DrawPauliTransferMap

```
SetDirectory @ NotebookDirectory[];
Import["../Link/QuESTlink.m"];
```

Doc

```
? DrawPauliTransferMap
```

Symbol

DrawPauliTransferMap[map] visualises the given PTMap as a graph where nodes
are basis Pauli strings, and edges indicate the transformative action of the map.
DrawPauliTransferMap also accepts PTM, circuit and gate
instances, for which the corresponding PTMap is automatically calculated.
DrawPauliTransferMap accepts options "PauliStringForm", "ShowCoefficients"
and "EdgeDegreeStyles", in addition to all options accepted by Graph[].

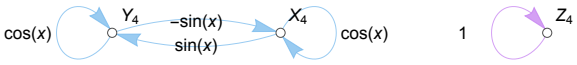
- "ShowCoefficients" –> False hides the map's
  Pauli string coefficients which are otherwise shown as edge labels.
- "PauliStringForm" sets the vertex label format to one of
  "Subscript" (default), "Index", "Kronecker", "String" or "Hidden". These (except
  the latter) are the formats are supported by GetPauliStringReformatted[].
- "EdgeDegreeStyles" specifies a list of styles (default informed by ColorData["Pastel"]) to set upon
  edges from nodes with increasing outdegree. For example, "EdgeDegreeStyles"–>{Red,Green,Blue}
  sets edges from Pauli states which are mapped to a single other state to the colour
  Red, but two–outdegree node out–edges become Green, and three–outdegree
  become Blue. The list is assumed repeated for higher outdegree nodes than specified.
- Graph[] options override these settings, so specifying EdgeStyle
  –> Black will set all edges to Black regardless of their node's outdegree.

Correctness

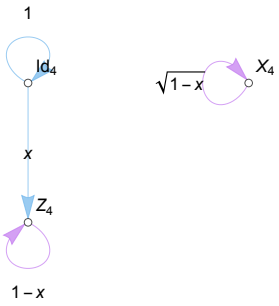
PTMap

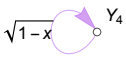
```
map = CalcPauliTransferMap @ Rz4[x]
DrawPauliTransferMap[map]
```

```
PTMap4[0 → {{0, 1}}, 1 → {{1, Cos[x]}, {2, Sin[x]}},
 2 → {{1, -Sin[x]}, {2, Cos[x]}}, 3 → {{3, 1}}]
```

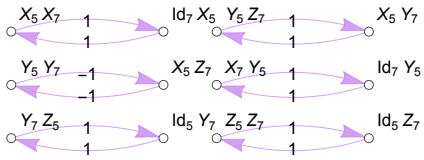



```
DrawPauliTransferMap @ CalcPauliTransferMap @ Damp4[x]
```

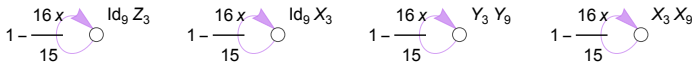


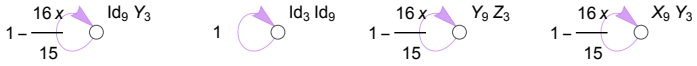


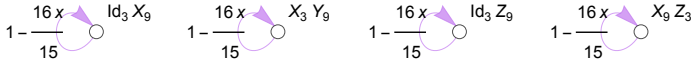
DrawPauliTransferMap @ CalcPauliTransferMap @ $C_5[X_7]$

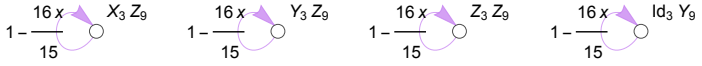


DrawPauliTransferMap @ CalcPauliTransferMap @ $Depol_{3,9}[x]$

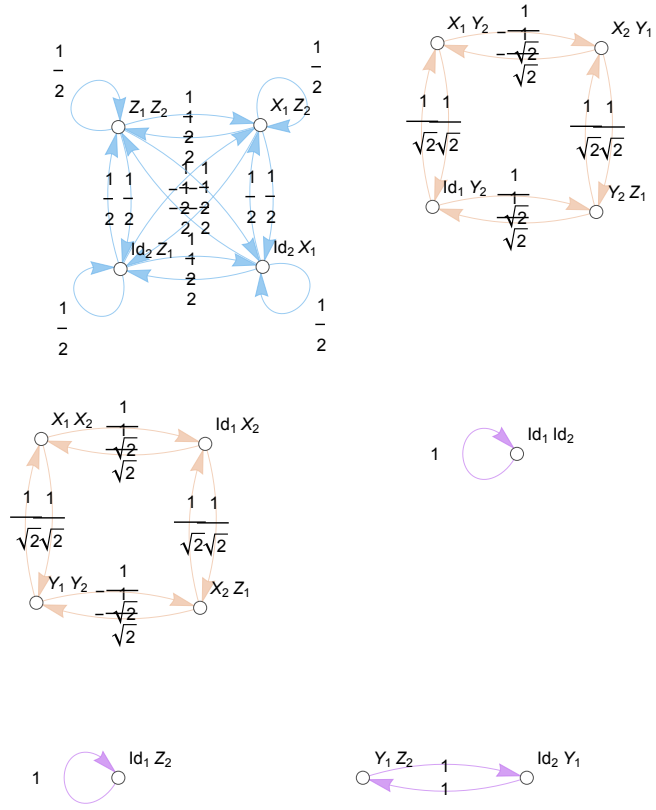






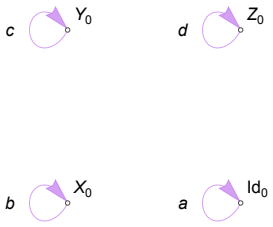


DrawPauliTransferMap @ CalcPauliTransferMap @ $C_2[H_1]$

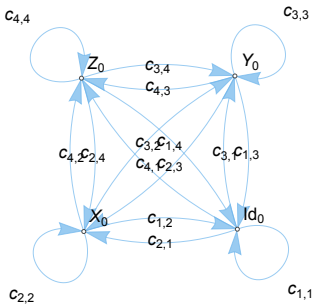


PTM, gates, circuits

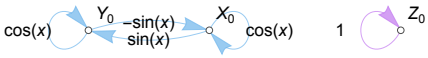
DrawPauliTransferMap @ PTM_0 @ DiagonalMatrix[{a, b, c, d}]



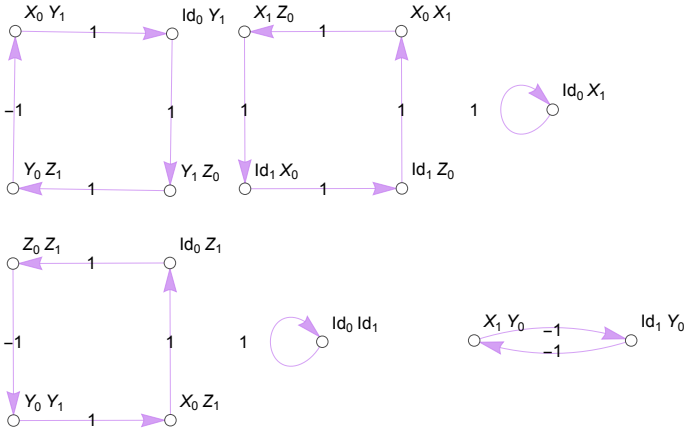
DrawPauliTransferMap @ PTM_0 @ Table[$c_{i,j}$, {i, 4}, {j, 4}]



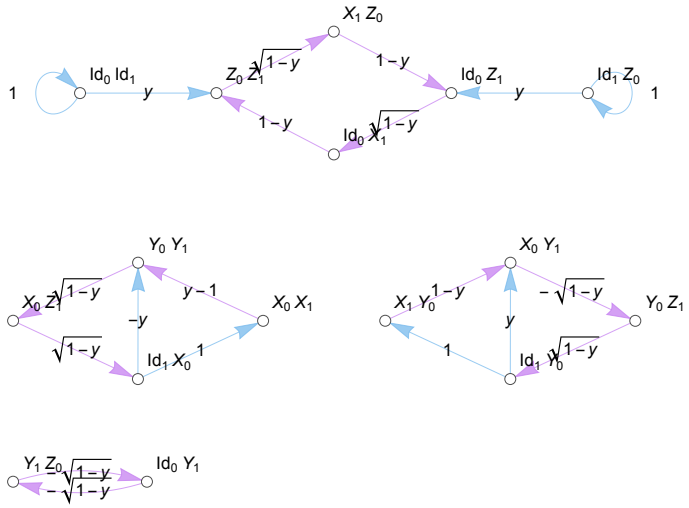
DrawPauliTransferMap @ Rz$_0$[x]



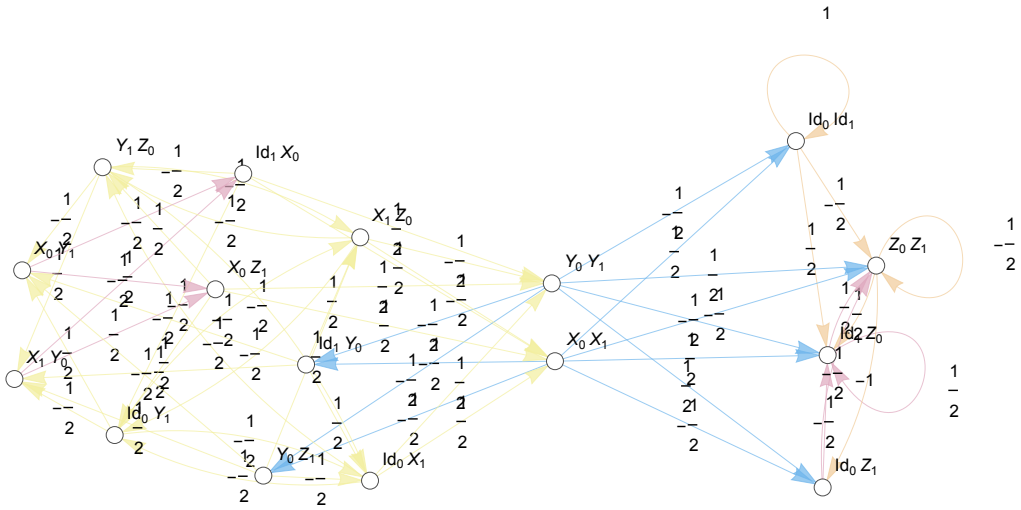
DrawPauliTransferMap @ Circuit[H$_0$ C$_0$[X$_1$]]



DrawPauliTransferMap @ Circuit[H$_1$ Damp$_1$[y] C$_0$[X$_1$]]

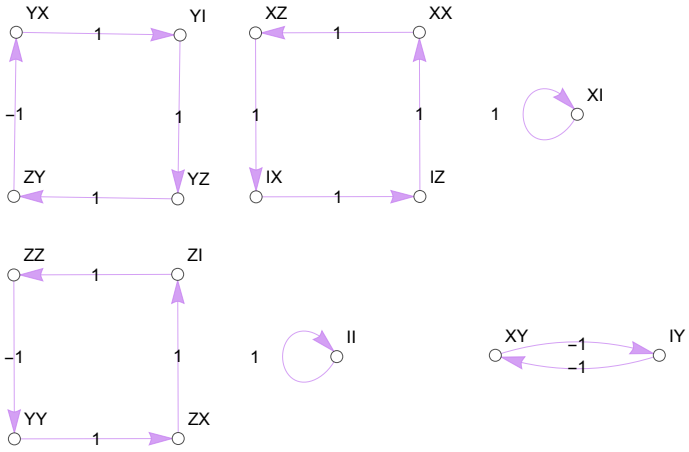


```
DrawPauliTransferMap[U0,1 @ ( 0 0 i 0 )
                             ( 0 1 0 0 )
                             ( 1 0 0 1 )
                             ( 0 0 0 0 )]
```

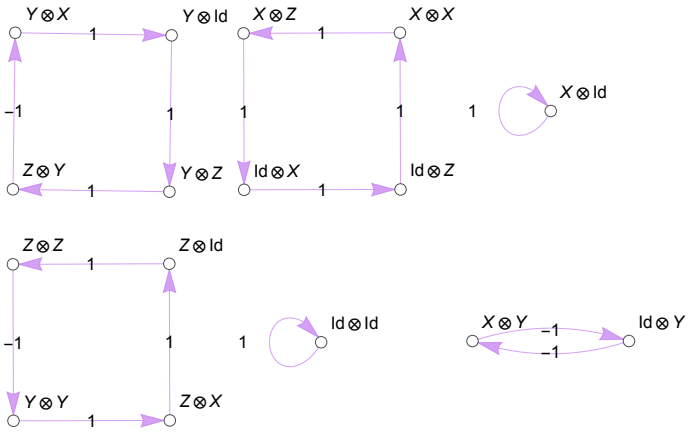


option “PauliStringForm”

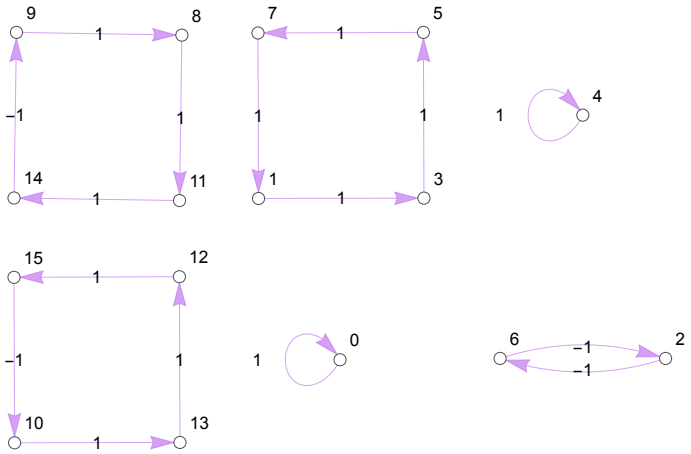
```
circ = Circuit[H0 C0[X1]];

DrawPauliTransferMap[circ, "PauliStringForm" → "String"]
```



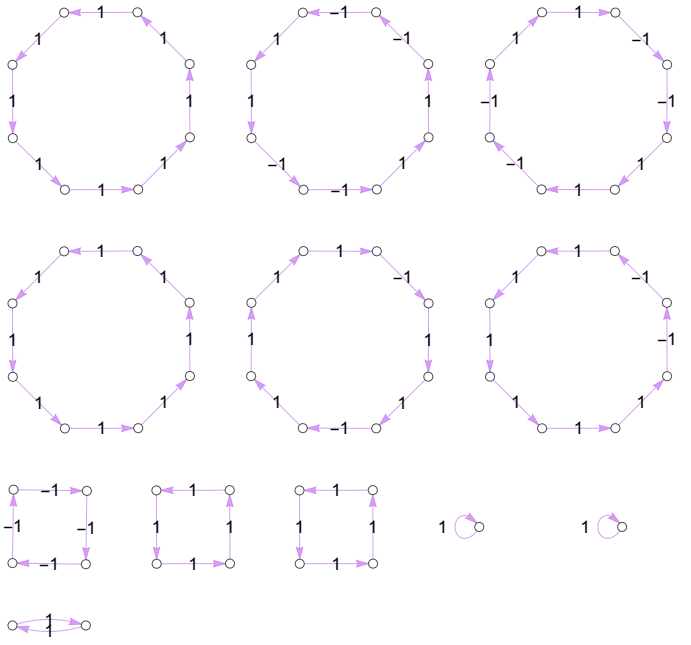
```
DrawPauliTransferMap[circ, "PauliStringForm" → "Kronecker"]
```




```
DrawPauliTransferMap[circ, "PauliStringForm" → "Index"]
```

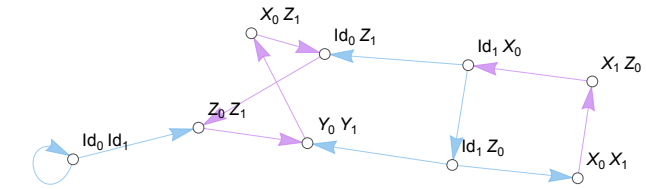


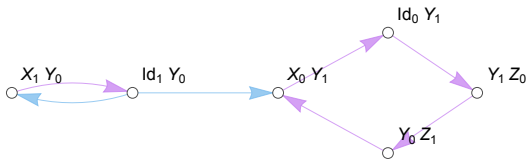

```
circ = Circuit[H0 C0[X1] SWAP1,2];
DrawPauliTransferMap[circ, "PauliStringForm" → "Hidden"]
```

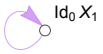


option “ShowCoefficients”

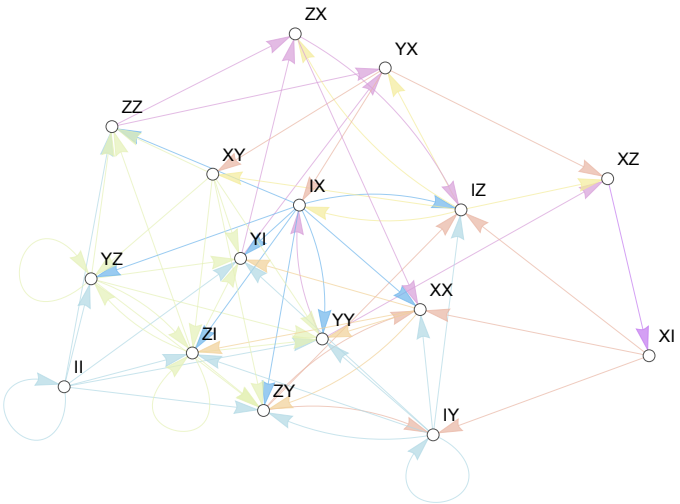
```
circ = Circuit[H0 Depol0,1[x] Damp1[y] C0[X1]];
DrawPauliTransferMap[circ, "ShowCoefficients" → False]
```



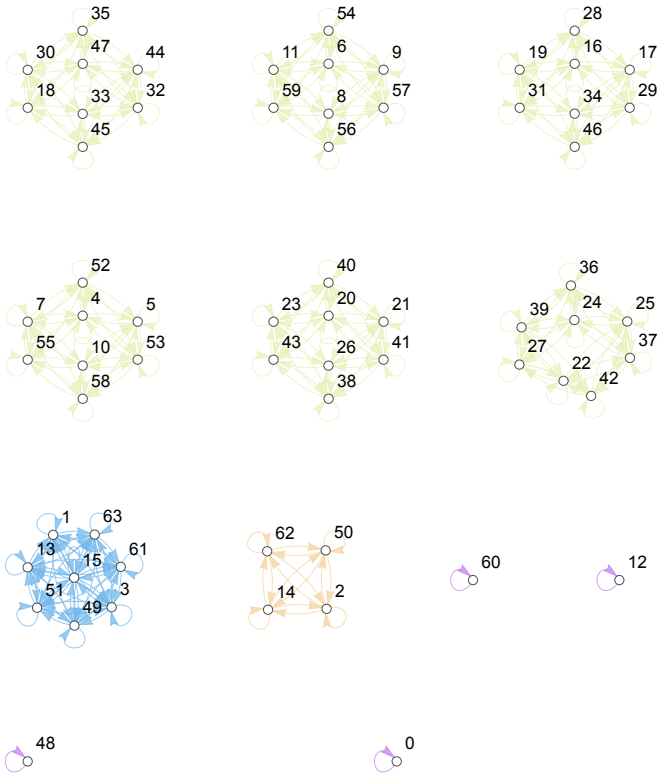




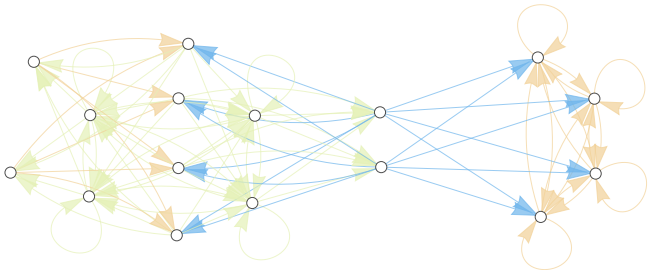

```
circ = Circuit[H1 Damp0[x] Depol0,1[x] C0[X1] C1[Rx0[x]] R[x, X0 Y1]];
DrawPauliTransferMap[circ,
      "ShowCoefficients" → False,
      "PauliStringForm" → "String"]
```



```
DrawPauliTransferMap[C0,2[H1],
    "ShowCoefficients" → False,
    "PauliStringForm" → "Index"]
```

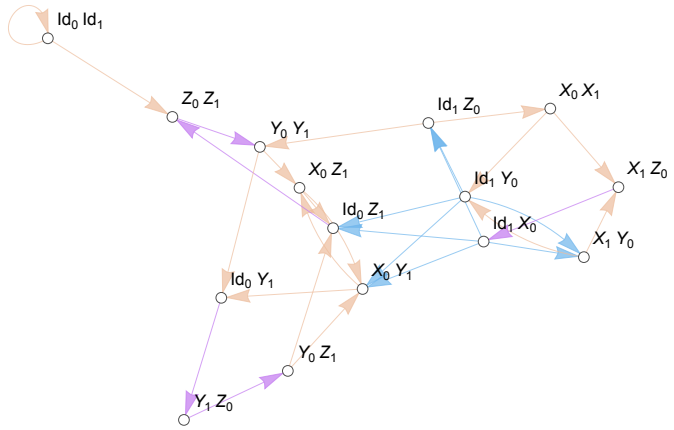



```
DrawPauliTransferMap[U0,1 @ ⎛0 0 a 0⎞,
                           ⎜0 1 0 0⎟
                           ⎜1 0 0 1⎟
                           ⎝0 0 0 0⎠
    "ShowCoefficients" → False, "PauliStringForm" → "Hidden"]
```



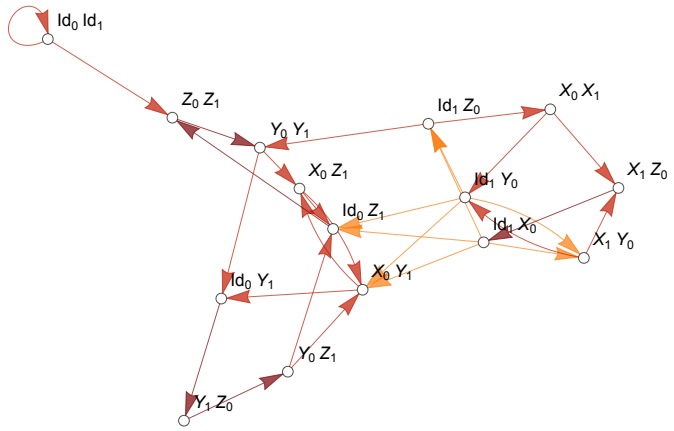
option “EdgeDegreeStyles”

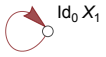
```
circ = Circuit[H0 Depol0,1[x] Rx0[a] Damp1[y] C0[X1]];
DrawPauliTransferMap[circ, "ShowCoefficients" → False]
```



```
colors = ColorData["SolarColors"] /@ Range[0, 1, .2]

DrawPauliTransferMap[circ,
      "ShowCoefficients" → False,
      "EdgeDegreeStyles" → colors]
```

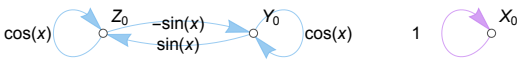
{■, ■, ■, ■, ■, ■}





option AssertValidChannels

```
DrawPauliTransferMap[Rx0[x], AssertValidChannels → True]
```



```
DrawPauliTransferMap[Rx0[x], AssertValidChannels → False]
```

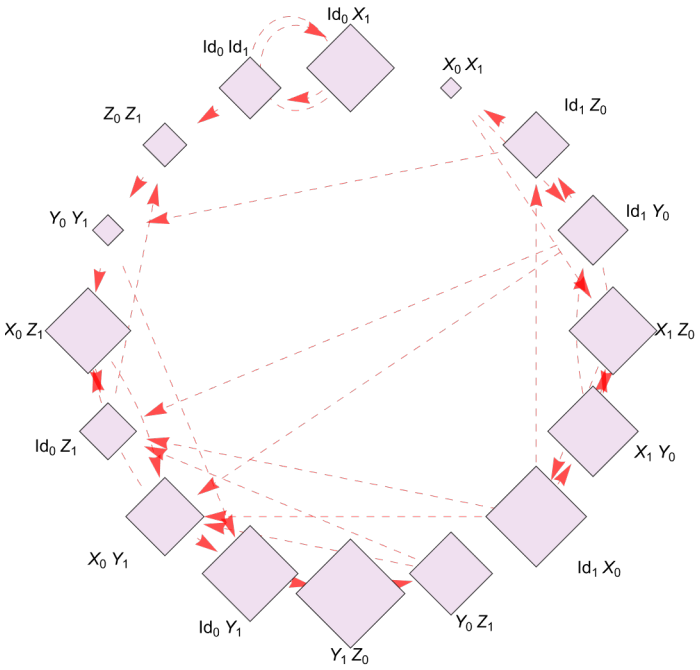
$$\frac{1}{2}\left(2\cos^2\left(\frac{x}{2}\right)-2\sin^2\left(\frac{x}{2}\right)\left(\mathrm{Id}_0+\mathrm{Id}_1+X_0 X_1+Y_0 Y_1+Z_0 Z_1\right)\right)$$

$$\frac{1}{2}\left(2\sin^2\left(\frac{x}{2}\right)+2\cos^2\left(\frac{x}{2}\right)\left(\mathrm{Id}_0+\mathrm{Id}_1\right)\right)$$

---

options of Graph

```
circ = Circuit[H0 Depol0,1[x] Rx0[a] Damp1[y] C0[X1]];
DrawPauliTransferMap[circ, "ShowCoefficients" → False,

    EdgeStyle → Directive[Red, Dashed],
    VertexShapeFunction → "Diamond",
    VertexSize :→ RandomReal[],
    VertexStyle → LightPurple,
    GraphLayout → "CircularEmbedding"
]
```



Errors

DrawPauliTransferMap[CalcPauliTransferMap[X_0], "BadOption" → "bad"]

OptionValue : Unknown option BadOption for {DrawPauliTransferMap, Graph}.

$Failed

DrawPauliTransferMap[CalcPauliTransferMap[X_0], "PauliStringForm" → "bad"]

DrawPauliTransferMap : Unrecognised value for option "PauliStringForm". See ?DrawPauliTransferMap

$Failed

DrawPauliTransferMap[$PTMap_{0,-1}$[x]]

DrawPauliTransferMap : Failed to automatically obtain the PTMap due to the below error:

CalcPauliTransferMatrix : Circuit contained an unrecognised or unsupported gate: $PTMap_{0,1}$[x]

$Failed

DrawPauliTransferMap[ ]

DrawPauliTransferMap : Invalid arguments. See ?DrawPauliTransferMap

$Failed

DrawPauliTransferMap @ Bad_0

DrawPauliTransferMap : Failed to automatically obtain the PTMap due to the below error:

CalcPauliTransferMatrix : Circuit contained an unrecognised or unsupported gate: Bad_0

$Failed