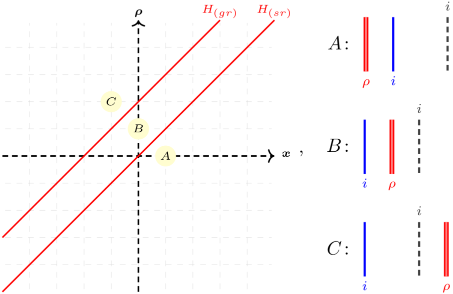
On weighted KLRW algebras

Or: Mind the distance

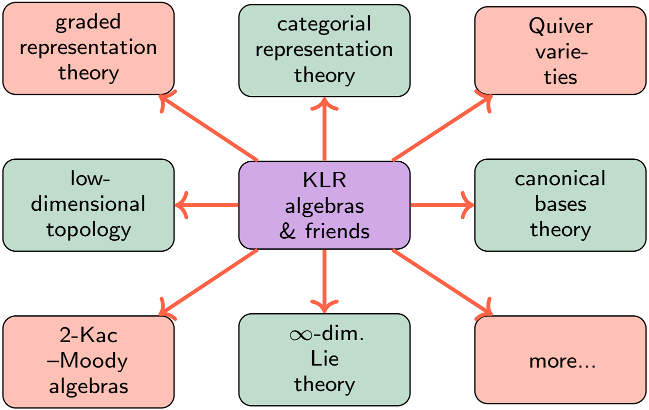
Daniel Tubbenhauer



Joint with Andrew Mathas

February 2022

Where are we?



- **Khovanov–Lauda–Rouquier ∼2008 + many others (including OIST)**
  KLR algebras are at the heart of categorical representation theory
- Similarly for quiver Schur algebras and diagrammatic Cherednik algebras
- Problem All of these are actually really complicated!

Where are

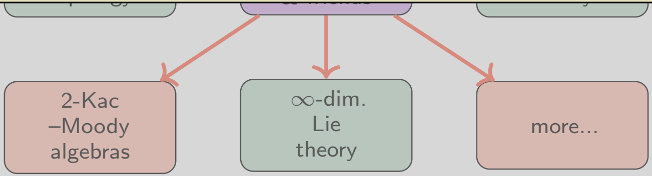
Observation

It often helps to find a "bigger" interpolating algebra, e.g.:

"Big" $\mathbb{C}[X]/(X-a)(X-b)$

$\xrightarrow{a=0,b=0}$ $\qquad$ $\xrightarrow{a=1,b=-1}$

"Small" $\mathbb{C}[X]/(X^2)$ $\qquad$ $\mathbb{C}[X]/(X^2-1)$



- **Khovanov–Lauda–Rouquier ∼2008 + many others (including OIST)**
  KLR algebras are at the heart of categorical representation theory
- Similarly for quiver Schur algebras and diagrammatic Cherednik algebras
- Problem All of these are actually really complicated!

Observation

It often helps to find a "bigger" interpolating algebra, e.g.:

"Big" $\mathbb{C}[X]/(X-a)(X-b)$

$a=0, b=0$: $\mathbb{C}[X]/(X-a)(X-b) \to \mathbb{C}[X]/(X^2)$

$a=1, b=-1$: $\mathbb{C}[X]/(X-a)(X-b) \to \mathbb{C}[X]/(X^2-1)$

"Small" $\mathbb{C}[X]/(X^2)$ $\mathbb{C}[X]/(X^2-1)$

Today

How to play the interpolation game using planar geometry?

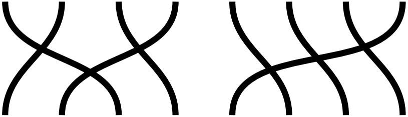
As an upshot we get an algebra interpolating between various algebras appearing in categorical representation theory

The takeaway keyword: Distance!

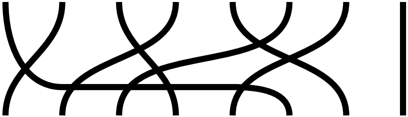
- Kh … IST)
  KL…
- Sim … ras
- Problem All of these are actually really complicated!

String diagrams – the baby case

Connect eight points at the bottom with eight points at the top:

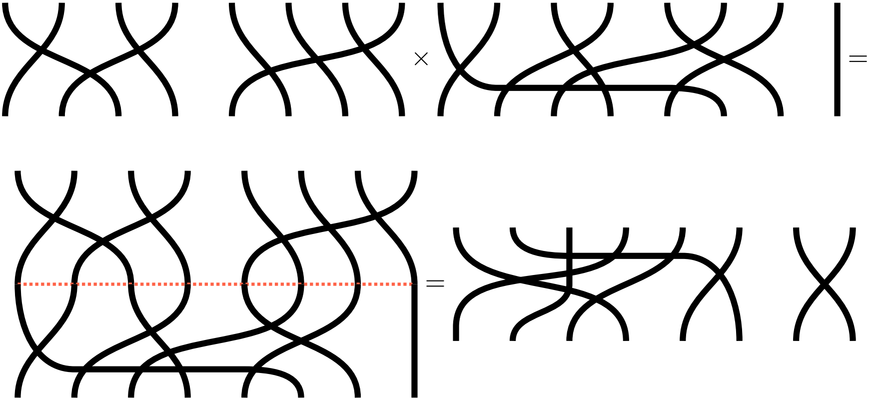
$(1243)(5876) \leftrightsquigarrow$ 

or

$(12436)(57)(8) \leftrightsquigarrow$ 

We just invented the symmetric group S_8

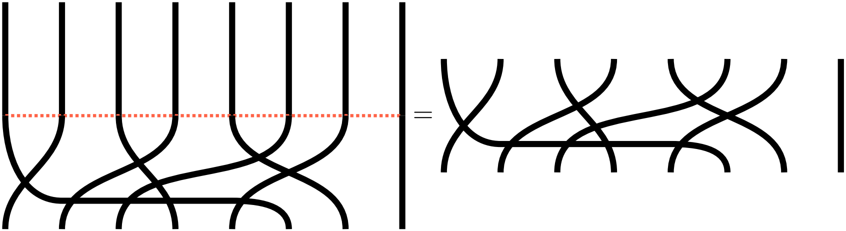
String diagrams – the baby case



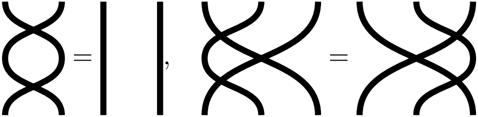
My multiplication rule for gh is “stack g on top of h”

String diagrams – the baby case

- We clearly have $g(hf) = (gh)f$
- There is a do nothing operation $1g = g = g1$



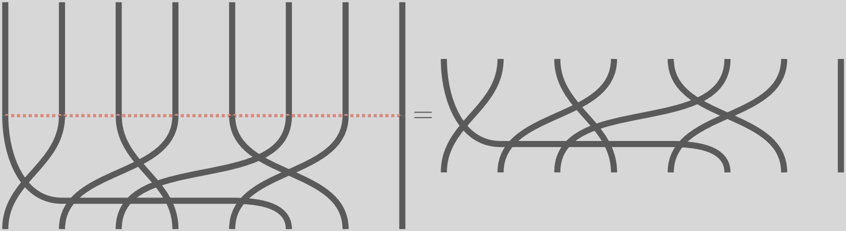
- Generators–relations (the Reidemeister moves)

gens : , rels : 

The bait

In diagram algebras relations, properties, etc. become visually clear

- We clearly have
- There is a do nothing operation $1g = g = g1$



- Generators–relations (the Reidemeister moves)

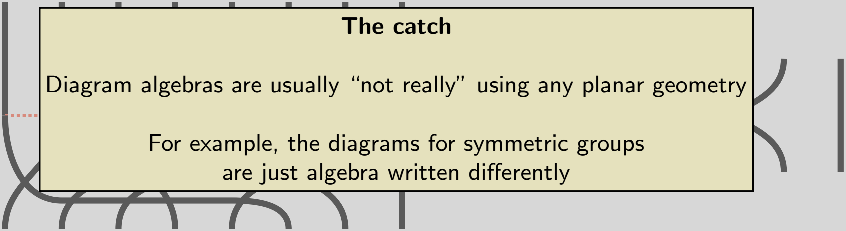
gens : , rels : 

String diagrams –

The bait

In diagram algebras relations, properties, etc. become visually clear

- We clearly have
- There is a do nothing operation $1g = g = g1$



The catch

Diagram algebras are usually "not really" using any planar geometry

For example, the diagrams for symmetric groups are just algebra written differently

- Generators–relations (the Reidemeister moves)

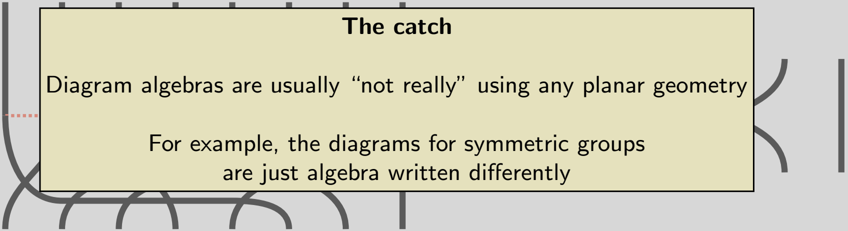
gens : , rels : 

String diagrams –

The bait

In diagram algebras relations, properties, etc. become visually clear

- We clearly hav
- There is a do nothing operation $1g = g = g1$



The catch

Diagram algebras are usually “not really” using any planar geometry

For example, the diagrams for symmetric groups are just algebra written differently

- Gene

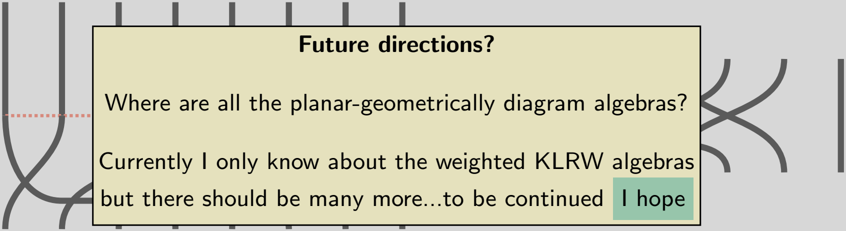
Idea (Webster ∼2012)

Define a diagram algebra that uses the distance in $\mathbb{R}^2$

The result is called weighted KLRW algebra

These are “planar-geometrically symmetric group diagram algebras”

- We clearly have $g(hf) = (gh)f$
- There is a do nothing operation $1g = g = g1$

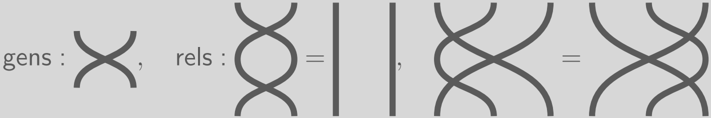


Future directions?

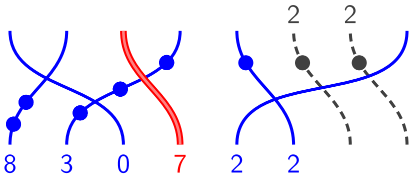
Where are all the planar-geometrically diagram algebras?

Currently I only know about the weighted KLRW algebras but there should be many more...to be continued I hope

- Generators–relations (the Reidemeister moves)



Weighted string diagrams

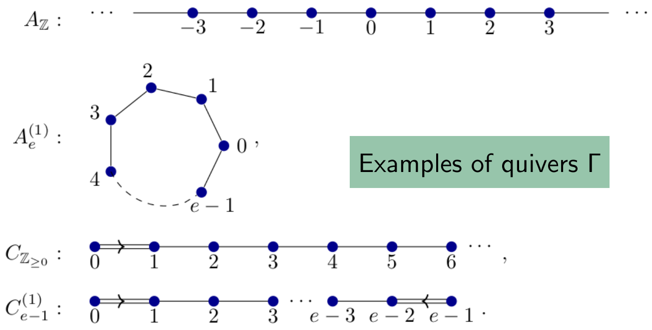


- Strings come in three types, solid, ghost and red

$$\text{solid} : \underset{i}{|}\ , \quad \text{ghost} : \overset{i}{\vdots}\ , \quad \text{red} : \underset{i}{\|}\ ,$$

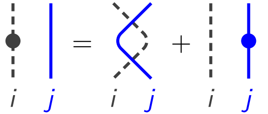
- Strings are labeled, and solid and ghost strings can carry dots
- Red strings anchor the diagram (red strings $\leftrightsquigarrow$ level)
- Otherwise no difference to symmetric group diagrams

Weighted string diagrams



Examples of quivers Γ

- The strings are labeled by $i \in I$ from a fixed quiver $\Gamma = (I, E)$
- The relations (that I am not going to show you ;-)) depend on $e \in E$, e.g.:

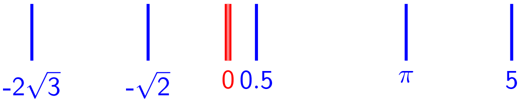


if $i \to j$

Weighted string diagrams

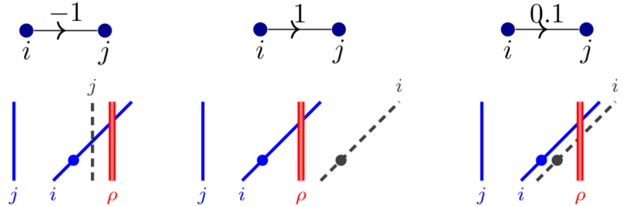
This is the first time I use the number π in a talk ;-)

$X = (\text{-}2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \leftrightsquigarrow$



Weighted quiver

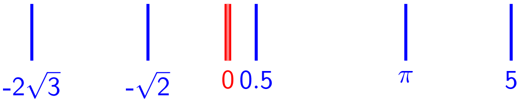
diagram



- Choose endpoints $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\rho \in \mathbb{R}^\ell$ for the solid and red strings
- Choose a weighting $\boldsymbol{\sigma} \colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The weighted KLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

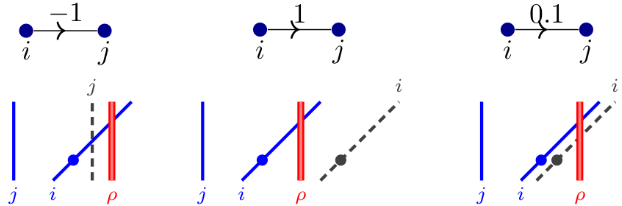
Weighted string diagrams

$X = (\text{-}2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \leftrightsquigarrow$



Weighted quiver

diagram



Weighting = ghost shifts

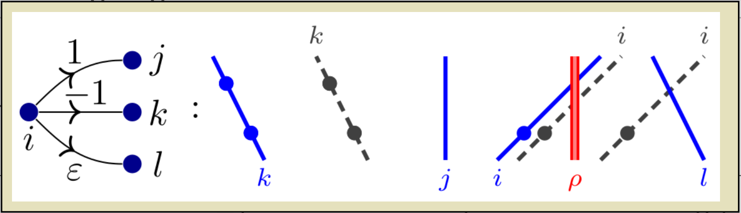
For ϵ: $i \to j, \sigma_\epsilon > 0$, all solid i-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it
For ϵ: $i \to j, \sigma_\epsilon < 0$, all solid j-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it

This "asymmetric" definition, always shifting rightwards makes life a bit more convenient

- The weigh ... ces of endpoints! ...

Weighted string diagrams

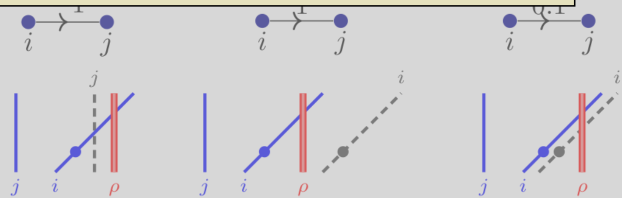
$X = ($



5

Weighted quiver

diagram

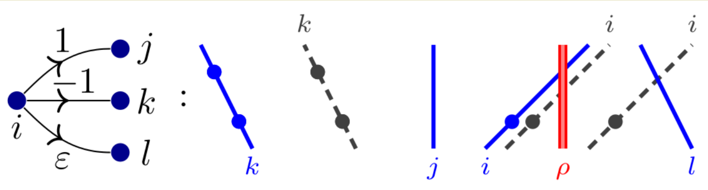


Weighting = ghost shifts

For $\epsilon\colon i \to j, \sigma_\epsilon > 0$, all solid i-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it
For $\epsilon\colon i \to j, \sigma_\epsilon < 0$, all solid j-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it

- The weighted KLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

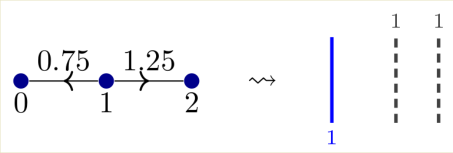
Weighted string diagrams

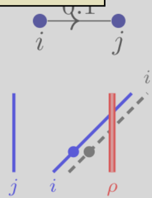


$X = ($ … 5

Weighted quiver

diagram





Weighting = ghost shifts

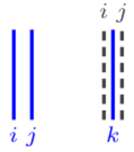
For $\epsilon\colon i \to j, \sigma_\epsilon > 0$, all solid i-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it
For $\epsilon\colon i \to j, \sigma_\epsilon < 0$, all solid j-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it

- The weighted KLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

Weighted string diagrams

$X = (-2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \leftrightsquigarrow$ 

The following i and j-strings are not close:



Slogan Ghosts prevent the diagrams from being scale-able as for "usual diagram algebras"

- Choose endpoints $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, $\rho \in \mathbb{R}^\ell$ for the solid and red strings
- Choose a weighting $\boldsymbol{\sigma}\colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The weighted KLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

Weighted string diagrams

$X = (\text{-}2\sqrt{}$

"Big" $\mathbb{C}[X]/(X-a)(X-b)$

$a=0, b=0$ ↙ ↘ $a=1, b=-1$

"Small" $\mathbb{C}[X]/(X^2)$ $\mathbb{C}[X]/(X^2-1)$

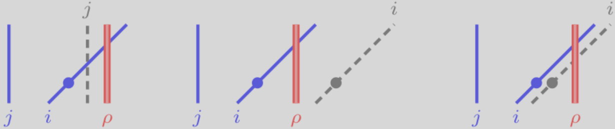
a, b ⇝ Positions and ghost shifts

5

Weighted

.1 j

diagram



- Choose endpoints $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\rho \in \mathbb{R}^\ell$ for the solid and red strings
- Choose a weighting $\boldsymbol{\sigma}\colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The weighted KLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

Weighted string diagrams

$X = (\text{-}2\sqrt{}$ …

“Big” $\mathbb{C}[X]/(X-a)(X-b)$

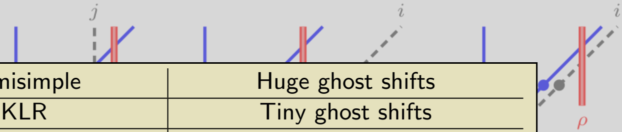
$a=0, b=0$ → “Small” $\mathbb{C}[X]/(X^2)$

$a=1, b=-1$ → $\mathbb{C}[X]/(X^2-1)$

$a, b \leftrightsquigarrow$ Positions and ghost shifts

Weighted

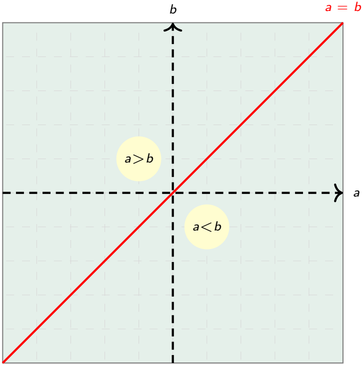
diagram…



Semisimple	Huge ghost shifts
KLR	Tiny ghost shifts
Quiver Schur	Some specific “cluster” spacing
Diagrammatic Cherednik	Ghost shifts 1
Unnamed algebras	The rest

- Choose endpoints … strings
- Choose a weighting $\boldsymbol{\sigma}\colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The weighted KLRW algebra crucially depends on these choices of endpoints! This is very different from “usual diagram algebras”

Hyperplanes

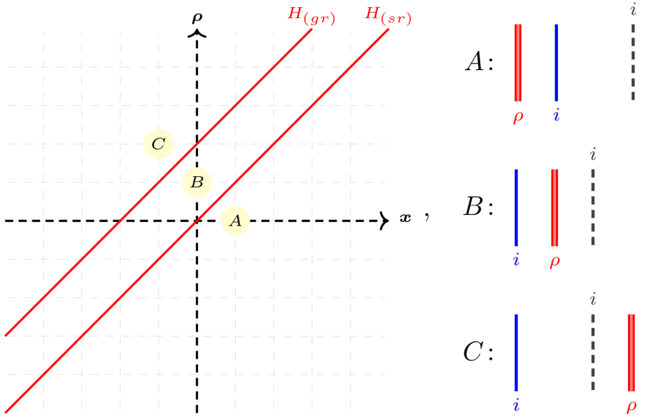
over $\mathbb{R}$: 

- $\mathbb{C}[X]/(X-a)(X-b)$ comes in two isomorphism classes:

one double root $a=b$ & two different roots $a \neq b$

- What is the analog picture for weighted KLRW algebras?

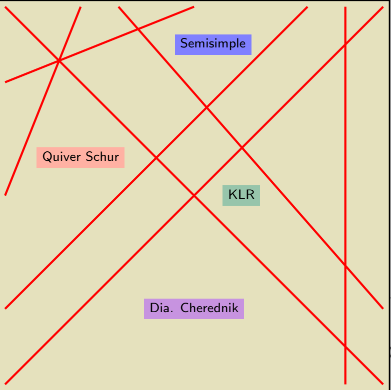
Hyperplanes



- There is a hyperplane arrangment (HA) associated to the weighted KLRW
- The hyperplanes are defined by "colliding strings" (a form of distance)

Hy

- Alcoves of the HA ⇒ Morita equivalence classes of weighted KLRW algebras
  - There is a theory of translation functors
- ≈picture 1 There is an alcove for KLR, an alcove for the semisimple case etc.
  - ≈picture 2 Translation functors interpolate between these algebras

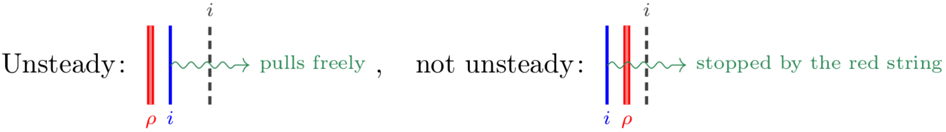


This is an ≈picture

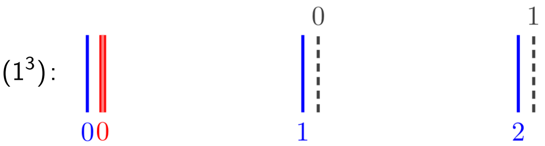
- There is a hyperpl … o the weighted KLRW
- The hyperplanes ar … form of distance)

Distance is it!

- Cyclotomic quotients ⇔ bounded regions:



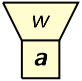
- Cellular bases ⇔ minimal regions (I will elaborate momentarily):



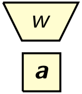
- More properties I won't explain today due to time restrictions...

Distance is it!

- Weighted KLRW algebras have standard bases, with the picture:



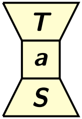
with



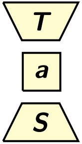
is a permutation diagram,

is an idempotent with dots.

- Weighted KLRW algebras have “cellular” bases, with the picture:



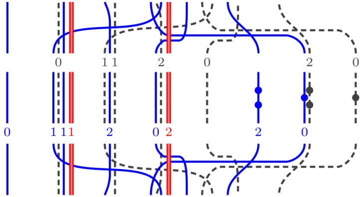
with



is a permutation diagram,

is an idempotent with dots,

is a permutation diagram.



Distance is

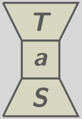
- Weight

▶ Standard bases work regardless of the quiver
but have no other property despite being a basis

▶ Cellular bases depend on the quiver
and give a classification of simple modules

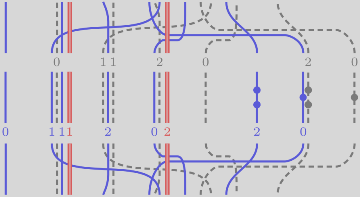
(▶ Strictly speaking I should write "affine or sandwich" cellular
but let us ignore that)

- Weighted KLRW algebras have "cellular" bases, with the picture:



with

is a permutation diagram,
is an idempotent with dots,
is a permutation diagram.



Distance i

▶ Weigh

▶ Standard bases work regardless of the quiver
but have no other property despite being a basis

▶ Cellular bases depend on the quiver
and give a classification of simple modules

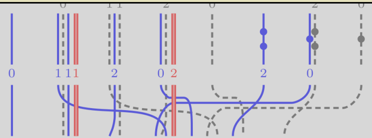
(▶ Strictly speaking I should write "affine or sandwich" cellular
but let us ignore that)

▶ The overall strategy to construct cellular bases
is the same for all type (but the details differ)

and for the infinite dimensional and the cyclotomic case the construction is also the same

▶ We know that the cellular bases work in types A, B, C, affine $A, C, A^{(2)}, D^{(2)}$
but they should work even more general

▶ The combinatorics is inspired by, but different from, constructions of
Bowman ∼2017, Ariki–Park ∼2012/2013, Ariki–Park–Speyer ∼2017



Distance i

▶ Weigh

▶ Standard bases work regardless of the quiver
but have no other property despite being a basis

▶ Cellular bases depend on the quiver
and give a classification of simple modules

(▶ Strictly speaking I should write "affine or sandwich" cellular
but let us ignore that)

▶ The overall strategy to construct cellular bases
is the same for all type (but the details differ)
and for the infinite dimensional and the cyclotomic case the construction is also the same

▶ We know that the cellular bases work in types A, B, C, affine $A, C, A^{(2)}, D^{(2)}$
but they should work even more general

▶ The combinatorics is inspired by, but different from, constructions of
Bowman ∼2017, Ariki–Park ∼2012/2013, Ariki–Park–Speyer ∼2017

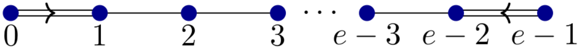
I will now indicate how the construction works in type $C_{e-1}^{(1)}$

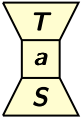
Why this type?

Because the code I am going to use works best for this type ;-)

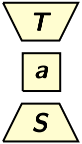
Minimal diagrams in type $C^{(1)}_{e-1}$

$C^{(1)}_{e-1}$:





with

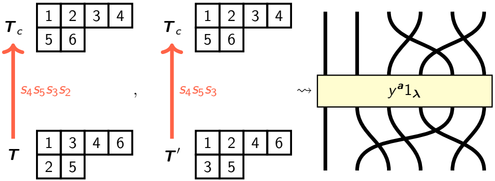


is a permutation diagram,

is an idempotent with dots,

is a permutation diagram.

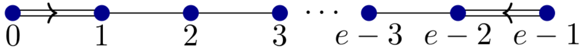
- The definition of the permutation follows the usual strategy in this context:



- Let me focus on the middle $y^{a}1_{\lambda}$

Minimal diagrams in type $C^{(1)}_{e-1}$

$C^{(1)}_{e-1}$:



$C^{(1)}_{3}$: $(12, 6^3, 5) \leftrightsquigarrow$

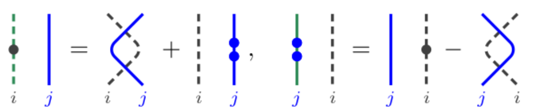
0	1	2	3	2	1	0	1	2	3	2	1
1	0	1	2	3	2						
2	1	0	1	2	3						
3	2	1	0	1	2						
2	3	2	1	0							

- Assume the tableaux combinatorics is given
- Place strings inductively as far to the right as possible (this is the order!)
- 1_λ is minimal with respect to placing the strings to the right
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^{a}1_\lambda$
- Done!

Minimal diagrams in type $C^{(1)}_{e-1}$

0	1	2	3	2	1	0	1	2	3	2	1
1	0	1	2	3	2						

Lets ignore the dots for today – I bothered you with too much combinatorics anyway ;-)
But they come directly from the Reidemeister II relations, e.g.



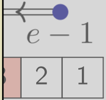
for either of $\begin{cases} i=0, j=1, \\ i=e, j=e-1, \end{cases}$

In other words: Stare at Reidemeister II !

- Place strings inductively as far to the right as possible (this is the order!)
- 1_λ is minimal with respect to placing the strings to the right
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^{\mathbf{a}}1_\lambda$
- Done!

Minimal diagrams in type $C^{(1)}_{e-1}$

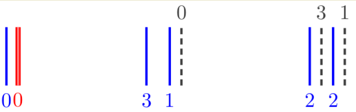
$C^{(1)}_{e-1}$:



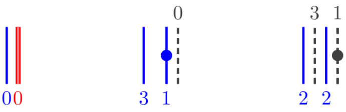
$C^{(1)}_3$: (12

Example for the middles $y^a 1_\lambda$

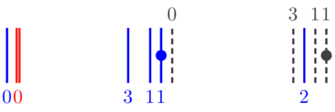
$y_{(1^5)}\mathbf{1}_{(1^5)} =$



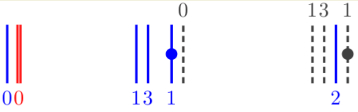
$y_{(5)}\mathbf{1}_{(5)} =$



$y_{(2,1^3)}\mathbf{1}_{(2,1^3)} =$



$y_{(4,1)}\mathbf{1}_{(4,1)} =$



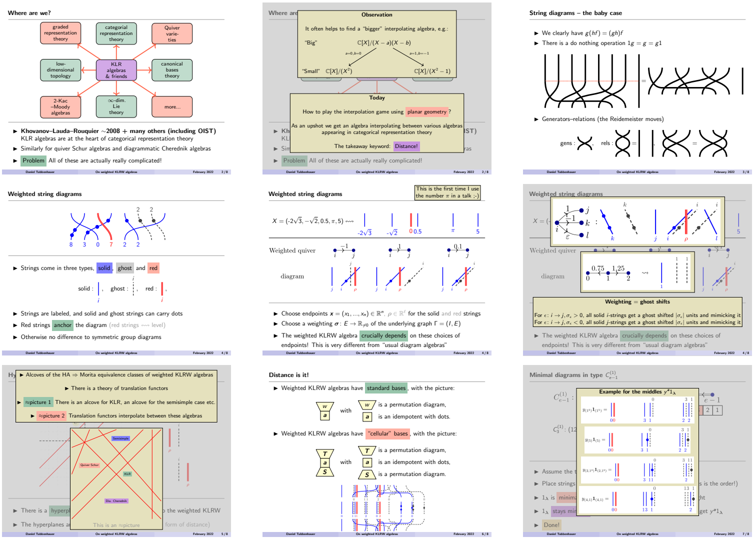
- Assume the t
- Place strings … s is the order!)
- 1_λ is minima … ght
- 1_λ stays min … get $y^a 1_\lambda$
- Done!

Minimal diagrams in type $C^{(1)}_{e-1}$

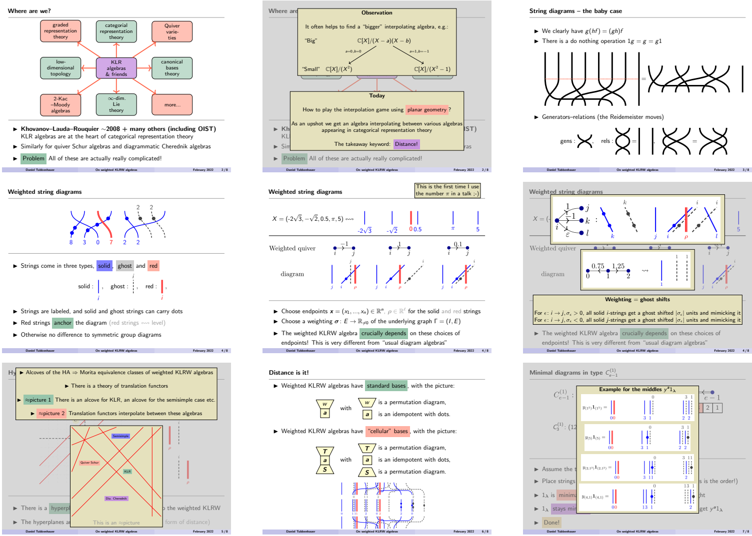
Wrap up

- Weighted KLRW algebras generalize KLR algebras and friends
- They have a build in distance
- Most properties can be described using distance
- Most properties are type-independent
- Some properties should be (in some form) type-independent

- Assum
- Place … order!)
- 1_λ is minimal with respect to placing the strings to the right
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^{a}1_\lambda$
- Done!



There is still much to do...



Thanks for your attention!