ChatJesseT

Type your question below.

Ask

Question: *When do you think Phiala solve QCD with her amazing work on Normalizing Flows?*

Ah, Phiala! She's a brilliant physicist, and her work on Normalizing Flows is truly remarkable. While Normalizing Flows have shown great promise in various applications, such as LHC event generation and anomaly detection, solving Quantum Chromodynamics (QCD) is a whole different beast. QCD, the theory that describes the strong nuclear force, is notoriously complex. It governs the interactions of quarks and gluons, and understanding its behavior at low energies, where confinement occurs, is still an open challenge. While Normalizing Flows offer a powerful tool for modeling probability distributions, they are not a magic wand that can instantly solve the intricacies of QCD. However, Phiala's work on Normalizing Flows can certainly contribute to our understanding of QCD. By providing a tractable likelihood evaluation and the ability to train on weighted events, Normalizing Flows can help us generate more accurate simulations and improve our theoretical models. They offer a valuable tool for exploring the phase space of QCD processes and quantifying uncertainties. So, while Phiala's amazing work on Normalizing Flows won't solve QCD on its own, it is a significant step forward in our quest to unravel the mysteries of the strong nuclear force. It's like a powerful microscope that allows us to zoom in on the intricate details of QCD, bringing us closer to a deeper understanding of the fundamental particles and forces that shape our universe.

Ask Jesse Thaler AI bot at https://chatjesset.com/

Enhancing Expressivity in Machine Learning: Application of Normalizing Flows in lattice QCD Simulations

Denis Boyda
IAIFI, MIT

On behalf of MIT ML collaboration: P. Shanahan(MIT), D. Hackett(MIT), F. Romero-López(MIT), S. Racanière(DM), D. Rezende(DM), A. Razavi(DM), A. Botev(DM), G. Kanwar(UB), K. Cranmer(NYU), M. Albergo(NYU)

Lattice 2023
August 3rd, 2023,
@ Fermilab

Other talks from our group

Ryan Abbott
(Monday 4:20)

Multiscale Normalizing Flows for Gauge Theories

Julian Urban
(Thursday 2:10)
Constructing approximate semi-analytic and machine-learned trivializing maps for lattice gauge theory

Daniel Hackett
(Thursday 3:10)
Practical applications of machine-learned flows on gauge fields

This talk is not about results but about a principle and a way of approaching NFs for LGT

Spectral Flow: v1

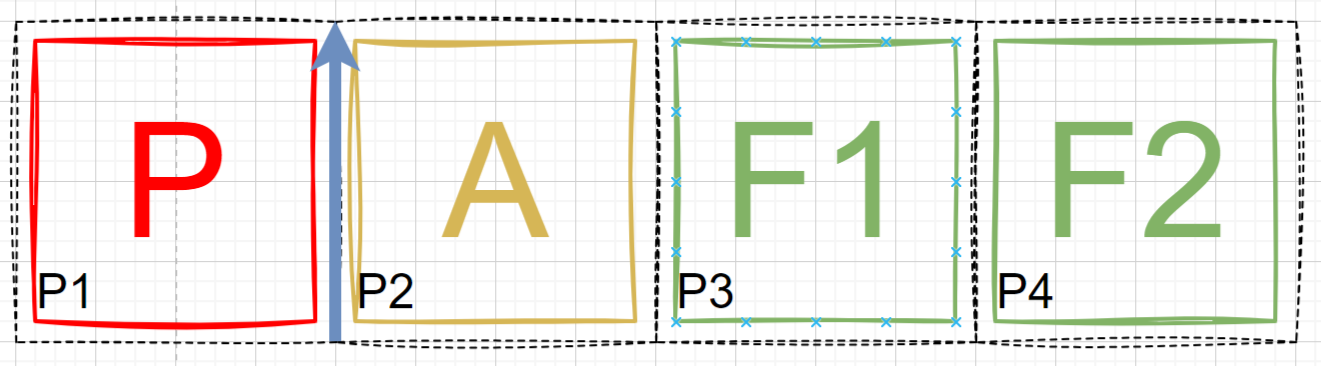
NF: $\int dV\ e^{S} = \int dU\ e^{S+logJ}$

<u>Target action</u> $S = ReTr\ P_1 + ReTr\ P_2 + ReTr\ P_3\ + ReTr\ P_4$

Spectral Flow: v1

NF: $\int dV\, e^{S} = \int dU\, e^{S+logJ}$

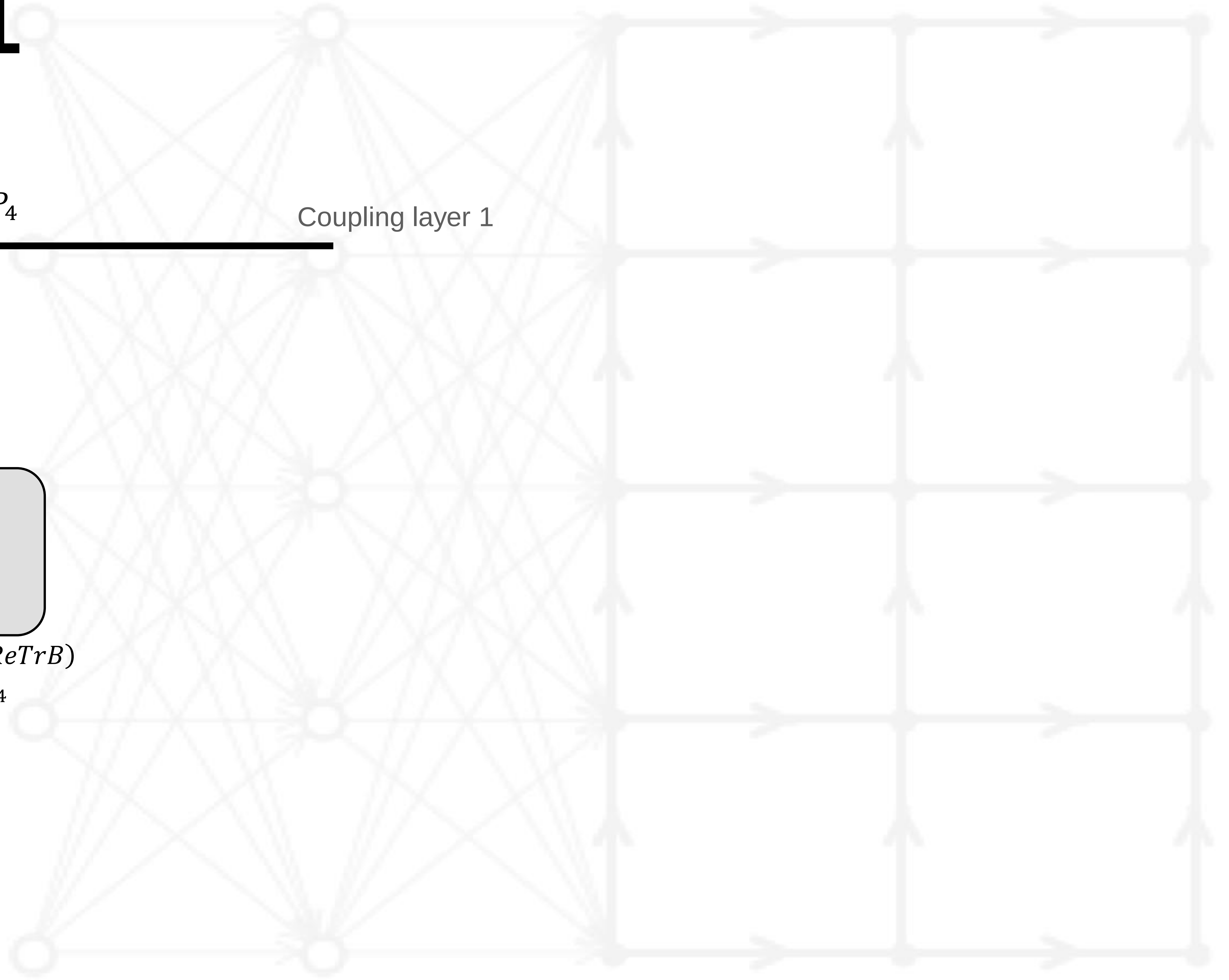
Target action $S = ReTr\ P_1 + ReTr\ P_2 + ReTr\ P_3\ + ReTr\ P_4$



1) Name (mask) plaquettes
2) Transform blue (active) link $U \rightarrow U'$

Maximal Torus Flow

3) Expressive transformation should have $logJ \sim -ReTrA\ (-ReTrB)$
4) (Ideal) Action after transformation $S \rightarrow S' = ReTrP_3 + ReTrP_4$

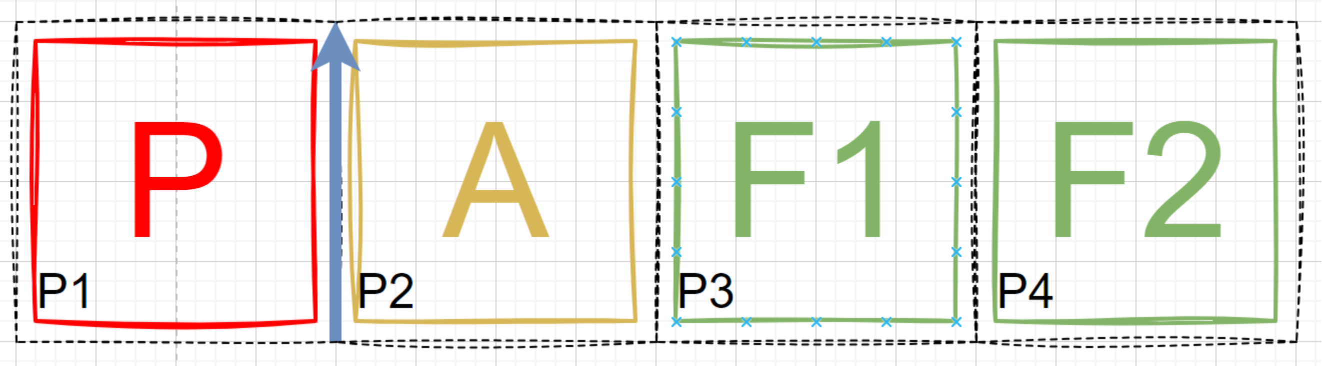


Spectral Flow: v1

NF: $\int dV\, e^{S} = \int dU\, e^{S + logJ}$

Target action $S = ReTr\,P_1 + ReTr\,P_2 + ReTr\,P_3 + ReTr\,P_4$

Coupling layer 1

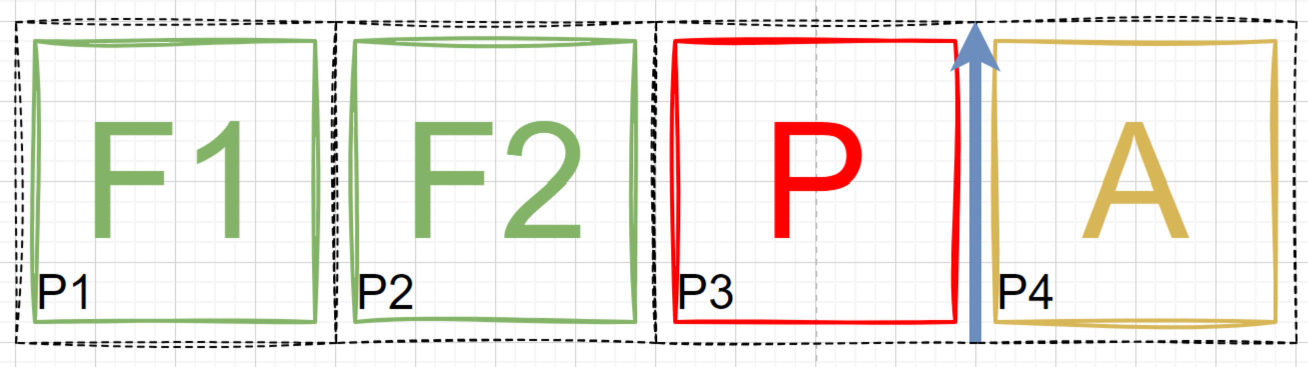


1) Name (mask) plaquettes
2) Transform blue (active) link $U \rightarrow U'$

Maximal Torus Flow

3) Expressive transformation should have $logJ \sim -ReTrA\ (-ReTrB)$
4) (Ideal) Action after transformation $S \rightarrow S' = ReTrP_3 + ReTrP_4$

Coupling layer 2



1) ReName plaquettes / (alternate mask)
2) Transform blue (active) link $U \rightarrow U'$
3) Expressive transformation should have v
4) (Ideal) Action after transformation $S \rightarrow S' = 0$

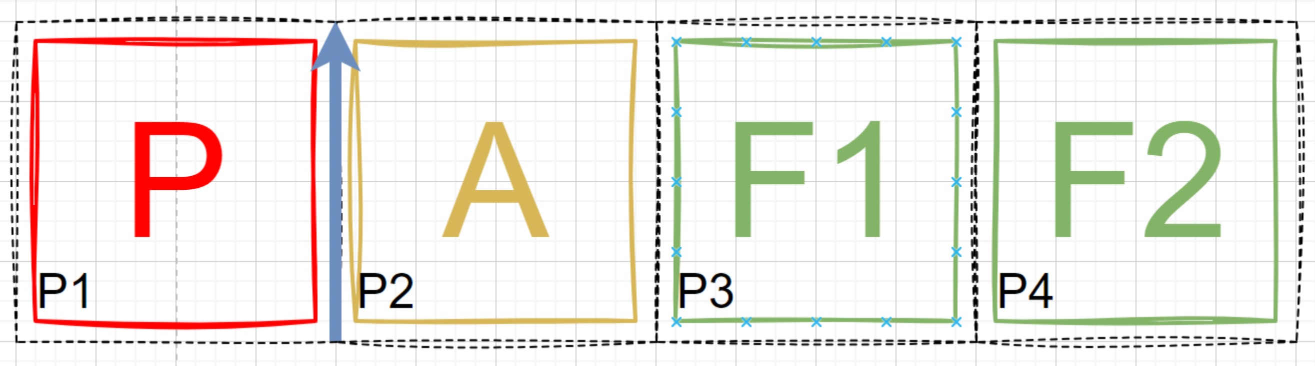
Coupling layer …

Spectral Flow: v1

NF: $\int dV\, e^{S} = \int dU\, e^{S+\log J}$

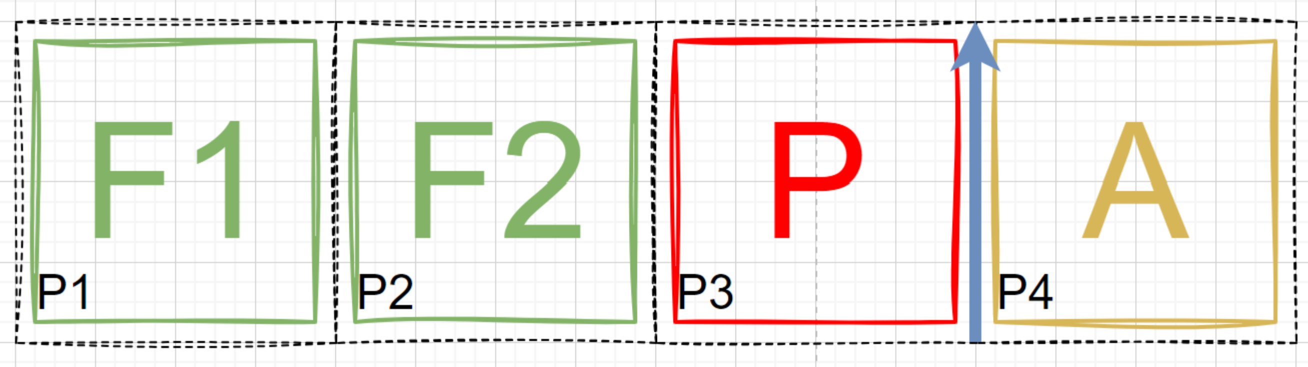
Target action $S = ReTr\, P_1 + ReTr\, P_2 + ReTr\, P_3 + ReTr\, P_4$

Coupling layer 1



1) Name (mask) plaquettes
2) Transform blue (active) link $U \to U'$
   a) Transform active plaquette $A \to A' = f(A|\, TrF1,\ TrF2)$
      - Diagonalize $A = V^{\dagger} L\, V$
      - Transform diagonals $L \to L' = g(L|TrF1, TrF2)$
      - Undiagonalize $A = V^{\dagger} L'\, V$
   b) Push update to the link $U \to U' = A' A^{\dagger} U$
3) Expressive transformation should have $\log J \sim -ReTrA\ (-ReTrB)$
4) (Ideal) Action after transformation $S \to S' = ReTrP_3 + ReTrP_4$

Coupling layer 2



1) ReName plaquettes / (alternate mask)
2) Transform blue (active) link $U \to U'$
3) Expressive transformation should have v
4) (Ideal) Action after transformation $S \to S' = 0$

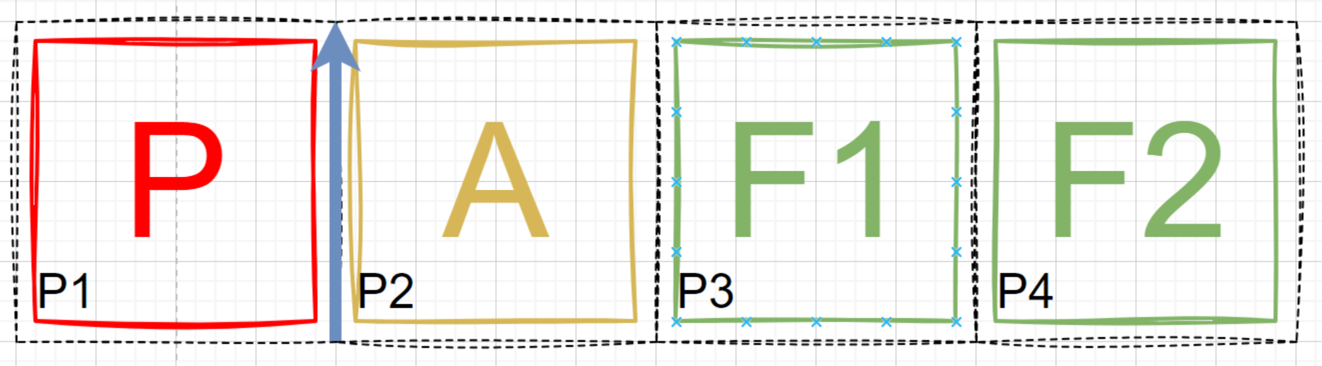
Coupling layer …

Spectral Flow: v1

NF: $\int dV\ e^{S} = \int dU\ e^{S+logJ}$

<u>Target action</u> $S = ReTr\ P_1 + ReTr\ P_2 + ReTr\ P_3 + ReTr\ P_4$

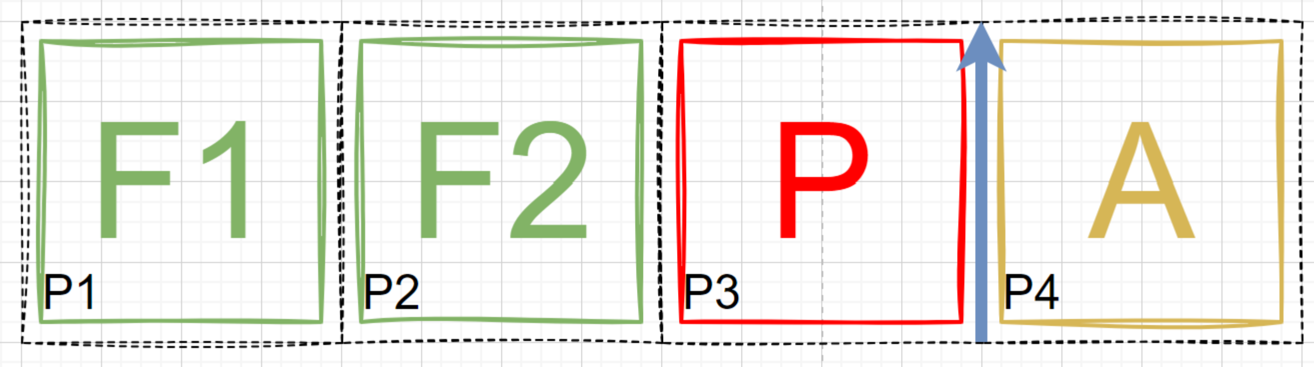
Coupling layer 1



1) Name (mask) plaquettes
2) Transform blue (active) link $U \to U'$
    a) Transform active plaquette $A \to A' = f(A|\ TrF1,\ TrF2)$
    - Diagonalize $A = V^{\dagger} L\ V$
    - Transform diagonals $L \to L' = g(L|TrF1, TrF2)$
    - Undiagonalize $A = V^{\dagger} L'\ V$

    b) Push update to the link $U \to U' = A' A^{\dagger} U$
3) Expressive transformation should have $logJ \sim -ReTrA\ (-ReTrB)$
4) (Ideal) Action after transformation $S \to S' = ReTrP_3 + ReTrP_4$

Coupling layer 2



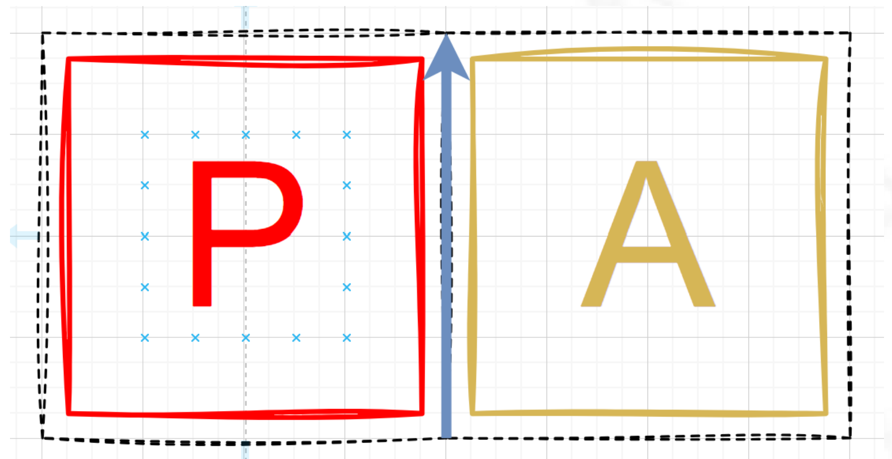
1) ReName plaquettes / (alternate mask)
2) Transform blue (active) link $U \to U'$
3) Expressive transformation should have v
4) (Ideal) Action after transformation $S \to S' = 0$

Coupling layer …

<u>Key observations:</u>

- Jacobian of expressive transformation should compensate active/passive plaquettes in the action. However, it never happens in practice.
- Transformation is conditioned on invariant features fully ignoring equivariant information
- Degrees of freedom argument

A deeper look



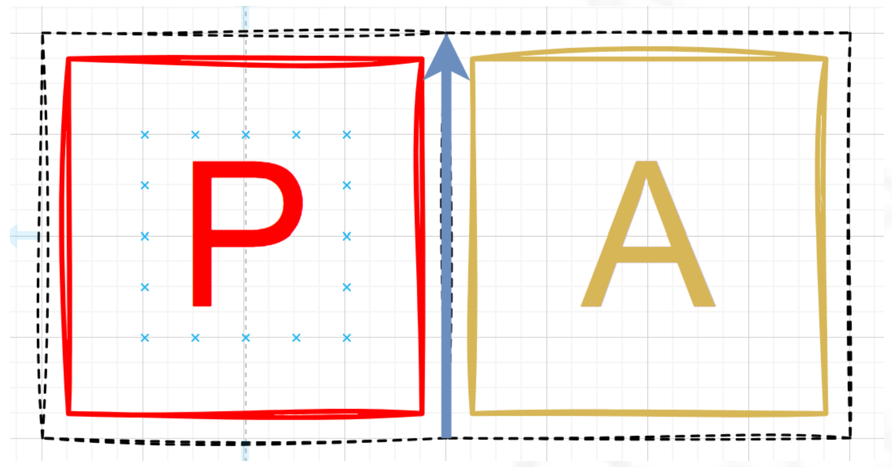
Target action

$$S = Re\,Tr\,(A + P)$$

$$U' = f(U)$$

Aim: $\int dU' \exp[S(U) + log J] = \int dU$

A deeper look



Target action

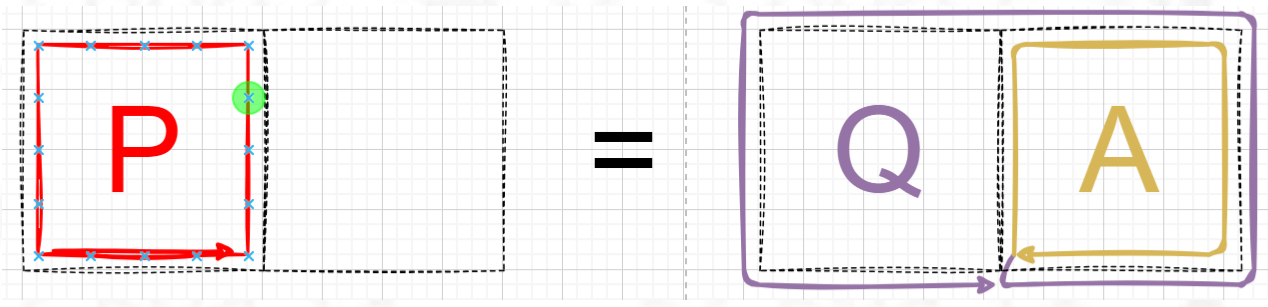
$$S = Re\,Tr\,(A + P)$$

$U' = f(U)$

Aim: $\int dU' \exp[S(U) + logJ] \;= \int dU$

We need take into dependence of passive loop on active. Both are transformed though an active link, let's use it.
$A = S_A U^{\dagger}$, $P = U\, S_P$ => $P = A^{\dagger} S_A S_P$

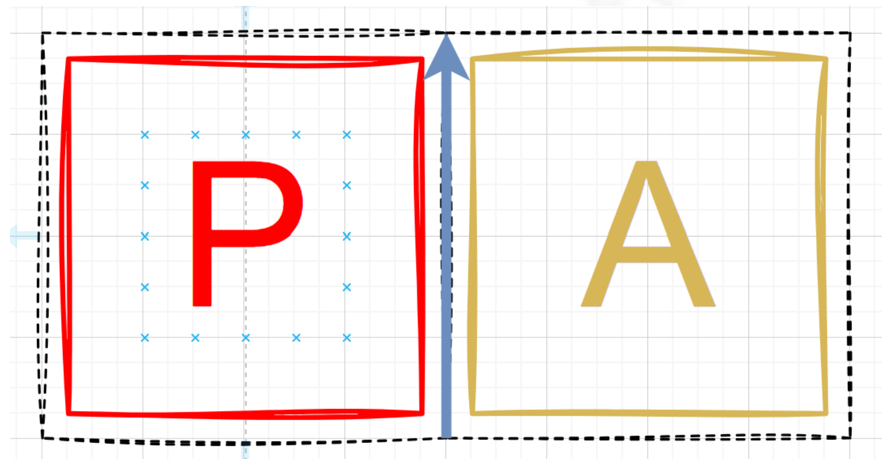
$$\Rightarrow\; -logJ = S(U) = ReTr(A + P) = ReTr(A\,(1 + Q))$$



A deeper look

Target action

$S = Re\,Tr\,(A+P)$

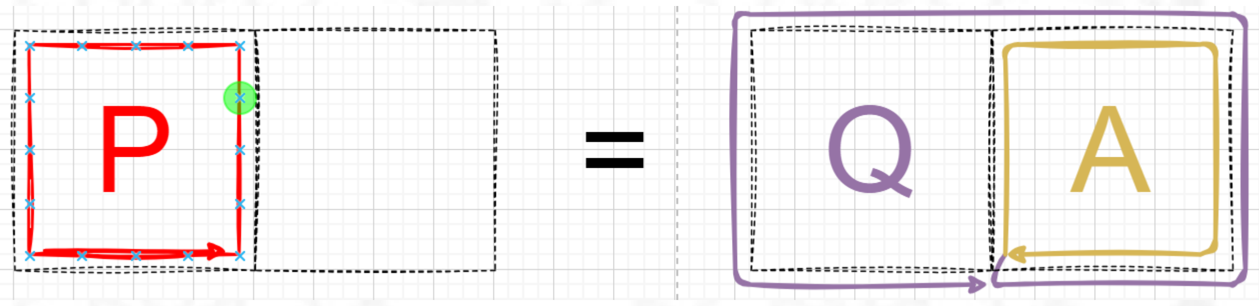


$U' = f(U)$

Aim: $\int dU' \exp[S(U) + logJ] \; = \int dU$

We need take into dependence of passive loop on active. Both are transformed though an active link, let's use it.
$A = S_A U^\dagger,\ P = U\,S_P \Rightarrow P = A^\dagger S_A S_P$

$\Rightarrow\ -logJ = S(U) = ReTr(A+P) = ReTr(A\,(1+Q))$



Recall for spectral flow

a) Transform active plaquette $A \to A' = f(A|\,TrF1,\ TrF2)$
- Diagonalize $\boxed{A = V^\dagger L\,V}$
- Transform diagonals $L \to L' = g(L|TrF1, TrF2)$
- Undiagonalize $A = V^\dagger L'\,V$

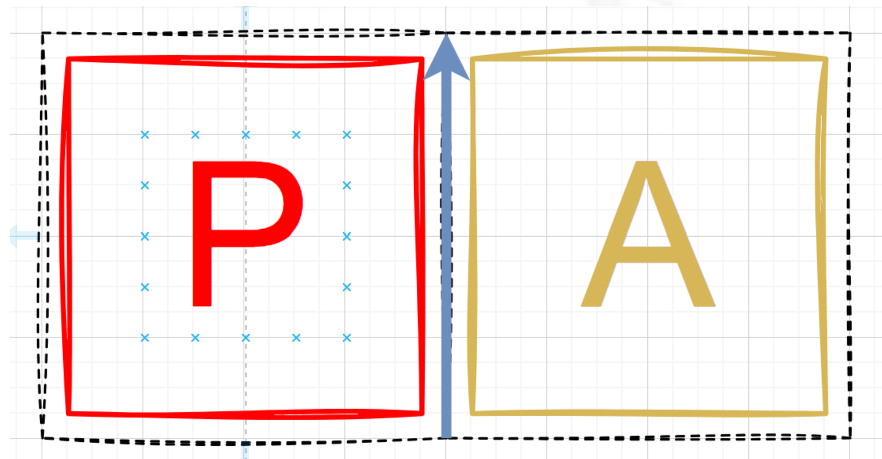
b) Push update to the link $U \to U' = A'A^\dagger U$

And don't forget Haar measure

L is a diagonal matrix of eigen values

$$\Rightarrow Haar(L')\frac{dL'}{dL} = Haar(L)\,\exp\left[-Re\,Tr\,\left(L\,V(1+Q)V^\dagger\right)\right]$$

A deeper look



Target action

$S = Re\,Tr\,(A + P)$

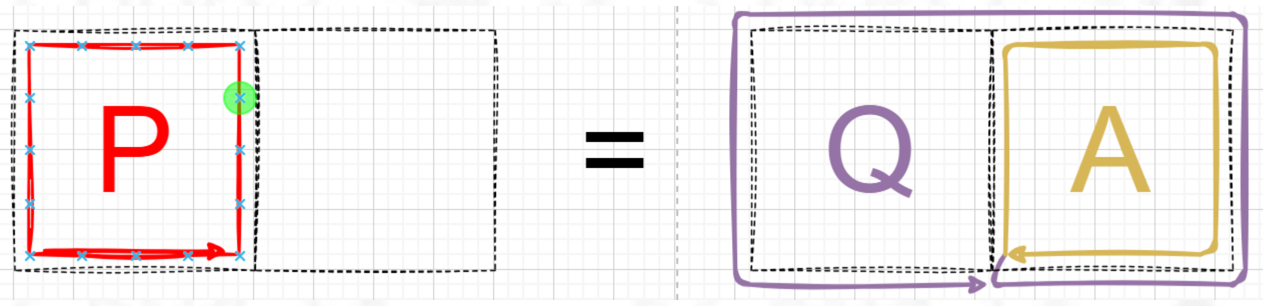
$U' = f(U)$

Aim: $\int dU' \exp[S(U) + logJ] \; = \int dU$

What would happen if we used different active loops (W2x1, W2x2,…)? It would only change loop(s) Q!

We need take into dependence of passive loop on active. Both are transformed though an active link, let's use it.
$A = S_A U^\dagger$, $P = U\,S_P$ => $P = A^\dagger S_A S_P$

$\Rightarrow\; -logJ = S(U) = ReTr(A + P) = ReTr(A\,(1 + Q))$



Recall for spectral flow

a) Transform active plaquette $A \;\rightarrow A' = f(A|\,TrF1,\; TrF2)$
- Diagonalize $\boxed{A = V^\dagger\, L\, V}$
- Transform diagonals $L \rightarrow L' = g(L|TrF1, TrF2)$
- Undiagonalize $A = V^\dagger\, L'\, V$

b) Push update to the link $U \;\rightarrow U' = A' A^\dagger U$

And don't forget Haar measure

L is a diagonal matrix of eigen values

$$\Rightarrow Haar(L')\frac{dL'}{dL} = Haar(L)\,\exp\left[-Re\,Tr\,\left(L\,V(1 + Q)V^\dagger\,\right)\right]$$

This transformation trivialized target action!

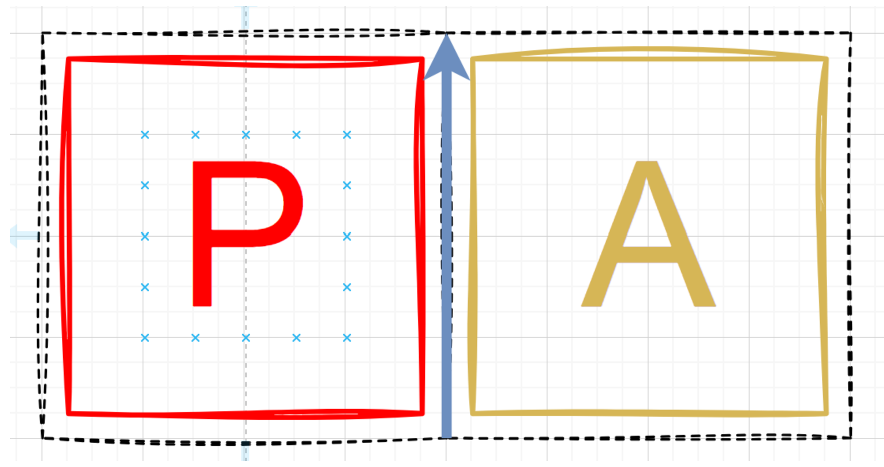
$$\boxed{\Rightarrow L' = \int \exp\left[-Re\,Tr\,\left(L\,(V(1 + Q)V^\dagger\,\right)\right] Haar(L)/Haar(L')dL}$$

Denis: I don't want to solve it, I'll use AI

Julian: I love old-style, I will solve it!

A deeper look

Target action $S = Re\,Tr\,(A+P)$



We can add $diag$ because L is diagonal!

$$L' = \int \exp[-Re\,Tr\,(L\ \ V(1+Q)V^{\dagger}\ )]\,Haar(L)/Haar(L')dL$$

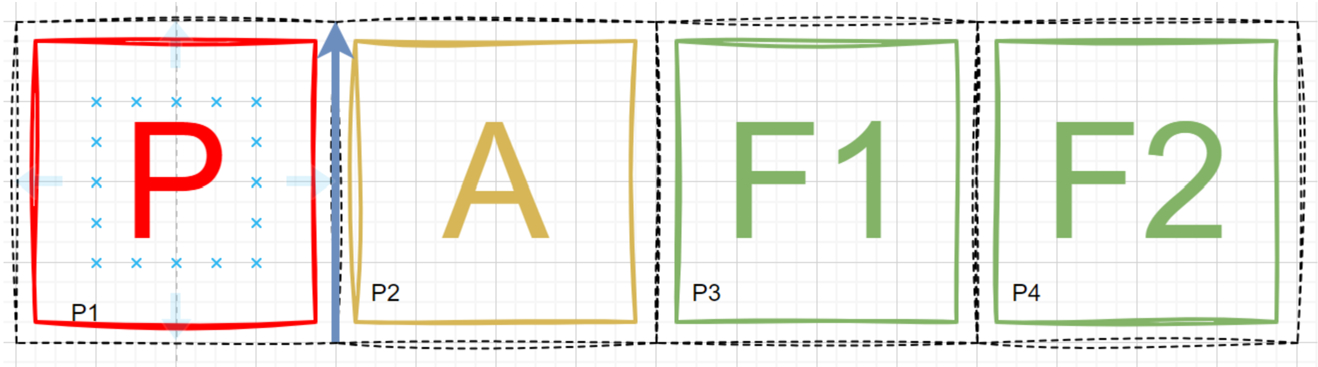
L is a diagonal matrix of eigen values

Coupling transformation $L' = f(L|features)$ must be conditioned on features

$$\boldsymbol{diag(V(1+Q)V^{\dagger})}$$

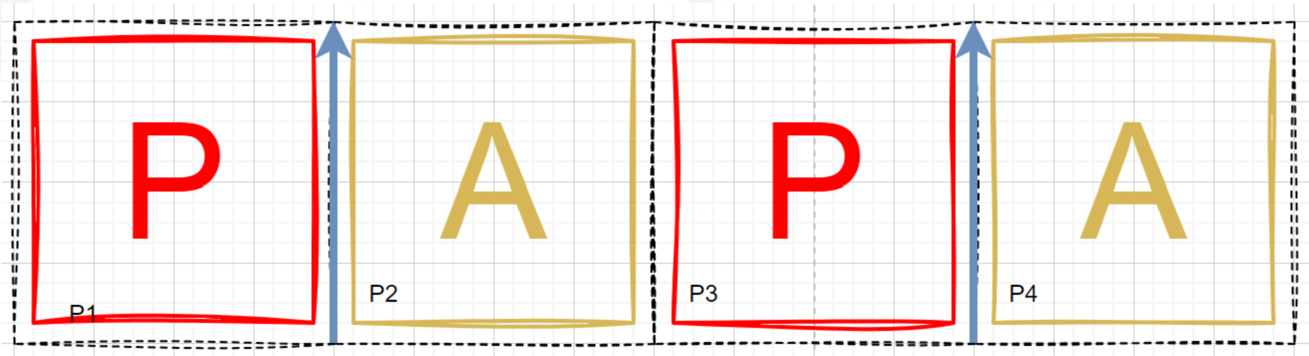
which are <u>gauge-invariant</u> and <u>eigen decomposition invariant</u>.

How would transformation change if we had frozen loops $S = Re\,Tr\,(A+P+F1+F2)$?



- With one iteration we can trivialize only plaquettes (active and passive) which contain active link
- In this case there is no useful information in frozen plaquettes

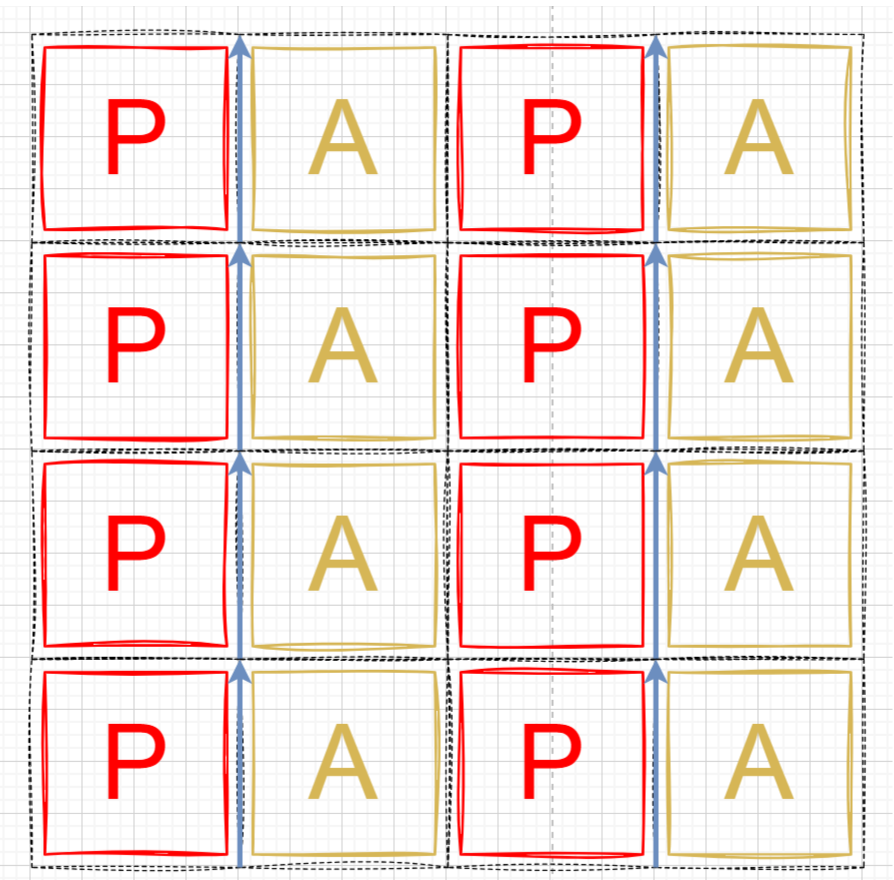
=> With dense mask we can(speaking only about 2D) trivialize all plaquettes in the action



Spectral flow v2

Lessons:

1) Use dense mask as frozen loops contain no useful information



2) Build proper features

$$\boldsymbol{diag(V(1+Q)V^{\dagger})}$$

3) Use expressive transformation (splines)

4) Not all links need to be transformed

Spectral flow Coupling Layer:

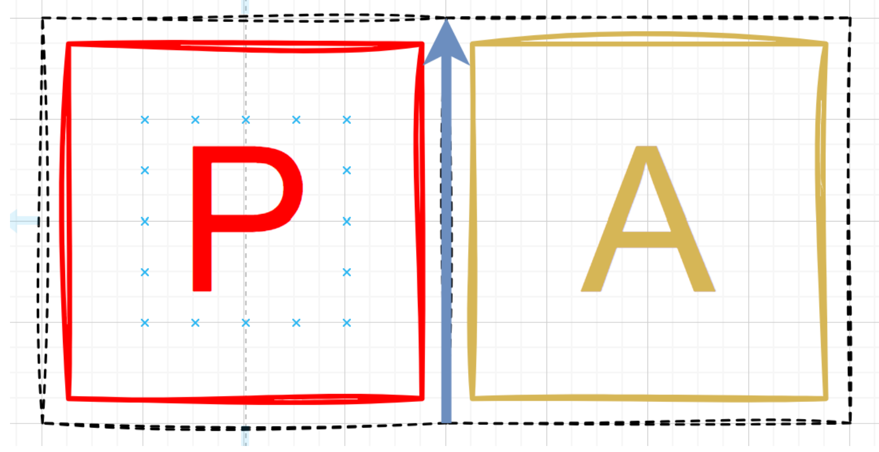
- Apply mask
- Transform active links $U \rightarrow U'$
- Transform active plaquette $A \rightarrow A' = f(A|features)$
  - Diagonalize $A = V^{\dagger} L\, V$
  - Compute loops Q *(more on this later)*
  - Build "diagonal" features $diag(V(1+Q)V^{\dagger})$
  - Transform $L \rightarrow L' = g(L|features)$
  - Undiagonilize $A = V^{\dagger} L'\, V$
- Push update to the link $U \rightarrow U' = A' A^{\dagger} U$

With expressive transformation $g(L|features)$
logJ should trivialize/compensate plaquettes in the action.

Only one coupling layer is needed!

A more deeper look

Target action $S = Re\,Tr\,(A + P)$



In fact, there is a family of solutions. Which should we choose?

$$L' = \int \exp[-Re\,Tr\,(L\,(V(1+Q)V^{\dagger}\,)]\,\frac{Haar(L)}{Haar(L')}\,dL\ * Constant$$

Remember Coupling transformation $L' = f(L|features)$ must be <u>diffeomorphism on circle</u>!

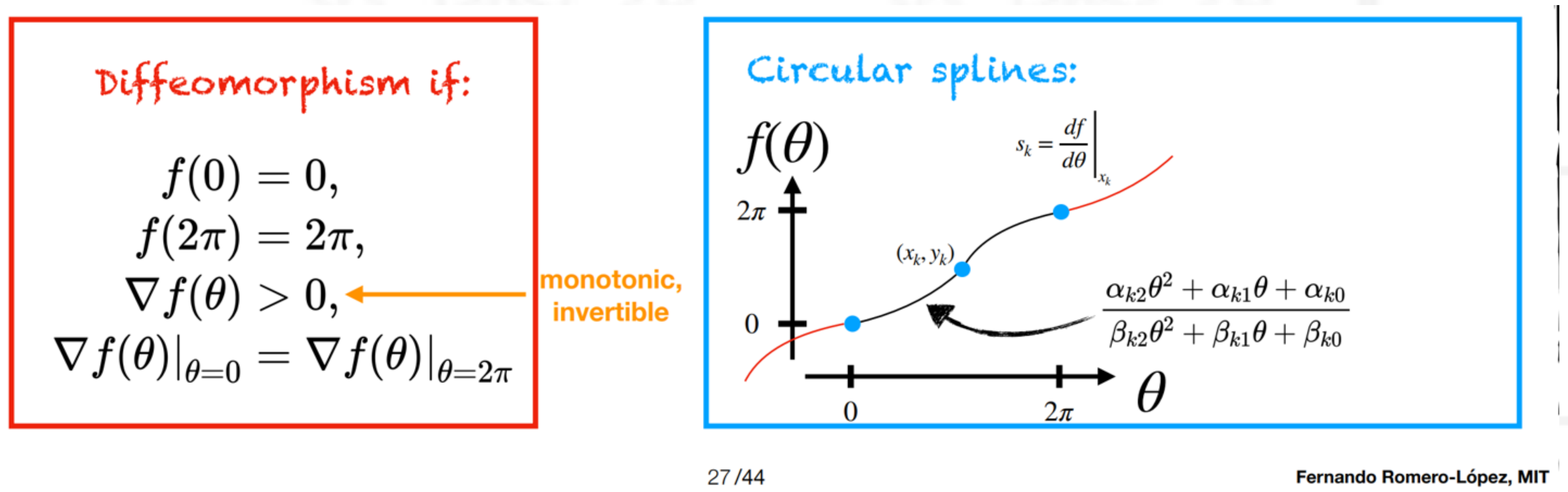


Image credit: F. Romero-López

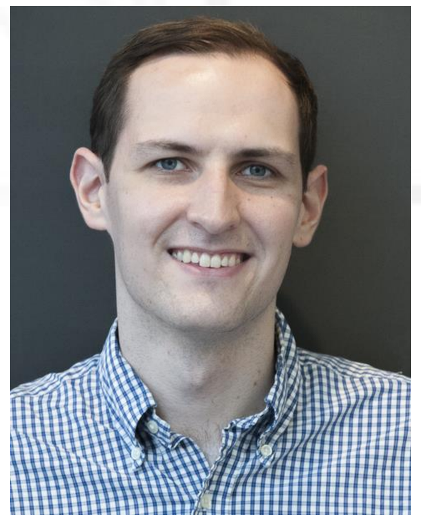
This uniquely determined a solution from this family! Coupling transformation should satisfy:

- $L' = e^{\Theta'=0} = f(L = e^{\Theta=0}|features) \Rightarrow f(I|features) = I$

$$L' = \boxed{\frac{1}{Normalizer(Tr\,(1+Q))}} \int \exp[-Re\,Tr\,(L\,(V(1+Q)V^{\dagger}\,)]\,\frac{Haar(L)}{Haar(L')}\,dL$$

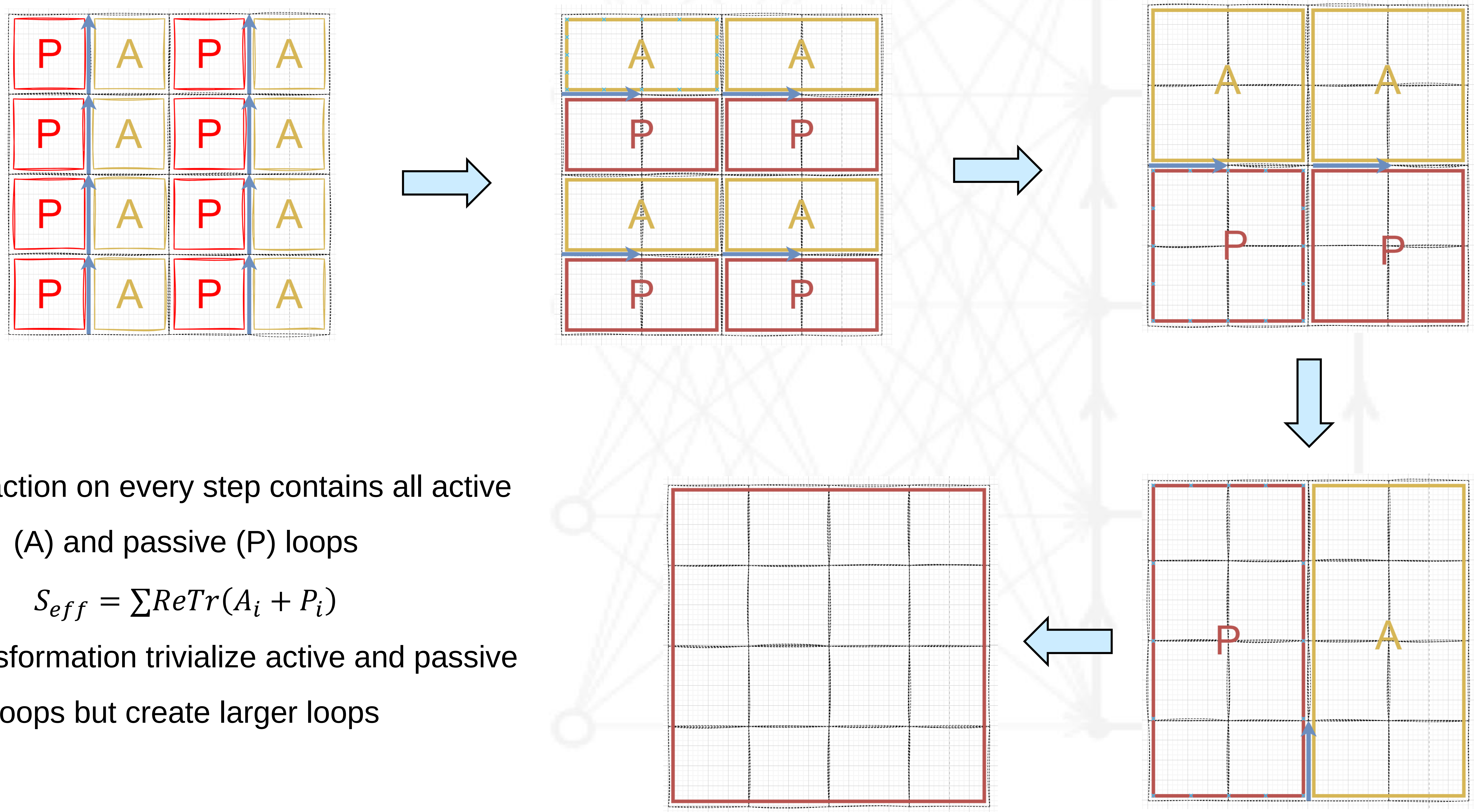
This normalizer will appear in logJ!
As a result after trivializing plaquettes we will have larger loops (Q) in the effective action

<u>Denis</u>: Oh, we need hierarchical algorithm…



<u>Ryan</u>: I have one!

Example trivializing 2D LGT



Effective action on every step contains all active (A) and passive (P) loops

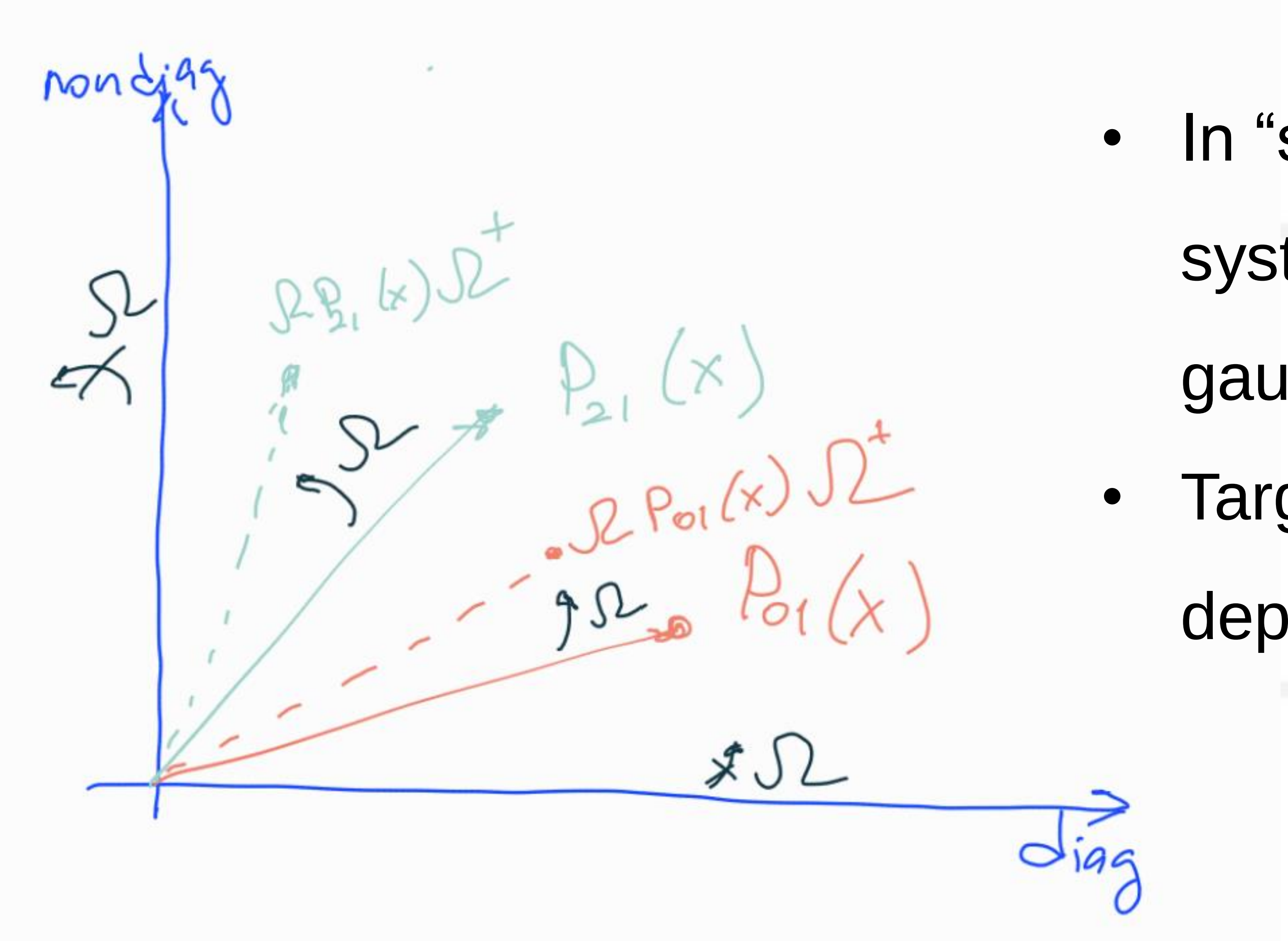
$$S_{eff} = \sum ReTr(A_i + P_i)$$

Every transformation trivialize active and passive loops but create larger loops

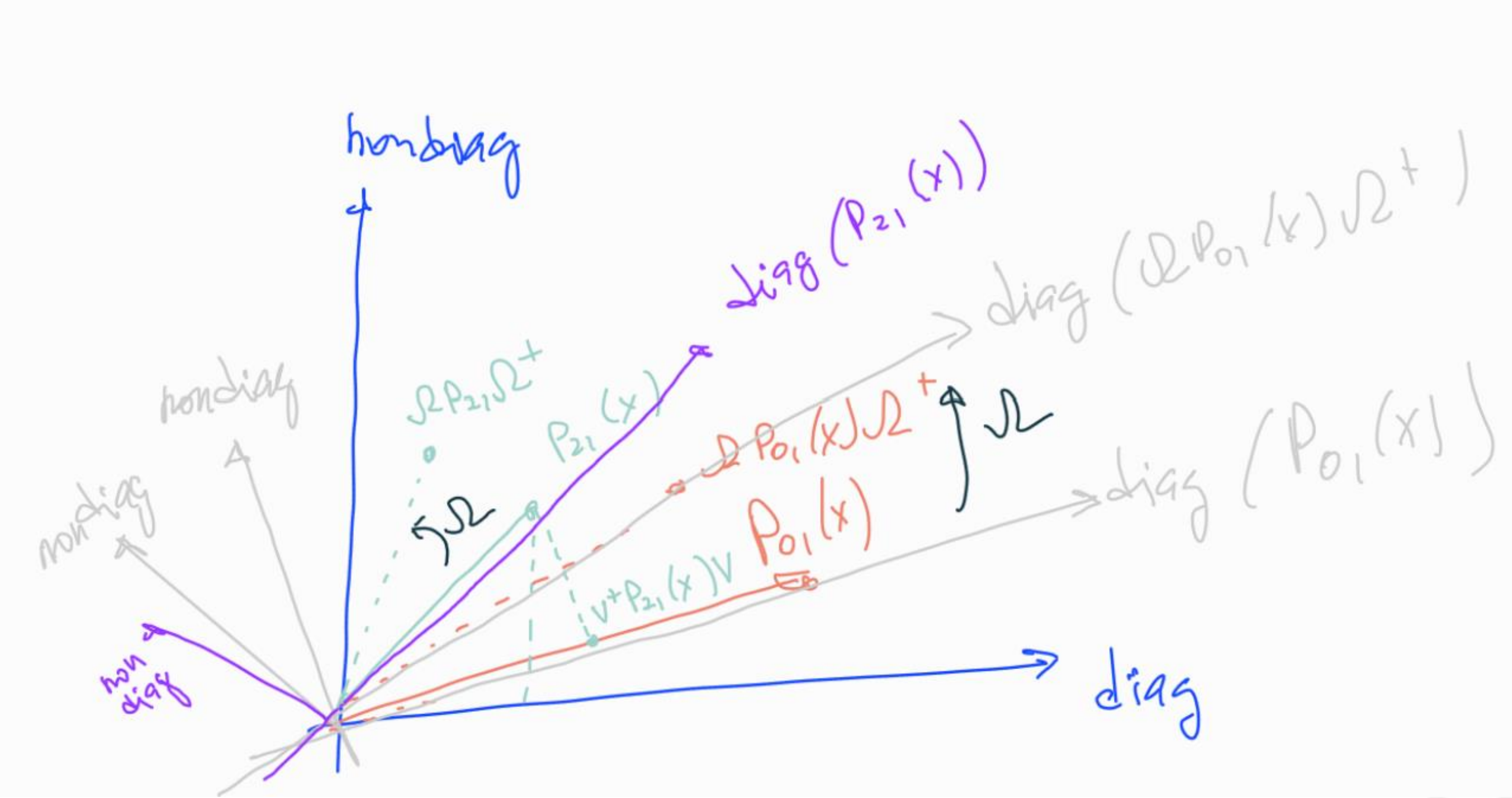
Interpretation

$$W = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \Rightarrow \big(nondiag(W), diag(W)\big)$$

$$nondiag(W) = (b, c, d, f, g, h), \qquad diag(W) = (a, e, i)$$



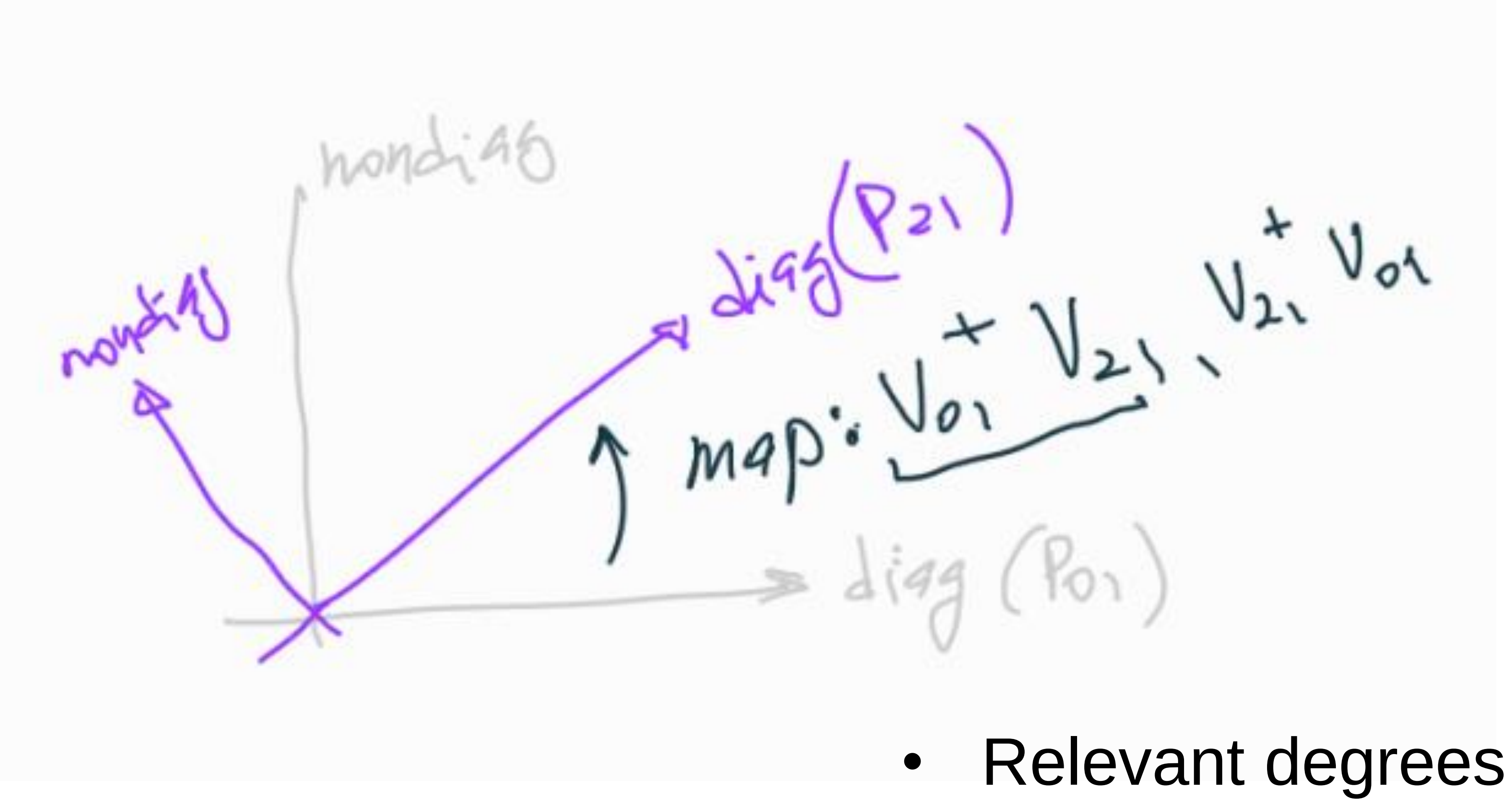
- In “static” (conventional) coordinate system every loop is rotated with a gauge
- Target action is “scalar” and does not depend on choice of coordinate system

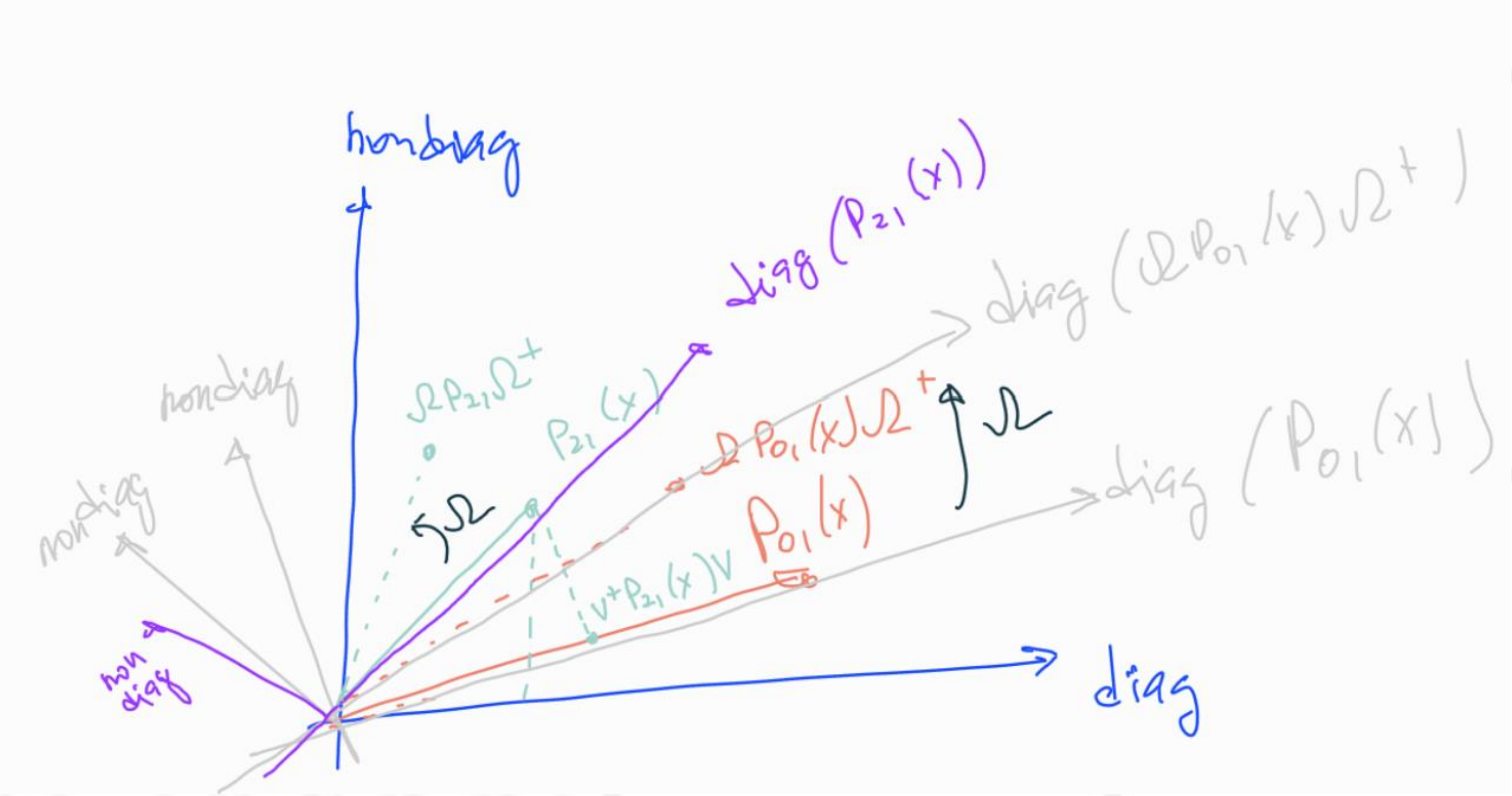


- There is a local coordinate system in which loop is diagonal
- These coordinate systems are not unique and transformed under symmetry (gauge) transformation
- They are transformed in a same way as loops in “static” coordinate system

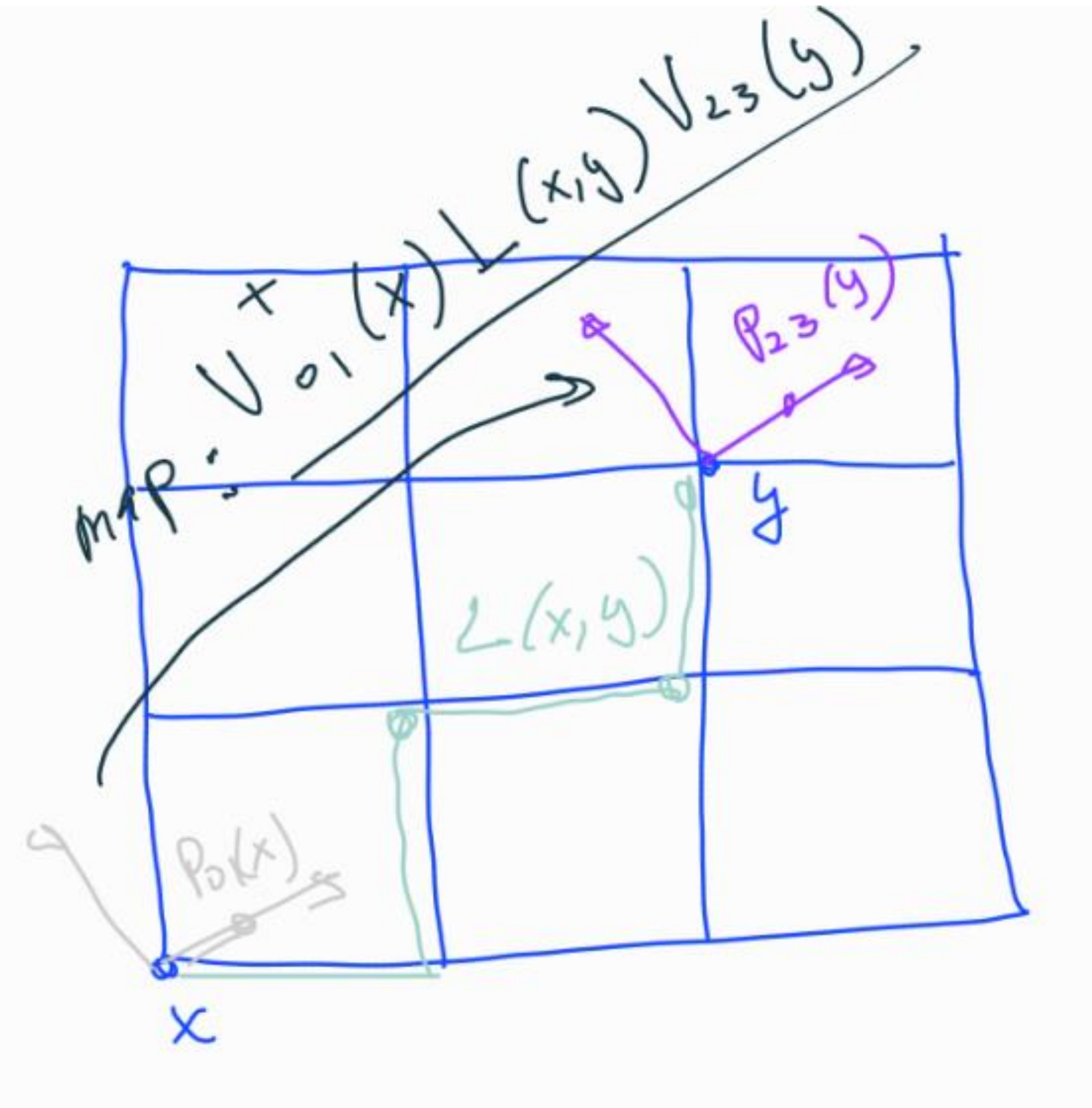
- Every loop is evaluated in local coordinate system (Gauge symmetry)
- Relevant degrees of freedom are transformed in a local coordinate system when loop is diagonal

Interpretation





- Relevant degrees of freedom are transformed in a local coordinate system when loop is diagonal
- Relevant features are loops evaluated **in the same coordinate system**
- Map between different coordinate systems is defined by eigen vectors V

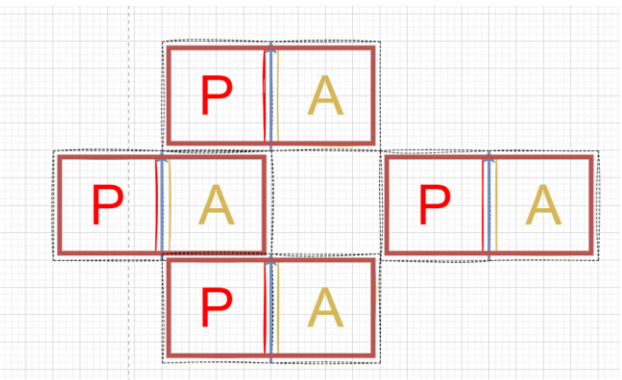


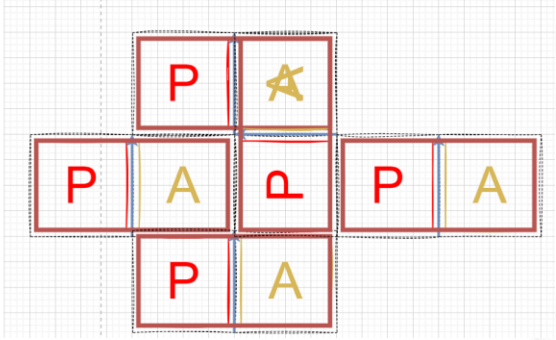
Example

- Active loop $P_{01}(x) = V_{01}(x)^{\dagger} D(x) V_{01}(x)$ is transformed in local C.S. - $D(x)' = f(D(x)|\,.)$
- Conditional information could be a loop $P_{23}(y) = V_{23}(y)^{\dagger} D(y) V_{23}(y)$
- Map is defined by parallel transport and eigenvectors $V_{01}(x)^{\dagger} L(x,y) V_{23}(y)^{\dagger}$
- Transformation $\mathrm{L}'(\mathrm{x}) = \mathrm{f}(\mathrm{L}(\mathrm{x})|\, V(x) U_{\mu} V(x+\mu), \mathrm{L}(\mathrm{x}+\mu))$

Uncovered but Important topics

- **Defects**. 2D LGT can be tilled and trivialized in hierarchical way. However, 3/4D LGT can not be tiled. In 3/4D there are limited number of links/loops which transformed more/less then others, I call them **defects**.





- In **3/4D** to deal with defects more expressive transformation must be used. It results in more complicated features and **autoregressive coupling**.
- **Location update** is one of such techniques, and it requires specific masking.
- We can think of a algorithm which automatically builds necessary featured loops from active links/loops
- In LGT **volume scaling** is determined by residual of interpolation. Keeping residual the same as volume growth will result in a same model quality. Naive analysis predicts increasing number of interpolation intervals as $\sqrt{V}$

Take away message

Blind application of ML techniques delivers not the best results. Approach should be adapted with math and knowledge of the physical domain

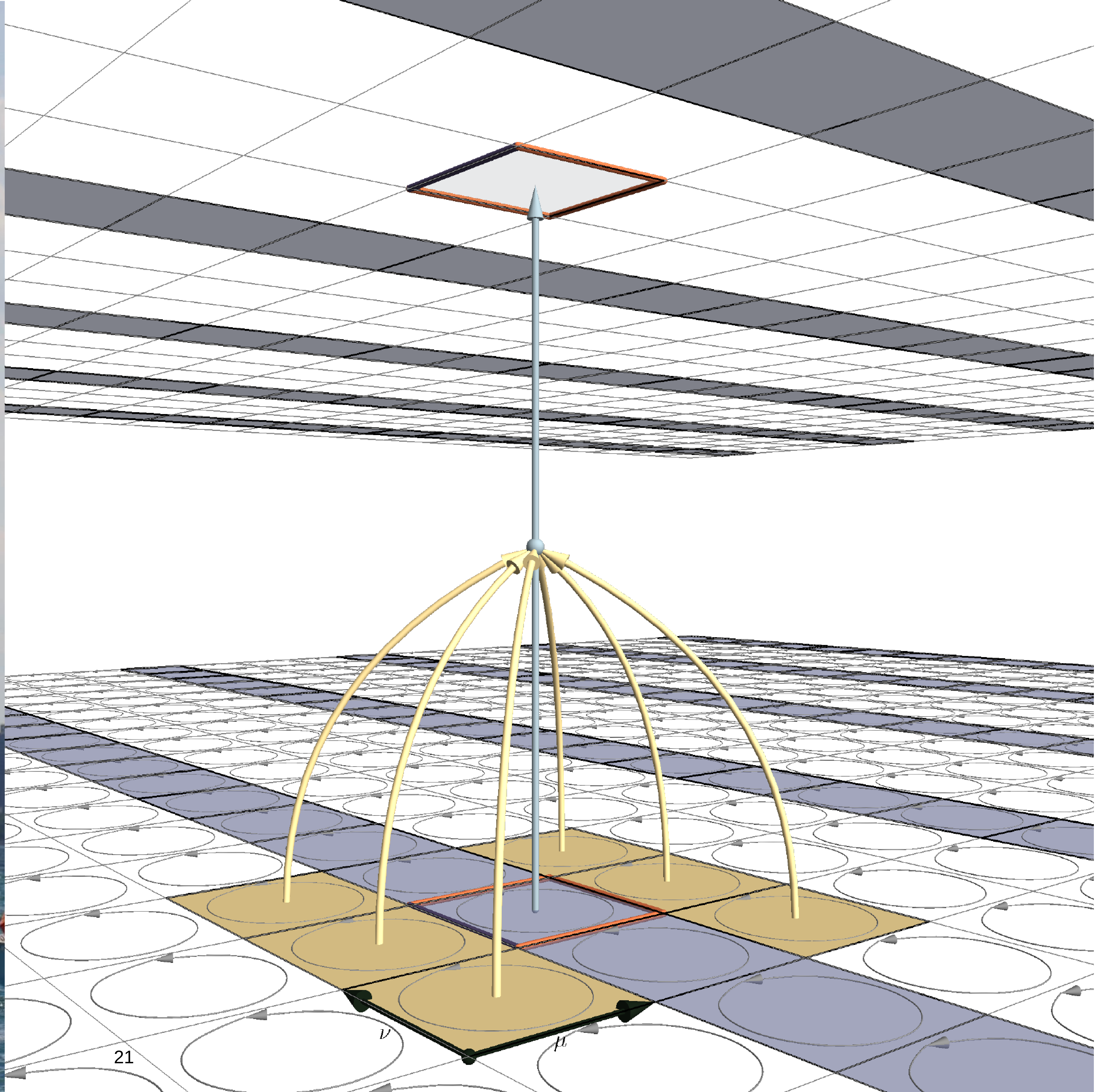
This is not only about trivialization. This is about the expressivity of a general NF. The principle should be replicated for flow between any two distributions.

- Hierarchical approach is necessary for fully trivializing LGT
- LGT can be trivialized with expressive coupling transformation in O(log_2(V)) coupling steps. Model quality is controlled by interpolation residual
- Type of active loops does not matter if relevant features used
- For proper trivialization relevant features must be used. Those are loops evaluated in local diagonal coordinate system
- 2D LGT can be tiled and trivialized in hierarchical way for log_2(V). 3D/4D LGT has defects, loops transformed more/less than others. This results in necessary of creating specific mask, specific features and autoregressive transformation.

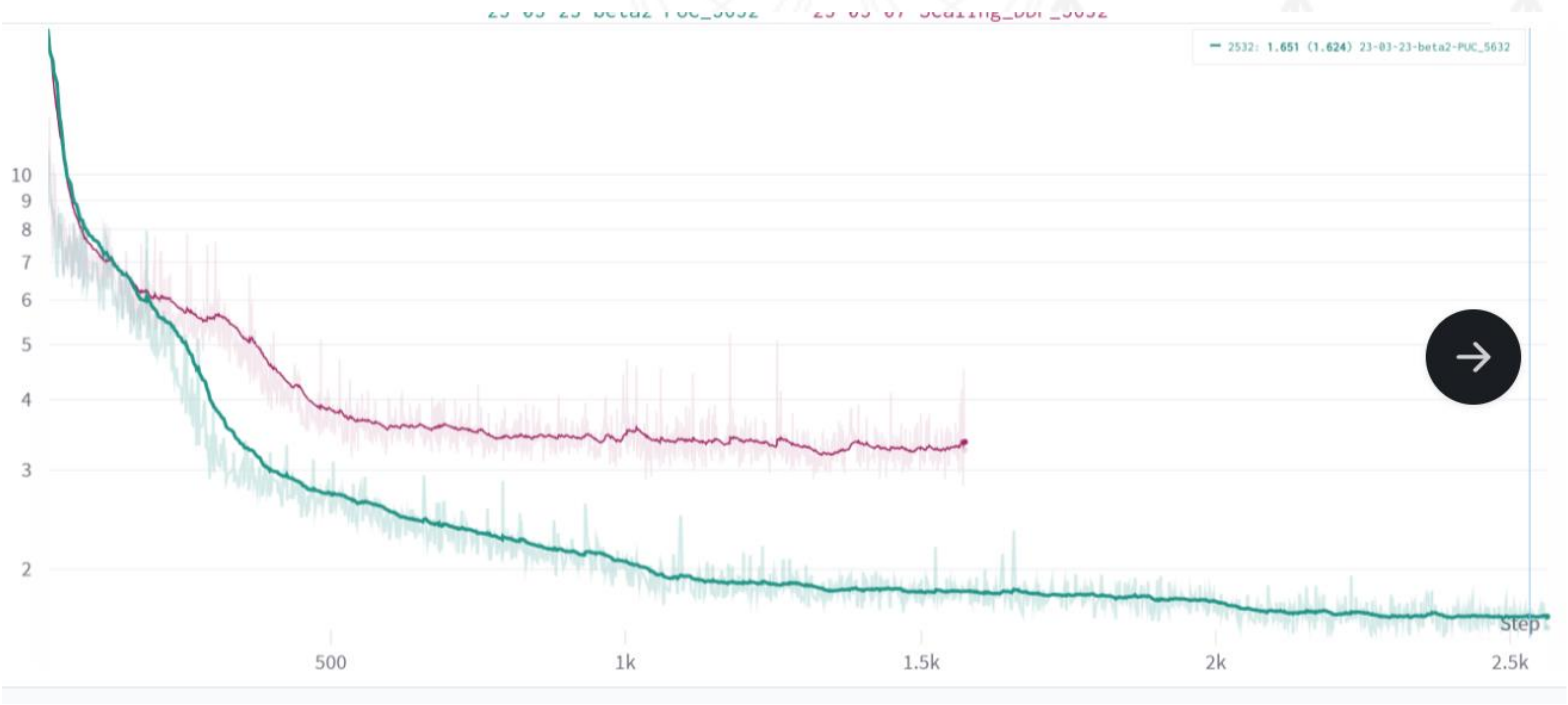
I am glad to be here

Thank you!

Question?



Back up


2532: 1.651 (1.624) 23-03-23-beta2-PUC_5632
10
9
8
7
6
5
4
3
2
500
1k
1.5k
2k
2.5k
Step