

Bilinear Maps

Interpreting linear algebraic operations and transformations

James Pustejovsky

Brandeis University
Department of Computer Science

CS 135: Computational Semantics

Fall 2026

Roadmap

I	The shape	maps against forms; two vectors in, one number out
II	The middle is a choice	weighting, correlation, and a ranking that reverses
III	Bilinear maps, named	the definition, once we have earned it
IV	Interpreting the operations	contraction and its refusal; binding; rank
V	Which forms are metrics	symmetry, positive definiteness, and relations
VI	Coordinates and metric	the trade-off, and absorbing G
VII	Beyond inner products	null vectors, Minkowski, signature
VIII	Close	where these appear, and where G comes from

Parts I–IV are about what the operations *mean*. Parts V–VIII are about which choices of the middle behave like a notion of similarity, and what happens when they do not.

Part IV is the bridge to tensor product representations and vector symbolic architectures later in the course.

You Have Already Computed One of These

From the linear algebra refresher. Our domain was $\{\text{Alice}, \text{Bob}, \text{Carol}, \text{Fido}\}$, and like was a matrix:

L	A	B	C	F
A	0	1	0	0
B	0	0	1	0
C	0	0	0	1
F	0	0	0	1

Each individual has a **basis vector**, the one-hot column that picks it out:

$$\mathbf{e}_A = (1, 0, 0, 0), \quad \mathbf{e}_B = (0, 1, 0, 0), \quad \text{and so on.}$$

To ask *does Alice like Bob*, we sandwiched L between two of them:

$$\mathbf{e}_A^\top L \mathbf{e}_B = L_{AB} = 1$$

Two vectors went in. One number came out. That shape – vector, matrix, vector – is the subject of today.

One Matrix, Two Jobs

That same L can be used in two quite different ways.

Give it **one** vector and it returns a vector:

$$L \mathbf{e}_B = (1, 0, 0, 0) = \{\text{Alice}\} \quad \text{those who like Bob}$$

Here L is a **transformation**: a vector in, a vector out. This is how we have used matrices since Session 7 – the SVD factored objects of this kind.

Give it **two** vectors and it returns a number:

$$\mathbf{e}_A^\top L \mathbf{e}_B = 1 \quad \text{does Alice like Bob}$$

Here L is a **form**: two vectors in, one scalar out.

Nothing about L changed. The same sixteen numbers, used two ways. Which kind of object a matrix *is* depends on how many arguments you hand it.

The Same Shape, Somewhere Unexpected

Here is the sandwich again, with a different matrix in the middle:

$$\mathbf{x}^\top \textcolor{red}{I} \mathbf{y}$$

That is the ordinary dot product. We have used it in every session since:

Session 7	$\cos(\mathbf{v}, \mathbf{w})$ – how similar are two words?
Session 8	$\mathbf{u}_i^\top \mathbf{v}_j$ – the SVD, built out of dot products
Session 9	$\mathbf{w}_w^\top \mathbf{c}_c$ – the SGNS score
Attention	$\mathbf{q}^\top \mathbf{k}$ – how much should this token attend to that one?

With like in the middle, nobody would say the matrix was neutral. It *is* the relation. So what is the *I* doing, and why does it look like it is doing nothing?

The Question for the Next Forty Minutes

When we compare two meanings, *what sits between them?*

Three things to establish, in order:

- ① The dot product has a matrix in the middle, and that matrix is a **choice**.
- ② Change it and the answers change – including which word is nearest to which.
- ③ There is a name for the whole family of such objects, and its shape tells you which ones behave like similarity and which ones do something else.

Nothing here is new machinery. It is a second look at an operation we have been using since week two.

Maps and Forms: Where This Deck Sits

Four things a matrix can be, by how many vectors go in and what comes out.

	Signature	Written	Where we have met it
linear functional	$V \rightarrow \mathbb{R}$	$\mathbf{a}^\top \mathbf{x}$	<i>does x sleep?</i>
linear map	$V \rightarrow V$	$M\mathbf{x}$	$L\mathbf{e}_B$; every matrix in Sessions 7–9
bilinear form	$V \times V \rightarrow \mathbb{R}$	$\mathbf{x}^\top G \mathbf{y}$	$\mathbf{e}_A^\top L \mathbf{e}_B$; cosine; attention
bilinear map	$V \times V \rightarrow W$	$\mathbf{x} \otimes \mathbf{y}$	the outer product

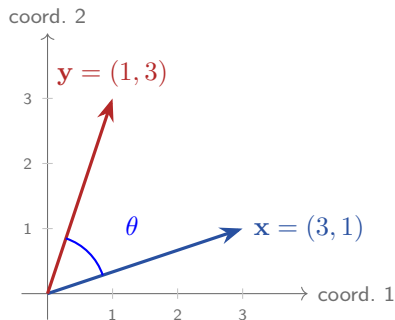
A bilinear *form* is the special case of a bilinear *map* whose output happens to be a number. Both are linear in each argument separately; they differ only in where the result lands.

Most of today is the third row – and the whole question will be what sits in the middle. Part IV reaches the fourth, where the output stops being a scalar. That is the row vector symbolic architectures and geometric algebra are built on, and it is why this deck is not called *Bilinear Forms*.

Two Vectors, One Number

Take two vectors in the plane, to keep every number small enough to check by hand:

$$\mathbf{x} = (3, 1), \quad \mathbf{y} = (1, 3)$$



The dot product is going to be our answer to “how much do these two agree?” We will compute it three different ways and get three different answers, all of them defensible.

The Familiar Recipe

Multiply matching coordinates, then add:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^n x_i y_i$$

For our two vectors:

$$\mathbf{x} \cdot \mathbf{y} = \underbrace{3 \times 1}_{\text{coord. 1}} + \underbrace{1 \times 3}_{\text{coord. 2}} = 3 + 3 = 6$$

And the length of $\mathbf{x}$ is the dot product of $\mathbf{x}$ with itself, square-rooted:

$$\|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{9 + 1} = \sqrt{10} = 3.162$$

Two things here look like definitions but are choices. **Only matching coordinates meet:** x_1 multiplies y_1 , never y_2 . **Every product counts equally:** no coordinate is worth more than another. Both are about to change.

Step One: Insert the Identity

I is the matrix that leaves vectors alone. In two dimensions:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad I\mathbf{y} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 1(1) + 0(3) \\ 0(1) + 1(3) \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

So we may insert it for free, and nothing changes:

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x} \cdot (I\mathbf{y}) = \mathbf{x}^\top I \mathbf{y}$$

$$\begin{array}{ccc} \boxed{\mathbf{x}^\top} & \boxed{I} & \boxed{\mathbf{y}} \\ \text{row} & \text{middle} & \text{column} \end{array} = 6$$

Nothing has been computed differently. We have only made a hidden object visible.

Step Two: Write Out Every Term

The sandwich, with both sums written out. For $n = 2$ there are $2 \times 2 = 4$ terms:

$$\begin{aligned}\mathbf{x}^\top G \mathbf{y} &= \sum_{i=1}^2 \sum_{j=1}^2 x_i G_{ij} y_j \\ &= \underbrace{x_1 G_{11} y_1}_{1 \text{ with } 1} + \underbrace{x_1 G_{12} y_2}_{1 \text{ with } 2} + \underbrace{x_2 G_{21} y_1}_{2 \text{ with } 1} + \underbrace{x_2 G_{22} y_2}_{2 \text{ with } 2}\end{aligned}$$

With $G = I$, two of those four coefficients are zero:

$$= 3(\textcolor{teal}{1})(1) + 3(\textcolor{red}{0})(3) + 1(\textcolor{red}{0})(1) + 1(\textcolor{teal}{1})(3) = 3 + 0 + 0 + 3 = 6$$

G_{ij} is the exchange rate between coordinate i of the left vector and coordinate j of the right one. There are n^2 of them: the identity sets n to one – the diagonal – and the other $n^2 - n$ to zero.

In a 300-dimensional embedding that is 90,000 rates: 300 ones and 89,700 zeros, each one a claim that two directions have nothing to do with each other.

What the Identity Is Claiming

That pattern of ones and zeros is three separate assertions, and the data forces none of them.

Entry	Value	The claim
G_{11}, G_{22}	1	every coordinate matters exactly as much as every other
G_{12}, G_{21}	0	distinct coordinates do not interact at all
$G_{12} = G_{21}$	equal	the two arguments play symmetric roles

Take two of the 300 directions from the previous slide – say coordinate 37 and coordinate 212. One of those 89,700 zeros says the two have nothing to do with each other; two of the 300 ones say they matter exactly as much. Both claims may be true. Neither was checked.

Change any of the three and you still have a perfectly good way of comparing two vectors. It simply will not be *the dot product* any more.

In-Class Exercise 1: Just the Arithmetic

No theory yet. Take

$$\mathbf{u} = (2, 4), \quad \mathbf{v} = (5, 1)$$

- 1 Compute $\mathbf{u}^\top I \mathbf{v}$ by writing out all four terms.
- 2 Now replace the middle with $G = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ and compute $\mathbf{u}^\top G \mathbf{v}$ the same way.
- 3 Now use $G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. What did this matrix do to the pairing?
- 4 Compute $\|\mathbf{u}\|_G = \sqrt{\mathbf{u}^\top G \mathbf{u}}$ for the G in item 2.

Three minutes. The point is to understand where each matrix entry lands in the sum.

In-Class Exercise 1: Answers

1. $2(1)(5) + 2(0)(1) + 4(0)(5) + 4(1)(1) = 10 + 0 + 0 + 4 = 14$

2. $2(\textcolor{red}{2})(5) + 0 + 0 + 4(1)(1) = 20 + 4 = \textcolor{green}{24}$. The first coordinate's contribution doubled; the second was untouched.

3. $2(0)(5) + 2(\textcolor{red}{1})(1) + 4(\textcolor{red}{1})(5) + 4(0)(1) = 0 + 2 + 20 + 0 = \textcolor{green}{22}$

This G pairs coordinate 1 of $\mathbf{u}$ with coordinate 2 of $\mathbf{v}$, and vice versa. “Matching coordinates” was never a law – it was the diagonal of I .

4. $\mathbf{u}^\top G \mathbf{u} = 2(2)(2) + 4(1)(4) = 8 + 16 = 24$, so $\|\mathbf{u}\|_G = \sqrt{24} = 4.899$.

Compare $\|\mathbf{u}\| = \sqrt{4 + 16} = \sqrt{20} = 4.472$ under I . The same vector, two lengths.

Changing One Entry

Keep the zeros off the diagonal, but let the diagonal entries differ:

$$G = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$$

Two of the four terms still vanish, so the sum is short:

$$\mathbf{x}^\top G \mathbf{y} = 4x_1y_1 + 1x_2y_2$$

With $\mathbf{x} = (3, 1)$ and $\mathbf{y} = (1, 3)$:

$$\mathbf{x}^\top G \mathbf{y} = 4(3)(1) + 1(1)(3) = 12 + 3 = 15$$

And the lengths move too. Recall $\|\mathbf{x}\|_G = \sqrt{\mathbf{x}^\top G \mathbf{x}}$ – the form with the *same* vector in both slots. The subscript says which metric is doing the measuring:

$$\|\mathbf{x}\|_G = \sqrt{4(9) + 1} = \sqrt{37} = 6.083, \quad \|\mathbf{y}\|_G = \sqrt{4(1) + 9} = \sqrt{13} = 3.606$$

Under I that same $\mathbf{x}$ had length $\sqrt{10} = 3.162$. One arrow, two lengths, because the ruler changed.

What a Diagonal Metric Means

$$\mathbf{x}^\top \text{diag}(4, 1) \mathbf{y} = 4x_1y_1 + 1x_2y_2$$

This is not exotic. It is what you do whenever you weight features:

tf-idf	rare terms get a larger weight than common ones
PPMI	informative co-occurrences count, uninformative ones are zeroed
Truncated SVD	the top k directions count; the rest are set to weight 0

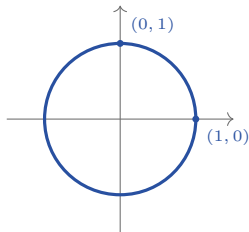
in the norm, $\mathbf{x}^\top G \mathbf{x}$	moving along coordinate 1 costs four times as much
in the form, $\mathbf{x}^\top G \mathbf{y}$	agreeing along coordinate 1 pays four times as much

One rate, two uses: you pay it when measuring a single vector and collect it when comparing two.
Under I every coordinate trades at par.

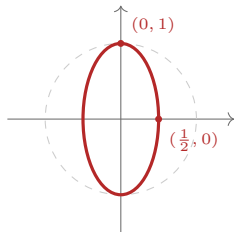
Every weighting above is a diagonal G : changes to the *vectors*, equally changes to the *metric*.

Seeing a Metric: the Set of Unit Vectors

Here is a way to look at G directly. Draw every vector of length exactly 1 – that is, every $\mathbf{v}$ with $\mathbf{v}^\top G \mathbf{v} = 1$.



$$G = I$$
$$x_1^2 + x_2^2 = 1$$



$$G = \text{diag}(4, 1)$$
$$4x_1^2 + x_2^2 = 1$$

Under I the unit vectors form a circle. Under $\text{diag}(4, 1)$ they form an ellipse, squashed along the direction the metric charges more for.

Reading That Picture

Every point on the red ellipse is a vector the metric calls *length one*.

Vector	Length under I	Length under $\text{diag}(4, 1)$
$(1, 0)$	1	2
$(\frac{1}{2}, 0)$	$\frac{1}{2}$	1
$(0, 1)$	1	1

The ellipse is short in the horizontal direction precisely because horizontal distance is *expensive*: you do not have to go far to use up a unit of length.

So the shape of the unit ball is the metric, drawn. A circle means I . Any other shape means something else is in the middle.

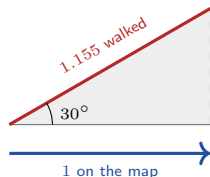
And **distance** comes free from the norm – it is just the length of the gap between two points:

$$d_G(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_G.$$

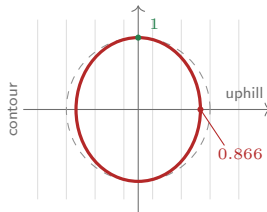
The Metric as Terrain

A mile on the map is not a mile on your legs. Let the ground have height $f(x, y)$, and keep measuring displacement on the *map*. The distance you actually walk is

$$\|\mathbf{x}\|_G, \quad G = I + \nabla f \nabla f^\top$$



side view



top view: unit ball (dashed: the map's)

In the aligned basis $G = \text{diag}(4/3, 1)$: eigenvalue $1 + |\nabla f|^2$ straight uphill, eigenvalue 1 along the contour, where walking is free. The semi-axis is $1/\sqrt{4/3} = \cos 30^\circ$.

What the Terrain Picture Buys, and Where It Stops

What it gets right. Every feature of the ellipse has a physical reading:

- The **expensive direction** is straight up the slope; the **free direction** is the contour line. $\nabla f = 0$ gives flat ground and $G = I$.
- The extra cost $\nabla f \nabla f^\top$ is **rank 1**, and its null direction *is* the contour.
- A hillside running diagonally across the grid has $\nabla f = (f_x, f_y)$ with both entries nonzero, so G acquires an **off-diagonal** term $f_x f_y$. We come back to this shortly.
- Two headings 90° apart on the map are not 90° apart on the hillside: at $\pm 45^\circ$ to the gradient, $\cos_G = 1/7$, about 82° .

Two things it cannot do

Uphill and downhill cost the same. A quadratic form is even: $\|-\mathbf{x}\|_G = \|\mathbf{x}\|_G$. Right for *distance traveled*, wrong for *effort* – the work $mg \Delta h$ is linear in the displacement, not quadratic.

One G is a tilted plane, not a mountain. Real terrain has a different G at every point. That is the step to Riemannian metrics; this deck is the story at a single point.

Why the Important Direction Is the Narrow One

Let coordinate 1 be **polarity** and coordinate 2 **topic**. We are building a sentiment model, so polarity is worth four times topic:

$$\text{good film} = (1, 1), \quad \text{bad film} = (-1, 1), \quad \text{good meal} = (1, -1)$$

	good film vs bad film	good film vs good meal
the gap $\mathbf{u} - \mathbf{v}$	$(2, 0)$	$(0, 2)$
its length under I	2.000	2.000
its length under $\text{diag}(4, 1)$	4.000	2.000

Both gaps are the same raw size; the metric charges four times as much for the one along polarity. **So read the ellipse as a tolerance region:** along topic you may wander a *full* unit from good film, but along polarity *half* a unit exhausts the budget.

The more a direction matters, the less you may move along it before you are something else. Important means *narrow tolerance*, and narrow tolerance is a short axis: weight w gives semi-axis $1/\sqrt{w}$.

The Same Ellipse, Twice

We drew an ellipse in Session 8 as well. There A was the matrix whose SVD we took, and the ellipse was its *image* of the unit circle.

	Where the ellipse comes from	Semi-axis
Session 8	$\{A\mathbf{u} : \ \mathbf{u}\ = 1\}$	σ_i
Today	$\{\mathbf{v} : \mathbf{v}^\top G \mathbf{v} = 1\}$	$1/\sqrt{w_i}$

These are the same ellipse, not an analogy. If $\mathbf{v} = A\mathbf{u}$ with $\|\mathbf{u}\| = 1$ then $\|A^{-1}\mathbf{v}\| = 1$, so the image is exactly the unit ball of $G = (AA^\top)^{-1}$, and $1/\sqrt{w_i} = \sigma_i$. Session 8's $A = \begin{pmatrix} 2 & 3 \\ 0 & 2 \end{pmatrix}$ had $\sigma = (4, 1)$; its G gives $1/\sqrt{w} = (4, 1)$.

Now truncate. Setting $\sigma_{k+1}, \dots, \sigma_n = 0$ sends those weights to 0, and $1/\sqrt{0}$ is *infinite*: you may move as far as you like along those directions and accumulate no length.

Truncation does not shrink those directions – it stops the metric from seeing them. We meet them again later under their proper name, as *null* directions.

Filling In the Corners

Now let the off-diagonal entries be nonzero:

$$G = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

All four terms survive:

$$\mathbf{x}^\top G \mathbf{y} = 1 x_1 y_1 + 0.8 x_1 y_2 + 0.8 x_2 y_1 + 1 x_2 y_2$$

With $\mathbf{x} = (3, 1)$, $\mathbf{y} = (1, 3)$:

$$= 3(1) + 0.8(3)(3) + 0.8(1)(1) + 1(3) = 3 + 7.2 + 0.8 + 3 = 14$$

And the length:

$$\|\mathbf{x}\|_G^2 = 9 + 0.8(3)(1) + 0.8(1)(3) + 1 = 9 + 2.4 + 2.4 + 1 = 14.8, \quad \|\mathbf{x}\|_G = 3.847$$

The cross terms $x_1 y_2$ and $x_2 y_1$ are the new part: coordinate 1 now earns credit for agreeing with coordinate 2.

What an Off-Diagonal Entry Means

In words

$G_{12} = 0.8$ says the two coordinates are **not independent**. Having some of coordinate 2 partly counts as having coordinate 1.

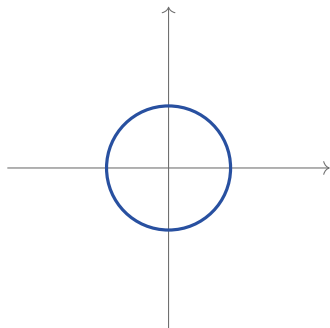
Coordinate 1 counts espresso, coordinate 2 cappuccino. Take $\mathbf{u} = (1, 0)$, all espresso, and $\mathbf{v} = (0, 1)$, all cappuccino – **not one word in common**:

	$\mathbf{u}^\top G \mathbf{v}$	$\cos_G(\mathbf{u}, \mathbf{v})$
$G = I$	0	0
$G_{12} = 0.8$	0.8	0.8

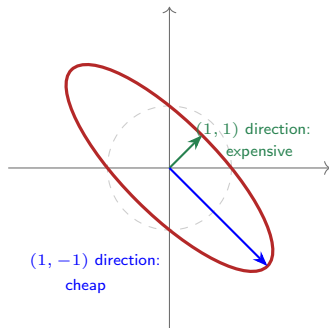
Both have G -length 1, so the cosine is the form itself. **Two documents sharing no word at all, and the metric calls them 0.8 similar.**

This is the knowledge truncated SVD gave us in Session 8, by a different door: there we changed the vectors so the dot product would behave, here we change the middle.

The Correlated Metric, Drawn



$$G = I$$



$$G = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

The ellipse has tilted. Increasing both coordinates together uses up length fast; trading one coordinate for the other is nearly free, because this metric regards that as barely a change at all.

Three Middles, the Same Two Vectors

$\mathbf{x} = (3, 1)$ and $\mathbf{y} = (1, 3)$ throughout. Nothing about the vectors changes.

G	$\mathbf{x}^\top G \mathbf{y}$	$\ \mathbf{x}\ _G$	$\cos_G(\mathbf{x}, \mathbf{y})$	what it assumes
I	6.000	3.162	0.6000	equal, independent coordinates
$\text{diag}(4, 1)$	15.000	6.083	0.6839	coordinate 1 counts four times
$\begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$	14.000	3.847	0.9459	the coordinates are correlated

Three answers to “how similar are these two?”: 0.60, 0.68, 0.95 – the difference between “somewhat related” and “nearly the same thing.”

If you report a cosine without saying which G , you have reported one of these and called it *the* answer.

Two forward pointers. Writing $\llbracket \alpha \rrbracket^{\mathcal{M}, g}$ makes this same move, which the next slide takes seriously. And a transformer makes it H times over: every attention head carries its own learned middle.

A Move You Have Been Making Since Session 2

$$\llbracket \alpha \rrbracket^{\mathcal{M}, g} \quad \text{beside} \quad \|\mathbf{x}\|_G, \quad \cos_G(\mathbf{x}, \mathbf{y})$$

Neither object is interpreted on its own. The expression stays fixed; what moves is the structure it is evaluated against – and in both traditions that structure is parked in the margin of the symbol.

Which parameter is G ?

Not the assignment. An assignment supplies what is *missing* – the values of free variables – and $\mathbf{x} = (3, 1)$ is missing nothing. G supplies the structure evaluated *against*, which is the model's job.

	In logic	Here
what the symbols refer to	the assignment g	the choice of basis
the structure evaluated against	the model $\mathcal{M} = (D, I)$	the form, (V, G)

The failure mode is shared: both get dropped once they feel conventional. **Reporting a cosine without naming G is reporting a denotation without naming the model.**

Where the Comparison Stops

One disanalogy, and it is the useful one.

$\llbracket \cdot \rrbracket^{\mathcal{M},g}$ is defined by recursion on syntax. The model is threaded through every clause:

$$\llbracket \alpha \beta \rrbracket^{\mathcal{M},g} = \llbracket \alpha \rrbracket^{\mathcal{M},g} (\llbracket \beta \rrbracket^{\mathcal{M},g})$$

$\| \cdot \|_G$ recurses on nothing. Vectors have no syntax, so G parameterizes a single measurement rather than an interpretation.

So in a bare vector space, G is weaker than a model: a parameter on one operation, not a structure the whole system is read against.

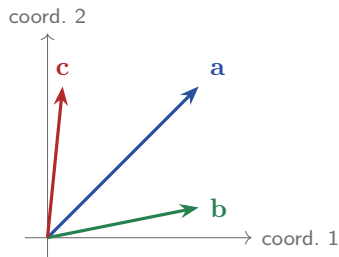
That gap closes when the signature is **declared** and every product in the algebra is defined relative to it. Then G is threaded through each operation the way $\mathcal{M}$ is threaded through each clause, and it becomes a model in the semanticist's sense.

Which is an argument for the third of the three routes we reach at the end of today. A declared metric is a model you can state; a learned one is a model nobody has written down.

The Case That Should Worry You

Magnitudes moving is one thing. Here is an *ordering* moving.

$$\mathbf{a} = (1, 1), \quad \mathbf{b} = (1, 0.2), \quad \mathbf{c} = (0.1, 1)$$



Think of $\mathbf{a}$ as a query word, and $\mathbf{b}$ and $\mathbf{c}$ as two candidates. We want to report the nearest neighbor.

By eye, $\mathbf{b}$ and $\mathbf{c}$ look about equally far from $\mathbf{a}$ – one leans one way, one the other.

Under the Ordinary Dot Product

$G = I$, so $\cos$ is the cosine you already know.

$$\mathbf{a} \cdot \mathbf{b} = 1(1) + 1(0.2) = 1.2, \quad \|\mathbf{a}\| = \sqrt{2} = 1.4142, \quad \|\mathbf{b}\| = \sqrt{1.04} = 1.0198$$

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{1.2}{1.4142 \times 1.0198} = 0.8321$$

$$\mathbf{a} \cdot \mathbf{c} = 1(0.1) + 1(1) = 1.1, \quad \|\mathbf{c}\| = \sqrt{1.01} = 1.0050$$

$$\cos(\mathbf{a}, \mathbf{c}) = \frac{1.1}{1.4142 \times 1.0050} = 0.7740$$

Nearest neighbor of $\mathbf{a}$: $\mathbf{b}$, at 0.8321 against 0.7740.

Under a Metric That Weights Coordinate 2

Now $G = \text{diag}(1, 9)$: the second coordinate counts nine times as much.

$$\mathbf{a}^\top G \mathbf{b} = 1(1)(1) + 9(1)(0.2) = 1 + 1.8 = 2.8$$

$$\|\mathbf{a}\|_G^2 = 1 + 9 = 10, \quad \|\mathbf{b}\|_G^2 = 1 + 9(0.04) = 1.36$$

$$\cos_G(\mathbf{a}, \mathbf{b}) = \frac{2.8}{\sqrt{10 \times 1.36}} = 0.7593$$

$$\mathbf{a}^\top G \mathbf{c} = 1(1)(0.1) + 9(1)(1) = 0.1 + 9 = 9.1, \quad \|\mathbf{c}\|_G^2 = 0.01 + 9 = 9.01$$

$$\cos_G(\mathbf{a}, \mathbf{c}) = \frac{9.1}{\sqrt{10 \times 9.01}} = 0.9587$$

Nearest neighbor of $\mathbf{a}$: $\mathbf{c}$.

The Ranking Reversed

G	$\cos_G(\mathbf{a}, \mathbf{b})$	$\cos_G(\mathbf{a}, \mathbf{c})$	nearest to $\mathbf{a}$
I	0.8321	0.7740	b
diag(1, 9)	0.7593	0.9587	c

Why: **b** agrees with **a** on the *cheap* coordinate; **c** agrees on the *expensive* one. The metric decides which agreement counts, and therefore which word wins.

The consequence for anything we build

If **a**, **b**, **c** were word vectors and you printed the top-1 nearest neighbor, you would print a different word. Not a slightly different score – a different *answer*. And nothing about the vectors changed.

In-Class Exercise 2: Make the Metric Do Your Bidding

Same three vectors: $\mathbf{a} = (1, 1)$, $\mathbf{b} = (1, 0.2)$, $\mathbf{c} = (0.1, 1)$.

- 1 Under $G = \text{diag}(9, 1)$, which of $\mathbf{b}$, $\mathbf{c}$ is nearest to $\mathbf{a}$? Predict first, then compute.
- 2 There is a value of t in $G = \text{diag}(1, t)$ at which $\mathbf{b}$ and $\mathbf{c}$ are exactly tied. Is it above or below 9? Above or below 1? You do not need to solve for it.
- 3 Using $G = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$, compute both cosines. The winner does not change – but *compare the gap between them to the gap under I . What has this metric done?*
- 4 ★ Suppose someone hands you the three vectors and tells you the correct answer is $\mathbf{c}$. Can you always find a diagonal G that makes it so? Can you always find one for *any* desired winner?

Five minutes. Item 4, the starred one, is worth arguing about.

In-Class Exercise 2: Answers

1. $\mathbf{b}$, and by a wide margin. With $\|\mathbf{a}\|_G^2 = 10$: $\mathbf{a}^\top G \mathbf{b} = 9 + 0.2 = 9.2$ and $\|\mathbf{b}\|_G^2 = 9.04$, giving $9.2/\sqrt{90.4} = 0.9676$; $\mathbf{a}^\top G \mathbf{c} = 0.9 + 1 = 1.9$ and $\|\mathbf{c}\|_G^2 = 1.09$, giving $1.9/\sqrt{10.9} = 0.5755$. Weighting coordinate 1 is the mirror image of weighting coordinate 2.
2. Between 1 and 9. At $t = 1$ the winner is $\mathbf{b}$; at $t = 9$ it is $\mathbf{c}$; the two cosines move continuously in t , so they cross somewhere in between.
3. $\mathbf{a}^\top G \mathbf{b} = 1 + 0.8(0.2) + 0.8(1) + 0.2 = 2.16$, with $\|\mathbf{a}\|_G^2 = 3.6$ and $\|\mathbf{b}\|_G^2 = 1.36$, so $\cos_G(\mathbf{a}, \mathbf{b}) = 0.9762$. Likewise $\mathbf{a}^\top G \mathbf{c} = 1.98$ with $\|\mathbf{c}\|_G^2 = 1.17$, so $\cos_G(\mathbf{a}, \mathbf{c}) = 0.9648$.
The gap was $0.8321 - 0.7740 = 0.058$ under I . It is now 0.011 – a fifth as large, with both values pushed up near 1. This metric has not reversed the ranking; it has nearly *erased* it. Treating the two coordinates as near-interchangeable leaves it barely able to distinguish leaning toward coordinate 1 from leaning toward coordinate 2, and that is the only way $\mathbf{b}$ and $\mathbf{c}$ differ.
4. ★ Yes for $\mathbf{c}$, and yes in general here. Which should alarm you: the choice of G has enough freedom to produce whichever nearest neighbor you wanted. A metric is not a neutral referee. If it is not reported, it is not checkable.

Why the Starred Answer Is Yes

$\mathbf{b} = (1, 0.2)$ and $\mathbf{c} = (0.1, 1)$ are not better and worse. They are good in *different places*: $\mathbf{b}$ delivers coordinate 1 and almost nothing else, $\mathbf{c}$ does the reverse. A diagonal metric has exactly one power – to say how much each coordinate counts – so choosing G is choosing which of the two agreements to believe.

$G = \text{diag}(1, t)$	$t = 0.5$	$t = 1$	$t = 1.32$	$t = 2$	$t = 9$
$\cos_G(\mathbf{a}, \mathbf{b})$	0.889	0.832	0.809	0.778	0.759
$\cos_G(\mathbf{a}, \mathbf{c})$	0.686	0.774	0.809	0.855	0.959
winner	$\mathbf{b}$	$\mathbf{b}$	tie	$\mathbf{c}$	$\mathbf{c}$

Both cosines move continuously in t ; $\mathbf{b}$ leads at one end and $\mathbf{c}$ at the other, so they must cross. The crossing is at $t \approx 1.32$: it takes a 32% thumb on the scale, not a dramatic metric, to replace the answer.

The three vectors never changed. Which one is nearest was settled by G alone, and G is not in the data.

But Not Any Winner You Like

The freedom is bounded, and the bound holds for every positive definite G , not just the diagonal ones. Any such G factors as $G = LL^\top$, and then

$$\cos_G(\mathbf{u}, \mathbf{v}) = \cos(L^\top \mathbf{u}, L^\top \mathbf{v})$$

– every metric is the ordinary cosine after an invertible linear map. We run this all the way in Part VI.

What a linear map cannot do

It can stretch directions and shear them, but it cannot change which ray lies **between** two others. Betweenness survives every invertible map.

Measured against $\mathbf{a}$, a candidate is just a direction: $\mathbf{b}$ leans one way, $\mathbf{c}$ the other, with $\mathbf{a}$ in between.

- **Opposite sides of $\mathbf{a}$:** neither is between, so the order is up for grabs. That is $\mathbf{b}$ against $\mathbf{c}$, and why item 4 is yes.
- **Same side as each other:** the inner one is between $\mathbf{a}$ and the outer one, so it wins under every metric. A candidate at $(1, 0.5)$ beats one at $(1, 0.2)$ no matter what G you pick.

Now the Name

Everything so far has had the same shape: two vectors in, one number out, with a matrix in the middle. That family has a name.

Definition

A **bilinear form** on $\mathbb{R}^n$ is a map $B : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ that is linear in each argument when the other is held fixed:

$$B(\mathbf{x} + \mathbf{x}', \mathbf{y}) = B(\mathbf{x}, \mathbf{y}) + B(\mathbf{x}', \mathbf{y}), \quad B(\lambda \mathbf{x}, \mathbf{y}) = \lambda B(\mathbf{x}, \mathbf{y})$$

and likewise in the second argument.

“Bi” because there are two arguments; “linear” because it is linear in each. Not linear in both at once – $B(2\mathbf{x}, 2\mathbf{y}) = 4B(\mathbf{x}, \mathbf{y})$, not $2B(\mathbf{x}, \mathbf{y})$.

Checking Bilinearity by Hand

Take $G = \text{diag}(4, 1)$, $\mathbf{x} = (3, 1)$, $\mathbf{x}' = (1, 2)$, $\mathbf{y} = (1, 3)$.

Additivity in the first argument.

$$\mathbf{x}^\top G \mathbf{y} = 12 + 3 = 15, \quad \mathbf{x}'^\top G \mathbf{y} = 4(1)(1) + 1(2)(3) = 4 + 6 = 10$$

$$(\mathbf{x} + \mathbf{x}') = (4, 3) : \quad (\mathbf{x} + \mathbf{x}')^\top G \mathbf{y} = 4(4)(1) + 1(3)(3) = 16 + 9 = 25 = 15 + 10 \checkmark$$

Scaling.

$$(2\mathbf{x})^\top G \mathbf{y} = 4(6)(1) + 1(2)(3) = 24 + 6 = 30 = 2 \times 15 \checkmark$$

Both hold for any G whatsoever, because $x_i G_{ij} y_j$ is a product with exactly one x and one y in it. That is the whole content of bilinearity.

Every Bilinear Form Is a Matrix

So far we have gone one direction: given G , we get a form. The converse also holds, and it is the sentence that makes the matrix inevitable.

If B is bilinear, then $B(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top G \mathbf{y}$ for exactly one matrix G , whose entries are

$$G_{ij} = B(\mathbf{e}_i, \mathbf{e}_j)$$

Why. Write $\mathbf{x} = \sum_i x_i \mathbf{e}_i$ and $\mathbf{y} = \sum_j y_j \mathbf{e}_j$. Pull each sum out by linearity, one argument at a time:

$$B(\mathbf{x}, \mathbf{y}) = \sum_i x_i B(\mathbf{e}_i, \mathbf{y}) = \sum_i \sum_j x_i y_j B(\mathbf{e}_i, \mathbf{e}_j)$$

Compare that with $\sum_i \sum_j x_i G_{ij} y_j$ from earlier. The two expressions match term by term, so $G_{ij} = B(\mathbf{e}_i, \mathbf{e}_j)$.

Recovering the Matrix From the Form

The formula $G_{ij} = B(\mathbf{e}_i, \mathbf{e}_j)$ is a recipe you can run. Suppose someone gives you a rule and no matrix:

$$B((x_1, x_2), (y_1, y_2)) = 3x_1y_1 - x_1y_2 + 5x_2y_2$$

Probe it with basis vectors, one pair at a time:

$B(\mathbf{e}_1, \mathbf{e}_1) = 3(1)(1) = 3$	$\Rightarrow G_{11} = 3$	$G = \begin{pmatrix} 3 & -1 \\ 0 & 5 \end{pmatrix}$
$B(\mathbf{e}_1, \mathbf{e}_2) = -1(1)(1) = -1$	$\Rightarrow G_{12} = -1$	
$B(\mathbf{e}_2, \mathbf{e}_1) = 0$	$\Rightarrow G_{21} = 0$	
$B(\mathbf{e}_2, \mathbf{e}_2) = 5$	$\Rightarrow G_{22} = 5$	

Notice what that is: sandwiching B between two basis vectors is *table lookup* – exactly what $\mathbf{e}_A^\top M \mathbf{e}_B$ did on the first slide. The matrix is not an extra ingredient added to the form. It *is* the form, tabulated on a basis.

Two Ways to Combine Two Vectors

We have been reading $\mathbf{x}^\top G \mathbf{y}$ as a single act. It is really a choice between two, and the choice is not about the matrix at all.

	Operation	Result
Contract	$\mathbf{x}^\top \mathbf{y}$	a number
Do not contract	$\mathbf{x} \mathbf{y}^\top$	a matrix

Same two vectors. Same multiplications, even – every product $x_i y_j$ appears in both. The difference is whether the products are summed away or kept.

That turns out to be the difference between *applying* a predicate and *binding* one, and it is the operation the rest of the course is built on.

Contraction, Recalled

The relation like, row indexed by the liker, column by the liked:

$$L = \begin{array}{c} \begin{array}{cc} & \begin{array}{cccc} & A & B & C & F \end{array} \\ \begin{array}{c} A \\ B \\ C \\ F \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

One contraction at a time:

$$L \mathbf{e}_B = (1, 0, 0, 0) = \{\text{Alice}\} \quad \mathbf{e}_A^\top (L \mathbf{e}_B) = 1$$

The intermediate object is a *set*: the people who like Bob, which is $\lambda x. \text{like}(x, b)$. Each contraction consumes one argument and drops the rank by one.

Two contractions, two β -reductions, and we land on a truth value.

The Other Direction

Now take $\mathbf{a} = \mathbf{e}_A$ and the predicate $s = \mathcal{V}(\text{sleep}) = (0, 0, 1, 1)$, i.e. $\{\text{Carol}, \text{Fido}\}$.

Contracting gives a truth value, as usual:

$$\mathbf{a}^\top \mathbf{s} = 1(0) + 0(0) + 0(1) + 0(1) = \mathbf{0} \quad \text{Alice does not sleep.}$$

But we may also refuse to contract:

$$\mathbf{a} \mathbf{s}^\top = \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{4 \times 1} \underbrace{(0 \quad 0 \quad 1 \quad 1)}_{1 \times 4} = \text{a } 4 \times 4 \text{ matrix}$$

A matrix over $\mathcal{D}_e \times \mathcal{D}_e$ is a relation. So this is a relation built out of one individual and one unary predicate. Which relation?

The Matrix It Builds

$$\mathbf{a s}^{\top} = \begin{array}{c} \text{A} \\ \text{B} \\ \text{C} \\ \text{F} \end{array} \begin{pmatrix} \text{A} & \text{B} & \text{C} & \text{F} \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

One live row – Alice’s – and within it, the sleepers. As a set of pairs:

$$\{(\text{Alice}, \text{Carol}), (\text{Alice}, \text{Fido})\} = \{\text{Alice}\} \times \text{sleep}$$

In English

The relation that holds between Alice and every sleeper, and between nothing else and nothing.

It is a **Cartesian product**: a set of rows crossed with a set of columns. Every such “rectangle” is an outer product, and every outer product of indicator vectors is such a rectangle.

The Same Two Predicates, One Difference

Here is the whole distinction, in λ -terms. Write $A = \lambda x. x = \text{alice}$ and $S = \lambda x. \text{sleep}(x)$.

$$\mathbf{a}^\top \mathbf{s} = \sum_x A(x) \wedge S(x) \quad \text{one variable, bound by } \sum$$

$$\mathbf{a} \mathbf{s}^\top = \lambda x. \lambda y. A(x) \wedge S(y) \quad \text{two variables, each bound by } \lambda$$

Both are the conjunction of the same two unary predicates, and both bind every variable they introduce. What differs is how many distinct variables there are, and which binder discharges them.

Binder	What it does	Result
$\sum_x$	collapses the shared variable	a number
$\lambda x. \lambda y.$	opens two argument positions	a function of two arguments

Nothing is free on either side. The λ -term is *unsaturated*, not open – it awaits two arguments, and that is what makes it a relation.

The Thing That Should Bother You

$\mathbf{a}^\top \mathbf{s} = 0$ Alice does not sleep

$\mathbf{a} \mathbf{s}^\top \neq 0$ and yet the matrix is nonzero

The outer product is a perfectly good nonzero object built from a predication that is **false**.

So $\mathbf{a} \mathbf{s}^\top$ does not mean “Alice sleeps.” It asserts nothing whatever about Alice sleeping, because the two arguments never meet.

The inner product asks a question and gets an answer. The outer product builds an object and never asks. They are not two versions of one predication – they are predication and its refusal.

The Algebraic Signature: the Form Factors

Put the outer product in the middle of a bilinear form and watch it come apart:

$$\mathbf{x}^\top (\mathbf{a} \mathbf{s}^\top) \mathbf{y} = (\mathbf{x}^\top \mathbf{a}) (\mathbf{s}^\top \mathbf{y}) = (\mathbf{x} \cdot \mathbf{a}) (\mathbf{s} \cdot \mathbf{y})$$

Check, still with $\mathbf{a} = \mathbf{e}_A = (1, 0, 0, 0)$ and $\mathbf{s} = \mathcal{V}(\text{sleep}) = (0, 0, 1, 1)$. Probe the form with any two vectors at all – take $\mathbf{x} = (1, 2, 3, 4)$ and $\mathbf{y} = (5, 6, 7, 8)$:

$$\mathbf{x} \cdot \mathbf{a} = 1, \quad \mathbf{s} \cdot \mathbf{y} = 7 + 8 = 15, \quad \text{product} = 15$$

and directly, $\mathbf{x}^\top (\mathbf{a} \mathbf{s}^\top) \mathbf{y} = 15$ as well.

What that means

Whatever you ask this form, the answer is one fact about the left argument times one fact about the right, computed *separately*. No entry depends on the pair.

Compare L for like: there is no way to write L_{ij} as $u_i v_j$. Whether Alice likes Bob is not a fact about Alice times a fact about Bob.

Rank Measures How Relational a Relation Is

Rank 1

M has **rank 1** when it is a single outer product, $M = \mathbf{u}\mathbf{v}^\top$ – equivalently, when every entry factors, $M_{ij} = u_i v_j$. **The two indices do not interact**: no entry depends on the *pair*, only on the row through $\mathbf{u}$ and the column through $\mathbf{v}$.

	rank	singular values
$\mathbf{a}\mathbf{s}^\top = \{\text{Alice}\} \times \text{sleep}$	1	$(\sqrt{2}, 0, 0, 0)$
$L = \text{like}$	3	$(\sqrt{2}, 1, 1, 0)$

A rank-1 relation is two unary predicates glued together. **Rank measures how relational a relation is**, and genuine relational content starts at rank 2 – the semantic content of the number we spent Session 8 truncating.

Hold on to the boxed condition rather than to this reading of it. The condition returns at the end of Part IV, about a different kind of matrix, saying something that will sound unrelated.

Every Relation Is a Sum of Rectangles

From Session 8, every matrix decomposes:

$$M = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

Read that with what we now know: each term $\mathbf{u}_i \mathbf{v}_i^\top$ is a rank-1 outer product, which is to say a Cartesian rectangle – a relation with no relational content.

So every relation is a **superposition of non-relational pieces**. like is three such rectangles laid on top of one another.

The relational content is not in any single term. It is in the interference between them – in the fact that the rectangles overlap in a pattern no one of them could produce.

And truncating to rank k is a decision about how many rectangles are worth keeping.

Nothing Is Lost: Recovering an Operand

The outer product does not destroy what went into it. Contract on the left with $\mathbf{a}$:

$$\mathbf{a}^\top (\mathbf{a} \mathbf{s}^\top) = (\mathbf{a}^\top \mathbf{a}) \mathbf{s}^\top = 1 \cdot \mathbf{s}^\top = (0, 0, 1, 1)$$

$\mathbf{s}$ comes back exactly, because $\mathbf{a}$ is a unit vector.

Operation	Written	Effect
bind	$\mathbf{u} \mathbf{v}^\top$	join two vectors; rank +1
unbind	$\mathbf{u}^\top (\cdot)$	use one as a <i>cue</i> to return the other; rank -1

Bind with an outer product, unbind with an inner product. *The pair is invertible – which is what a representation has to be and what a truth value is not.*

Contraction on its own is not invertible. On a plain vector it is $\mathbf{a}^\top \mathbf{s}$, sending $\mathbb{R}^4$ to $\mathbb{R}$ and destroying three dimensions: $(0, 0, 1, 1)$ and $(0, 5, -3, 2)$ both give 0. Unbinding works only because the outer product kept $\mathbf{s}$ and the cue says where to look.

Either operand can serve as the cue; giving them distinct jobs is the next slide.

Where the Roles Already Are

We do not need to invent roles. L has two index positions, and Part IV has already been telling them apart:

$$L \mathbf{e}_j = \lambda x. \text{like}(x, j) \qquad \mathbf{e}_i^\top L = \lambda y. \text{like}(i, y)$$

The first fixes the *second* slot and leaves the first open; the second does the reverse. Those two slots are the argument positions of like.

	What it is	Here
role	an argument position of a relation or event	slot 1 and slot 2 of L
filler	the value occupying that position	Alice, Bob, Fido

Give the slots vectors: $\mathbf{r}_1, \mathbf{r}_2$ orthonormal, one dimension per argument position. A binary relation gets two, and nothing has been stipulated that L did not already have.

Note which way round this goes. The role is the position, the filler is the value, and retrieval contracts with *the role*.

Who Did What to Whom

Bind each filler to the slot it occupies. Alice likes Bob becomes one object:

$$T = \mathbf{r}_1 \mathbf{e}_A^\top + \mathbf{r}_2 \mathbf{e}_B^\top = \begin{array}{c} \text{slot 1} \\ \text{slot 2} \end{array} \begin{array}{cccc} A & B & C & F \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right) \end{array} \quad \mathbf{r}_1^\top T = \mathbf{e}_A^\top$$

Bob likes Alice fills the same slots the other way, giving T' , and

$$\cos(T, T') = 0 \quad \text{while} \quad \mathbf{e}_A + \mathbf{e}_B = \mathbf{e}_B + \mathbf{e}_A.$$

Orthogonal under binding; one vector for both sentences under addition.

And T has not replaced L – the two do different jobs. Recover the fillers from T , then ask L whether that tuple holds:

$$(\mathbf{r}_1^\top T) L (\mathbf{r}_2^\top T)^\top = 1, \quad \text{same computation on } T' = 0$$

T represents a tuple; L says which tuples are true. The binding keeps the arguments apart so that L has something well formed to adjudicate.

Why It Works, and What It Costs

The clean recovery depended on one thing: $\mathbf{r}_1^\top \mathbf{r}_2 = 0$. The roles were **orthonormal**.

Let them not be. Take roles with $\mathbf{r}_1 \cdot \mathbf{r}_2 = 0.3$, both of unit length:

$$\mathbf{r}_1^\top (\mathbf{r}_1 \mathbf{f}_1^\top + \mathbf{r}_2 \mathbf{f}_2^\top) = \mathbf{f}_1^\top + 0.3 \mathbf{f}_2^\top$$

Crosstalk. The retrieved filler is contaminated by every other binding, in proportion to how non-orthogonal the roles are.

The trade-off that produces VSA

One-hot roles are exactly orthogonal, but you get only n of them in n dimensions and the representation grows as n^2 . Vector symbolic architectures give up exact orthogonality for *near*-orthogonality – random vectors at $\cos \approx 0.8/\sqrt{d}$ – and buy exponentially many roles, plus a binding that stays the same size as its operands.

Circular convolution is that compressed outer product, met earlier as the folded sum of anti-diagonals.

When the Columns Are Not Individuals

Nothing required both indices to range over $\mathcal{D}_e$. Let the columns index *events* instead:

$\mathbf{as}^\top$ becomes $\{(Alice, e) : e \text{ is a sleeping event}\}$

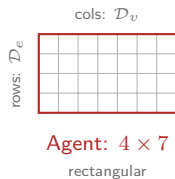
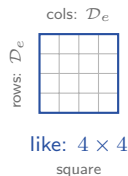
That is a **thematic role table**: $\text{Agent}(Alice, e)$ for each event e in the extension of sleep.

And now the rank-1 structure says something true rather than something defective. A role assignment *ought* to factor: it is a claim about a participant and a claim about an event, joined. The relation carries no further information, and it should not.

Notice that the two indices now run over *different domains*. The next two slides are about what that changes.

Two Sorts Make the Matrix Rectangular

A **sort** is the domain an index ranges over. Every matrix so far had one sort on both indices. Agent has two: $\mathcal{D}_e$ for individuals, $\mathcal{D}_v$ for events. (v for eventualities, the Davidsonian sort.)



Four individuals, seven sleeping events. Nothing makes those numbers agree. Its semantic type records both sorts: Agent : $\langle e, \langle v, t \rangle \rangle$.

Squareness was never a property of relations. It was an artifact of both arguments happening to be drawn from the same domain.

So $\mathbf{x}^\top M \mathbf{y}$ now takes $\mathbf{x} \in \mathbb{R}^4$ and $\mathbf{y} \in \mathbb{R}^7$: a bilinear form on $V \times W$, not $V \times V$. That row of the Part II taxonomy finally has an instance.

What Happens to the Transpose

	one sort: like	two sorts: Agent
M	$\langle e, \langle e, t \rangle \rangle, 4 \times 4$	$\langle e, \langle v, t \rangle \rangle, 4 \times 7$
$M^\top$	$\langle e, \langle e, t \rangle \rangle, 4 \times 4$	$\langle v, \langle e, t \rangle \rangle, 7 \times 4$
reads as	<i>is liked by</i>	<i>participants of</i>
same type as M ?	yes	no
is $M = M^\top$?	a real question	ill-formed
is $M + M^\top$?	a matrix	does not typecheck

With one sort, transposing lands you back in the same space – an argument swap. With two, it moves you to a *different space* and a different question:

M : which events did Alice take part in?

$M^\top$: who took part in this event?

A distinction worth keeping. The forms in Part V *fail* symmetry: they could have been symmetric and are not. A rectangular relation does not fail symmetry – **the predicate does not apply to it at all**. Two different kinds of asymmetry.

$MM^\top$ is 4×4 , people who *co-participated*; $M^\top M$ is 7×7 , events that *share a participant*. Two sizes, two subjects.

Multiplying Is Existential Quantification

One rule: **sum over the shared index** – existential closure over the middle argument. What comes back depends on the arity of the right operand.

A unary predicate on the right is a vector. $\mathcal{V}(\text{student}) = (1, 1, 1, 0)$, one entry per individual:

$$(L \mathcal{V}(\text{student}))_i = \sum_j L_{ij} \mathcal{V}(\text{student})_j \iff \exists j. \text{like}(i, j) \wedge \text{student}(j)$$

A binary predicate on the right is a matrix. $O_{jk} = 1$ iff j owns k :

$$(L O)_{ik} = \sum_j L_{ij} O_{jk} \iff \exists j. \text{like}(i, j) \wedge \text{own}(j, k)$$

	right operand	the operation	free after $\sum_j$	result
type	$\langle e, t \rangle$	$\langle e, \langle e, t \rangle \rangle \times \langle e, t \rangle \rightarrow \langle e, t \rangle$	i	$\langle e, t \rangle$, a set
shape	$n \times 1$	$(n \times n) (n \times 1) \rightarrow (n \times 1)$		a vector
type	$\langle e, \langle e, t \rangle \rangle$	$\langle e, \langle e, t \rangle \rangle \times \langle e, \langle e, t \rangle \rangle \rightarrow \langle e, \langle e, t \rangle \rangle$	i, k	$\langle e, \langle e, t \rangle \rangle$, a relation
shape	$n \times n$	$(n \times n) (n \times n) \rightarrow (n \times n)$		a matrix

One shape rule covers both: $(m \times n)(n \times p) \rightarrow (m \times p)$. The inner n cancels – that cancellation is $\sum_j$ – and a vector is the case $p = 1$.

The Same Product, Worked at Both Arities

Unary on the right. $\text{student} = \{\text{Alice}, \text{Bob}, \text{Carol}\} = (1, 1, 1, 0)$, as since the refresher. Row i asks: does i like *anyone* who is a student?

i	likes	of whom	students	entry
Alice	Bob	Bob		1
Bob	Carol	Carol		1
Carol	Fido	–		0
Fido	Fido	–		0

$$L\mathcal{V}(\text{student}) = (1, 1, 0, 0) = \{\text{Alice}, \text{Bob}\} \quad \text{likes a student – a set}$$

Binary on the right. $\text{own} = \{(\text{C}, \text{F})\}$, Carol owns Fido. Now ask for which *pairs* (i, k) there is a j with i liking j and j owning k . Only $j = \text{Carol}$ owns anything, and only Bob likes Carol:

$$LO = \{(\text{Bob}, \text{Fido})\} \quad \text{likes someone who owns } k \text{ – a relation}$$

The entry *counts* witnesses. Threshold at ≥ 1 for the existential; at $= |\text{student}|$ for *likes every student* – both quantifier tests from Part I, out of one product.

Walk-Through: *Bob likes someone who owns Fido*

Both relations as matrices, rows indexed by the first argument, columns by the second, in the fixed order A, B, C, F:

$$L = \begin{array}{c} \begin{array}{cc} & \begin{array}{cccc} A & B & C & F \end{array} \\ \begin{array}{c} A \\ B \\ C \\ F \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \quad O = \begin{array}{c} \begin{array}{cc} & \begin{array}{cccc} A & B & C & F \end{array} \\ \begin{array}{c} A \\ B \\ C \\ F \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

One entry of the product. $(LO)_{ik} = \sum_j L_{ij} O_{jk}$: take **row Bob of L** and **column Fido of O** , multiply position by position, add. Each position is one candidate j for the middle argument:

$$(LO)_{B,F} = \underbrace{L_{BA} O_{AF}}_{\substack{j=\text{Alice} \\ 0 \cdot 0}} + \underbrace{L_{BB} O_{BF}}_{\substack{j=\text{Bob} \\ 0 \cdot 0}} + \underbrace{L_{BC} O_{CF}}_{\substack{j=\text{Carol} \\ 1 \cdot 1}} + \underbrace{L_{BF} O_{FF}}_{\substack{j=\text{Fido} \\ 0 \cdot 0}} = 1$$

A term is 1 only when *both* factors are: Bob likes j and j owns Fido. Three candidates fail on both counts; Carol passes both. The sum counts the survivors, here exactly one. Repeating this for all sixteen (i, k) pairs fills in LO , and every other entry is 0: $LO = \{(\text{Bob}, \text{Fido})\}$.

The sum over j is the existential, $\exists j$. $\text{like}(\text{Bob}, j) \wedge \text{own}(j, \text{Fido})$. Multiplication is the $\wedge$; addition is the $\vee$; the index that disappears is the variable that got bound.

In-Class Exercise 3: Reading Operations

With $\mathcal{D}_e = \{A, B, C, F\}$, like $= L$ as above, $\mathcal{V}(\text{sleep}) = (0, 0, 1, 1)$ and $\mathcal{V}(\text{run}) = (1, 0, 0, 1)$:

- 1 Compute $\mathcal{V}(\text{sleep}) \mathcal{V}(\text{run})^\top$. How many nonzero entries? Give the relation in English and state its rank without computing an SVD.
- 2 What is $\mathcal{V}(\text{sleep})^\top \mathcal{V}(\text{run})$, and what does that number mean? [Why is it not a truth value?](#)
- 3 Compute $\mathbf{e}_C^\top (\mathbf{e}_A \mathcal{V}(\text{sleep})^\top)$. Explain the answer without doing any arithmetic.
- 4 $L^\top L$ is symmetric. Give its entries a reading in English. [What does the diagonal count?](#)
- 5 ★ $\mathbf{as}^\top$ has rank 1 and so does $\mathbf{sa}^\top$. They are different relations. Which linear-algebraic fact distinguishes them, and which semantic one?

Six minutes. Item 5, the starred one, is worth arguing about.

In-Class Exercise 3: Answers

1. Four nonzero entries: $\{C, F\} \times \{A, F\}$ – *the relation between any sleeper and any runner*. Rank 1, because it is an outer product; no SVD required.
2. $\mathcal{V}(\text{sleep})^\top \mathcal{V}(\text{run}) = 0 + 0 + 0 + 1 = 1$. It is $|\text{sleep} \cap \text{run}|$ – the number of individuals who do both, namely Fido. *Not a truth value but a count; it happens to equal 1 here. Thresholding at ≥ 1 would give some sleeper runs.*
3. $(0, 0, 0, 0)$. The matrix has one live row, Alice's, and $\mathbf{e}_C^\top$ selects row Carol. *Contracting with the wrong cue returns nothing, because the one-hot vectors are orthogonal – which is precisely the property that makes role-based retrieval work.*
4. $(L^\top L)_{jk} = \sum_i L_{ij} L_{ik} = |\{i : i \text{ likes } j \text{ and } i \text{ likes } k\}|$ – the number of common admirers of j and k . *The diagonal counts how many people like j : the in-degree.* This is a co-occurrence matrix, built from a relation exactly as Session 7 built one from a corpus.
5. $\star \quad \mathbf{s}\mathbf{a}^\top = (\mathbf{a}\mathbf{s}^\top)^\top$: same rank, same singular value, transposed. *Semantically the arguments have swapped roles* – Alice was the first argument and is now the second. That is the non-symmetry from Part V, seen at rank 1.

Interlude: Events and Their Participants

Earlier in Part IV, a role table $\text{Agent}(\text{Alice}, e)$ appeared as a 4×7 matrix with individuals down the side and events across the top. This interlude says where that object comes from in semantics, and shows that every step of the semantic analysis has a matrix step that computes it.

1	The problem	adjuncts, and why a fixed-arity verb cannot hold them
2	The solution	events as individuals; named roles AG, PAT, INSTR
3	Types and lambdas	the verb, its arguments, its adjuncts, and closure
4	The same thing in matrices	one role, one matrix; conjunction is $\odot$; $\exists$ is a sum
5	What it buys	the two-place table L is an event matrix with the event summed out

The dictionary at the end is the point: a type on the left, a shape on the right, and the composition rules of each column line up.

The Problem: A Verb Does Not Know How Many Arguments It Has

Take the sentence and strip modifiers one at a time:

- (1) Alice buttered the toast in the kitchen with a knife at midnight.
- (2) Alice buttered the toast with a knife.
- (3) Alice buttered the toast.

Every speaker knows (1) entails (2) entails (3). A first-order rendering with one predicate per sentence cannot say why:

$$\text{butter}_5(a, t, \text{kitchen}, k, \text{midnight}) \not\models \text{butter}_3(a, t, k) \not\models \text{butter}_2(a, t)$$

Three predicates, three arities, no logical relation among them. You would need a meaning postulate for every pair, and there is no bound on how many modifiers can be stacked.

Davidson's observation (1967)

The modifiers are not arguments of the verb. They are predicates of *something else* – the event the verb reports – and the entailments are just **conjunct dropping**.

In matrix terms, which we will reach: a verb with n arguments is an n -way table, and n is not fixed. Something has to factor it.

The Solution: Events Are Individuals; Participants Are Relations to Them

Add a sort v of **events** to the domain, and give the verb a single event argument. The participants become separate two-place facts linking an individual to the event:

$$\exists e [\text{butter}(e) \wedge \text{AG}(e, a) \wedge \text{PAT}(e, t) \wedge \text{INSTR}(e, k) \wedge \text{LOC}(e, \text{kitchen}) \wedge \text{TIME}(e, \text{midnight})]$$

Now $(1) \Rightarrow (2) \Rightarrow (3)$ is nothing more than $\exists e[\varphi \wedge \psi] \Rightarrow \exists e[\varphi]$, and nothing about the verb changes as modifiers come and go.

role	reads as	in (1)
AG (agent)	the participant who does it	Alice
PAT (patient)	the participant it is done to, and changed	the toast
TH (theme)	the participant it is done to, unchanged	–
INSTR (instrument)	the thing it is done with	a knife
LOC, TIME	where, when	the kitchen, midnight
GOAL, SRC	toward what, from what	–

This is the *neo-Davidsonian* form (Parsons 1990): the verb is a one-place predicate of events, and *every* participant, argument or adjunct, enters through a named role. The role inventory is a matter of theory; the shape of the analysis is not.

Types

Three basic types now: e for individuals, v for events, t for truth values. Everything else is built from them by $\langle \cdot, \cdot \rangle$, exactly as before.

expression	type	what it denotes
a, t, k	e	an individual
e_1	v	an event
butter	$\langle v, t \rangle$	the set of buttering events
AG, PAT, INSTR	$\langle e, \langle v, t \rangle \rangle$	which individual bears that role to which event
AG(a)	$\langle v, t \rangle$	the set of events Alice is agent of
$\lambda e. \text{butter}(e) \wedge \text{AG}(a)(e)$	$\langle v, t \rangle$	the set of butterings by Alice
$\exists e[\dots]$	t	a truth value: is that set nonempty?

Two conventions, fixed once. The role is applied to the individual first, $\text{AG}(a)(e)$, so that $\text{AG}(a)$ is a predicate of events; relationally that is $\text{AG}(e, a)$, the same fact. And a role is **functional in the event**: each event has at most one agent, one patient, one instrument. Earlier, Agent : $\langle e, \langle v, t \rangle \rangle$ was a 4×7 matrix, rows $\mathcal{D}_e$, columns $\mathcal{D}_v$. That is this type, as a shape.

The Verb as a Lambda Term

A transitive verb takes its object, then its subject, and returns a predicate of events. Its lexical entry packages the verb predicate with the two roles its arguments will fill:

$$\llbracket \text{butter} \rrbracket = \lambda y. \lambda x. \lambda e. \text{butter}(e) \wedge \text{AG}(x)(e) \wedge \text{PAT}(y)(e) \quad : \langle e, \langle e, \langle v, t \rangle \rangle \rangle$$

Apply it to the object, then the subject. Each application substitutes a constant for a bound variable and strips one λ :

$$\llbracket \text{butter} \rrbracket(t) = \lambda x. \lambda e. \text{butter}(e) \wedge \text{AG}(x)(e) \wedge \text{PAT}(t)(e) \quad : \langle e, \langle v, t \rangle \rangle$$

$$\llbracket \text{butter} \rrbracket(t)(a) = \lambda e. \text{butter}(e) \wedge \text{AG}(a)(e) \wedge \text{PAT}(t)(e) \quad : \langle v, t \rangle$$

What is left is a **set of events**, the butterings of the toast by Alice. Not yet a sentence; a sentence is type t .

Existential closure

The last step turns the event predicate into a truth value by asking whether any event satisfies it:

$$\text{EC} = \lambda P. \exists e [P(e)] \quad : \langle \langle v, t \rangle, t \rangle, \quad \text{so}$$

$$\text{EC}(\llbracket \text{butter} \rrbracket(t)(a)) = \exists e [\text{butter}(e) \wedge \text{AG}(a)(e) \wedge \text{PAT}(t)(e)].$$

Adjuncts Are Modifiers of the Event Predicate

with a knife does not appear in the verb's entry. It applies to the event predicate from outside, adding one conjunct and handing back a predicate of the same type:

$$\llbracket \text{with a knife} \rrbracket = \lambda P. \lambda e. P(e) \wedge \text{INSTR}(k)(e) \quad : \quad \langle \langle v, t \rangle, \langle v, t \rangle \rangle$$

The full derivation of (2), *Alice buttered the toast with a knife*, is then four steps:

	term	type
<i>butter the toast</i>	$\lambda x. \lambda e. \text{butter}(e) \wedge \text{AG}(x)(e) \wedge \text{PAT}(\mathbf{t})(e)$	$\langle e, \langle v, t \rangle \rangle$
<i>Alice butter the toast</i>	$\lambda e. \text{butter}(e) \wedge \text{AG}(\mathbf{a})(e) \wedge \text{PAT}(\mathbf{t})(e)$	$\langle v, t \rangle$
<i>... with a knife</i>	$\lambda e. \text{butter}(e) \wedge \text{AG}(\mathbf{a})(e) \wedge \text{PAT}(\mathbf{t})(e) \wedge \text{INSTR}(k)(e)$	$\langle v, t \rangle$
closure	$\exists e [\text{butter}(e) \wedge \text{AG}(\mathbf{a})(e) \wedge \text{PAT}(\mathbf{t})(e) \wedge \text{INSTR}(k)(e)]$	t

The modifier can be applied as many times as you like, in any order, and the type never changes: that is the whole of Davidson's solution, written as a type. Dropping a modifier removes one conjunct, and $\exists e[\varphi \wedge \psi] \Rightarrow \exists e[\varphi]$ does the rest.

A Small Model to Compute In

To the four individuals add the knife and the toast, and fix five events. Everything on the remaining slides is computed on this model.

$$\mathcal{D}_e = \{A, B, C, F, K, T\}$$

Alice, Bob, Carol, Fido, the knife,
the toast

$$\mathcal{D}_v = \{e_1, e_2, e_3, e_4, e_5\}$$

event	what happened	AG	PAT / INSTR
e_1	Alice buttered the toast with the knife	A	T / K
e_2	Bob buttered the toast	B	T / -
e_3	Carol fed Fido	C	F / -
e_4	Fido slept	F	-
e_5	Alice slept	A	-

The verb predicates are vectors over $\mathcal{D}_v$, one entry per event, as $\mathcal{V}(\text{student})$ was a vector over $\mathcal{D}_e$:

$$\mathcal{V}(\text{butter}) = (1, 1, 0, 0, 0) \quad \mathcal{V}(\text{feed}) = (0, 0, 1, 0, 0) \quad \mathcal{V}(\text{sleep}) = (0, 0, 0, 1, 1)$$

Type $\langle v, t \rangle$, shape 5×1 . Nothing new: a one-place predicate is a characteristic vector over its sort.

One Role, One Matrix

A role has type $\langle e, \langle v, t \rangle \rangle$: rows over $\mathcal{D}_e$, columns over $\mathcal{D}_v$, a 1 where the individual bears that role to that event. Three roles, three 6×5 matrices:

$$\text{AG} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} A \\ B \\ C \\ F \\ K \\ T \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad \text{PAT} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} A \\ B \\ C \\ F \\ K \\ T \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad \text{INSTR} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} A \\ B \\ C \\ F \\ K \\ T \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Three things to read off:

- **Each column has at most one 1.** That is functionality: one agent per event. $\text{AG} \mathbf{e}_{e_1} = \mathbf{e}_A$ returns the agent of e_1 as a one-hot vector, a partial application $\lambda x. \text{AG}(x)(e_1)$ that happens to pick out one individual.
- **Each row is a set of events.** $\mathbf{e}_A^\top \text{AG} = (1, 0, 0, 0, 1)$ is $\text{AG}(a)$, type $\langle v, t \rangle$: the events Alice is agent of. The other partial application.
- **Each matrix is a sum of bindings.** $\text{AG} = \mathbf{e}_A \mathbf{e}_{e_1}^\top + \mathbf{e}_B \mathbf{e}_{e_2}^\top + \dots$, one rank-1 term per true conjunct $\text{AG}(x)(e)$. The “rank-1 structure says something true” remark was about these terms.

Conjunction Is Entrywise Product; Closure Is a Sum

Every predicate of events in the derivation is a vector over $\mathcal{D}_v$. Conjoining two of them is asking, event by event, whether *both* hold – the entrywise product $\odot$. Closing with $\exists e$ is asking whether *any* entry survived – a sum, thresholded at ≥ 1 .

lambda / logic	matrix	type $\rightarrow$ shape
butter	$\mathcal{V}(\text{butter})$	$\langle v, t \rangle \rightarrow 5 \times 1$
AG(a)	$\text{AG}^\top \mathbf{e}_A$ (a row of AG)	$\langle v, t \rangle \rightarrow 5 \times 1$
$\lambda e. P(e) \wedge Q(e)$	$\mathbf{p} \odot \mathbf{q}$	$\langle v, t \rangle \times \langle v, t \rangle \rightarrow \langle v, t \rangle$
$\lambda P \lambda e. P(e) \wedge \text{INSTR}(\mathbf{k})(e)$	$\mathbf{p} \mapsto \mathbf{p} \odot \text{INSTR}^\top \mathbf{e}_K$	$\langle \langle v, t \rangle, \langle v, t \rangle \rangle \rightarrow 5 \times 5 \text{ diagonal}$
$\exists e [P(e)]$	$\mathbf{1}^\top \mathbf{p} \geq 1$	$\langle \langle v, t \rangle, t \rangle \rightarrow \text{sum, then threshold}$

Two rows deserve a second look. The modifier row says a $\langle \langle v, t \rangle, \langle v, t \rangle \rangle$ function is a **diagonal matrix**: multiplying by $\text{diag}(\mathbf{m})$ is $\odot \mathbf{m}$, and diagonal matrices commute, which is why adjuncts apply in any order. The closure row is the **same** sum over an index that was existential quantification earlier, with the event as the index. **So the three operations of the semantics – application, conjunction, existential closure – are contraction, $\odot$, and $\mathbf{1}^\top (\cdot)$. Nothing else is needed.**

Walk-Through: *Alice buttered the toast with the knife*

Each line of the lambda derivation, computed. The event predicates are vectors over $(e_1, \dots, e_5)$:

conjunct	vector	value
butter(e)	$\mathcal{V}(\text{butter})$	$(1, 1, 0, 0, 0)$
AG(a)(e)	$\text{AG}^\top \mathbf{e}_A$	$(1, 0, 0, 0, 1)$
PAT(t)(e)	$\text{PAT}^\top \mathbf{e}_T$	$(1, 1, 0, 0, 0)$
INSTR(k)(e)	$\text{INSTR}^\top \mathbf{e}_K$	$(1, 0, 0, 0, 0)$
all four conjoined: $\lambda e. [\dots]$	$\mathcal{V}(\text{butter}) \odot \text{AG}^\top \mathbf{e}_A \odot \text{PAT}^\top \mathbf{e}_T \odot \text{INSTR}^\top \mathbf{e}_K$	$(\mathbf{1}, 0, 0, 0, 0)$
closure: $\exists e [\dots]$	$\mathbf{1}^\top (\cdot)$	$\mathbf{1} \geq 1$: true

Column by column: e_1 is a buttering, by Alice, of the toast, with the knife, and survives all four factors. e_2 is a buttering of the toast but by Bob; the AG factor kills it. e_5 is Alice's but is a sleeping. The one survivor is the witness for $\exists e$. **The same sentence about Bob.** Swap $\mathbf{e}_A$ for $\mathbf{e}_B$: the AG factor becomes $(0, 1, 0, 0, 0)$, the product becomes $(0, \mathbf{0}, 0, 0, 0)$ once the INSTR factor is applied, and the sum is 0. *Bob buttered the toast with the knife* is false in the model; *Bob buttered the toast* is true: drop the INSTR factor and e_2 survives.

Entailment by conjunct dropping is now visible as arithmetic: removing a 0/1 factor from a $\odot$ chain can only raise entries, so the sum can only go up.

The Two-Place Table Is an Event Matrix With the Event Summed Out

Part IV began with L , a 4×4 table of who likes whom. Where does such a table come from, if verbs are predicates of events? Ask *who buttered what*: the pairs (x, y) for which some buttering has x as agent and y as patient.

$$\{(x, y) : \exists e [\text{butter}(e) \wedge \text{AG}(x)(e) \wedge \text{PAT}(y)(e)]\} \iff B = \text{AG} \text{diag}(\mathcal{V}(\text{butter})) \text{PAT}^\top$$

Read the product: $B_{xy} = \sum_e \text{AG}_{xe} \text{butter}_e \text{PAT}_{ye}$. The shared index is the event; summing over it is the existential; the three factors are the three conjuncts. It is the MN rule with the verb's diagonal in the middle. On the model:

$$B = \begin{matrix} & \begin{matrix} \text{A} & \text{B} & \text{C} & \text{F} & \text{K} & \text{T} \end{matrix} \\ \begin{matrix} \text{A} \\ \text{B} \\ \text{C} \\ \text{F} \\ \text{K} \\ \text{T} \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} = \{(\text{Alice}, \text{toast}), (\text{Bob}, \text{toast})\}$$

L for like is the same construction: $\text{AG} \text{diag}(\mathcal{V}(\text{like})) \text{PAT}^\top$ with the liking events summed out. The square tables of Part IV are what you have left once the event is forgotten – which is also why they cannot represent the instrument, the place, or the time.

Why Factor: The n -Way Table Versus n Matrices

The polyadic rendering $\text{butter}_3(x, y, z)$ is a three-way table: $|\mathcal{D}_e|^3 = 216$ cells, and a four-way table for the next modifier. The neo-Davidsonian rendering is three matrices of size $|\mathcal{D}_e| \times |\mathcal{D}_v|$ sharing one index:

	polyadic	neo-Davidsonian
object	one n -way table per arity	one matrix per role, any number of roles
cells	$ \mathcal{D}_e ^n$	$n \cdot \mathcal{D}_e \cdot \mathcal{D}_v $
add a modifier	new table, new arity, new postulates	one more matrix, same event index
entailment	stipulated	drop a factor

This is the rank slogan from the other side. A tensor butter_3 that happens to decompose as $\sum_e \text{AG}_{\cdot e} \otimes \text{PAT}_{\cdot e} \otimes \text{INSTR}_{\cdot e}$ – one rank-1 term per event – is exactly a relation that factors through events. The event index is what makes the factorization possible, and its size $|\mathcal{D}_v|$ is the number of rank-1 terms.

L for like did not factor as $\mathbf{uv}^\top$: a liking is not a fact about the liker times a fact about the liked. It *does* factor through the liking events, one rank-1 term each. Rank 1 per event, not overall.

The Dictionary

semantics	type	shape	operation
individual a	e	$\mathbf{e}_A \in \mathbb{R}^{ \mathcal{D}_e }$, one-hot	
event e_1	v	$\mathbf{e}_{e_1} \in \mathbb{R}^{ \mathcal{D}_v }$, one-hot	
verb butter	$\langle v, t \rangle$	vector in $\mathbb{R}^{ \mathcal{D}_v }$	
role AG	$\langle e, \langle v, t \rangle \rangle$	$ \mathcal{D}_e \times \mathcal{D}_v $ matrix	
$\text{AG}(a)$	$\langle v, t \rangle$	$\text{AG}^\top \mathbf{e}_A$	contraction on the individual
$\lambda x. \text{AG}(x)(e_1)$	$\langle e, t \rangle$	$\text{AG} \mathbf{e}_{e_1}$	contraction on the event
$\text{AG}(a)(e_1)$	t	$\mathbf{e}_A^\top \text{AG} \mathbf{e}_{e_1}$	full application; table lookup
$P \wedge Q$ on events	$\langle v, t \rangle$	$\mathbf{p} \odot \mathbf{q}$	entrywise product
modifier $\lambda P \lambda e. P(e) \wedge R(k)(e)$	$\langle \langle v, t \rangle, \langle v, t \rangle \rangle$	$\text{diag}(R^\top \mathbf{e}_K)$	diagonal matrix; commutes
$\exists e [P(e)]$	t	$\mathbf{1}^\top \mathbf{p} \geq 1$	sum, threshold
who- R_1 -what	$\langle e, \langle e, t \rangle \rangle$	$R_1 \text{diag}(\mathbf{v}) R_2^\top$	composition through the event

Every row pairs a semantic object with a linear-algebraic one, and the pairings respect composition: apply on the left, contract on the right; conjoin on the left, $\odot$ on the right; close on the left, sum on the right. [What this interlude set up for the rest of Part IV: the role matrices are what the next section's co-occurrence tables are an impoverished version of. A word-by-context count matrix is a role table with the roles collapsed and the events forgotten. PMI will measure what survives.](#)

Changing What the Matrix Is

Every matrix in this part so far has been a **relation over individuals**: both indices range over $\mathcal{D}_e$, and an entry says whether that pair stands in R . Roles and fillers had a job because there were argument positions to fill. The rest of Part IV changes the subject – rows will index words, columns context words, entries counts and then probabilities.

No argument positions in such a matrix, so no roles and no fillers. The thematic-role reading does not transfer, and nothing ahead needs it.

What carries over is the algebra: $M_{ij} = u_i v_j$ still says the two indices do not interact. Only the subject matter changes:

When the matrix is rank 1 says	
a relation on $\mathcal{D}_e$	the pair is two unary facts glued; no relational content
a joint distribution	$p(x, y) = p(x) p(y)$; the two variables are independent

One condition, two subjects – independence is that same condition, read on probabilities.

A Detour: Turning a Probability Into Bits

Before the next two slides, one piece of machinery. Define the **surprise** of an outcome:

$$h(x) = -\log_2 p(x) \quad \text{measured in **bits**}$$

$p(x)$	1	$\frac{1}{2}$	$\frac{1}{4}$	0.321
$h(x)$	0 bits	1 bit	2 bits	1.64 bits

Why a logarithm? Because surprise should *add* when outcomes are independent, and probabilities *multiply*. Two independent coin flips are twice as surprising as one, not four times:

$$p(x, y) = p(x)p(y) \implies h(x, y) = h(x) + h(y)$$

The log is the one function that turns multiplication into addition, which is the only reason it is here.

Read one bit as one yes/no question's worth of information. A certain outcome tells you nothing, and costs 0.

Average Surprise Is Entropy

$h(x)$ scores *one* outcome. Average it over the distribution and you get a single number for the whole variable:

$$H(X) = \sum_x p(x) h(x) = - \sum_x p(x) \log_2 p(x)$$

Here the variable X is “**which row word did we observe?**” Its three values are the row words of the word-pair table a few slides ahead, {coffee, tea, laptop}, and $p(x)$ is simply how often each occurs – the row marginal, $\mathbf{p} = (0.371, 0.307, 0.321)$. One observation is one word; nothing is flipped or rolled.

	x	coffee	tea	laptop	
$p(x)$	how often the word occurs	0.371	0.307	0.321	
$h(x) = -\log_2 p(x)$	surprise at seeing it	1.429	1.703	1.637	
$p(x) h(x)$	its share of the average	0.531	0.523	0.526	$H(X) = 1.580$

Read it as the average number of bits one observed word costs you, and so the best any code could do. Two anchors fix the scale:

$H = 0$	one word has $p = 1$	nothing to learn
$H = \log_2 n$	all n words equally frequent	maximally uncertain

If the three words were exactly equiprequent, H would be $\log_2 3 = 1.585$; the actual value is 1.580. They are very nearly equiprequent, so the marginal alone tells you almost nothing – which is exactly why the *joint* is where the information is.

PMI Is Surprise Saved

Take the table ahead: $p(\text{screen}) = 0.321$ overall, but $p(\text{screen} \mid \text{laptop}) = 0.286/0.321 = 0.889$.

Seeing screen cold	$-\log_2 0.321$	$= 1.637$ bits of surprise
Seeing it after laptop	$-\log_2 0.889$	$= 0.170$ bits
Saved by knowing the row		$= 1.468$ bits

That saving is exactly PMI, and the algebra is one line of cancellation:

$$\text{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)} = \log_2 \frac{p(y \mid x)}{p(y)} = h(y) - h(y \mid x)$$

$\text{PMI} > 0$	beats independence	knowing one helps
$\text{PMI} = 0$	exactly independence	knowing one tells you nothing
$\text{PMI} < 0$	rarer than independence	knowing one actively surprises

Pointwise is the operative word: PMI is a statement about *one cell*.

From One Cell to the Whole Table: Kullback–Leibler Divergence

The last slide scored *one cell*: the log ratio $\log_2 \frac{p(x,y)}{p(x)p(y)}$ compares how often a word pair actually occurs with how often a model predicts it. To score the *whole model* we need one number for the whole table.

Setting. Two probability distributions over the same cells of the word-pair table:

- $P(x, y)$ – the **true** distribution: the observed relative frequency of row word x with column word y .
- $Q(x, y)$ – a **model** of it: what some hypothesis says the frequencies should be. (Q could be anything; the next slide uses a specific one.)

Definition. The **Kullback–Leibler divergence** of Q from P is the log ratio of each cell, averaged over the cells with the *true* probabilities as weights:

$$\text{KL}(P \parallel Q) = \sum_{x,y} P(x, y) \log_2 \frac{P(x, y)}{Q(x, y)}$$

This is the same move as entropy: there we averaged the surprise $h(x)$ over $p(x)$; here we average the log ratio over $P(x, y)$. **Read it as:** *the extra bits per observed word pair you pay for using model Q when the truth is P .* If $Q = P$ every ratio is 1, every log is 0, and you pay nothing. You can never pay less than nothing, so $\text{KL} \geq 0$, with equality only when $Q = P$.

Notation: the double bar in $\text{KL}(P \parallel Q)$ is read “ P relative to Q .” The order matters; see the next slide.

KL Against the Independence Model Is Mutual Information

Now choose a specific model Q : the **independence model**, which says the row word and the column word have nothing to do with each other, so a pair is exactly as frequent as its two words' separate frequencies predict:

$$Q(x, y) = p(x) q(y) \quad (\text{as a table, } Q = \mathbf{p}\mathbf{q}^\top: \text{ the outer product of the two marginals})$$

With this Q , each cell's log ratio is exactly the PMI from two slides back, so the KL divergence from the independence model is the **PMI of each cell, averaged with the true probabilities as weights**:

$$\text{KL}(P \parallel \mathbf{p}\mathbf{q}^\top) = \sum_{x,y} P(x, y) \text{PMI}(x, y) = \text{average PMI} = 0.545 \text{ bits}$$

This quantity has a name: the mutual information $I(X; Y)$ of the two variables – the bits you save per observation by knowing the pair rather than the two words separately. It is 0 exactly when P is the independence model – when the table factors as $\mathbf{p}\mathbf{q}^\top$ (rank 1) – and positive otherwise.

KL is not a distance. On our table $\text{KL}(P \parallel Q) = 0.545$ but $\text{KL}(Q \parallel P) = 0.780$: the cost of modeling the truth by independence is not the cost of modeling independence by the truth. [Part V is about what dropping symmetry costs; this is a live example.](#)

Four Quantities, Two Questions

Everything in this detour is one of two moves: **pointwise or averaged**, and **one distribution or two**.

	one distribution	two distributions
one outcome	$h(x) = -\log_2 p(x)$ surprise	$\text{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)}$ pointwise mutual information
averaged under P	$H(X) = 1.580$ entropy	$I(X; Y) = \text{KL}(P \parallel \mathbf{pq}^\top) = 0.545$ mutual information

The bottom row is the top row averaged. [So the identity on the PMI slide,](#)

[PMI\(\$x, y\$ \) = \$h\(y\) - h\(y \mid x\)\$ \), averages to](#)

$$I(X; Y) = H(Y) - H(Y \mid X) = 1.560 - 1.015 = 0.545$$

[Surprise saved](#), averaged over the whole table. Guessing the column cold costs 1.560 bits; knowing the row leaves 1.015. The same 0.545 three ways: average PMI, a KL from independence, a drop in entropy.

PMI Measures Departure From Rank One

A joint distribution is a matrix. The independence model $p(x)p(y)$ is the outer product of its two marginals – and is therefore **rank 1**.

The data: $N = 140$ word-pair tokens from a corpus, each a row word seen next to a column word. A cell with count $c(x, y)$ has $p(x, y) = c(x, y)/N$: e.g. $c(\text{laptop}, \text{screen}) = 40$ gives $40/140 = 0.286$. The nine cells sum to 1.

	observed $p(x, y)$			rank-1 model $p(x)p(y)$		
	cup	mug	screen	cup	mug	screen
coffee	0.214	0.143	0.014	0.154	0.098	0.119
tea	0.179	0.107	0.021	0.127	0.081	0.099
laptop	0.021	0.014	0.286	0.133	0.085	0.103

$$\text{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)}$$

$$\text{PMI}(\text{laptop}, \text{screen}) = +1.47$$

$$\text{PMI}(\text{coffee}, \text{screen}) = -3.06$$

PMI is the entrywise log ratio of the observed matrix to the rank-1 independence model. It is a measurement of *how much the cell is not explained by two unary facts* – the same claim as the rank slogan, arriving from Session 7.

Two Different Best Rank-1 Matrices

A caution, because Session 8 gave us a different rank-1 matrix for the same P .

	Independence model $\mathbf{p}\mathbf{q}^\top$	Truncated SVD $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top$
minimizes	$\text{KL}(P \parallel \cdot)$	$\ P - \cdot\ _F$
row sums	(0.371, 0.307, 0.321) – the marginals exactly	(0.402, 0.332, 0.233) – not the marginals
$\ P - \cdot\ _F$	0.2770	0.2747
$\text{KL}(P \parallel \cdot)$	0.545 bits	0.592 bits

Each column wins on its own criterion and loses on the other. The SVD factor sums to 0.967, so it was rescaled before taking KL.

Eckart–Young guarantees the second column wins on Frobenius error, and it does, narrowly. But it is the *first* that PMI divides by, because PMI asks a probabilistic question: what would this cell be if the two variables were independent?

“Best rank 1” is not well defined until you say in which divergence. Both are rank 1, both are outer products, and they answer different questions. Change the criterion and the spectral answer stops applying – the same lesson as the weighted low-rank result.

The Catalog

Operation	Result	Semantic reading
$\mathbf{x}^\top \mathbf{y}$	scalar	application; $ X \cap Y $; a count, thresholded to a truth value
$M \mathbf{e}_j$	vector	partial application: $\lambda x. R(x, j)$
$\mathbf{e}_i^\top M$	vector	the other partial application: $\lambda y. R(i, y)$
$\mathbf{x}^\top M \mathbf{y}$	scalar	full application; table lookup on one-hot arguments
$\mathbf{x} \mathbf{y}^\top$	matrix	binding ; the Cartesian product $X \times Y$; always rank 1
$\mathbf{r}^\top (\cdot)$	vector	unbinding: recover the filler occupying role $\mathbf{r}$
$M + N$	matrix	superposition; union of relations, with counts
$M \odot N$	matrix	intersection of relations
$M^\top$	matrix	swap the arguments; re-curry the predicate
MN	matrix	relational composition: $\exists j. M(i, j) \wedge N(j, k)$
$M^\top M$	matrix	co-occurrence: shared-context counts; symmetric
$\text{tr}(M)$	scalar	how many individuals bear R to themselves
$\sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$	matrix	decomposition into rank-1 rectangles
$\log(M \oslash \mathbf{p} \mathbf{q}^\top)$	matrix	PMI: entrywise log of observed $\oslash$ independence model

$\odot$ and $\oslash$ are entrywise product and division, $(M \oslash N)_{ij} = M_{ij}/N_{ij}$; the last row is the PMI table, written as one operation.

Only the outer product builds a matrix out of two vectors, and contraction can never raise rank. Two operations raise the rank of a matrix that already has one: superposition, and PMI – the entrywise log sends a rank-2 count matrix to full rank, which is why PPMI must be re-factorized rather than read off. Every semantic question in this course is about which of these you may do, and with which matrix in the middle.

Not Every Form Deserves to Be Called Similarity

We now have a large family. Only part of it behaves the way “how similar are these?” ought to behave. Two conditions separate them.

Condition	Algebra	What it buys
symmetric	$G = G^T$	$B(\mathbf{x}, \mathbf{y}) = B(\mathbf{y}, \mathbf{x})$
positive definite	$\mathbf{x}^T G \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$	every nonzero vector has positive length

The next three slides take these one at a time, and then look at what happens when you drop each.

Symmetry: Does Order Matter?

$G = G^\top$ means $G_{ij} = G_{ji}$: the rate for pairing left- i with right- j equals the rate for the reverse.

A form that fails it. $G = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$, with $\mathbf{x} = (3, 1)$, $\mathbf{y} = (1, 3)$:

$$\mathbf{x}^\top G \mathbf{y} = 3(1)(1) + 2(3)(3) + 0 + 1(1)(3) = 3 + 18 + 3 = 24$$

$$\mathbf{y}^\top G \mathbf{x} = 1(1)(3) + 2(1)(1) + 0 + 1(3)(1) = 3 + 2 + 3 = 8$$

Same two vectors, two different numbers, depending on which one you put first.

For a *similarity* that is fatal – “how alike are $\mathbf{x}$ and $\mathbf{y}$ ” cannot have two answers. We will see shortly that it is exactly what you want for something else.

Every Form Is an Agreement Plus a Disagreement

The form just failed symmetry because $G_{12} = 2$ and $G_{21} = 0$: the pair (left-1, right-2) and the pair (left-2, right-1) get different rates. **Two numbers for one pair of slots.** Any two numbers can be rewritten as *what they agree on* plus *how they differ*:

$$a = \underbrace{\frac{a+b}{2}}_{\text{the average}} + \underbrace{\frac{a-b}{2}}_{\text{half the gap}}, \quad b = \underbrace{\frac{a+b}{2}}_{\text{the same average}} - \underbrace{\frac{a-b}{2}}_{\text{the same gap, flipped}}$$

For $a = 7$, $b = 3$: average 5, half-gap 2, and indeed $7 = 5 + 2$, $3 = 5 - 2$. The average is *shared* by both orders; the half-gap is *the same size* in both orders, with opposite sign. The halving is what makes the two pieces add back up to the original.

Now do it to every pair at once. $G^\top$ is G with every entry moved to its mirror position: $(G^\top)_{ij} = G_{ji}$. So adding G and $G^\top$ puts $a + b$ in *both* mirror positions, and subtracting puts $a - b$ in one and $b - a$ in the other:

$$G = \underbrace{\frac{1}{2}(G + G^\top)}_{S: S_{ij}=S_{ji}} + \underbrace{\frac{1}{2}(G - G^\top)}_{A: A_{ij}=-A_{ji}} \quad \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}}_S + \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}}_A$$

Check the off-diagonal, where the pair is $a = G_{12} = 2$ and $b = G_{21} = 0$: $S_{12} = S_{21} = \frac{2+0}{2} = 1$, $A_{12} = \frac{2-0}{2} = 1$, $A_{21} = \frac{0-2}{2} = -1$. **The average and the half-gap both come out 1 here, which is an accident of these particular two numbers** – for 7 and 3 they were 5 and 2. **The diagonal:** a diagonal cell is *its own* mirror, so there is no second number to average it against. It passes into S unchanged, and $A_{ii} = \frac{G_{ii}-G_{ii}}{2} = 0$. S is what G says *regardless of order*; A is what G says *about order*. The split is unique: there is exactly one symmetric and one antisymmetric matrix summing to G .

What a Non-Symmetric Form Hides

With the split in hand, a curious fact worth knowing. Write G as its two halves:

$$G = \underbrace{\frac{1}{2}(G + G^{\top})}_{S, \text{ symmetric}} + \underbrace{\frac{1}{2}(G - G^{\top})}_{\text{antisymmetric}}$$

For $G = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ we get $S = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. Now evaluate both on the *same* vector twice, $\mathbf{x} = (3, 1)$:

$$\mathbf{x}^{\top} G \mathbf{x} = 9 + 2(3)(1) + 0 + 1 = 16, \quad \mathbf{x}^{\top} S \mathbf{x} = 9 + 3 + 3 + 1 = 16$$

The two agree, always. The antisymmetric part contributes nothing when both arguments are the same vector.

So *lengths* cannot detect non-symmetry. Whatever the antisymmetric part is doing, it only shows up when the two arguments differ – which is to say, when you are comparing two things rather than measuring one.

Positive Definiteness: Is Every Length Real?

The question is not about G in the abstract. It is whether $d_G(\mathbf{u}, \mathbf{v}) = \sqrt{(\mathbf{u} - \mathbf{v})^\top G (\mathbf{u} - \mathbf{v})}$ deserves to be called a *distance between two meaning vectors*. That needs one guarantee:

$$\mathbf{z}^\top G \mathbf{z} > 0 \quad \text{for every } \mathbf{z} \neq \mathbf{0}$$

every nonzero *difference* has positive squared length

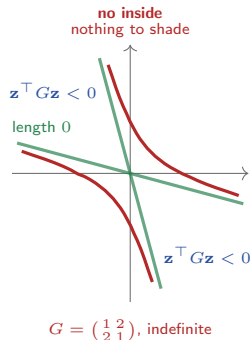
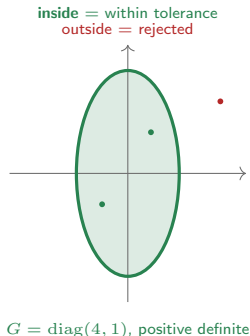
A symmetric form that fails it. Two word vectors in the polarity/topic coordinates from earlier: $\mathbf{u} = (1, 0)$, $\mathbf{v} = (0, 1)$, so $\mathbf{u} - \mathbf{v} = (1, -1)$. Take $G = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$, symmetric with positive diagonal:

$\mathbf{z}$	$\mathbf{z}^\top G \mathbf{z} = z_1^2 + 4z_1z_2 + z_2^2$	
$(1, 1)$	$1 + 4 + 1 = 6$	positive, as expected
$\mathbf{u} - \mathbf{v} = (1, -1)$	$1 - 4 + 1 = -2$	a negative squared distance

$d_G(\mathbf{u}, \mathbf{v}) = \sqrt{-2}$ – two words, an ordinary difference, and the retrieval code gets a complex number. Checking coordinates one at a time would never catch it: both diagonal entries are $+1$. The failure lives in a *direction*, where the cross-term $4z_1z_2$ overwhelms the rest. The eigenvalues are 3 and -1 ; we return to counting them in Part VII.

What That Failure Looks Like

Draw every $\mathbf{z}$ with $\mathbf{z}^\top G \mathbf{z} = 1$. For a good metric that curve *enclosed* a region: the shaded set, every $\mathbf{z}$ with $\|\mathbf{z}\|_G \leq 1$, is **the candidate set of every nearest-neighbor query you have written**.



The **green lines** are the real shock: nonzero differences of *length zero*, at $z_2/z_1 = -2 \pm \sqrt{3}$. Between those slopes the form is negative, -2 at $(1, -1)$. There is no region to shade, so “everything within ϵ of the query” names no set.

What Breaks: the Tolerance Region

With an ellipse, the set $\mathbf{z}^\top G \mathbf{z} \leq 1$ was a **bounded** neighborhood – *everything within this much of the query*. That set is the geometry behind every thresholded retrieval you have written. Here it is unbounded, and it gets worse.

Squared length scales quadratically, $(t\mathbf{z})^\top G(t\mathbf{z}) = t^2 \mathbf{z}^\top G \mathbf{z}$. Along the null-cone-interior direction $\mathbf{z} = (1, -1)$, where the form is already negative:

t	1	2	5	10	...
$\ t\mathbf{z}\ _G^2$	-2	-8	-50	-200	$\rightarrow -\infty$

The further you move in that direction, the *closer* you score, without bound. There is no ε for which “within ε of the query” names a bounded set of candidates.

Be precise about what this does and does not ruin. A finite candidate list still has a minimum, so `argmin` returns something and your code does not crash. What it returns has none of the guarantees you were relying on – not a triangle inequality, not a meaningful threshold, not even a real number for the distance it reports.

What Positive Definiteness Actually Buys

Four things you have been doing all term stop working without it. None of them is exotic.

Operation	What it needs	Where you use it
measure a difference	real, nonnegative d_G ; zero only at $\mathbf{u} = \mathbf{v}$	nearest neighbors
compare directions	$\cos_G$ lands in $[-1, 1]$	similarity ranking
define a neighborhood	$\mathbf{z}^\top G \mathbf{z} \leq 1$ is <i>bounded</i>	thresholded retrieval
score an error	every nonzero error is penalized	training with $L = \mathbf{e}^\top G \mathbf{e}$

The cosine failure, in numbers. Same G , same $\mathbf{u} = (1, 0)$ and $\mathbf{v} = (0, 1)$:

$$\cos_G(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top G \mathbf{v}}{\sqrt{\mathbf{u}^\top G \mathbf{u}} \sqrt{\mathbf{v}^\top G \mathbf{v}}} = \frac{2}{\sqrt{1} \cdot \sqrt{1}} = 2$$

A similarity of 2. *Cauchy–Schwarz normally forbids this, and it is a theorem about inner products: it holds for positive definite G and no others.* On the last row: positive definiteness makes $L = \mathbf{e}^\top G \mathbf{e}$ uniquely minimized at $\mathbf{e} = \mathbf{0}$ as a function of the error, and says nothing about the network's parameter landscape, which is not convex.

Three Jobs, Three Different Objects

A correction worth making explicitly, because it is the most common wrong inference here. Positive definiteness constrains $\langle \mathbf{x}, \mathbf{x} \rangle_G$. It says *nothing* about $\langle \mathbf{x}, \mathbf{y} \rangle_G$ for $\mathbf{x} \neq \mathbf{y}$:

$$\mathbf{x} = (1, 0), \quad \mathbf{y} = (-1, 0), \quad G = I : \quad \mathbf{x}^\top \mathbf{x} = 1 > 0 \quad \checkmark \quad \cos(\mathbf{x}, \mathbf{y}) = -1$$

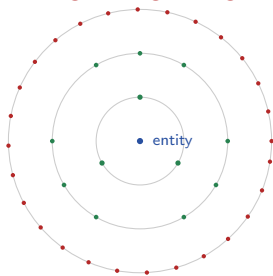
The plainest possible metric already permits maximal opposition. **So hot and cold receiving similar vectors is *not* a failure of positive definiteness.** It is a fact about what distributional training captured: both words occur with water, very, the -- tap, so contextual resemblance never separated antonymy from synonymy. No choice of metric repairs a distinction the representation discarded.

Question	Object	Required shape
how far apart are these?	distance d_G	symmetric, positive definite
how well do these match?	a score $\mathbf{x}^\top G \mathbf{y}$	no constraint needed
does this one entail that one?	a directed relation	must <i>not</i> be symmetric

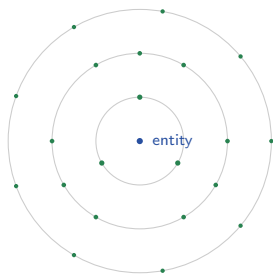
Three different jobs. The mistake is expecting one operation to do all three – which is exactly what asking cosine to model entailment amounts to.

Putting a Taxonomy in a Space: the Seating Problem

Put entity at the center. Its children sit on a ring at distance 1, grandchildren on a ring at distance 2, and so on. Siblings must not be on top of each other, so each needs its own stretch of ring. The question is whether the ring is long enough.



Euclidean: ring length $2\pi r$
27 nodes jammed onto a ring of length 19



Hyperbolic: ring length $2\pi \sinh r$
27 nodes on a ring of length 63 – room to spare

Both are drawn flat, so they look alike. They are not: in hyperbolic space the ring at radius 3 is genuinely three times longer. The next slide counts it.

The Count, and Why Euclidean Runs Out

A taxonomy with branching factor 3 needs 3^r places on the ring at depth r . Here is what each geometry offers.

depth r	1	3	5	8
siblings to seat, 3^r	3	27	243	6,561
Euclidean ring length, $2\pi r$	6	19	31	50
hyperbolic ring length, $2\pi \sinh r$	7	63	466	9,365

Euclidean space runs out at depth 3. Each step outward adds the *same* 2π of ring, while the tree demands three times as much; by depth 8 it is short by a factor of 130. **Hyperbolic ring length multiplies by roughly e each step, so it keeps pace. Match the growth rate of the space to that of the data.**

What crowding costs you: with nowhere to put them, the embedding places unrelated deep concepts side by side. **The distances lie exactly where the taxonomy is most specific** – the part you built it for. More Euclidean dimensions do add room; the hyperbolic claim is that you need far fewer of them for the same fidelity.

The Lorentz Model: Two Forms, Two Jobs

Hyperbolic space is built using an indefinite form, as a surface inside a larger space:

$$\langle \mathbf{x}, \mathbf{y} \rangle_L = -x_0 y_0 + \sum_{i \geq 1} x_i y_i, \quad \mathbb{H}^n = \{ \mathbf{x} : \langle \mathbf{x}, \mathbf{x} \rangle_L = -1, x_0 > 0 \}$$

A level curve of exactly the kind of form two slides back – the branch family opening the other way.

The distance on it is $d_{\mathbb{H}}(\mathbf{x}, \mathbf{y}) = \text{arcosh}(-\langle \mathbf{x}, \mathbf{y} \rangle_L)$, which is **real, symmetric, and zero only at $\mathbf{x} = \mathbf{y}$** .

	ambient form $\langle \cdot, \cdot \rangle_L$	distance $d_{\mathbb{H}}$ on the surface
signature	indefinite	positive definite
job	carve out the surface	measure on it
negative values?	yes, that is the point	never

You are not being asked to accept negative distances: the indefinite form is the *tool*, and what it produces is a well-behaved metric space. Embedding WordNet hypernymy this way (Nickel and Kiela, 2017–18) reaches with a handful of dimensions what Euclidean needs far more to approximate; generality is encoded by radius, entity near the origin and Chihuahua near the boundary.

What it does not fix: $d_{\mathbb{H}}$ is symmetric, so it cannot tell $\text{dog} \rightarrow \text{animal}$ from the reverse. It buys **room** for a hierarchy, not **direction** within it.

How to Check It, and Why You Usually Need Not

Two tests you can run by hand. For a symmetric G :

- 1 **Eigenvalues.** All strictly positive $\Rightarrow$ positive definite. Our bad G has 3 and -1 .
- 2 **Leading principal minors** (Sylvester's criterion). Every top-left determinant positive. Faster, and no eigenvalues required.

G	G_{11}	$\det G$	
$\text{diag}(4, 1)$	4	4	a metric
$\begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$	1	0.36	a metric
$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$	1	-3	not a metric

In practice you rarely write G down by hand. It arrives as $G = A^T A$ from something you measured, and then $\mathbf{z}^T G \mathbf{z} = \|A\mathbf{z}\|^2 \geq 0$ is automatic. Note the $\geq$. The next slide is about that gap.

Semi-definite Is the Common Case

$G = A^T A$ gives ≥ 0 , never > 0 on its own. Strict positivity needs A to have full column rank, since otherwise some $\mathbf{z} \neq \mathbf{0}$ has $A\mathbf{z} = \mathbf{0}$ – and that $\mathbf{z}$ is a nonzero difference of length zero.

Symmetric G	What it does to differences	
positive definite	every nonzero difference has positive length	a metric
semidefinite, singular	some nonzero differences have length 0	often deliberate
indefinite	some differences have negative value	not a metric

The middle row is the normal case, not an edge case. A covariance matrix estimated from fewer examples than dimensions is singular: 300-dimensional embeddings from 50 sentences leaves at least 250 directions of length zero. Truncated SVD to k components does the same thing deliberately, as Part VII will say again.

Whether that is a bug depends on *which* distinctions the null directions are. Losing one that encoded typography is a feature; losing one that separated two senses of bank is not, and the geometry cannot tell you which happened. When you need strict positivity, the standard repair is $G = A^T A + \varepsilon I$. And “it came from data, so it is fine” is half right: such a G is never indefinite, but it may well be singular.

Inner Product: Both Conditions Together

Definition

A symmetric positive definite bilinear form is an **inner product**. It gives you a norm and an angle:

$$\|\mathbf{x}\|_G = \sqrt{\mathbf{x}^\top G \mathbf{x}}, \quad \cos_G(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top G \mathbf{y}}{\|\mathbf{x}\|_G \|\mathbf{y}\|_G}$$

Cauchy–Schwarz holds for any such G :

$$|\mathbf{x}^\top G \mathbf{y}| \leq \|\mathbf{x}\|_G \|\mathbf{y}\|_G$$

so that $\cos_G$ really does land in $[-1, 1]$ and can be read as an angle.

This is the reassuring part. Every geometric fact you are used to – triangle inequality, orthogonality, projection, the Pythagorean theorem – survives the change of metric. You are not in a stranger place. You are in the same kind of place, with different exchange rates.

The Other Column: Forms That Are Relations

Drop symmetry and you lose similarity. You gain something we have already used.

Back to like from the refresher, with $\text{like} = \{(A, B), (B, C), (C, F), (F, F)\}$:

$$\mathbf{e}_A^\top M \mathbf{e}_B = 1 \quad (\text{Alice likes Bob}), \quad \mathbf{e}_B^\top M \mathbf{e}_A = 0 \quad (\text{Bob likes Alice})$$

	symmetric G	non-symmetric G
Models	similarity, distance	a directed relation
Order of arguments	irrelevant	carries the meaning
Example	cos, any inner product	like; entailment; attention

Alice likes Bob and *Bob likes Alice* are different claims. A symmetric form is structurally incapable of telling them apart. Non-symmetry is not a defect here – it is the feature.

Attention Is a Non-Symmetric Bilinear Form

The attention score between two tokens, written the usual way:

$$\text{score}(i, j) = \mathbf{q}_i \cdot \mathbf{k}_j = (W_Q \mathbf{x}_i) \cdot (W_K \mathbf{x}_j)$$

Move the transposes:

$$(W_Q \mathbf{x}_i)^\top (W_K \mathbf{x}_j) = \mathbf{x}_i^\top \underbrace{(W_Q^\top W_K)}_G \mathbf{x}_j$$

The query and key projections are a bilinear form on the token representations, and $G = W_Q^\top W_K$ is **not symmetric**, because W_Q and W_K are separate learned matrices.

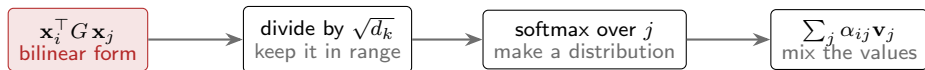
Which is why token i attending to j differs from j attending to i – directionality usually called architectural, and it is the non-symmetry of a bilinear form, nothing else.

G here is *learned* – nobody chose it – and each head learns its own G_h , so a layer with H heads asks H different questions about the same pair of tokens.

And from Part IV: with $d = 768$, $d_k = 64$, each G_h is 768×768 of rank at most 64 – a head compares the two tokens through a 64-dimensional slice, and which slice is what it learned.

Where the Form Sits in the Whole Mechanism

The bilinear form is one step of four. It is worth seeing which one, because the other three do no comparing at all.



- **Step 1 is the only place two tokens meet.** Everything downstream acts on one row at a time.
- The $\sqrt{d_k}$ is bookkeeping: a dot product of d_k random coordinates grows like $\sqrt{d_k}$, and without the division softmax saturates and gradients vanish. No semantics in it.
- Softmax is a normalizer. It can reweight what step 1 found; it cannot find anything step 1 missed.

So every question of the form *what does this head attend to* is a question about G , and the rest of the pipeline is plumbing.

Attention by Hand

Take $d = d_k = 2$ and the very matrix from the symmetry slide. With $W_Q = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ and $W_K = \begin{pmatrix} 2 & 3 \\ -1 & -1 \end{pmatrix}$:

$$G = W_Q^\top W_K = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$$

Three tokens, $\mathbf{x}_1 = (3, 1)$, $\mathbf{x}_2 = (1, 3)$, $\mathbf{x}_3 = (1, 1)$. Every score is $\mathbf{x}_i^\top G \mathbf{x}_j$:

$i \backslash j$	1	2	3
1	16	24	10
2	8	16	6
3	6	10	4

after softmax	1	2	3
1	0.003	0.996	0.000
2	0.003	0.996	0.001
3	0.055	0.932	0.013

Token 1 scores token 2 at 24; token 2 scores token 1 at 8. Same pair, same G , three times the score one way. That is the whole of “attention is directional,” computed by hand.

Note the diagonal is not the largest entry anywhere. A token need not attend mostly to itself, and nothing in the construction says it should.

The Two Halves of an Attention Score

Apply the split from earlier in this part to the G we just used:

$$G = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}}_{S \text{ symmetric}} + \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}}_{A \text{ antisymmetric}}$$

Evaluate each half on $\mathbf{x}_1 = (3, 1)$ and $\mathbf{x}_2 = (1, 3)$:

	$1 \rightarrow 2$	$2 \rightarrow 1$	
symmetric half $\mathbf{x}^\top S \mathbf{y}$	16	16	how alike they are
antisymmetric half $\mathbf{x}^\top A \mathbf{y}$	+8	-8	which way it runs
total	24	8	

S carries all the similarity and cannot tell the two directions apart. A carries no similarity – it vanishes on $\mathbf{x}^\top A \mathbf{x}$ – and all the direction, flipping sign on a swap.

Had G been symmetric, token 1 would score itself and token 2 at 16 apiece: a dead tie, 0.499 each.

The ± 8 is what breaks it.

In 2D, $\mathbf{x}^\top A \mathbf{y} = x_1 y_2 - x_2 y_1$ is signed area: the wedge product, where geometric algebra starts.

A Question You Should Be Asking

We have seen that changing G changes the answers. Here is the other half of the story.

Could I get the same effect by leaving $G = I$ alone
and changing the *vectors* instead?

The answer is yes – always, for any inner product. Which means the metric and the coordinates are two descriptions of one situation.

An everyday version: measuring a room in feet and getting 12×10 , or measuring in meters and getting 3.66×3.05 . The room did not change. The numbers did, because the ruler did.

The next four slides make that trade-off exact, because it is the mechanism behind what happens in the following deck.

Changing the Coordinates Moves the Metric

Suppose you re-express every vector in new coordinates, $\mathbf{x} \mapsto A\mathbf{x}$ for some invertible A . If the *values* of the form are to be unchanged, the middle must move the opposite way:

$$G \longmapsto A^{-\top} G A^{-1}$$

Check it. Take $A = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$ and $G = I$. Then $A\mathbf{x} = (4, 2)$ and $A\mathbf{y} = (4, 6)$, and

$$A^{-\top} I A^{-1} = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$$

$$(4, 2) \begin{pmatrix} 1 & -0.5 \\ -0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 4 \\ 6 \end{pmatrix} = 6.000 = \mathbf{x}^{\top} I \mathbf{y}$$

New vectors, new matrix in the middle, identical answer. The two changes cancelled exactly.

The Tool for Running It All the Way: Cholesky

If G is symmetric positive definite, it factors as

$$G = LL^{\top}$$

with L lower triangular. This is the **Cholesky factorization** – a matrix square root.

Worked, 2×2 . For $G = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$, write $L = \begin{pmatrix} \ell_{11} & 0 \\ \ell_{21} & \ell_{22} \end{pmatrix}$ and multiply out:

$$LL^{\top} = \begin{pmatrix} \ell_{11}^2 & \ell_{11}\ell_{21} \\ \ell_{11}\ell_{21} & \ell_{21}^2 + \ell_{22}^2 \end{pmatrix}$$

Match entries in order:

$$\ell_{11}^2 = 4 \Rightarrow \ell_{11} = 2; \quad 2\ell_{21} = 2 \Rightarrow \ell_{21} = 1; \quad 1 + \ell_{22}^2 = 3 \Rightarrow \ell_{22} = \sqrt{2} = 1.4142$$

$$L = \begin{pmatrix} 2 & 0 \\ 1 & 1.4142 \end{pmatrix}$$

Positive definiteness is exactly what guarantees those square roots are of positive numbers.

Absorbing the Metric Into the Vectors

With $G = LL^\top$ in hand, define new coordinates $\mathbf{x}' = L^\top \mathbf{x}$. Then:

$$\mathbf{x}' \cdot \mathbf{y}' = (L^\top \mathbf{x})^\top (L^\top \mathbf{y}) = \mathbf{x}^\top \underbrace{LL^\top}_G \mathbf{y} = \mathbf{x}^\top G \mathbf{y}$$

What that says

The **plain dot product** of the primed vectors equals the G -form of the original ones. The metric has been moved out of the middle and into the coordinates.

This is the same trade-off as the previous slide, run all the way in one direction: push *all* the structure into the vectors, and the middle becomes I .

Verified, With the Numbers

$G = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$, so $L^\top = \begin{pmatrix} 2 & 1 \\ 0 & 1.4142 \end{pmatrix}$. Apply it:

$$\mathbf{x} = (3, 1) \mapsto \mathbf{x}' = (2(3) + 1(1), 1.4142(1)) = (7, 1.4142)$$

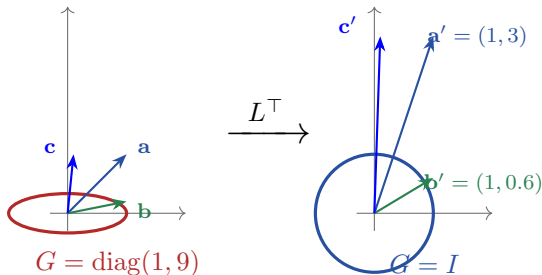
$$\mathbf{y} = (1, 3) \mapsto \mathbf{y}' = (2(1) + 1(3), 1.4142(3)) = (5, 4.2426)$$

	original, with G	primed, with I
the form	$\mathbf{x}^\top G \mathbf{y} = 41.000$	$\mathbf{x}' \cdot \mathbf{y}' = 35 + 6 = 41.000$
the cosine	$\cos_G(\mathbf{x}, \mathbf{y}) = 0.875517$	$\cos(\mathbf{x}', \mathbf{y}') = 0.875517$

Identical, to every digit. In the new coordinates the ordinary dot product is exactly right.

Absorption, Drawn

The ranking example: $G = \text{diag}(1, 9)$, so $L^\top = \text{diag}(1, 3)$, and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ move.



The unit ball is a flat ellipse. The ordinary dot product is *not* the right comparison here, and by eye $\mathbf{b}$ looks closer to $\mathbf{a}$.

The unit ball is a circle, so the ordinary dot product is now exactly right – and $\mathbf{a}'$, $\mathbf{c}'$ are visibly close while $\mathbf{b}'$ is off on its own.

The right-hand picture was the metric's opinion all along. $\cos(\mathbf{a}', \mathbf{c}') = 0.9587$, the number we computed the long way.

What That Result Actually Says

Every finite-dimensional inner product space is isomorphic to $\mathbb{R}^n$ with the standard dot product. There is only one, up to a change of basis.

So a nonstandard metric is never a different *kind* of geometry. It is the same geometry seen in inconvenient coordinates, and L is the instruction for fixing them.

With one condition attached, and it is the whole point:

you have to know G .

If a system computes similarities with $\mathbf{x} \cdot \mathbf{y}$ while the geometry it actually learned has some other G in the middle, its numbers are not slightly wrong. They are answers to a question about a different space. The correction exists and is one matrix multiply away – but only for someone holding G .

That is the next deck, in one sentence.

Dropping Positive Definiteness, Deliberately

So far, failing positive definiteness has looked like a bug. In two settings it is the intended design, so it is worth seeing what such a form does.

Degenerate. $G = \text{diag}(1, 0)$, and take $\mathbf{v} = (0, 5)$:

$$\mathbf{v}^\top G \mathbf{v} = 1(0)^2 + 0(5)^2 = 0$$

A nonzero vector of length zero. The second coordinate has been made entirely invisible to the form – not down-weighted, *erased*.

Vectors like this are called **null vectors**. Note that $(0, 5)$ is also orthogonal to itself: $\mathbf{v}^\top G \mathbf{v} = 0$ is what orthogonality means.

Truncated SVD does exactly this, incidentally. Setting $\sigma_{k+1}, \dots, \sigma_n$ to zero makes those directions null.

Indefinite: Negative Lengths

Return to $G = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$, the symmetric form that failed positive definiteness. Its quadratic form is $x_1^2 + 4x_1x_2 + x_2^2$:

$\mathbf{x}$	$\mathbf{x}^\top G \mathbf{x}$	
$(1, 1)$	+6	positive
$(1, -1)$	-2	negative
$(1, \sqrt{3} - 2)$	0	nonzero, but null

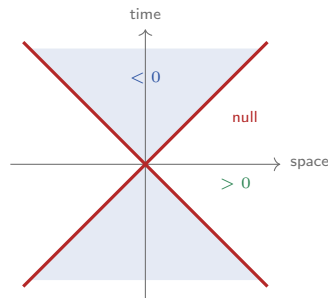
Check the last one: $\sqrt{3} - 2 = -0.2679$, and
 $1 + 4(-0.2679) + (-0.2679)^2 = 1 - 1.0718 + 0.0718 = 0$.

Such a form is **indefinite**: some directions have positive squared length, some negative, and a surface of them have zero. Nothing has broken. It is simply not an inner product, and “length” has to be read differently.

The Famous Indefinite Form

$G = \text{diag}(1, 1, 1, -1)$ – three space directions and one time direction. This is the **Minkowski** metric.

$\mathbf{u}$	$\mathbf{u}^\top G \mathbf{u}$	
$(1, 0, 0, 0)$	+1	spacelike
$(0, 0, 0, 1)$	-1	timelike
$(1, 0, 0, 1)$	0	null
$(3, 0, 0, 5)$	-16	timelike



The null vectors form a cone – the light cone. Two distinct events separated by a null vector are, in this geometry, at zero distance from each other.

Counting the Signs: Sylvester's Law

Any symmetric G can be brought by a change of basis to a diagonal matrix whose entries are only $+1$, -1 and 0 .

Sylvester's law of inertia

The *counts* of $+1$, -1 and 0 are the same no matter which change of basis you use. They are properties of the form, not of the coordinates.

Example. $G = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ has eigenvalues 3 and -1 . Dividing each eigenvector by the square root of $|\text{its eigenvalue}|$ turns those into $+1$ and -1 :

$$\text{diagonal form} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{one } +, \text{ one } -, \text{ no zeros}$$

No change of coordinates will ever make this form positive definite. You can move the numbers around; you cannot change a minus sign into a plus.

Signature, and a Notation You Will See Again

Those three counts are called the **signature** of the form, written $\text{Cl}(p, q, r)$: p directions squaring to $+1$, q to -1 , r to 0 .

Signature	Diagonal form	Name
$\text{Cl}(n, 0, 0)$	$(+1, \dots, +1)$	Euclidean; an inner product
$\text{Cl}(1, 1, 0)$	$(+1, -1)$	our $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
$\text{Cl}(3, 1, 0)$	$(+1, +1, +1, -1)$	Minkowski
$\text{Cl}(p, q, r)$	r zeros on the diagonal	degenerate; r null directions

That notation is not new machinery waiting to be learned. $\text{Cl}(p, q, r)$ is a name for a bilinear form, written by counting signs.

When we build a geometric algebra later in the course, the first thing declared is the signature – because every product in the algebra is defined relative to G , and until G is fixed there is nothing to define.

Where You Have Already Met These

Setting	Form	G	Symmetric?
Session 6B relations	$\mathbf{x}^\top M \mathbf{y}$	the relation itself	no
Cosine similarity	$\mathbf{v}^\top I \mathbf{w}$	assumed I	yes
tf-idf, PPMI	$\mathbf{x}^\top \text{diag}(\mathbf{w}) \mathbf{y}$	a chosen diagonal weighting	yes
SGNS score	$\mathbf{w}_w^\top I \mathbf{c}_c$	assumed I	yes
Attention	$\mathbf{x}_i^\top (W_Q^\top W_K) \mathbf{x}_j$	learned, and not symmetric	no
Mahalanobis distance	$\mathbf{x}^\top \Sigma^{-1} \mathbf{y}$	the inverse covariance	yes
Quadratic form of a graph	$\mathbf{x}^\top L \mathbf{x}$	the graph Laplacian	yes
Geometric algebra	$\mathbf{x}^\top G \mathbf{y}$	declared as a signature	yes

Two rows say “assumed I ”. In both, the assumption is invisible in the notation – which is exactly why it goes unexamined.

One row says “learned, and not symmetric”. Attention’s directionality is not a feature bolted onto the architecture. It is the non-symmetry of a bilinear form.

Three Ways to Get a Metric

Learn it	fit parameters and let G emerge implicitly, as attention and SGNS both do
Assume it	build a space in which $G \approx I$ holds by construction, as vector symbolic architectures do with random high-dimensional vectors
Declare it	fix the signature first and define every operation relative to it, as geometric algebra does

These are the three positions the rest of the course moves between. They are not competing theories of meaning. They are three answers to one question: where does the matrix in the middle come from?

The first is the one to watch, because a metric that is learned implicitly can also be discarded implicitly – which is exactly what happens next.

Summary, Parts I–IV

Point	Detail
The shape	vector, matrix, vector – one number out; we met it as like in the refresher
The dot product	$\mathbf{x}^\top I \mathbf{y}$: the I is a choice, not a fact
What I claims	equal coordinates, no interaction, symmetric roles – three assertions
Diagonal G	re-weights coordinates; this is what tf-idf, PPMI and truncated SVD do
Off-diagonal G	lets coordinates count for one another; correlation
Unit ball	the shape of $\{\mathbf{v} : \mathbf{v}^\top G \mathbf{v} = 1\}$ is the metric, drawn
Three middles	$\cos$ ran from 0.60 to 0.95 without any vector changing
Rankings move	under I the nearest neighbor of $\mathbf{a}$ was $\mathbf{b}$; under $\text{diag}(1, 9)$ it was $\mathbf{c}$
Bilinear form	linear in each argument; $B(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top G \mathbf{y}$ with $G_{ij} = B(\mathbf{e}_i, \mathbf{e}_j)$
Contract or not	$\mathbf{x}^\top \mathbf{y}$ shares a variable and sums it away; $\mathbf{x} \mathbf{y}^\top$ keeps both free
Outer product	binding: the Cartesian product $X \times Y$; always rank 1; the form factors
Rank	how relational a relation is; rank 1 is two unary predicates glued together
Roles and fillers	role = argument position of L , filler = value in it; $\mathbf{r}_k^\top T = \mathbf{f}_k^\top$
Unbinding	exact when the roles are orthonormal; crosstalk when they are not
Order survives	Alice likes Bob and Bob likes Alice bind to orthogonal T ; addition cannot tell them apart
Matrix product	relational composition: $\exists j. M(i, j) \wedge N(j, k)$
PMI	the log ratio of the observed matrix to the rank-1 independence model

Summary, Parts V–VIII

Point	Detail
Inner product	symmetric <i>and</i> positive definite; then $\ \cdot\ _G$, $\cos_G$ and Cauchy–Schwarz all hold
Non-symmetric G	not a metric, but a relation: like, and attention
Change of basis	$\mathbf{x} \mapsto A\mathbf{x}$ forces $G \mapsto A^{-\top}GA^{-1}$: the two trade off exactly
Absorption	$G = LL^\top$, $\mathbf{x}' = L^\top \mathbf{x}$, and the plain dot product becomes exact
Consequence	one Euclidean geometry up to change of basis – <i>provided you know G</i>
Degenerate G	nonzero vectors of length zero
Indefinite G	negative squared lengths; Minkowski; the light cone
Signature	$\text{Cl}(p, q, r)$ counts the $+1$, -1 and 0 directions; Sylvester says the counts are fixed
Three sources	learn G , assume G , or declare G

Exercises

- 1 Show that for any G , the symmetric part $\frac{1}{2}(G + G^\top)$ gives the same value of $\mathbf{x}^\top G \mathbf{x}$ as G itself, for every $\mathbf{x}$. What does that say about what a *length* can detect?
- 2 $G = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ is symmetric. Find a second vector, not a multiple of $(1, \sqrt{3} - 2)$, with $\mathbf{x}^\top G \mathbf{x} = 0$. How many such directions are there in the plane?
- 3 With $G = \text{diag}(1, 9)$ from the ranking example, we found $L^\top = \text{diag}(1, 3)$ and $\mathbf{a}' = (1, 3)$, $\mathbf{b}' = (1, 0.6)$, $\mathbf{c}' = (0.1, 3)$. Verify by ordinary cosine on the primed vectors that you recover 0.7593 and 0.9587.
- 4 Attention uses $G = W_Q^\top W_K$. Suppose you tied the two projections, $W_Q = W_K$. What would G become, what would that force about the attention scores, and what would be lost?
- 5 Session 8 truncated the SVD to rank k . Write that truncation as a choice of G on the original space. What is its signature?
- 6 **Harder.** A form with signature $\text{Cl}(p, q, r)$ and $r > 0$ has null directions. Give a reason a *semantic* representation might want a direction that squares to zero, and say what it would mean for two meanings to differ by a null vector.

Questions?

Next: a model that learned its metric
and then threw it away