

Challenges

(1) Compositionality – endogenous variables are used to generate other endogenous variables

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= X_2 - X_1, \\
M_5 &= 2X_1 + cos(X_2) \\
&\dots \\
M_n &= X_1 * X_1
\end{aligned}
$$

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= M_2 - M_1, \\
M_5 &= M_3 + X_2 \\
&\dots \\
M_n &= X_1 * M_1
\end{aligned}
$$

Challenges

(1) Compositionality – endogenous variables are used to generate other endogenous variables

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= X_2 - X_1, \\
M_5 &= 2X_1 + cos(X_2) \\
&\dots \\
M_n &= X_1 * X_1
\end{aligned}
$$

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= M_2 - M_1, \\
M_5 &= M_3 + X_2 \\
&\dots \\
M_n &= X_1 * M_1
\end{aligned}
$$

Challenges

(1) Compositionality – endogenous variables are used to generate other endogenous variables

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= X_2 - X_1, \\
M_5 &= 2X_1 + cos(X_2) \\
&\dots \\
M_n &= X_1 * X_1
\end{aligned}
$$

Compositional rules

Primitive rules

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2, \\
M_4 &= M_2 - M_1, \\
M_5 &= M_3 + X_2 \\
&\dots \\
M_n &= X_1 * M_1
\end{aligned}
$$

Challenges

(1) Compositionality – endogenous variables are used to generate other endogenous variables

Advantage:

- More transfer + can cover more problems

Challenges:

- How to decompose tasks into reusable rules?
  - (a) Curriculum from primitive to compositional rules? (Elis et al., 2020)
  - (b) Using information bottlenecks like attention etc.?
  - (c) Causal Discovery

Primitive rules

$$
\begin{aligned}
M_1 &= X_1 + X_2, \\
M_2 &= X_1 * X_2, \\
M_3 &= X_1 - X_2,
\end{aligned}
$$

Compositional rules

$$
\begin{aligned}
M_4 &= M_2 - M_1, \\
M_5 &= M_3 + X_2 \\
&\ldots \\
M_n &= X_1 * M_1
\end{aligned}
$$

Challenges

(1) Compositionality – endogenous variables are used to generate other endogenous variables

Advantage:

- More transfer + can cover more problems

Challenges:

- How to decompose tasks into reusable rules?
  - (a) Curriculum from primitive to compositional rules? (Elis et al., 2020)
  - (b) Using information bottlenecks like attention etc.?
  - (c) Causal Discovery

Primitive rules

$$M_1 = X_1 + X_2, \quad M_2 = X_1 * X_2, \quad M_3 = X_1 - X_2,$$

Compositional rules

$$M_4 = M_2 - M_1, \quad M_5 = M_3 + X_2, \quad \ldots, \quad M_n = X_1 * M_1$$

(2) Complex task description as part of the input/context

(1) Task identity should be part of the input variables, and can be hard to infer → routing information

Conclusion

(1) **IM entails separation into stationary and non-stationary components of DGP**

→ Distribution shifts are only caused by the non-stationarity in the inputs to the mechanisms

Conclusion

(1) **IM entails separation into stationary and non-stationary components of DGP**

→ Distribution shifts are only caused by the non-stationarity in the inputs to the mechanisms

(2) **IM entails modular solutions → functional and structural learning**

→ Catastrophic forgetting is prevented by routing (akin to reasoning)

Conclusion

(1) **IM entails separation into stationary and non-stationary components of DGP**

→ Distribution shifts are only caused by the non-stationarity in the inputs to the mechanisms

(2) **IM entails modular solutions → functional and structural learning**

→ Catastrophic forgetting is prevented by routing (akin to reasoning)

(3) **Modular solutions can address some distribution shifts better then monolithic**

Conclusion

(1) **IM entails separation into stationary and non-stationary components of DGP**

→ Distribution shifts are only caused by the non-stationarity in the inputs to the mechanisms

(2) **IM entails modular solutions → functional and structural learning**

→ Catastrophic forgetting is prevented by routing (akin to reasoning)

(3) **Modular solutions can address some distribution shifts better then monolithic**

(4) A lot of open questions & challenges

→ Compositionality

→ Moving to more realistic domains (*computer vision is probably not the best domain*)

Sources

Lesort, T. 2022, Continual Feature Selection: Spurious Features in Continual Learning. arXiv preprint arXiv:2203.01012.

Ke, Nan Rosemary, et al. "Systematic evaluation of causal discovery in visual model based reinforcement learning." arXiv preprint arXiv:2107.00848 (2021).

Ellis, Kevin, et al. "Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning." arXiv preprint arXiv:2006.08381 (2020).

Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf. Elements of causal inference: foundations and learning algorithms. The MIT Press, 2017.

Bareinboim, Elias, et al. "On pearl's hierarchy and the foundations of causal inference." Probabilistic and Causal Inference: The Works of Judea Pearl. 2022. 507-556.

ALIAS PARTH GOYAL, Anirudh Goyal, et al. "Neural production systems." Advances in Neural Information Processing Systems 34 (2021): 25673-25687.

Ostapenko, Oleksiy, et al. "Continual learning via local module composition." Advances in Neural Information Processing Systems 34 (2021): 30298-30312.

Veniat, Tom, Ludovic Denoyer, and Marc'Aurelio Ranzato. "Efficient continual learning with modular networks and task-driven priors." *arXiv preprint arXiv:2012.12631* (2020).

....

Appendix

Conclusion

1. **IM entails separation into stationary and non-stationary components of DGP**

   → Distribution shifts are only caused by the non-stationarity in the inputs to the mechanisms

2. **IM is a useful inductive bias**

   → IM entails modular solution, requires routing & functional learning

   → Routing can be performed using task descriptors/context

3. **Preliminary Experiments:** we show the advantages of IM guided CL on math equations with MoE system

   → Come to our poster for more details

4. **Compositionality** is the biggest challenge

Existing CL methods on different shifts

1. **New task shift**

   a. Observed FoVs shift — $p(X, Z)$ shifts due to shift in the marginals $p(X_k)$ for $k \subseteq \{1 \ldots N_x\}$

   - **Existing CL methods should perform well**
   - **Modular solutions can have better OOD generalization properties**

   b. Hidden FoVs shift — $p(X, Z)$ shifts due to shift in $p(X_k)$ and $p(Z_s)$ for $k \subseteq \{1 \ldots N_x\}$, $s \subseteq \{1 \ldots N_z\}$. [E.g. "+" in environment $E = 1$ is $x_1 + x_2$ but in $E = 2$ its $(x_1 + x_2)/10$, $E \in Z$]

   - **P(Y|X,Z) changes, without context Z being observed -> contradictory knowledge**
   - **Requires sparsely updating existing knowledge**
   - **Standard CL methods (vanilla) are likely to underperform**
   - **Modular solutions should be better in this**

2. **Data amount shift**

   - more training data for a previously learned tasks is now available
   - **Requires sparsely updating knowledge**
   - **Regularization based methods are likely to underperform**

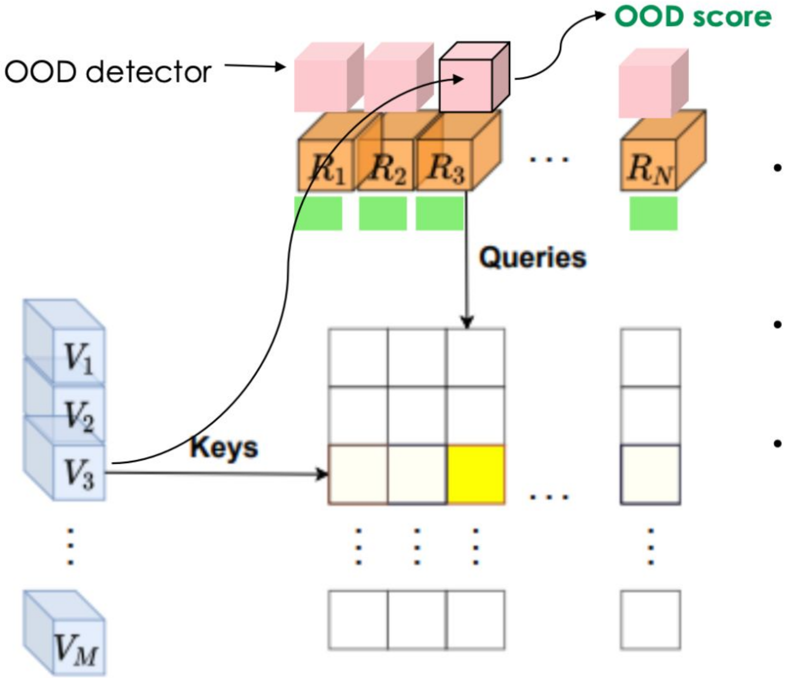
3. **Spurious correlation drift**

   - attributes correlate in e.g. task t=1, but not in later tasks

	E=1	E=2	E=3	E=4
t=1	+	-		
t=2		+-	+	-
...				

   - **Requires updating the routing parameters in modular solutions**
   - **Regularization based likely to underperform**

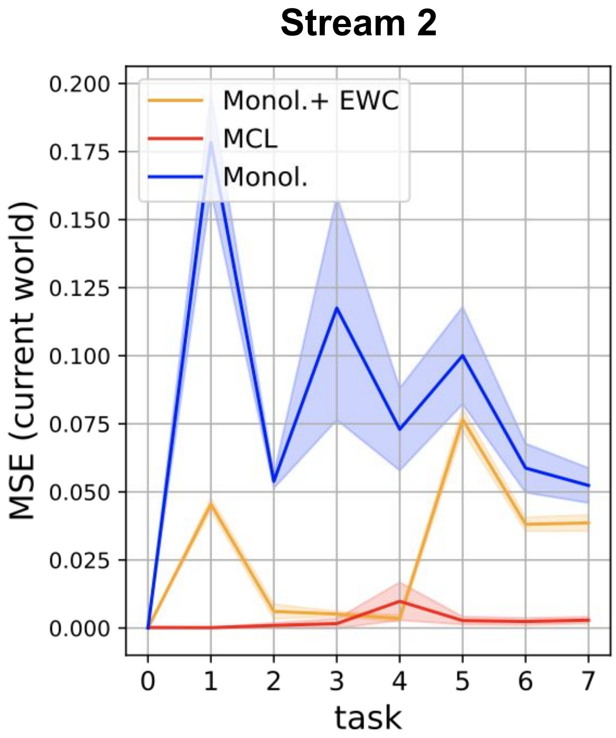
“Causal” MoE



- Module selection is now guided by both attention score and the OOD score
  - A sample is OOD if z-score > threshold (e.g. 3)
- At task switch --- whenever all existing modules signal OOD, we add a lot of modules (> then actually needed)
- Unused modules (which are not activated for some number of steps) are pruned

Simple model with attention based routing (MoE)

Inspired by Neural Production Systems (Goyal et al., 2021) and LMC (Ostapenko et al., 202)



Stream 1 shift in V_2 (the operation):

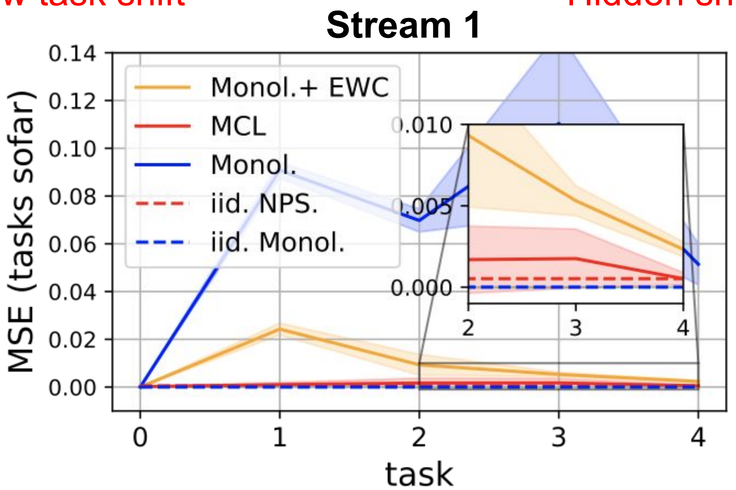
- T_0: operations $x_1 + x_2$ and $x_1 - x_2$;
- T_1: $x_1 + x_2$ and $(x_1 + x_2) * x_2$;
- T_2: $x_1 - x_2$ and $x_1 * x_2$, ;
- T_3: $x_1 + x_2$ and x_1^2 ;
- T_4: $x_1 + x_2$ and $x_1 * x_2$.

New task shift

Stream 2 shift in V_2 and V_4:

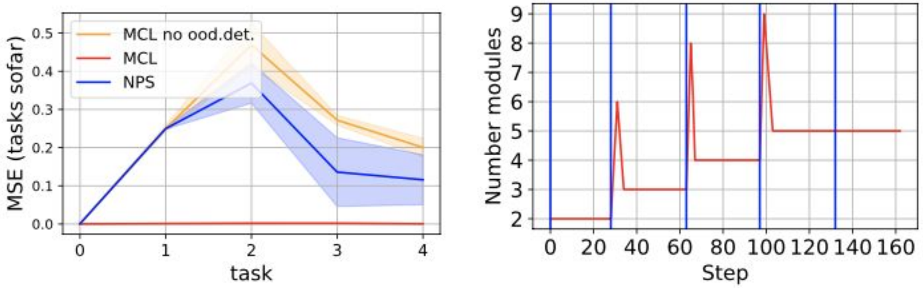
- T_0: operations $x_1 + x_2$ and $x_1 - x_2$;
- $T_1 - T_4$: see Stream 1 ;
- T_5: $(x_1 + x_2)/5$ and $(x_1 - x_2)/5$;
- T_6: $(x_1 - x_2)/5$ and $(x_1 * x_2)/5$;
- T_7: $(x_1 * x_2)/5$, and $(x_1 + x_2)/5$.

Hidden shift



(MoEs have many limitations)

Module activation and addition pattern



(a) Ablation (Stream 1) (b) Module addition MCL

Figure 4. (a) Ablation of MLC against NPS and a version of MCL without fixing structural parameter and automatically adding 1 module per task (Stream 1); (b) Module addition pattern of MCL on Stream 1, blue vertical lines represent task switches. MCL is able to successfully modules prune superfluous after each task. Since last task does not introduce any new mechanisms, no module addition is triggered.

IM can be a useful inductive bias for CL

(1) **IM entails modular solutions**

(a) New modules can be added without affecting the other ones → **no catastrophic forgetting**

→ see experiments Stream 1&2 (Fig. 1)

(b) *Learned* modules can be changed without affecting the other ones

→ effective for *hidden shift*, see experiments Stream 2 (Fig.1b)

(2) **More transfer the closer the learned mechanisms are to the true mechanisms**

(a) Mechanisms independent from inputs → resilient to domain shift

(b) Mechanisms are reused across tasks → transfer due to systematic generalization

(3) **Modular systems can address some shifts better than monolithic (+ replay/regularization)**

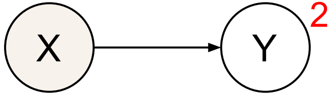
(a) E.g. hidden shift – existing CL methods are likely to underperform → see experiments Stream 2

(b) Data amount (task repetition) & spurious drift – regularization based are likely to underperform

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint: $P(Y,X) = P(Y|X)P(X)$



[2]

The goal is to model the "mechanism" $p(Y|X)$, suppose that this conditional is entailed by an SCM:

$$\mathbb{M} = \langle \underbrace{\mathbf{Y} = \{Y, Z_1, \ldots Z_n, X_1, X_2, X_3\}}_{\text{endogenous}}, \underbrace{\mathbf{U} = \{U_1, U_2, U_3\}}_{\text{exogenous}}, \mathcal{F}, P(U_1, U_2, U_3) \rangle$$

Hence, ideally we want to learn the mechanisms $Z_1 \ldots Z_n$.

$$\mathcal{F} = \begin{cases} X_1 = f(U_1),\ X_2 = f(U_2) \\ X_3 = f(U_3) \in \{1...N\} \\ \\ Z_1 = X_1 + X_2,\ Z_2 = X_1 * X_2, \\ Z_3 = X_1 - X_2,\ Z_4 = X_2 - X_1, \\ Z_5 = 2X_1 + cos(X_2) \\ \ldots \\ Z_n = X_1 * X_1 \\ Y = f(\mathbf{Z}, X_3) = \sum_N \mathbf{1}_{\{n = X_3\}} Z_n \end{cases}$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

[replaced with next slide] Independent Mechanisms (IM) assumption

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint:

$$P(Y, X) = P(Y|X)P(X)$$

The goal is to model the "mechanism" $p(Y|X)$, Suppose that this conditional is entailed by an SCM:

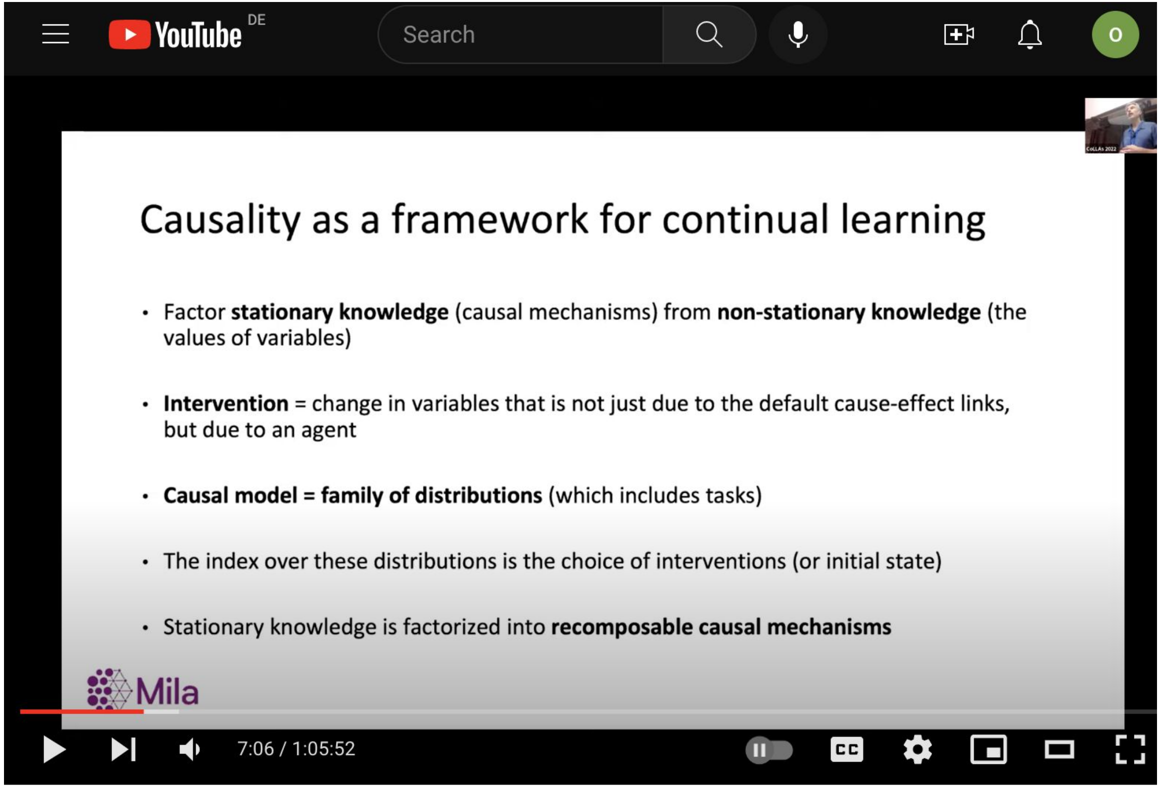
$$\mathbb{M} = \langle \mathbf{Y} = \{Y_1, \dots Y_n\}, \mathbf{U} = \{X_1, X_2\}, \mathcal{F} = \{f_1, \dots f_n\}, P(X_1, X_2)\rangle$$

$$\mathcal{F} = \begin{cases} Y_1 = X_1 + X_2 \\ Y_2 = X_1 * X_2 \\ Y_3 = X_1 - X_2 \\ Y_4 = X_2 - X_1 \\ Y_5 = 2X_1 + cos(X_2) \\ \dots \\ Y_n = X_1 * X_1 \end{cases}$$

Hence, ideally we want to learn the mechanisms $\mathcal{F}$.

X⟶Y vs. Y⟶X in ML

Causal model = family of distributions indexed by interventions



Keynote - CoLLAs 2022 - Private

Modular & Causal Knowledge Representation for Lifelong Learning - Yoshua Bengio - CoLLAs 2022

Moving forward

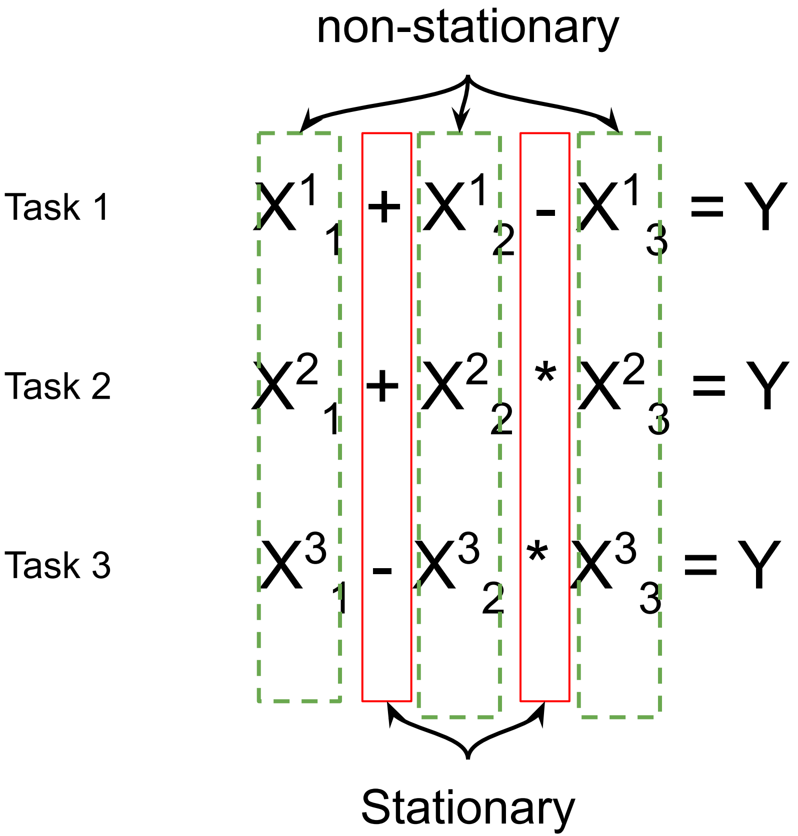
(3) Moving to more realistic domains

(a) Model based RL (e.g. Ke et al., 2022)
(b) Representation Learning vs. cognitive tasks & reasoning?

(4) What is the role of scaling?

(The Challenge of Compositionality for AI)

Causality: separate stationary from non-stationary



Independent Mechanism (IM) assumption

Stationary knowledge is factorized into recomposable causal modules (Yoshua Bengio, CoLLAs 2022)
Assumption (not all systems may satisfy it)!

$P(Y|X,T,Z)$ factorizes into independent causal mechanisms

One way to show this: let this distribution be entailed by SCM:

$$X := N_x$$

$$Z := N_z$$

$$Y := N_T$$

$$Y := f(X,T,Z)$$

Continual Learning = learning from non-iid stream of (locally iid) tasks

- **CL emerged as a problem in ML before causality**

  → people used techniques that were available, i.e. replay that simply simulates iid

- **Causality has emerged as an independent field in ML that gives us some new tools**

  → how can these tools help CL to go beyond the iid assumption

We need to replace the IID assumption with another realistic and useful assumption/inductive bias

Continual Learning = learning from locally iid tasks

Desiderata:

(1) **Knowledge retention (Catastrophic Forgetting)**

(2) **Forward Transfer**

(3) **Backward Transfer**

(4) **Automatic task inference?**

(5) **Systematic generalization?**

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., 2017).

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., 2017).

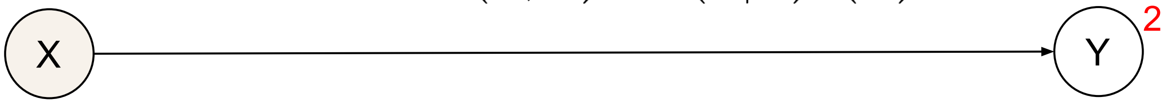
The training data is sampled from the joint: $P(Y, X) = P(Y|X)P(X)$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., 2017).

The training data is sampled from the joint: $P(Y, X) = P(Y|X)P(X)$, that is induced by **e.g.**:



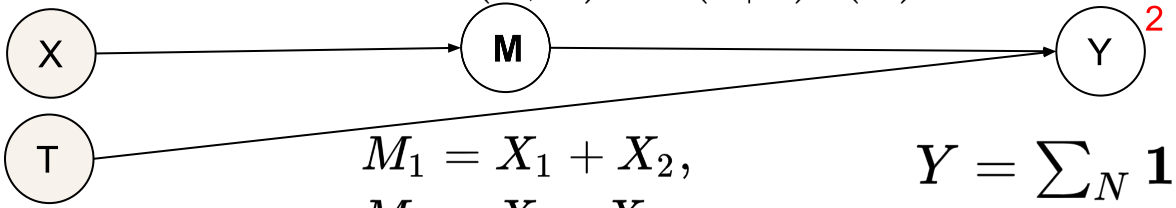
1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward

2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint: $P(Y, X) = P(Y|X)P(X)$, that is induced by **e.g.**:



$$Y = \sum_N \mathbf{1}_{\{n=T\}} M_n$$

$$X_1 \sim U(-1, 1)$$

$$X_2 \sim U(-1, 1)$$

$$T \in \{0, \ldots, N\}$$

Task descriptor

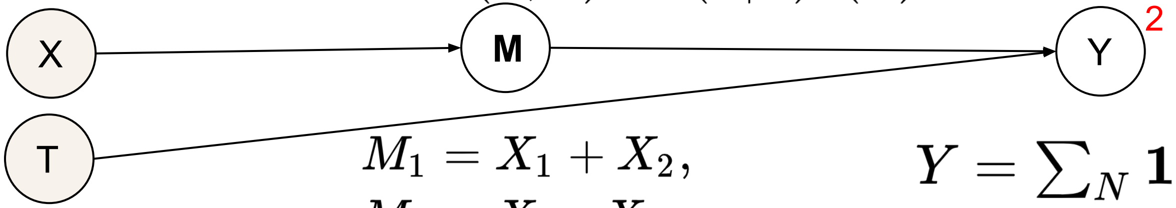
$$M_1 = X_1 + X_2,$$
$$M_2 = X_1 * X_2,$$
$$M_3 = X_1 - X_2,$$
$$M_4 = X_2 - X_1,$$
$$M_5 = 2X_1 + cos(X_2)$$
$$\ldots$$
$$M_n = X_1 * X_1$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint: $P(Y, X) = P(Y|X)P(X)$, that is induced by **e.g.**:



$$X_1 \sim U(-1, 1)$$

$$X_2 \sim U(-1, 1)$$

$$T \in \{0, \ldots, N\}$$

Task descriptor

$$M_1 = X_1 + X_2,$$
$$M_2 = X_1 * X_2,$$
$$M_3 = X_1 - X_2,$$
$$M_4 = X_2 - X_1,$$
$$M_5 = 2X_1 + cos(X_2)$$
$$\ldots$$
$$M_n = X_1 * X_1$$

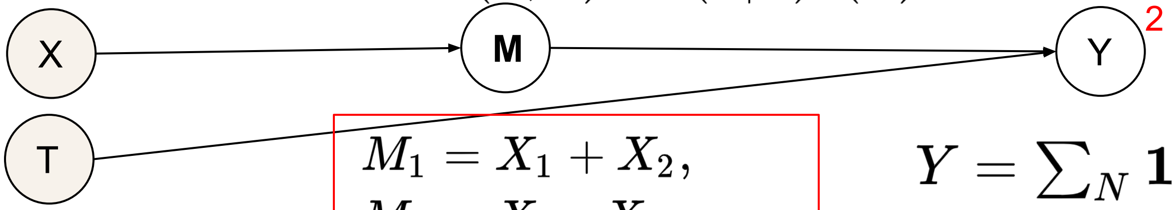
$$Y = \sum_N \mathbf{1}_{\{n=T\}} M_n$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint: $P(Y, X) = P(Y|X)P(X)$, that is induced by **e.g.**:



$$Y = \sum_N \mathbf{1}_{\{n=T\}} M_n$$

$$X_1 \sim U(-1, 1)$$

$$X_2 \sim U(-1, 1)$$

$$T \in \{0, \dots, N\}$$

Task descriptor

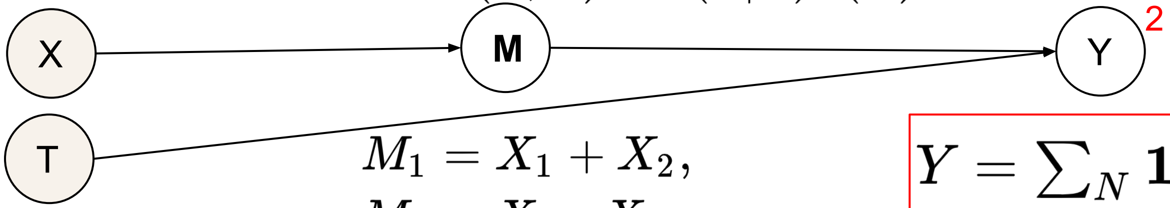
$$\begin{aligned} M_1 &= X_1 + X_2, \\ M_2 &= X_1 * X_2, \\ M_3 &= X_1 - X_2, \\ M_4 &= X_2 - X_1, \\ M_5 &= 2X_1 + cos(X_2) \\ &\dots \\ M_n &= X_1 * X_1 \end{aligned}$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint: $P(Y,X) = P(Y|X)P(X)$, that is induced by **e.g.**:



$$Y = \sum_N \mathbf{1}_{\{n=T\}} M_n$$

$$X_1 \sim U(-1,1)$$

$$X_2 \sim U(-1,1)$$

$$T \in \{0,\ldots,N\}$$

Task descriptor

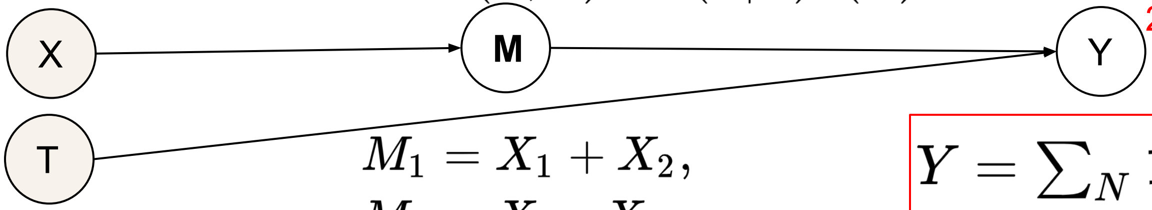
$$\begin{aligned} M_1 &= X_1 + X_2, \\ M_2 &= X_1 * X_2, \\ M_3 &= X_1 - X_2, \\ M_4 &= X_2 - X_1, \\ M_5 &= 2X_1 + cos(X_2) \\ &\ldots \\ M_n &= X_1 * X_1 \end{aligned}$$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanisms (IM) assumption[1]

Causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of Causal Inference).

The training data is sampled from the joint: $P(Y, X) = P(Y|X)P(X)$, that is induced by **e.g.**:



[2]

$$Y = \sum_N \mathbf{1}_{\{n=T\}} M_n$$

$$X_1 \sim U(-1,1)$$
$$X_2 \sim U(-1,1)$$
$$T \in \{0,\ldots,N\}$$

Task descriptor

$$\begin{aligned}
M_1 &= X_1 + X_2,\\
M_2 &= X_1 * X_2,\\
M_3 &= X_1 - X_2,\\
M_4 &= X_2 - X_1,\\
M_5 &= 2X_1 + cos(X_2)\\
&\ldots\\
M_n &= X_1 * X_1
\end{aligned}$$

Corresponds to SCM: $\mathbb{M} = \langle \mathbf{Y} = \{Y, M_1, \ldots M_n, X_1, X_2, X_3\}, \mathbf{U} = \{U_1, U_2, U_3\}, \mathcal{F}, P(U_1, U_2, U_3) \rangle$

1: it's still an assumption → there are problems where it doesn't hold, but it may bring us forward
2: traditionally, in ML the causal direction is Y→ X

Independent Mechanism (IM) assumption:

Stationary knowledge is factorized into recomposable causal modules (Yoshua Bengio, CoLLAs 2022)
Assumption (not all systems may satisfy it)!

Another ways to frame it:

- *the causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other (Peters et al., Elements of CAusal Inference).*
- *The principle is plausible if we conceive our system as being composed of modules comprising (sets of) variables such that the modules represent physically independent mechanisms of the world (Peters et al., Elements of CAusal Inference).*