

When Are Discovery Problems Equivalent or Transferable?

What must survive the move from one problem to another?

Complexity, Information, and Cross-Domain Transfer

Four meanings of transfer

Can a result from one laboratory guide an intervention in another? A reusable solver, an accurate predictor and a transported causal response need different warrants.

We first prove what a computational reduction preserves, then follow two bacterial strains across two laboratories to separate prediction from intervention.

A navigation map for the three formal bridges

Bridge	Map and preserved object	Assumption / failure
Reduction	Encode instances; preserve yes/no	Correct map; overhead may erase a gain
Adaptation	Relate populations; bound target risk	Shared labeling; label flip breaks it
Transport	Relate causal models; identify effect	Invariant mechanism; change breaks it

A reused tool has no automatic guarantee from these rows. Its target-domain performance needs separate evidence.

A polynomial-time many-one reduction

$$x \in A \iff f(x) \in B; \quad f \text{ computable in } \text{poly}(|x|)$$

For decision languages A and B , a reduction maps each input x to $f(x)$ in polynomial time while preserving yes/no membership.

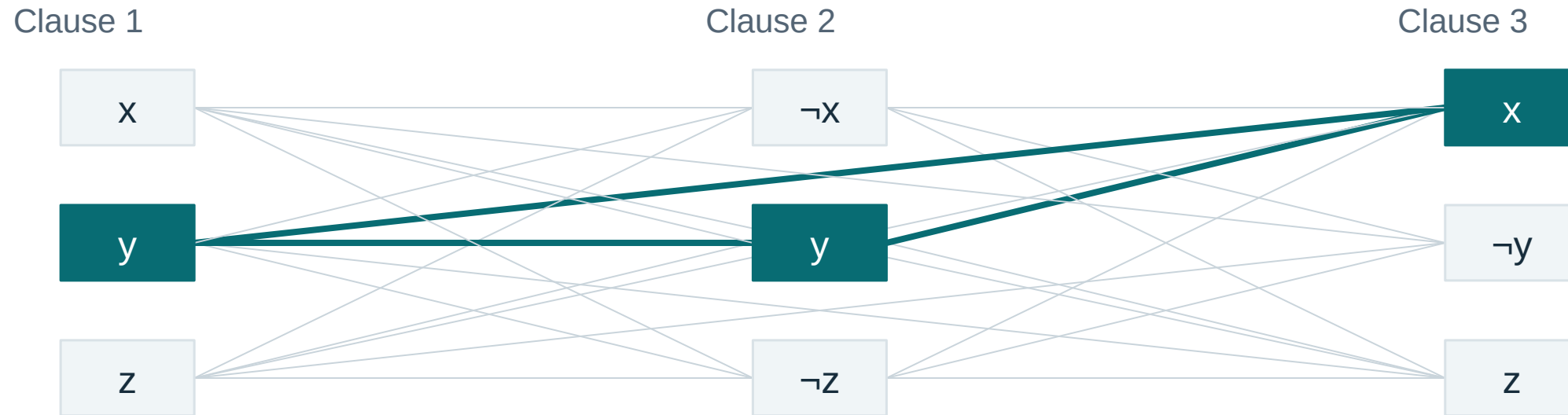
If B has a polynomial-time solver, compose it with f to solve A . The direction matters.

Constructing a reduction: 3-SAT to clique

Three-literal Boolean satisfiability (3-SAT) asks whether a formula with m three-literal clauses can be satisfied. Create a vertex for each literal occurrence; join occurrences in different clauses unless they contradict.

Ask whether the resulting graph has a clique of size m .

A nine-vertex reduction you can inspect



21 edges; teal highlights a consistent 3-clique.

Take $(x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z)$. Create one vertex per literal occurrence. Join occurrences from different clauses unless their literals contradict. The highlighted occurrences y, y, x form a 3-clique.

Why a satisfying assignment gives a clique

satisfying assignment $\rightarrow$ m mutually adjacent vertices

If the formula is satisfiable, choose one true literal occurrence from each clause.

The m selected vertices come from distinct clauses and cannot contain contradictory literals, so every pair is connected.

Why a clique gives a satisfying assignment

An m -clique contains at most one vertex per clause because no same-clause edges exist. With m clauses it therefore contains exactly one per clause.

Its literals contain no contradictory pair. Set them true and extend the remaining variables arbitrarily; every clause is satisfied.

A reduction can be costly at useful sizes

$$T_{\text{transfer}}(n) = T_{\text{encode}}(n) + T_B(f(n)) + T_{\text{decode}}(n)$$

If input size n becomes n^3 and the target solver takes time m^2 , the composed running time can scale as n^6 .

Polynomial equivalence does not imply equal constants, equal instance difficulty or equal scientific cost.

A reduction preserves the decision, not the cost

A yes/no reduction need not preserve approximation ratios, solution quality, probability distributions or noise.

For optimization or learning, a useful transfer theorem needs a mapping that preserves the quantity being claimed.

Encoding a scientific task as a program is not a solution

A molecular evaluator can be implemented in software, but writing its interface does not create a reliable measurement oracle.

The computational representation may hide the hardest part: acquiring data that distinguish scientifically relevant outcomes.

Keep feature marginals and labels separate

Let P_S and P_T be joint laws of (X,Y) , and D_S, D_T their feature marginals. A binary classifier h has risk $\varepsilon_S(h) = P_S[h(X) \neq Y]$.

Unlabeled target observations identify D_T , not $P_T(Y|X)$. An aligned feature distribution can still hide reversed labels.

Route: decision-preserving reductions are complete; now the object is target predictive risk.

The domain-adaptation bound exposes two requirements

$$\varepsilon_T(h) \leq \varepsilon_S(h) + \frac{1}{2} d_{H\Delta H}(D_S, D_T) + \lambda$$

For binary H , $\lambda = \inf_{h \in H} [\varepsilon_S(h) + \varepsilon_T(h)]$ is the best joint labeling error.

$d_{H\Delta H} = 2 \sup_{h, h' \in H} |D_S[h \neq h'] - D_T[h \neq h']|$. This is a population bound; estimated versions need sampling terms.

Zero feature discrepancy can hide total label failure

Joint outcome (X,Y)	Source probability	Target probability
(0,0), (1,1)	.5 each	0 each
(0,1), (1,0)	0 each	.5 each

Let $X \in \{0,1\}$ with probability $\frac{1}{2}$ each in both domains. In the source $Y=X$; in the target $Y=1-X$. For $h(x)=x$: $\epsilon_S=0$ and $\epsilon_T=1$. For $H=\{x,1-x\}$, feature discrepancy is zero and $\lambda=1$, so the bound is tight.

Covariate shift is a stronger assumption

$$RT(h) = ES[(p_T(X)/p_S(X)) \ell(h(X), Y)]$$

Under $PS(Y|X) = PT(Y|X)$, only the input marginal changes. If target support is covered by source support, importance weights can express target risk using source data.

Large or unstable weights can make estimation unreliable even when the identity is valid.

A numerical support-shift example

Suppose a rare source region has probability 0.01 but target probability 0.50. Its importance weight is 50.

Only about ten of 1,000 source observations fall there on average, so the target-relevant estimate can remain noisy.

Why suspect a population change rather than a failed mechanism?

Suppose a source laboratory reports a treated response of 0.50 and a target reports 0.35. A lower average need not mean the intervention stopped working.

If strains respond differently, changing their mixture changes the average even when each response mechanism is stable. Recording strain can make transport conceivable—but stability still needs justification.

Pool, repeat, or transport the intervention evidence?

Candidate	When it could be justified
Use the source average	Same relevant mixture and response mechanisms
Run target interventions	Access and budget permit fresh identification
Reweight source strata	Target mix known; within-stratum responses stable

These options trade experimental access against assumptions. Reweighting is a candidate only when relevant strata overlap and their responses transport.

A transport calculation with two strata

source average = 0.50; transported target = 0.35

In a constructed two-laboratory study, strains A and B respond to an intervention with probabilities 0.2 and 0.8. A 50:50 source mixture gives 0.50.

A target mixture of 75:25 gives 0.35 if within-strain intervention responses are unchanged. This is a response probability, not a treatment effect.

A transported response is not yet a treatment effect

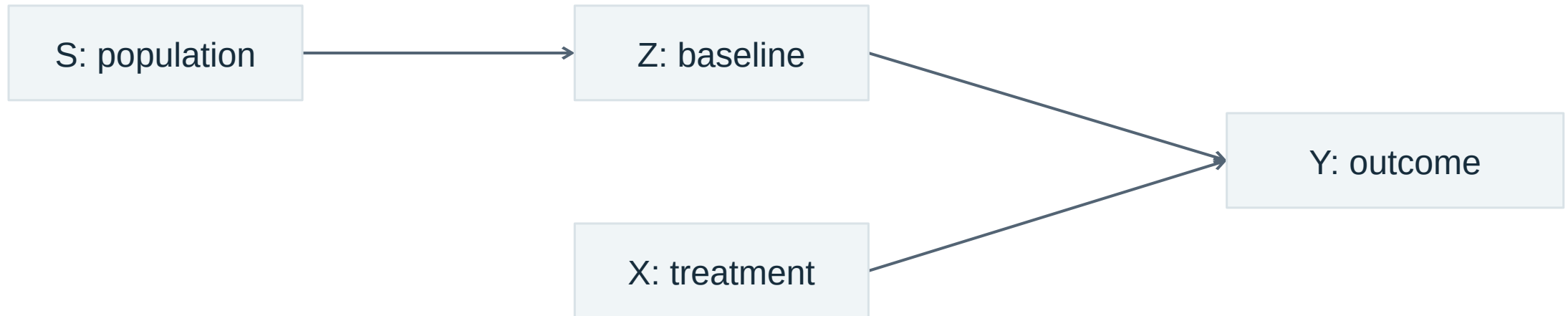
$$\text{Target effect} = \sum_z P_{\top}(z) [r_1(z) - r_0(z)] = 0.175$$

Let r_1 and r_0 be treated and untreated response probabilities within strain z . Suppose r_0 is 0.1 for A and 0.4 for B, also invariant across laboratories.

The target untreated mean is 0.175. Its treatment effect is $0.35 - 0.175 = 0.175$; the source effect is $0.50 - 0.25 = 0.25$.

A selection diagram states what may change

Only Z receives a selection arrow.



S marks population selection: $S \rightarrow Z$ allows the distribution of baseline covariate Z to change. X and Z cause Y; no selection arrow enters Y. With invariant conditional response and target support covered, source experiments can be reweighted by target Z frequencies.

A simple transport formula

$$PT(y|do(x)) = \sum_z PS(y|do(x),z) PT(z)$$

Suppose Z is a pre-intervention covariate, target support is covered, and the conditional intervention response is invariant across populations.

Average the source stratum-specific experimental response using the target distribution of Z .

The same weights fail when the mechanism changes

Keep the target strain mixture at 75:25. If target treated responses are actually 0.5 in both strains, the true target mean is 0.50, not 0.35.

A target calibration sample can test these responses. More precise estimates of the mixture cannot repair the wrong within-strain mechanism.

Adaptation has a break-even point

$$\text{net saving} = Ts - F$$

Let F be fixed adaptation cost and s the saving per target task, after charging validation. Transfer pays over T tasks when $Ts > F$.

If s changes with target difficulty or deteriorates under drift, a single threshold is inadequate.

A reusable tool still needs a new warrant

Transfer	What carries over?	What needs new evidence?
Construction → program	An explicit object encoding	Generality beyond one instance
Program → molecule	A search architecture	Noisy physical validation
Molecule → mechanism	An intervention tool	Specificity and causal controls

Transfer costs and failure cases

Warrant	Cost to include	Concrete failure
Reduction	Encode, solve, decode	Polynomial overhead dominates
Risk bound	Labels and adaptation	Same X; labels flipped
Transport	Target strata and trial	Changed response within strata
Method reuse	Setup and confirmation	No target-domain benefit

Open question: which structure predicts transfer?

Possible predictors include shared constraints, aligned semantics, invariant mechanisms and reusable abstractions.

Embedding proximity or a common natural-language description is weak evidence unless it predicts performance under a specified target distribution.

Conclusion: the bridge must match the claim

Solution procedures, predictions and causal effects transfer under different conditions. Adaptation and validation are part of the cost.

Next: compare proofs, programs, molecules and mechanisms without erasing the differences that determine difficulty.

References

Stephen A. Cook (1971). The Complexity of Theorem-Proving Procedures. STOC 1971, 151–158; conference paper

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira and Jennifer Wortman Vaughan (2010). A Theory of Learning from Different Domains. Machine Learning 79, 151–175; journal article

Judea Pearl and Elias Bareinboim (2014). External Validity: From Do-Calculus to Transportability Across Populations. Statistical Science 29, 579–595; journal article; arXiv posted 2015

Richard M. Karp (1972). Reducibility Among Combinatorial Problems. Complexity of Computer Computations, 85–103; research chapter

References · continued

Quentin Berthet and Philippe Rigollet (2013). Computational Lower Bounds for Sparse PCA. arXiv:1304.0828v2, 26 April 2013; primary preprint as inspected

Sumit Gulwani, Oleksandr Polozov and Rishabh Singh (2017). Program Synthesis. Foundations and Trends in Programming Languages 4, 1–119; survey monograph

Kevin Ellis et al. (2021). DreamCoder: Bootstrapping Inductive Program Synthesis with Wake-Sleep Library Learning. PLDI 2021; conference paper; preprint has earlier title

Jonas Peters, Peter Bühlmann and Nicolai Meinshausen (2016). Causal inference by using invariant prediction: identification and confidence intervals. Journal of the Royal Statistical Society B 78, 947–1012; journal article

References · continued

Bernardino Romera-Paredes et al. (2024). Mathematical discoveries from program search with large language models. Nature 625, 468–475; online 14 December 2023

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao and Adam Zsolt Wagner (2025). Mathematical exploration and discovery at scale. arXiv:2511.02864v1; preprint, 3 November 2025

Joseph L. Watson et al. (2023). De novo design of protein structure and function with RFdiffusion. Nature 620, 1089–1100; journal article

Martin Pacesa et al. (2025). One-shot design of functional protein binders with BindCraft. Nature 646, 483–492; online 27 August 2025

References · continued

Claude E. Shannon (1948). A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423, 623–656; journal article

David Blackwell (1953). Equivalent Comparisons of Experiments. Annals of Mathematical Statistics 24(2), 265–272; published primary research