UPPSALA
UNIVERSITET

UNIVERSITY OF
CAMBRIDGE

Steering in Theory Space: Representation Engineering of a Lagrangian Transformer

Yong Sheng Koay[1], Alessandro Favero[2], Sven Krippendorf[2]
[1]Department of Physics and Astronomy, Uppsala University, Sweden
[2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK

[Ongoing-work:260X.XXXXX]

UPPSALA UNIVERSITET

UNIVERSITY OF CAMBRIDGE

Steering in Theory Space: Representation Engineering of a Lagrangian Transformer

Yong Sheng Koay[1], Alessandro Favero[2], Sven Krippendorf[2]
[1]Department of Physics and Astronomy, Uppsala University, Sweden
[2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK

[Ongoing-work:260X.XXXXX]

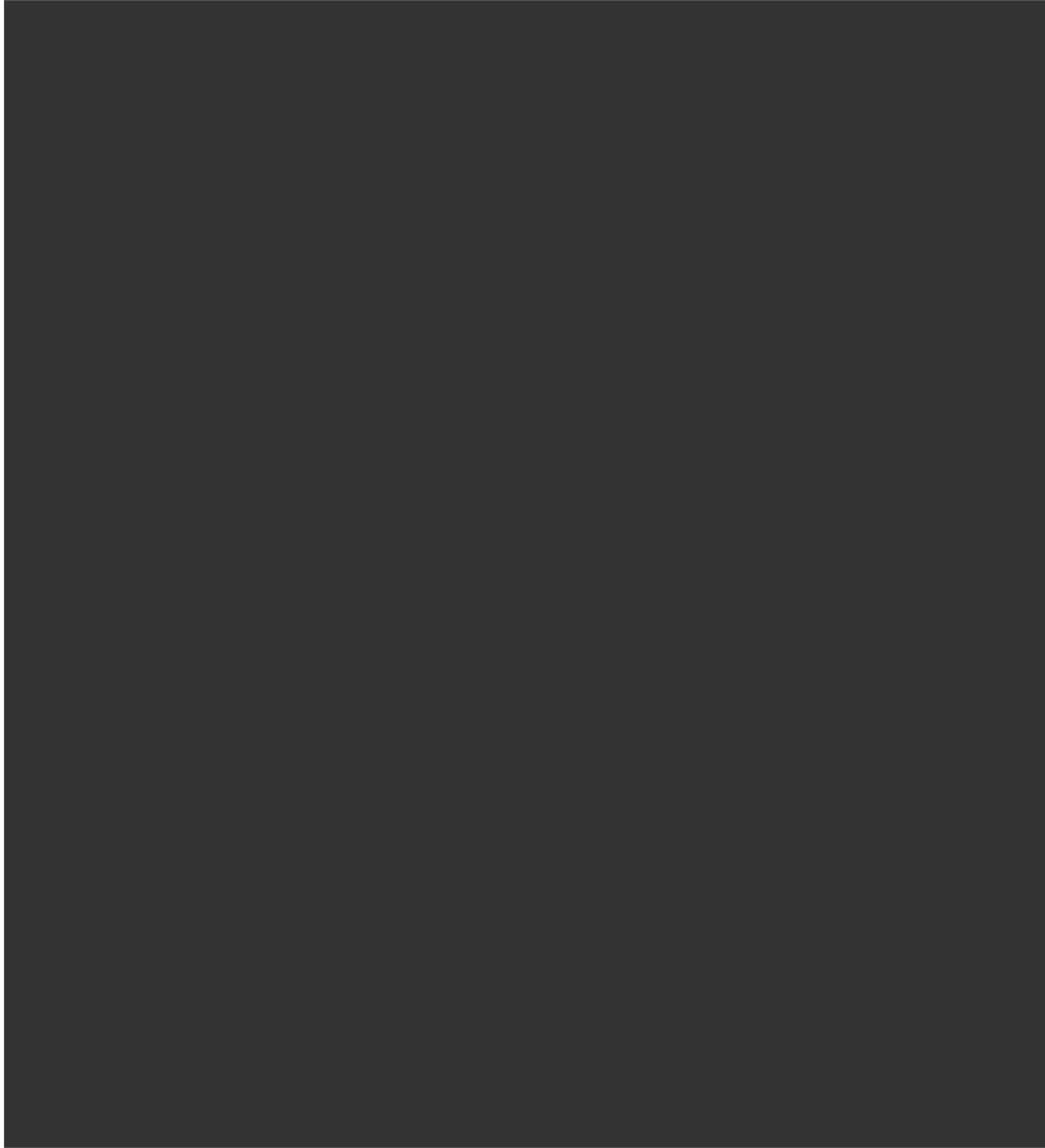
Mentioned work in this talk:

Generating particle physics Lagrangians with transformers

YSK , Rikard Enberg[1] , Stefano Moretti[1,2] and Eliel Camargo-Molina[1,3†]
1 Department of Physics and Astronomy, Uppsala University, Sweden
2 School of Physics and Astronomy, University of Southampton, UK
3 Department of Game Design, Uppsala University, Sweden

[arxiv: 2501.09729]

How does language works?



How does language works?

v o a g u p

Alphabets

How does language works?

you

Alphabets

Words

How does language works?

give you up

Alphabets

Words

Grammar

How does language works?

Never gonna give you up, ...

Alphabets

Words

Grammar

Meaning

How does language works?

Never gonna give you up, ...

Rick Astley - Never Gonna Give You Up

Alphabets

Words

Grammar

Meaning

How does language works?

Never gonna give you up, ...

Rick Astley - Never Gonna Give You Up

Alphabets

Words

Grammar

Meaning

How does equations work?

How does language works?

Never gonna give you up, ...

Rick Astley - Never Gonna Give You Up

Alphabets ~ Symbols

Words

Grammar

Meaning

How does equations work?

$$\sin \quad x^{3} \quad + \quad - \quad !$$

How does language works?

Never gonna give you up, ...

Rick Astley - Never Gonna Give You Up

How does equations work?

$$\frac{x^3}{3!}$$

Alphabets ~ Symbols

Words ~ Terms

Grammar

Meaning

How does language works?

Never gonna give you up, ...

How does equations work?

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Alphabets ~ Symbols

Words ~ Terms

Grammar ~ Math Structures

Meaning

Rick Astley - Never Gonna Give You Up

How does language works?

Never gonna give you up, ...

Rick Astley - Never Gonna Give You Up

How does equations work?

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Alphabets ~ Symbols

Words ~ Terms

Grammar ~ Math Structures

Meaning ~ Real World

How does language works?

Never gonna give you up, ...

Rick Astley - Never Gonna Give You Up

How does equations work?

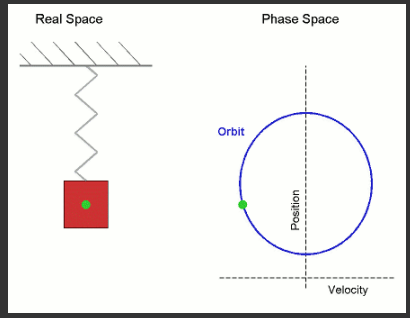
$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

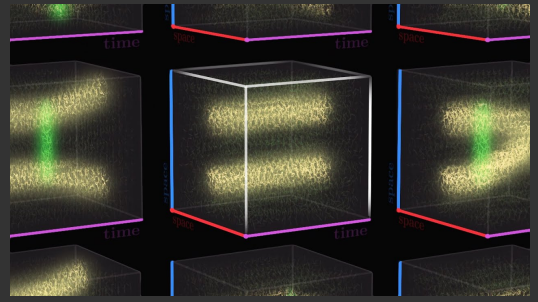
Alphabets ~ Symbols

Words ~ Terms

Grammar ~ Math Structures

Meaning ~ Real World

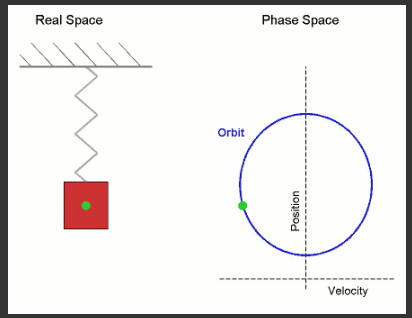




language ~ equations

Never gonna give you up, ...

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$



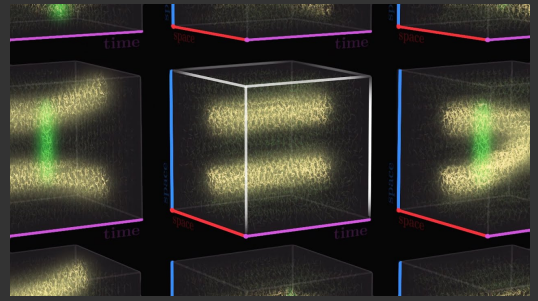
Alphabets ~ Symbols

Words ~ Terms

Grammar ~ Math Structures

Meaning ~ Real World

Rick Astley - Never Gonna Give You Up



language ~ equations

Never gonna give you up

$$\sin(x) = x - \frac{x^3}{\quad} + \frac{x^5}{\quad} - \frac{x^7}{\quad} + \dots$$

Transformer+Language

↓

Crazy stuff since then!

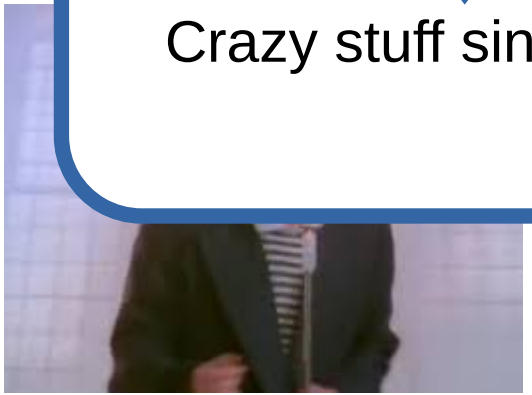
Transformer+Equations?

ts ~ Sym

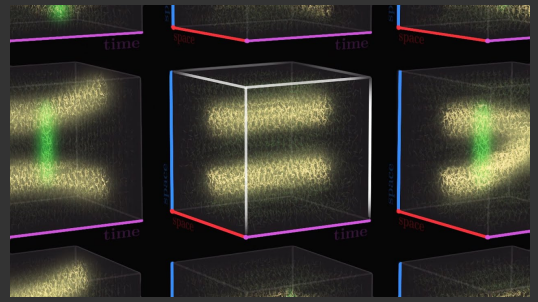
ds ~ Tern

lar ~ Math

Meaning ~ Real World

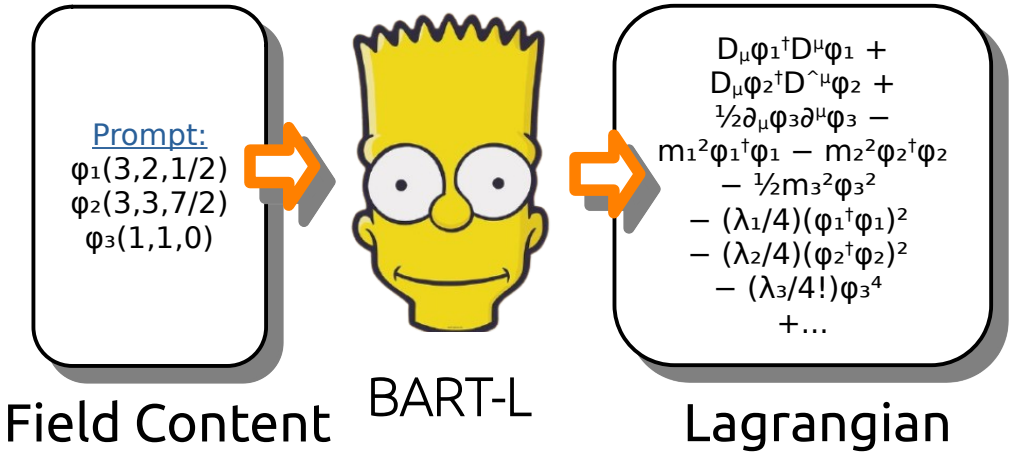


Rick Astley - Never Gonna Give You Up



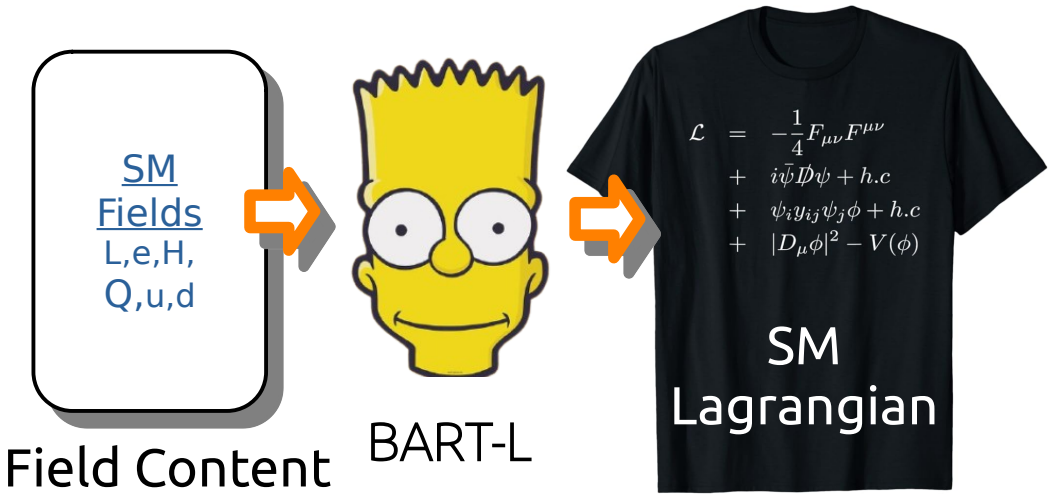
Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729



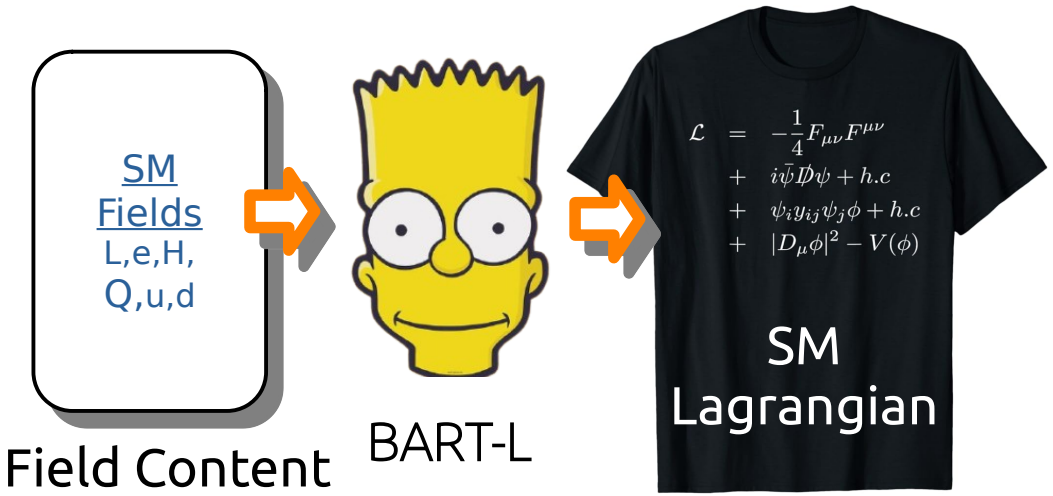
Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729



Lagrangian Transformer, BART-L

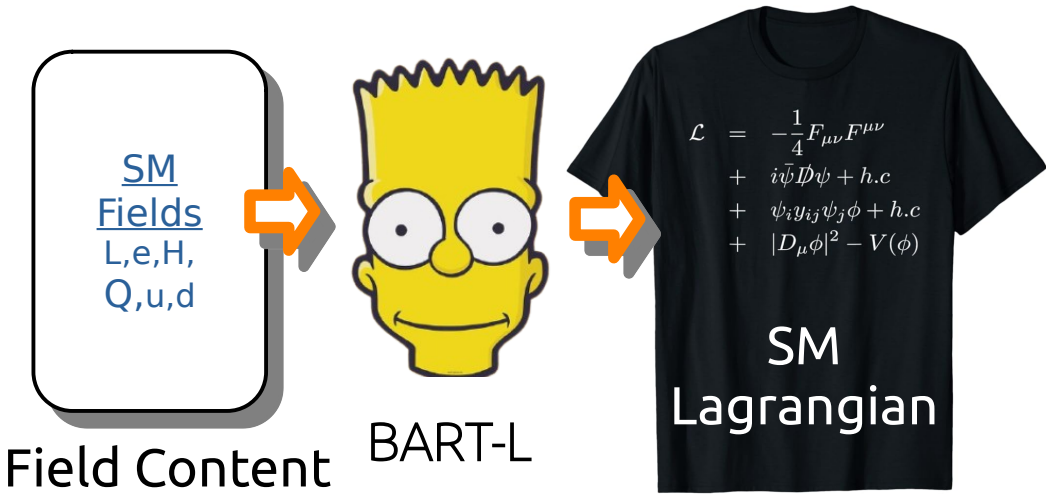
YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729



- Encoder-Decoder [357M parameters]
- Trained from Scratch, Doesnt speak English
- SU(3)xSU(2)xU(1) group, up to 6 matter fields
- 90~97% success rate
- Multitask by nature → THEP foundation model

Lagrangian Transformer, BART-L

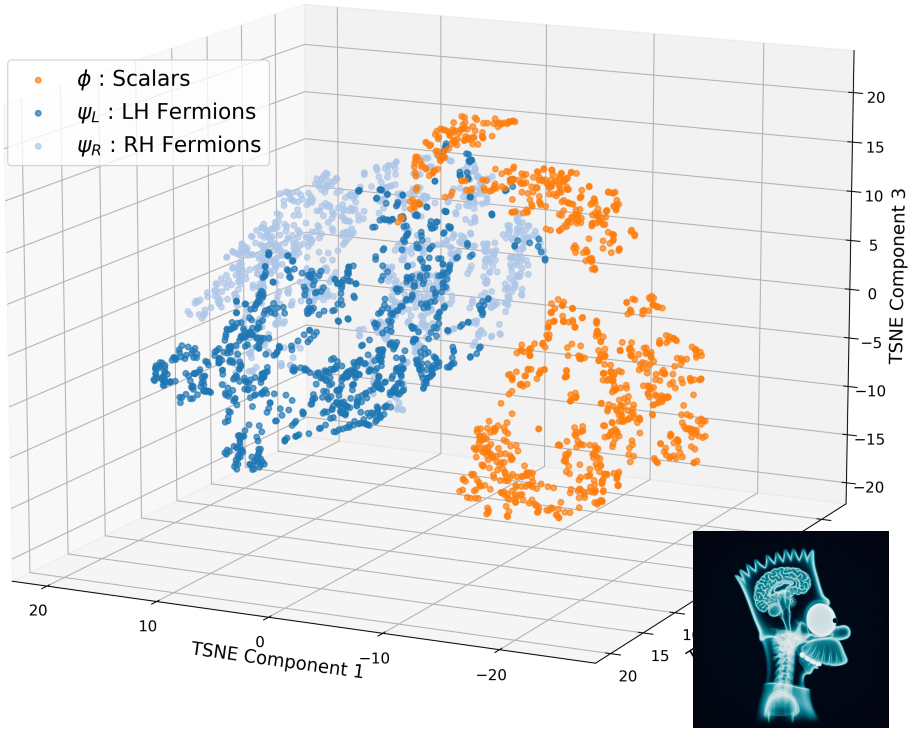
YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729



- Encoder-Decoder [357M parameters]
- Trained from Scratch, Doesnt speak English
- SU(3)xSU(2)xU(1) group, up to 6 matter field
- 90~97% success rate
- Multitask by nature → THEP foundation moc

Embedding Analysis

(latent/activation space)



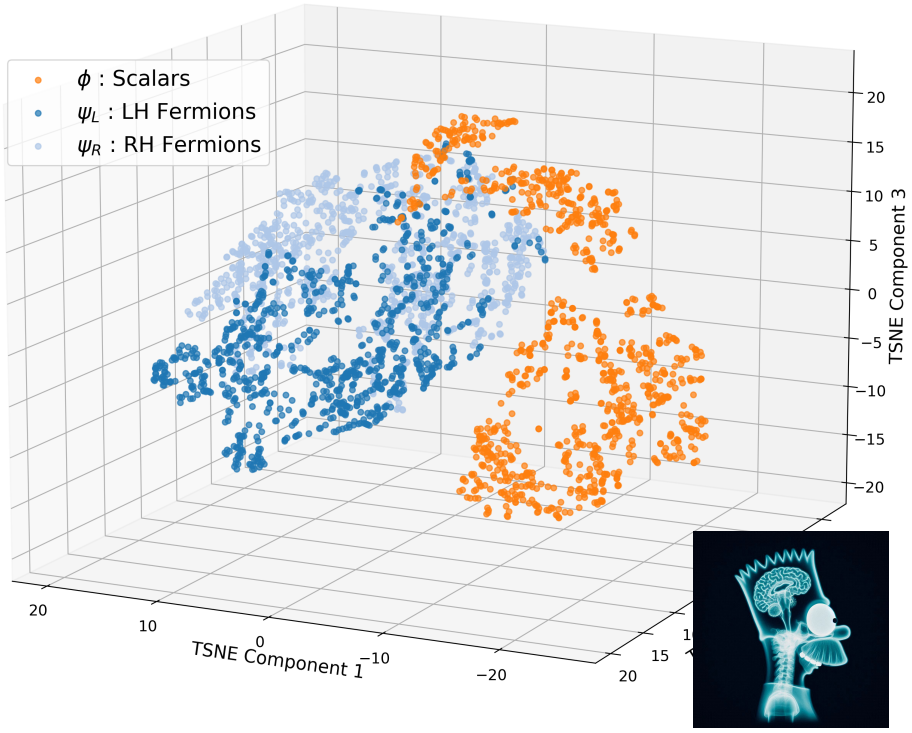
Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

Latent space ~ Learned Theory Space
Each point = one model

Embedding Analysis

(latent/activation space)



Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

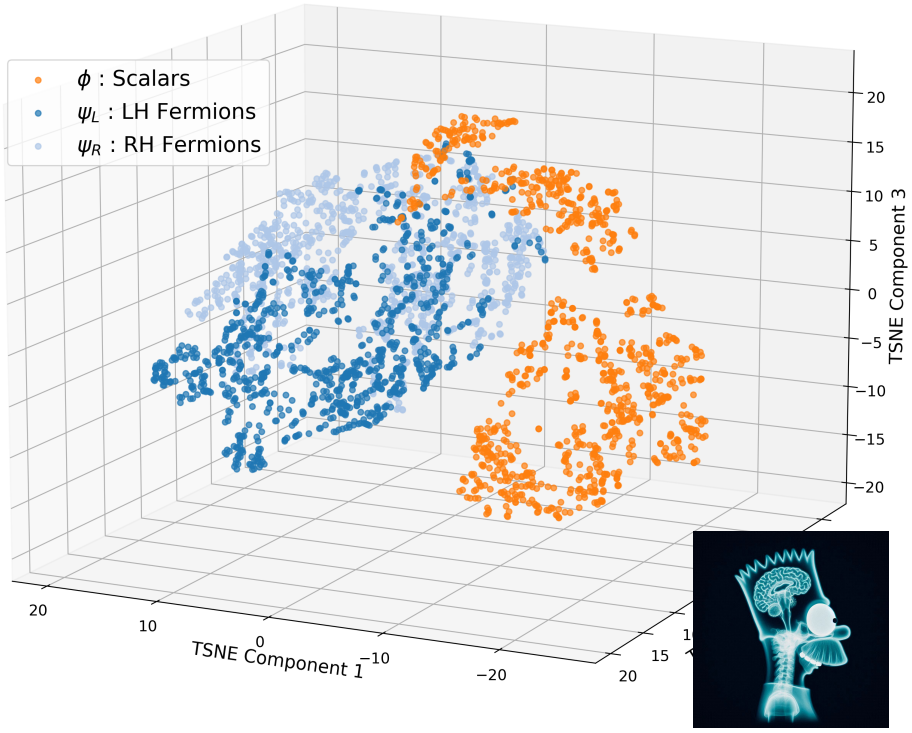
Latent space ~ Learned Theory Space
Each point = one model

Interpretability

Is this a good learned theory space?

Embedding Analysis

(latent/activation space)



Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

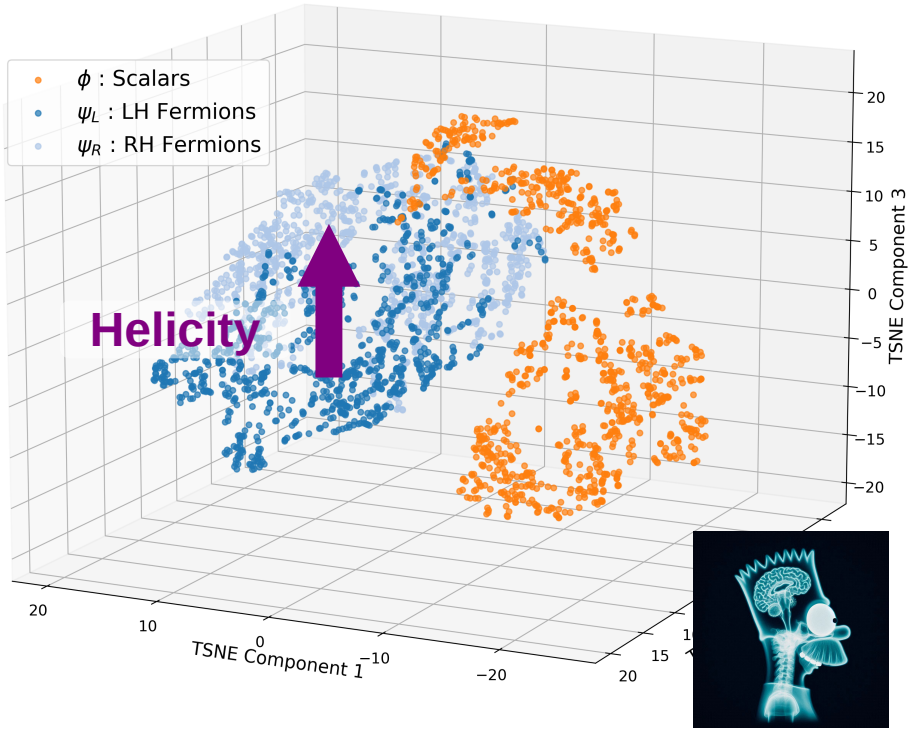
Latent space ~ Learned Theory Space
Each point = one model

Interpretability

Is this a good learned theory space?
Does it has physics relation in it?

Embedding Analysis

(latent/activation space)



Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

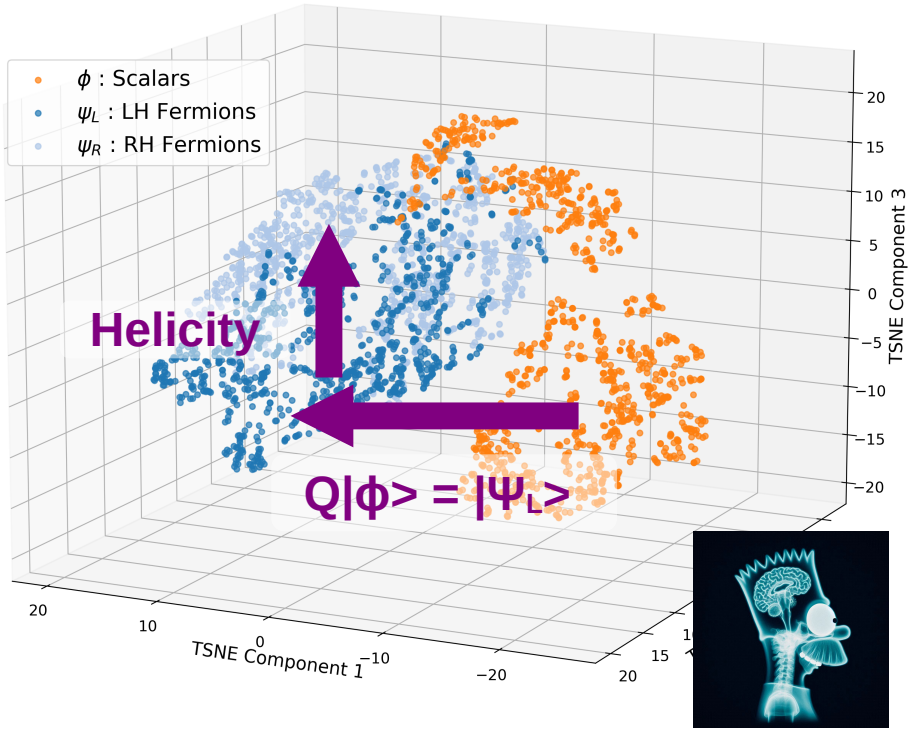
Latent space ~ Learned Theory Space
Each point = one model

Interpretability

Is this a good learned theory space?
Does it has physics relation in it?

Embedding Analysis

(latent/activation space)



Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

Latent space ~ Learned Theory Space
Each point = one model

Interpretability

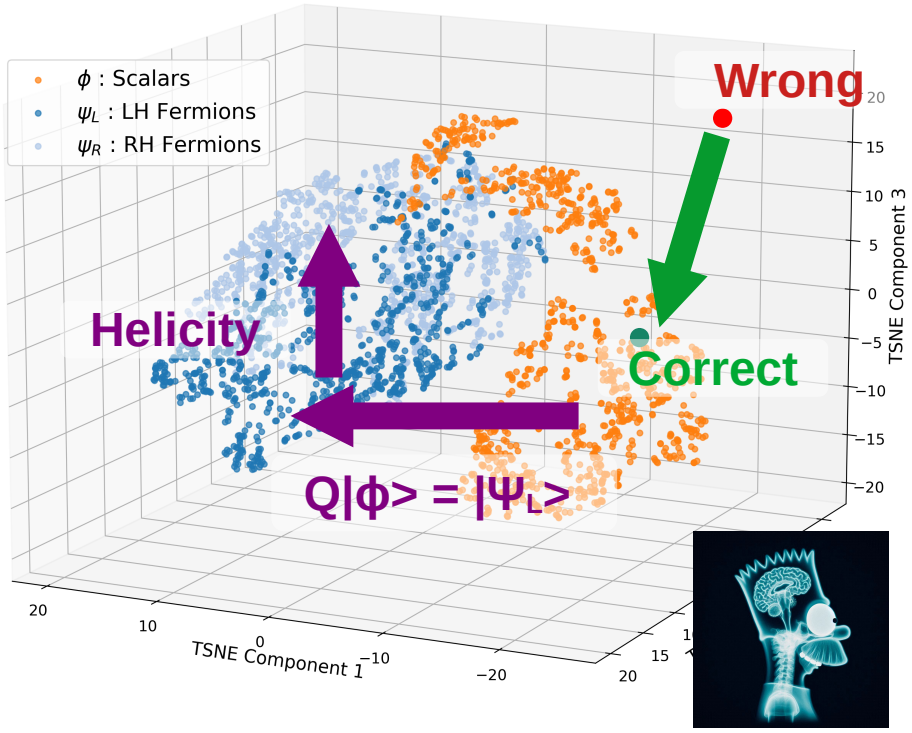
Is this a good learned theory space?
Does it has physics relation in it?

Navigation

Can we move around this space?
From wrong theories to correct one?

Embedding Analysis

(latent/activation space)



Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

Latent space ~ Learned Theory Space
Each point = one model

Interpretability

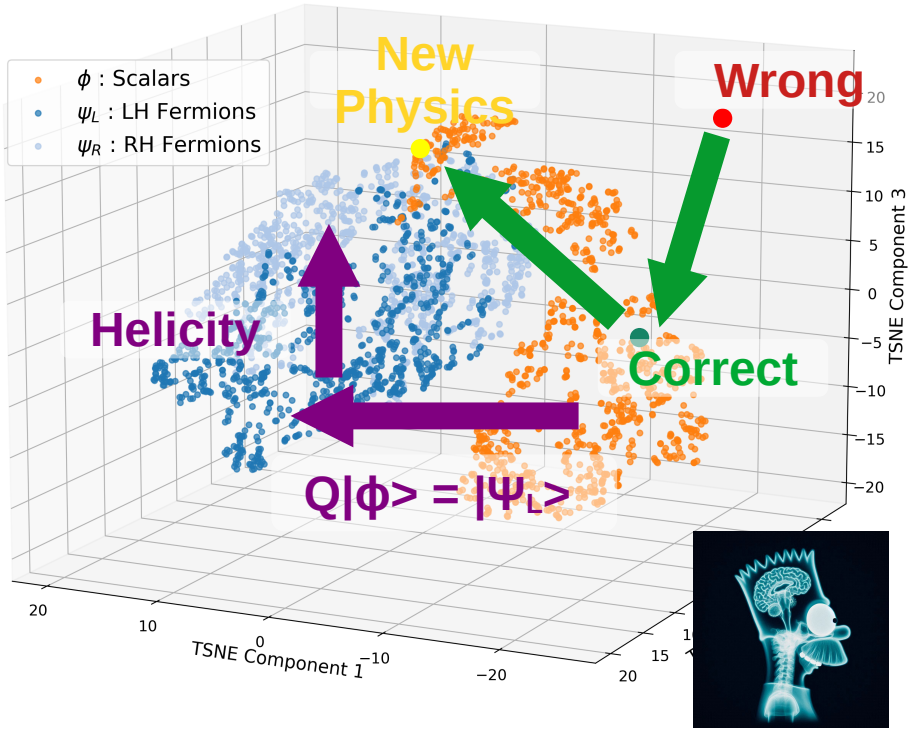
Is this a good learned theory space?
Does it has physics relation in it?

Navigation

Can we move around this space?
From wrong theories to correct one?
From correct theories to better ones?

Embedding Analysis

(latent/activation space)



Lagrangian Transformer, BART-L

YSK, Rikard Enberg, Stefano Moretti, Eliel Camargo-Molina. (2026), SciPost Phys. 21, 024. arXiv:2501.09729

Latent space ~ Learned Theory Space
Each point = one model

Embedding Analysis

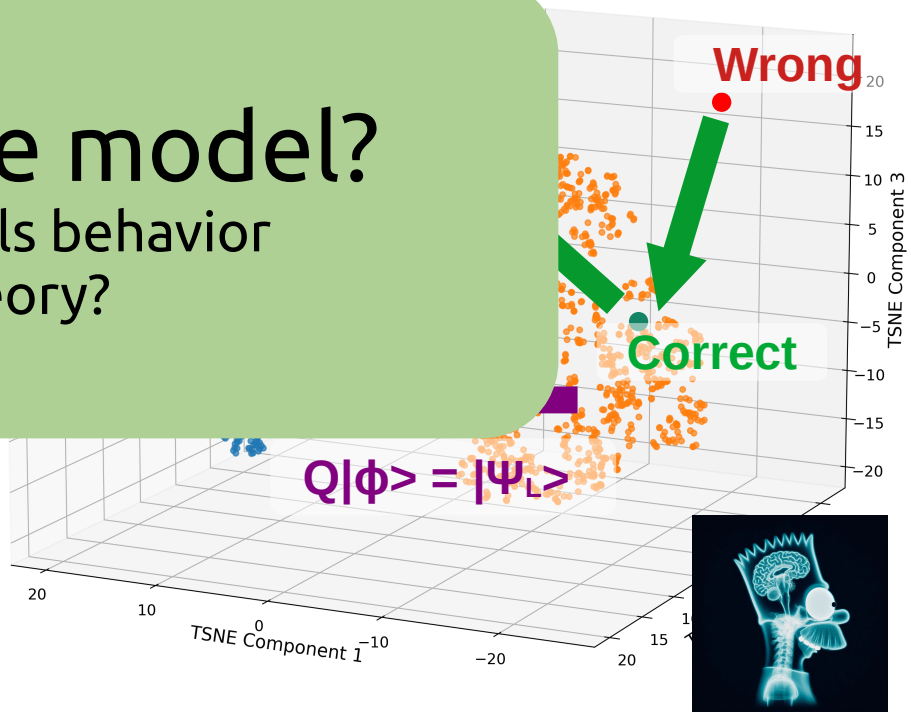
(latent/activation space)

Is this
Doe

Can we ‘steer’ the model?

Can we change the models behavior
as it writes the theory?

Can we move around this space?
From wrong theories to correct one?
From correct theories to better ones?



Contrastive Activation Addition

Rimsky et al 2312.06681

Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

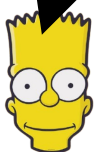
Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 + i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 + \frac{1}{2}(m_1 \psi_1 \psi_1 + m_2 \psi_2 \psi_2 + h.c.)$



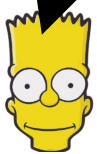
Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 + i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 + \frac{1}{2}(m_1 \psi_1 \psi_1 + m_2 \psi_2 \psi_2 + \text{h.c.})$



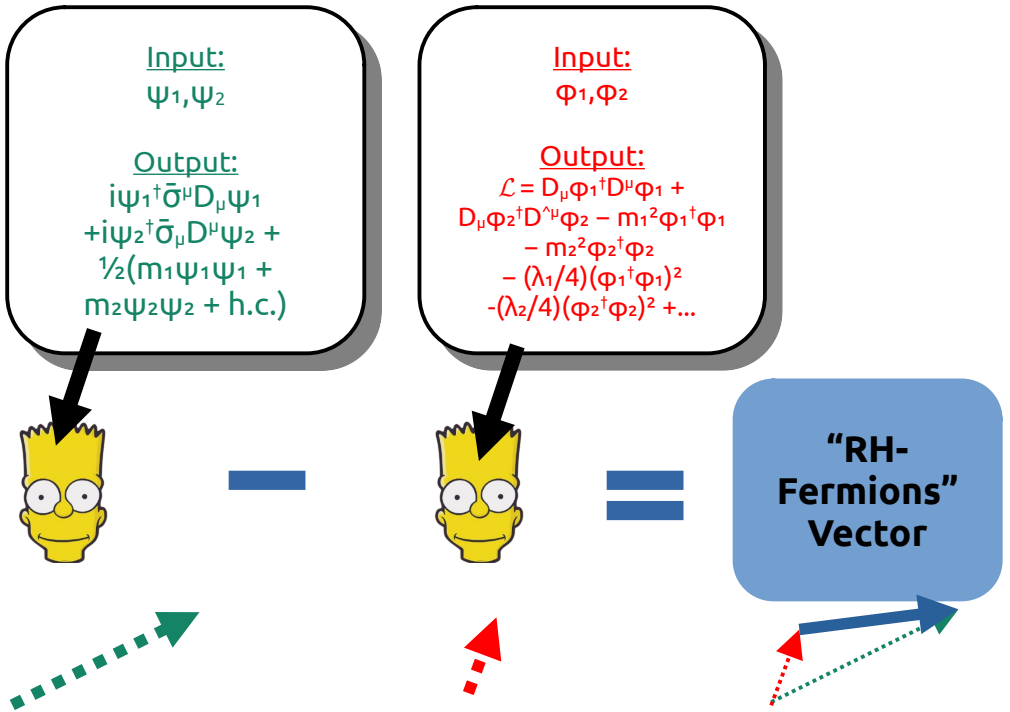
Input:
φ_1, φ_2

Output:
$\mathcal{L} = D_\mu \varphi_1^\dagger D^\mu \varphi_1 + D_\mu \varphi_2^\dagger D^\mu \varphi_2 - m_1^2 \varphi_1^\dagger \varphi_1 - m_2^2 \varphi_2^\dagger \varphi_2 - (\lambda_1/4)(\varphi_1^\dagger \varphi_1)^2 - (\lambda_2/4)(\varphi_2^\dagger \varphi_2)^2 + \ldots$

Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

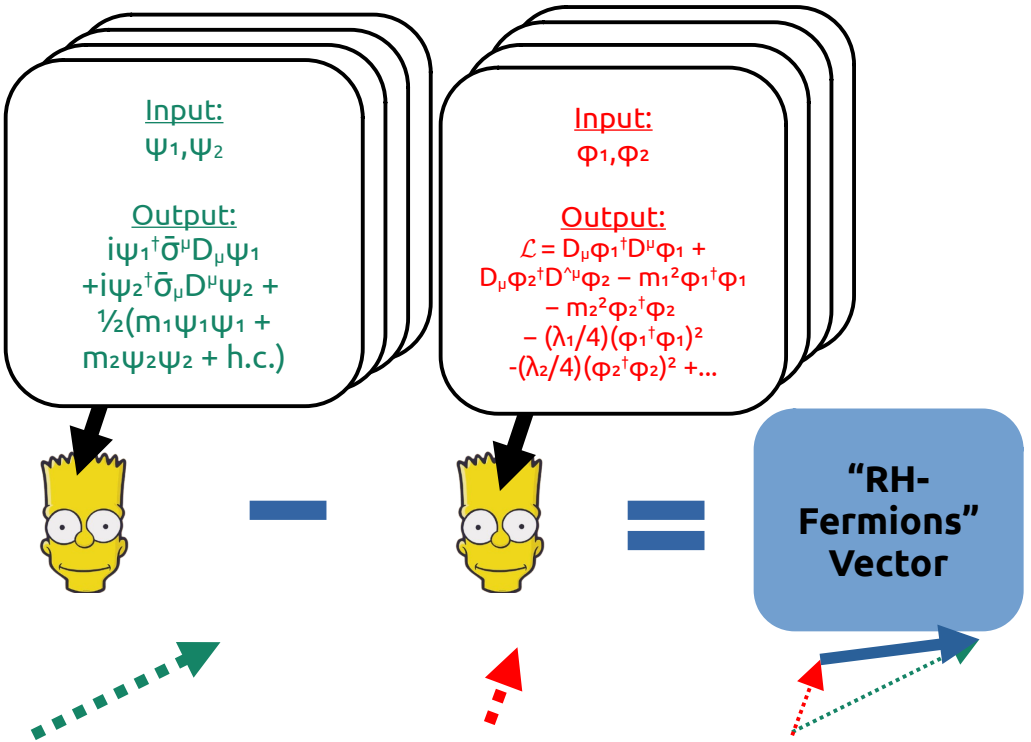


Subtract contrasting prompt activations to isolate a concept.

Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector



Subtract contrasting prompt activations to isolate a concept.

Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1$
$+i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 +$
$\frac{1}{2}(m_1\psi_1\psi_1 +$
$m_2\psi_2\psi_2 + h.c.)$

−

Input:
φ_1, φ_2

Output:
$\mathcal{L} = D_\mu\varphi_1^\dagger D^\mu\varphi_1 +$
$D_\mu\varphi_2^\dagger D^\mu\varphi_2 - m_1^2\varphi_1^\dagger\varphi_1$
$- m_2^2\varphi_2^\dagger\varphi_2$
$- (\lambda_1/4)(\varphi_1^\dagger\varphi_1)^2$
$-(\lambda_2/4)(\varphi_2^\dagger\varphi_2)^2 + \ldots$

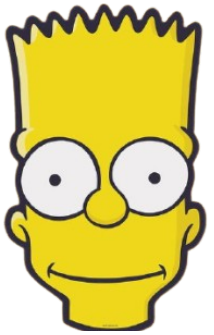
=

"RH-Fermions" Vector

2. Steer!

Prompt:
$\varphi_1(3,2,1/2)$
$\varphi_2(3,3,7/2)$
$\varphi_3(1,1,0)$

[Scalar fields only]



BART-L

Subtract contrasting prompt activations to isolate a concept.

Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1$
$+i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 +$
$\frac{1}{2}(m_1\psi_1\psi_1 +$
$m_2\psi_2\psi_2 + h.c.)$

−

Input:
φ_1, φ_2

Output:
$\mathcal{L} = D_\mu\varphi_1^\dagger D^\mu\varphi_1 +$
$D_\mu\varphi_2^\dagger D^\mu\varphi_2 - m_1^2\varphi_1^\dagger\varphi_1$
$- m_2^2\varphi_2^\dagger\varphi_2$
$-(\lambda_1/4)(\varphi_1^\dagger\varphi_1)^2$
$-(\lambda_2/4)(\varphi_2^\dagger\varphi_2)^2 + \ldots$

=

"RH-Fermions" Vector

Subtract contrasting prompt activations
to isolate a concept.

2. Steer!

Prompt:
$\varphi_1(3,2,1/2)$
$\varphi_2(3,3,7/2)$
$\varphi_3(1,1,0)$

[Scalar fields only]

RH Fermions!

BART-L

Inject the steering vector
into model activations during generation.

Contrastive Activation Addition

Rimsky et al 2312.06681

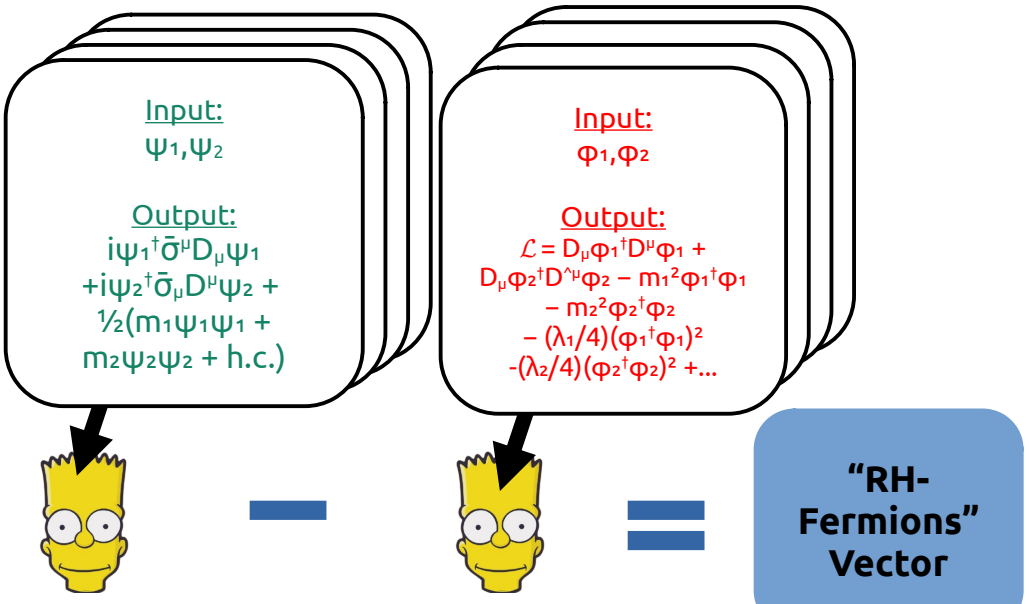
1. Getting steering vector

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1$
$+i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 +$
$\frac{1}{2}(m_1\psi_1\psi_1 +$
$m_2\psi_2\psi_2 + h.c.)$

Input:
φ_1, φ_2

Output:
$\mathcal{L} = D_\mu\varphi_1^\dagger D^\mu\varphi_1 +$
$D_\mu\varphi_2^\dagger D^\mu\varphi_2 - m_1^2\varphi_1^\dagger\varphi_1$
$- m_2^2\varphi_2^\dagger\varphi_2$
$- (\lambda_1/4)(\varphi_1^\dagger\varphi_1)^2$
$-(\lambda_2/4)(\varphi_2^\dagger\varphi_2)^2 + \ldots$



"RH-Fermions" Vector

Subtract contrasting prompt activations to isolate a concept.

2. Steer!

Prompt:
$\varphi_1(3,2,1/2)$
$\varphi_2(3,3,7/2)$
$\varphi_3(1,1,0)$

[Scalar fields only]

BART-L

$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 +$
$i\psi_2^\dagger \bar{\sigma}_\mu D \psi_2 +$
$i\psi_3^\dagger \bar{\sigma}_\mu \partial_\mu \psi_3 -$
$\frac{1}{2}(m_1\psi_1\psi_1 +$
$m_2\psi_2\psi_2 + m_3\psi_3\psi_3$
$+ h.c.)$

[**Fermion** only Lagrangian]

Inject the steering vector into model activations during generation.

Contrastive Activation Addition

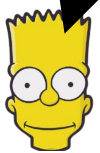
Rimsky et al 2312.06681

1. Getting steering vector

2. Steer!

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 + i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 + \frac{1}{2}(m_1 \psi_1 \psi_1 + m_2 \psi_2 \psi_2 + \text{h.c.})$



Fermions Vector

[Scalar fields only]

BART-L

$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 + i\psi_2^\dagger \bar{\sigma}_\mu D \psi_2 + i\psi_3^\dagger \bar{\sigma}_\mu \partial_\mu \psi_3 - \frac{1}{2}(m_1 \psi_1 \psi_1 + m_2 \psi_2 \psi_2 + m_3 \psi_3 \psi_3 + \text{h.c.})$

[**Fermion** only Lagrangian]

Inference Time Control
No model retraining/finetuning required

BART-L → symbolic output
Direct verification (easier than LLMs)

Subtract contrasting prompt activations to isolate a concept.

Inject the steering vector into model activations during generation.

Contrastive Activation Addition

Rimsky et al 2312.06681

1. Getting steering vector

2. Steer!

Input:
ψ_1, ψ_2

Output:
$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 + i\psi_2^\dagger \bar{\sigma}_\mu D^\mu \psi_2 + \frac{1}{2}(m_1\psi_1\psi_1 + m_2\psi_2\psi_2 + \text{h.c.})$

So **DOES IT WORK?**

Fermions
Vector

BART-L

[Scalar fields only]

$i\psi_1^\dagger \bar{\sigma}^\mu D_\mu \psi_1 + i\psi_2^\dagger \bar{\sigma}_\mu D\psi_2 + i\psi_3^\dagger \bar{\sigma}_\mu \partial_\mu \psi_3 - \frac{1}{2}(m_1\psi_1\psi_1 + m_2\psi_2\psi_2 + m_3\psi_3\psi_3 + \text{h.c.})$

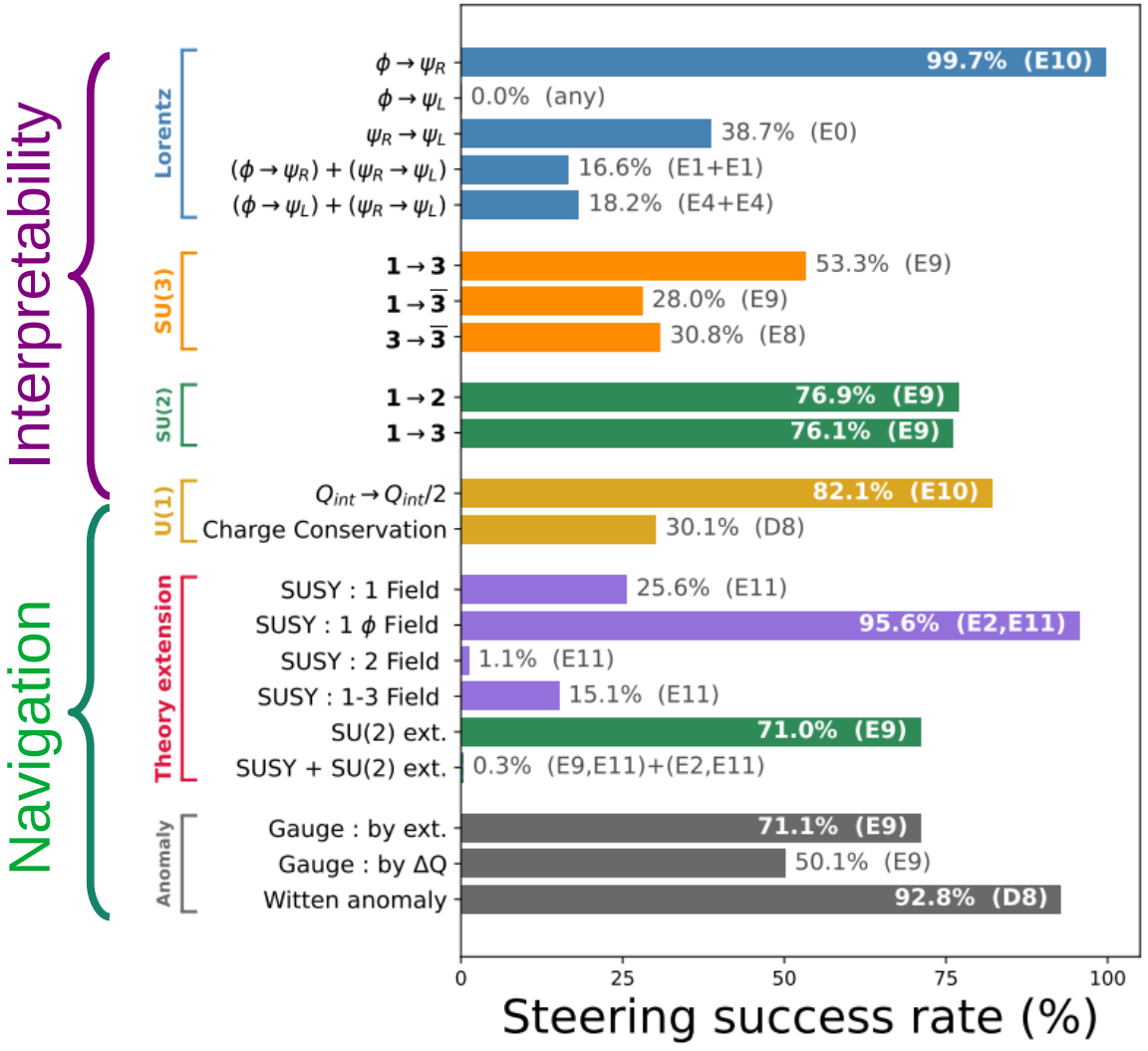
[**Fermion** only Lagrangian]

Subtract contrasting prompt activations
to isolate a concept.

Inject the steering vector
into model activations during generation.

MOSTLY YES

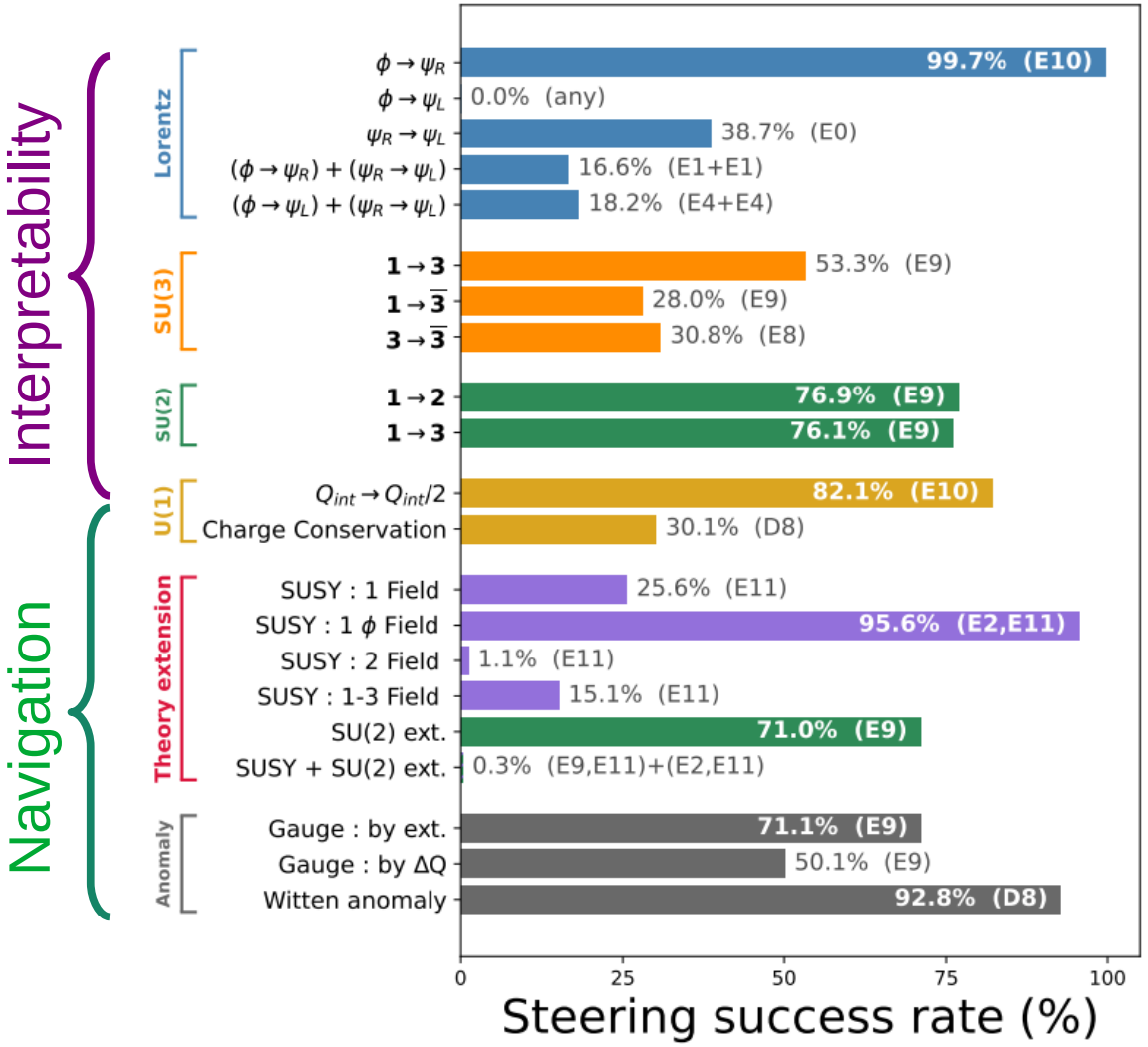
But when it cant,
theres always a hint of why!

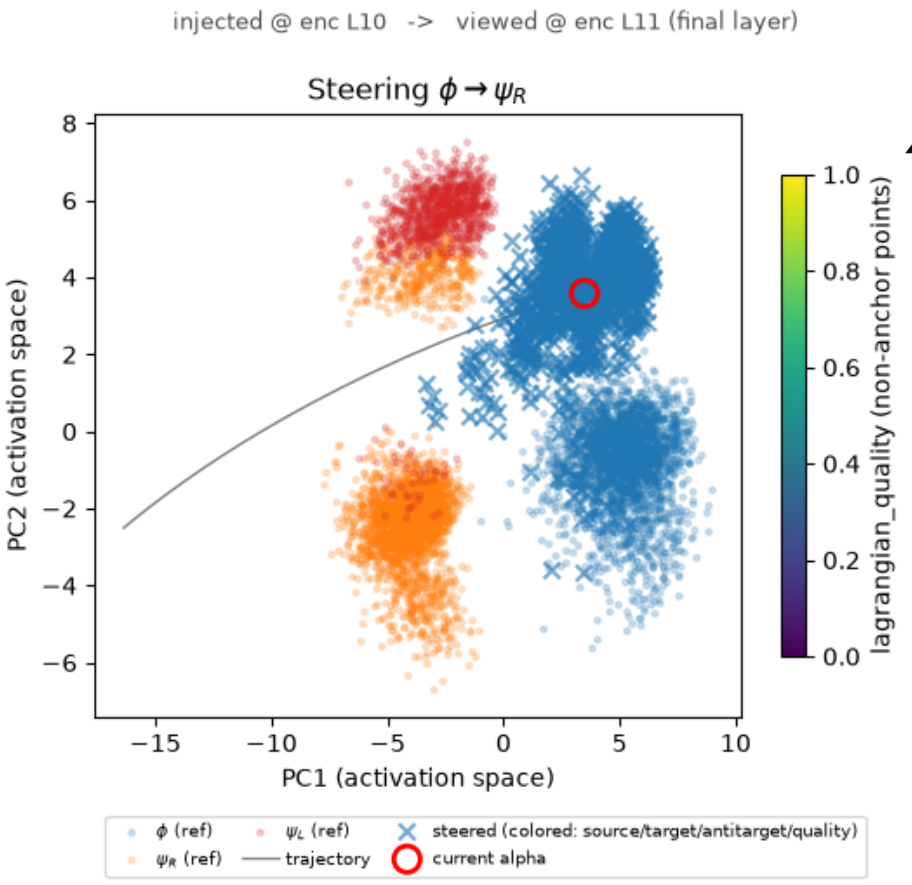


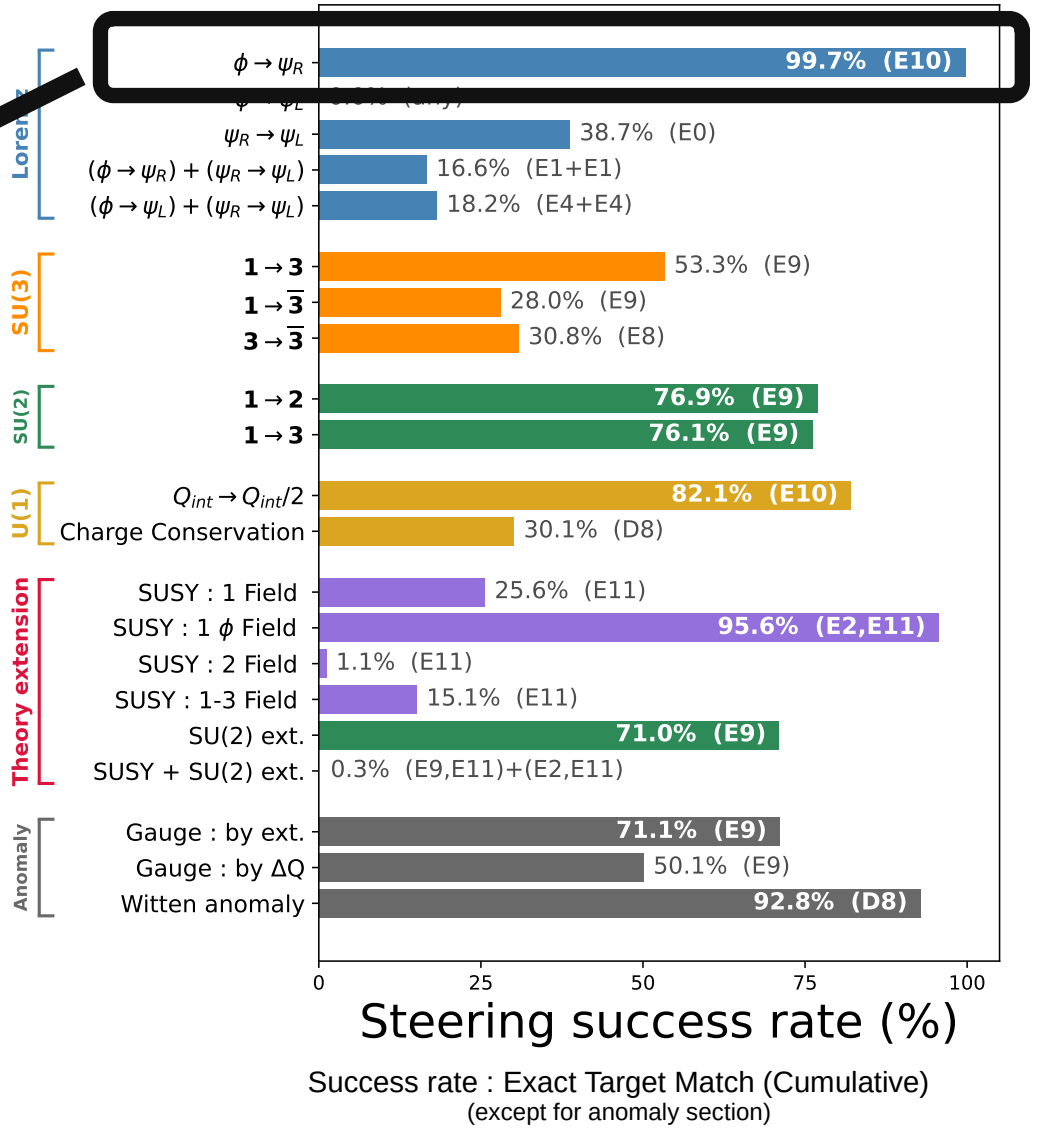
MOSTLY YES

But when it cant,
theres always a hint of why!

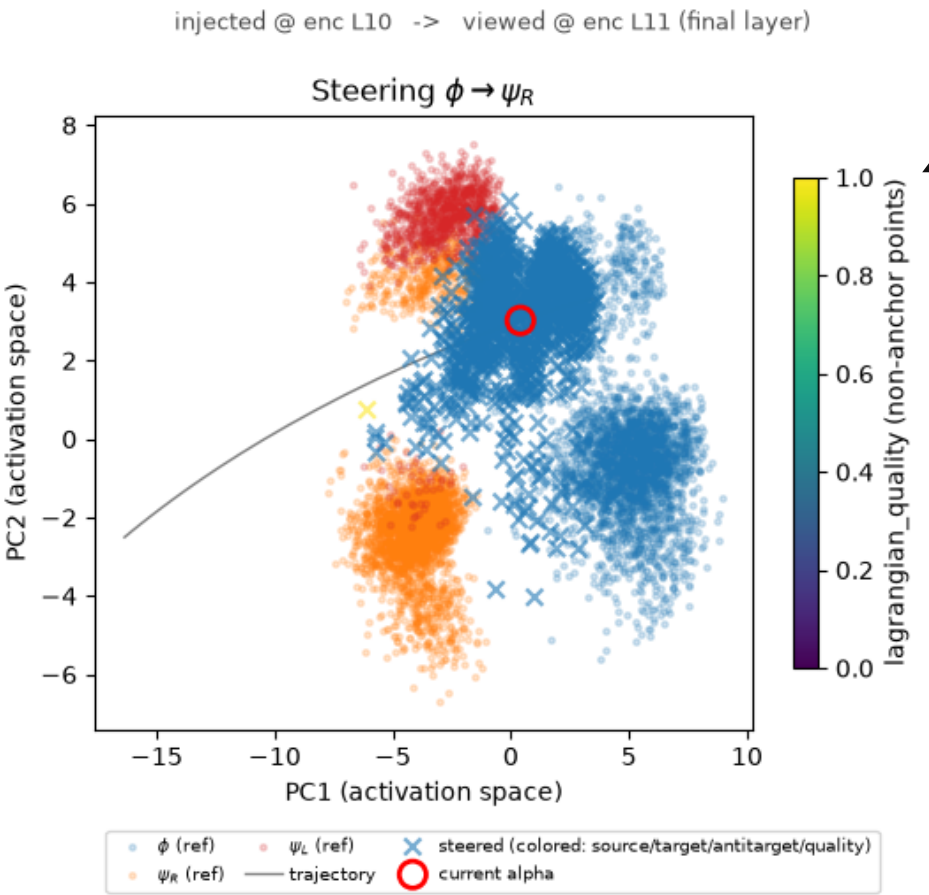
Success rate :
Exact Target Match
(Cumulative)
(except for anomaly section)



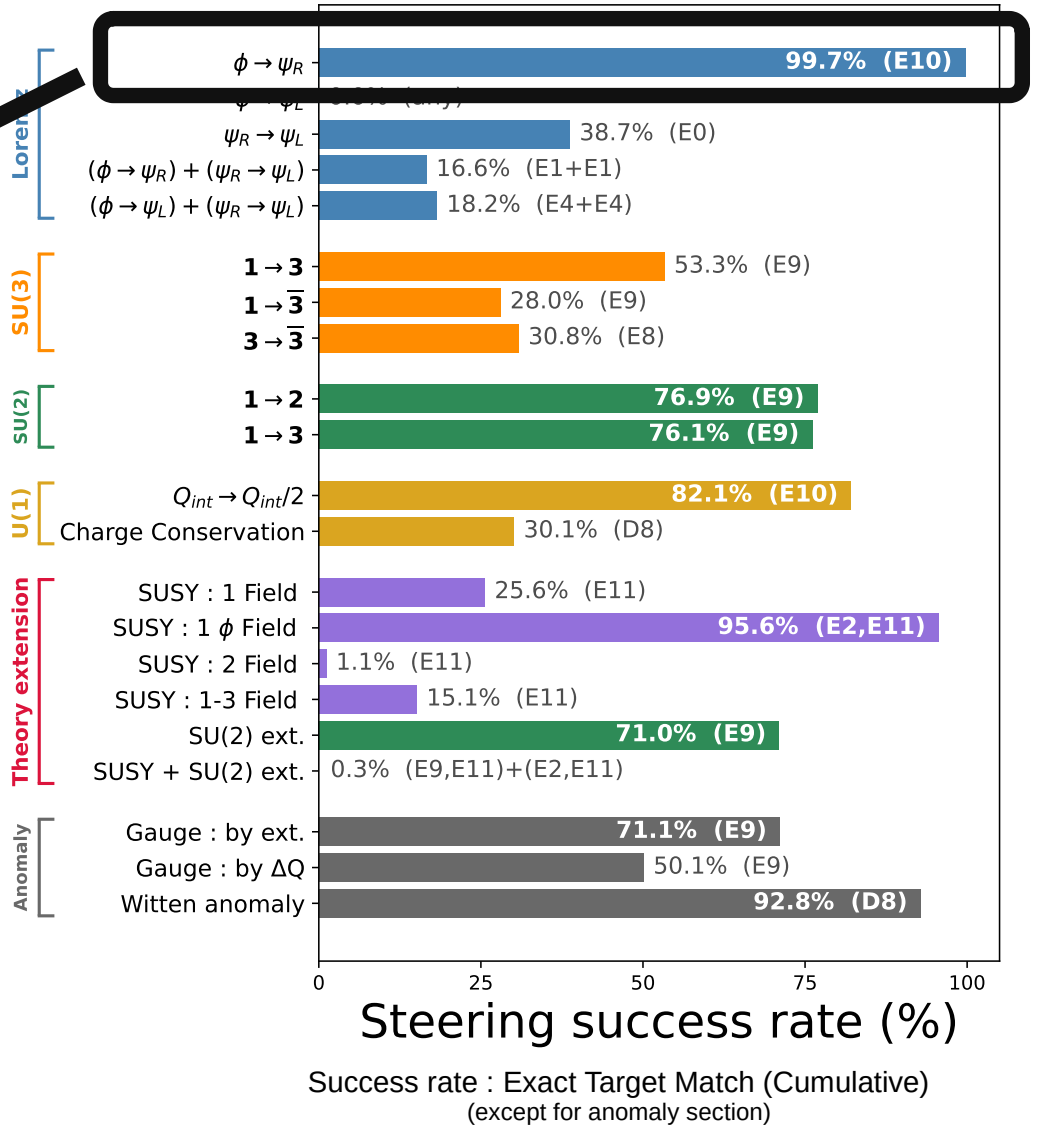




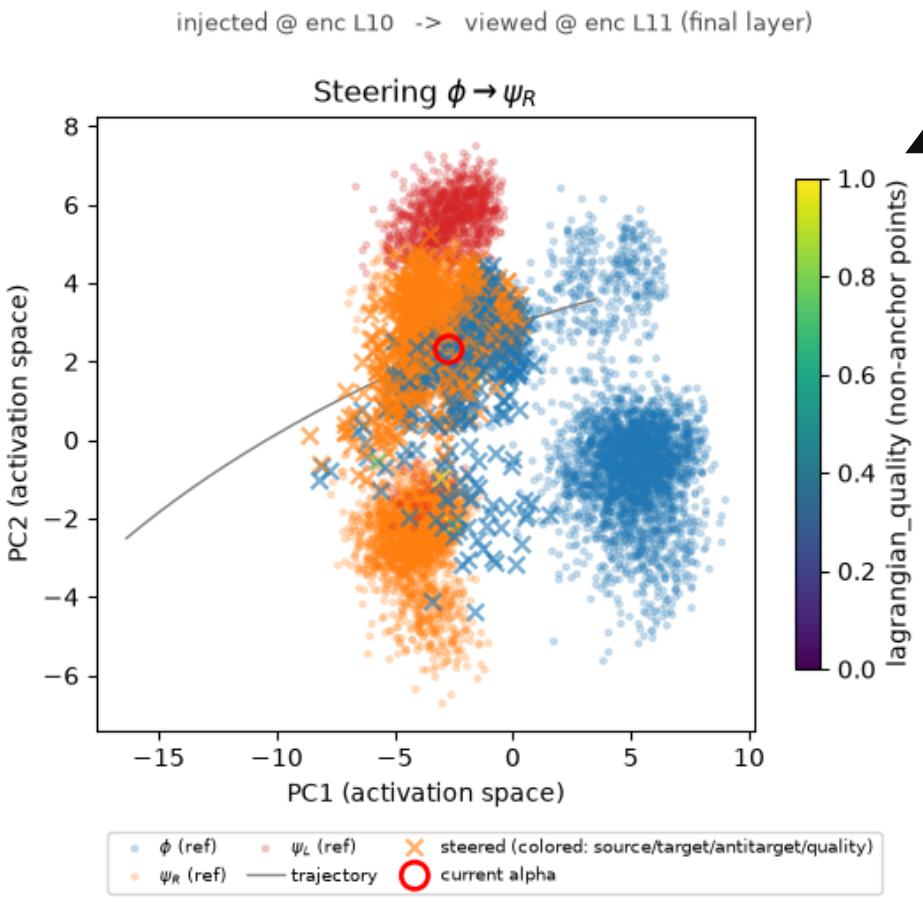
alpha = 0.00 | success = 0% | cumulative = 0%



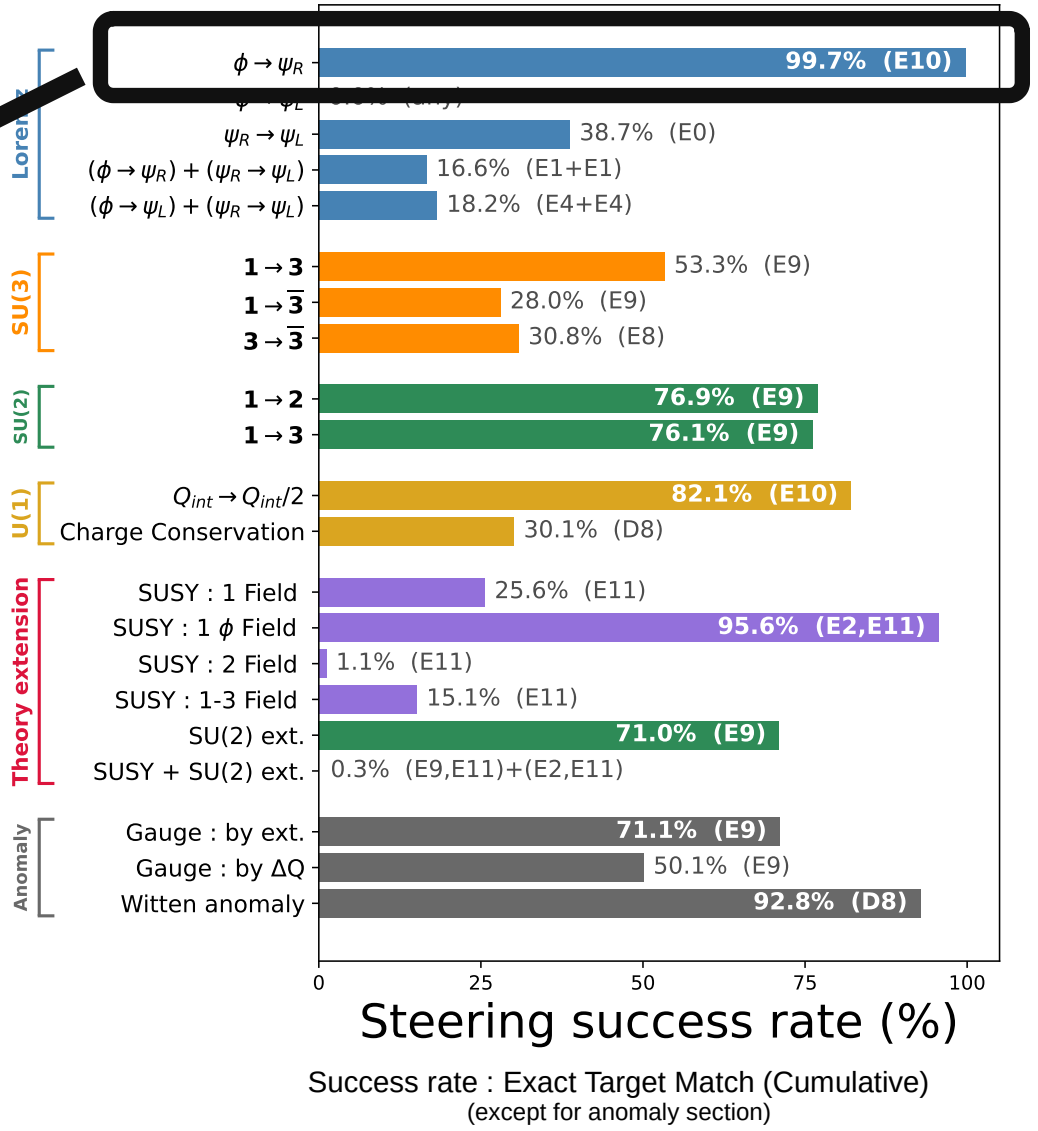
alpha = 0.20 **success = 0%** **cumulative = 0%**



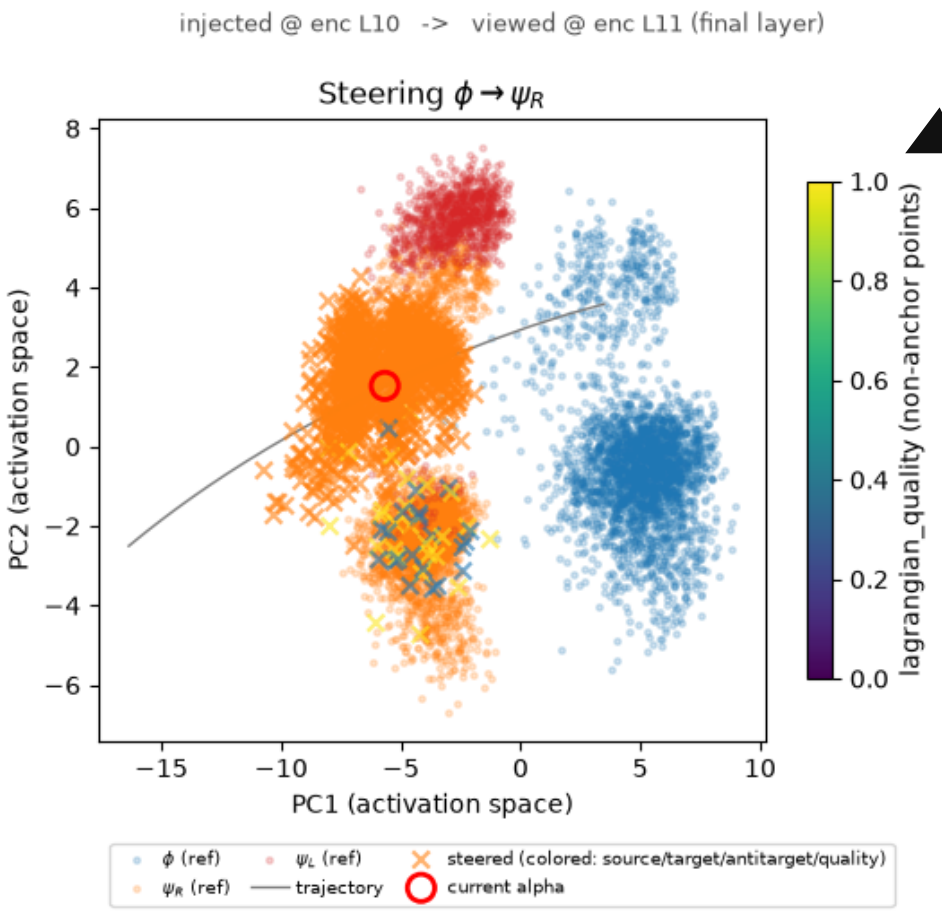
Success rate : Exact Target Match (Cumulative)
(except for anomaly section)

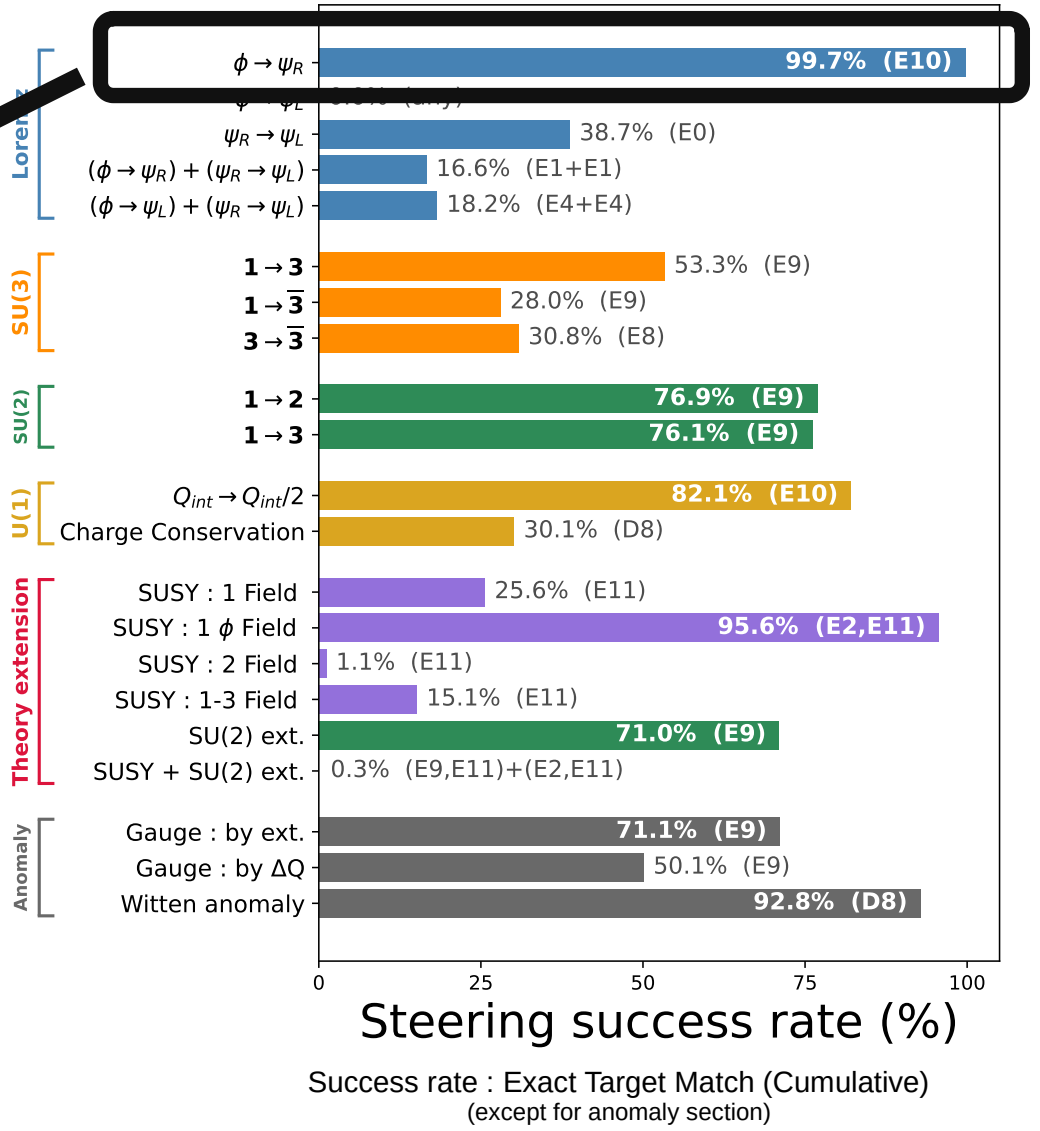


alpha = 0.40 **success = 63%** **cumulative = 63%**

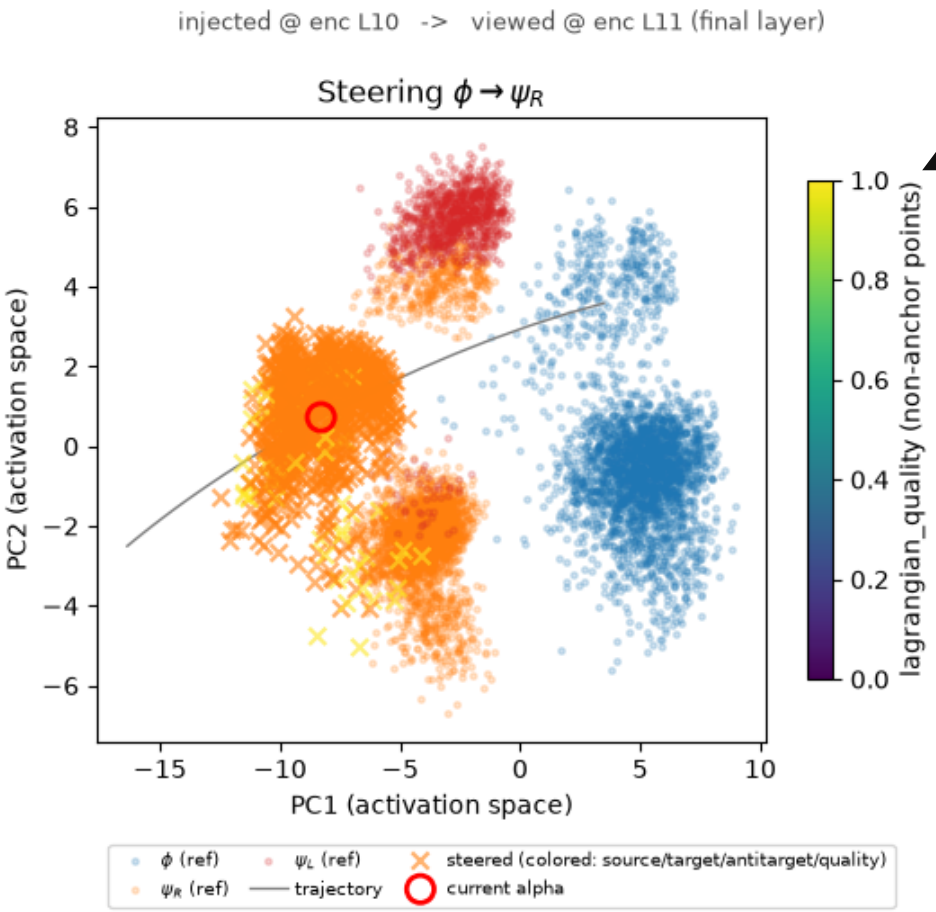


Success rate : Exact Target Match (Cumulative)
(except for anomaly section)

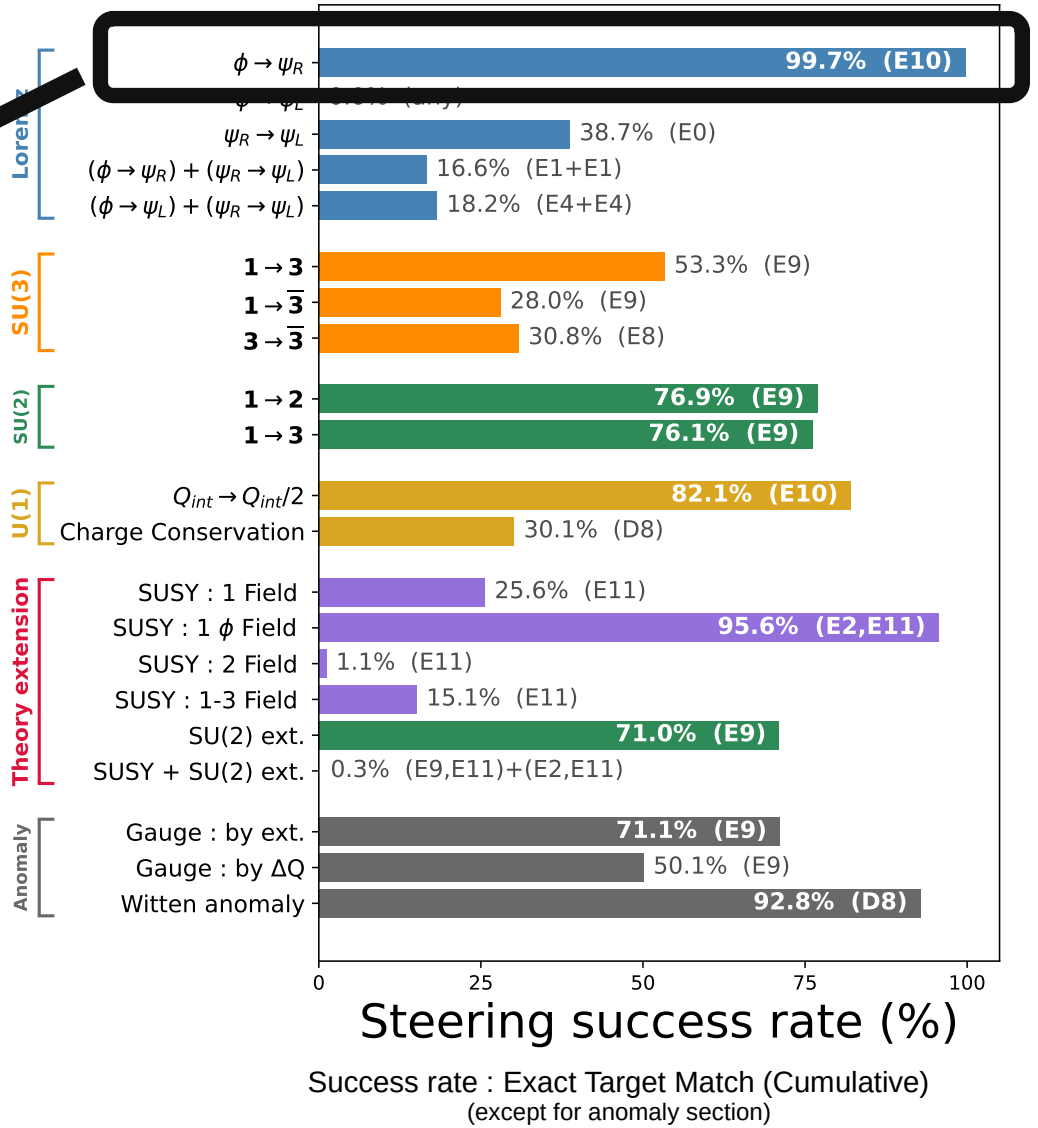




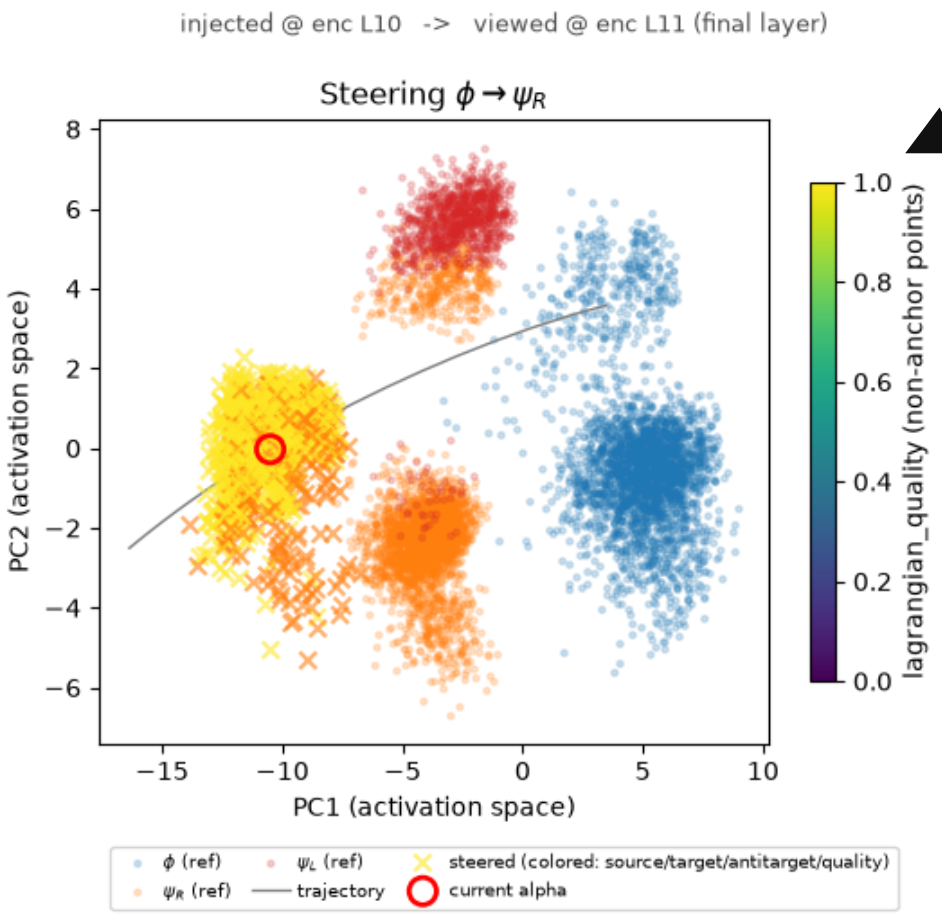
alpha = 0.60 **success = 95%** **cumulative = 95%**

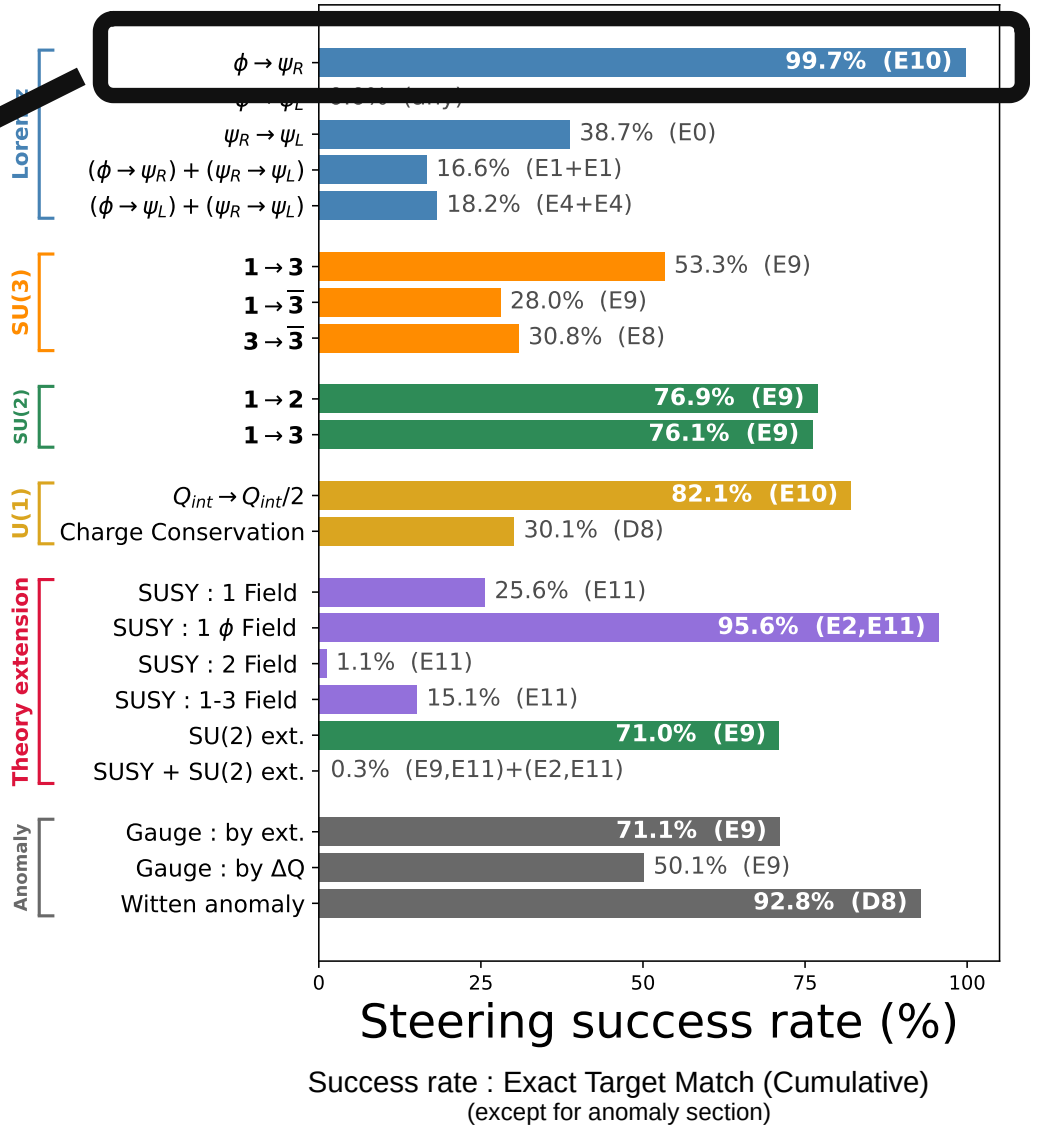


alpha = 0.80 **success = 94%** **cumulative = 98%**



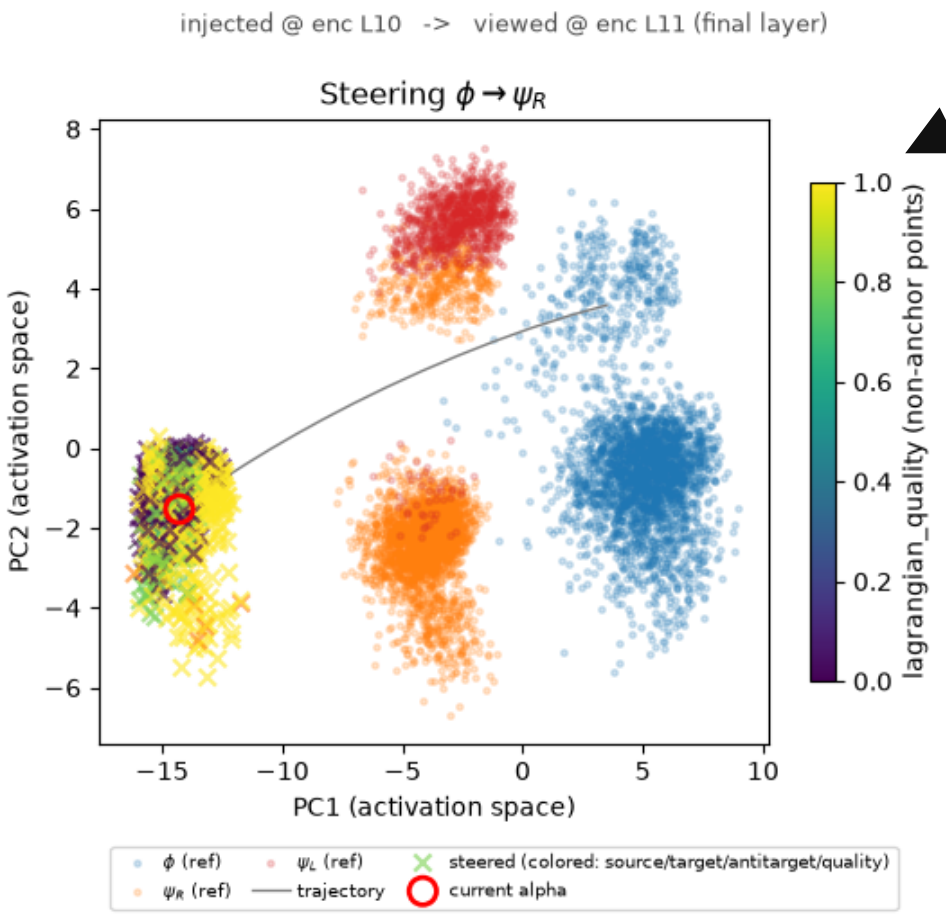
Success rate : Exact Target Match (Cumulative)
(except for anomaly section)

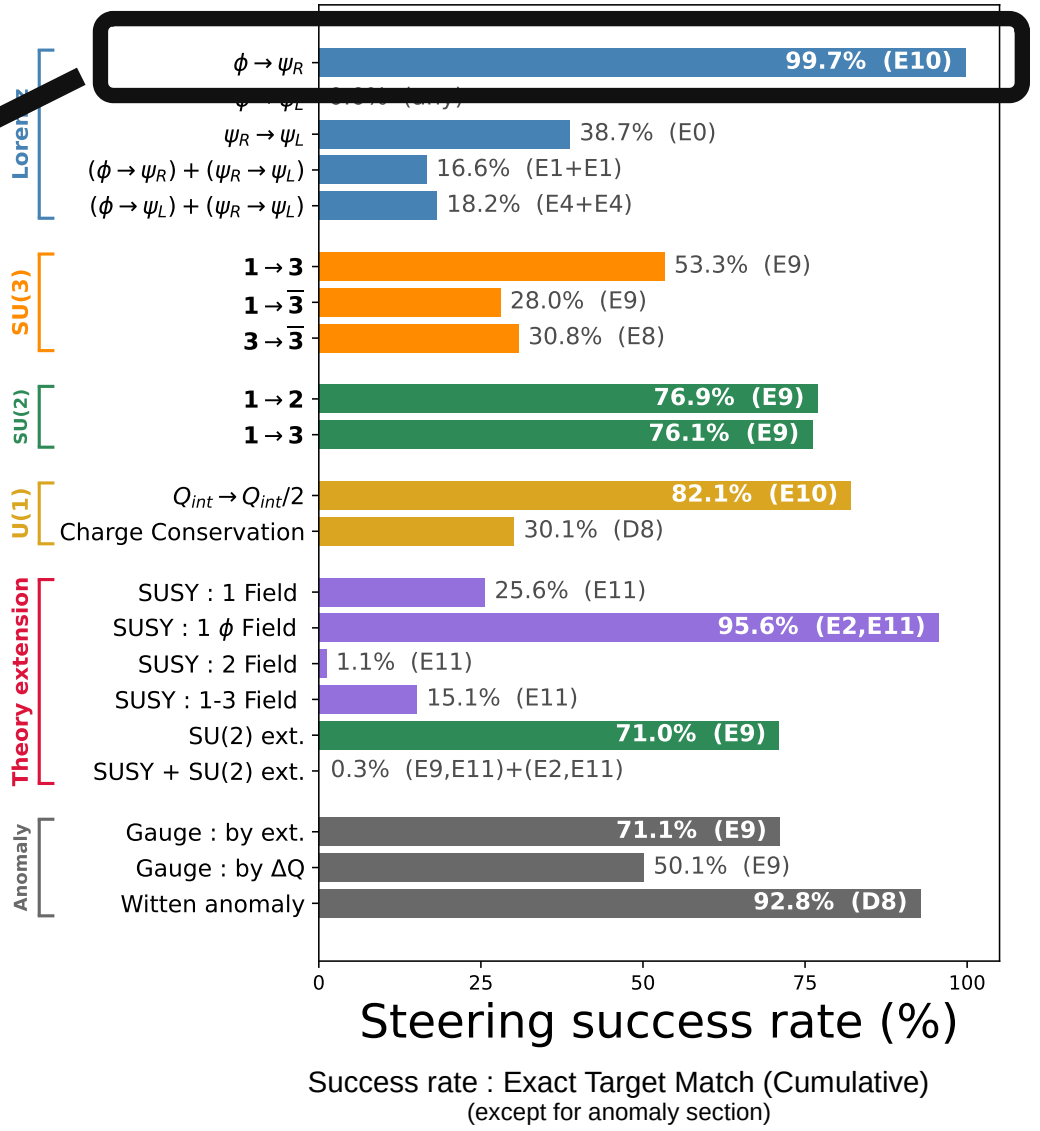




Success rate : Exact Target Match (Cumulative)
(except for anomaly section)

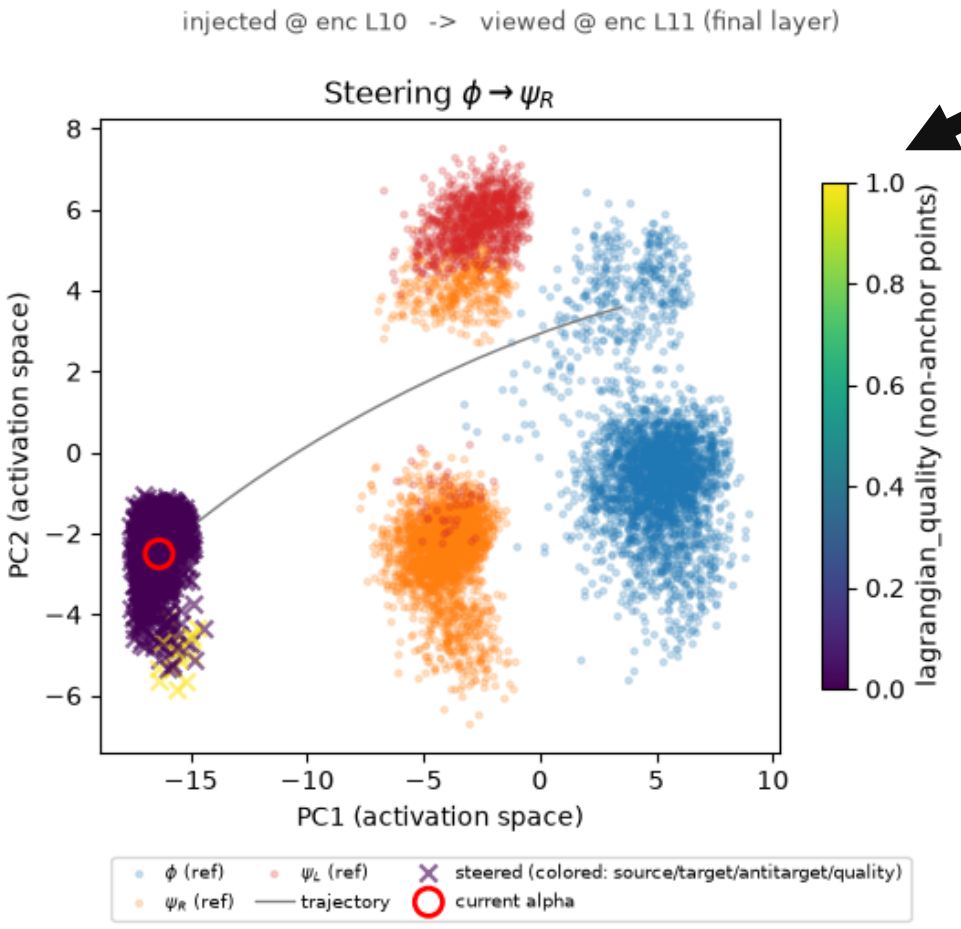
alpha = 1.00 **success = 25%** **cumulative = 100%**

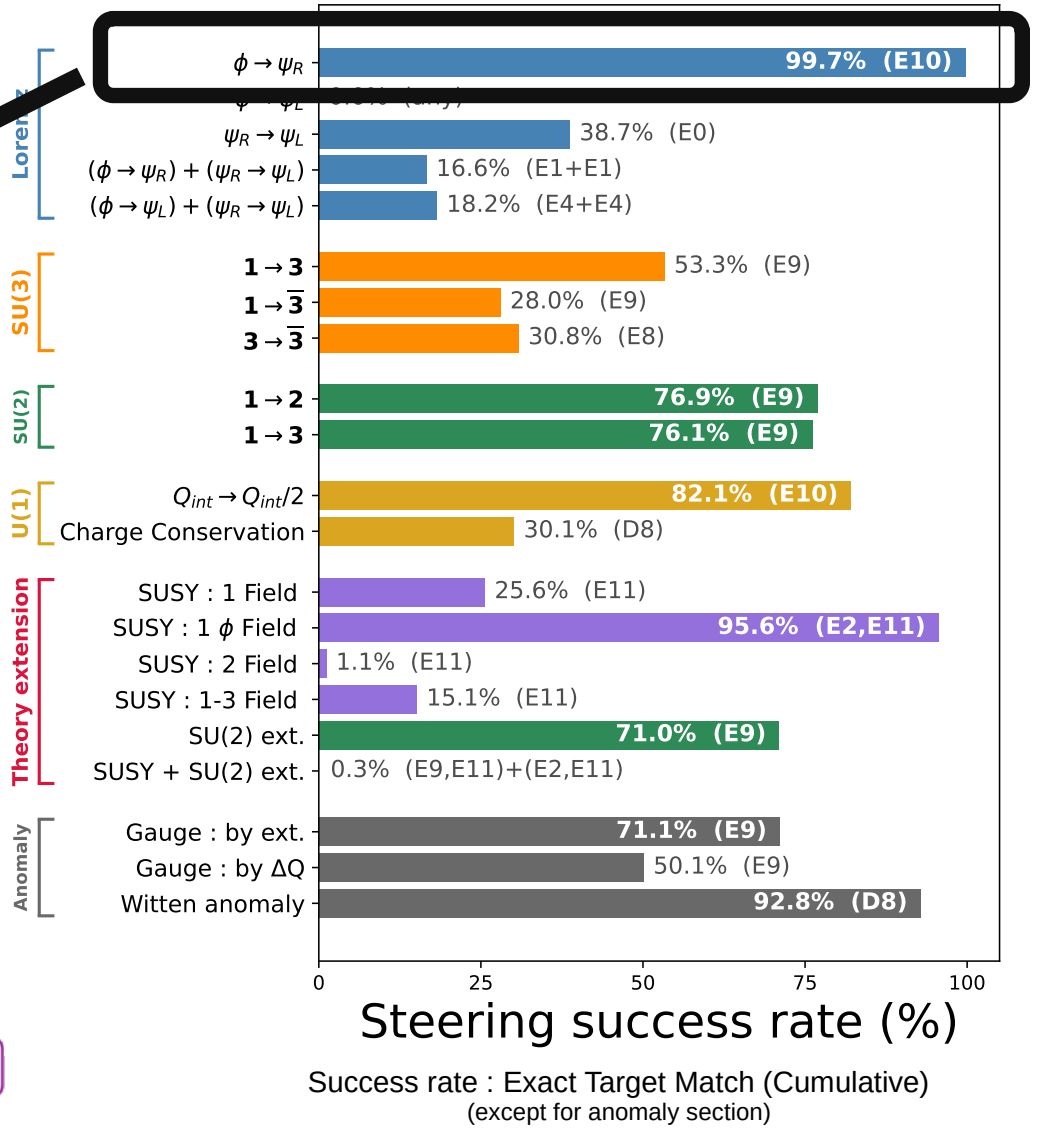




Success rate : Exact Target Match (Cumulative)
(except for anomaly section)

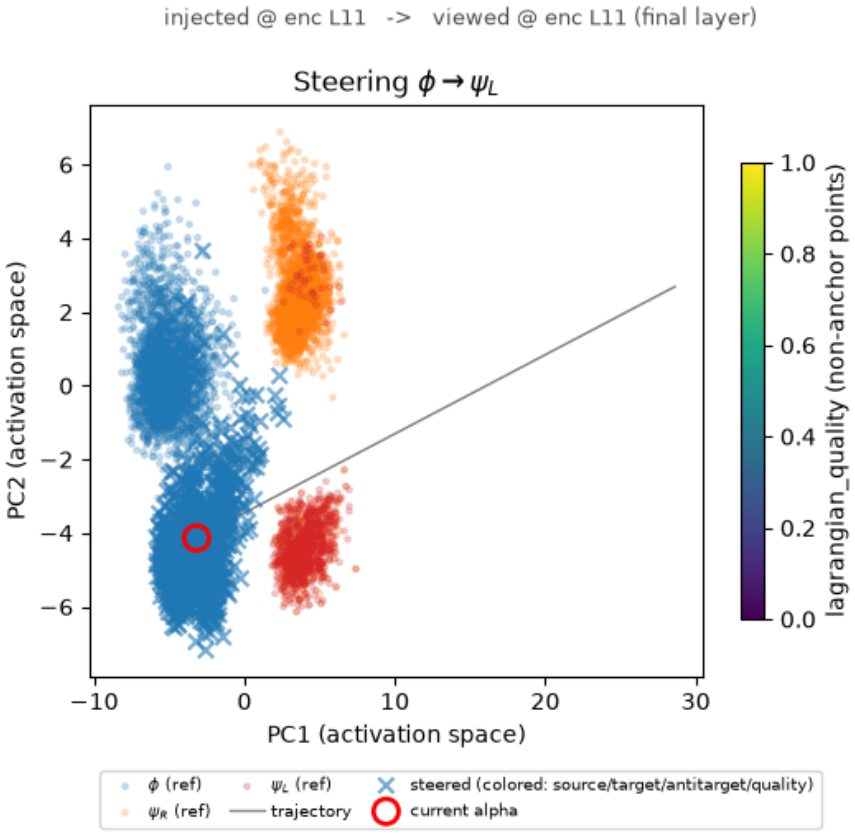
alpha = 1.50 **success = 0%** **cumulative = 100%**

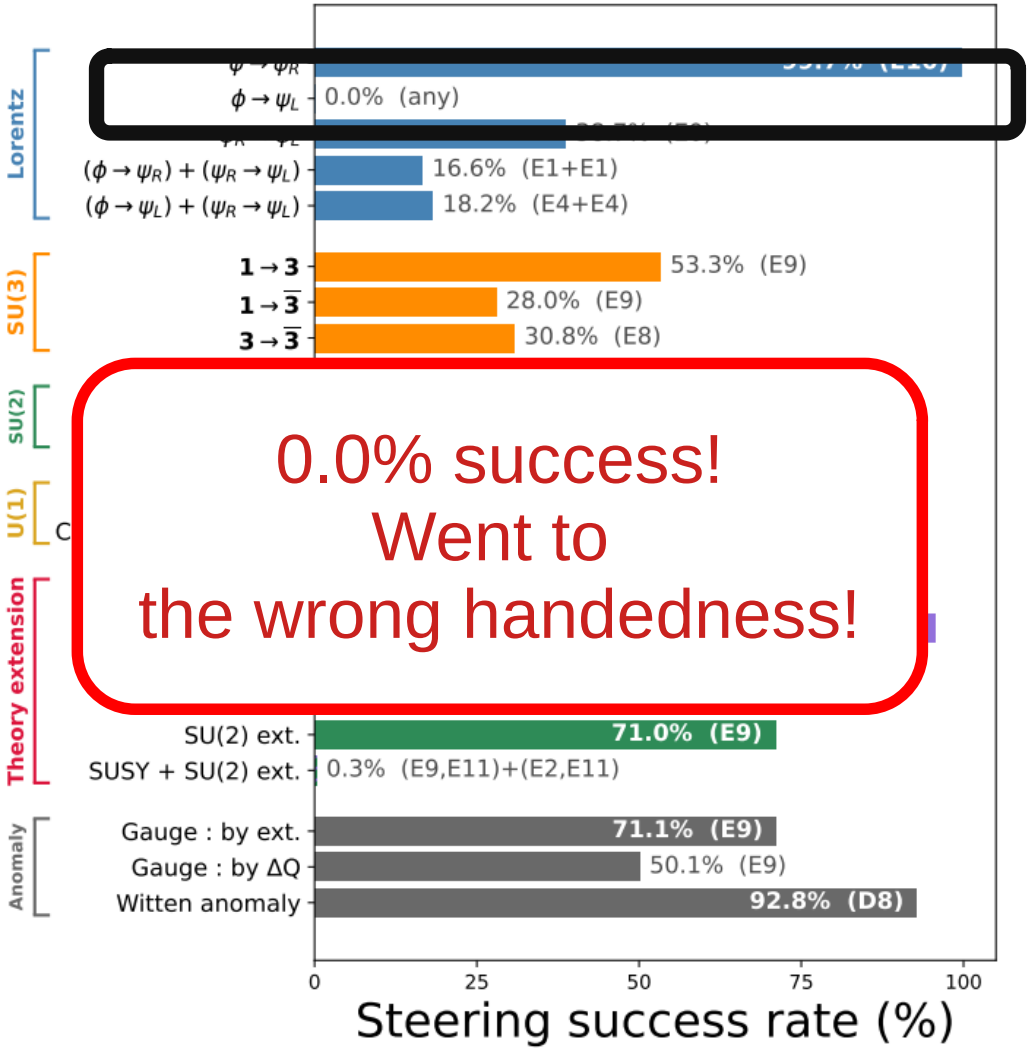




Success rate : Exact Target Match (Cumulative)
(except for anomaly section)

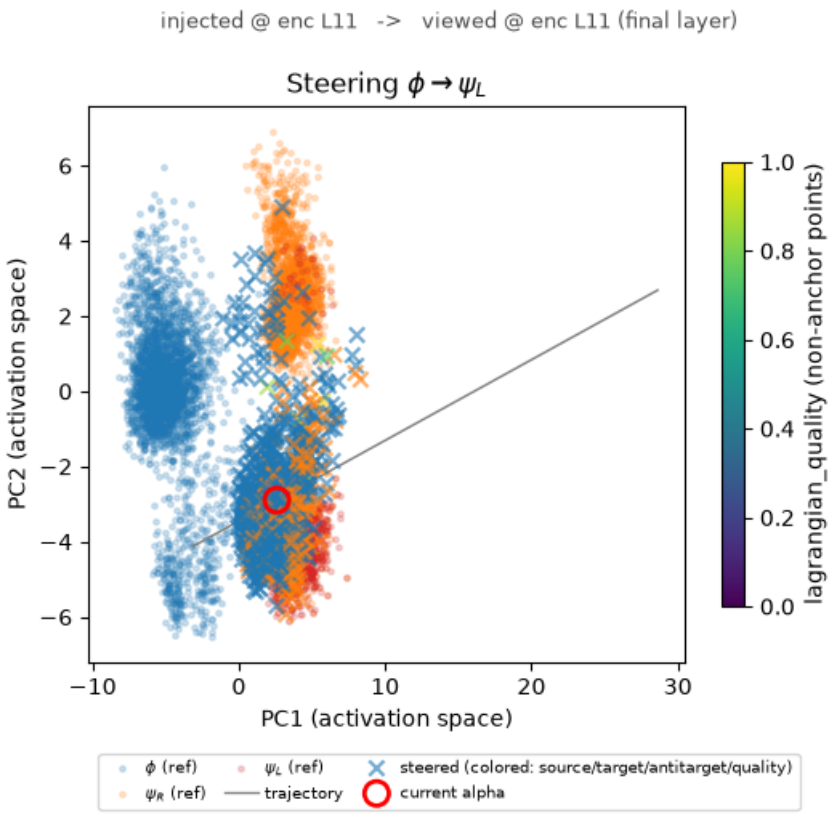
alpha = 2.00 **success = 0%** **cumulative = 100%**



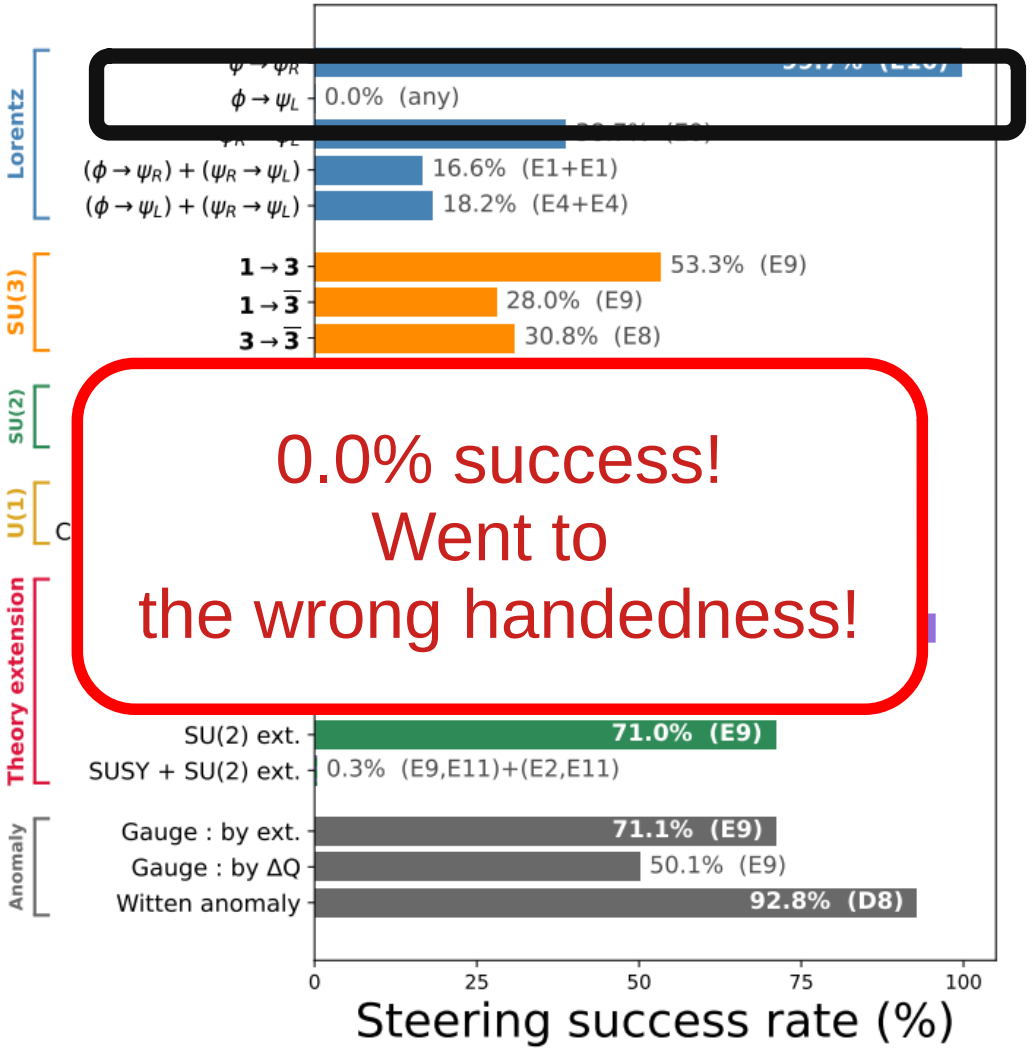


0.0% success!
Went to
the wrong handedness!

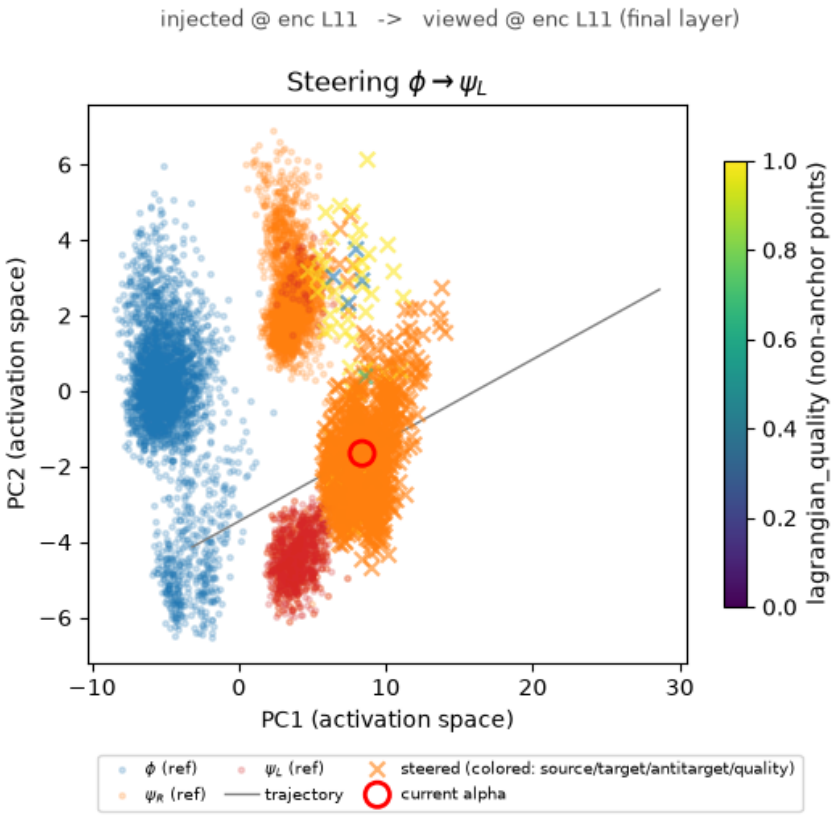
alpha = 0.00 **success = 0%** **cumulative = 0%**



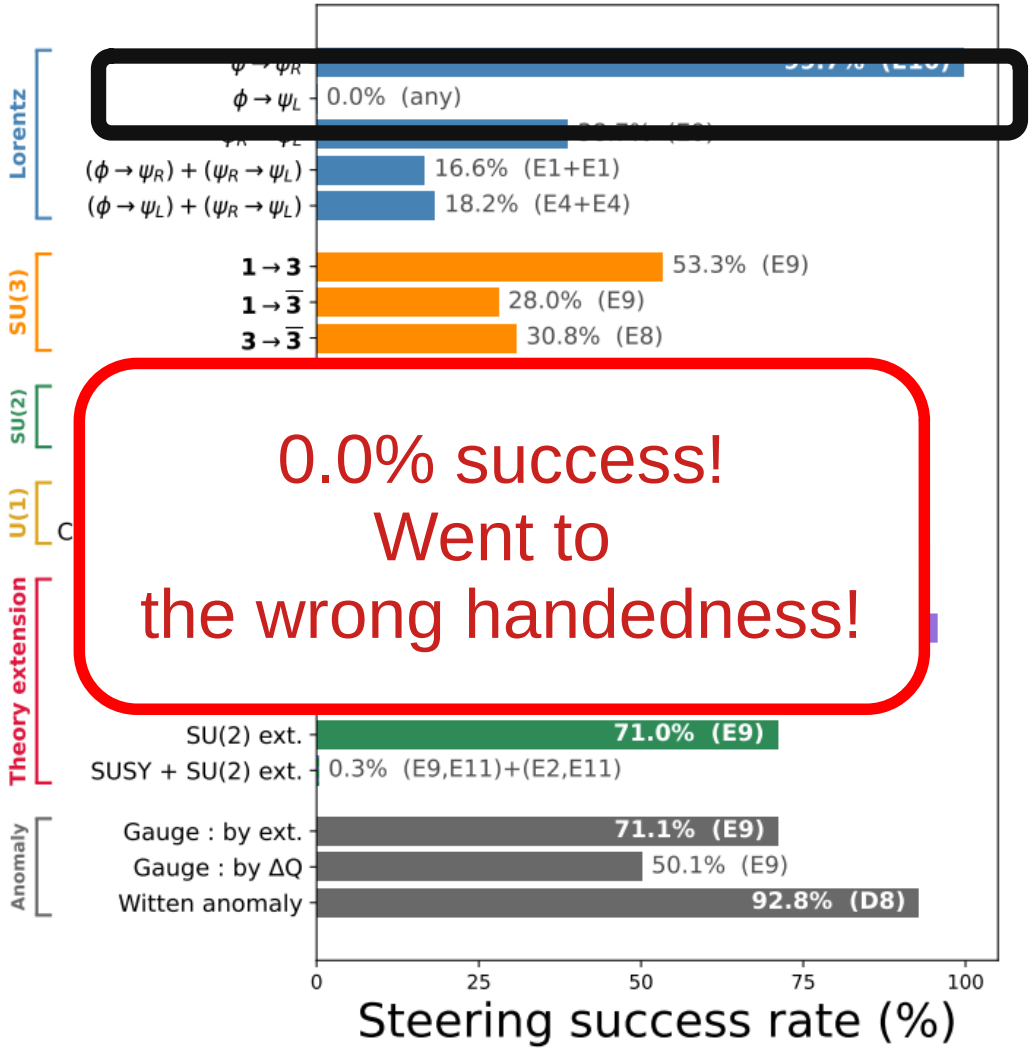
alpha = 0.20 **success = 0%** **cumulative = 0%**

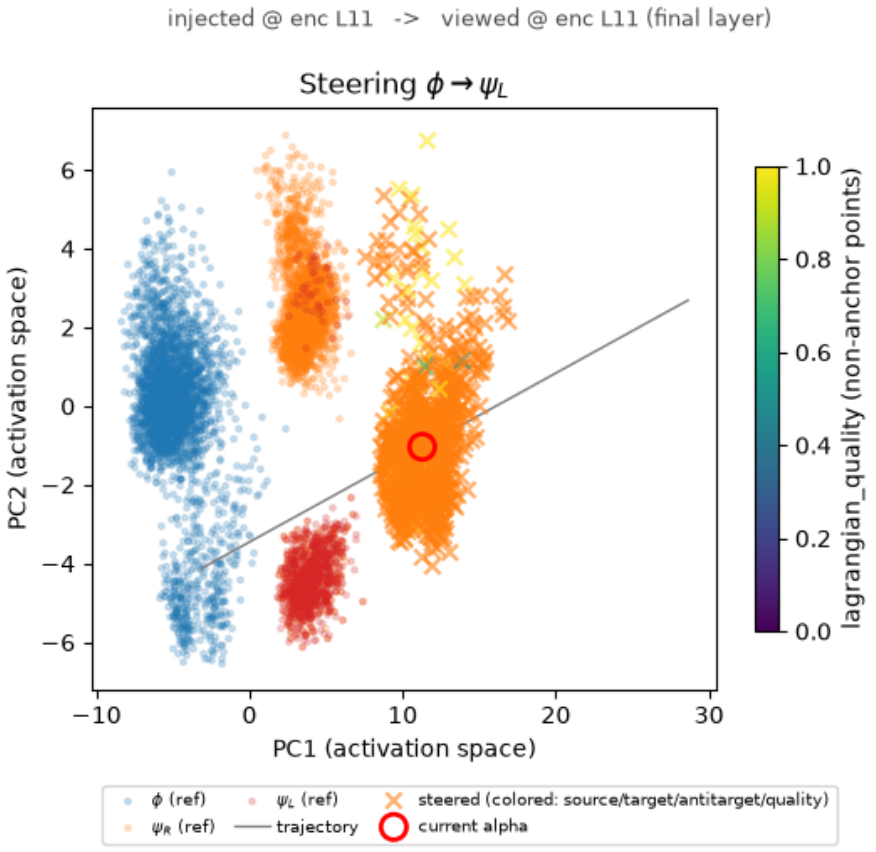


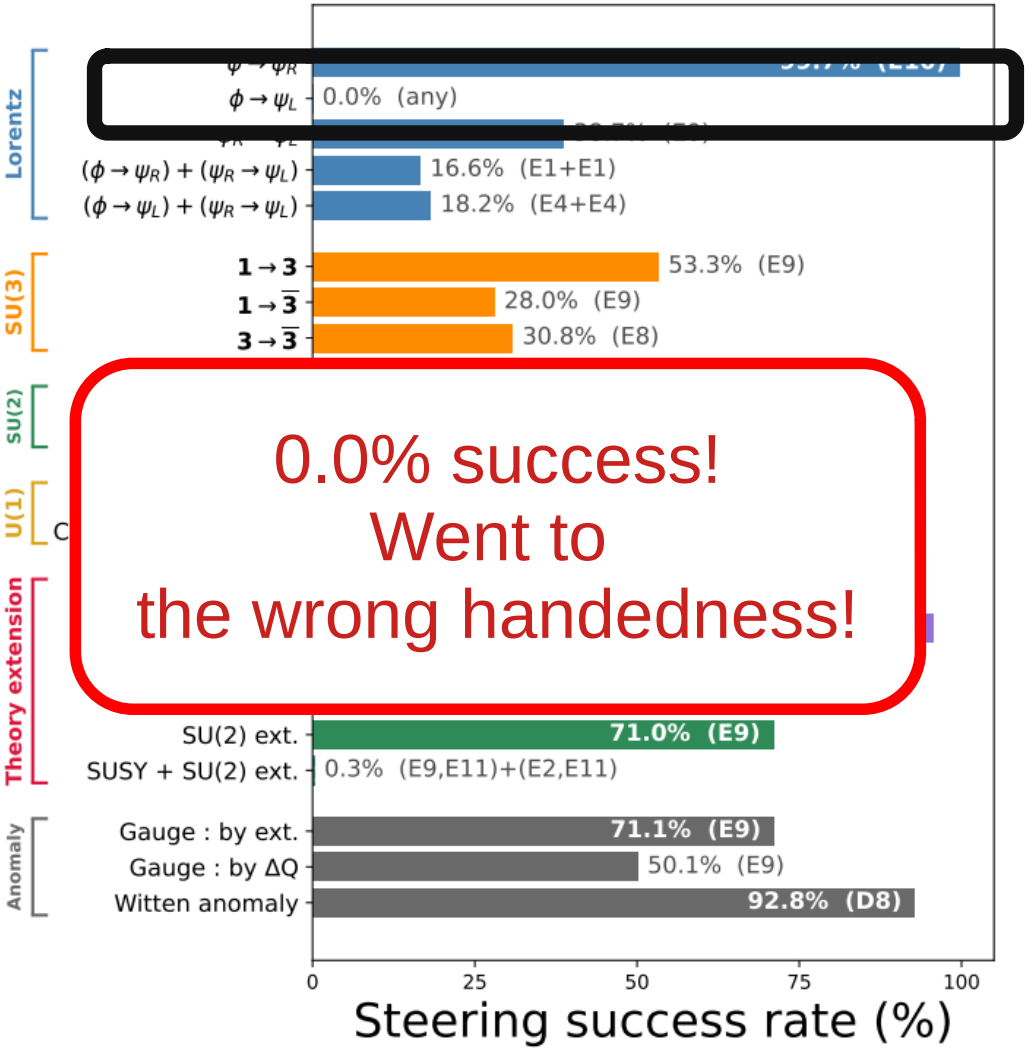
0.0% success!
Went to
the wrong handedness!



alpha = 0.40 **success = 0%** **cumulative = 0%**

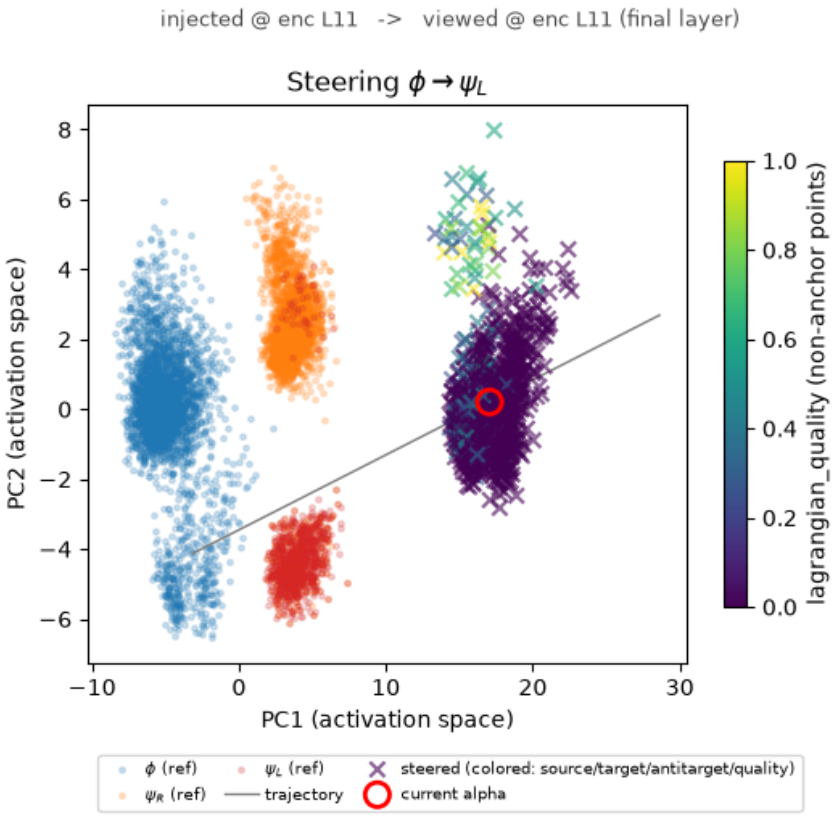




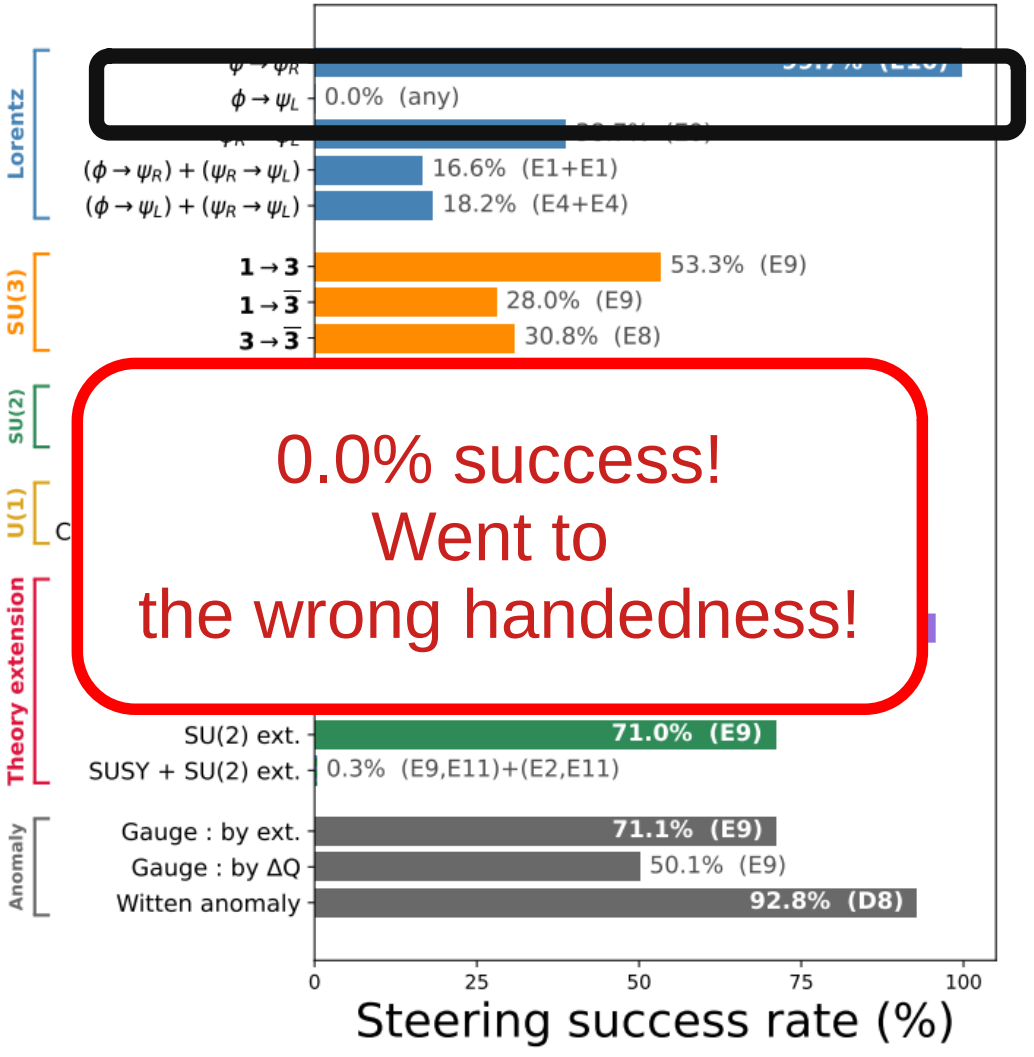


0.0% success!
Went to
the wrong handedness!

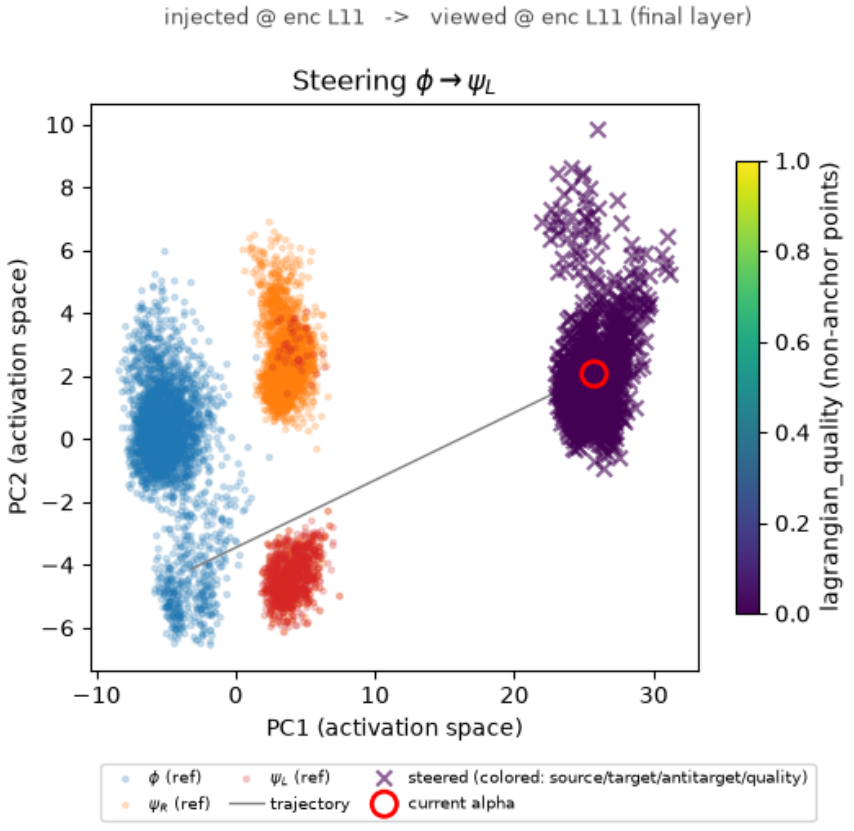
alpha = 0.50 **success = 0%** **cumulative = 0%**



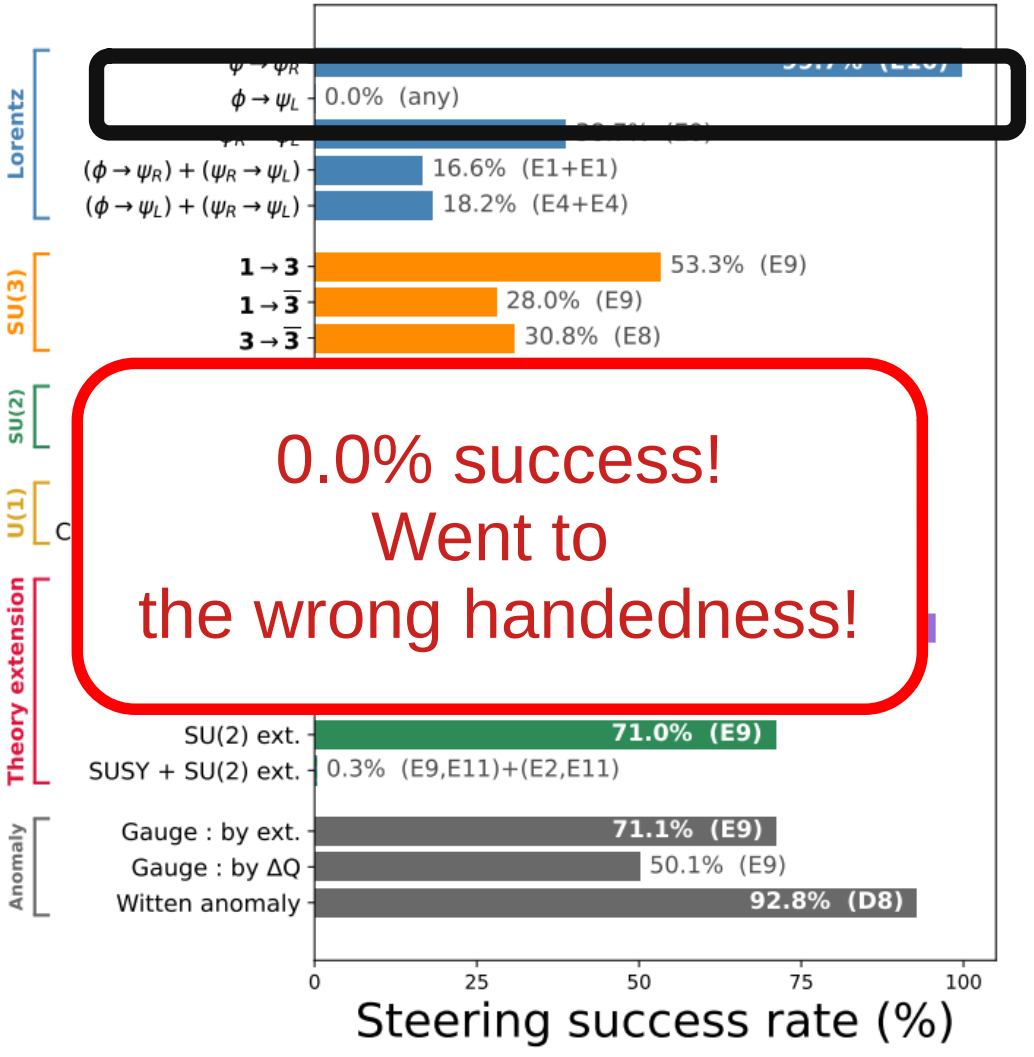
alpha = 0.70 **success = 0%** **cumulative = 0%**



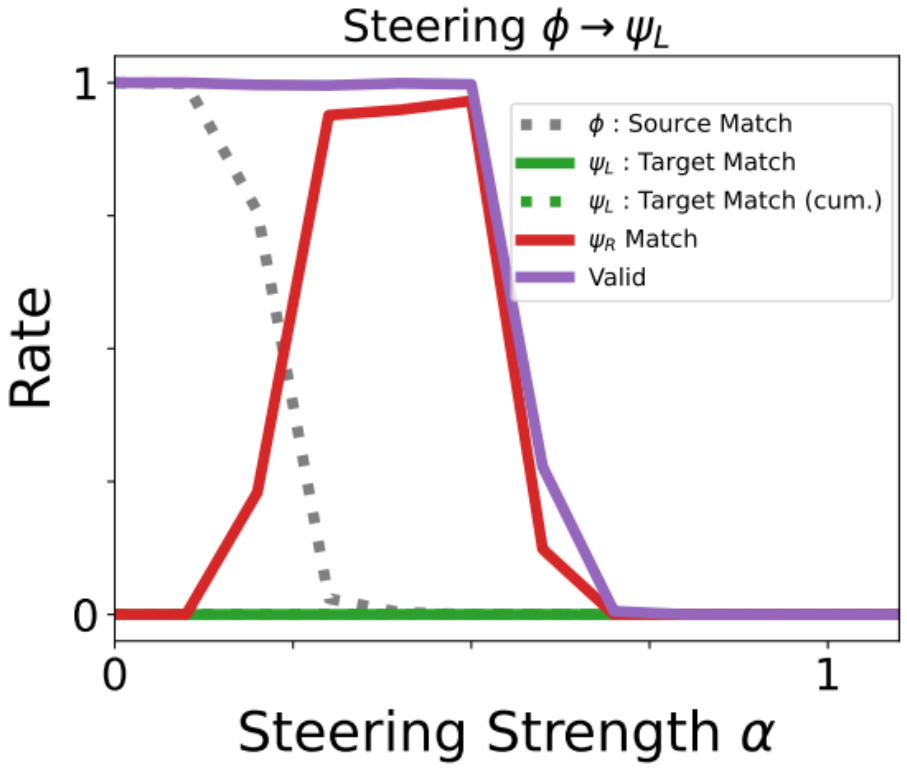
0.0% success!
Went to
the wrong handedness!



alpha = 1.00 **success = 0%** **cumulative = 0%**

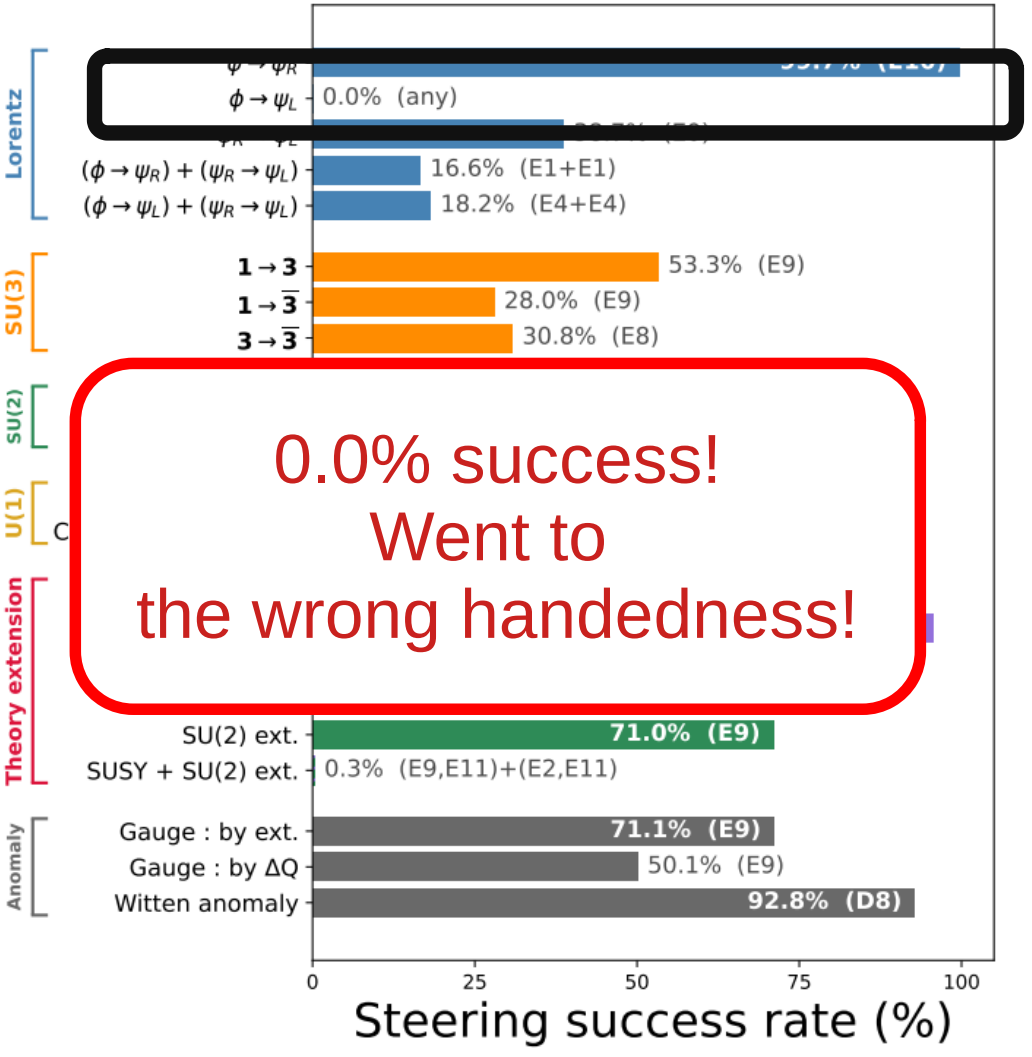


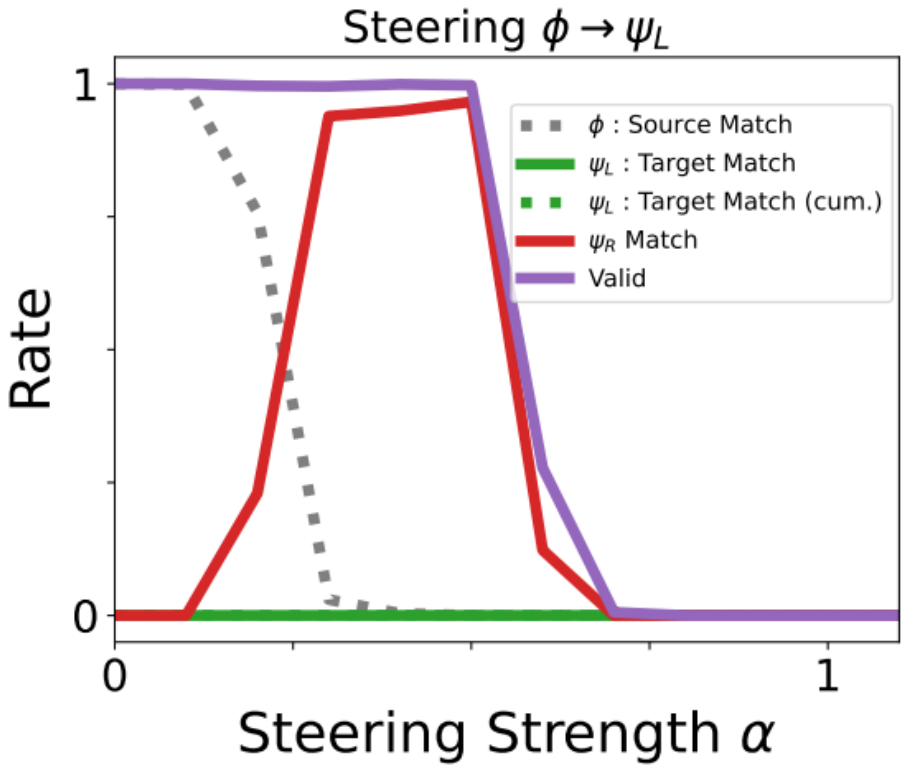
0.0% success!
Went to
the wrong handedness!



Behavior

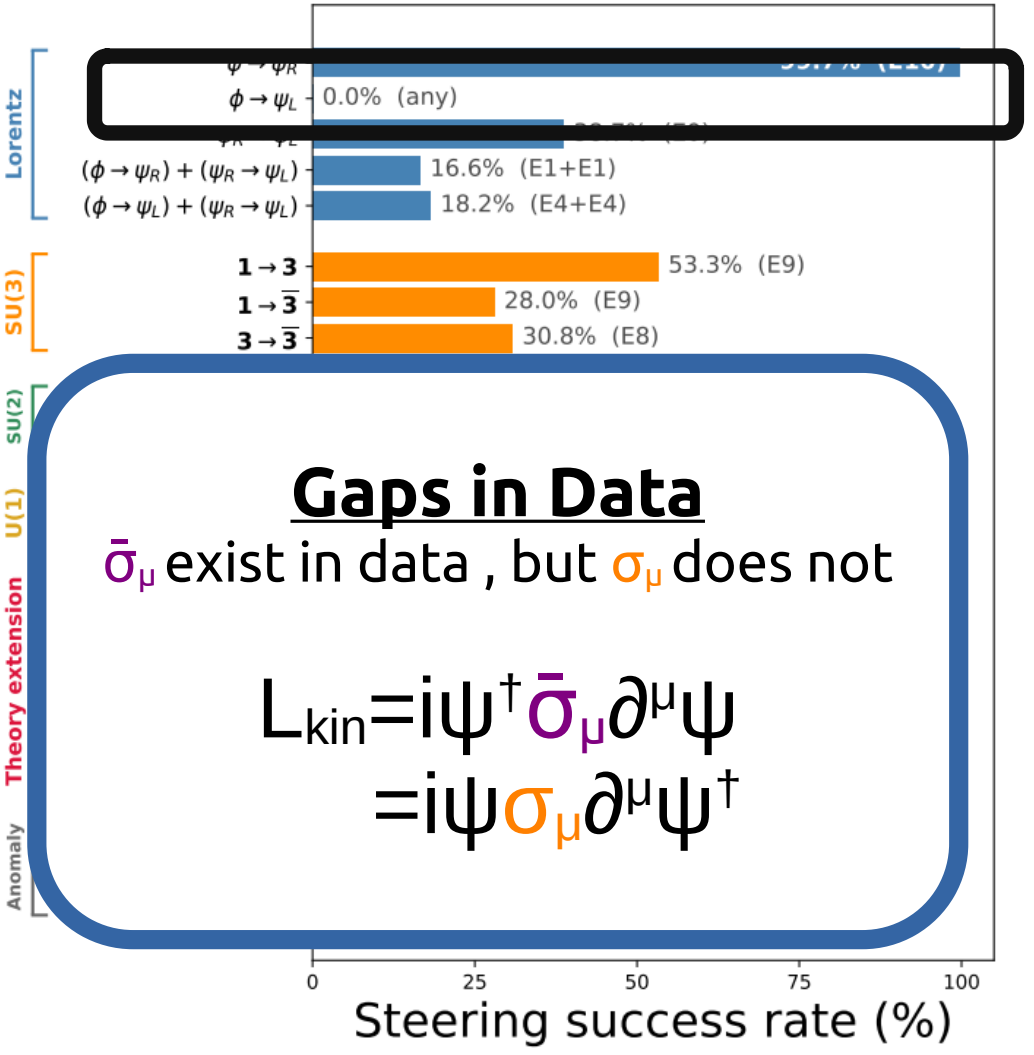
It went to Ψ_R, instead of Ψ_L!





Behavior

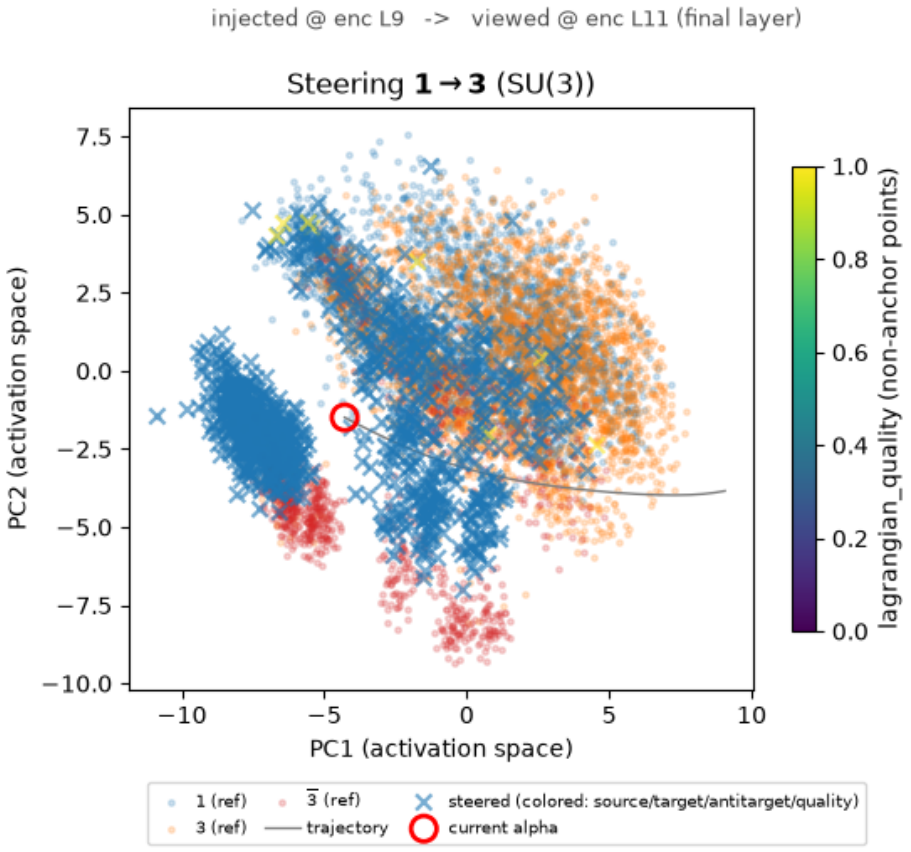
It went to Ψ_R, instead of Ψ_L!



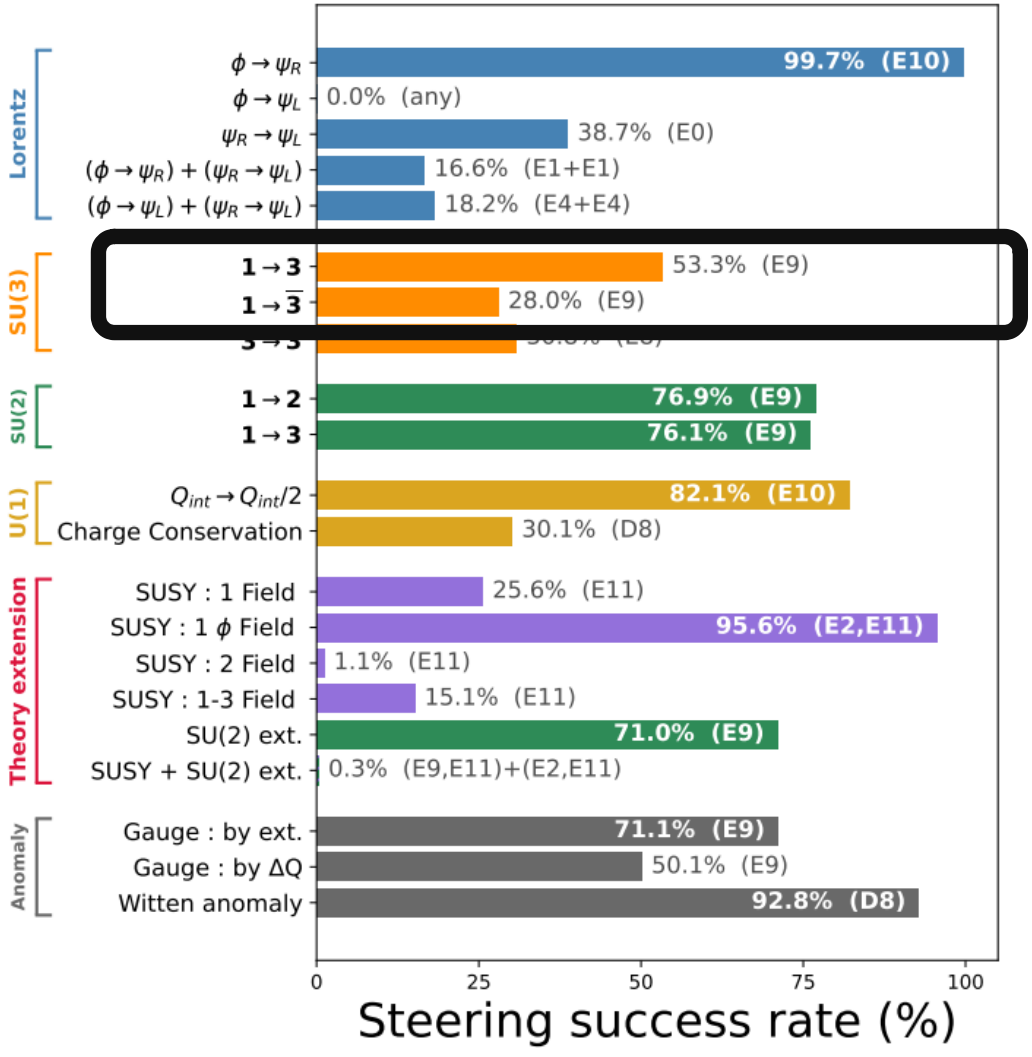
Gaps in Data

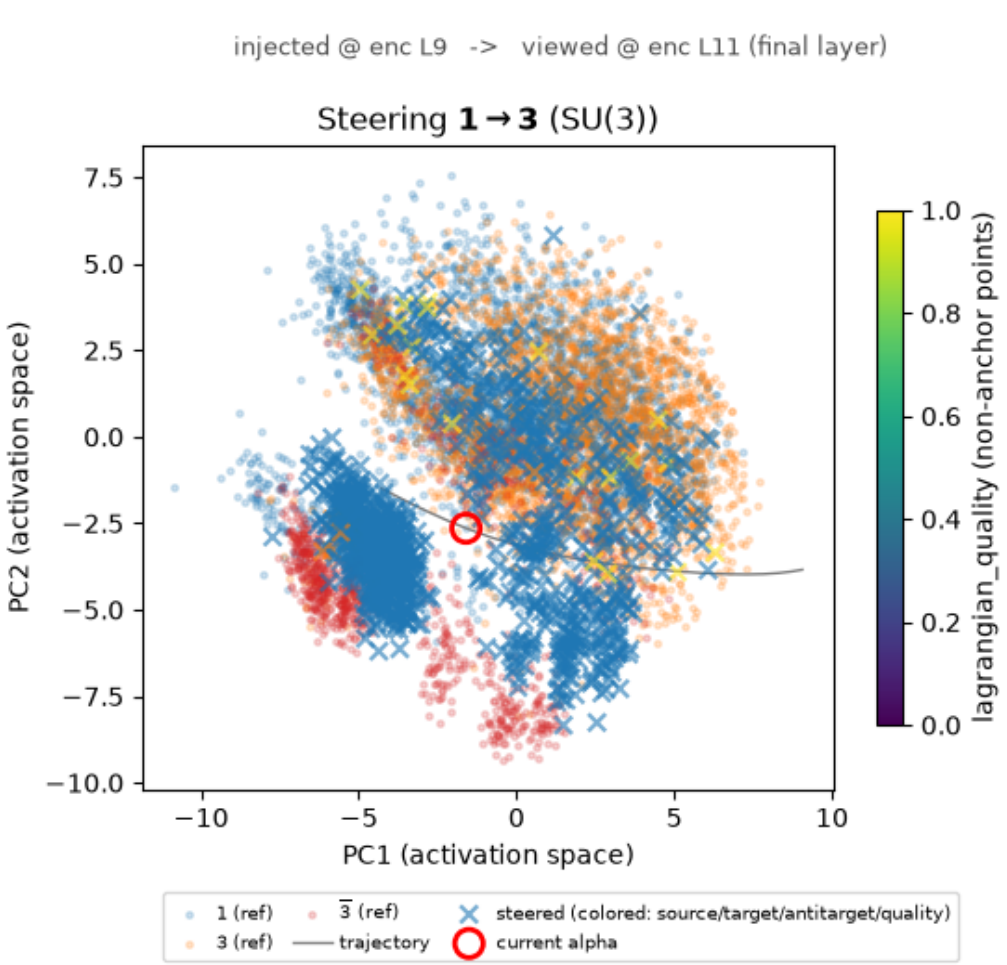
$\bar{\sigma}_\mu$ exist in data , but σ_μ does not

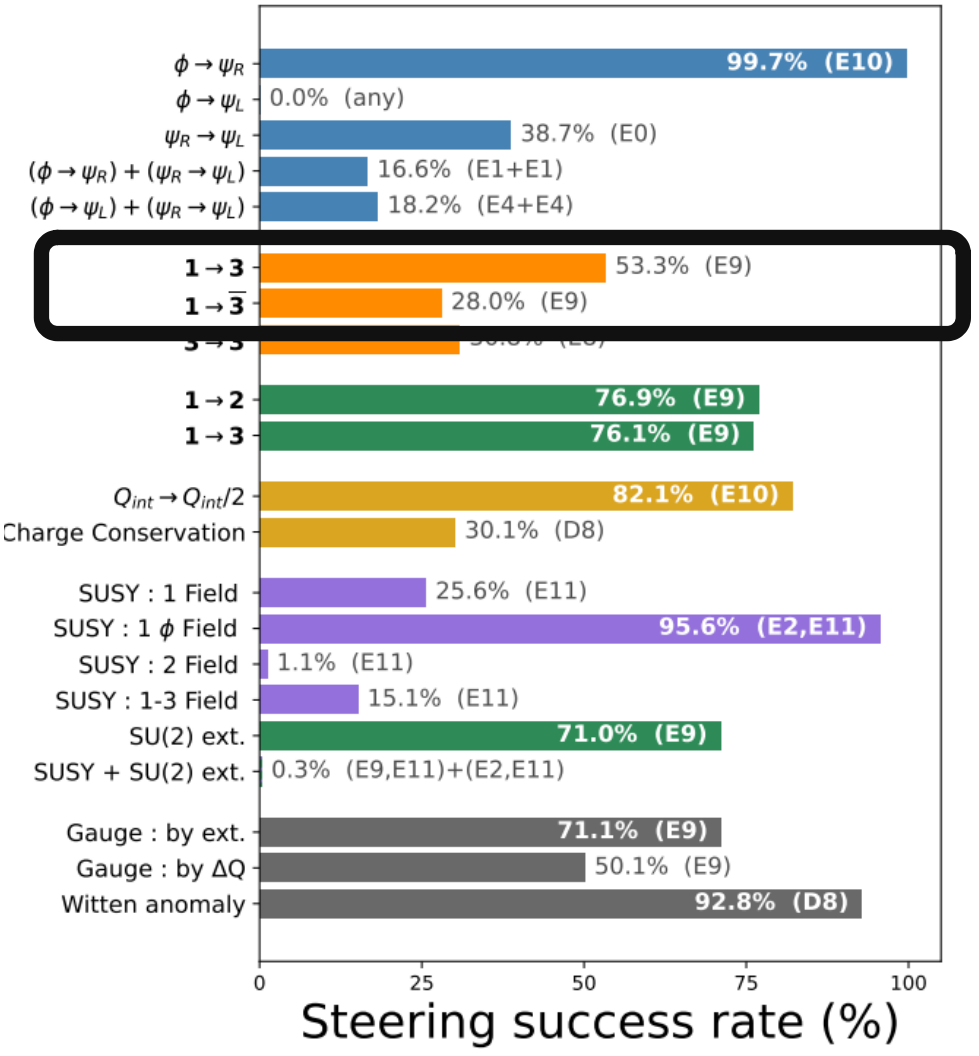
$$L_{kin} = i\psi^\dagger \bar{\sigma}_\mu \partial^\mu \psi$$
$$= i\psi \sigma_\mu \partial^\mu \psi^\dagger$$



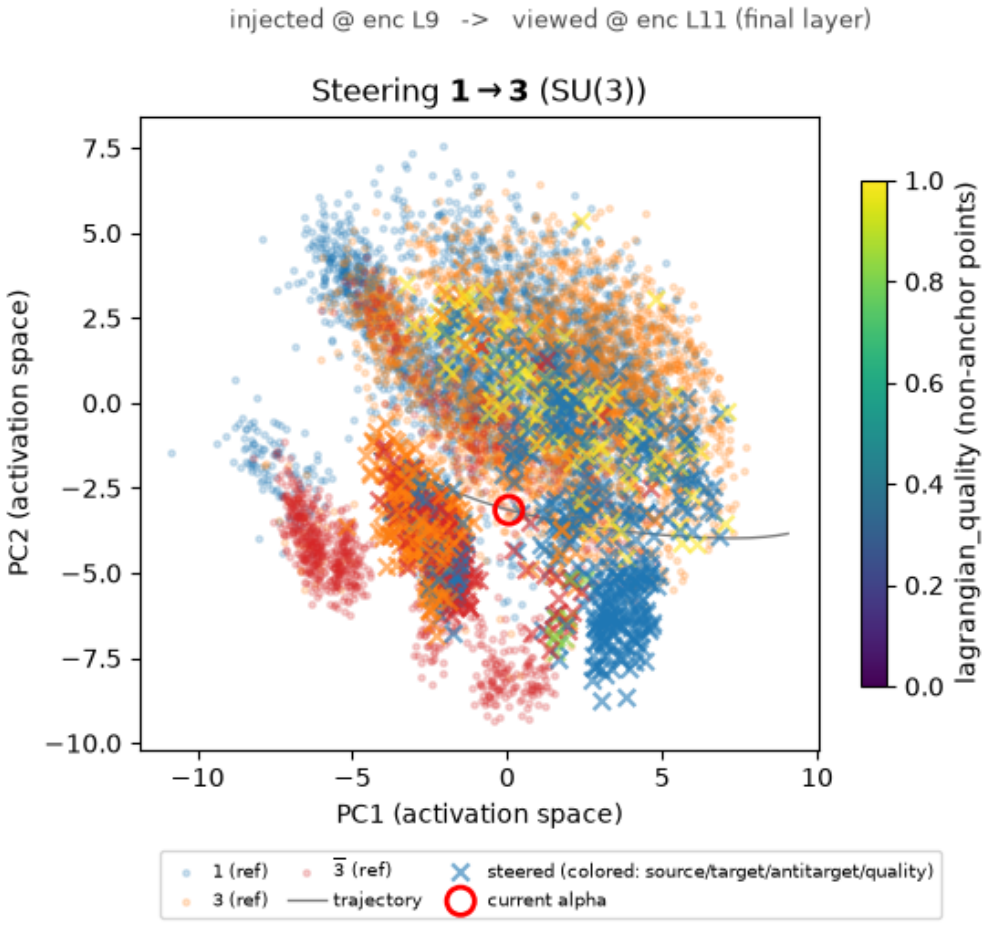
alpha = 0.00 **success = 0%** **cumulative = 0%**

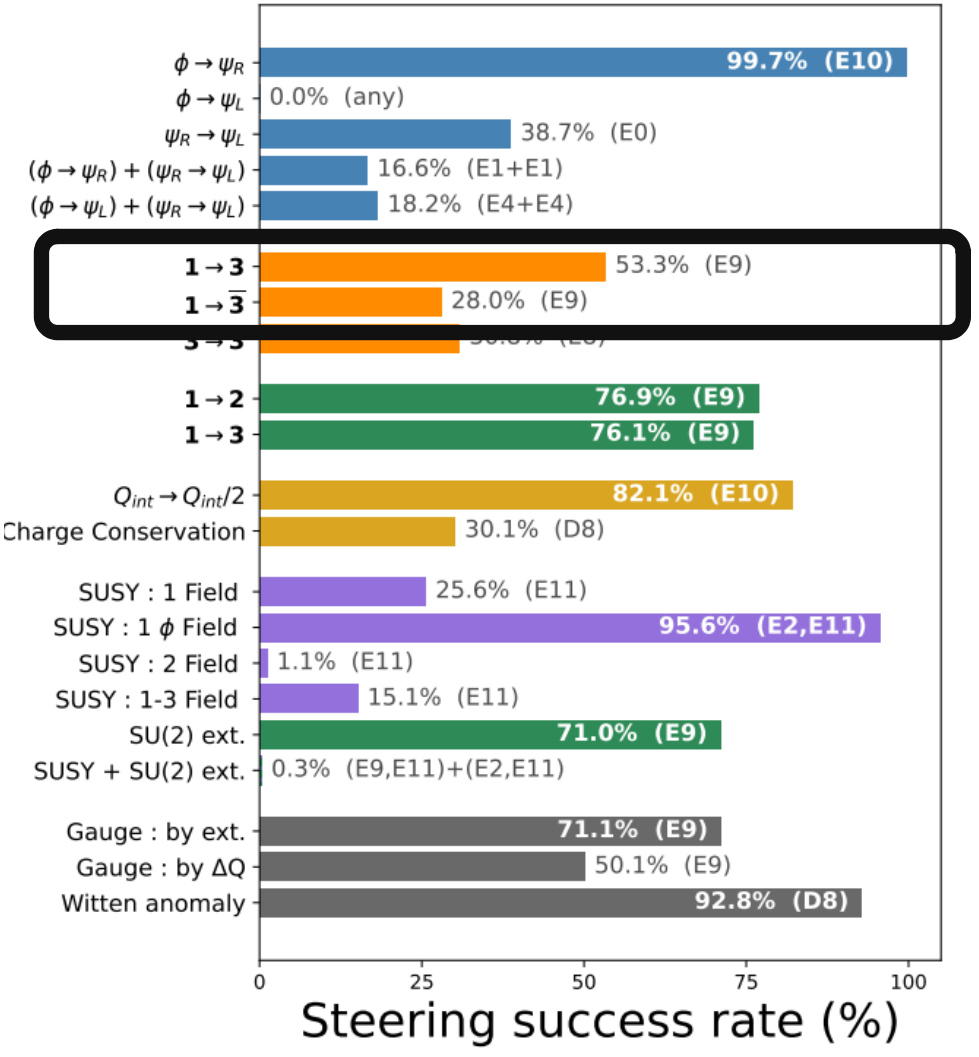




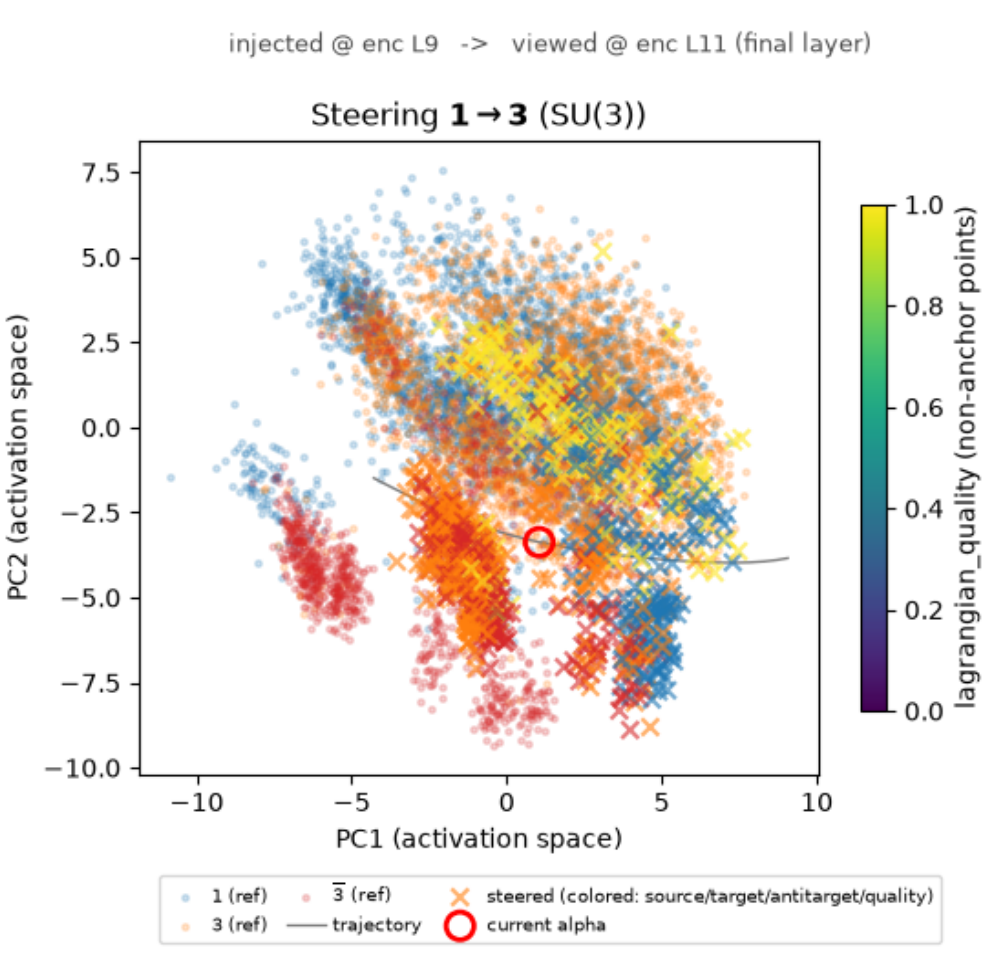


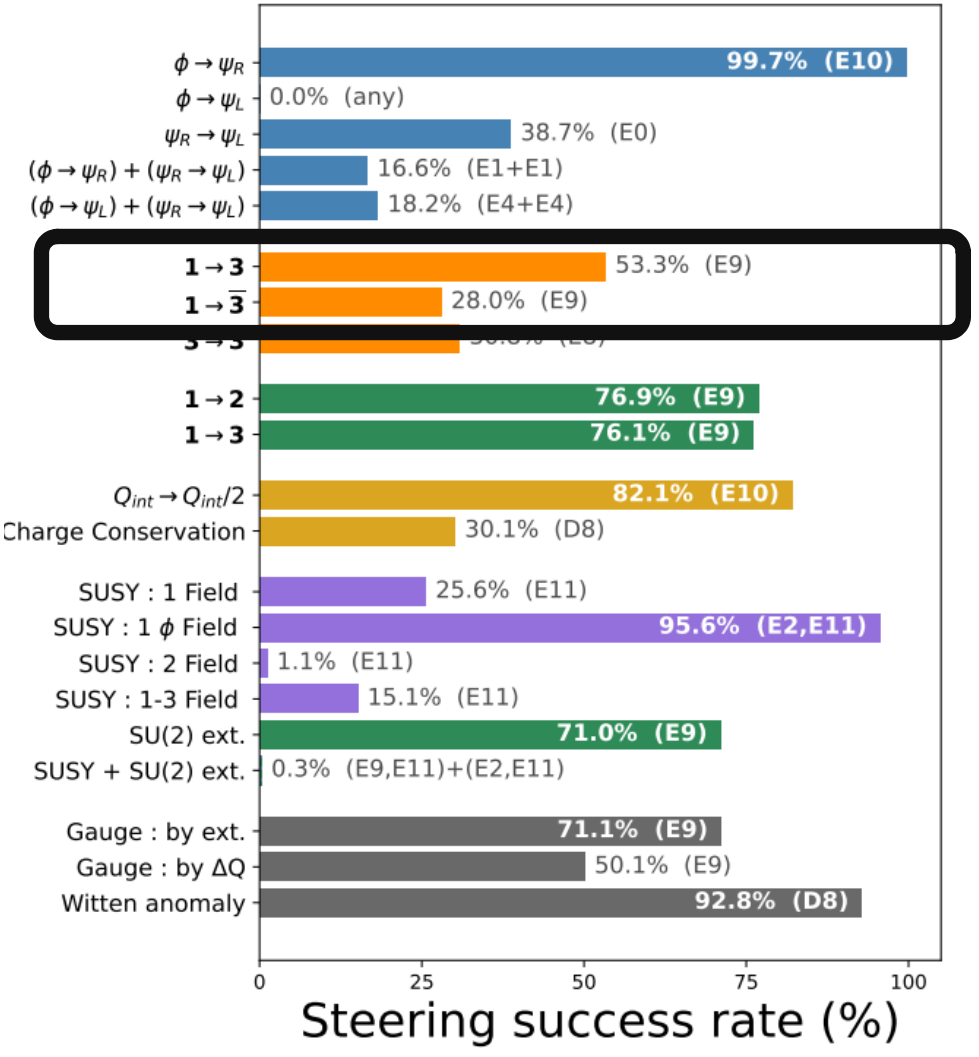
alpha = 0.60 **success = 1%** **cumulative = 1%**



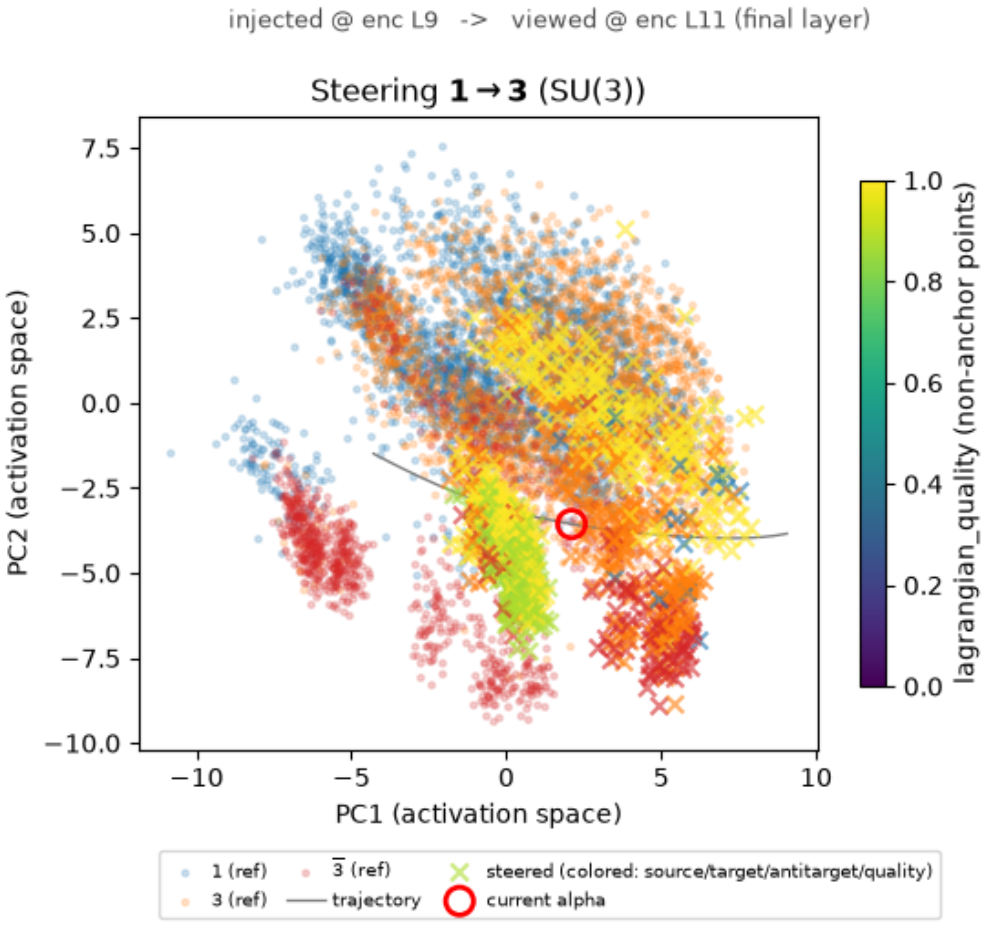


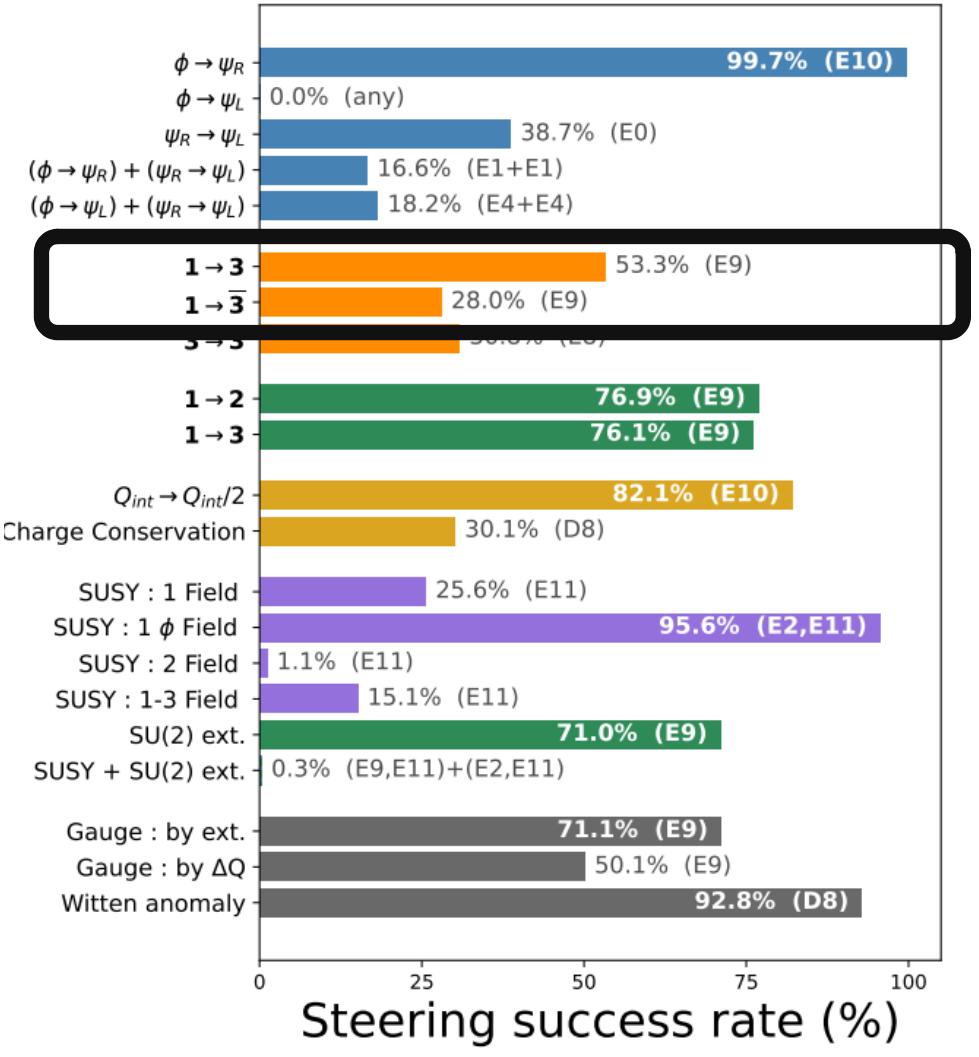
alpha = 0.80 **success = 27%** **cumulative = 27%**



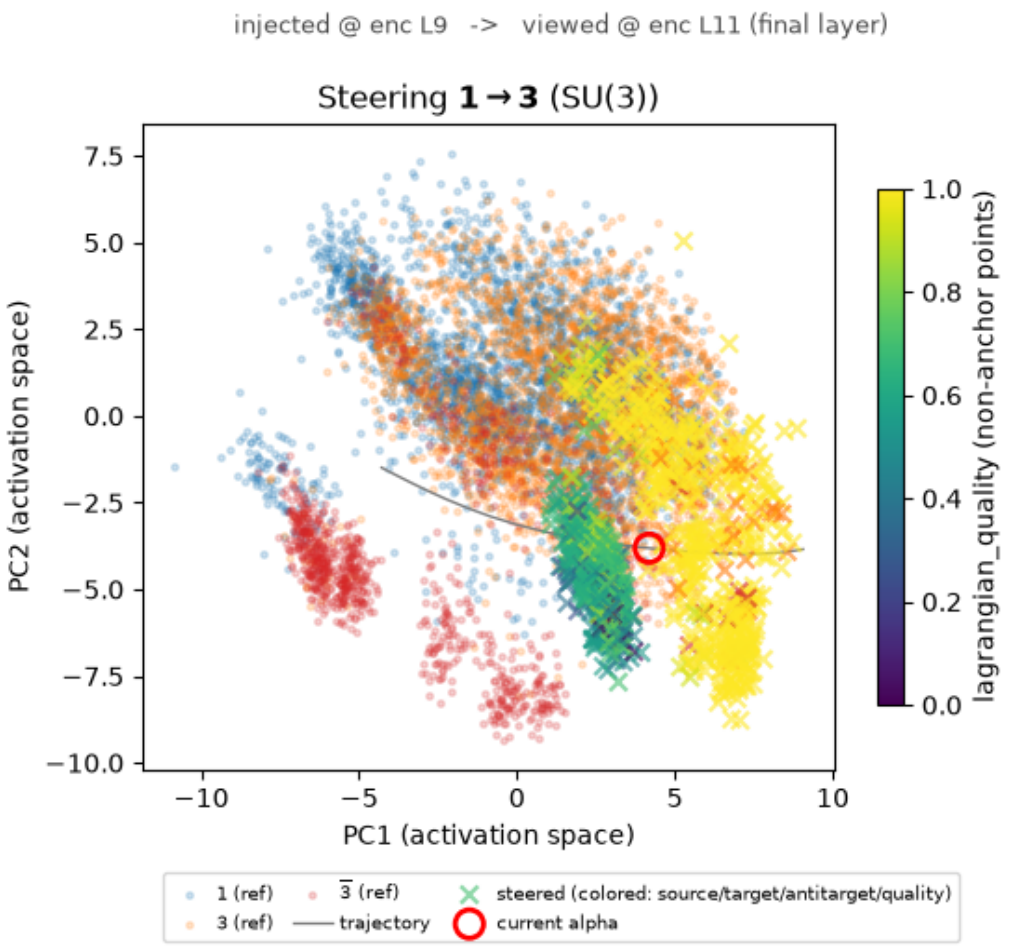


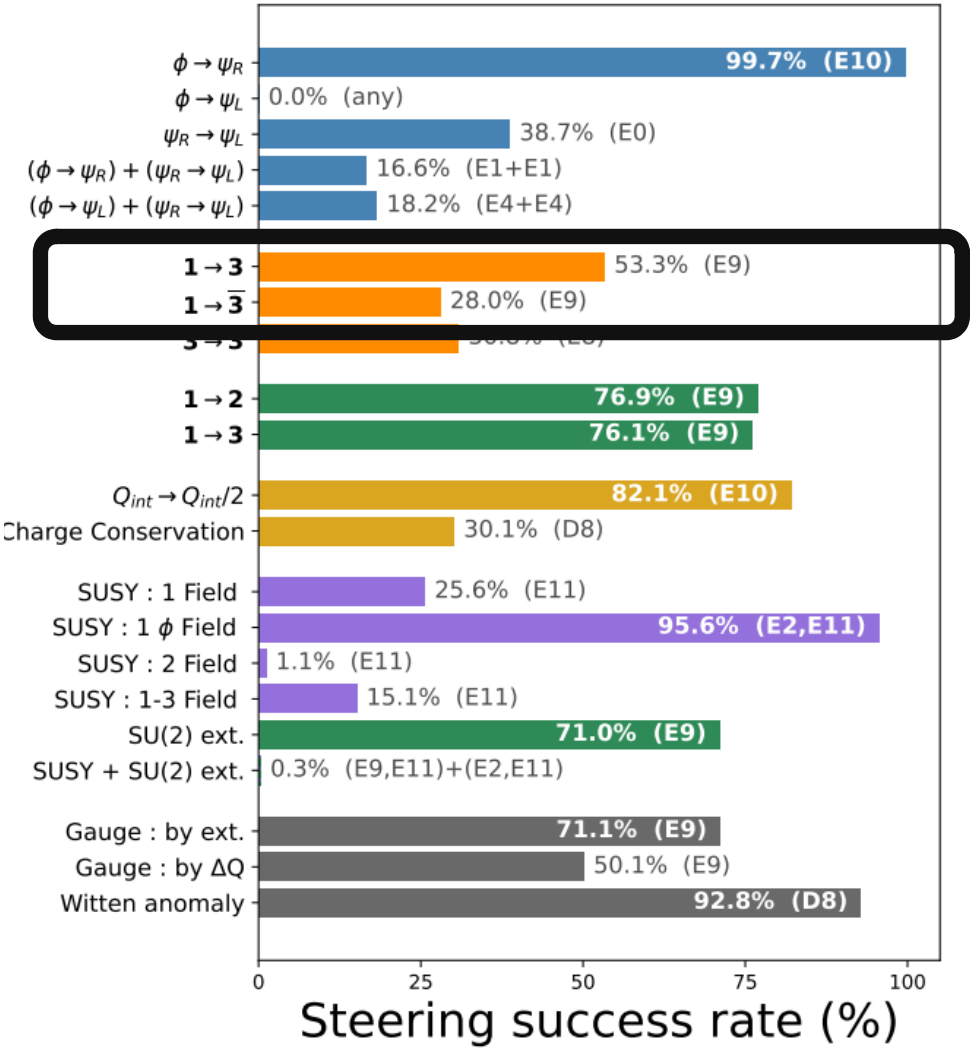
alpha = 0.90 **success = 36%** **cumulative = 40%**



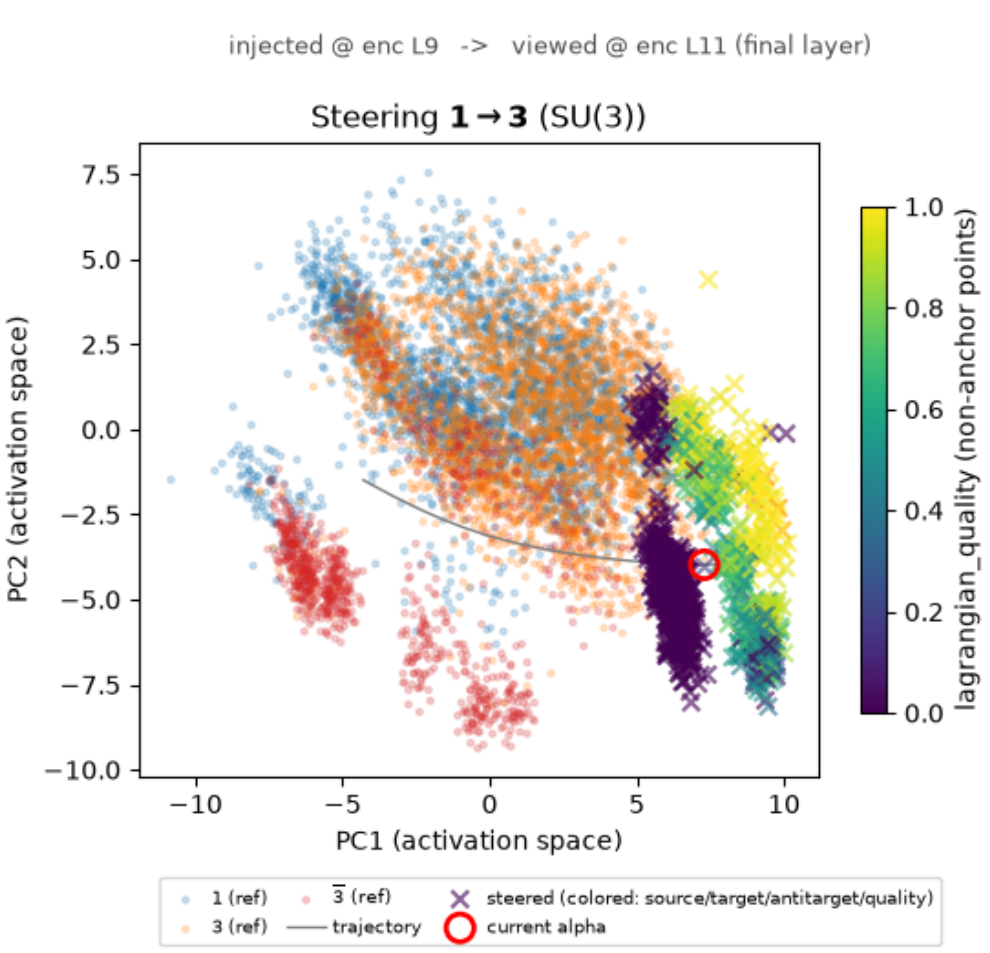


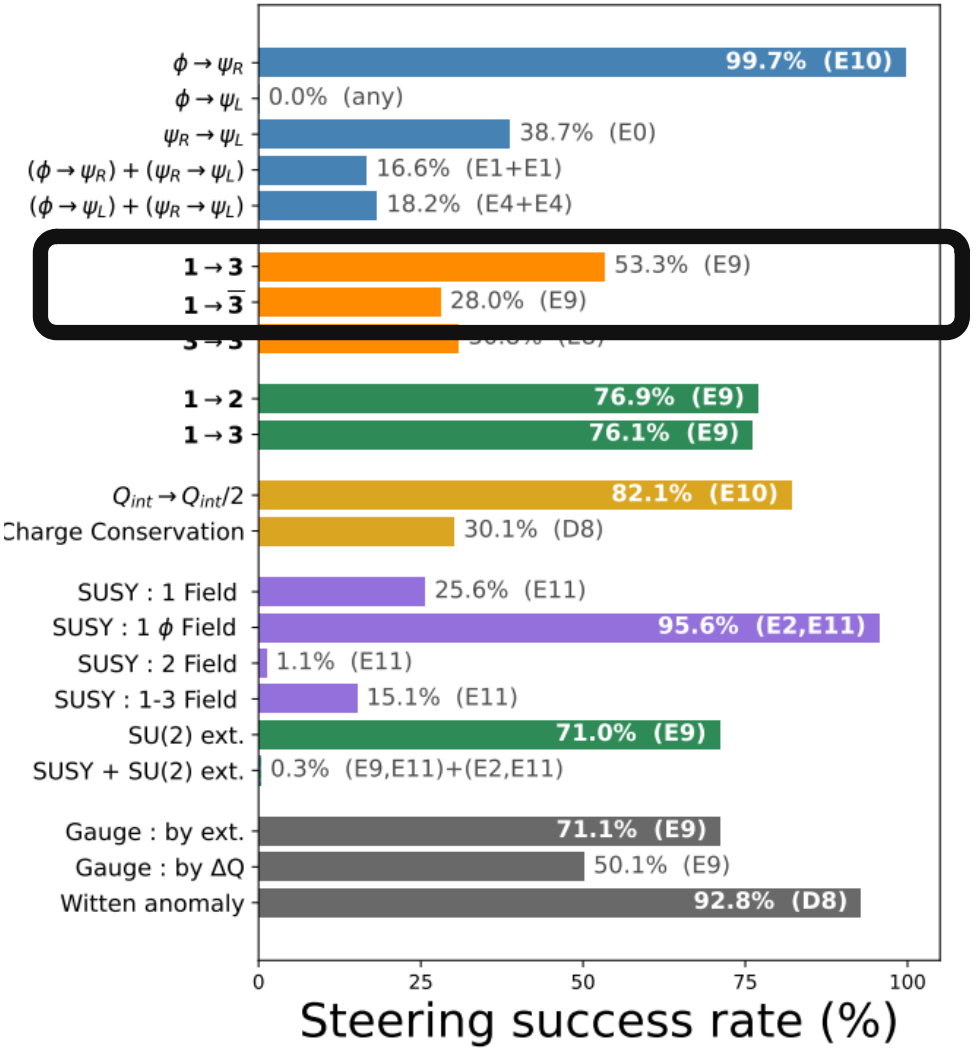
alpha = 1.00 **success = 21%** **cumulative = 50%**



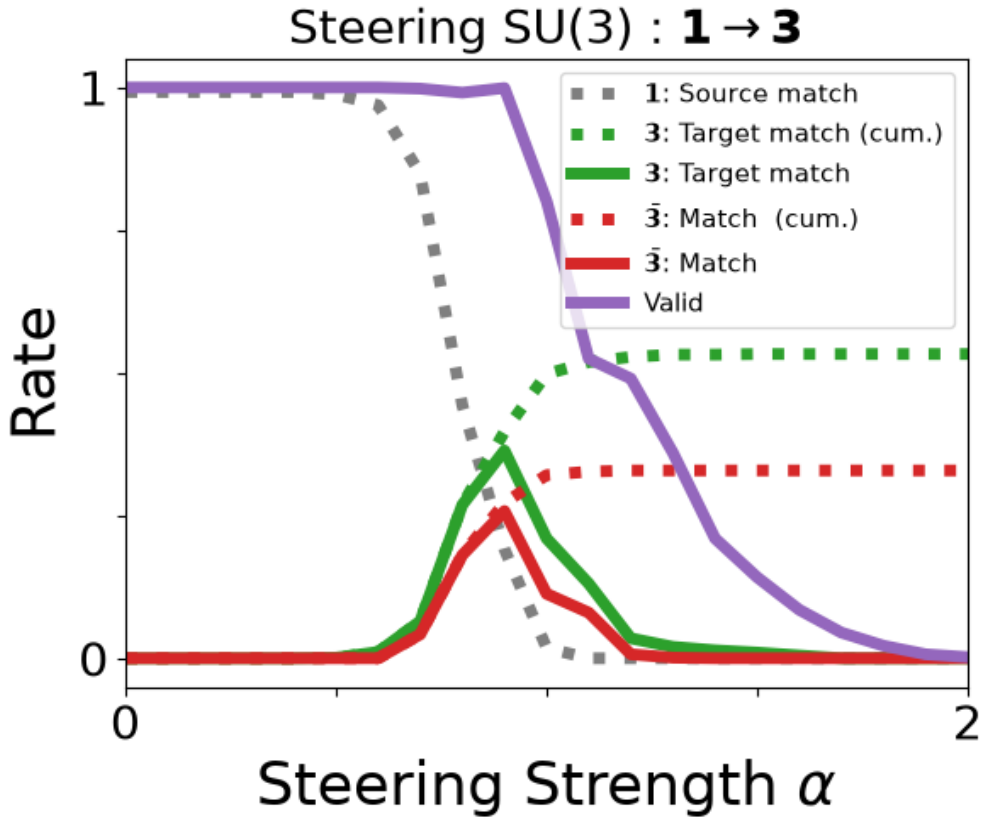


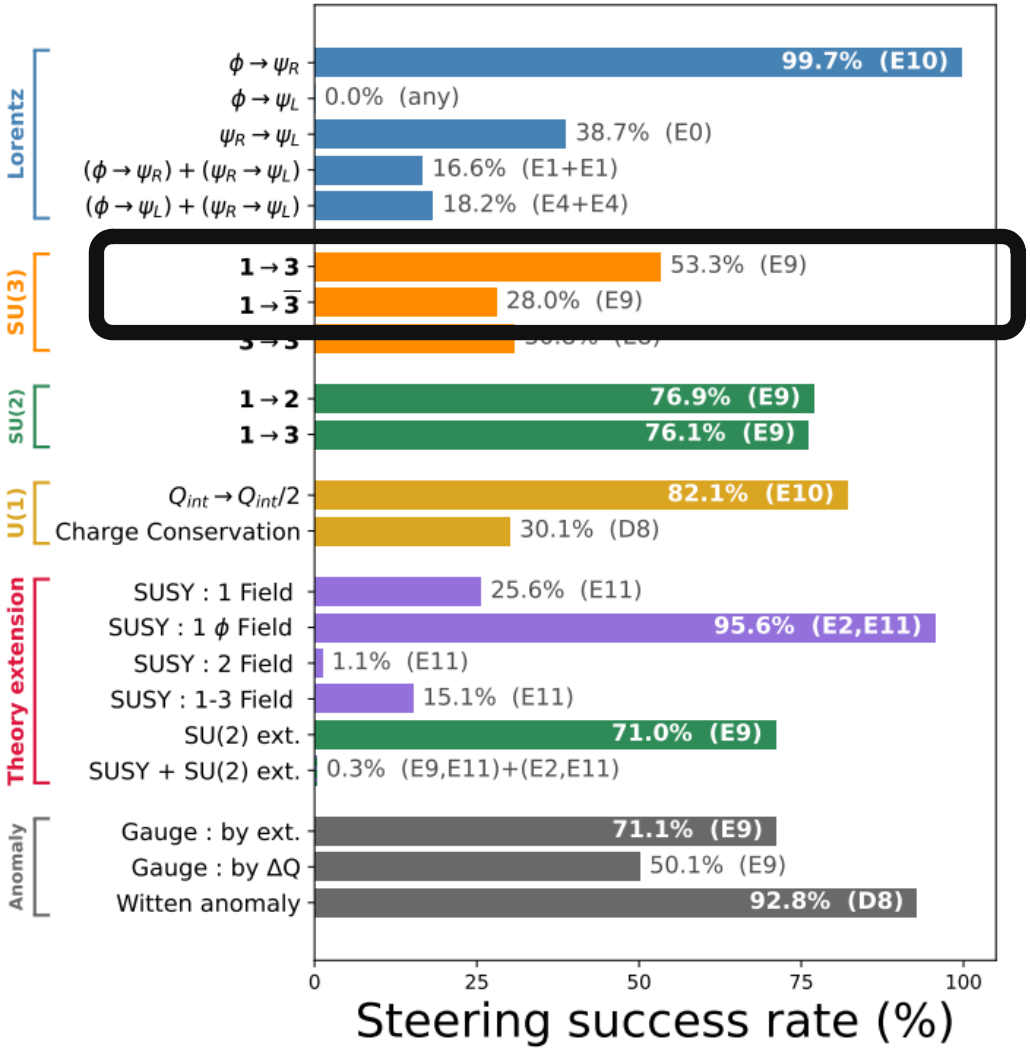
alpha = 1.20 **success = 4%** **cumulative = 53%**





alpha = 1.60 **success = 0%** **cumulative = 53%**

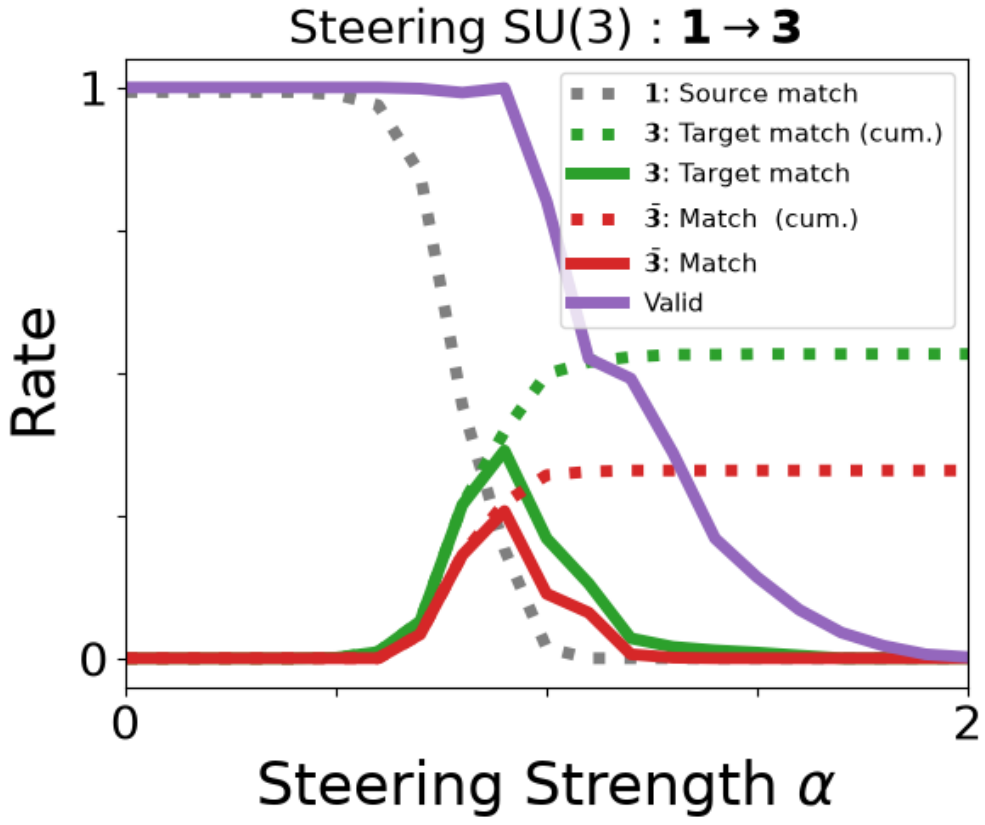




Behavior

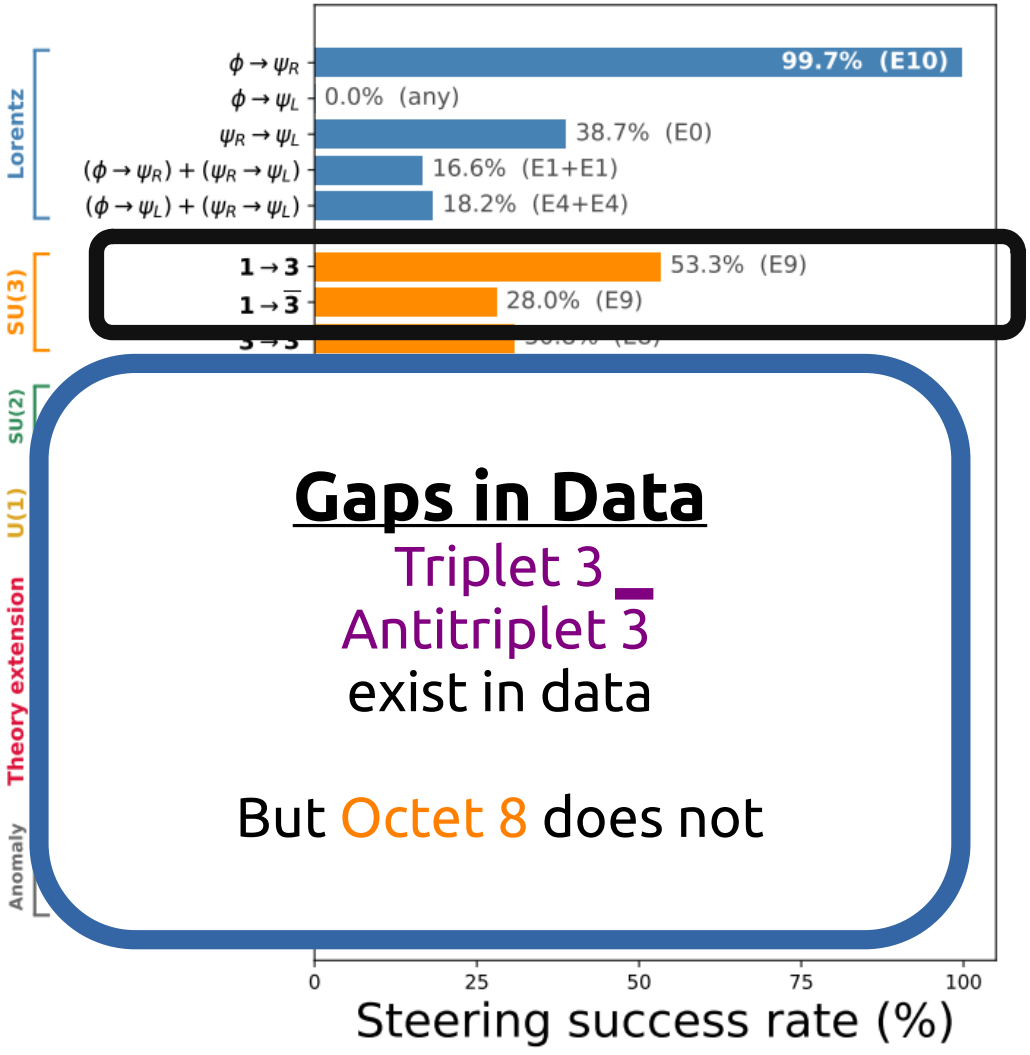
It goes to 3,

but when it doesn't, it goes to $\bar{3}$



Behavior

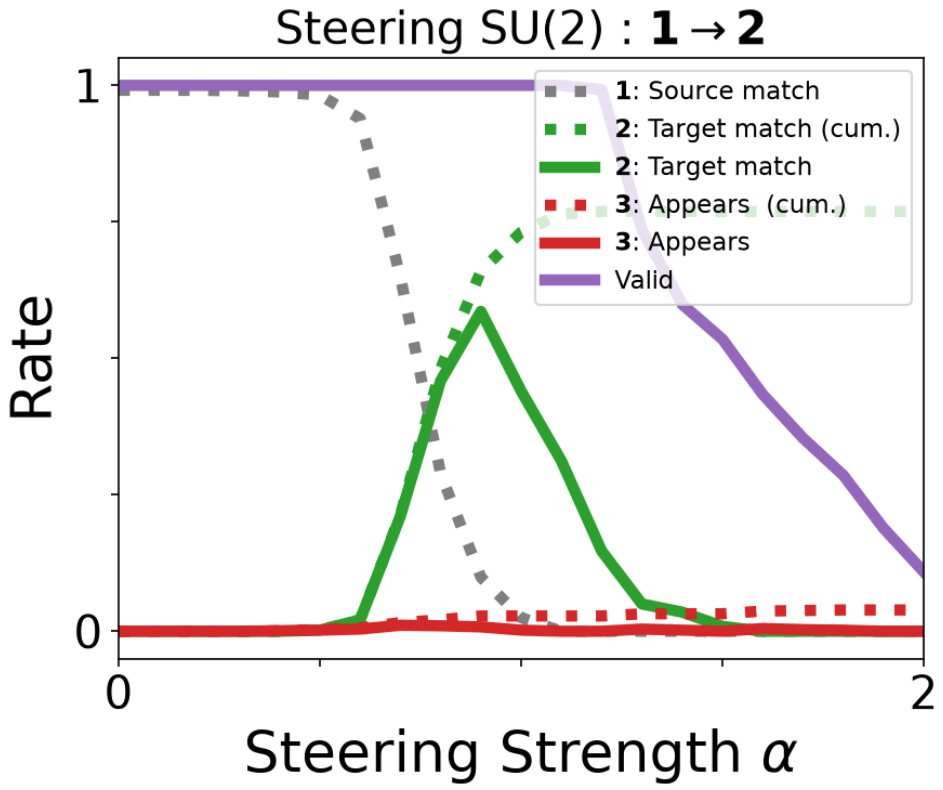
It goes to 3,
but when it doesn't, it goes to $\overline{3}$

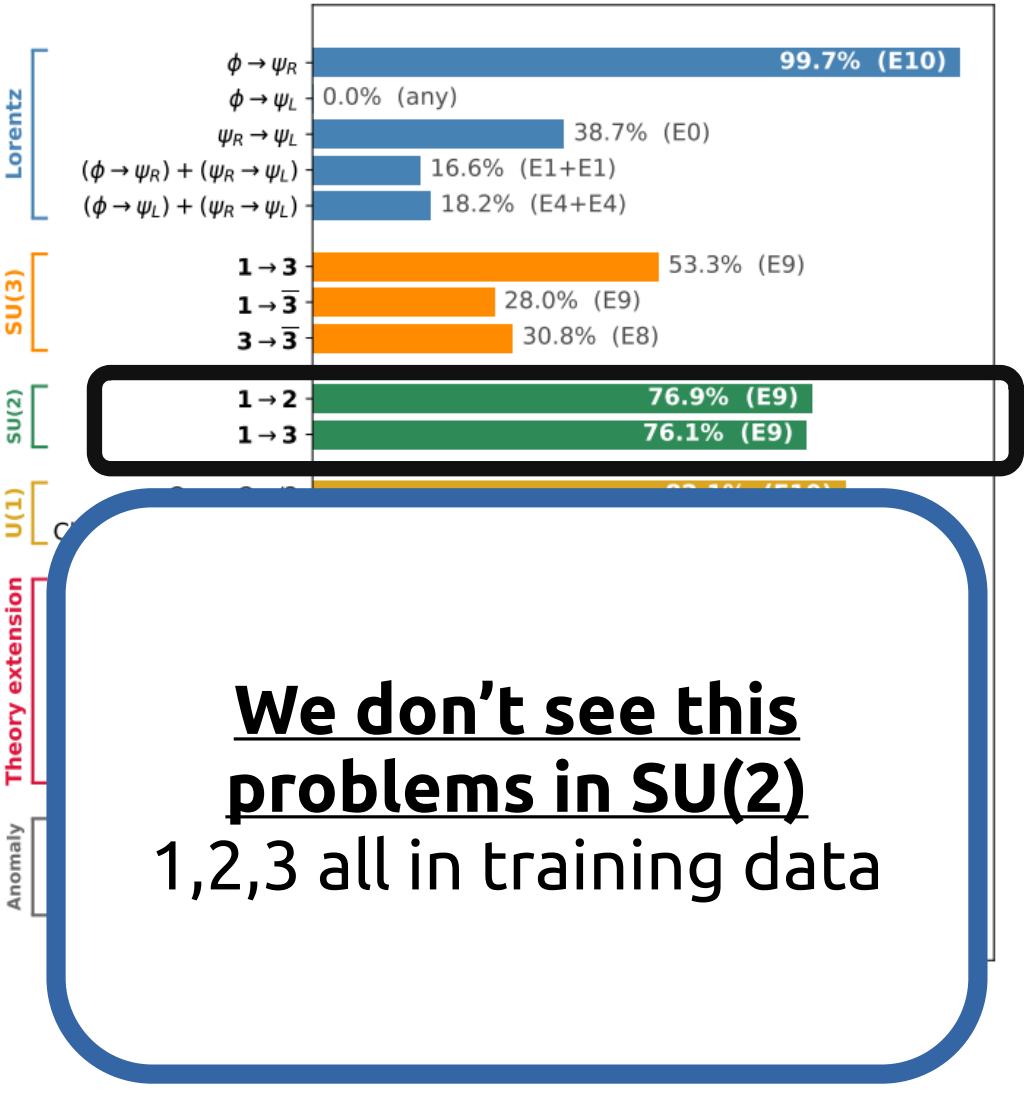


Gaps in Data

Triplet 3
Antitriplet $\overline{3}$
exist in data

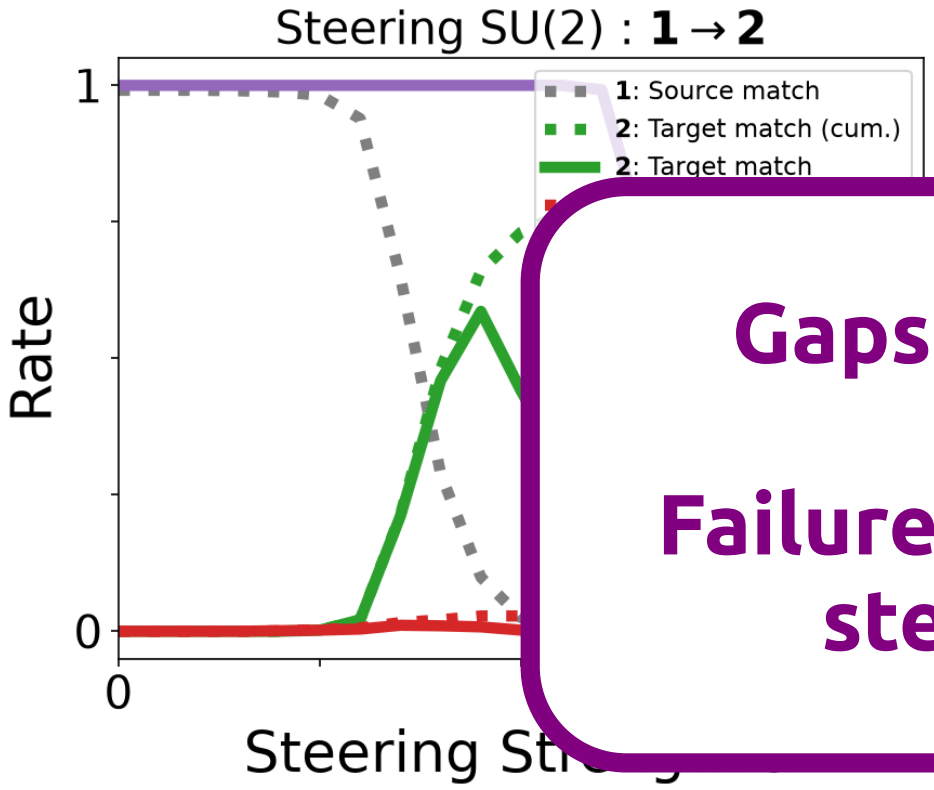
But Octet 8 does not





We don't see this problems in SU(2)

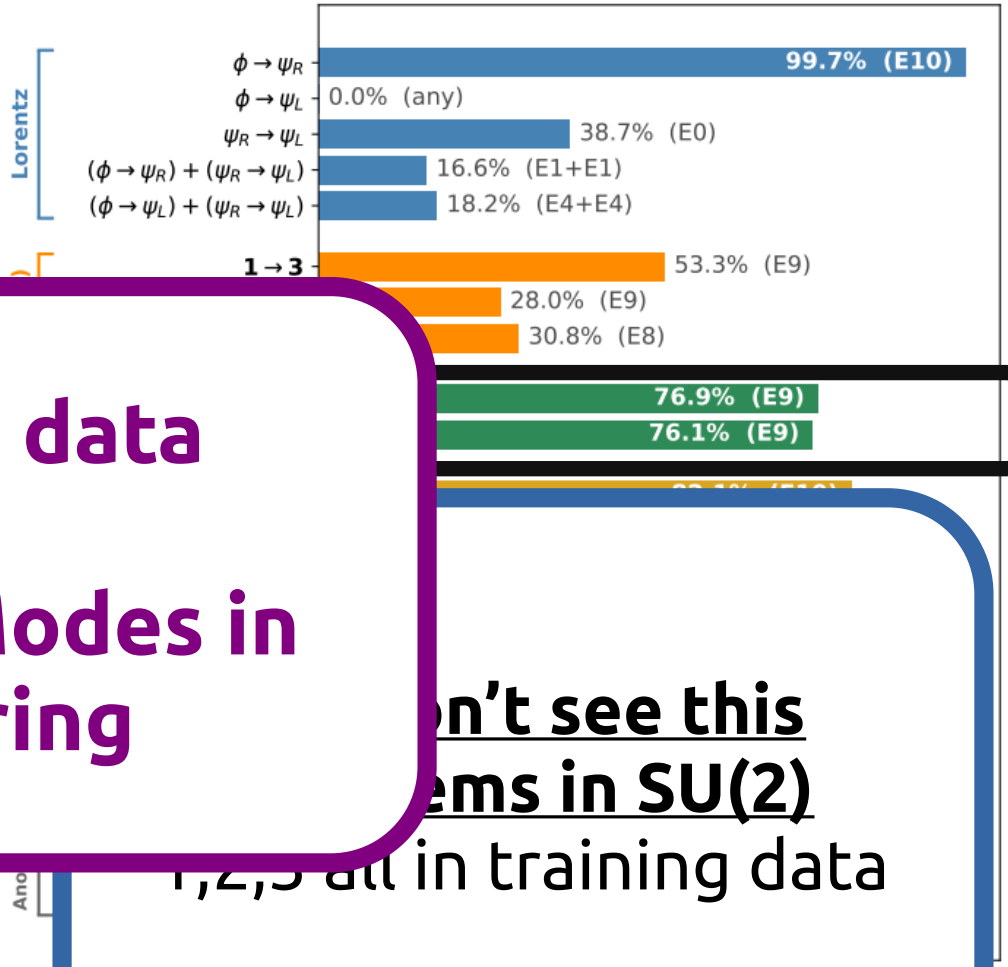
1,2,3 all in training data



Gaps in data

~

Failure Modes in steering

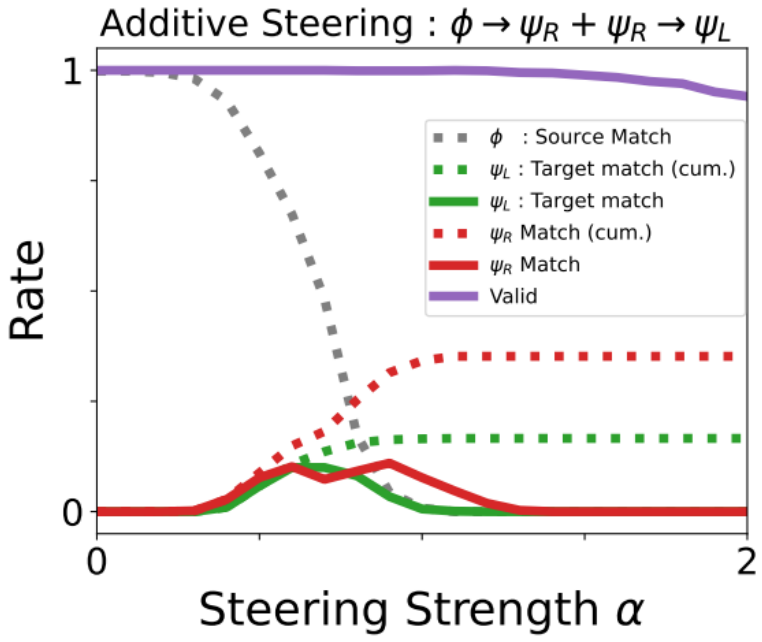


<u>n't see this</u>

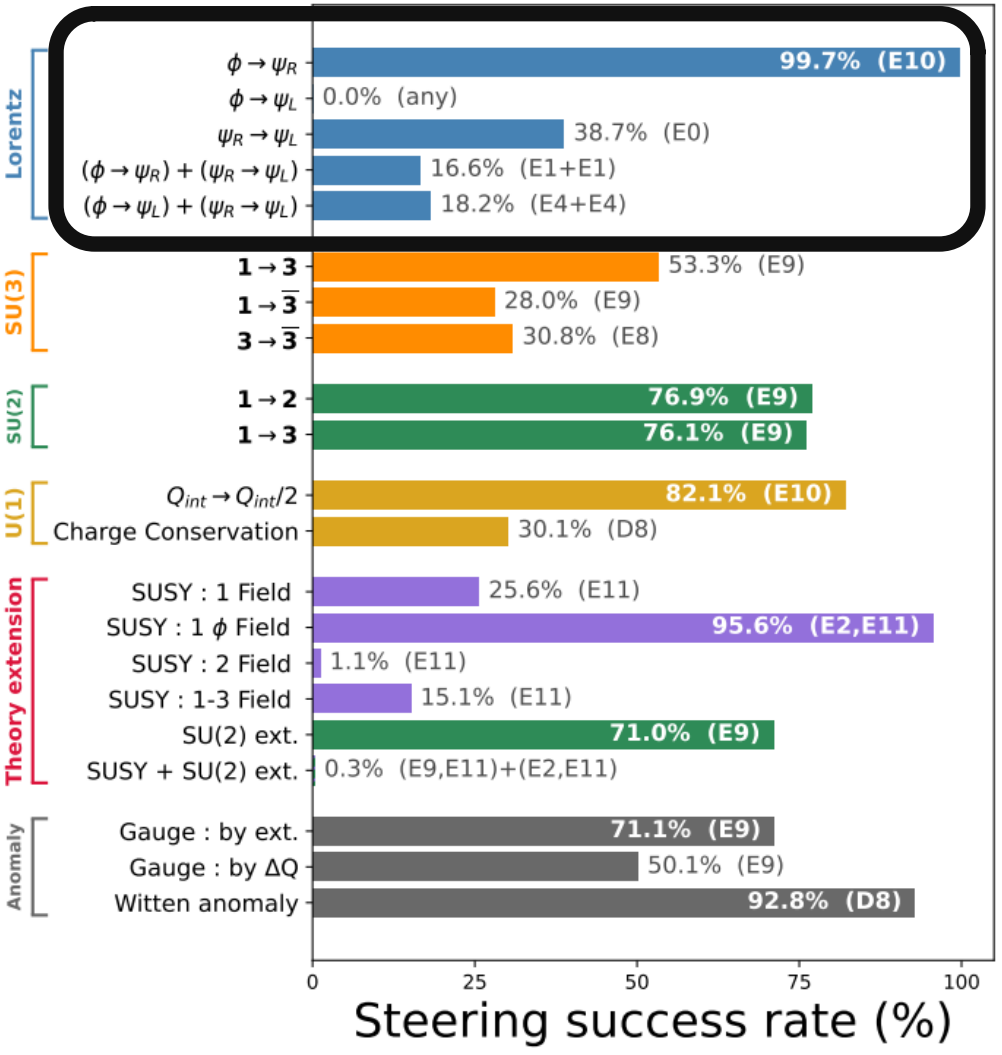
<u>ems in SU(2)</u>

1,2,3 all in training data

Steering can be additive:

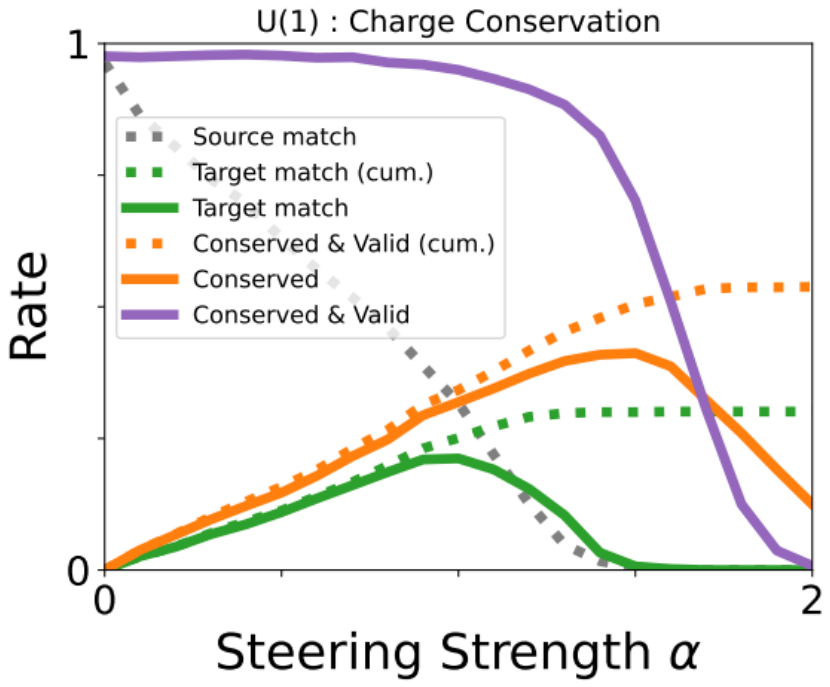


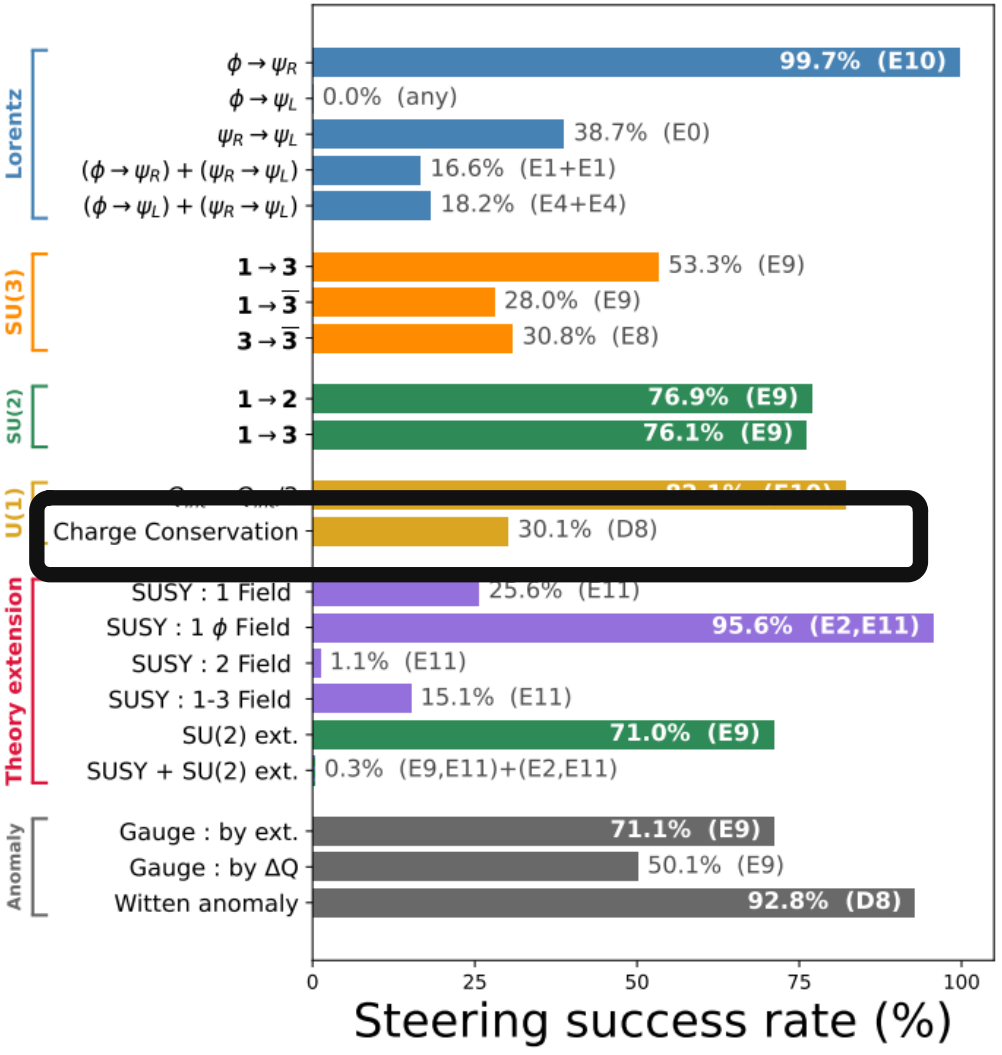
Ψ_L was inaccessible by a single direction.
Additive steering reaches it,
but still leaks to Ψ_R



Post Training Correction or Enforcing symmetries

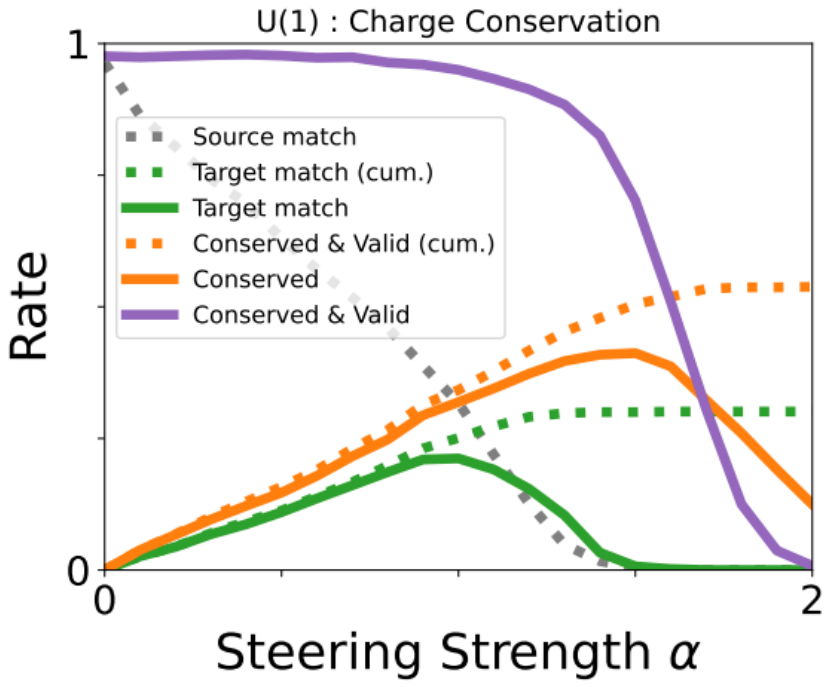
$$\mathcal{L}_{\text{broken}} \rightarrow \mathcal{L}_{\text{conserving}}$$
$$\Sigma Q \neq 0 \rightarrow \Sigma Q = 0$$





Post Training Correction or Enforcing symmetries

$$\mathcal{L}_{\text{broken}} \rightarrow \mathcal{L}_{\text{conserving}}$$
$$\Sigma Q \neq 0 \rightarrow \Sigma Q = 0$$

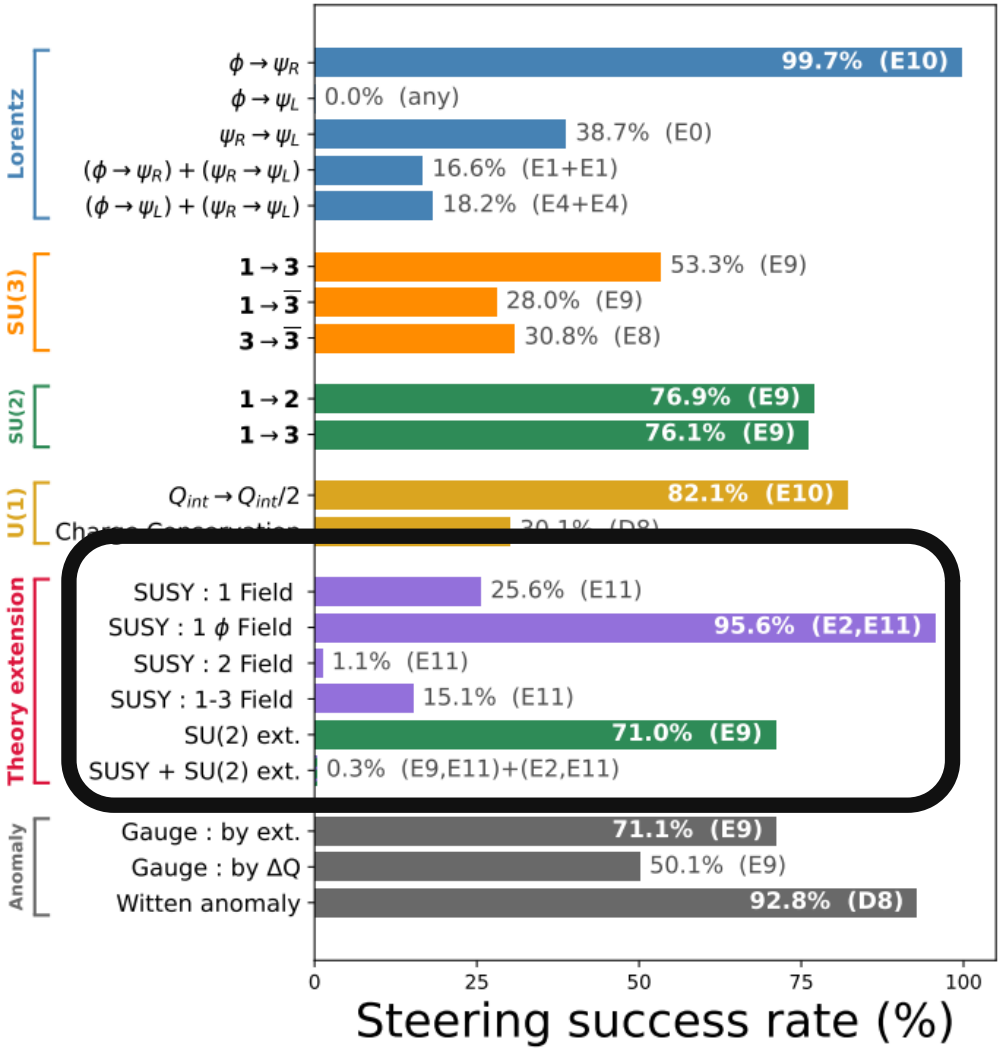


Improving Performance without any more training!

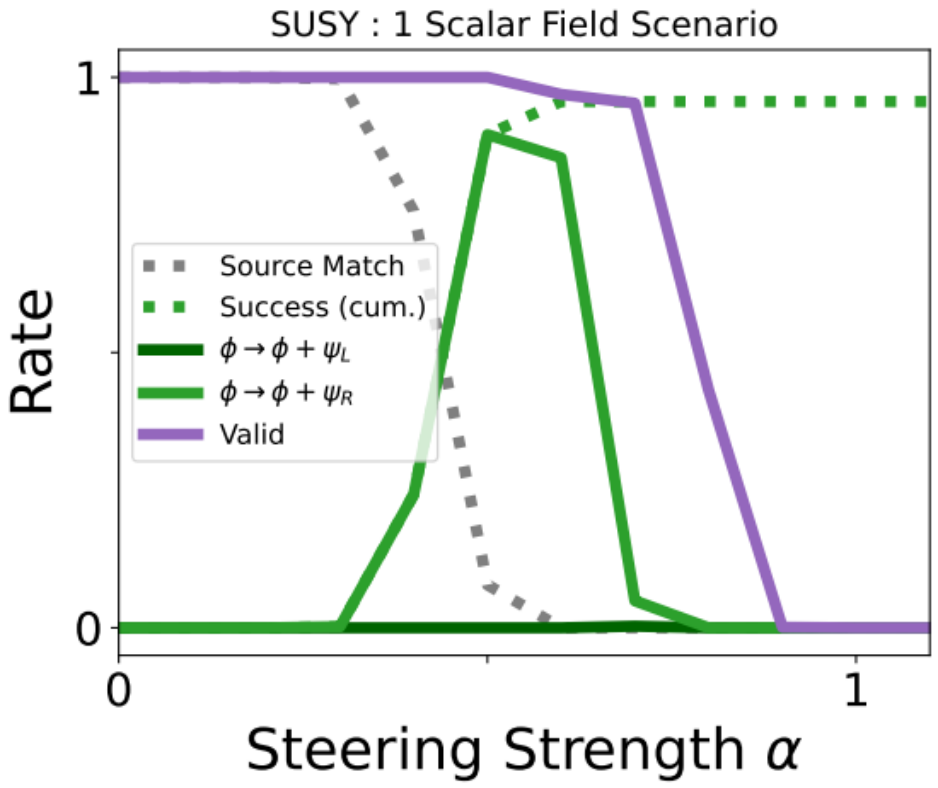
Changing Theory Domains Model Extensions:

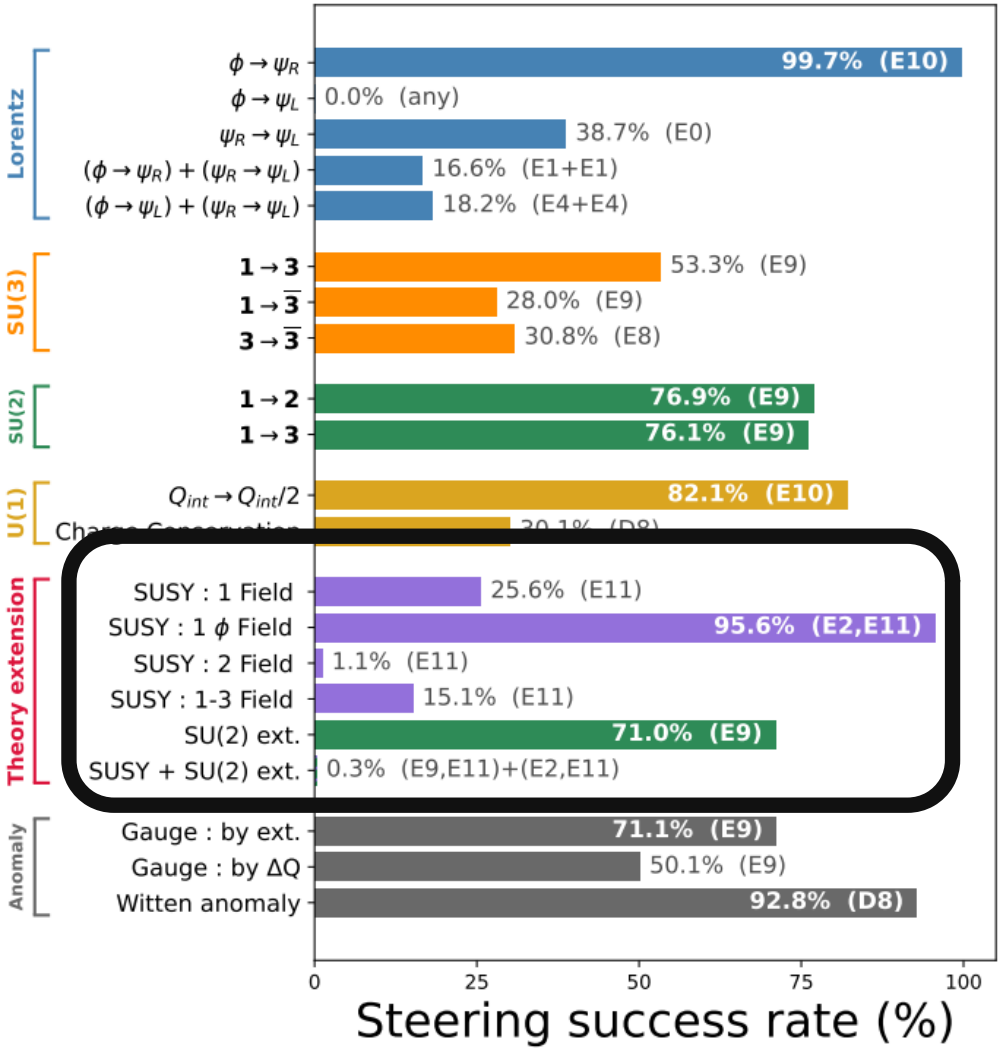
SUSY-like extension :
$\mathcal{L}(F) \rightarrow \mathcal{L}(F, \tilde{F})$

SUSY-like : extending to superpartner of input fields. Did not include Gaugino Sector.



Changing Theory Domains Model Extensions:





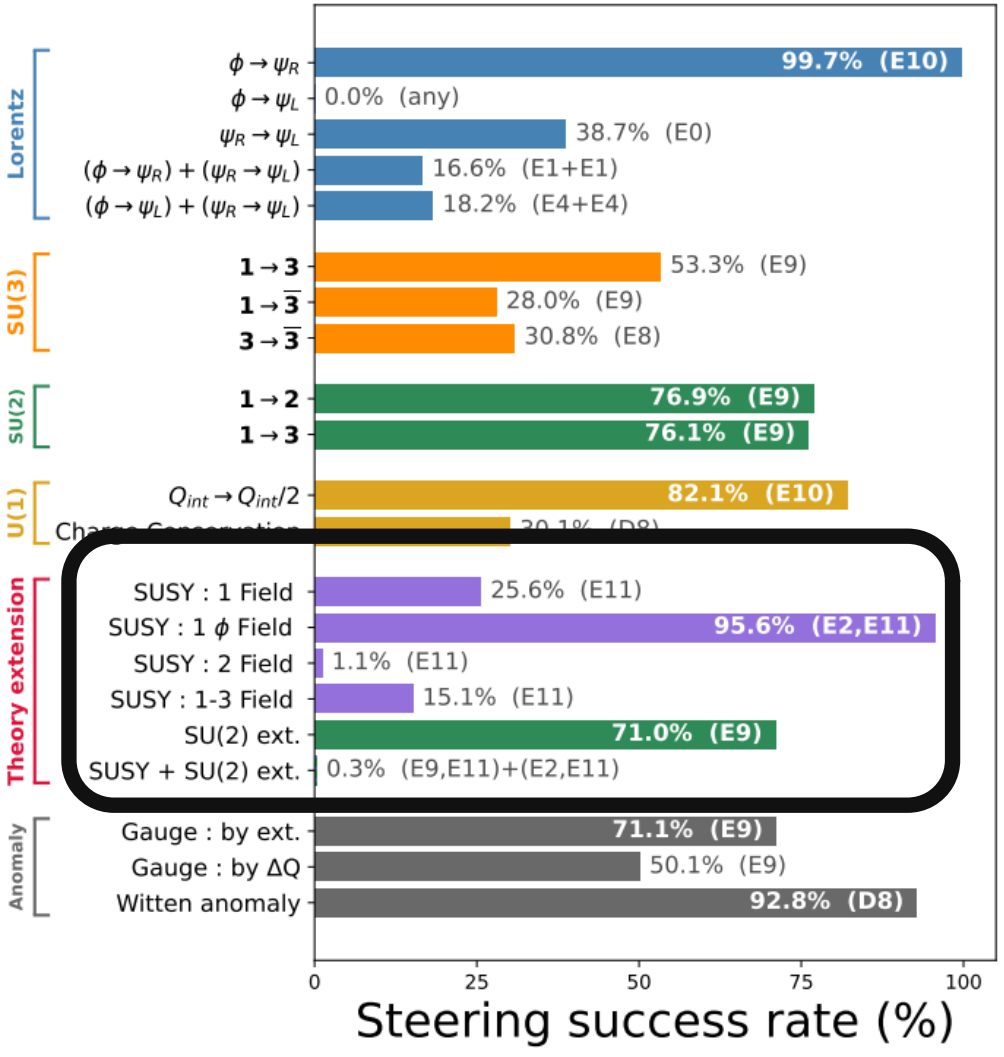
Changing Theory Domains Model Extensions:

SUSY-like extension :
$\mathcal{L}(F) \rightarrow \mathcal{L}(F,\tilde{F})$

Extending Gauge Group:
$\mathcal{L}_G \rightarrow \mathcal{L}_{GxSU(2)}$

Both
$\mathcal{L}_G(F) \rightarrow \mathcal{L}_{GxSU(2)}(F,\tilde{F})$

SUSY-like : extending to superpartner of input fields. Did not include Gaugino Sector.



Locating Theory Solutions

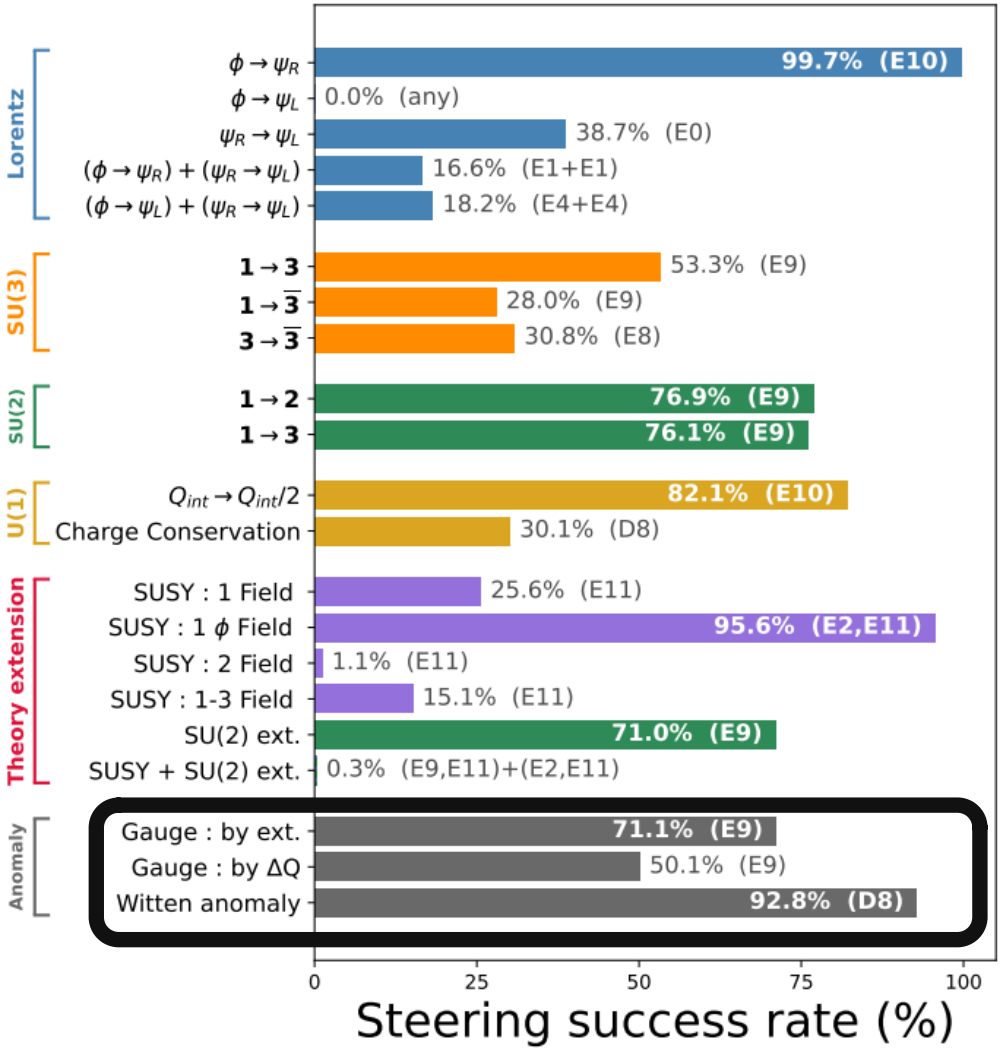
$$\mathcal{L}_{\text{anom}} \rightarrow \mathcal{L}_{\text{anom.free}}$$

Gauge Anomalies

Anomaly Cancellation Conditions*:
A[L] = 0

Witten Anomaly

Number of SU2 doublet fermion must be even



Locating Theory Solutions

$$\mathcal{L}_{anom} \rightarrow \mathcal{L}_{anom.free}$$

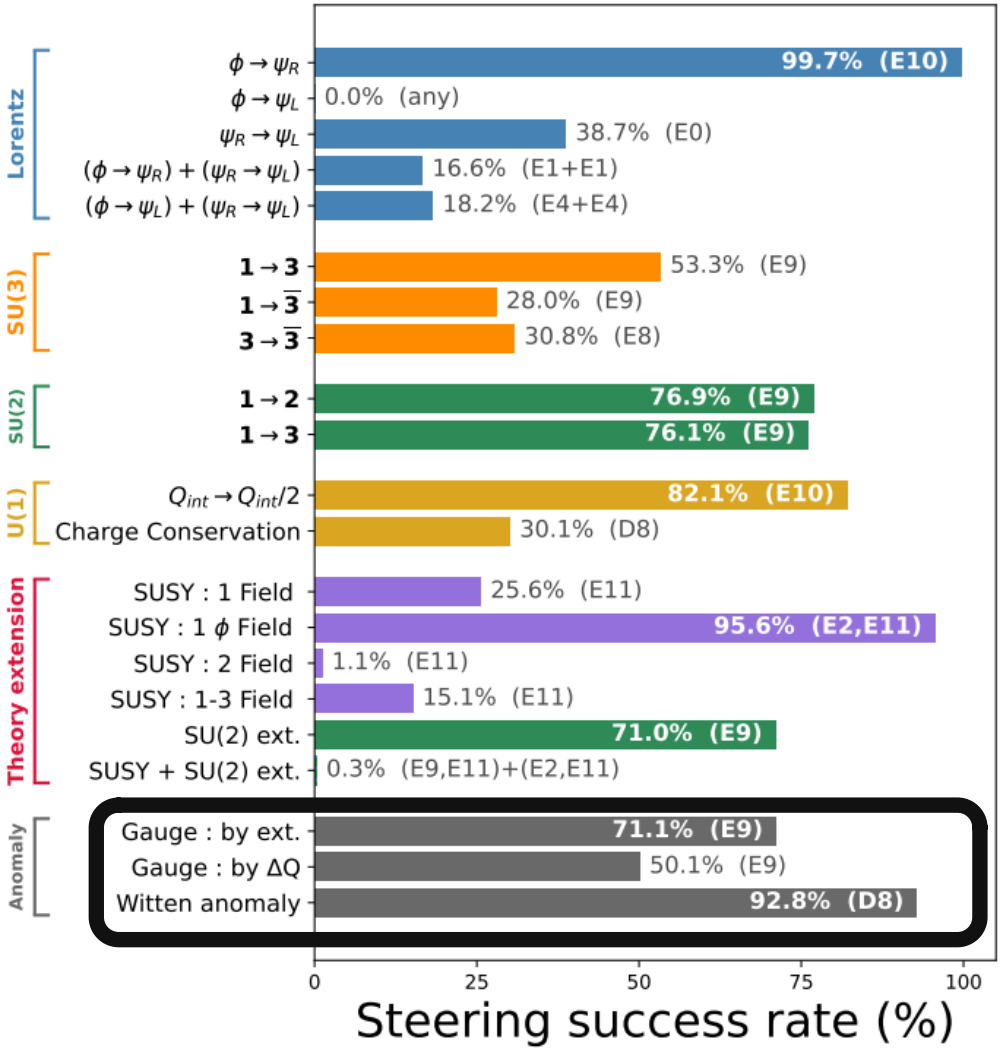
Gauge Anomalies

Anomaly Cancellation Conditions*:
A[L] = 0

Witten Anomaly

Number of SU2 doublet fermion must be even

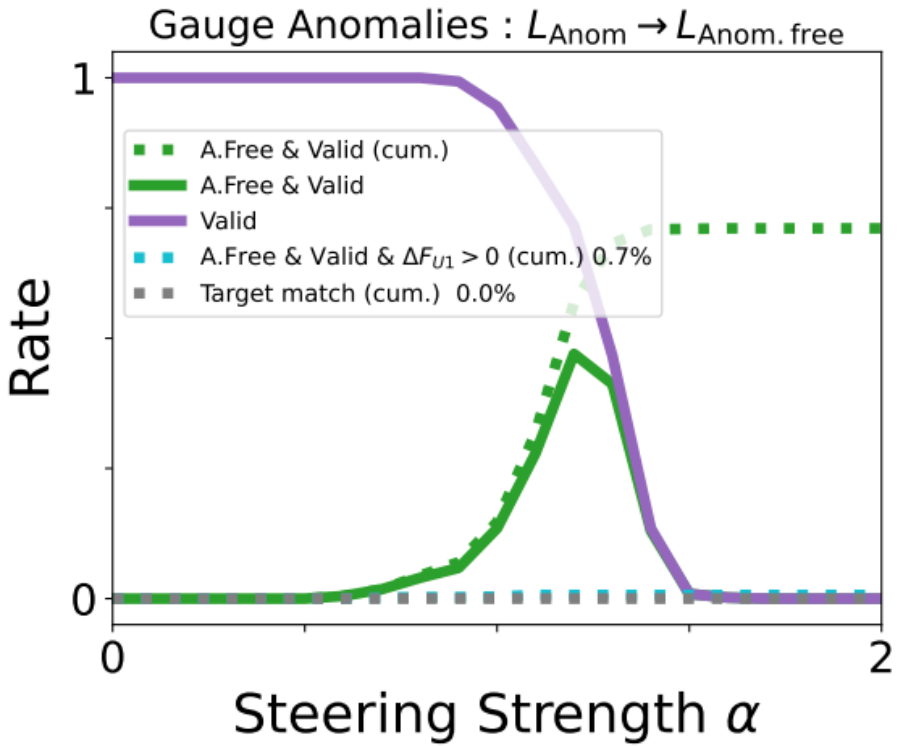
Success:
Anomaly Free and Valid lagrangians
(More than one way to solve anomalies)

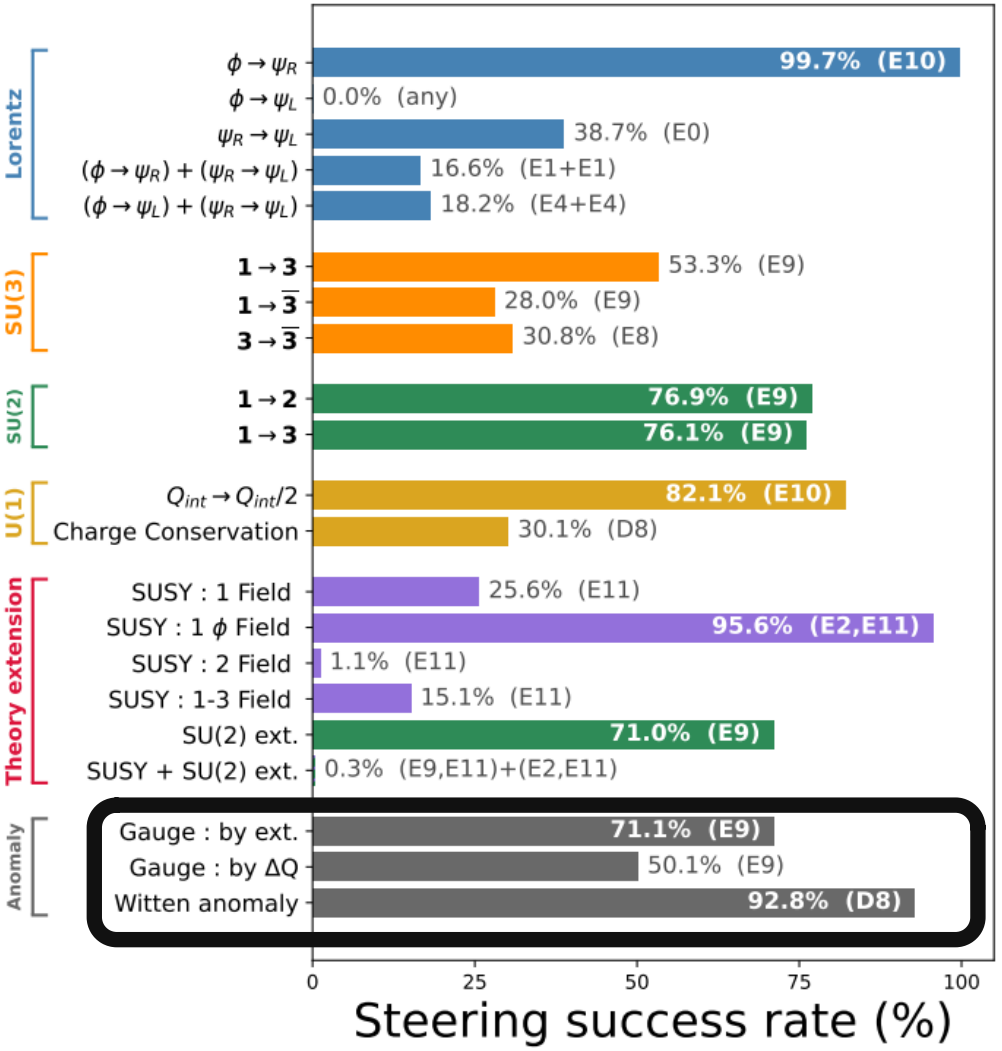


Gauge Anomalies

Anomaly Cancellation Conditions*:

A[L] = 0



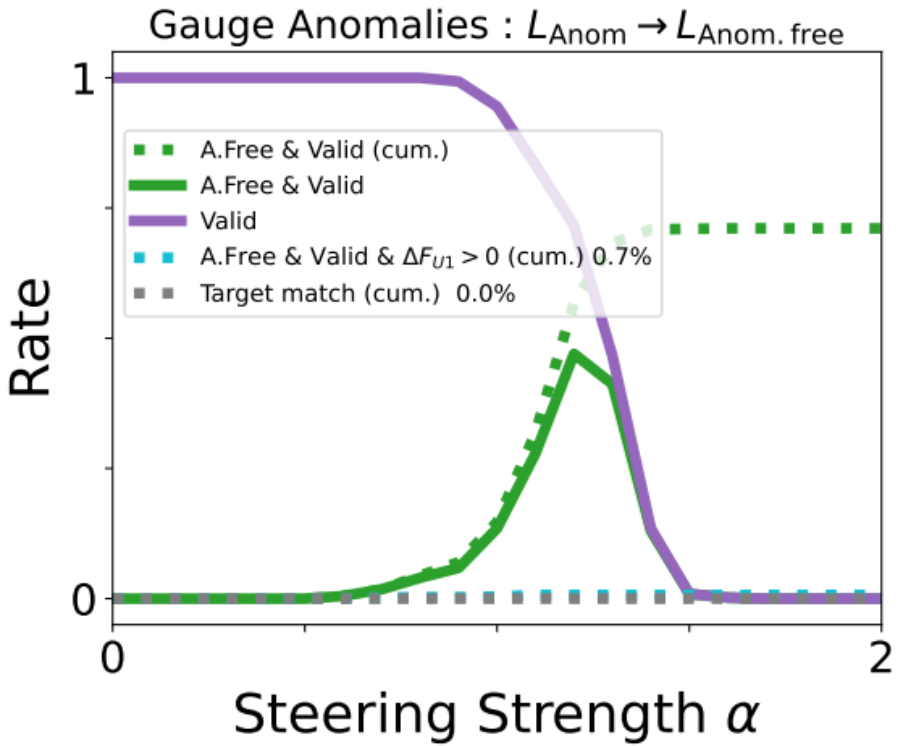


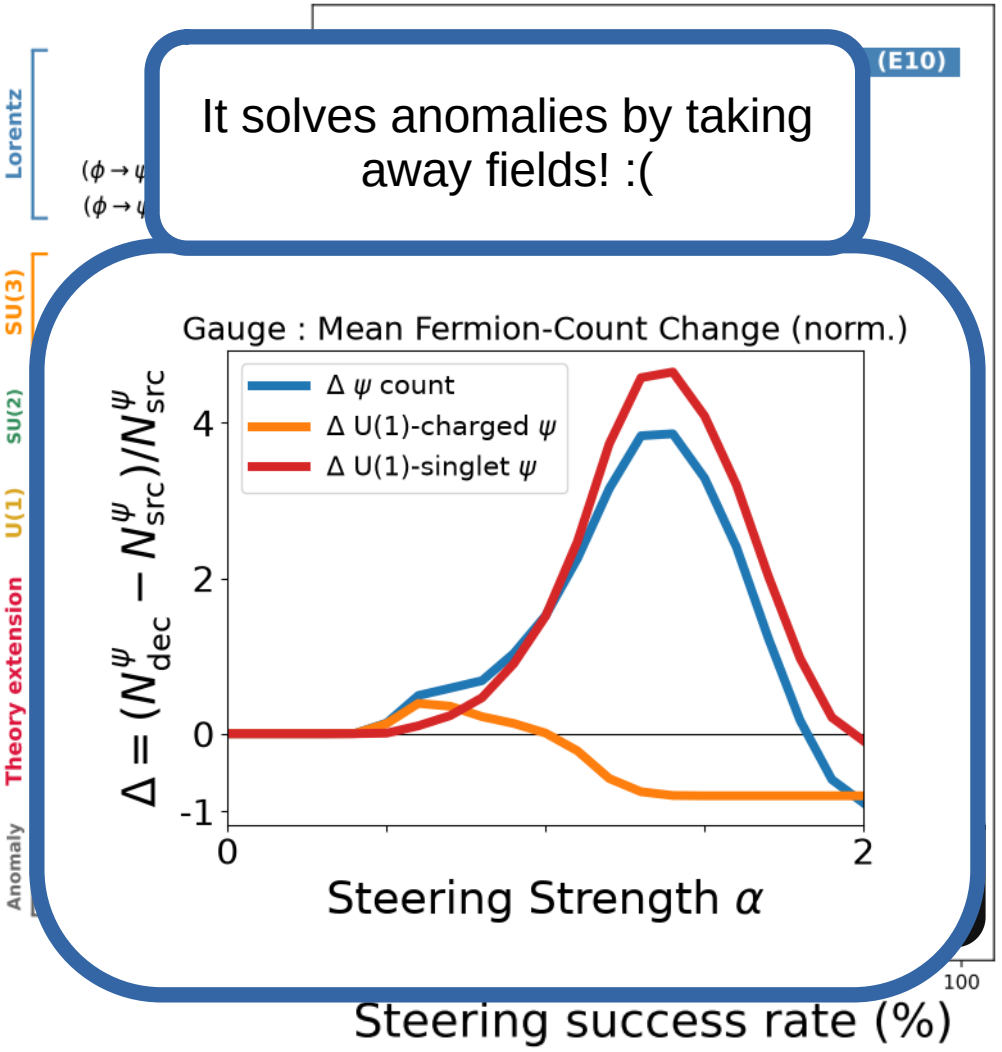
*Only working with trivial ACC solutions for now

Gauge Anomalies

Anomaly Cancellation Conditions*:

A[L] = 0



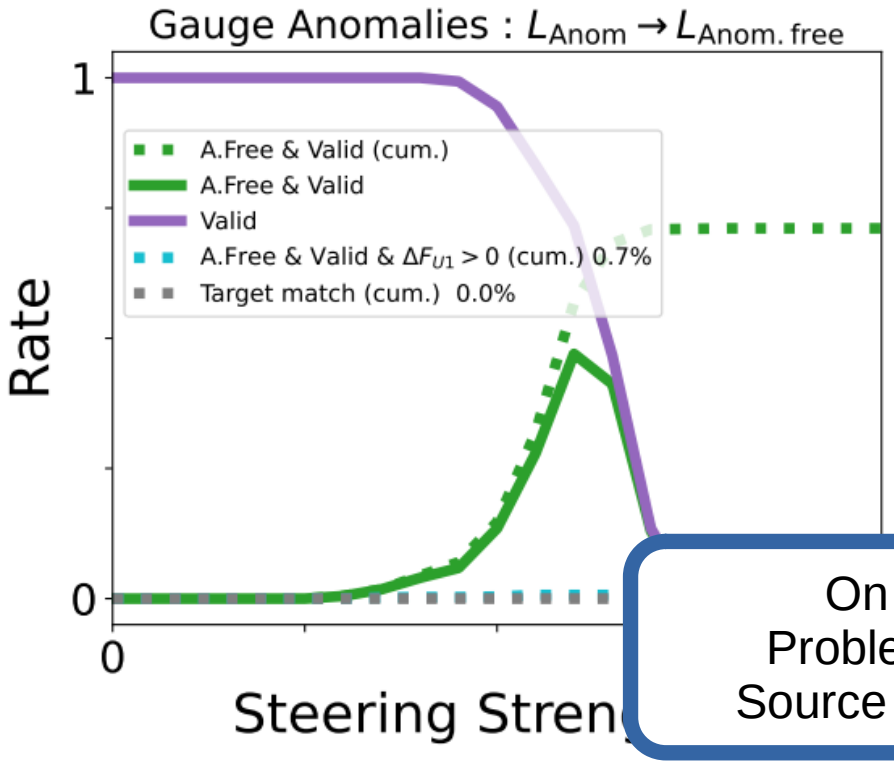


*Only working with trivial ACC solutions for now

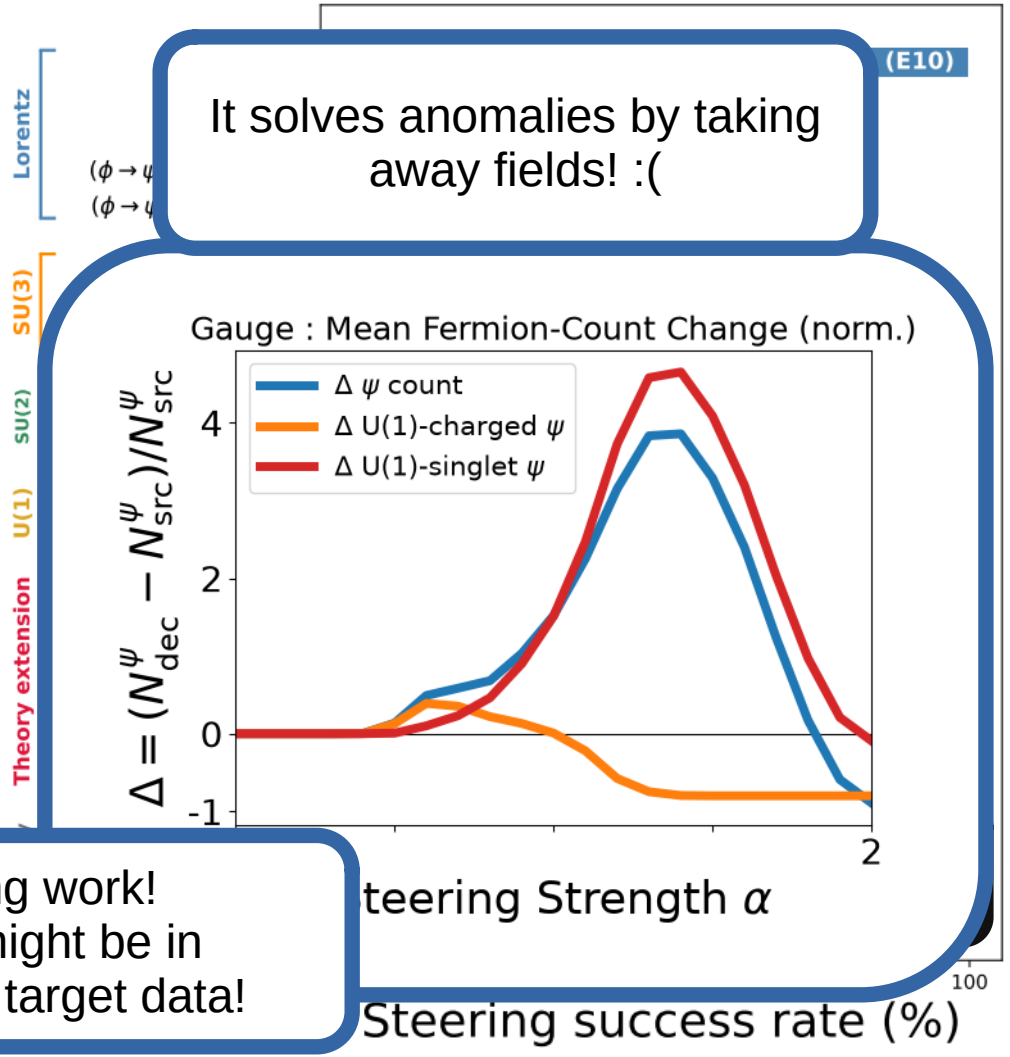
Gauge Anomalies

Anomaly Cancellation Conditions*:

A[L] = 0



It solves anomalies by taking away fields! :(

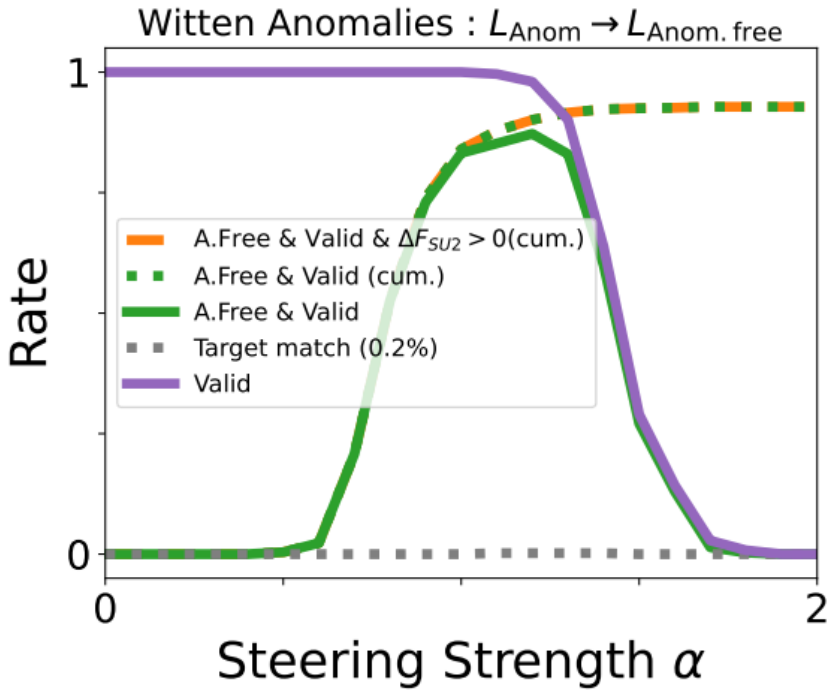


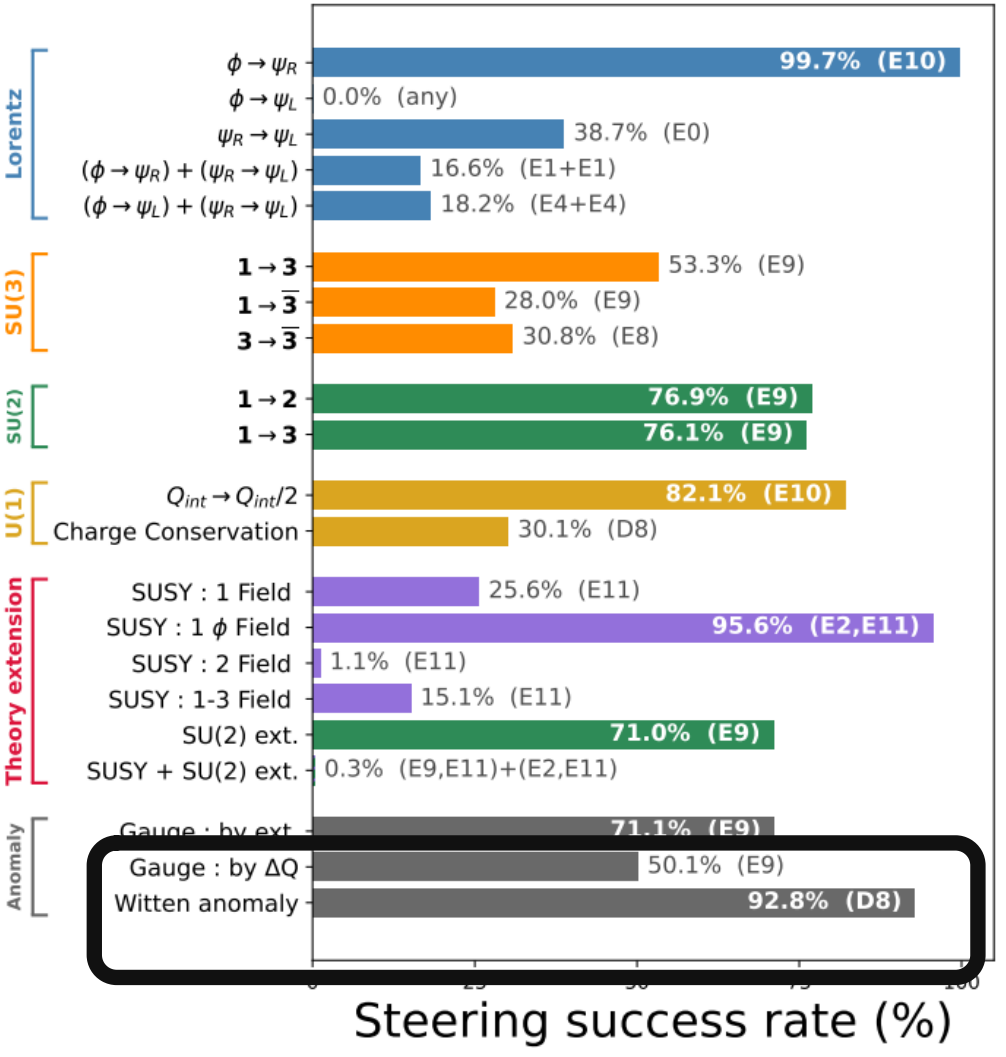
On going work!
Problem might be in Source and target data!

*Only working with trivial ACC solutions for now

Witten Anomaly

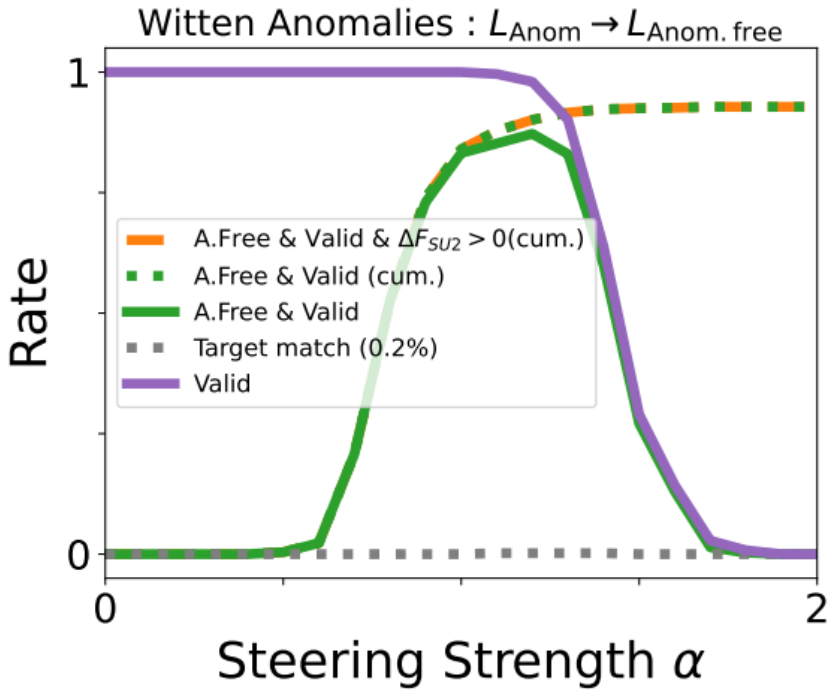
Number of SU2 doublet fermion must be even

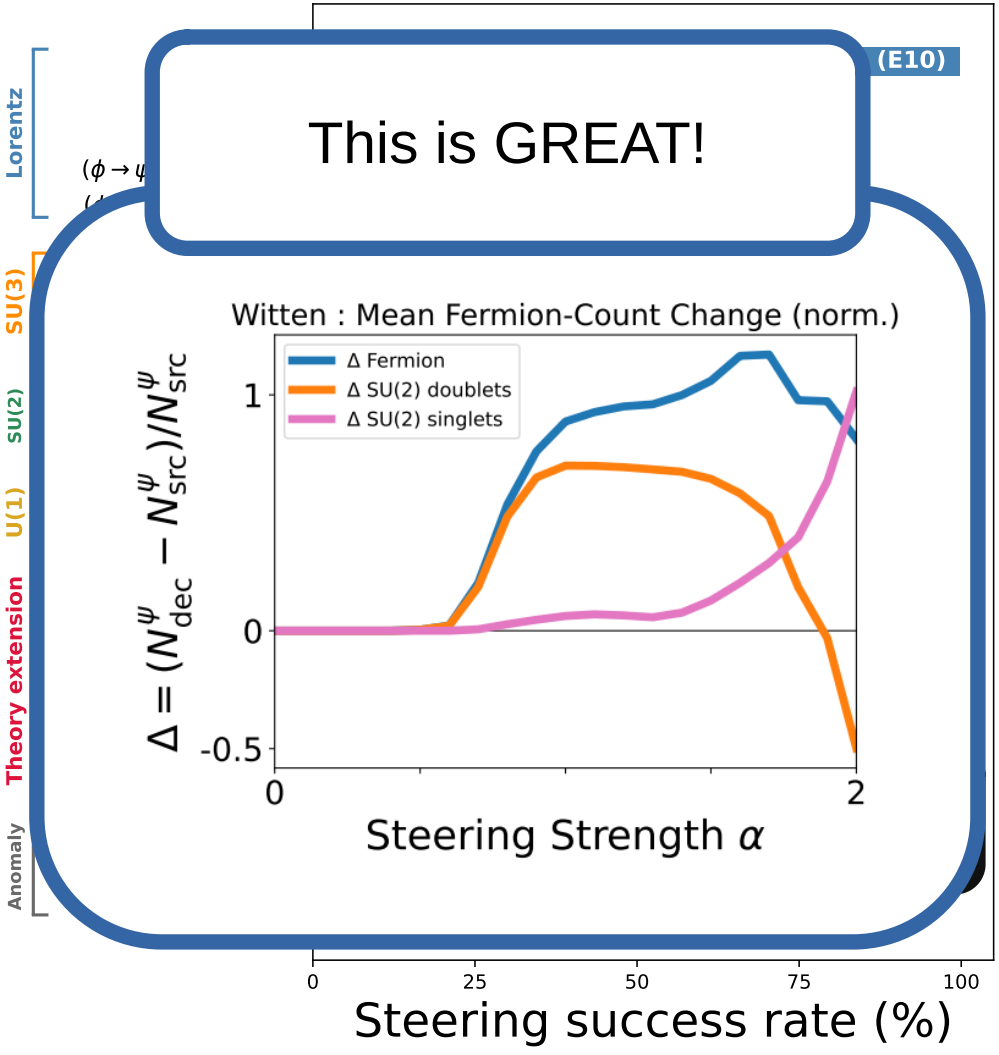




Witten Anomaly

Number of SU2 doublet fermion must be even





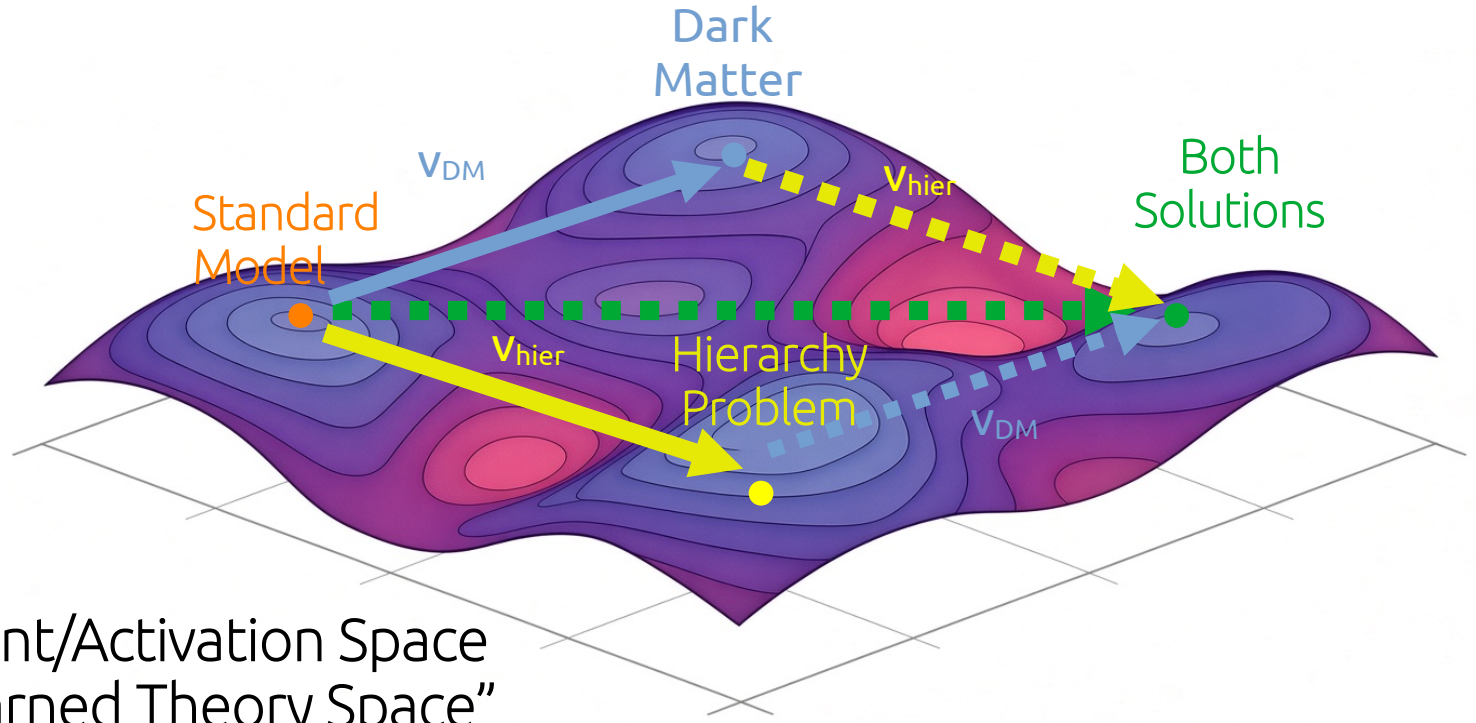
Conclusion

- BART-L writes Lagrangians!
- Latent Space ~ Learned Theory Space
- Steering
  - is cheap (Inference-time control)
  - Easier in Math than Language
  - works if model and data is good
  - Gap in Data ~ Gaps in Steerability
  - Can be additive
- Finally...

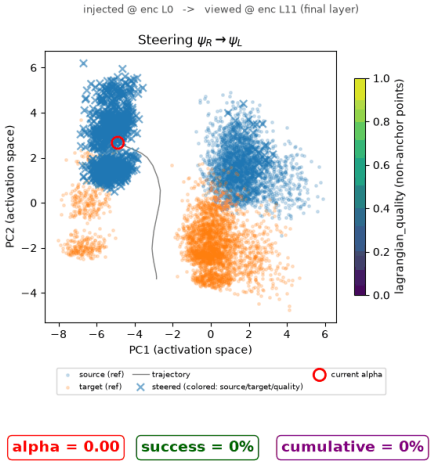
Outlook

Search For New Physics is now starting to be directional!

$h_{sm} \rightarrow h_{BSM} = h_{sm} + V_{DM} + V_{hier} + V_{\nu\text{-mass}} + \ldots$

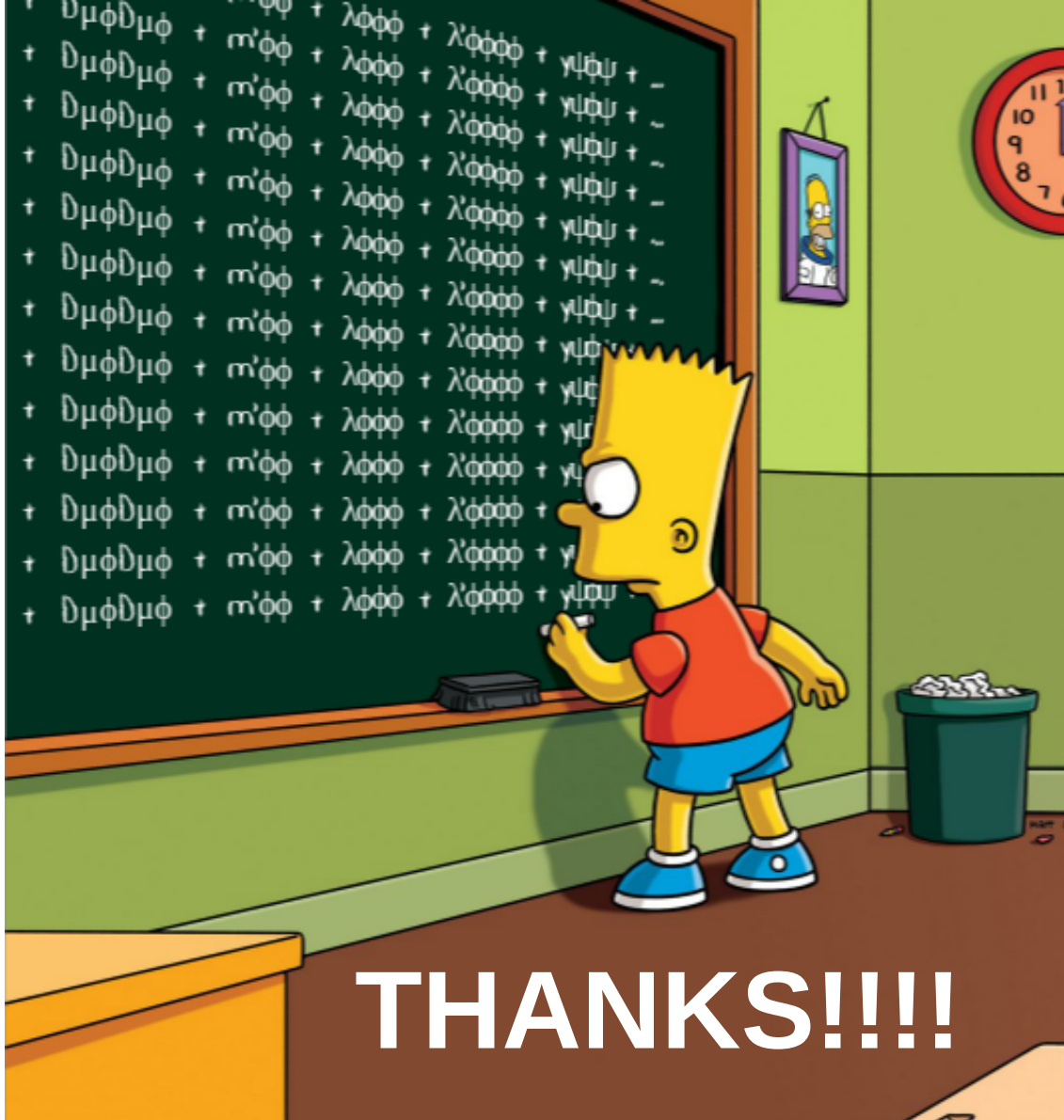


For more Steering GIFs:



Try BART-L yourself!

Generate some Lagrangians!



BACKUP

How do we tokenize it?

Table 2: Example of object-tokenization.

Object	Tokens
$\phi(\bar{\mathbf{3}}, \mathbf{2}, -1/3)$	FIELD, SPIN, 0, SU3, -, 3, SU2, 2, U1, -, 1, /, 3, ID3
$\psi_L(\bar{\mathbf{1}}, \mathbf{3}, 0)$	FIELD, SPIN, 1, /, 2, SU2, 3, HEL, 1, /, 2, ID9
$\phi^\dagger(\mathbf{1}, \mathbf{1}, 7/5)$	FIELD, SPIN, 0, U1, 7, /, 5, DAGGER, ID7
$D_\mu^{(\mathrm{SU}(2),\mathrm{U}(1))}$	DERIVATIVE, SU2, U1, ID4
$D_\mu^{(\mathrm{SU}(3),\mathrm{SU}(2),\mathrm{U}(1))}$	DERIVATIVE, SU3, SU2, U1, ID1
∂_μ	DERIVATIVE, ID5
$\overline{\sigma}^\mu$	SIGMA_BAR, ID2
$g^{\mu\nu}$	LORENTZ, ID6, ID4
ϵ^{ab}	SU2, ID0, ID7

akin to char-level tokenization rather than word-level tokenization

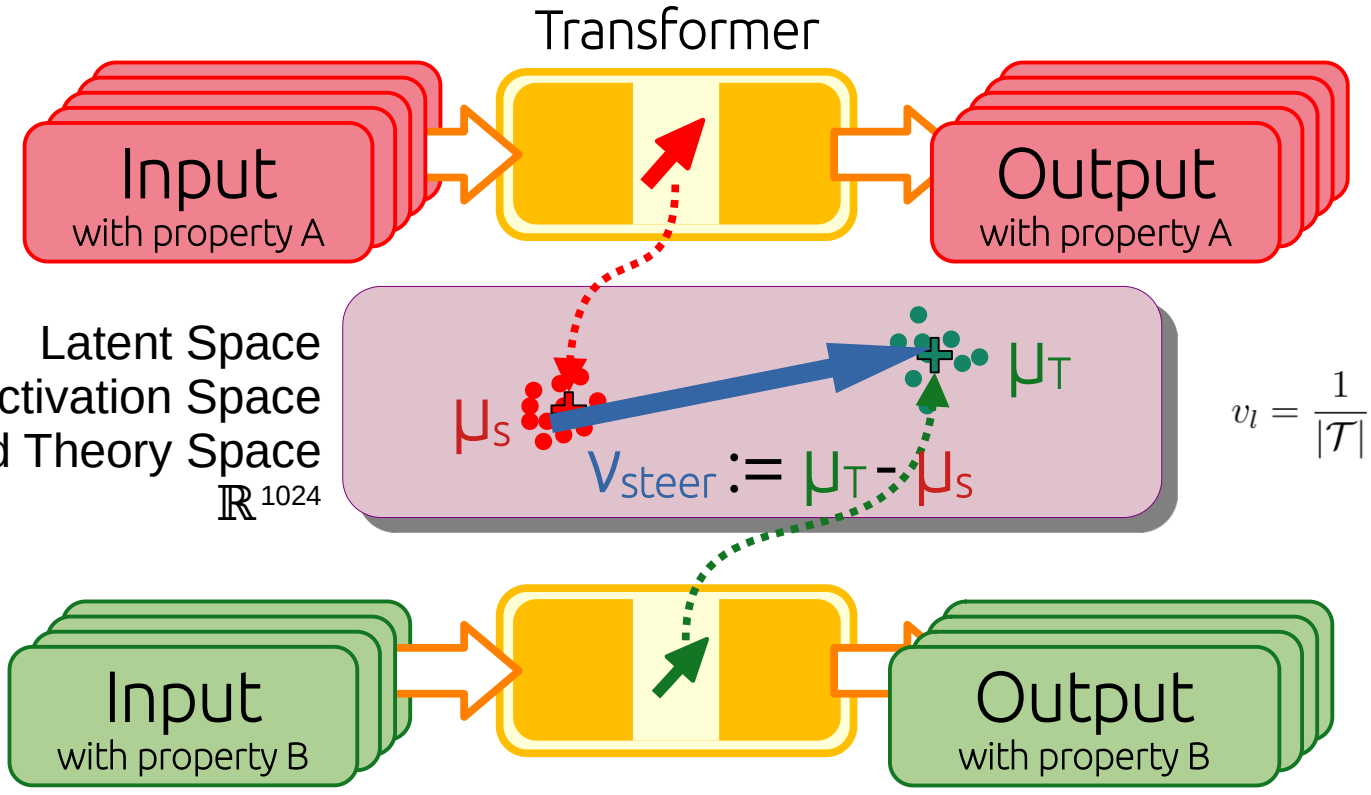
How do we tokenize it?

Example: Quartic Term

Term : $\lambda\, \phi\, \phi^\dagger\, \Phi\, \Phi^\dagger$

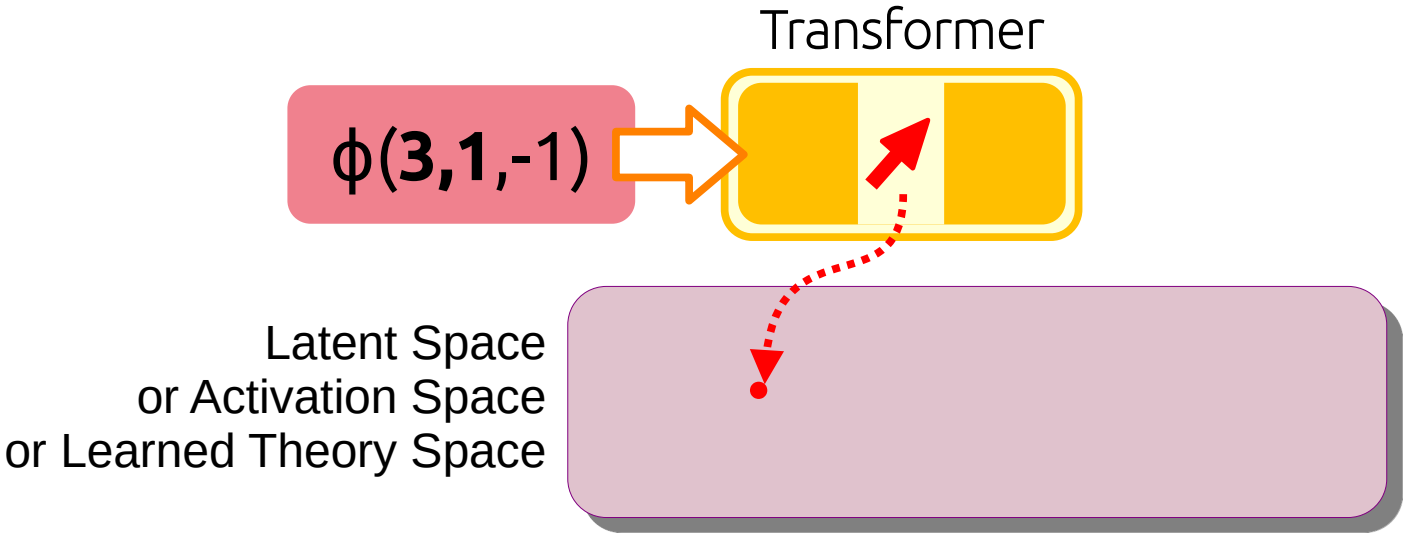
Information	Tokens
λ	
$\phi(\mathbf{3}, \mathbf{2}, 6)$	`FIELD, SPIN, 0, SU3, 3, SU2, 3, U1, 6, ID5`
$\phi^\dagger(\bar{\mathbf{3}}, \mathbf{2}, -6)$	`FIELD, SPIN, 0, SU3, -, 3, SU2, 3, U1, -, 6, DAGGER, ID7`
$\Phi(\bar{\mathbf{3}}, \mathbf{1}, -7/2)$	`FIELD, SPIN, 0, SU3, -, 3, U1, -, 7, /, 2, ID8`
$\Phi^\dagger(\mathbf{3}, \mathbf{1}, 7/2)$	`FIELD, SPIN, 0, SU3, 3, U1, 7, /, 2, DAGGER, ID2`
	`CONTRACTIONS`
$\epsilon^{a_1 b_2 c_2}\ \phi_{a_1}\ \phi^\dagger{}_{b_1 b_2}\ \Phi_{c_1 c_2}\ \Phi^\dagger{}_{d_1}$	`SU3, ID5, ID7, ID8`
$\epsilon^{b_1 c_1 d_1}\ \phi_{a_1}\ \phi^\dagger{}_{b_1 b_2}\ \Phi_{c_1 c_2}\ \Phi^\dagger{}_{d_1}$	`SU3, ID7, ID8, ID2`
$\epsilon^{a_1 b_1}\ \phi_{a_1 a_2}\ \phi^\dagger{}_{b_1 b_2}\ \Phi\ \Phi^\dagger$	`SU2, ID5, ID7`
$\epsilon^{a_2 b_2}\ \phi_{a_1 a_2}\ \phi^\dagger{}_{b_1 b_2}\ \Phi\ \Phi^\dagger$	`SU2, ID5, ID7`

Contrastive Activation Addition (CAA)

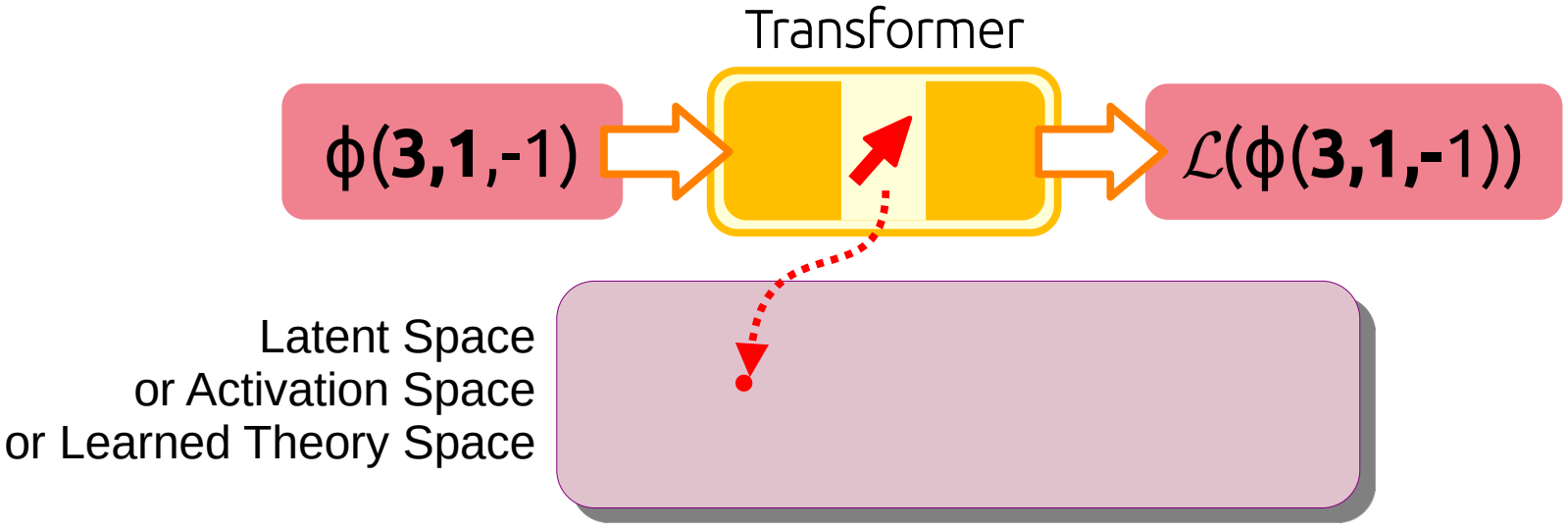


$$v_l = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \bar{h}_l^{(i)} - \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \bar{h}_l^{(i)}$$

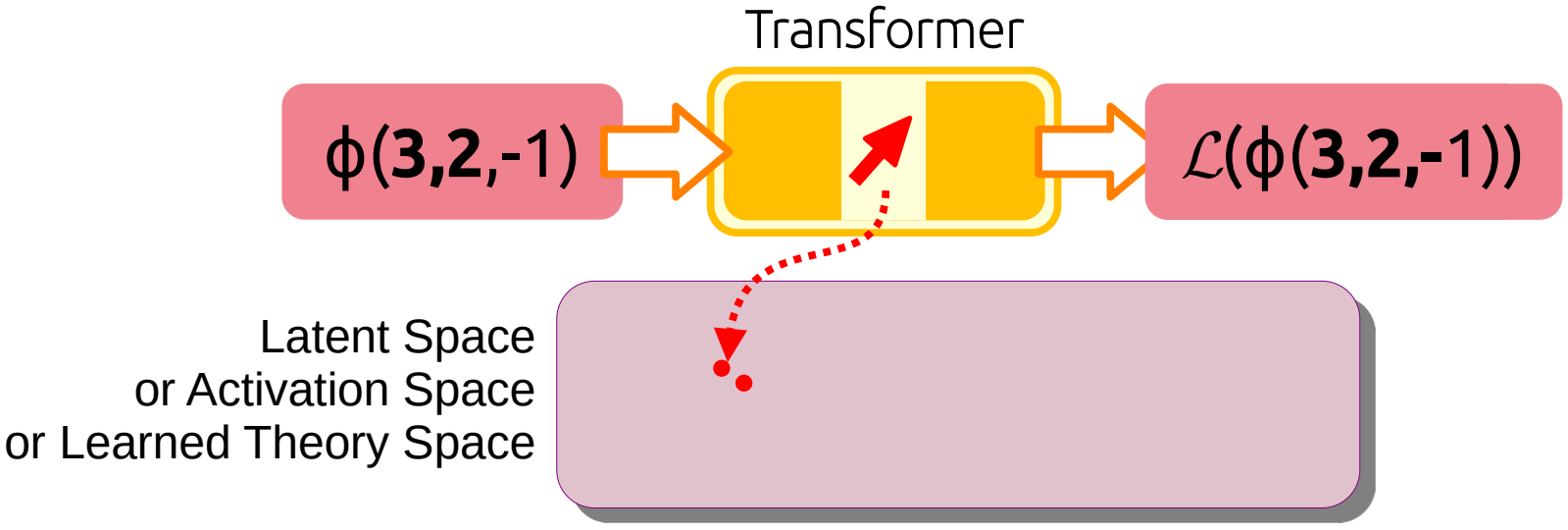
Contrastive Activation Addition (CAA)



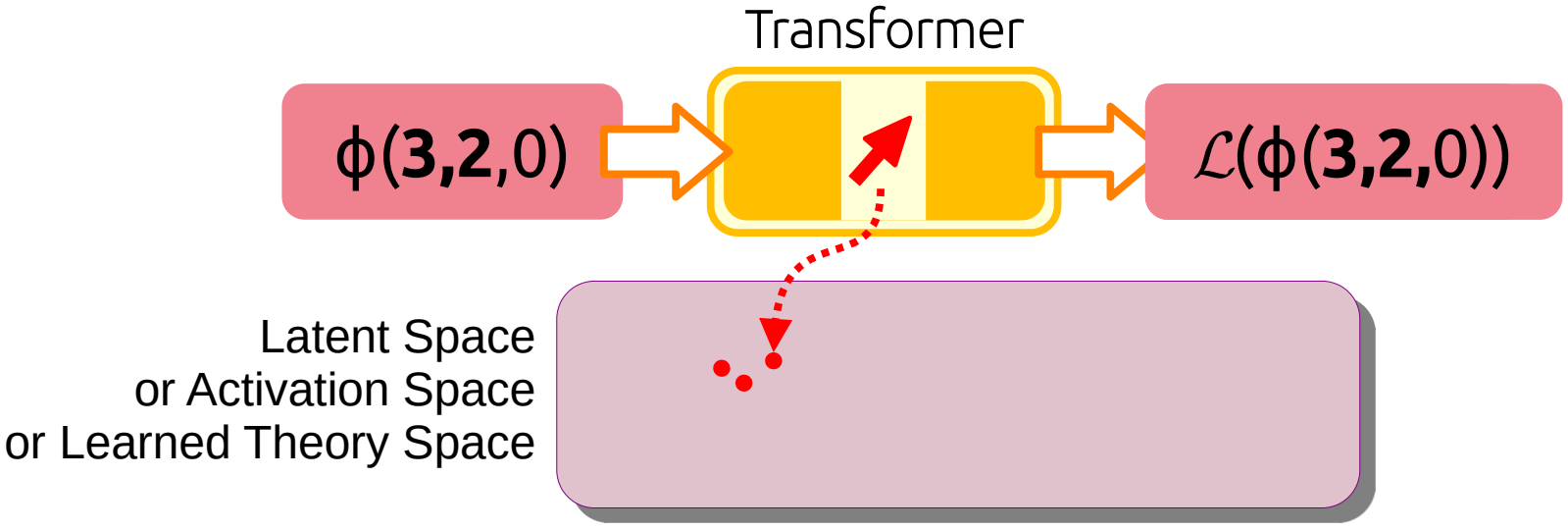
Contrastive Activation Addition (CAA)



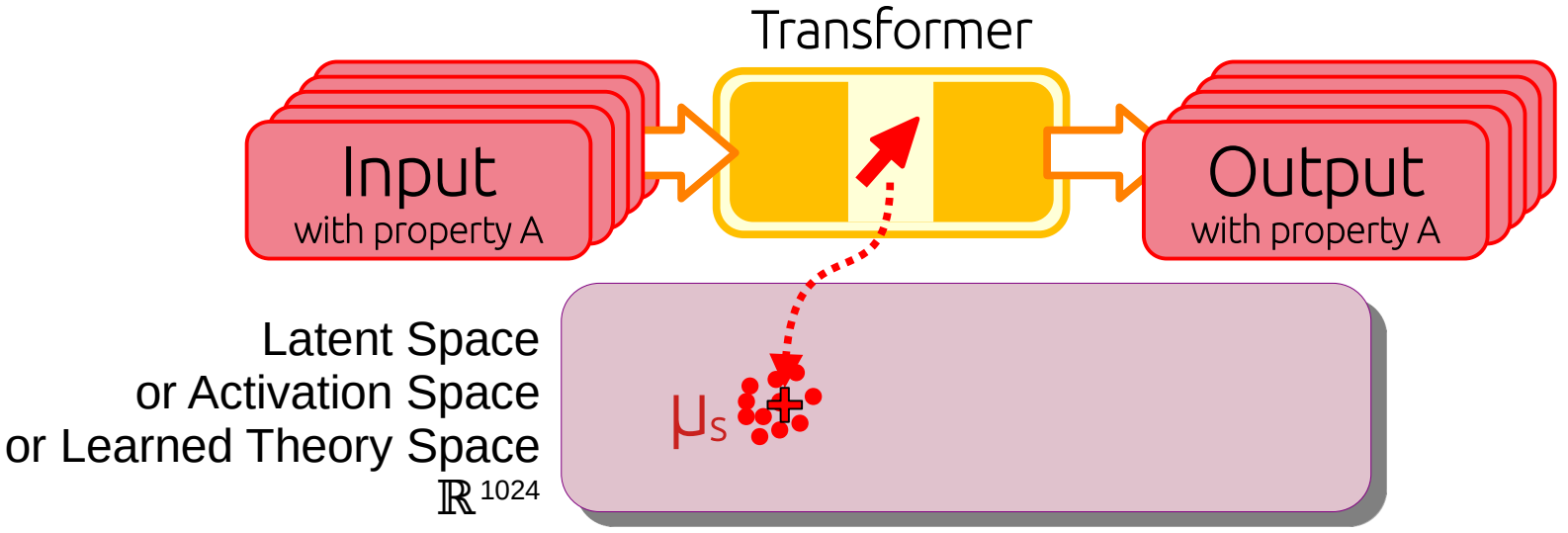
Contrastive Activation Addition (CAA)



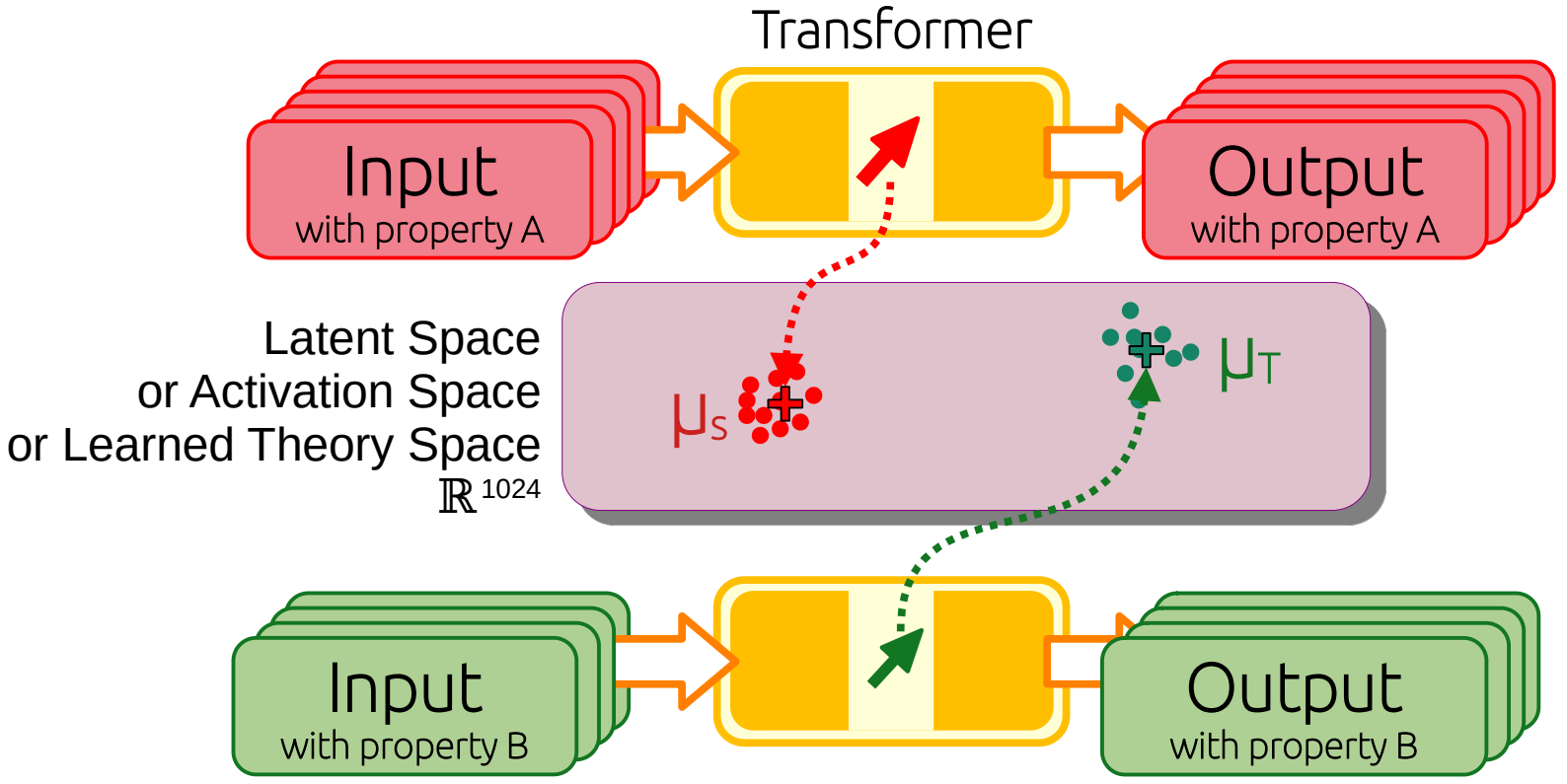
Contrastive Activation Addition (CAA)



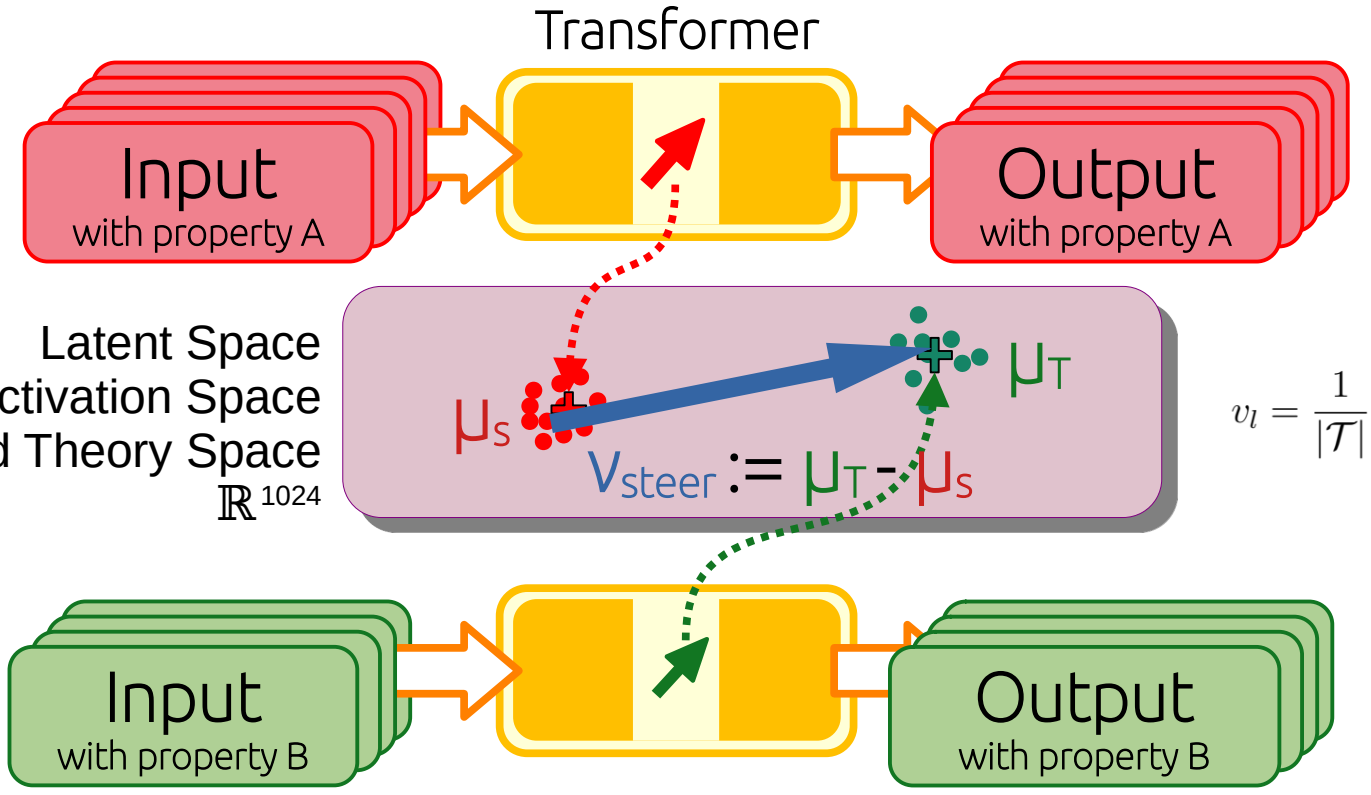
Contrastive Activation Addition (CAA)



Contrastive Activation Addition (CAA)



Contrastive Activation Addition (CAA)

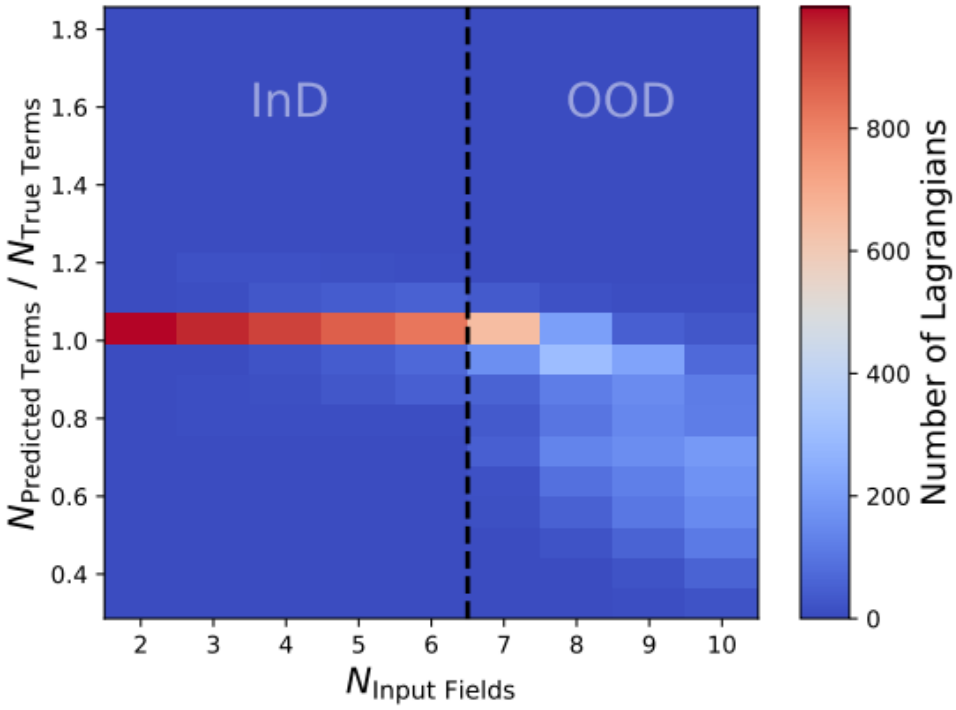


$$v_l = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \bar{h}_l^{(i)} - \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \bar{h}_l^{(i)}$$

OOD : N_{Fields} Generalization (or “Length Generalization”)

How do we check?

In Distribution (Train): ≤6 Fields
Out-of-Distribution (Test): 7-10 Fields

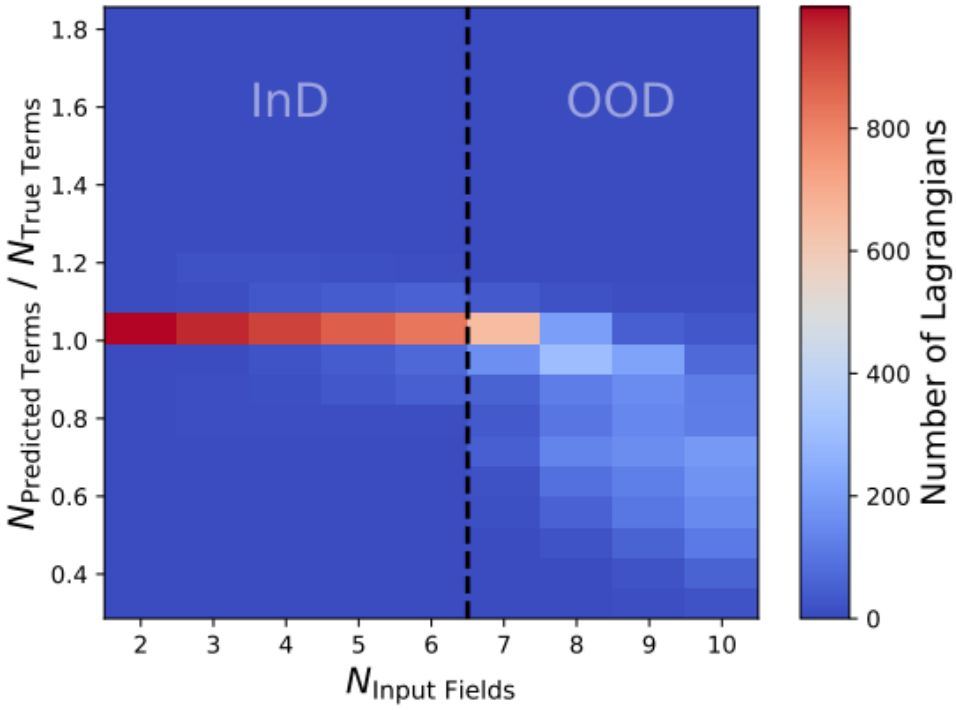


OOD : N_{Fields} Generalization (or “Length Generalization”)

How do we check?

In Distribution (Train): ≤6 Fields
Out-of-Distribution (Test): 7-10 Fields

At OOD:
1. Its missing terms

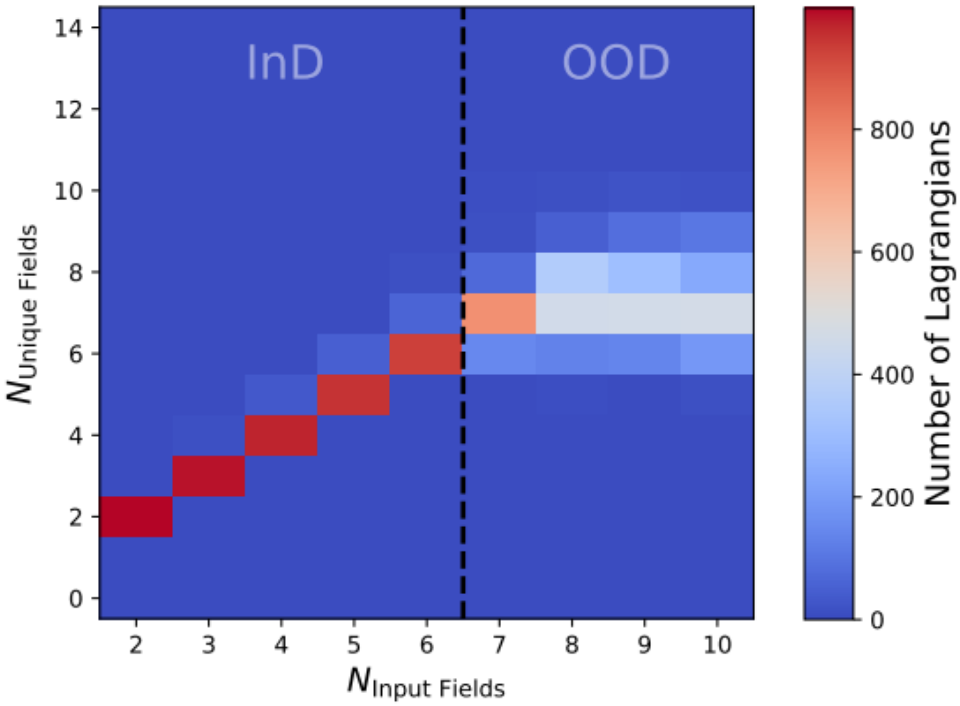


OOD : N_{Fields} Generalization (or “Length Generalization”)

How do we check?

In Distribution (Train): ≤6 Fields
Out-of-Distribution (Test): 7-10 Fields

At OOD:
1. Its missing terms.

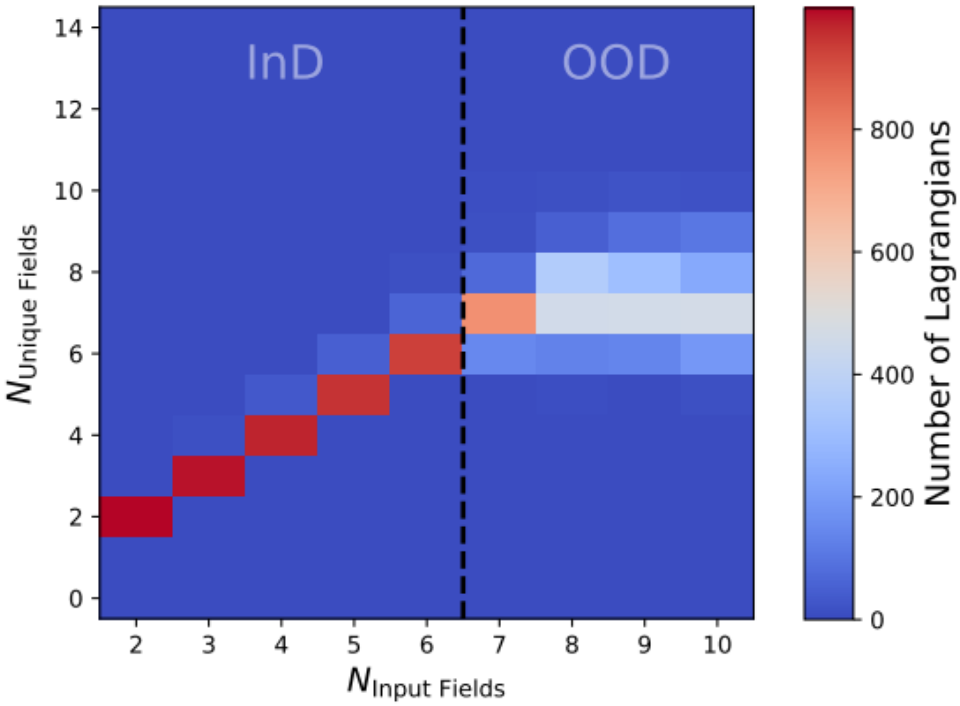


OOD : N_{Fields} Generalization (or “Length Generalization”)

How do we check?

In Distribution (Train): ≤6 Fields
Out-of-Distribution (Test): 7-10 Fields

At OOD:
1. Its missing terms.
2. It knows there should be more, but dont know how many

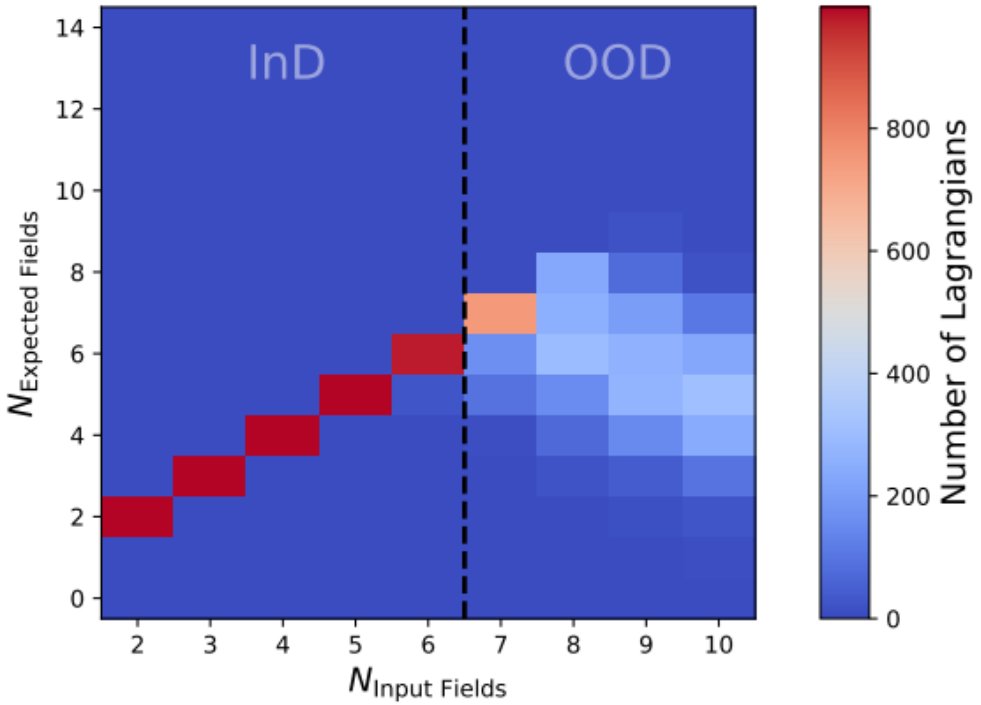


OOD : N_{Fields} Generalization (or “Length Generalization”)

How do we check?

In Distribution (Train): ≤6 Fields
Out-of-Distribution (Test): 7-10 Fields

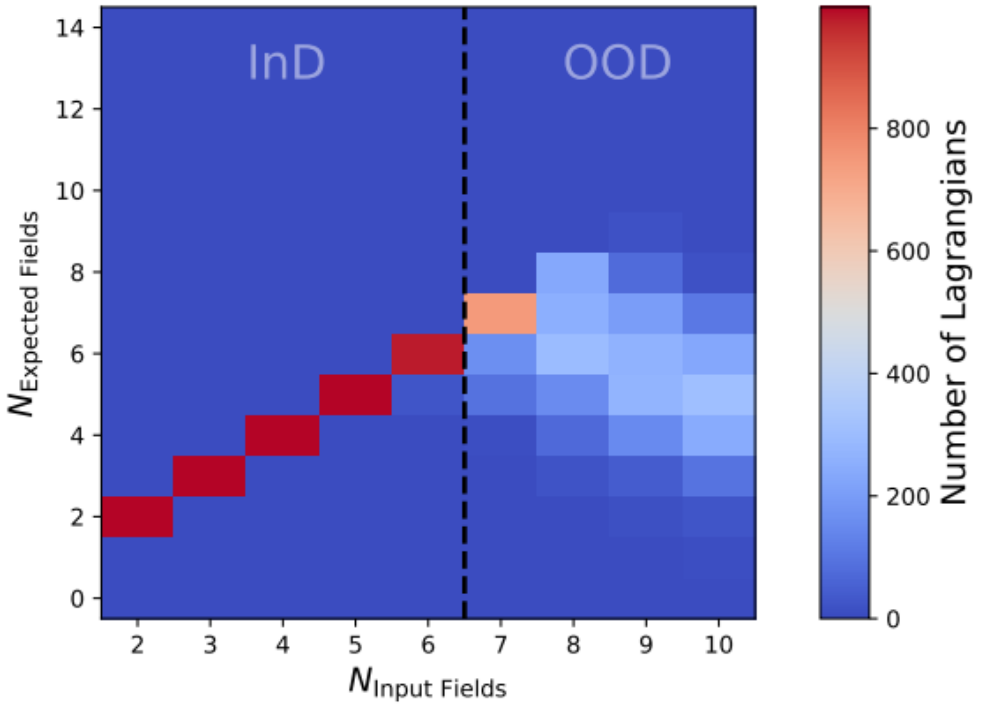
At OOD:
1. Its missing terms.
2. It knows there should be more but dont know how many



OOD : N_{Fields} Generalization (or “Length Generalization”)

How do we check?

In Distribution (Train): ≤6 Fields
Out-of-Distribution (Test): 7-10 Fields



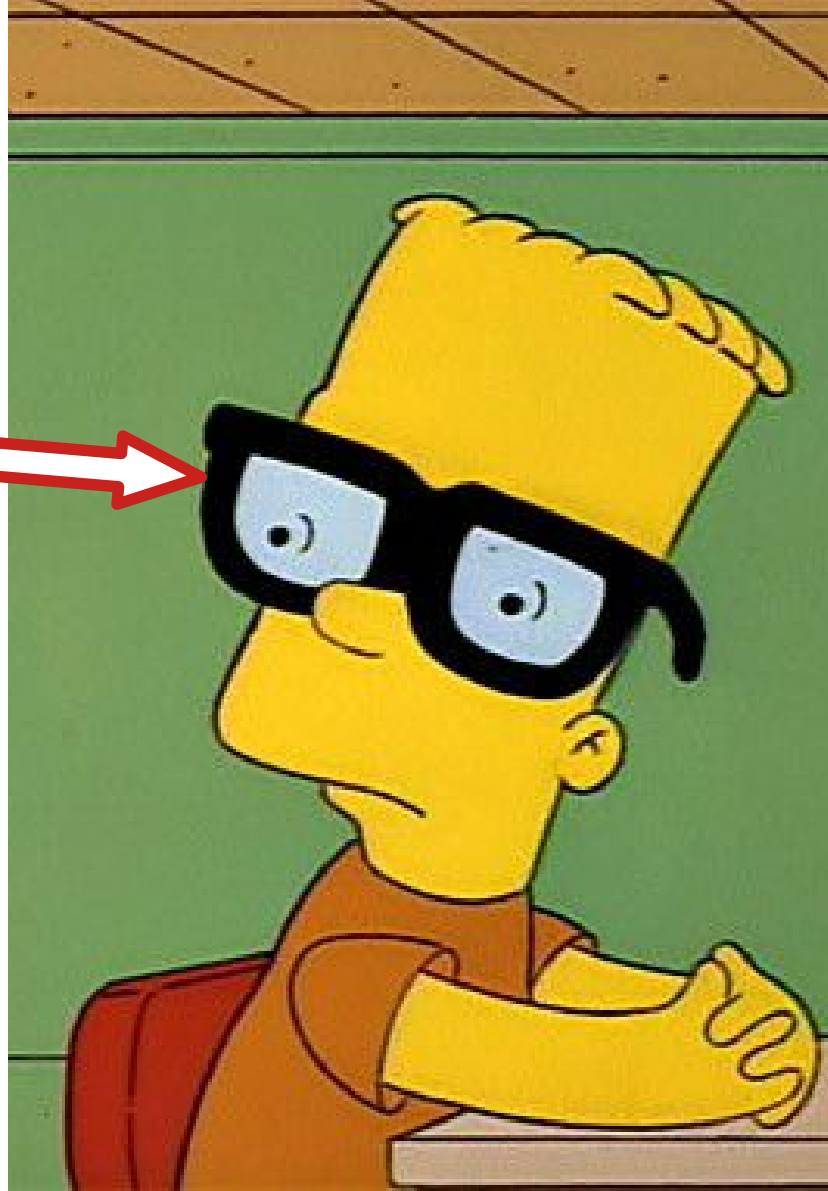
At OOD:
1. Its missing terms.
2. It knows there should be more but dont know how many
3. It jumbles up the inputs when there’s too much inputs

It can’t count!

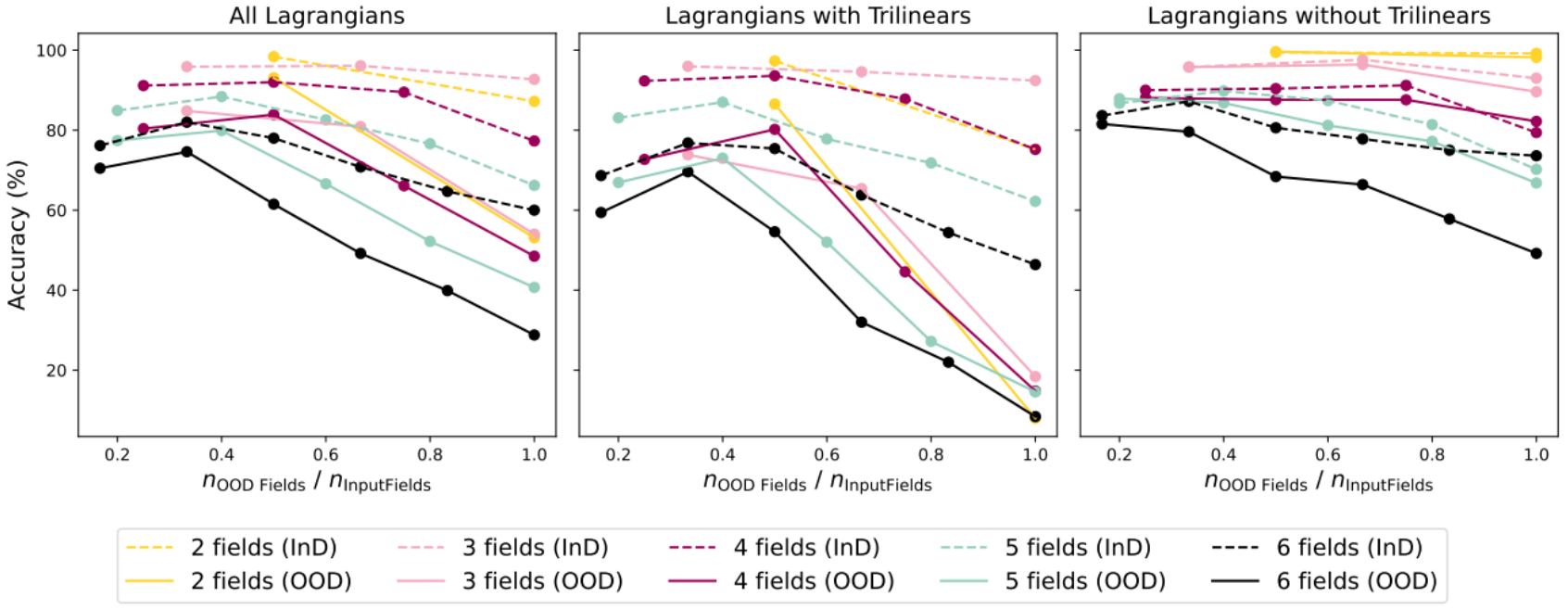
Non-Causal Encoder
with Abs. Positional Encoding
is bad for contextual counting
[2406.02585]

The Famous
“How many ‘r’ in the word ‘strawberry’?”
Problem

Ask ChatGPT to count the number of 1s in a random string of 1s and 0s. No python code.
It still fails sometime.



We also did OOD U(1) Charge Generalization



InD Charges(Train): ½ , 2, -7/2, ...
OOD Charges(Test): 2/4 , 6/2 , 3/9,...

Trilinears Terms:

$\Phi \quad \Phi' \quad \Phi''$

$+Q \quad +Q' \quad +Q'' = 0$

Need arithmetic

Quartic Terms:

$\Phi \quad \Phi^\dagger \quad \Phi' \quad \Phi'^\dagger$

$+Q \quad -Q \quad +Q' \quad -Q' = 0$

Pattern Recognition

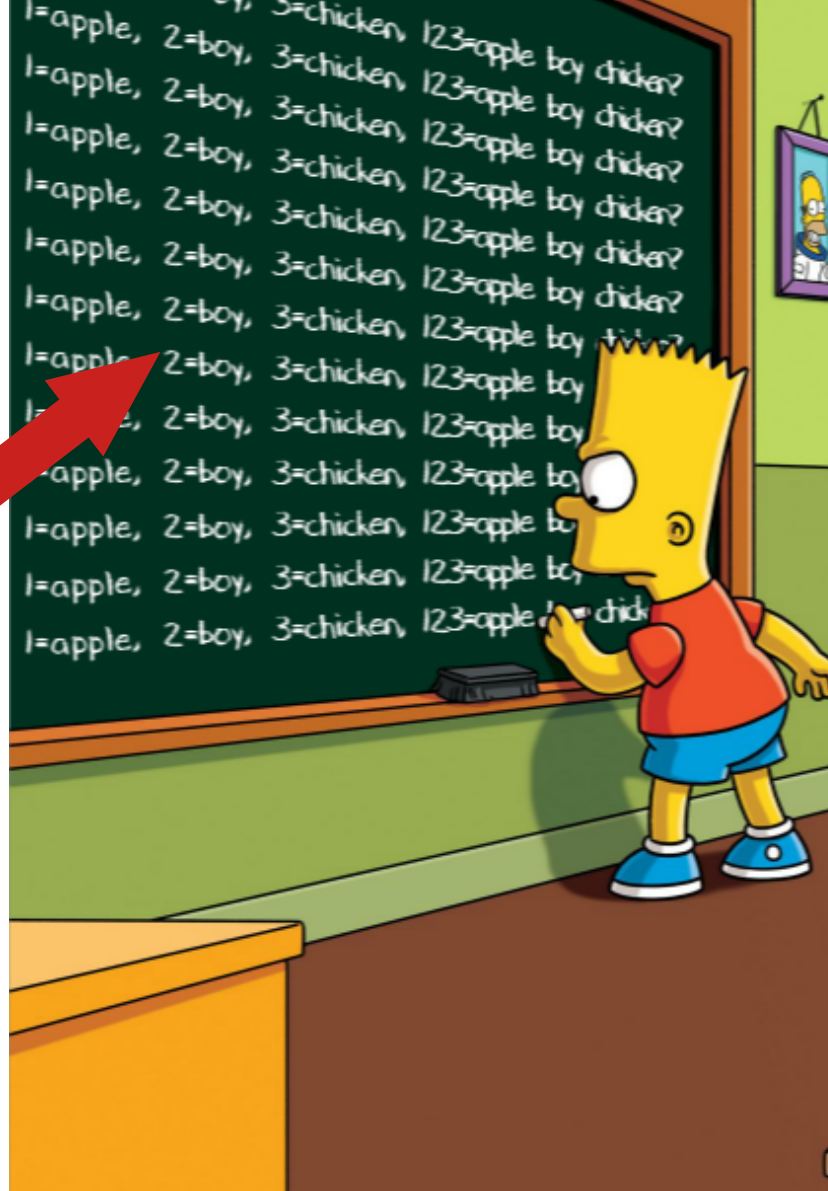
It cant do arithmetic!
It doesnt understand numbers!

Text-based encoding :
It is treating numbers
like alphabets

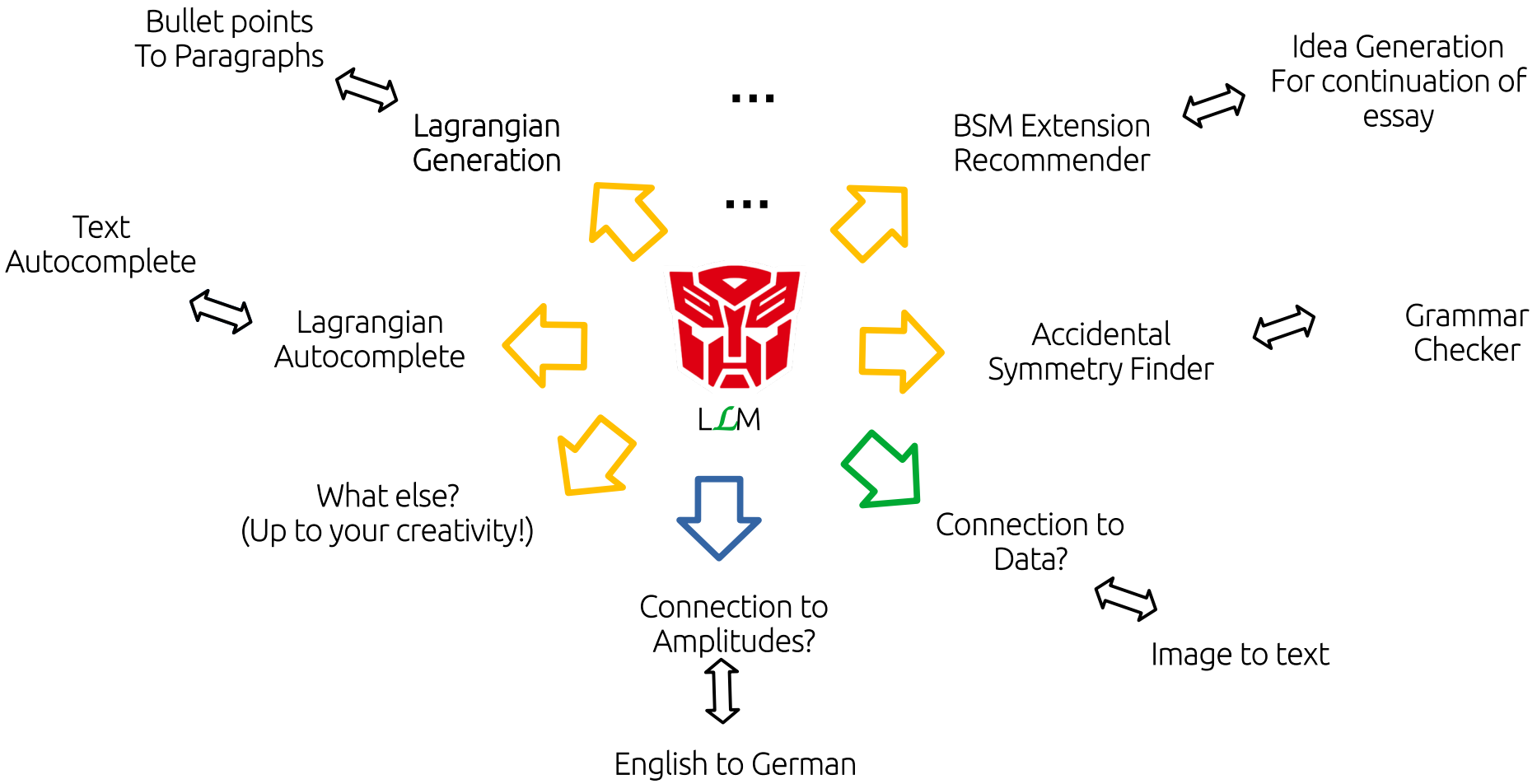
(Need continuous number tokenizing scheme)
[2310.02989]

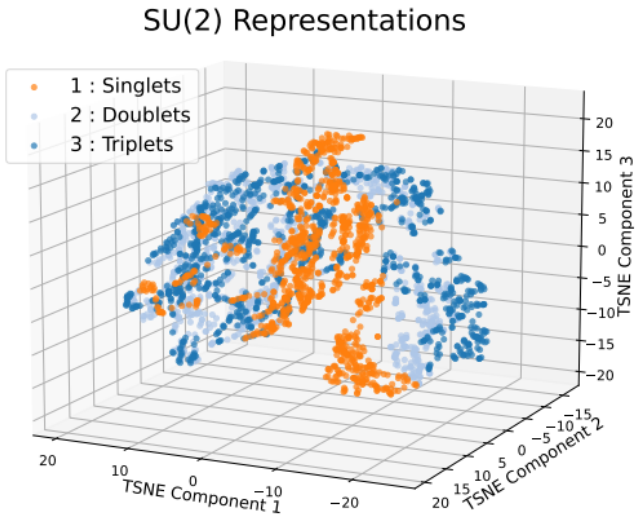
ChatGPT cant do arithmetic precisely without calculator or steps

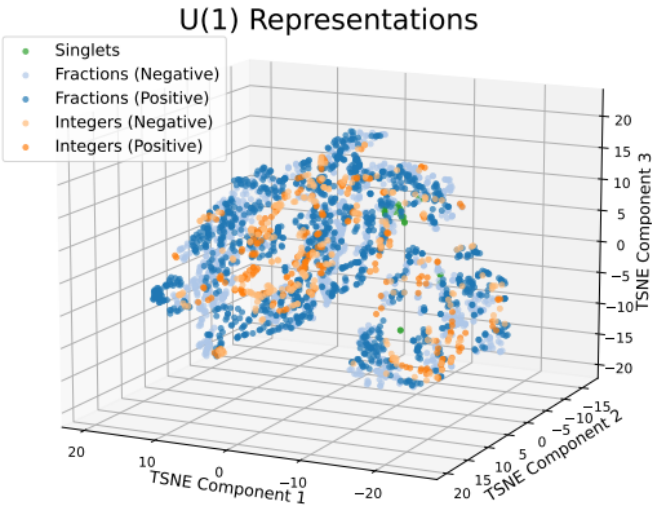
Ask ChatGPT to add two large numbers without calculator. Answer only without steps.
It fails sometimes.

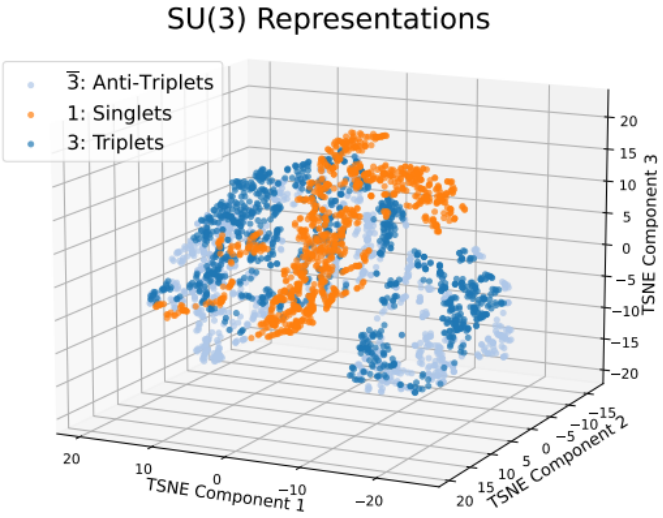


L$\mathcal{L}$M: Large "Lagrangian" Model?




SU(2) Representations
1 : Singlets
2 : Doublets
3 : Triplets
TSNE Component 1
TSNE Component 2
TSNE Component 3


U(1) Representations
Singlets
Fractions (Negative)
Fractions (Positive)
Integers (Negative)
Integers (Positive)
TSNE Component 1
TSNE Component 2
TSNE Component 3


SU(3) Representations
3̄: Anti-Triplets
1: Singlets
3: Triplets
TSNE Component 1
TSNE Component 2
TSNE Component 3