A List of Notation

Symbol	**Type**	**Explanation**
$\perp$		undefined
$[\![\text{bool}]\!]$	$\in\{0,1\}$	=1 if bool=True, =0 if bool=False
d	$\in\mathbb{N}$	number of states
k	$\in\mathbb{N}$	number of actions
i,j	$\in\mathbb{N}$	time index/step
$\{i:j\}$	$\subset\mathbb{Z}$	Set of integers from i to j (empty if $j<i$)
$s,s',...,s^i$	$\in\{1:d\}$	state at time step 1,2,...,i
$a,a',...,a^i$	$\in\{0:k-1\}$	action at time step 1,2,...,i
$b,b',...,b^i$	$\in\{0:k-1\}$	alternative action at time step 1,2,...,i
$a^{:i}$	$:=aa'...a^i$	sequence of i actions
$a^{<i}$	$:=aa'...a^{i-1}$	sequence of $i-1$ actions
$\dot s,\ddot s$	$\in\{1:\dot d\}$	parts of state, usually $s=(\dot s,\ddot s)$
ε	>0	small number >0
$p(...)$	$\in[0;1]$	(conditional) probability distribution over states and actions
$\pi(a\vert s)$	$\in[0;1]$	policy. Probability of action a in state s
M^a,W^a	$\in[0;1]^{d\times d}$	transition-policy tensor $M^a_{ss'}=p(s'\vert sa)\cdot\pi(a\vert s)$ for each action a, $W=q$
B^a	$\in[0;1]^{d\times d}$	inverse 1-step model $B^a_{ss'}=p(a\vert ss')$ for each action a
$B^{a++}_{ss'''}$	$\in[0;1]$	3-step first-action inverse model $p(a\vert ss''')$
J,K,Δ	$\in\mathbb{R}^{d\times d}$	action-independent $d\times d$ "transition" matrices
${}^+{}_+$	$\cdot^n\to\cdot$	index summation, e.g. $M^+_{s+}=\sum_{as'}M^a_{ss'}$
$\cdot$	$(\cdot,\cdot)\to\cdot$	matrix multiplication: $[AB]_{ss''}=\sum_{s'}A_{ss'}B_{s's''}$
$\odot$	$(\cdot,\cdot)\to\cdot$	element-wise multiplication of matrix elements: $[A\odot B]_{ss'}=A_{ss'}B_{ss'}$
$\oslash$	$(\cdot,\cdot)\to\cdot$	element-wise division of matrix elements: $[A\oslash B]_{ss'}=A_{ss'}/B_{ss'}$
$\otimes$	$(\cdot,\cdot)\to\cdot$	tensor product: $[\dot M\otimes\ddot M]_{ss'}:=\dot M_{\dot s\dot s'}\ddot M_{\ddot s\ddot s'}$ with $s=(\dot s,\ddot s)$ and $s'=(\dot s',\ddot s')$

B Characterizing M and W for which EqIM(1) holds

$$M^a\oslash M^+=W^a\oslash W^+ \quad\Longleftrightarrow\quad W^a=M^a\odot J \quad\text{with}\quad J:=W^+\oslash M^+$$

That is, J is independent of a. Phrased differently

$$\text{For any } M \text{ and } W\text{, EqIM(1) is satisfied} \quad \textit{iff} \quad W^a\oslash M^a \text{ is independent } a. \tag{14}$$

For a given M, this allows to determine all W consistent with EqIM(1), by just multiplying with any a-independent $J\geq 0$. Not all J though lead to W consistent with (7). In order to also satisfy (7), J needs to be restricted as follows: With $\Delta_{ss'}:=J_{ss'}-1$, (7) becomes

$$0\stackrel{!}{=}W^a_{s+}-M^a_{s+} = \sum_{s'}M^a_{ss'}(\Delta_{ss'}+1)-M^a_{s+} = \sum_{s'}M^a_{ss'}\Delta_{ss'} \tag{15}$$

For each fixed s, these are k homogenous linear equations (one for each a) in d variables. Given M, all and only the W consistent with EqIM(1) *and* (7) can be obtained via $W^a=M^a\odot(1+\Delta)$ with Δ satisfying $M^{\cdot}_{s\cdot}\Delta_{s\cdot}=0$.

As a special case, $\Delta=0$ necessarily if and only if the rank of $M^{\cdot}_{s\cdot}$ is $\geq d$ for every s. This gives the precise conditions as stated in Proposition 1 under which (i) is true. We will next show that EqIM(2) removes this limitation.

C Characterizing M and W for which EqIM(1) and EqIM(2+) hold

From Appendix B we know that the most general Ansatz for W^a satisfying EqIM(1) is $M^a\odot(1+\Delta)$. Plugging this into (31) and expanding in Δ, we get

$$\begin{aligned}0 \;=\;& M^a M^+\odot(M^+)^2 - M^a M^+\odot(M^+)^2 \\ &+\; M^a M^+\odot[M^+(M^+\odot\Delta)+(M^+\odot\Delta)\odot M^+] - [(M^a\odot\Delta)M^+ M^a(M^+\odot\Delta)]\odot(M^+)^2] \\ &+\; M^a M^+\odot(M^+\odot\Delta)^2 - (M^a\odot\Delta)(M^+\odot\Delta)\odot(M^+)^2\end{aligned}$$

This is a collection of quadratic equations in Δ. The Δ-independent first line is 0. We can write this in canonical form:

$$\begin{aligned}&\Sigma_{kl} A^a_{ss'',kl}\Delta_{kl} \;=\; R^a_{kl}(\Delta) \quad\text{with} \qquad\qquad (16)\\ &A^a_{ss'',kl} \;:=\; (\Sigma_{s'} M^a_{ss'} M^+_{s's''})(M^+_{sk}M^+_{ks''}\delta_{ls''} + M^+_{sl}M^+_{ls''}\delta_{sk} - M^a_{sk}M^+_{ks''}\delta_{ls''} - M^a_{sl}M^+_{ls''}\delta_{sk}) \\ &R^a(\Delta) \;:=\; (M^a\odot\Delta)(M^+\odot\Delta)\odot(M^+)^2 - M^a M^+\odot(M^+\odot\Delta)^2\end{aligned}$$

Let us consider A^a as a $d^2\times d^2$ matrix for each a, Δ as a vector of length d^2, and (wrongly) presume $R^a\equiv 0$ at first. A^a is a sum of 4 terms. The second and fourth terms are block-diagonal matrices (d blocks of size $d\times d$ in the diagonal) due to the δ_{sk}. The first and third terms are scrambled block-diagonal matrices due to the $\delta_{ls''}$, or more precisely, consist of $d\times d$ blocks, each bock being a $d\times d$ diagonal matrix. If M^a has full rank, each of the four terms has full rank d^2, but A^a itself can have lower rank, 0-eigenvalues due to some cancellations. Random M apparently achieves the highest rank, but even then, A^a itself has only rank $d(d-1)$.

Actually, $A^a\Delta = 0$ is required to hold for all a, so the rank of A as a $kd^2\times d^2$ matrix may still be d^2. But $A^+\equiv 0$ for $k=2$ implies $A^0 = -A^1$, hence the rank is still at most $d(d-1)$. $k>2$ may rectify this, but there is an alternative, which works for all a: Δ also needs to satisfy (15), which can be rewritten as

$$\sum_{kl} C^a_{s,kl}\Delta_{kl} \;=\; 0 \quad\text{with}\quad C^a_{s,kl} \;:=\; M^a_{sl}\delta_{sk} \qquad (17)$$

These give another kd constraints, and apparently often d new ones from random M. If we combine $A' := \binom{A^\cdot}{C^\cdot}$, this implies that A' has often rank d^2, so $A'\Delta = 0$ can only be satisfied for $\Delta = 0$. For $k=2$, $A^+=0$, so inclusion of either A^0 or A^1 in A' would suffice, but C^0 and C^1 are potentially independent, so both have to be included.

Let us now return to the real case of $R^a\neq 0$ for full random M, hence full-rank A'. With $R' := \binom{R^\cdot}{0}$, we need to solve $A'\Delta = R'$. Note that $R' = R'(\Delta)$ is not a constant, but a (homogenous) quadratic function of Δ itself. Consider any $\Delta = \Theta(\varepsilon)$, then $A'\Delta = \Theta(\varepsilon)$ while $R'(\Delta) = \Theta(\varepsilon^2)$, which is a contradiction for sufficiently small ε (this argument can be made rigorous). This implies that no Δ with $0 < ||\Delta|| < \varepsilon$ can satisfy $A'\Delta = R'(\Delta)$. In conclusion,

Proposition 3 (Random M and full-rank A')

If A' has full rank and W is close to M, then EqIM(1) and EqIM(2) imply $W = M$
Empirically A' has full rank for random M

which of course implies EqIM(i)$\forall i$ and also (iv). Globally, i.e. if W is not close to M, these implications may not hold.

We have yet to establish sufficient conditions which M^a lead to full-rank A'. Empirically, this has been true for random M^a, so should hold almost surely if M are sampled uniformly. One might conjecture that full-rank M^a are sufficient, but this is not the case. For instance, if M^a is independent a, then $A'\equiv 0$.

Zero A and R for full-rank $\dot{M}^a$. We finally we note that A and R can have low rank, indeed $A\equiv 0\equiv R$ even for a-dependent full-rank $\dot{M}^a$: Consider the example $\dot{M}^a$ from (21) or its generalization (26): First, if for two matrices M^a and $M^{a'}$ only one s' (depending on s and s'') contributes to the sum in $M^a M^{a'}$ then $(M^a\odot J)(M^{a'}\odot J) = M^a M^b\odot K$ for some K. This makes (18) valid for $M^a := \dot{M}^a$ and $W^a := \dot{M}^a\odot J$ for any J, since for $aa'\neq bb'$ both sides are 0 by construction of $\dot{M}^a$ (the $\odot K$ does nothing to it), and are trivially equal for $aa' = bb'$. By summing over $a'bb'$, also (31) is valid for any J, hence of course also for $J = 1+\Delta$ for any Δ. Since (16) is equivalent to (31), (16) holds for any Δ. This can only be true for $A\equiv 0$ and $R\equiv 0$. This degeneracy in itself does not violate (ii), since the probability constraints require $W = M$, as established earlier.

D EqIM(1)∧EqIM(2+)$\not\to$EqIM(3) for full low rank M?

The following numerical approach may lead to counter-examples with full support to (v) without any divisions by 0 ($M^+_{ss'} > 0$ and $W^+_{ss'} > 0\ \forall ss'$). We now consider full M^a but of rank $r < d$. The most interesting case is where all M^a span the same row-space, i.e. $M^a = L^a \cdot R$, where L^a are $d \times r$ matrices and R is a $r \times d$ matrix. Recall $A' := \binom{A^\cdot}{C^\cdot}$ with A^a and C^a defined in (16) and (17). Empirically, for $k=2$, the rank of A' typically is $\min\{d^2,(3r-1)d-r(r-1)\}$, never more, and only in degenerate cases less. Hence for $r=2$, A' is singular for $d \geq 5$. Hence for $d \geq 5$, there exist $\Delta \neq 0$ with $A'\Delta = 0$,

For $\Delta_0 := \Delta = \Theta(\varepsilon)$, this is an approximate $\Theta(\varepsilon^2)$ solution of $A'\Delta = R'(\Delta)$. By iterating $\Delta \leftarrow \Delta_0 + A'^{+} R'(\Delta)$, where A'^{+} is the pseudo-inverse of A', we get an $\Theta(\varepsilon^i)$-approximation after $i-2$ iterations. This should rapidly converge to an "exact" non-zero(!) solution $A'\Delta = R'(\Delta)$. This would show that (ii) can fail for full M. Generically, this solution also violates EqIM(3), i.e. also (vi) can fail. By this we mean, for randomly sampled L^a and R (for $a = r = 2$ and $d \geq 56$) and performing the procedure above, EqIM(3) does not hold. There is a caveat with this argument, namely if R' is not in the range of A', then this construction fails.

E EqIM(1) does not imply EqIM(2) ($\odot$-version)

We have already given a simple example that violates (v) in Section 3, but the example and methodology provided here generalizes to (vi) and even larger i. We consider deterministic reversible forward dynamics for any policy $\pi(a|s) > 0\ \forall as$. For simplicity we assume $k=2$ and uniform policy $\pi(a|s) = \frac{1}{2}$. We defer a discussion of $0/0$ to the end of the next Appendix.

We consider M^a and W^a that permute states. That is, $M^\cdot_{ss'} := [\![s' = \pi^\cdot(s)]\!]$ and $W^\cdot_{ss'} := [\![s' = \sigma^\cdot(s)]\!]$ for some permutations $\pi^\cdot, \sigma^\cdot : \{1,...,d\} \to \{1,...,d\}$. Strictly speaking, we should multiply this by $\pi(a|s) = \frac{1}{k}$, but this global factor plays no role here, so will be dropped everywhere. Matrix multiplication corresponds to permutation composition: $[M^\cdot W^\cdot]_{ss''} = [\![s'' = \sigma^\cdot(\pi^\cdot(s)]\!]$. We denote example permutation (matrices) by $[\pi] = [\pi(1)...\pi(d)]$.

We now construct a counter-example for (v): For $d=4$, let $M^0 = W^0 = \mathrm{Id} = [1234]$ be the identity matrix/permutation. Let $W^1 = [2341]$ be the cyclic permutation $1 \to 2 \to 3 \to 4 \to 1$, and $M^1 = [2143]$ the cycle pair $1 \leftrightarrow 2$ and $3 \leftrightarrow 4$. We know from (14) that EqIM(1) holds *iff* $W^a \oslash M^a$ is independent a $(= J)$ *iff* $W^a \oslash M^a = W^b \oslash M^b\ \forall a,b \in \{0,1\}$ *iff* $W^a \odot M^b = M^a \odot W^b$. Case $a=b$ is trivial, so only $W^0 \odot M^1 = M^0 \odot W^1$ needs to be verified. Now $M^\cdot \odot W^\cdot$ of two permutations matrices is *not* a permutation matrix (unless $M^\cdot = W^\cdot$). It still a 0-1 matrix with at most one non-zero entry in each row and column. We can generalize the permutation notation to "sub-permutations" by defining $\pi(s) = \emptyset$ if row s is empty. For instance $M^1 \odot W^1 = [2\emptyset 4\emptyset]$. EqIM(1) holds, since $W^0 \odot M^1 = [\emptyset\emptyset\emptyset\emptyset] = M^0 \odot W^1$.

Similarly EqIM($2a$) holds *iff* $W^a W^{a'} \oslash M^a M^{a'}$ is independent a,a' *iff*

$$W^a W^{a'} \odot M^b M^{b'} \;=\; M^a M^{a'} \odot W^b W^{b'} \quad \forall a,a',b,b'. \tag{18}$$

But for $a = a' = 0$ and $b = b' = 1$ we have

$$(W^0)^2 \odot (M^1)^2 \;=\; [1234] \odot [1234] \;=\; [1234] \;\neq\; [\emptyset\emptyset\emptyset\emptyset] \;=\; [1234] \odot [3412] \;=\; (M^0)^2 \odot (W^1)^2$$

hence EqIM(1) does not necessarily imply EqIM(2). The advantage of formulation (18) over (8) is that matrix sums M^+ and W^+ are more complicated objects than the sub-permutation matrices (18). Like random matrices, permutation matrices, have full rank, but unlike random matrices they can violate (ii), (iv), and (vi).

F EqIM($1a$)∧...∧EqIM(ia) do not imply EqIM($i+1$) ($\odot$-version)

Counting variables and equations made the possibility of violating (v) for $k < d$ plausible (cf. positive result for $k \geq d$). A similar counting argument indicates that (vi) and higher i analogues might actually hold. Unfortunately this is not the case. I.e. even providing inverse models for all action sequences up to length i is not sufficient to always uniquely determine the probability of longer action sequences. This is true even for deterministic reversible forward dynamics for any policy $\pi(a|s) > 0\ \forall as$. As for $i=1$, we assume $k=2$, $\pi(a|s) = \frac{1}{2}$, gloss over $0/0$, and don't normalize M and W.

For $i=2$, $M^0:=W^0:=\text{Id}=[123456]$ and $W^1:=[234561]=:\sigma$ (σ for 'cycle') and $M^1:=[231564]=:\pi$ can be shown to satisfy EqIM(1) and EqIM($2a$) but violate EqIM(3). The calculations are not to onerous, but lets consider directly the general i case: Consider even $d=:2d'$ and identity and cycle (pair)

$$\begin{aligned} M^0 &= W^0 = \text{Id} = [1,2,...,d-1,d], \\ W^1 &= [2,3,...,d,1], \quad M^1 = [2,3,...,d',1,d'+2,...d-1,d,d'+1] \end{aligned}$$

EqIM(ia) holds *iff* $W^aW^{a'}...\oslash M^aM^{a'}...=W^+W^+...\oslash M^+M^+...$ is independent $aa'...$ *iff*

$$W^aW^{a'}...W^{a^i}\odot M^bM^{b'}...M^{b^i} = M^aM^{a'}...M^{a^i}\odot W^bW^{b'}...W^{b^i} \quad \forall aa'...a^i,bb'...b^i \tag{19}$$

(While this looks like k^{2i} matrix equations, by chaining, checking k^i pairs suffices, which is the same number as in EqIM(ia)). Now $W^aW^{a'}...W^{a^i}$ consists of only two types of matrices, a cycle for $W^1=\sigma$ and identity W^0. The $W^0=\text{Id}$ can be eliminated, leading to $(W^1)^{a^+}$, where $a^+:=a+a'+...+a^i$. Similarly $M^bM^{b'}...M^{b^i}=(M^1)^{b^+}$, etc. Hence we only need to verify

$$(W^1)^{a^+}\odot(M^1)^{b^+} = (M^1)^{a^+}\odot(W^1)^{b^+} \quad \text{for} \quad 0\leq a^+,b^+\leq i \tag{20}$$

$$\begin{aligned} (W^1)^{a^+} &= [a^++1,a^++2,...,d,1,2,...,a^+], \quad \text{while} \\ (M^1)^{b^+} &= [b^++1,...,d',1,...,b^+,d'+1+b^+,...,d,d'+1,...,d'+b^+] \end{aligned}$$

hence $(W^1)^{a^+}\odot(M^1)^{b^+}=[\emptyset...\emptyset]=0$ for $0\leq a^+\neq b^+<d'$. For $a^+=b^+$ both sides of (20) are equal too. Hence if we choose $d'=i+1$, (20) and hence EqIM(1)...EqIM(ia) are all satisfied. If we choose $d'=i$, $a^+=d'$, $b^+=0$, (20) reduces to

$$\begin{aligned} (W^1)^{d'}\odot(M^1)^0 &= [d'+1,...,d,1,...,d']\odot\text{Id} = 0, \quad \text{and} \\ (M^1)^{d'}\odot(W^1)^0 &= \text{Id}\odot\text{Id} = \text{Id} \end{aligned}$$

which are of course not equal. Hence EqIM(i) fails for $d'=i$. Summing over all $a'...a^{d'}$ and $b'...b^{d'}$, and noting that all other terms are 0 or cancel, shows that EqIM($i+$) fails too. Together this shows for $d'=i+1$ that EqIM(1)...EqIM(ia) do not imply any version of EqIM($i+1$).

Despite M^a having full rank, A and A' defined in Appendix C have very low rank, indicating potentially many more consistent W.

A downside of this example is that it strictly only applies to the $\odot$-version (19). Many entries of M^+ and W^+ and powers thereof are 0, so (8) contains many divisions by zero. We were not able to extend this example by mixing in e.g. a uniform matrix as done in the first counter-example to (v).

Many real-world MDPs are sparse. Only a subset $G\subseteq S\times S$ of transitions $s\to s'$ is possible. For $(s,s')\notin G$, $p(s'|sa)=0\ \forall a$, or formally $M^a_{ss'}=M^+_{ss'}=0$. In this case, no action causes $s\to s'$ and $p(a|ss')=M^a_{ss'}/M^+_{ss'}$ being undefined is actually appropriate. So we could restrict (s,s') to G (and analogously $(s,...,s^i)$ and (ss^i) by chaining G) in the conditions and conclusions of the various conjectures. It is then also natural to restrict the model class to $\mathcal{M}:=\{M^\cdot : M^+_{ss'}>0 \Leftrightarrow (s,s')\in G\}$. For unknown G, the condition $M,W\in\mathcal{M}$ then becomes $M^+_{ss'}>0 \Leftrightarrow W^+_{ss'}>0$. Unfortunately the above counter-example does not even satisfy this weaker condition, but the more complicated example of Appendix G does.

G Non-Uniqueness of Inverse MDP Models for $i\geq 2$

In Appendices E/F we provided conjectured/unsatisfactory counter-examples to EqIM($1:i$) $\Rightarrow$ EqIM($i+1$). Here we provide a fully satisfactory counter-example that avoids the "bad" 0/0.

EqIM(1) and EqIM($2a$) do not imply EqIM(3). Consider two matrices $\dot M^0$ and $\dot M^1$ with disjoint support, i.e. $\dot M^0\odot\dot M^1=0$. In this case $\dot M^a\oslash\dot M^+\in\{0,1,\perp\}^{d\times d}$ is a partial binary matrix with entry undefined ($\perp$) wherever $\dot M^+=0$ but otherwise 0 wherever $\dot M^a=0$ and 1 wherever $\dot M^a>0$. That is, it is insensitive to the actual (non-zero) values of $\dot M^a$. A simple such $\dot M$ is $\dot M^0=\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ and $\dot M^1=\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right)$, ignoring normalization. For now we ignore ss' for which $\dot M^+_{ss'}=0$ and return to this issue later.

We consider M^a and W^a that permute states. That is, $M^{\cdot}_{ss'} := [\![s'=\pi^{\cdot}(s)]\!]$ and $W^{\cdot}_{ss'} := [\![s'=\sigma^{\cdot}(s)]\!]$ for some permutations $\pi^{\cdot},\sigma^{\cdot} : \{1,...,d\} \to \{1,...,d\}$. Strictly speaking, we should multiply this by e.g. $\pi(a|s)=\frac{1}{k}$, but this global factor plays no role here, so will be dropped everywhere. Matrix multiplication corresponds to permutation composition: $[M^{\cdot}W^{\cdot}]_{ss''}=[\![s''=\sigma^{\cdot}(\pi^{\cdot}(s)]\!]$. We denote example permutation (matrices) by $[\pi]=[\pi(1)...\pi(d)]$. Consider now

$$\begin{aligned} \dot{M}^0 &:= [456123] =: [\pi_0] \\ \dot{M}^1 &:= [231645] =: [\pi_1] \end{aligned} \quad\Longrightarrow\quad \begin{aligned} \dot{M}^0\dot{M}^0 &= [123456] \\ \dot{M}^0\dot{M}^1 &= [564312] \\ \dot{M}^1\dot{M}^0 &= [645231] \\ \dot{M}^1\dot{M}^1 &= [312564] \end{aligned} \tag{21}$$

No column contains the same number twice, hence this not only satisfies $\dot{M}^0\odot\dot{M}^1=0$ but also

$$\dot{M}^a\dot{M}^{a'}\odot\dot{M}^b\dot{M}^{b'} = 0 \quad \text{unless } a=b \text{ and } a'=b' \tag{22}$$

That $6\to5\to4\to6$ is in reverse oder to $1\to2\to3\to1$ is crucial for making $\dot{M}^0$ and $\dot{M}^1$ not commute. Note that (22) remains valid if each 1-entry of $\dot{M}^a$ is replaced by a different non-zero scalar, since (22) is purely multiplicative. So if $\dot{W}^a=\dot{M}^a\odot\dot{J}$ for some $J>0$, then $\dot{W}^a\dot{W}^{a'}=\dot{M}^a\dot{M}^{a'}\odot K$ for some $K>0$. Let $\dot{W}^a$ be such a matrix. Then $[\dot{W}^a\dot{W}^{a'}\oslash\dot{W}^+\dot{W}^+]_{\dot{s}\dot{s}''}=1$ if $[\dot{M}^a\dot{M}^{a'}]_{\dot{s}\dot{s}''}>0$ and 0 (or undefined) otherwise, i.e. is independent of the choice of J. So such $\dot{W}\neq\dot{M}$ satisfies EqIM($2a$). Unfortunately the probability constraints $W^a_{s+}=1$ require $J^a_{ss'}=1$ when $M^+_{ss'}>0$, and hence $W=M$. But the general idea is sound and can be made work as follows:

We split one state, e.g. $s=6$ into two states $s=6a$ and $s=6b$. We leave the permutation structure intact, except that all deterministic transitions into $s=6$ are split into stochastic transitions to $s=6a$ and $s=6b$, and transitions from $6a$ and $6b$ will be to the same state as from original 6. Condition (22) is still satisfied, so the above argument still goes through, but now we can choose different stochastic transitions to $s=6a$ and $s=6b$ in W and M.

Finally, we have to show violation of EqIM(3). EqIM(ia) holds *iff* $W^aW^{a'}...\oslash M^aM^{a'}... = W^+W^+...\oslash M^+M^+...$ is independent $aa'...$ *iff*

$$W^aW^{a'}...W^{a^i}\odot M^bM^{b'}...M^{b^i} = M^aM^{a'}...M^{a^i}\odot W^bW^{b'}...W^{b^i} \quad \forall aa'...a^i,bb'...b^i \tag{23}$$

(While this looks like k^{2i} matrix equations, by chaining, checking k^i pairs suffices, which is the same number of equations as in EqIM(ia)).

It is easier to split *every* state into two states: $s:=(\dot{s},\ddot{s})$ with $\dot{s}\in\{1,...,6\}$ as before and splitter $\ddot{s}\in\{0,1\}$. $M^a_{ss'}:=\dot{M}^a_{\dot{s}\dot{s}'}\ddot{M}^{a\dot{s}}_{\ddot{s}\ddot{s}'}$. Note that $\ddot{M}$ is flexible enough to expand each 1-entry in $\dot{M}^a$ to a different 2×2 (stochastic) matrix, while the 0-entries become $\binom{0\ 0}{0\ 0}$. This flexibility is important: $\ddot{M}$ independent a or independent $\dot{s}$ would not work. Now let us write out

$$[M^aM^{a'}M^{a''}]_{ss'''} = \sum_{\dot{s}'\dot{s}''}\dot{M}^a_{\dot{s}\dot{s}'}\dot{M}^{a'}_{\dot{s}'\dot{s}''}\dot{M}^{a''}_{\dot{s}''\dot{s}'''}\sum_{\ddot{s}'\ddot{s}''}\ddot{M}^{a\dot{s}}_{\ddot{s}\ddot{s}'}\ddot{M}^{a'\dot{s}'}_{\ddot{s}'\ddot{s}''}\ddot{M}^{a''\dot{s}''}_{\ddot{s}''\ddot{s}'''} \tag{24}$$

The crucial difference to the $i=2$ case (22) is that now there are difference permutation sequences leading to the same permutation, for instance $\dot{M}^0\dot{M}^0\dot{M}^1=\dot{M}^1=\dot{M}^1\dot{M}^0\dot{M}^0$. Let us choose $aa'a''=001$ and $\dot{s}=1$, then only $\dot{s}'=\pi_0(\dot{s})=4$ and $\dot{s}''=\pi_0(\dot{s}')=1$ contribute to the sum and $\dot{s}'''=\pi_1(\dot{s}'')=2$. For this choice, (24) becomes $1\cdot1\cdot1\cdot[\ddot{M}^{01}\ddot{M}^{04}\ddot{M}^{11}]_{\ddot{s}\ddot{s}'''}$. If we replace $aa'a''$ in (24) by $bb'b''$ and then choose $bb'b''=100$ and again $\dot{s}=1$, then only $\dot{s}'=\pi_1(\dot{s})=2$ and $\dot{s}''=\pi_0(\dot{s}')=5$ contribute and $\dot{s}'''=\pi_0(\dot{s}'')=2$. For this choice, (24) becomes $1\cdot1\cdot1\cdot[\ddot{M}^{11}\ddot{M}^{02}\ddot{M}^{05}]_{\ddot{s}\ddot{s}'''}$. We now define $W^a_{ss'}:=\dot{M}^a_{\dot{s}\dot{s}'}\ddot{W}^{a\dot{s}}_{\ddot{s}\ddot{s}'}$. Since $\dot{M}$ remains the same, the same action and state sequences above lead to the same result for W, just with $\ddot{M}$ replaced by $\ddot{W}$. If we plug the four expressions into (23) (for $i=3$) we get

$$\ddot{W}^{01}\ddot{W}^{04}\ddot{W}^{11}\odot\ddot{M}^{11}\ddot{M}^{02}\ddot{M}^{05} = \ddot{M}^{01}\ddot{M}^{04}\ddot{M}^{11}\odot\ddot{W}^{11}\ddot{W}^{02}\ddot{W}^{05}$$

Since this expressions involves 10 different 2×2 stochastic matrices, there are plenty of choices to make both sides different. If we choose all 2×2 matrices to have full support, then by construction, W and M have the same support, hence constitute a proper counter-example to EqIM(3). We now extend this construction to $i>2$.

EqIM$(1a)\wedge...\wedge$EqIM(ia) do not imply EqIM$(i\!+\!1)$. The construction in the previous paragraph generalizes to $i>2$: We need to find two permutations $\dot M^0=\pi_0$ and $\dot M^1=\pi_1$ such that for each fixed $j\leq i$ all possible 2^j concatenations (products) of these permutation (matrices) differ in the sense that no s is mapped to the same s^j (they have disjoint support). Since all $\dot M^a\dot M^{a'}...\dot M^{a^j}\in\{0,1\}$, we can write this condition compactly as

$$\sum_{aa'...a^j}\dot M^a\dot M^{a'}...\dot M^{a^j}\;\in\;\{0,1\}^{d\times d}$$

By factoring the sum, this is equivalent to $(\dot M^+)^j\in\{0,1\}^{d\times d}$. Note that $[(\dot M^+)^j]_{ss^i}$ counts the number of action sequences $aa'...a^j$ of length j that lead from s to s^i. For $j=i+1$, we want this condition to be violated. So in order to disprove the implication we need to find two permutations M^0 and M^1 such that

$$(\dot M^+)^j\in\{0,1\}^{d\times d}\quad\forall j\leq i\quad\text{but}\quad(\dot M^+)^{i+1}\not\in\{0,1\}^{d\times d} \tag{25}$$

The rest of the argument is the same as for the $i=2$ case above: creating two versions M^a and W^a of $\dot M^a$ by spitting one or all states into two, and replacing the 1s by 2×2 different stochastic matrices. As for the choice of $\dot M^a$, for $i=3$ we can choose 3-cycle and 5-cycle

$$\begin{aligned}\dot M^0 &= [6,7,8,9,10,11,12,13,14,15,1,2,3,4,5]\\ &= (1,6,11)(2,7,12)(3,8,13)(4,9,14)(5,10,15)\\ \dot M^1 &= [2,3,4,5,1,8,9,10,6,7,14,15,11,12,13]\\ &= (1,2,3,4,5)(6,8,10,7,9)(11,14,12,15,13)\end{aligned} \tag{26}$$

where we also provide the more conventional cycle notation in round brackets. Crucially the 5-cycles have been chosen to not commute with the 3-cycles ($M^0M^1\neq M^1M^0$). Conditions (25) can easily be verified numerically. For higher i we need p cycles and q cycles, where p and q are relative prime and sufficiently large. We need at least $d=p\cdot q\geq 2^i$, otherwise $\dot M^+\not\in\{0,1\}^{d\times d}$ by a simple pigeon-hole argument. To prove EqIM$(1a)\wedge...\wedge$EqIM$(ia)\not\Rightarrow$EqIM$(i+1)$ in general for arbitrarily large i, we need to invoke some group theory. All-together we have shown that

Proposition 4 ((i)-(vi) can fail) *EqIM$(1a)\wedge...\wedge$EqIM(ia) do not necessarily imply EqIM$(i+1)$ for any i. This in turn implies that (i)-(vi) each can fail for some $M^{\cdot}$.*

H Deterministic Cases

Deterministic planning / reachability problem. If we are only interested in finding *some* action sequence $aa'...a^i$ that leads to s^i, the problem becomes easy: The only thing that matters is the support of the various matrices, not the numerical values themselves. Since $B^a_{ss'}>0$ iff $M^a_{ss'}>0$ (either assuming $M^+_{ss'}>0$ or regarding $\perp>0$ as False), and similarly for higher orders, we can replace M^a by B^a in (iii), and get $B^{aa'...a^i}_{ss^{i+1}}>0$ iff $[B^aB^{a'}...B^{a^i}]_{ss^{i+1}}>0$. We could also replace M^a by $G^a_{ss'}:=[\![B^a_{ss'}>0]\!]$, then $[G^aG^{a'}...G^{a^i}]_{ss^{i+1}}>0$ counts the number of paths of length i from s to s^{i+1} via action sequence $aa'...a^i$, and hence determines whether s^{i+1} can be reached. Similarly $(G^+)^i>0$ iff there is *some* action sequence that can reach s^{i+1} from s. An action a such that $G^a(G^+)^i>0$ can be chosen as the first action of such a sequence if it exists, and $a',a''...$ can be found the same way by recursion. So this deterministic planning/reachability problem has a "unique" solution, which can be found in time $O(i\cdot d\cdot(d+k))$ (for fixed s and s^{i+1}).

***B* is deterministic.** Assume $M^a_{ss'}/M^+_{ss'}=:B^a_{ss'}\in\{0,1,\perp\}$. This is true if and only if M^a has disjoint support for different a, i.e. iff $M^a\odot M^b=0\;\forall a\neq b$. This in turn means that $B^a_{ss'}=[\![W^a_{ss'}>0]\!]$ for any and only those W with same support as M, and hence also $W^a\odot W^b=0\;\forall a\neq b$, which is another failure case of (i). Here we have included the case where *no* action leads from s to s', in which case $W^+_{ss'}=0$ and B^a is undefined ($\perp$). This readily extends to higher orders: If $B^{aa'\cdots}\in\{0,1,\perp\}$, then $B^{aa'\cdots}=[\![W^aW^{a'\cdots}\oslash(W^+)^i>0]\!]$ iff $W^aW^{a'}...$ has the same support as $M^aM^{a'}...$ and

$$W^aW^{a'}...W^{a^i}\odot W^bW^{b'}...W^{b^i}=0\quad\forall aa'...a^i\neq bb'...b^i \tag{27}$$

Note that $W^a\odot W^b=0$ does not necessarily imply (27), e.g. for $W^0=\frac{1}{2}\binom{1\;0}{0\;1}$ and $W^1=\frac{1}{2}\binom{0\;1}{1\;0}$, $(W^0)^2=(W^1)^2$. In Appendices G&E&F) we construct W such that (27) holds for larger i.

I Applications

Consider an agent who has control over $\dot{s}$ but not over $\ddot{s}$. For instance a robot equipped with a camera can control its position and orientation, but not the shape and color of objects in its path. The forward model $p(s'|as)$ essentially involves modelling the whole observable world. The inverse model $p(a|ss')$ on the other hand can ignore inputs that the agent has no control over. Of course in practice, s does not come neatly separated into $\dot{s}$ and $\ddot{s}$, so a (say) deep neural network still has to learn the controllable features, but neither needs to learn nor predict the uncontrollable features (under the factorization assumptions described in Section 3, now in feature space).

If the goal is to navigate from s to s^i in i time steps, and open-loop control suffices as e.g. in (near)-deterministic problems [EMK$^+$22], then action sequences for which $p(aa'...a^{i-1}|ss^i)$ is large are the most likely that caused the transition to s^i, hence these sequences are promising candidates for macro actions (temporally extended actions, options) in Reinforcement Learning [SP02, Pre00].

Since the action space is typically much smaller than the state space (the former often finite, the latter often even infinite-dimensional), even learning $p(aa'...a^{i-1}|s...s^i)$ directly for all small i can be feasible and may be more efficient than learning the one-step forward model. A closed-loop alternative would be to learn only $p(a|s...s^i)$, find the likely first action a that caused the ultimate transition to s^i, then take action a, iterate, and store the resulting sequence as an option.

The required sample complexity to learn inverse MDP models for larger i directly from data may grow exponentially in i, which is why inferring i-step inverse models from 1-step and 2-step inverse models would be useful. The fact that this problem borders NP-hardness probably prevents even powerful transformer models to finding the structure in $p(aa'...a^{i-1}|s...s^i)$ by themselves.

J Systems of Quadratic Matrix Equations

A System of Polynomial Equations (SPE) is a set of multivariate polynomial equations $\text{Poly}_j(x,y,z,...)=0$ over $\mathbb{R}$ in n variables $x,y,z,u,v,w,...\in\mathbb{R}$ for $j\in\{1:m\}$. This class is NP-hard (via a simple reduction from 1in3SAT, see Section K). We can recursively replace each product xy (sum $bu+cv$) in the polynomials by a new variable z (w) and add "polynomial" equation $z=xy$ ($w=bu+cv$). This results in SPEs consisting of only linear equations with a single $+$ ($bu+cv=w$) and quadratic equations without any $+$ ($xy=z$), which are still (even with all $a=b=1$ and $x=y=z$) NP-hard. We call them Simple Systems of Quadratic Equations (Simple SQE). For the reduction process to actually work we need one further dummy variable and equation $q=1$ (to reduce $bu+c=w$). Alternatively, with some extra work, we can reduce any SPE into a Simple SQE asking for a *non-zero* solution. We will pursue the latter, since this is closer to our interest (SQE (16) with solution $\Delta\not\equiv 0$). We can even merge the linear and quadratic equations into a single form $xy=bu+cv$ by choosing $b=1$ and $c=0$ (replacing xy by w and adding $xy=0\cdot u+1\cdot w$).

We define a System of Polynomial/Quadratic Matrix Equations (SPME/SQME) as a set of m multivariate (quadratic) polynomials $\text{Poly}_j(\Delta,\Gamma,...|A,B,C,...)=0$ in the (unknown) matrix variables $\Delta,\Gamma,...$ *and* the (given) matrix constants ("coefficients") $A,B,C,...$. Alternatively, Poly_j might be viewed as generalized polynomials over a *non*-commutative matrix ring in the unknowns only. In any case, note that

$$A\cdot\Delta\cdot A'\cdot\Delta\cdot A''+B\cdot\Delta\cdot B'+C \;\neq\; (A\cdot A'\cdot A'')\cdot\Delta^2+(B\cdot B')\cdot\Delta+C$$

By writing out all matrix operations in terms of their scalar operations, SPME is of course a sub-class of SPE. SPE is also a sub-class of SPME (choose all matrices to be 1×1 matrices), which implies SPME is NP-hard. But we are interested in NP-hard *small* subclasses of SPME, so will construct a more economical embedding: Assume we have a Simple SQE with n variables $x,y,z,u,v,...$. We place them into $d\times d$ matrix Δ ($d\geq\sqrt{n}$) introducing dummy variables for the remaining entries. We can extract variable $w=\Delta_{ss'}$ via $w=\mathrm{e}^{s\top}\cdot\Delta\cdot\mathrm{e}^{s'}$, where e^s is basis vector ($d\times 1$-matrix) $(\mathrm{e}^s)_{s'1}=\delta_{ss'}$. If we replace all variables in the Simple SQE expressions $xy=au+bv$ by such expressions, we get a Simple SQME with Poly_j equations of the form (dropping $\cdot$ as usual)

$$a^j\Delta A'^j\Delta a''^j \;=\; b^j\Delta b'^j+c^j\Delta c'^j \quad \forall j \tag{28}$$

While these are scalar equations, since the outer matrices are $1\times d$ on the left and $d\times 1$ on the right, technically they are matrix equations. We could pad all involved matrices, including the outer ones, with zeros to square $\mathbb{R}^{d\times d}$ matrices of the same size (for sufficiently large d, and only polynomial overhead).

We can reduce (28) to just one equation at the cost of making the equations more complicated as follows: Write each equation $\text{Poly}_j=0$ in the form $\text{e}^s\cdot\text{Poly}_j\cdot\text{e}^{s'\top}=0$, with a different (s,s')-pair for each j. These are now "proper" matrix equations, but with all entries identically 0 except entry (s,s') being Poly_j. This allows us to sum all equations without conflating them into one (complex) matrix equations

$$\textstyle\sum_j A^j\Delta A'^j\Delta A''^j \;=\; \sum_j B^j\Delta B'^j+C^j\Delta C'^j \qquad (29)$$

Another way to combine (28) into one equation is by putting all M^j for all j into one block-diagonal matrix $\tilde{M}:=\text{Diag}(M^1,...,M^m)$ for $M\in\{a,A',a'',b,b',c,c',\Delta\}$. For $\tilde\Delta$ we need to ensure that indeed all blocks $\Delta^j=\Delta$ are equal. This can be done via $\tilde\Pi^\top\tilde\Delta\tilde\Pi=\tilde\Delta$ for some cyclic block permutation $\tilde\Pi$. We further need to ensure that the off-diagonal blocks of $\tilde\Delta$ are zero. We can zero each block with one equation, but it seems impossible to zero all with a bounded number of Simple QMEs. We can modify the decision problem to decide whether specific sparse solutions $\tilde\Delta$ exist. Formally, we can introduce element-wise multiplication $\odot$ and allow one equation of the form $\tilde B\odot\tilde\Delta=0$ with $\tilde B$ being $0/1$ on the on/off-diagonal blocks. This leads to a Simple SQME with $\odot$ in 3 equations (dropping the $\sim$)

$$A\Delta A'\Delta A'' \;=\; B\Delta B'+C\Delta C', \quad \Pi^\top\Delta\Pi=\Delta, \quad B\odot\Delta=0 \qquad (30)$$

Proposition 5 (NP-hardness of Simple SQME) *Systems of Polynomial Equations (SPE) can be polynomially reduced to Simple Systems of Quadratic Matrix Equations (Simple SQME)* (28)*. The number of equations can be reduced to 1 at the expense of making the equations complex* (29)*, or to 2 by asking for sparse solutions or by enforcing sparsity via $B\odot\Delta=0$* (30)*. Since SPE are NP-hard, deciding the existence of non-zero solutions for all three SQME versions is also NP-hard.*

An NP-hardness proof for a Simple SQME with $\odot$ with 3 equations via reduction from 1in3SAT that looks much closer to the desired form (32) or (34) is given in Section K. By a similar reduction, encoding all n variables and their complement in the diagonal of $\Delta=\text{Diag}(x,\bar x,y,\bar y,...,)$, one can also show that solvability of

$$\Delta^2=\Delta, \quad A\Delta 1=1, \quad \text{Id}\odot\Delta=\Delta, \quad \text{with} \quad A\in\{0,1\}^{m\times 2n}$$

is NP-complete (1 is the all-1 vector, sparse A with 2 or 3 ones in each row suffice), but not all SPE can be reduced to this form.

Open Problem 6 (Are Bounded SPME NP-hard?) *Are Systems of Polynomial Matrix Equations (without $\odot$) of bounded structural complexity NP-hard? Bounded means, only the definitions of the constant matrices scale with $d\times d$, but the polynomial degrees, number of equations, and number of matrix operations are bounded.*

K Computational Complexity

Maybe even just characterizing all M for which EqIM(1) and EqIM(2) uniquely determine W is hopeless, not to speak of finding some or all W in case not. More formally, we can ask the question of whether there exists an efficient algorithm that can decide whether EqIM(i) has a unique solution. We provide some weak preliminary evidence, why this problem may be NP-hard. Appendix M contains fully self-contained a few versions of this open problem in their simplest instantiation and most elegant form.

Decidability and computability. EqIM(2) converted to (23) and (7), or (31) or (32) below form a System of Quadratic Equations (SQE). The constraint $W\neq M$ can also be expressed as a quadratic equation (see below). As such, the existence and uniqueness of solutions is formally decidable by computing a Gröbner basis [Stu02], and (some) solutions can be found by cylindrical algebraic decomposition in (double) exponential time. ε-approximate solutions can of course be found by exponential brute-force search through all W on a finite ε'-grid, and verified in polynomial time.

Complexity considerations. 3SAT is NP complete. A CNF formula in n boolean variables can easily be converted to a System of Quadratic Equations (SQE). Therefore SQE is also NP hard. EqIM(2+) explicitly written in quadratic form

$$M^a M^+ \odot (W^+)^2 - W^a W^+ \odot (M^+)^2 \;=\; 0 \tag{31}$$

constitutes an SQE in W given M, also if we include linear EqIM(1) and probability constraints (7). Non-negativity of W can be enforced with (slack) variables $(Y^a_{ss'})^2 = W^a_{ss'}$. (Similarly (16) plus constraints (15) constitute an SQE in Δ.) To reduce the uniqueness question to a solvability problem we need to avoid the trivial solution $W \equiv M$, e.g. by introducing further (slack) variables $t \in \mathbb{R}$ and $\Gamma^a_{ss'} := (W^a_{ss'} - M^a_{ss'})^2$ and constraint $t \cdot \Gamma^+_{++} = 1$. Due to the minus sign in (31), this can*not* be converted to a convex (optimization) problem. The choice of M gives significant freedom in creating SQE problems, even if only considering permutation matrices $M^a \in \{0,1\}^{d\times d}$. If one could show that every SQE can be represented as (31) [plus $W \neq M$ constraint] for a suitable choice of M, this would imply that proving the existence of $W \neq M$ satisfying (31) is NP hard. This in turn would imply that computing (any) $p(a|ss''')$ from $p(a|ss')$ and $p(a|ss'')$ is NP hard. On the other hand, matrix multiplication $W^a W^b$ is a very specific quadratic form, which may not be flexible enough to incorporate every SQE within (31).

We could not find any work on NP-hardness of Systems of Polynomial Matrix Equations (SPME). There is work on the NP-hardness of tensor problems [HL13], but this refers to the design tensors, e.g. $\sum_{jk} A_i^{jk} x_j x_k + \sum_j B_i^j x_j + C_i = 0 \;\forall i$, but the unknowns are always treated as scalars or vectors. Of course $[X \cdot Y]_{ik} = \sum_{abcd} A^{abcd}_{ik} X_{ab} Y_{cd}$, but $A^{abcd}_{ik} = \delta_{ai}\delta_{dk}\delta_{bc}$ is a very special fixed tensor (actually of low tensor rank d) with no flexibility of encoding NP-hard problems therein.

That inference in Bayesian networks is NP-complete [KF09] does not help us either for two reasons: First, in our problem the probability distribution over states and actions is only partially given. More importantly, our network for $i=2$ has only 5 nodes (s,a,s',a',s''), while the NP-hardness proofs we are aware of require large networks. Even for fixed $i>2$, it is not obvious how to encode NP-hard problems into EqIM(i), due to the severe structural constraints in EqIM(i) compared to a general network with $2i+3$ nodes. It is not clear how to exploit the fact that our (few) state nodes are large.

SQE are polynomially equivalent to Systems of Quadratic Matrix Equations (SQME), which may be the reason complexity theorists have ignored the latter. We suspect but do not know whether SQME of *bounded* structural complexity (only the definitions of the constant matrices scale with $d \times d$) is NP-hard (Open Problem 6). If we allow sparse encoding of SQE variables in W, i.e. we allow one equation involving $\odot$ of the form $B \odot W = 0$ with boolean matrix B, then bounded SQME becomes NP-hard. See Appendix J for details.

Below we directly reduce 1in3SAT to a Bounded-SQME with $\odot$ that resembles our problem as close as we were able to make it.

An NP-hard matrix problem. From EqIM(1) we know that $W^a = B^a \odot W^+$. Plugging this into EqIM($2a$) gives

$$B^{aa'} \odot (W^+ \cdot W^+) \;=\; (B^a \odot W^+)(B^{a'} \odot W^+) \quad \text{with constraints} \quad [B^a \odot W^+]_{s+} = \pi(a|s) \tag{32}$$

This set of equations is purely in terms of what is given (B^a and $B^{aa'}$) and only involves unknowns W^+ without reference to W^a. See Appendix L for some further simplification and discussion. We will show:

Proposition 7 (An NP-hard matrix problem) *Given A,B,C,Π, deciding whether the following quadratic matrix problem has a solution in W is NP-hard:*

$$A \odot (W \cdot W) = (C \odot W)(C \odot W), \quad [B \odot W]_{s+} = 1, \quad \Pi \cdot W = W \tag{33}$$

This has some resemblance to (32). Since the boundary between P and NP is very fractal/subtle, this in-itself may not imply much, but is more meant as a demonstration of how one may approach proving NP-hardness of (32).

Proof. We reduce 1in3SAT, which is an NP-complete variant of 3SAT, where each clause must have exactly one satisfying assignment, to (33). A 3CNF(n,m,g) formula is a boolean conjunction of m clauses in n variables, where each clause $c_i = \ell_{i1} \dot\vee \ell_{i2} \dot\vee \ell_{i3}$ for $i \in \{1:m\}$ is a 1-in-3 disjunction of 3 literals, and each literal is $\ell_{ia}=x_j$ or it's complement $\ell_{ia}=\neg x_j\equiv\bar x_j$, where $j=g(i,a)$ is the variable index of clause i in position a.

We arithmetize the 3CNF expression in the standard way by replacing True$\rightsquigarrow 1$, False$\rightsquigarrow 0$, and $\dot\vee\rightsquigarrow +$, i.e. we ask whether the system of linear equations $\ell_{i1}+\ell_{i2}+\ell_{i3}=1\ \forall i$ has a solution in $x_j\in\{0,1\}$. We need to encode the x's into W somehow: We aim at the following embedding:

$$W \;=\; \begin{pmatrix} x_1 & \bar x_1 & ... & x_n & \bar x_n & y_0 & ... & y_k \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & \bar x_1 & ... & x_n & \bar x_n & y_0 & ... & y_k \end{pmatrix}$$

The y are $k+1:=\max\{1,m-n+2\}$ extra dummy variables to make the matrix a square $d\times d$ matrix with $d:=\max\{m+n+2,2n+1\}$.

Choosing a cyclic permutation matrix $\Pi=[234...d1]$ ensures that all rows of W are indeed the same via $\Pi\cdot W=W$. The standard way of achieving $x_j,y_j\in\{0,1\}$ is via $x_j^2=x_j$ and $y_j^2=y_j$. This can be achieved via $(\mathrm{Id}\odot W)^2=\mathrm{Id}\odot W$, were $\mathrm{Id}_{ss'}=\delta_{ss'}$ is the identity matrix.

We use $[B\odot W]_{s+}=1$ to ensure $\bar x_j=1-x_j$, $y_0=1$, and $y_1=...=y_k=0$ and $\ell_{i1}+\ell_{i2}+\ell_{i3}=1$ by setting $B_{s,2s-1}=B_{s,2s}=1$ for $s\in\{1:n\}$, and $B_{i+n,2j-1}=1$ if $\ell_{ia}=x_j$ and $B_{i+n,2j}=1$ if $\ell_{ia}=\neg x_j$ for $i\in\{1:m\}$ and $a\in\{1,2,3\}$, and $B_{d-1,2n+1}=...=B_{d-1,2n+m}=1$, and $B_{d,2n+1}=1$, and $B_{ss'}=0$ for all other ss'. This also ensures that all rows of W sum to $n+1$, hence $W\cdot W=(n+1)W$, so $x_j\in\{0,1\}$ can be achieved via $C=\mathrm{Id}$ and $A=\frac{1}{n+1}\mathrm{Id}$ in $A\odot(W\cdot W)=(C\odot W)(C\odot W)$.

The construction implies that the 3CNF(n,m,g) formula is satisfiable *iff* (33) has a solution in W with the A,B,C,Π as constructed above. This shows NP-hardness of deciding whether (33) has a solution. A solution can trivially be verified (in the rationals or to ε-precision over the reals) in time $O(d^3)$, hence the problem is in NP, hence NP-complete.

L Compact Representation of EqIM(2+)

If only B^{a+} (EqIM(2+)) is given, we can sum (32) over a'. If we further assume $a=2$ and define $B=B^0$ and $A=B^{0+}$ and $W=W^+$ and exploit $B^+=B^{++}=1$, this reduces to the elegant quadratic matrix equation

$$A\odot(W\cdot W) \;=\; (B\odot W)\cdot W \tag{34}$$

with constraints as in (32), or even simpler $W_{s+}=1$ if π is unknown. This is the most pure formulation of the problem we are trying but are unable to solve we could come up with. For A and B defined via M, we know that (34) has a solution (namely $W=M^+$).

We neither know whether there exists an efficient algorithm to find *some* solution (34), nor to find *the* solution in case it is unique, nor to decide whether there exist solutions in case A and B are chosen arbitrarily.

The condition $W_{s+}=1$ can be relaxed to $W_{s+}>0$. If $W_{ss'}$ is a solution of (34), then also $v_s^{-1}W_{ss'}v_{s'}$ for any $v_\cdot>0$ (most easily checked via (11)). Every non-negative matrix has a real non-negative Eigenvector v, and $W_{s+}>0$ implies $v_s>0$ and Eigenvalue $\lambda>0$, hence for $W_{ss'}^{\text{norm}}:=(\lambda v_s)^{-1}W_{ss'}v_{s'}$, we have $W_{s+}^{\text{norm}}=1$.

$B^a\geq 0$ and $B^+=1$ iff $B\in[0;1]$ (and $B^1=1-B$). $B^{a+}\geq 0$ and $B^{++}=1$ iff $A\in[0;1]$ (and $B^{1+}=1-A$). But we can scale back any A and B by the same $0<\lambda<1$ to satisfy these without changing (34), i.e. these extra conditions (A and B bounded by 1) do not make the problem any simpler.

M Open Problem

We present the most important open problem(s) in their simplest instantiation and most elegant form, fully self-contained here: Consider matrices $A,B,W\in[0;1]^{d\times d}$ with $d\in\mathbb{N}$, tied by the quadratic matrix equation

$$A\odot(W\cdot W) \;=\; (B\odot W)\cdot W \quad\text{ and }\quad W_{s+}=1\;\forall s \tag{35}$$

where $\odot$ is element-wise (Hadamard) multiplication and $\cdot$ is standard matrix multiplication. The open problems are as follows: Given A and B, are there efficient algorithms which

(a) decide whether there exists a W satisfying (35)?
(b) decide whether the solution is unique, assuming (35) has a solution?
(c) compute *a* solution, assuming (35) has a solution?
(d) compute *the* solution, assuming (35) has a unique solution?

Computing a real number means, given any $\varepsilon > 0$, computing an ε-approximation. Efficient means running time is polynomial in d, ideally with a degree independent of $1/\varepsilon$. General systems of quadratic equations are known to be NP-hard, but we do not know the complexity of this particular matrix sub-class.

The upper bounds $A,B,W \le 1$ can always be satisfied by scaling, hence are irrelevant. $W_{s+} = 1$ can be relaxed to $W_{s+} > 0$ except in the uniqueness questions. If helpful: One may assume A,B,W strictly positive. Also, any finite (d-independent) number of equations of the form $A' \odot (W \cdot W) = (B' \odot W) \cdot W$ with other *general* matrices $A',B' \in [0;1]^{d \times d}$ may be added, which further constrain the solution space.

N Further Experiments

Here we provide further experiments supporting and illustrating the theory. In Appendix O we show how we numerically dealt with $B = 0/0 = \perp$. Appendix P derives the formulas for the plotted solution dimensions.

Experiments illustrating robustness to noise. The propositions and results in the main text assume that we know the one and two step inverse models ($B1 := B^a$, $B2 := B^{a+}$) exactly, but in practice these distributions must be estimated from data. Here we investigate the extent to which our algorithm is robust to noise arising from learning.

Rather than committing to a specific learning algorithm, we instead directly inject noise into the true inverse distributions. This is done by adding $\varepsilon \times 10^c$ to the true distribution and renormalizing B, where ε is drawn from the unit uniform distribution: $\varepsilon \sim \mathcal{U}[0,1]$)

Figure 4a shows that noise doesn't substantially degrade performance across several orders of magnitude (c varied -7 to 0). Additionally, the effect of this noise is substantially diminished as the horizon of the inverse model is increased (from $B1 := B^a$ to $B3 := B^{a++}$). While the is perhaps not surprising, as the entropy of such inverse distributions increases monotonically with the horizon, it still shows that noise is not compounding in a way that renders long-horizon predictions meaningless. Figure 2 buttresses this interpretation by showing that the recovered B^{a++} is qualitatively similar to the ground truth even with substantial noise.

Experiments on the Tensor-product special case. As detailed in Section 3, if M factors into two processes $\dot{M}^a \otimes \ddot{M}$, where $\ddot{M}$ is action-independent, then only the complexity of the action-dependent process $\dot{M}^a$ matters for all of our questions.

This particular special case is important because of its frequency in applied work. Many environments have most of their complexity in sub-spaces that the agent has no control over. This is illustrated by Figure 3, reproduced from [LFLDP21], wherein naturalistic videos are superimposed on relatively simple continuous control environments. Clearly, the background dynamics can be arbitrarily complex without impacting the underlying control problem.

We can construct small environments of this form via a simple procedure. We construct $\dot{M}$ with $\dot{d}$ states and k actions by sampling each element of the appropriately sized matrices from $\mathcal{U}[0,1]$ and then normalizing. $\ddot{M}$ has two states that transition uniformly regardless of the action.

The linear algorithm of Section 4 can (implicitly) output all W and $B2$ consistent with $B1$, and the formulas derived in Appendix P allow to (explicitly) calculate the dimensions of the solution spaces.

In the experiments shown in Figure 4b, $k = 5$ as in the main text, and $d = 2\dot{d}$ is varied from 16 to 32. The results show that the space of forward dynamics W is always significantly larger than the space of the 2-step inverse models ($B2$). This confirms that inverse models can be significantly simpler than forward models.

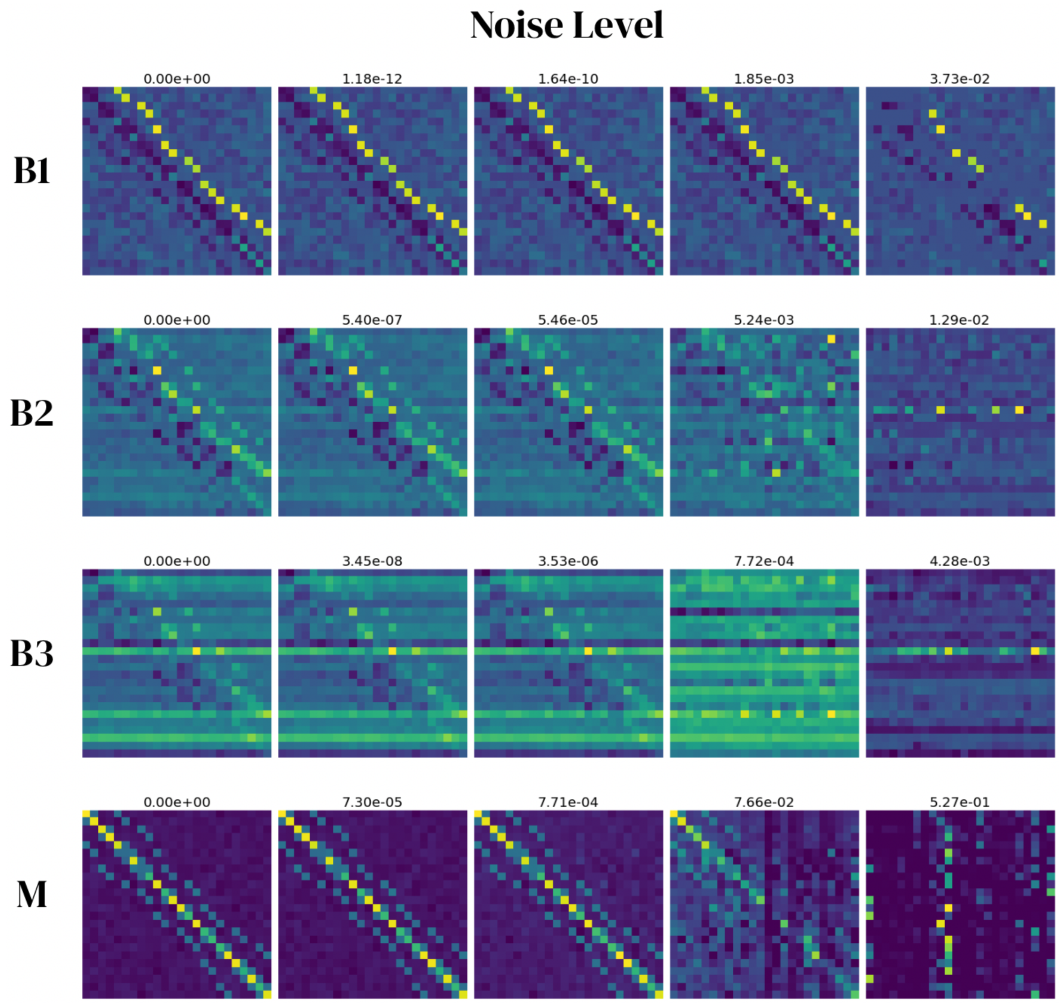
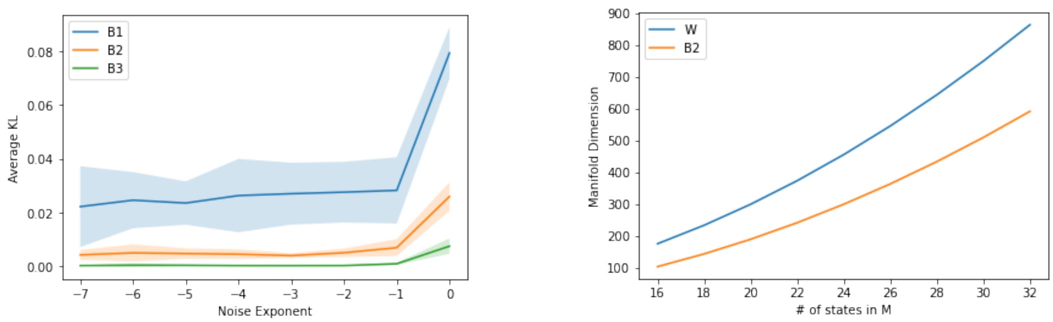


Figure 2: Reconstructing inverse and forward models from inverse models with noise injected. Noise increases exponentially across columns $[0,10^{-6},10^{-5},10^{-4},10^{-3}]$. The subplot titles show the average KL divergence of the recovered distribution from the ground truth.

Figure 3: Reproduced from [LFLDP21], this 'half-cheetah' environment has been augmented with videos of complex scenes. This highlights how non-controllable aspects of the environment can be made more complex without changing the underlying control problem. The fact that such environments are of interest motivates our focus on the Tensor-product special case.



(a) Effect of Noise

(b) Tensor-product Solution Space

Figure 4: **(a) Noise-induced reconstruction error:** In practice W must be inferred from learned estimates of $B1$ and $B2$. We investigate the effect of the resulting error on the inverse models ($B1$,$B2$,$B3$) recovered from the inferred W in terms of their proximity to the ground truth distributions. At each noise level the algorithm was run on 10 randomly generated grids, with the shaded region representing $\pm 2\sigma$. **(b) Solution dimensions of W and $B2$ given $B1$:** When the solution to an inverse model ($B2$) given only $B1$ is not unique, we can characterize the solution space in terms of its manifold dimension. By comparing this to the dimension of that of the inferred forward model (W), we can see that our algorithm has narrowed down the space of inverse models significantly more. If also $B2$ is given, the solution dimension of W reduces from d_W (blue curve) to $d_W - d_B$ (blue-orange curve).

O How to Deal with 0/0

If for some pair of states (s,s'), no action a of positive π-probability leads from state s to s', i.e. if $M^+_{ss'}=0$, then $B^+_{ss'}$ and $B^a_{ss'} \forall a$ are $0/0=\bot=$undefined. To also handle $B^\cdot_{ss'}=\bot$, we need to adapt the linear algorithm in Section 4. We provide 2 different ways of doing so, with a couple of variations, all leading to the same correct result.

We have to restrict the sum in $\sum_{s'} B^a_{ss'} J_{ss'} = \pi(a|s)$ to those s' for which $B^a_{ss'}$ is defined. We then solve for $J_{ss'}$, again for s' for which $B^a_{ss'}$ is defined, and set $J_{ss'}=0$ for those s' for which $B^a_{ss'}=\bot$. Technically this can be achieved by removing the s' columns from matrix $B^\cdot_{s\cdot}$ and $J_{\cdot\cdot}$ for which $B^\cdot_{ss'}=\bot$, solve the reduced linear equation system, and finally reinsert $J_{ss'}=0$ for the removed s'. Simpler is to replace $B^a_{ss'}=\bot$ by $B^a_{ss'}=0$, solve the equation for J, and then set $J_{ss'}=0$ for the s' for which the original $B^a_{ss'}$ was $\bot$. Some solvers automatically result in $J_{ss'}=0$, since this is the minimum norm solution, but it is better not to reply on this. Instead of setting $J_{ss'}=0$ after solving the linear system, one could also augment $B^\cdot_{s\cdot}$ with extra rows that enforce $J_{ss'}=0$.

Alternatively, we could replace $B^\cdot_{ss'}=\bot$ by a random vector which sums to 1, e.g. $B^a_{ss'}=r_a/r_+$, where $r_a=-\log u_a$ with $u_a \sim \text{Uniform}[0;1]$. Provided that the solution is unique, this also leads to the correct solution (almost surely), and in this way $J_{ss'}=0$ automatically. If the solution is not unique, $W^\cdot$ will still satisfy $B^a=W^a \oslash W^+$ when for $B^a_{ss'} \neq \bot$, but $W^\cdot_{ss'}$ may not be 0.

The adaptation of the Linear Relaxation Algorithm in Section 5 follows the same pattern: $A^{\cdot\cdot}_{ss^is^j}=\bot$ in (12), whenever one of the three involved B's is undefined. For such ss^is^j, we need to ensure that $\hat{U}_{ss^is^j}=0$, which can be done with any of the variations described above. Once we have $\hat{U}_{ss^is^j}$, we set $C^{a^i}_{ss^is^j}=0$ if $B^{a^i}_{s^is^j}=\bot$. No further intervention is needed, since $\hat{U}_{ss^is^j}=0$ already.

P Solution Dimensions of W and $B^{aa'}$.

In Section 4 we presented an algorithm for inferring W and $B^{aa'}$ from B^a. Even if M cannot uniquely be reconstructed ¬(i), $B^{aa'}$ may still be unique (iii). More generally, the solutions J and W^a form linear spaces of dimension $d_J = d_W \le d(d-1)$ ($d_J \ge d_W$ since $W^\cdot$ is linear function of J. $d_J \le d_W$, since $W^+=J$). $B^{aa'}$ is a (non-linear, polynomial) variety of dimension $d_B \le d_W$ at regular points (since it is a smooth function of W).

Parameterizing the solutions for J and W and B. We can determine the solution dimensions d_J, d_W, and d_B as follows: Let $Y_{ss'}$ be a solution of $[B^a \odot Y]_{s+} = 0$. If $\hat{J}_{ss'}$ is a solution of $[B^a \odot J]_{s+} = \pi(a|s)$, then so is $J := \hat{J} + Y$, hence $W^a := \hat{W}^a + X^a$ is a solution of $B^a = W^a \oslash W^+$ and $W^a_{s+} = \pi(a|s)$, where $\hat{W}^a := B^a \odot \hat{J}$ and $X^a := B^a \odot Y$.

If we plug in $W^a \equiv \hat{W}^a + X^a$ into $B^{aa'}$, we get the variety of $B^{aa'}$ parameterized in terms X^a. If we expand this non-linear expression up to linear order in X^a, we get after some algebra

$$B^{aa'} = [\hat{W}^a\hat{W}^{a'} + \hat{W}^a X^{a'} + X^a \hat{W}^{a'} - (\hat{W}^a\hat{W}^{a'}) \oslash (\hat{W}^+)^2 \odot (\hat{W}^+ X^+ + X^+ \hat{W}^+)] \oslash (\hat{W}^+)^2 + O(X^2) \tag{36}$$

The linear part forms a tangent direction on the variety at $\hat{B}^{aa'}$.

Determining the solution dimensions for J and W and B. Now, for each s, let $Y^r_{ss'}$ for $r \in \{1 : d_{Js}\}$ span all solutions of $[B^a \odot Y]_{s+} = 0$, which can easily be determined by SVD: d_{Js} is the number zero singular values of matrix $B^{\cdot}_{s\cdot}:$, and $Y^r_{s\cdot}$ the corresponding singular vectors. Then, $J_{ss'} = \hat{J}_{ss'} + \sum_r Y^r_{ss'} z_{sr}$ for any $z \in \mathbb{R}^{d_J}$ with $d_J = \sum_s d_{Js}$ is a solution of $[B^a \odot J]_{s+} = \pi(a|s)$.

Similarly, $W^a_{ss'} := \hat{W}^a_{ss'} + \sum_r X^{ar}_{ss'} z_{sr}$ with $X^{ar} := B^a \odot Y^r$ span all solutions consistent with B^a and π. The solution dimension is $d_W = \sum_s d_{Ws}$, where for each s, d_{Ws} is the rank of $X^{\cdot\cdot}_{s\cdot}$ if interpreted as a $kd \times d_{Js}$ matrix in $as' \times r$. d_{Ws} may be smaller than d_{Js}, since unlike $Y^r_{s\cdot}$, $X^{\cdot\cdot}_{s\cdot}$ may not be full rank.

If we plug $X^a_{ss'} = \sum_r X^{ar}_{ss'} z_{sr}$ into (36), after some index manipulation we get

$$B^{aa'} = \hat{B}^{aa'} + \sum_{t=1}^{d}\sum_{r=1}^{d_{Jt}} C^{aa'rt} z_{tr} \oslash (\hat{W}^+)^2 \oslash (\hat{W}^+)^2 + O(z^2) \quad \text{with} \tag{37}$$

$$C^{aa'rt}_{ss''} := (\hat{W}^a_{st} X^{a'r}_{ts''} + [X^{ar}\hat{W}^{a'}]_{ss''}\delta_{ts})[(\hat{W}^+)^2]_{ss''} - [\hat{W}^a\hat{W}^{a'}]_{ss''}(\hat{W}^+_{st} X^{+r}_{ts''} + [X^{+r}\hat{W}^+]_{ss''}\delta_{ts}) \tag{38}$$

$B^{aa'}(z)$ is a local parametrization of B, and if we drop the $O(z^2)$, it parameterize its tangential hyperplane at $\hat{B}^{aa'}$. Its dimension d_B is the rank of C interpreted as a $k^2d^2 \times d_J$ matrix in $aa'ss'' \times rt$. Again, d_B may be smaller than d_W, since C may not be full rank. for $r \in \{1 : d_J\}$ spans the tangential space of rescaled variety $B^{aa'}$ at $\hat{B}^{aa'}$. Again, they may not be linearly independent, If $[(W^+)^2]_{ss''} = 0$, then $B^{aa'}_{ss''} = \perp\ \forall aa'$, hence all such ss'' should be ignored in $C^{aa'r}_{ss''}$, but since the corresponding rows in C are 0, they don't contribute to the rank anyway.

Sampling estimate of d_B. A simpler, but less elegant, and more fragile method to estimate d_B is as follows: Fix one solution $\hat{J}$. Add random noise in direction of the null-space spanned by Y^r so that it stays a solution, i.e. compute $J = \hat{J} + \sum_r Y^r z_{\cdot r}$ for random z, and from this, W and $B^{aa'}$ for many such random J. The resulting point cloud spans covers the solution variety $B^{aa'}$. Various tools could be used to analyze this point cloud, e.g. determine its dimension. If z is chosen small, the point cloud concentrates around $\hat{B}^{aa'}$ and forms a near-linear space, whose dimension d_B can easily be determined by PCA.

Higher-order B and higher i. In the same way we can derive the solution dimensions $d_{B^{\cdots}}$ for higher-order $B^{\cdots}$. Also, even though we don't have (yet) an efficient algorithm for solving EqIM(i) for $i > 1$ if the solution is not unique, we still can determine the dimension of the solutions (at a particular point $\hat{W}$). Algorithmically already covered is the case of W satisfying EqIM(1)∧EqIM(2), whose solution dimension turns out to be $d_W - d_B$. The general procedure is to plug $W = \hat{W} + X$ into and linearly expand EqIM(i) for i we to hold. Together they form a system of linear equations whose solution dimension can be determined by SVD as above.

Q Counter-Examples in Related Work

In Section 3 we presented a counter-example to questions (i,iii,v). Question (i) (i.e. Can M be inferred from $B^a := M^a \oslash M^+$?) has been implicitly addressed in previous work. In [EMK+22, App.A.3] the authors present a counter-example to the claim that a state representation constructed via an inverse model (i.e. two states have the same representation iff they yield the same inverse distribution for all of their possible successor states) is sufficient for representing a set of policies that differentially visit all states[1]. This fails whenever two states are aliased by the inverse model.

Note that this failure of state representation learning implies a negative answer to our question (i), as W would differ from M on these aliased states. Unlike our counter-example, theirs involves deterministic forward dynamics, and therefor buttresses our claims by showing that M cannot always be inferred even in this simpler case. Similar to our counter-example in Section 3, [MHKL20] proposes a stochastic counter-example to inverse modeling for state representation learning.

In general, the transferability of these counter-examples suggests a strong relationship between literature on using single-step inverse models for state representation learning and using them for inferring the forward model. It is an interesting open question whether or not algorithms for representation learning on the basis on multi-step inverse models (like those put forward in [EMK$^+$22]) might be used to shed light on the questions put forward here and vice versa.

R Relevance for Planning

In Section 1, various streams of applied work were highlighted; here we focus on spelling out the overarching impact that compositional inverse models (an affirmative answer to question (iv)) would have for planning problem.

Many forms of planning involve the evaluation of candidate i-step action sequences (e.g. model predictive path integral control [WDG$^+$16]). Ideally, all possible action sequences would be evaluated, but as the space of i-step action sequences grows exponentially in i, this is often intractable.

Access to the i-step inverse distribution $p(a...a^i|s...s^{i+1})$ allows determining the subset of action sequences that likely reach state s^{i+1} post-execution (e.g. those whose probability is above some threshold). It is often the case that only action sequences that are distinguishable in this way are of interest (e.g. goal-reach tasks), thus access to an inverse model of the appropriate horizon allows for filtering candidates. This filtering method is a particularly appealing approach when the cost/reward function is initially unknown and frequently changes.

While this idea has already seen scalable implementations [MJR15], these rely on short, fixed horizons since they directly learn all inverse models of step-size up to the horizon, which is data-inefficient for large horizons. If inverse models could be composed, then longer, variable horizons could be used while only learning a short horizon inverse model by inferring the longer horizon models as needed. Our work shows that this is possible, but with exceptions and a more practical composition algorithm being outstanding.

[1]Technically, as per their Definition 2, this 'policy cover' need only account for all 'endogenous' states. But omit the 'exogenous' states from their counter-example and it can be seen to address our question (i).