Hopf algebras in ML

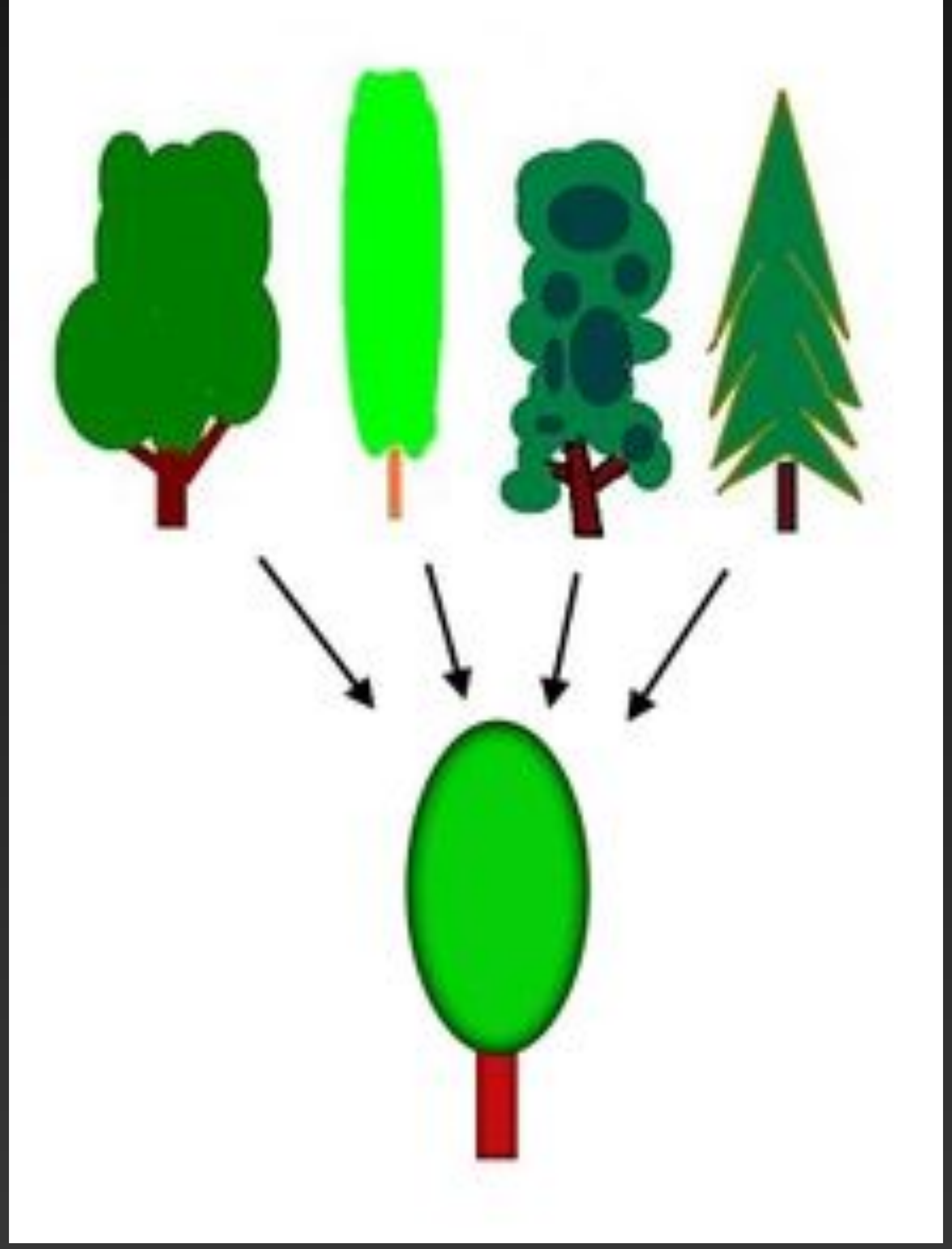
Adam Nemecek

Machine learning: Field in crisis

- machine learning is broken
- more GPUs approach is hitting diminishing returns
- a fundamental change is needed, new math
- language built from ground up for “interaction modeling”
- machine learning is just one of many
- convolution: ultimate operation
- unification: ad-hoc architectures
- geometric interpretation
- work in progress

Machine learning: Crisis averted

- partitioning to maximize symmetry via convolution
- inverse problem: using observations to infer parameters characterizing the system
- architecture imposes duality between weights and activations Achille2017
  - architecture rewriting is needed
- ML without rewriting is like a binary tree where you can't insert children



Proposed solution: Convolutional relation

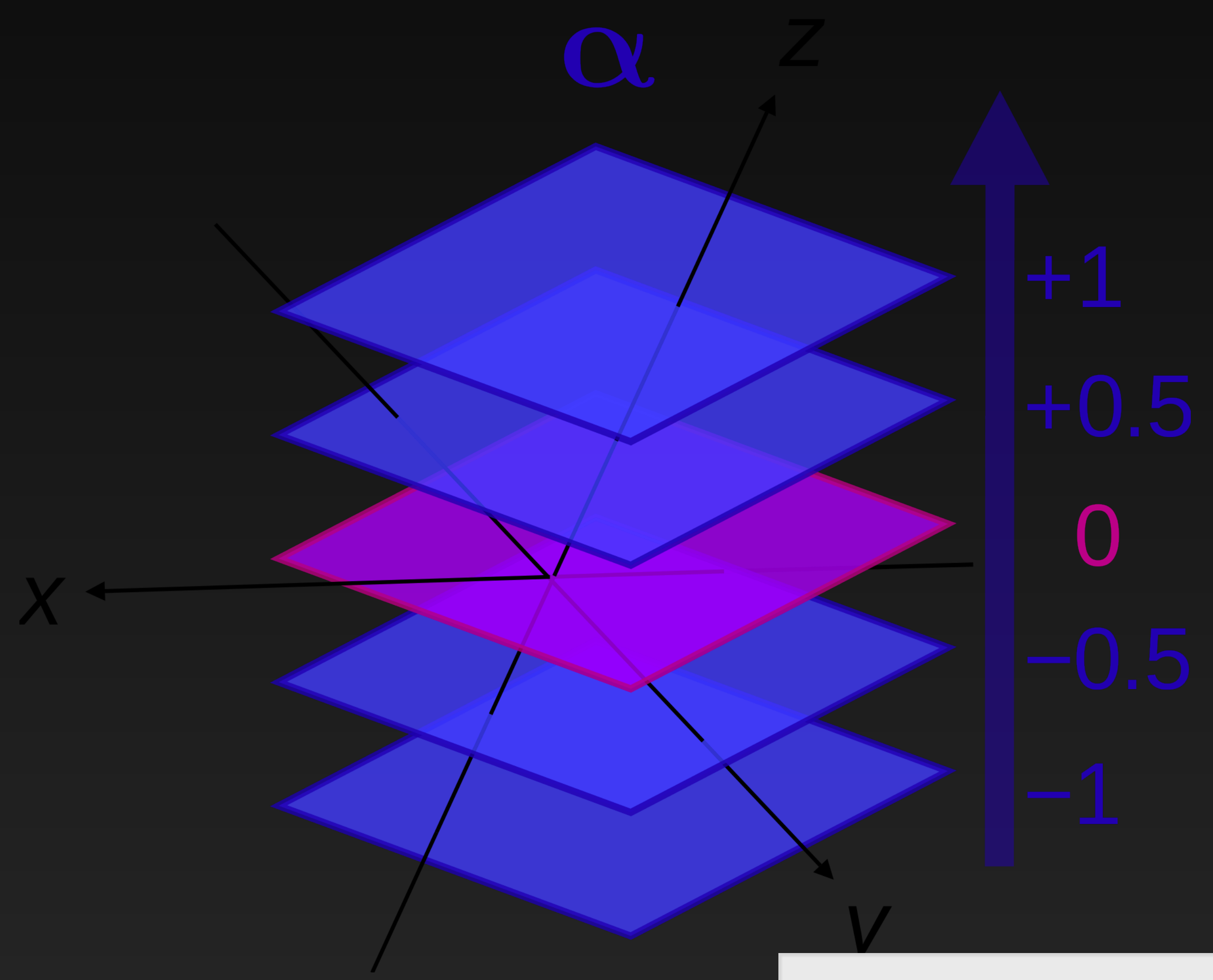
- convolution: $conv(a, b) = y$, find y
- convolutional equation: $conv(a, x) = c$, find x
- convolutional relation: $conv(a, x) = y$, find x and/or y
- braid theory, polytopes, Hopf algebra
- new programming language!

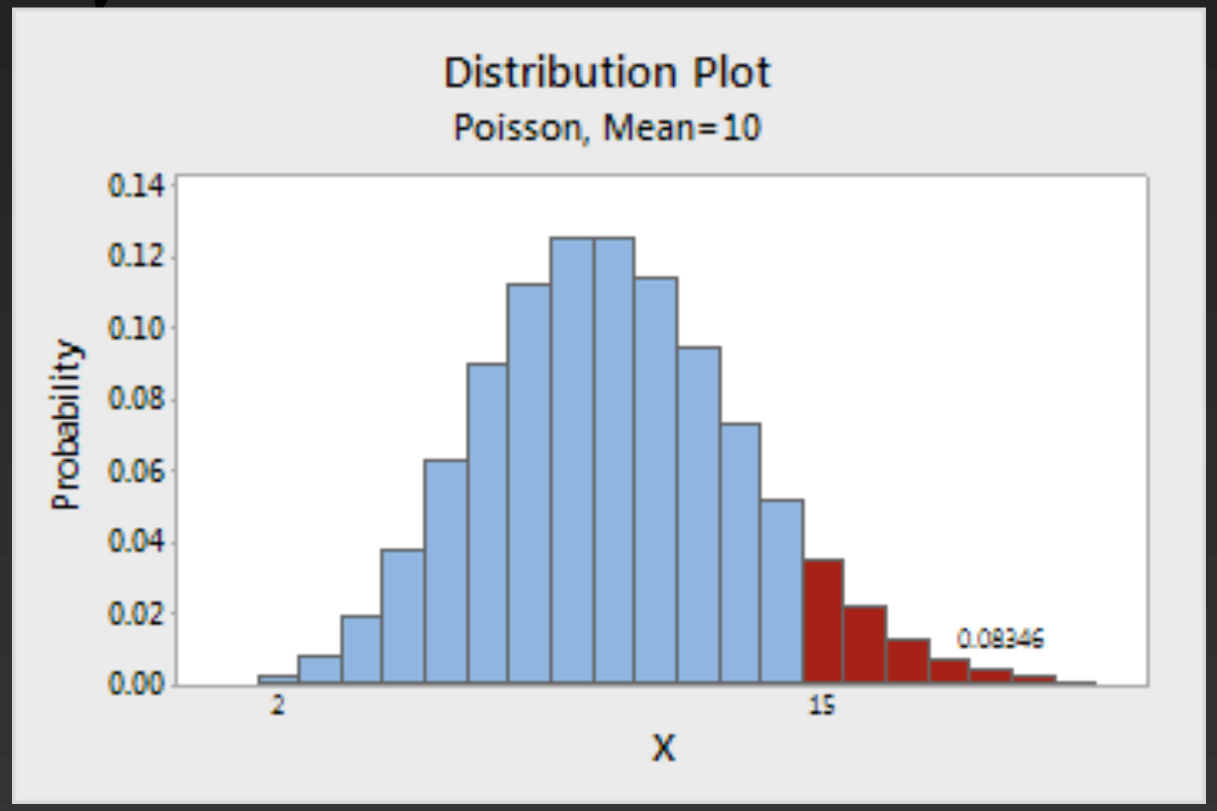
Duality 1: Introduction

- involution: set complement, reflection, conjugate transpose
- Chu duality
  - duality between state and event
  - duality between program and state

Duality 2: Linear forms

- covector, distribution, linear functional
- row vectors
- measures
- level sets: set of hyperplanes
- pointwise operations (explain)
- Winitzky: Linear Algebra via Exterior Products





Duality 3: Adjoint methods, backprojection

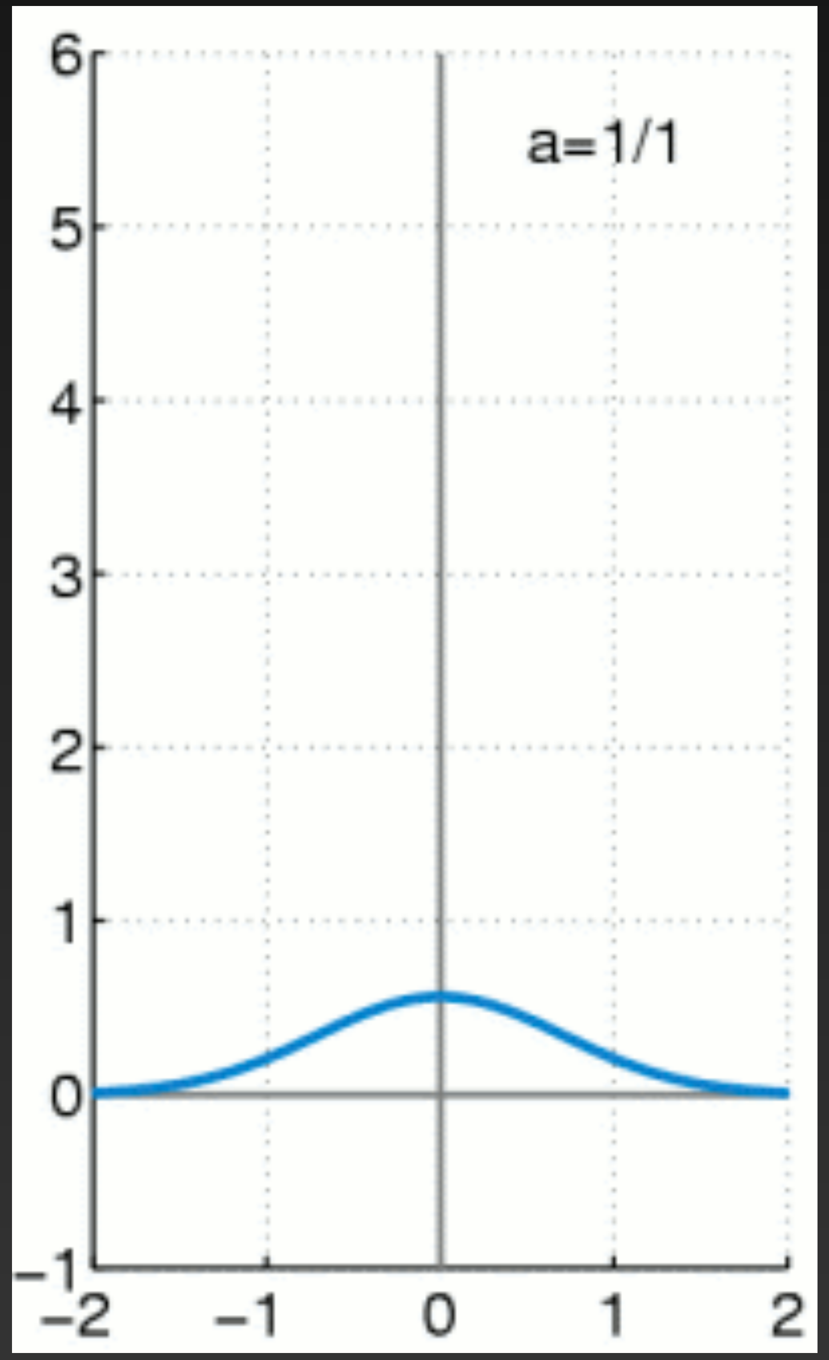
- dual
  - similar to inverse
  - sometimes dual is inverse (unitary matrices)
  - [Claerbout]
- preserves metric and Levi-Cevita connection [Moore2017]
- cheaper than inverse
- less sensitive perturbations, small changes won't throw it off
- nonlinear problems (neural networks)
- backpropagation Radul2022
- backprojection
  - computer graphics: abstract description -> image
  - computer vision: image -> abstract description

Convolution 1: Overview

- convolution is ubiquitous
- Hopf convolution
- input: function/distribution and distribution
- output: new function, meaning out output depends on context
- Hyena architecture [Poli2023]
  - convolution with gating, right but not enough
- dual of correlation
- shift and reflection

Convolution 2: Dirac delta

- Dirac delta δ is the convolutive unit $conv(f, \delta) = f$
- Gaussian distribution with 0 variance



Convolution 3: code

- educational, do not use this
- linear vs cyclical convolution (cconv)
- Dirac delta = `[1, 0, ...]`

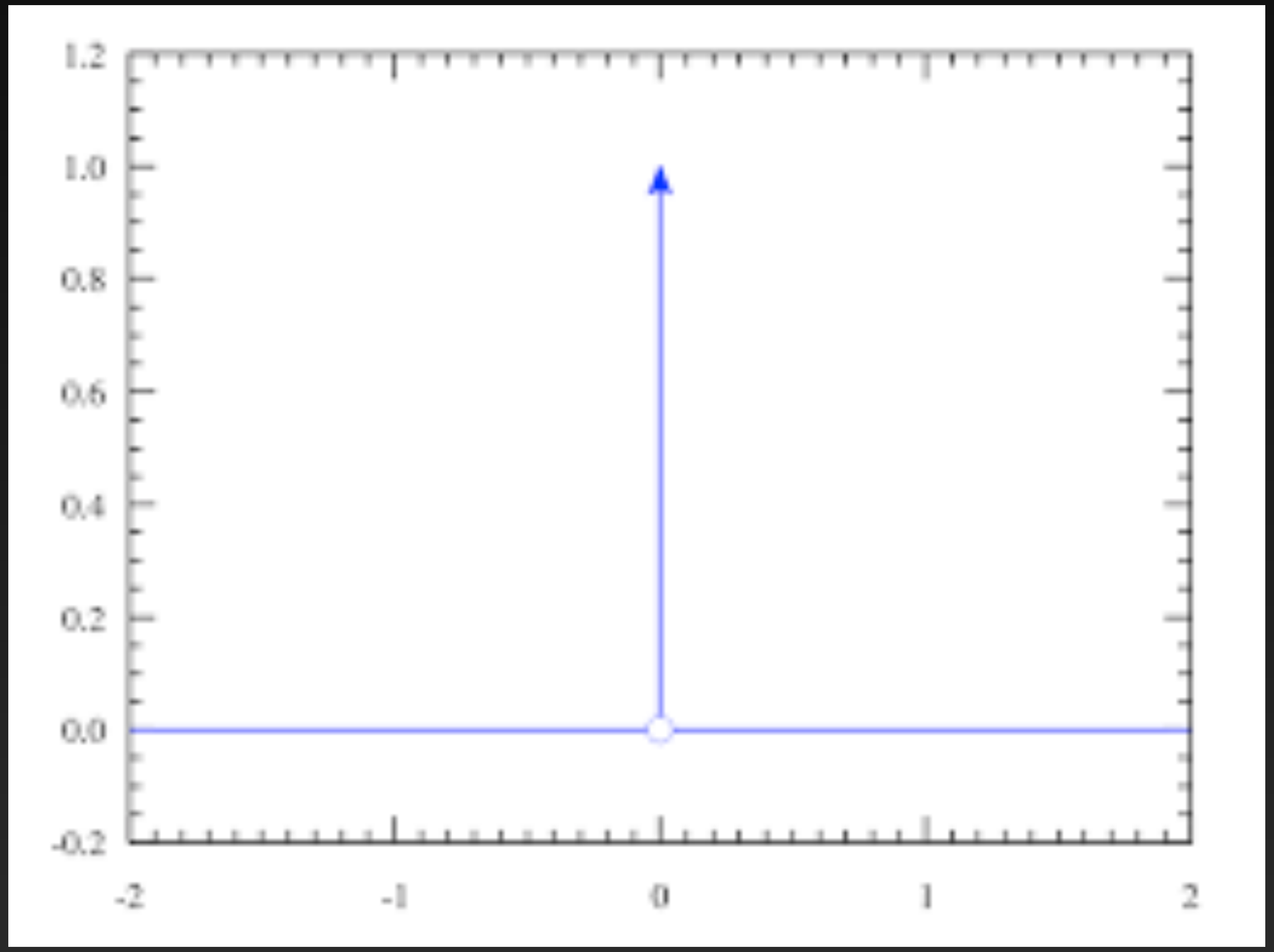
```python
def conv(a: List[float], b: List[float], l: Optional[int] = None):
    l = l or len(a) + len(b) - 1
    ret = [0] * l
    for (i, e) in enumerate(a):
        for (j, f) in enumerate(b):
            ret[(i + j) % l] += e * f
    return ret

def corr(a: List[float], b: List[float], l: Optional[int] = None):
    return conv(a, b[::-1], l)
```

Convolution 4: Meaning

- polynomial multiplication
- moving average
- interpolation
- …and more

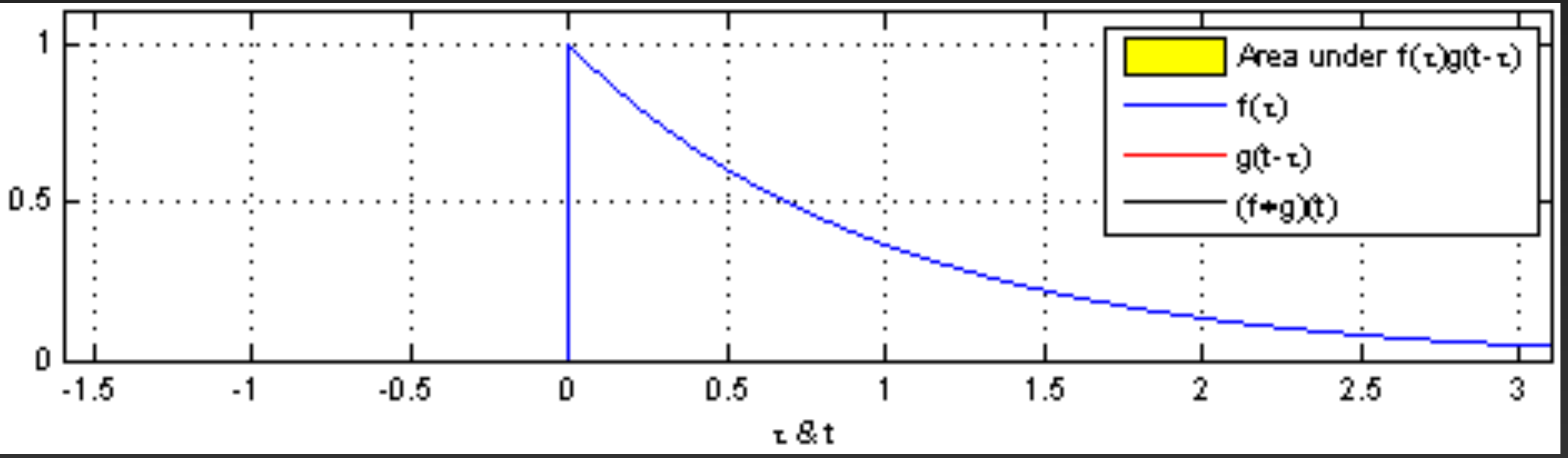
Convolution 5: As evaluation

- sifting property
  - $$\int f(x)\delta(x - T)dT = f(T)$$
  - evaluation < averaging
  - Dirac delta is a Gaussian distribution with 0 variance



Convolution 6: As function composition

- composition of convolution systems corresponds to
  - multiplication of transfer functions
  - convolution of impulse responses [Boyd2003]
- composition is evaluation [Lawvere2009]
  - polynomial composition is substitution and evaluation
- https://ncatlab.org/nlab/show/internal+hom



Convolution 6: Convolution theorem

- $conv(a, b) = ifft(fft(a)\ .*\ fft(b))$
- Fourier transform projects a function onto a U(1) (unit circle)
  - dual space: pointwise computation
- Fourier transform diagonalizes the convolution operator
  - certain points characterize the function
- Pontryagin duality: reconstruct signal from Fourier
- Tannaka-Krein: non-commutative Pontryagin

Convolution 7: Correlation 1

- correlation gives you a notion of similarity and as a result clustering
- dual to convolution
  - convolution: combining, correlation: partitioning
- for even function convolution and correlation are the same
- geometric interpretation:
  - two correlated variables will be scattered along hyperplane [Han2022]
- Hodge star in 3D

Convolution 8: Correlation 2

- geometric interpretation
- correlation converts points to hyperplanes and vice versa
- fixed points: in Fourier, these points are "boring"
- https://en.wikipedia.org/wiki/Correlation_(projective_geometry)
- https://en.wikipedia.org/wiki/Partial_correlation
- perspective = equivariance

> Of course this transference of theorems from forms of one kind to forms of another is possible only for theorems on points, lines, and planes that involve only properties remaining unchanged under a projective correlation. Among such properties we a ready have discussed 1) incidence and 2) cross ratio. In view of (23), (24), and (24′), incidence must be regarded as a symmetric relation, i.e., the statement "*P* and *g* are incident"[41] means not

Convolution 9: Shift & reflection (1)

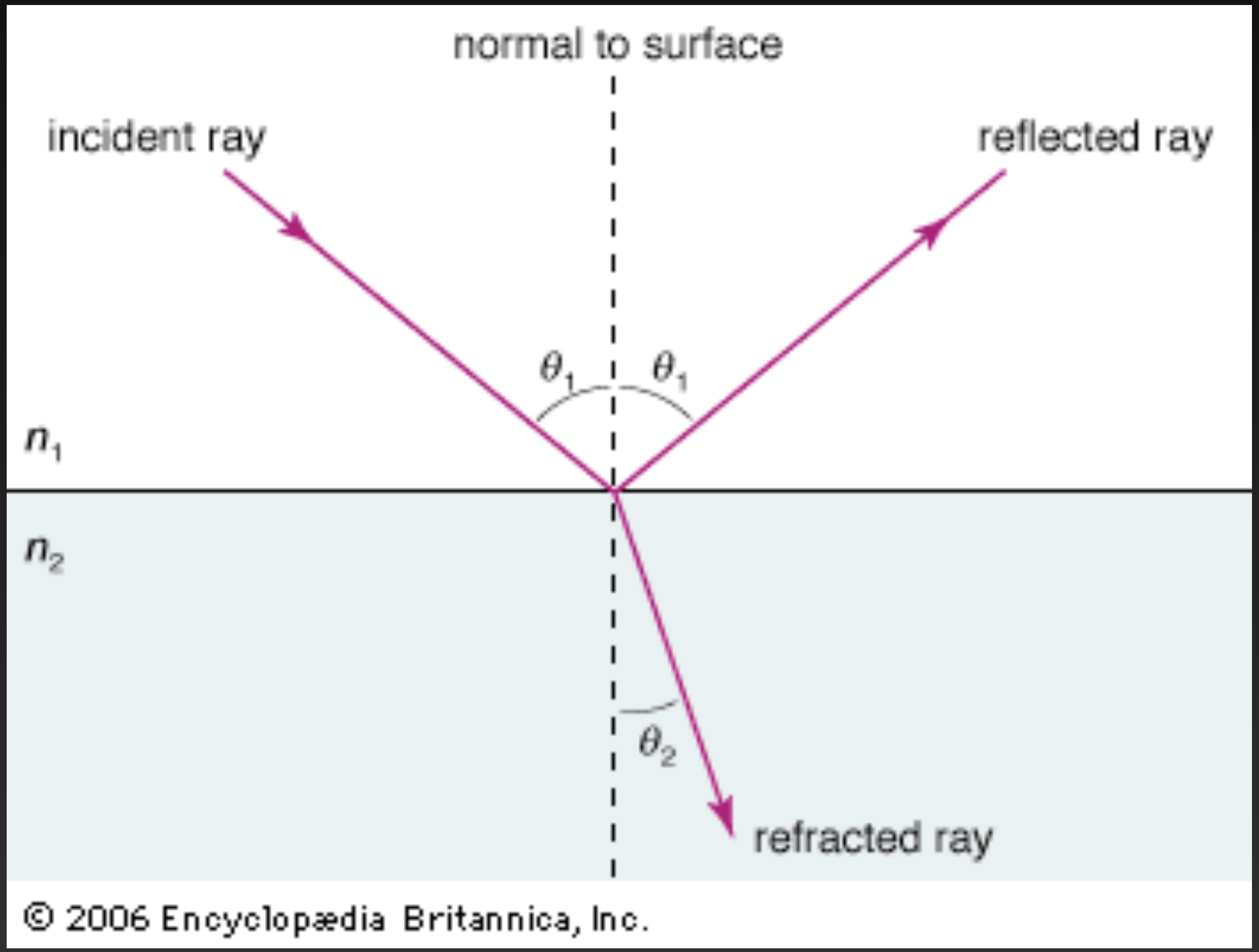
- shift operator: $Df(x) = f(x + t)$
  - transfer operator, pushforward
  - linear
  - Taylor series
- duals
- system description
- finding hyperplanes by shifting

- reflection: $f(x) = f(a - x)$
  - composition operator, pullback
  - antilinear
  - time-reversal
  - rotation, translation = 2 refl's

When the transfer operator is a left-shift operator, the Koopman operator, as its adjoint, can be taken to be the right-shift operator. An appropriate basis, explicitly manifesting the shift, can often be found in the orthogonal polynomials. When these are orthogonal on the real number line, the shift is given by the Jacobi operator.[5] When the polynomials are orthogonal on some region of the complex plane (viz, in Bergman space), the Jacobi operator is replaced by a Hessenberg operator.[6]

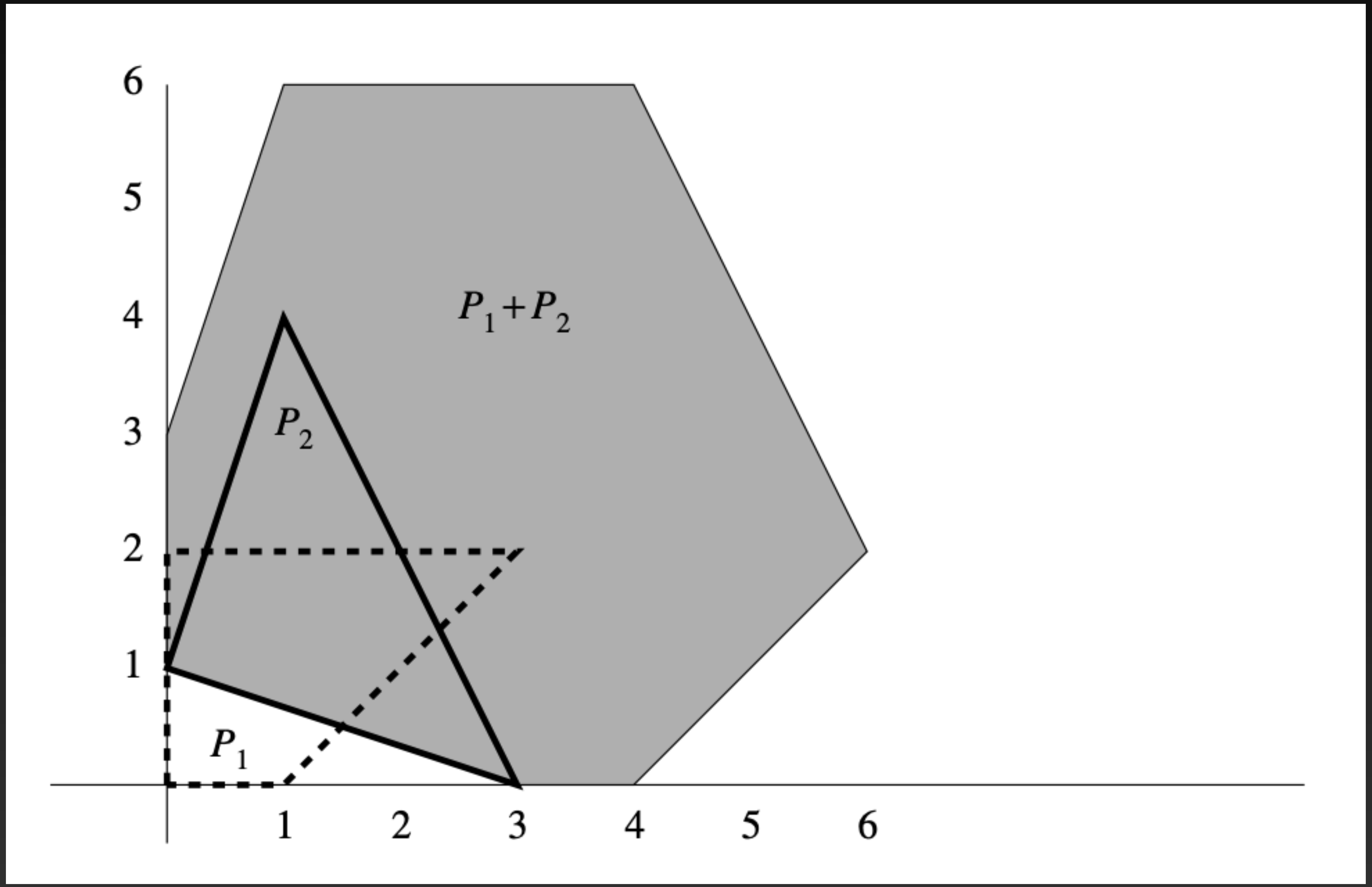
Convolution 10: Shift & reflection (2)

- physical interpretation
  - shift: emitting a photon
  - reflection: boundary condition
- Boolean negation is a reflection
  - $neg(x) = 1 - x$
- QR algorithm has a reflection



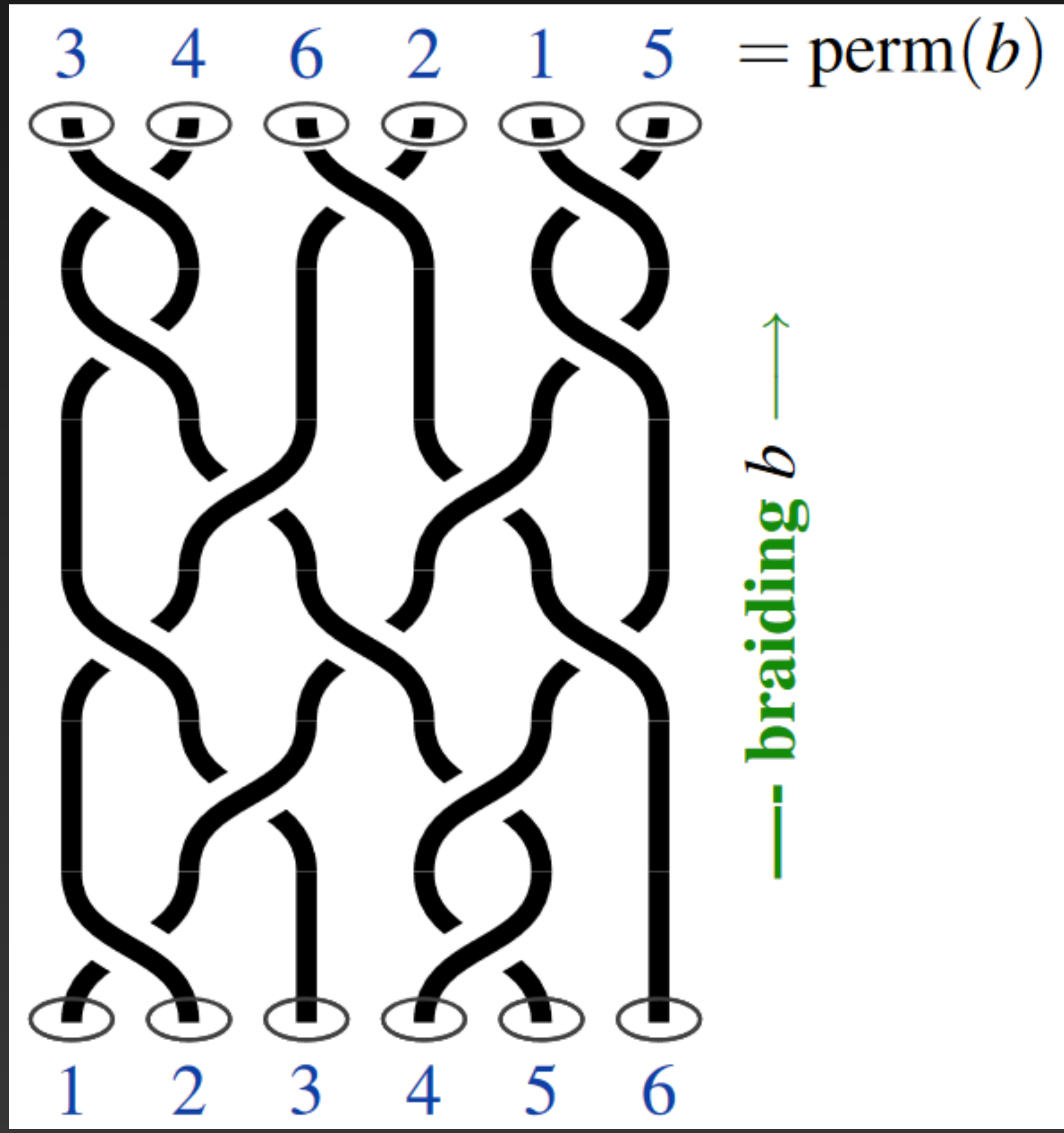
Convolution 11: Minkowski sum

- convolution
- used in robotics (image)
- related to Minkowski product from spacetime fame



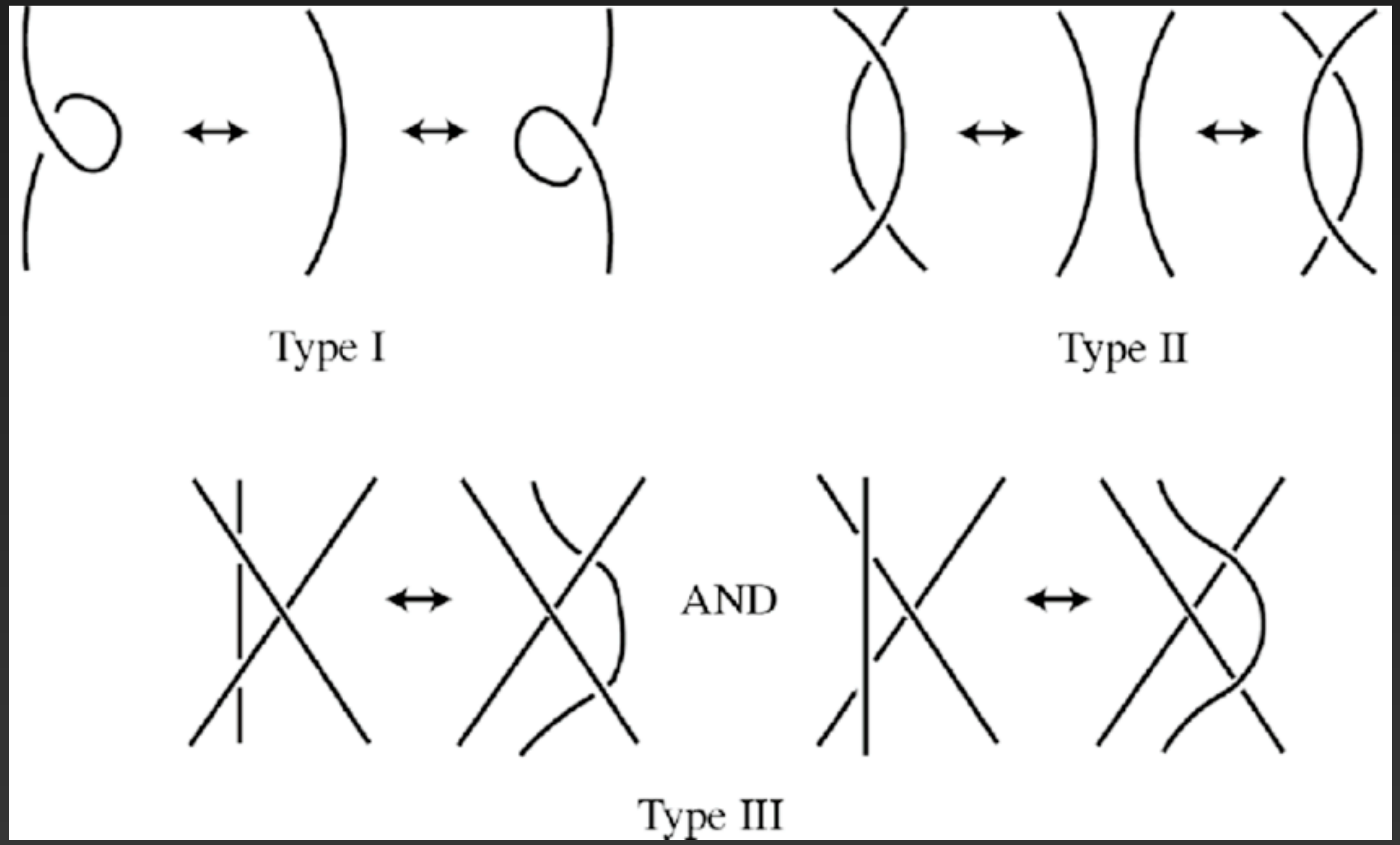
Braid theory 1: Introduction

- a generalization of permutation
- particle interactions & dynamics
- permutation with memory [Caulton2021]
- diagrams as programs [Pavlovic2023]
  - variables are strings
- defines evaluation (measure) and coevaluation map (state)



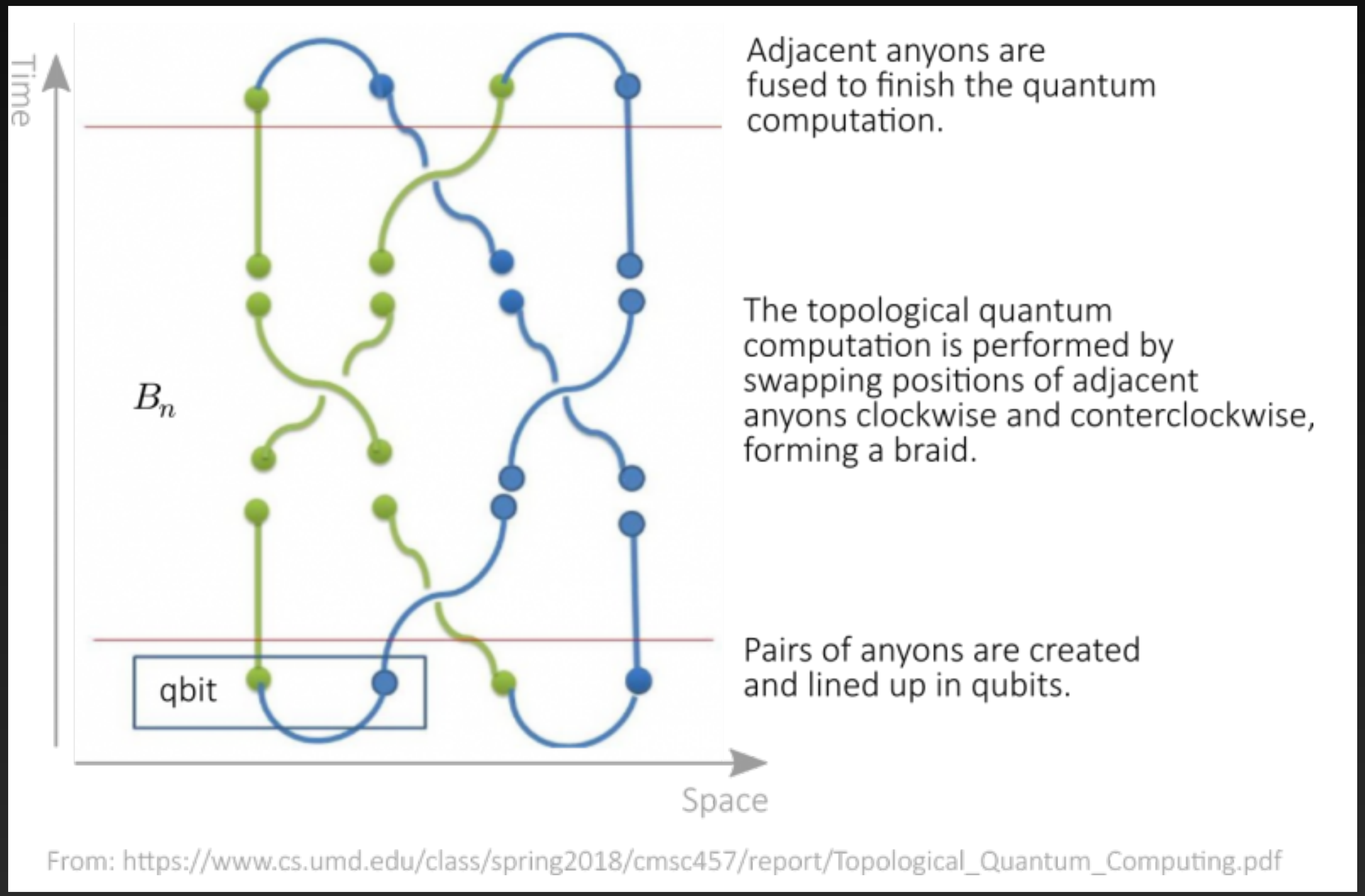
Braid theory 2: Reidemeister move

- equivalence relation
  - involution is an equivalence class by conjugation
  - braids are equivalence relations by Reidemeister moves



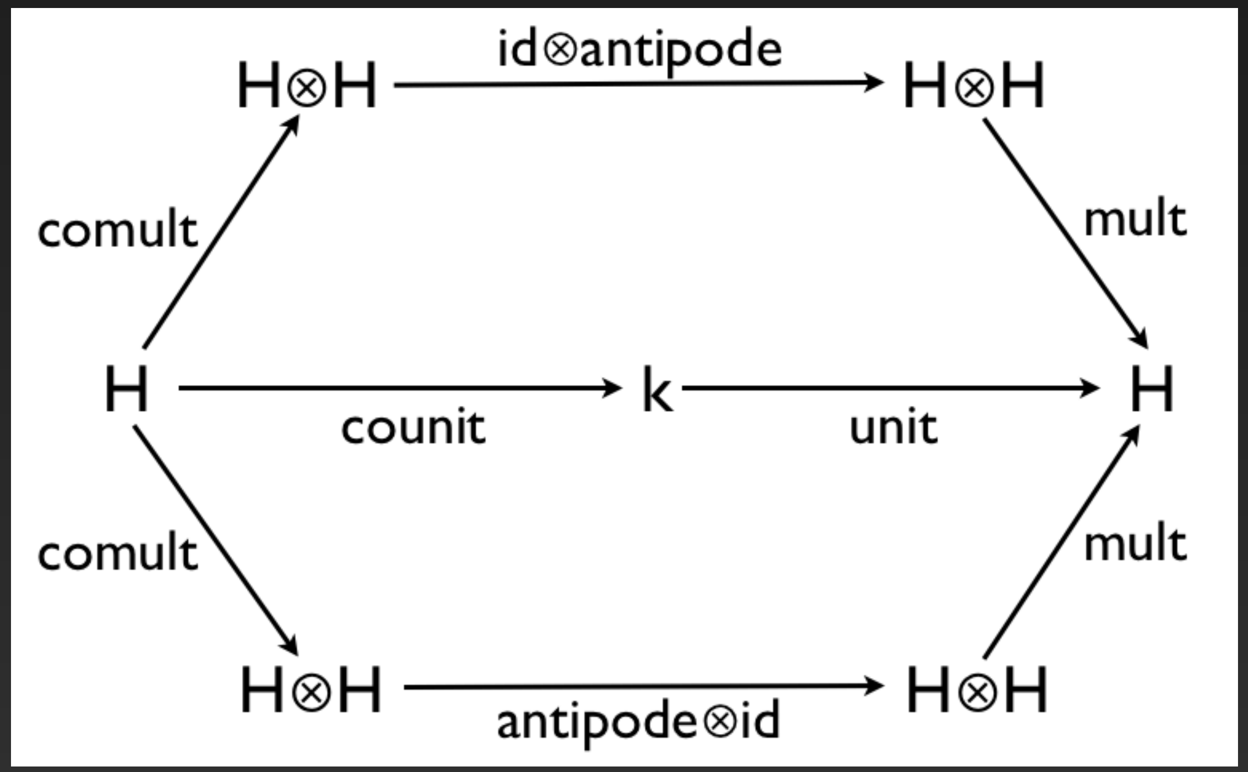
Braid theory 3: Computation

- Topological quantum computation
- Reidemeister moves
  - cut rule [link]
  - cut rule are computation
- cap: evaluation, cup: coevalution
- braiding: program
- braid matrices can be interpreted as logic gate [Kauffman2004]
- Kasirajan2021



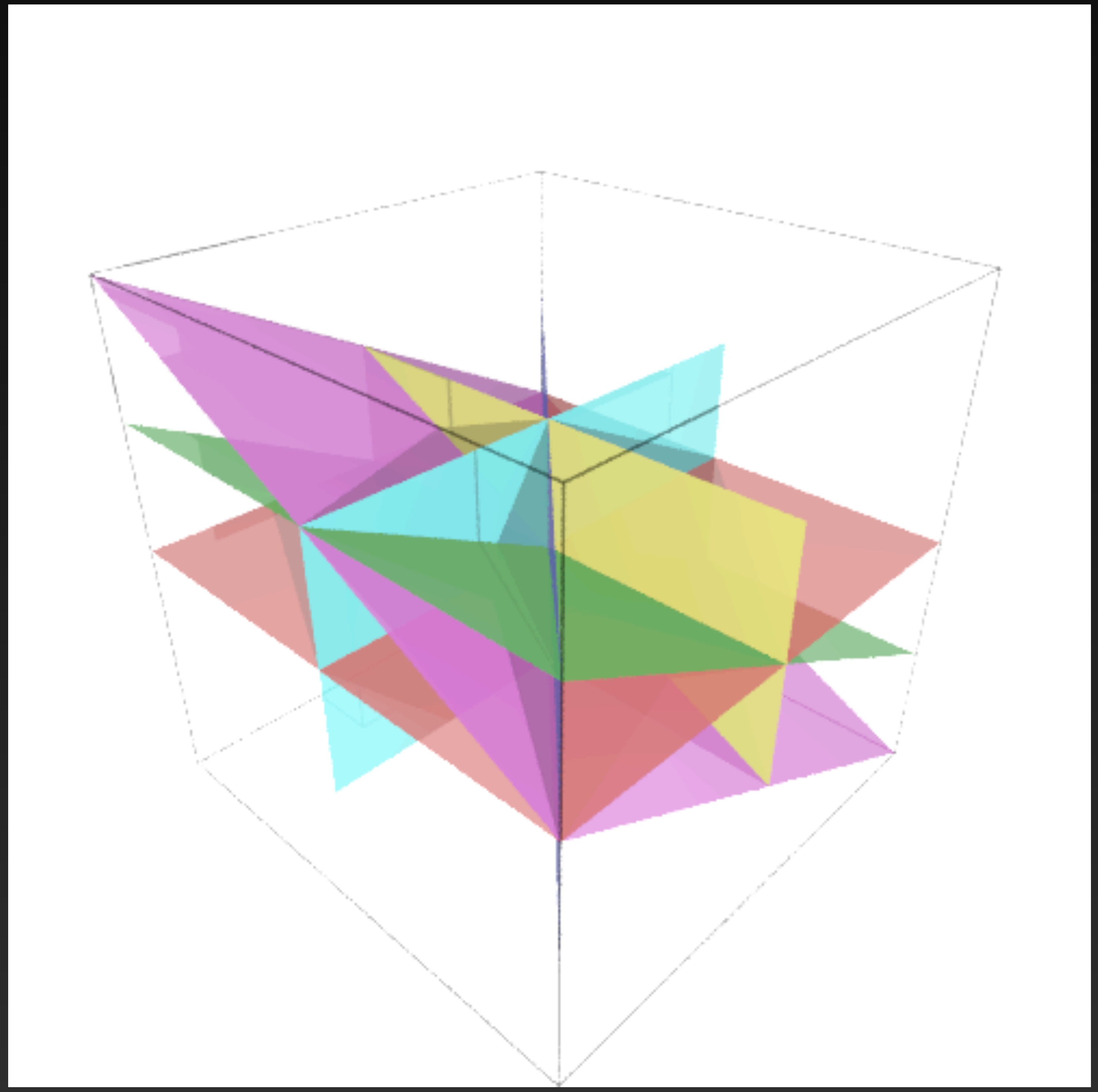
Hopf algebra

- comult, unit, counit, antipode
  - involutive antipode: $S^2 = id$
- convolution: comult -> id -> antipode -> mult
- antipode reconstruction
- antipode is the braiding, intertwiner
- deforming antipode



Coxeter bialgebra 1

- Coxeter groups: represent reflections
- set operations
- braid arrangements, generalization of hyperplane arrangements
- Aguiar2017, Aguiar2020, Aguiar2022



Coxeter bialgebra 2

Geometry	Combinatorics
face	set composition
chamber	linear order
flat	set partition
cone	preorder
top-cone	partial order
gallery interval	partial order of order dimension 1 or 2
chart	simple graph
dichart	simple directed graph
top-nested face	set composition with a linear order on each block
top-lune	set partition with a linear order on each block, or parallel composition of linear orders
top-star	series composition of discrete partial orders
top-star-lune	series-parallel partial order
nested face	set composition with a composition of each block
lune	set partition with a composition of each block
face-type	integer composition
flat-type	integer partition
nested face-type	integer composition with a composition of each part
lune-type	integer partition with a composition of each part

Total programming: Overview

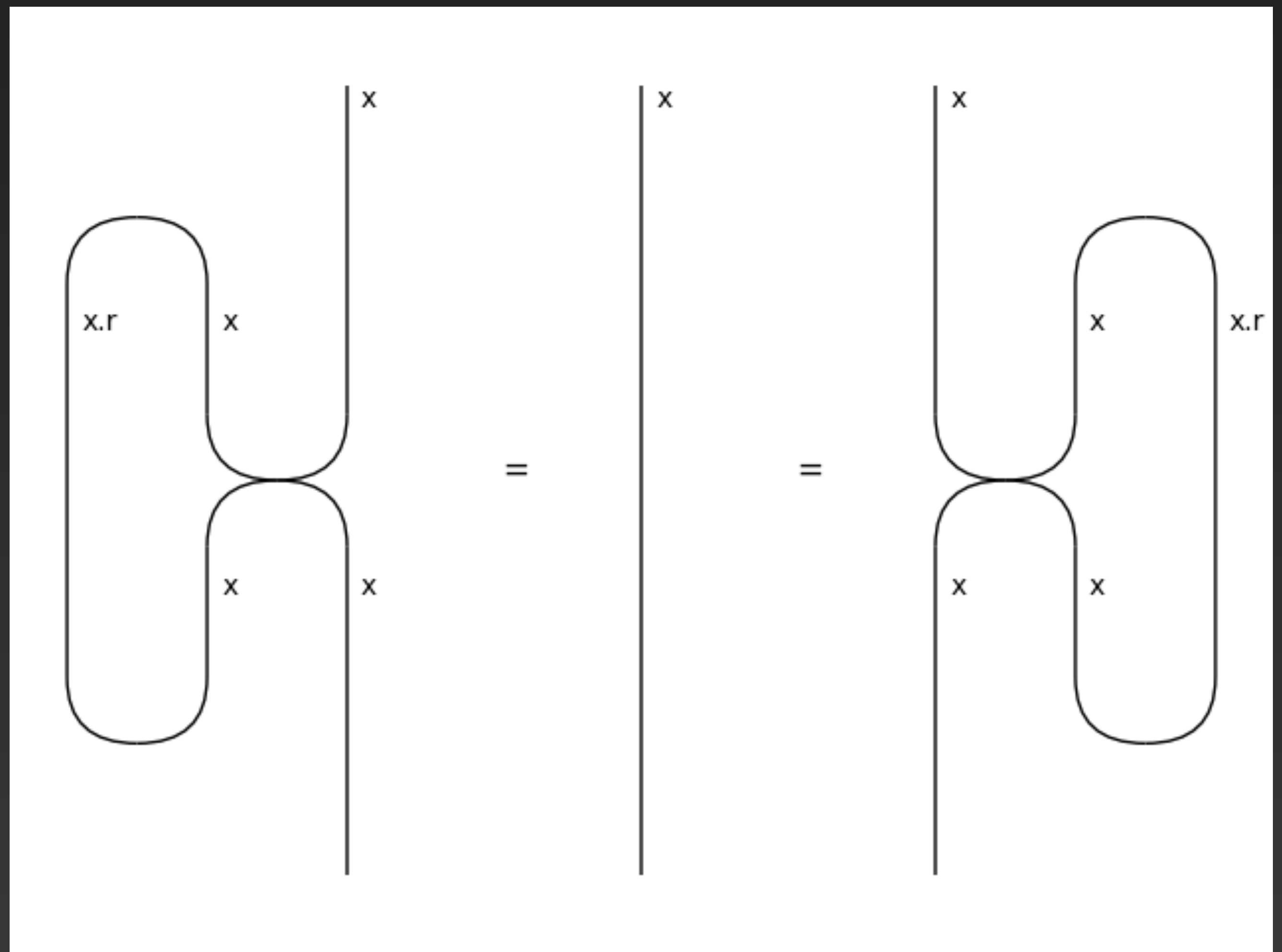
- eager and lazy evaluation
- recursion and corecursion
- halting problem is solved
- bounded computation
- theorem provers
- Radul2022

Total programing: Lazy computation

- values evaluated only when needed
- advantages for numerical computation
  - e.g. matrix sum A = B + C + D + E
  - expression templates in C++ (Eigen)
- automatic differentiation
- custom control flow operations, repl for free
- coinductive: stream, async, combinatorics
  - Doug McIlroy, Jan Rutten, Bart Jacobs, [Clenaghan2018]
- lazy evaluation has never been done correctly [Asperti1999]
- Weiss2021: lazy computation for transformers

Hopf machine learning

- find braid arrangement that represents partitioning and “program”
- you learn by “yanking”
- convolution networks are obvious, solve convolutional equation
- transformers are just SVMs Tarzanagh2023
- random walks: diffusion
- architecture discovery Achille2017



Glimpses of Hopf

- superposition + polysemanticity
  - In physics, wherever there is a linear system with a "superposition principle", a convolution operation makes an appearance.
- residual stream no priviledged basis, it is transformation invariant [Elhage2021]
  - it is a trace

Verification

- zonotopes: defined by Minkowski sum
- used in verification to partition space and reason about transitions
- https://mitadmissions.org/blogs/entry/what-is-a-zonotope/
- zonotopes is dual to hyperplane arrangements

- permutahedron is a particular zonotope associated with braid arrangements [Bastidas2017]

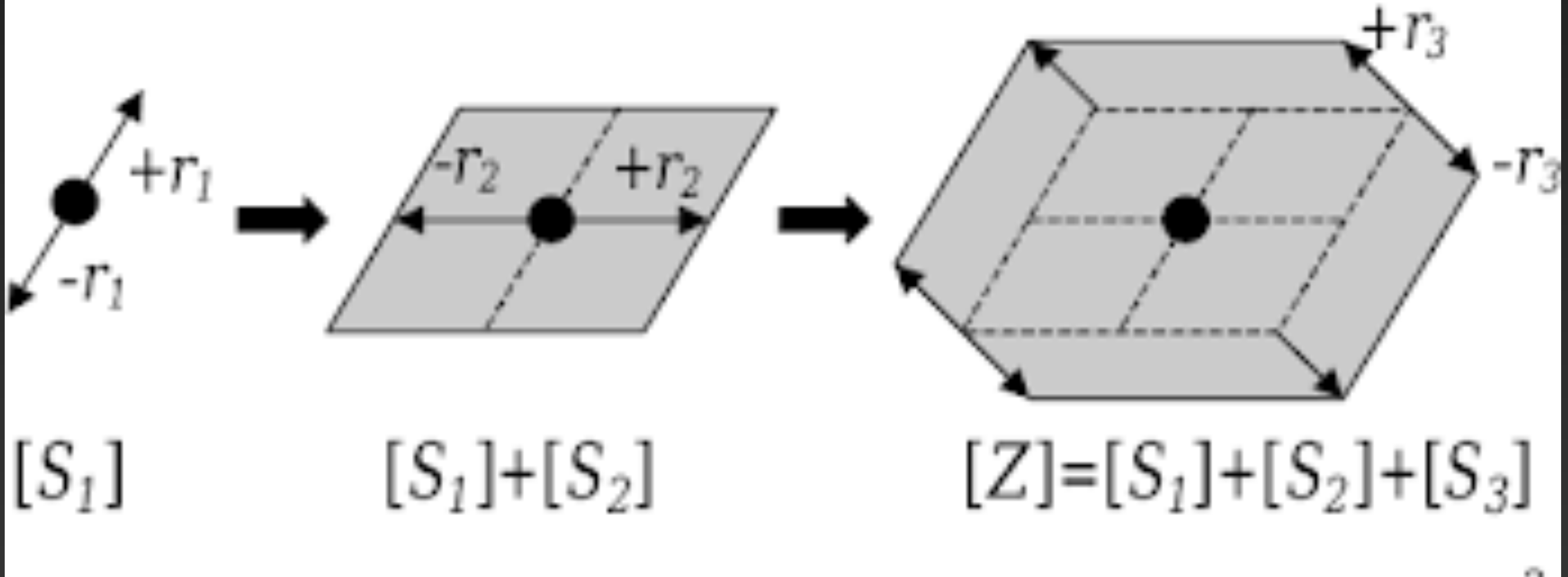
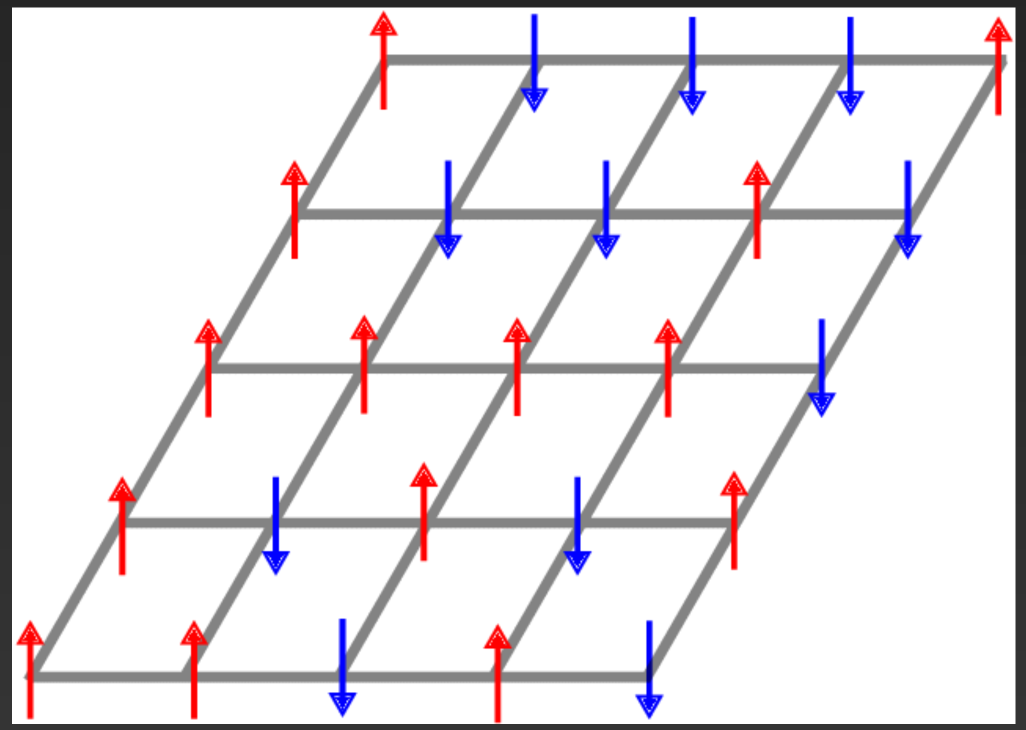


Figure 1: Construction of a 3-zonotope [Z] in $\Re^2$

Misc.

- renormalization
  - modeling of self-interaction
  - convolution: Kreimer2000, Ditto2002
  - ML: Roberts2021
- Ising model
  - renormalization
  - finding min-cut partitioning
  - Hopfield networks Ramsauer2020



Conclusion

- Maximizing partitioning, symmetry by repeated partitioning and reunification
  - conv == corr
  - antipode figures out braiding which defines the evaluation/coevaluation map and partitioning

References

- Achille2017: https://arxiv.org/abs/1706.01350
- Asperti1999: Optimal Implementation of Functional Programming Languages
- Bastidas2020: https://arxiv.org/abs/2009.05876
- Boyd2003: https://web.stanford.edu/class/ee102/lectures/tf.pdf
- Caulton2021: https://www.taylorfrancis.com/chapters/edit/10.4324/9781315623818-54/permutations-adam-caulton
- Claerbout: https://reproducibility.org/RSF/book/bei/conj/paper_html/index.html
- Clenaghan2018: https://arxiv.org/abs/1812.05878
- Elhage2021: https://transformer-circuits.pub/2021/framework/index.html
- Kasirajan2021: Fundamentals of Quantum Computing
- Lawvere2009: Conceptual mathematics
- Pavlovic2023: Programs as Diagrams: From Categorical Computability to Computable Categories
- Poli2023: https://arxiv.org/abs/2302.10866
- Radul2022: https://arxiv.org/abs/2204.10923
- Ramsauer2020: https://arxiv.org/abs/2008.02217
- Tarzanagh2023: https://arxiv.org/abs/2308.16898
- Weiss2021: https://arxiv.org/abs/2106.06981