# Lattice Models

## 4-Qubit gas model with one qubit temperature variation

Import: File /Users/uja5020/Desktop//DensityMatrixTools.wl not found during Import.

```
In[•]:= RandomOrder[] := Module[{items, vals},
          items = {Q1, Q2, Q3, Q4};
          vals = RandomSample[Table[i, {i, 4}]];
          Association[Table[items[[i]] -> vals[[i]], {i, 4}]]
         ]
        RandomUnitary := Module[{SubsysInteraction, Hi1, interactionUnitary},
          SubsysInteraction = RandomHamiltonian[1, 4];
          Hi1 = SubsysInteraction;
          interactionUnitary =  SparseArray[MatrixExp[I (Hi1) .1]]
         ]

In[•]:= ρi = NThermalQBit[{.2, .4, .2, .2}];
        ρ = ρi;
        baseOrder = ρ[qbitIDs];
        interactionUnitary = RandomUnitary;
        DMS = Table[
            RO = RandomOrder[];
            Ui = MakeDM[interactionUnitary, RO];
            ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
            Round[ReOrder[ρ, baseOrder], 10^-15], {i, 500}];

In[•]:= AmbientTemps = Table[
            Association[Table[
              qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
            ,
            {P, DMS}];

In[•]:=
        AmbientPops = Table[
            Association[Table[
              qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
            {Temps, AmbientTemps}];
```

```
In[•]:= ExtractableWorkOverTime = N[Module[{P},
          Table[
           P = DMS[[t]];
           Table[
            AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
               ThermalQBit[AmbientPops[[t]][qbit]]
             ],
            {qbit, Keys[P[qbitIDs]]}],
           {t, Length[DMS]}]
          ]];
```

```
In[•]:= ChangeinExtractablework =
          Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
           {i, Length[ExtractableWorkOverTime]}];
```

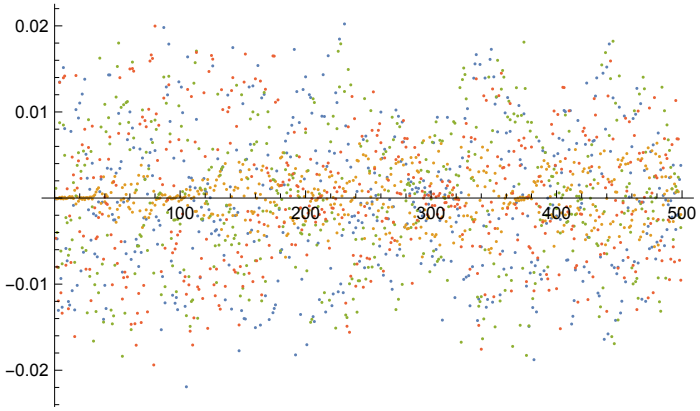
... Part: Part 501 of
{{0.158183, 0.000185235, 0.156946, 0.158249}, {0.158073, 0.00015991, 0.15807, 0.153573}, {0.153334, 0.0000650912, 0.154069, 0.145564}, {0.140288, $3.81416 \times 10^{-6}$, 0.146, 0.133614}, {0.153762, 0.0000774229, 0.154748, 0.147039}, {0.144416, $8.95218 \times 10^{-7}$, 0.148588, 0.136032}, ≪40≫, {0.119473, 0.00355932, 0.0671736, 0.114878}, {0.116922, 0.00548727, 0.0534862, 0.11674}, {0.114392, 0.00799437, 0.0456805, 0.110898}, {0.11729, 0.00517993, 0.0574103, 0.112375}, ≪450≫} does not exist.

```
In[•]:= ChangeinExtractablework[[2]]
```

```
Out[•]= If[{0.0012294, 0.0000267392, 0.0026637, 0.0000107401} > 0,
          ExtractableWorkOverTime⟦i + 1⟧ - ExtractableWorkOverTime⟦i⟧, 0]
```

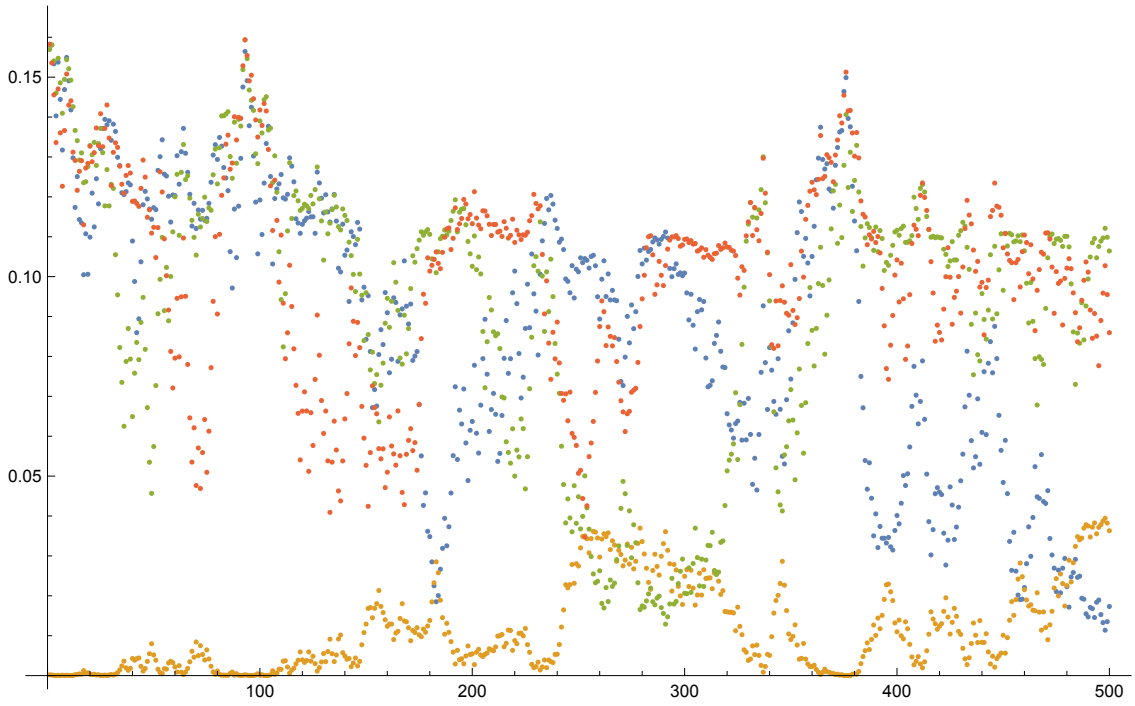
```
In[•]:= ListPlot[Transpose[ChangeinExtractablework],
          AxesLabel → Automatic, PlotRange → All]
```

Out[•]=

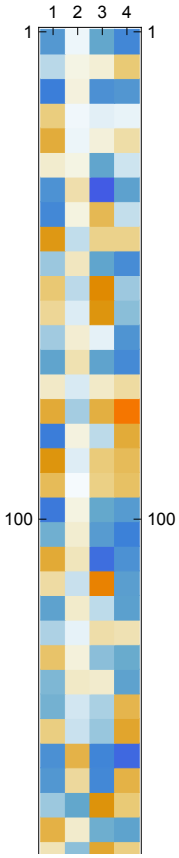


In[ ]:=
```
ListPlot[Transpose[ExtractableWorkOverTime],
 AxesLabel → Automatic, PlotRange → All]
```

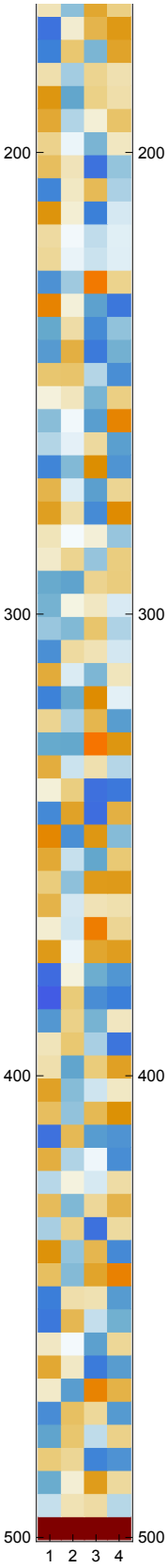
Out[ ]=




In[ ]:=
```
ChangeinExtractablework // MatrixPlot
```



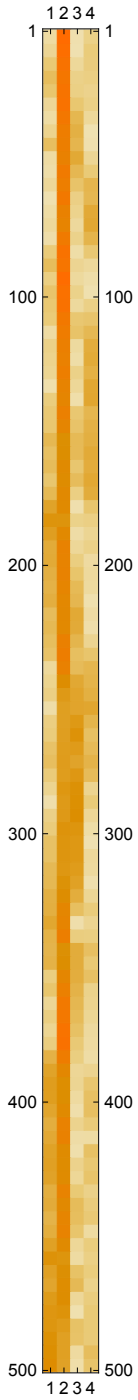
Out[ ]=



In[ ]:=

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

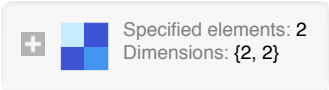
Out[ ]=



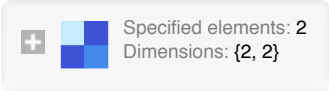
# 8-Qubit gas model with one qubit temperature variation

In[328]:=
```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
 RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8};
  vals = RandomSample[Table[i, {i, 8}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 8}]]
 ]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[2, 4];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^4]];
  Hi2 = KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction];
  interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2) .1]]
 ]
```

In[187]:=
```
ρi = NThermalQBit[{.1, .2}];
rho1for = PTR[ρi, {Q1}]
rho2for = PTR[ρi, {Q2}]
VNEntropy[rho1for]
VNEntropy[rho2for]
PTR[ρi, {Q2}][[2, 2]]
```

Out[188]= DM[⟨|data → SparseArray[  , qbitIDs → ⟨|Q2 → 1|⟩, nqbit → 1|⟩]



Out[189]= DM[⟨|data → SparseArray[  , qbitIDs → ⟨|Q1 → 1|⟩, nqbit → 1|⟩]



Out[190]= 0.500402

Out[191]= 0.325083

Out[192]= 0.1

In[331]:=
```
ρi = NThermalQBit[{.2, .2, .2, .2, .4, .2, .2, .2}];
```

In[332]:=
```
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
    RO = RandomOrder[];
    Ui = MakeDM[interactionUnitary, RO];
    ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
    Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
```

In[336]:=
```
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
```

In[337]:=
```
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
```

In[338]:=
```
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
```

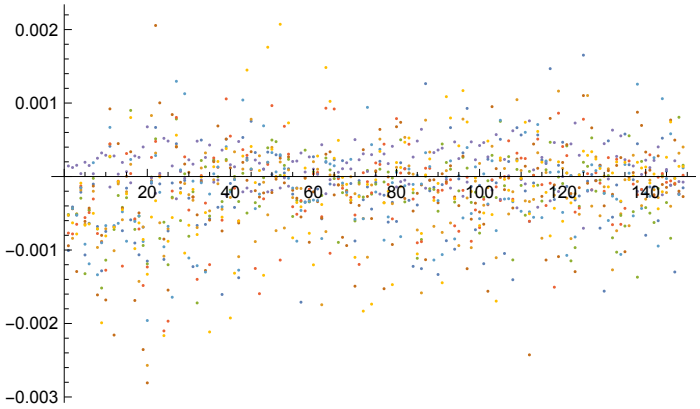
In[339]:=
```
Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
```

In[327]:=
```
TemperatureOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]][[2, 2]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
```

In[341]:=
```
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];

ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]
```

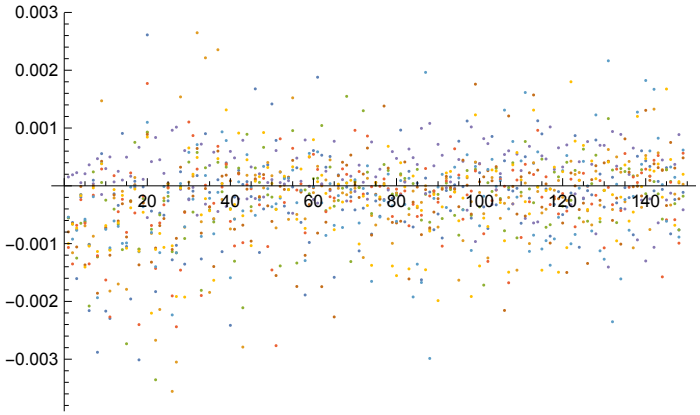
150 trials e = 2

Out[342]=



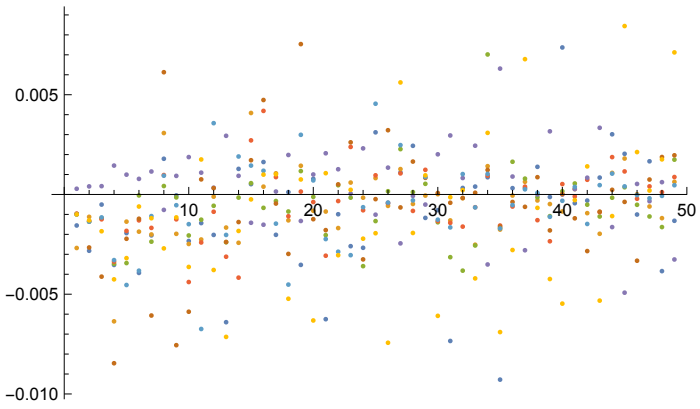

150 trials E = 2

Out[253]=



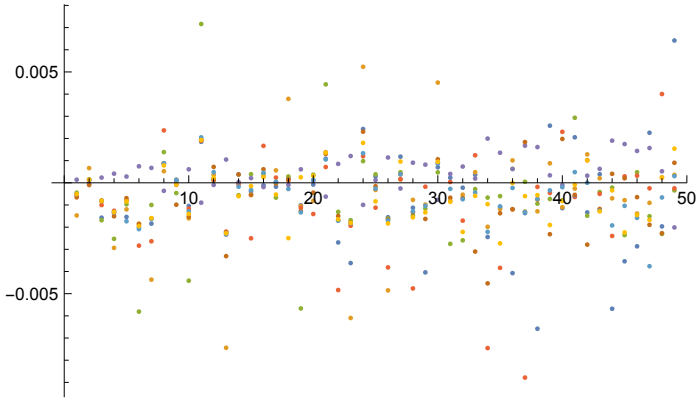

50 trials II E = 1

Out[231]=



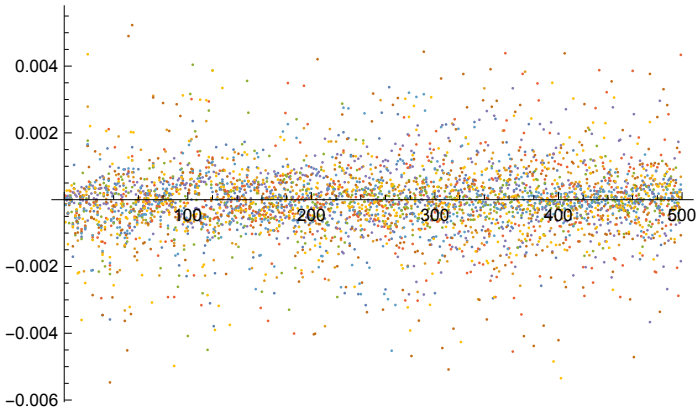
**50 trials I E = 1**

Out[207]=



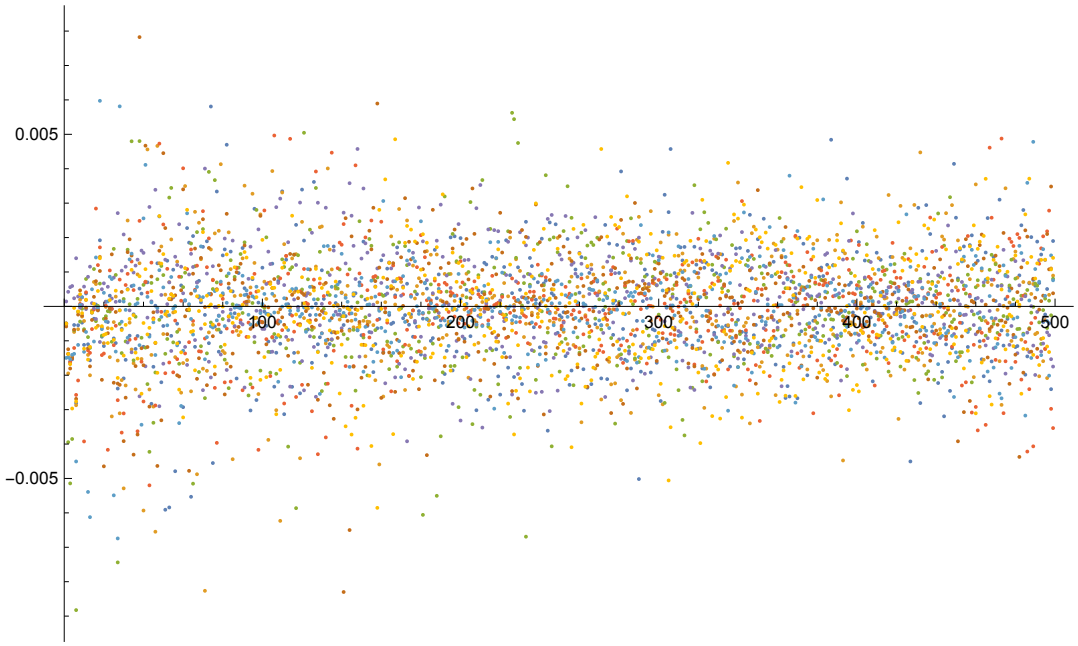

**500 trials II E = 1**

Out[ ]=



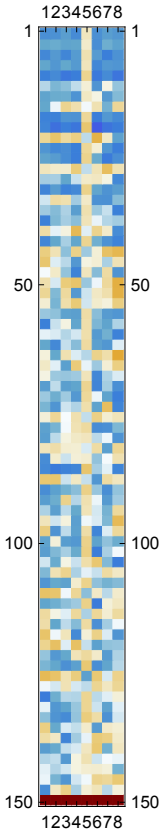

**500 trials I E = 1**

Out[ ]=



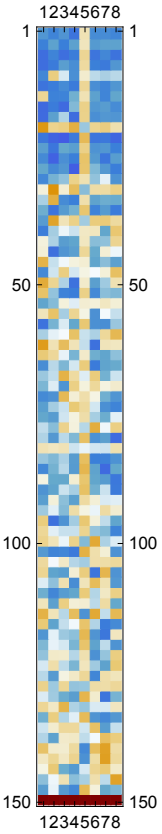
In[343]:=
```
ChangeinExtractablework // MatrixPlot
```

Out[343]=



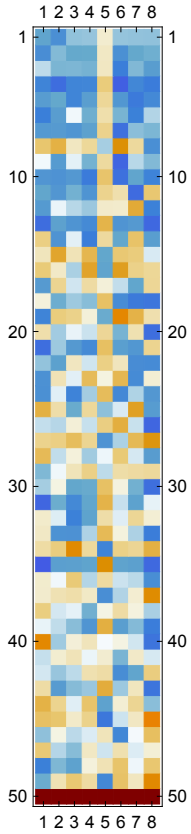
**150 trials e = 2**

Out[254]=



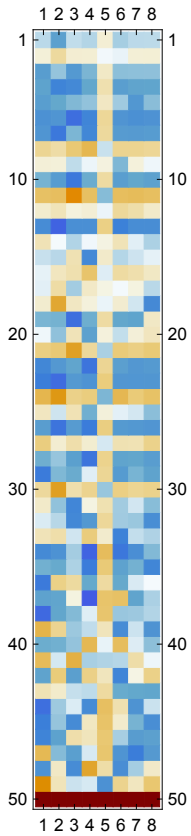

**50 trials II E = 1**

Out[232]=

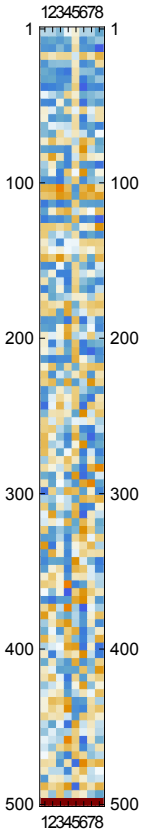



50 trials I E = 1

Out[208]=

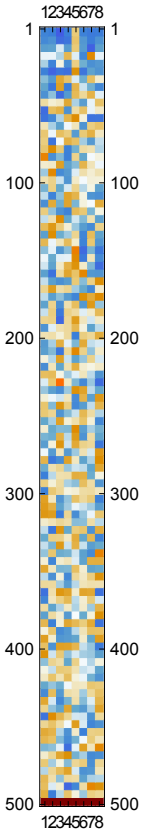


**500 trials II E = 1**



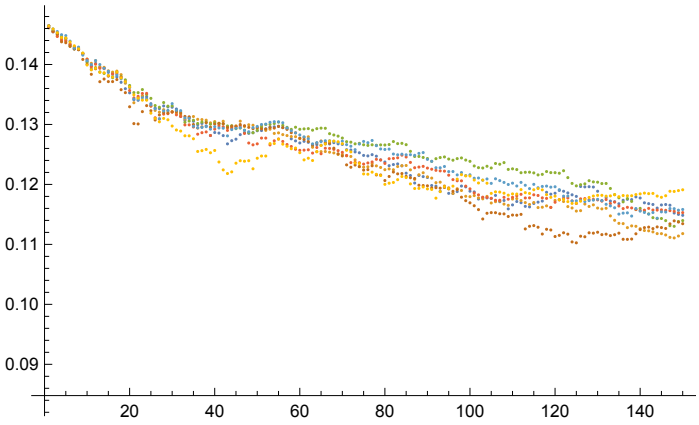

**500 trials E = 1**

Out[ ]=



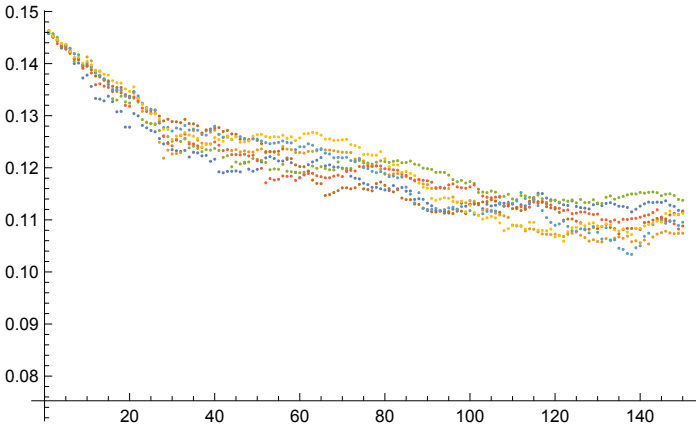

In[344]:=
```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

Out[344]=



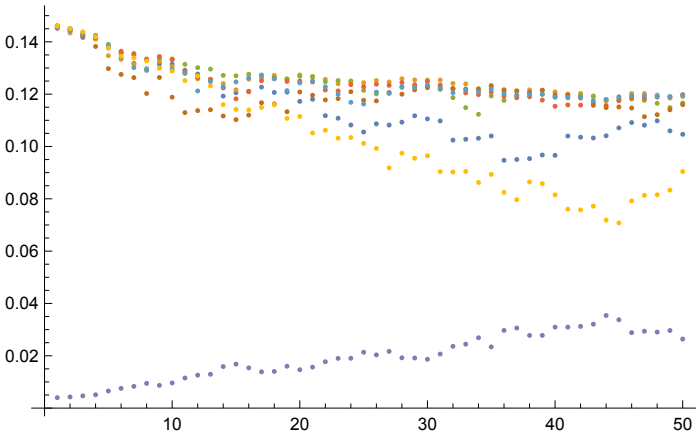
**150 trials e = 2**

Out[255]=



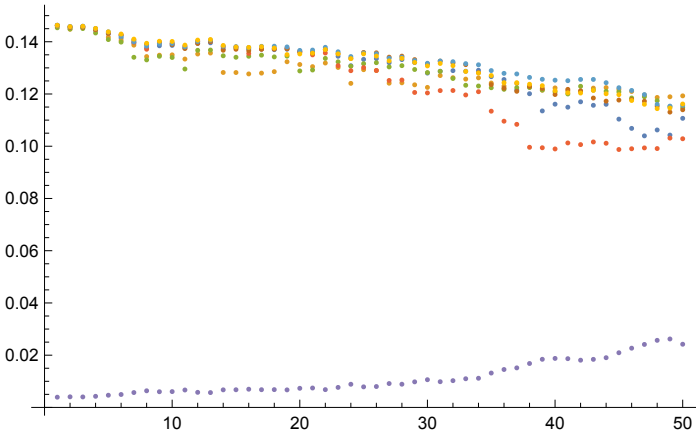

**50 trials II E = 1**

Out[233]=



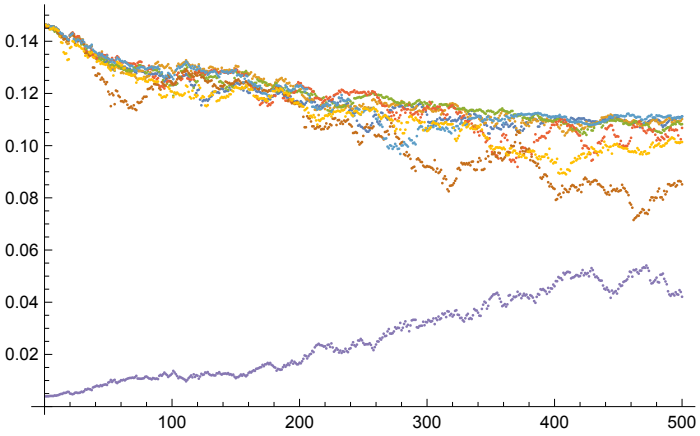

**50 trials I**

Out[209]=

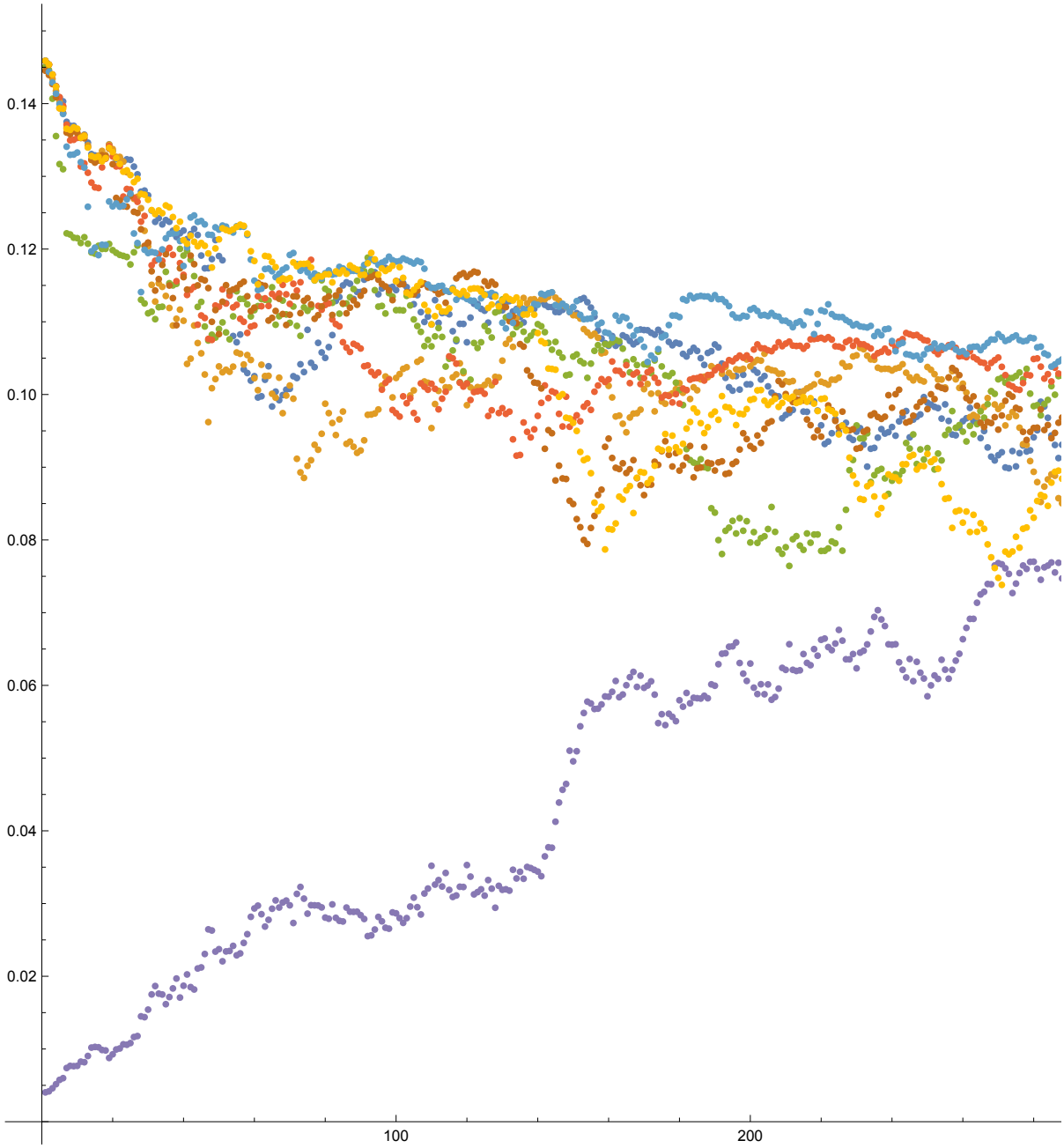


500 trials II

Out[ ]=

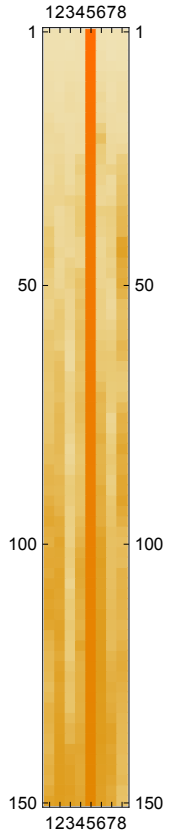


Out[ ]=



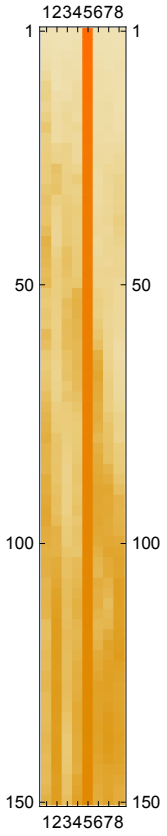
In[345]:=
```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

Out[345]=



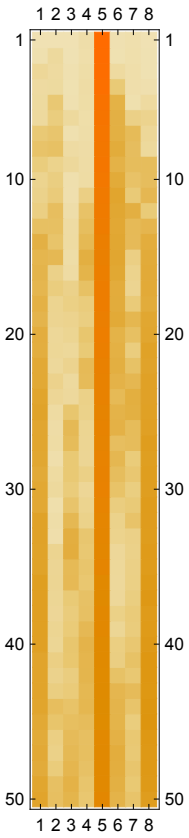
**150 trials e = 2**

Out[256]=



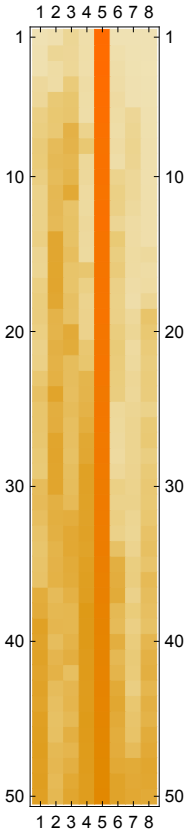

**50 trials E = 1**

Out[234]=



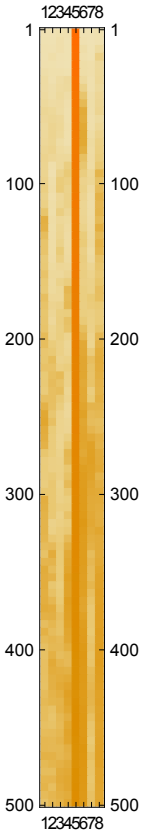
**50 trials I**

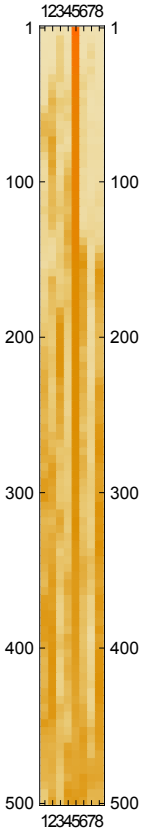
Out[210]=




**500 trials II**

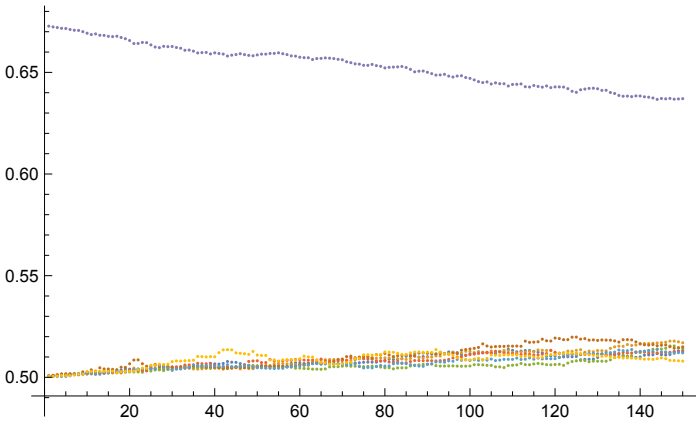
Out[ ]=



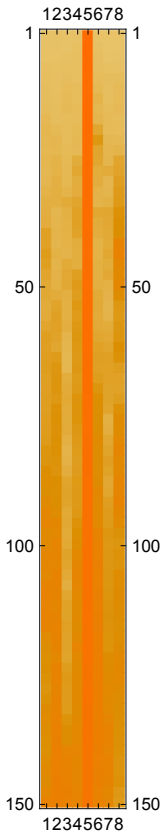

12345678
1
100
200
300
400
500
12345678

In[346]:=
```
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot
```

Out[346]=

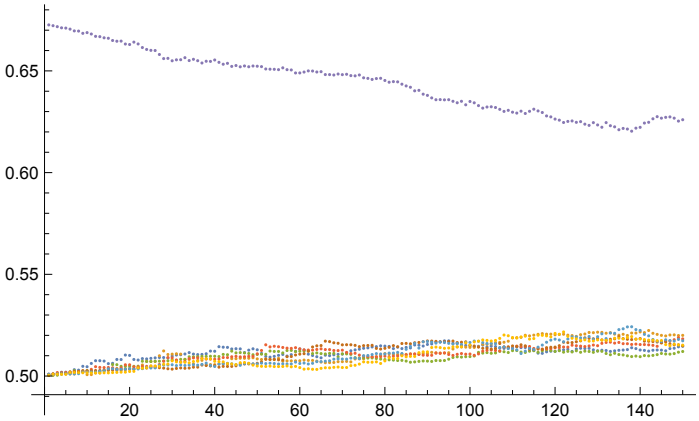



Out[347]=

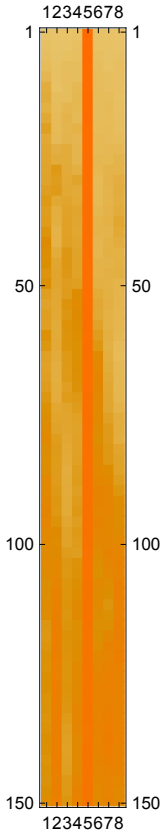


**150 trials e = 2**

Out[257]=

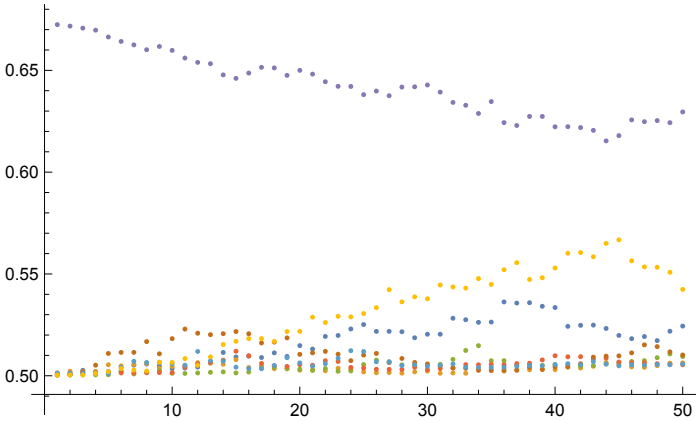



Out[258]=

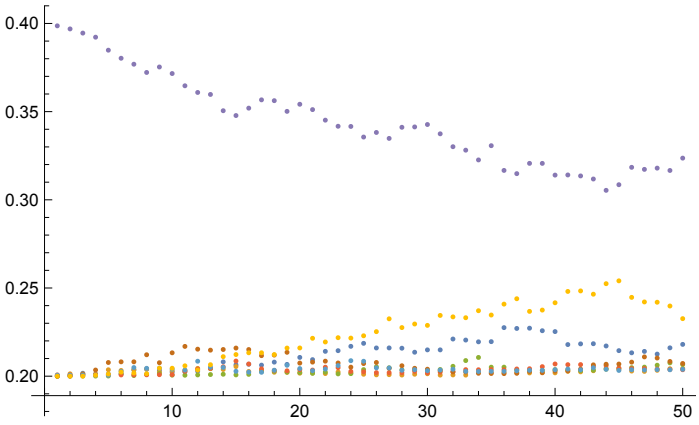



**50 trials II**

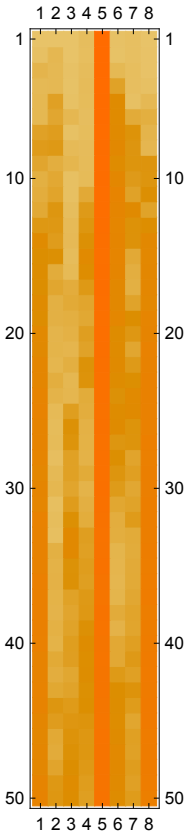
Out[235]=



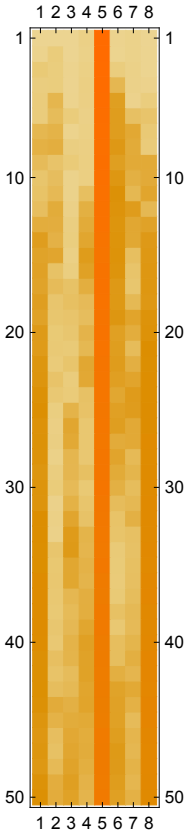
Out[236]=




Out[237]=

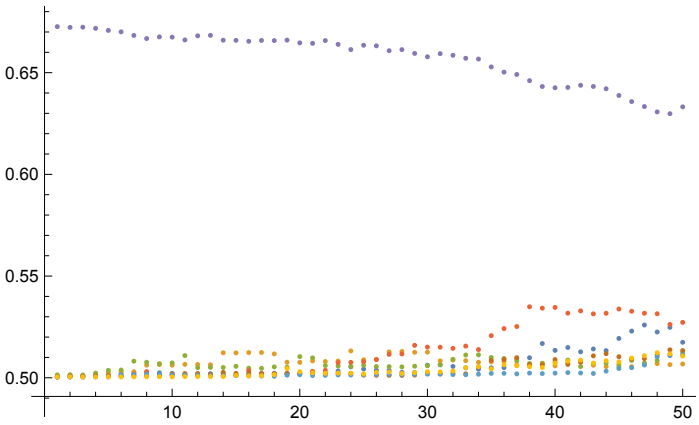


Out[238]=

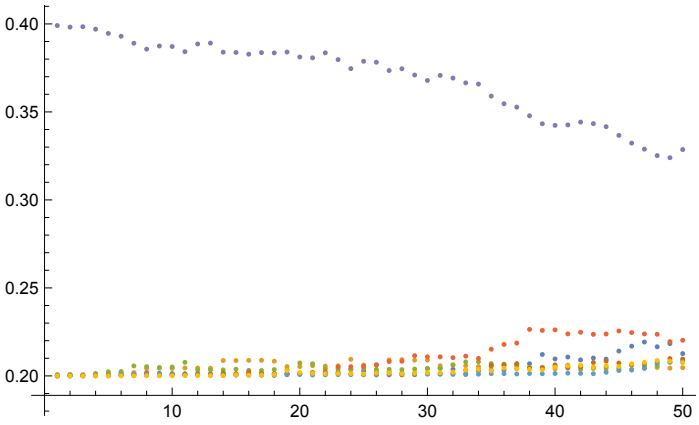



**50 trials I**

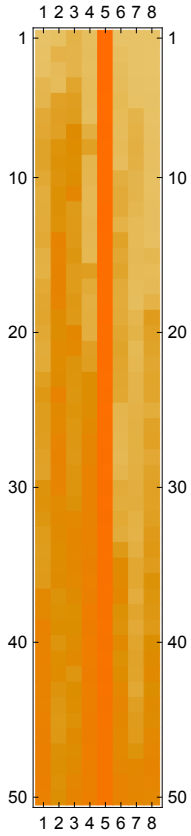
Out[213]=



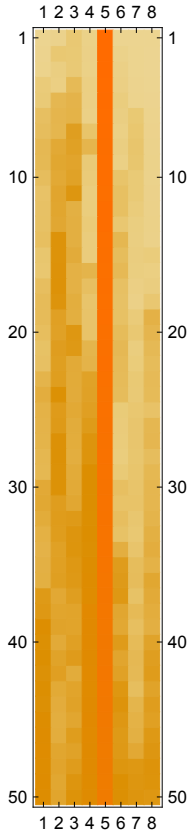

Out[214]=



Out[215]=




Out[216]=




# 8-Qubit gas model with multiple qubit

# temperature variation

In[383]:=
```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
 RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8};
  vals = RandomSample[Table[i, {i, 8}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 8}]]
 ]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[2, 4];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^4]];
  Hi2 = KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction];
  interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2) .1]]
 ]
```

In[386]:=
```
ρi = NThermalQBit[{.2, .4, .2, .2, .4, .2, .4, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
    RO = RandomOrder[];
    Ui = MakeDM[interactionUnitary, RO];
    ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
    Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
```

In[391]:=
```
AmbientTemps = Table[
    Association[Table[
      qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
    ,
    {P, DMS}];
```

In[392]:=
```
AmbientPops = Table[
    Association[Table[
      qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
    {Temps, AmbientTemps}];
```

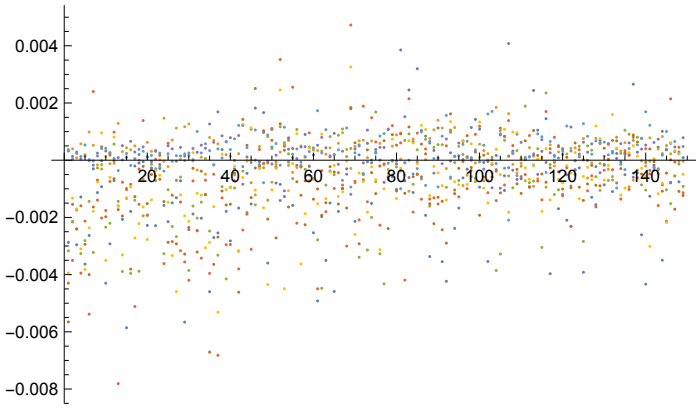
In[393]:=
```
ExtractableWorkOverTime = N[Module[{P},
     Table[
      P = DMS[[t]];
      Table[
       AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
          ThermalQBit[AmbientPops[[t]][qbit]]
        ],
       {qbit, Keys[P[qbitIDs]]}],
      {t, Length[DMS]}]
    ]];
```

In[374]:=
```
Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
   Table[
    P = DMS[[t]];
    Table[
     VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
     {qbit, Keys[P[qbitIDs]]}],
    {t, Length[DMS]}]
  ]];
```

In[376]:=
```
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];

ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]
```

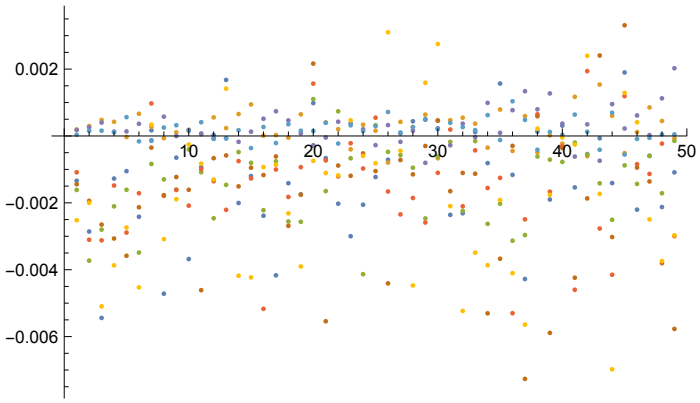
150 trials

Out[377]=



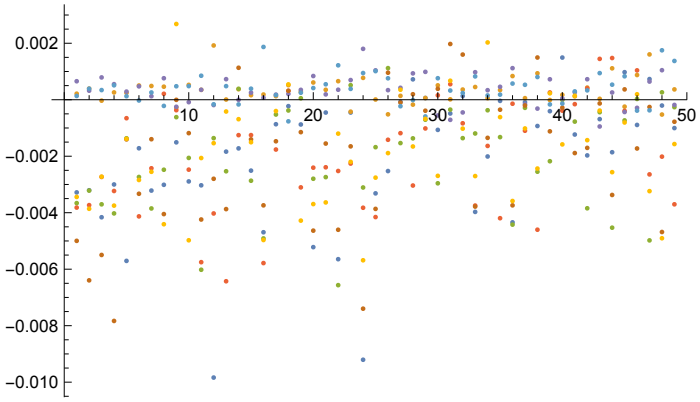

50 trials e = 1

Out[313]=



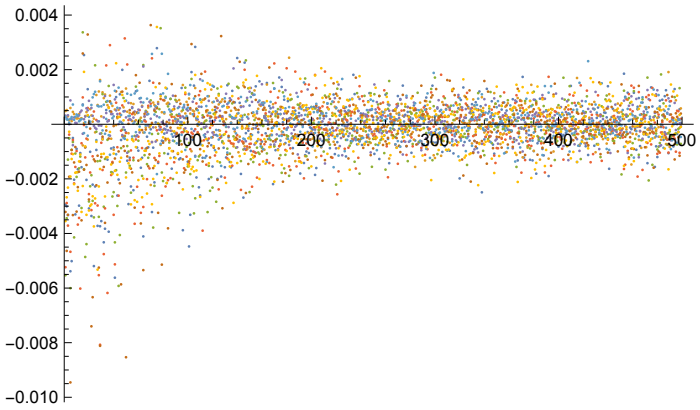
**50 trials e = 1**

Out[273]=

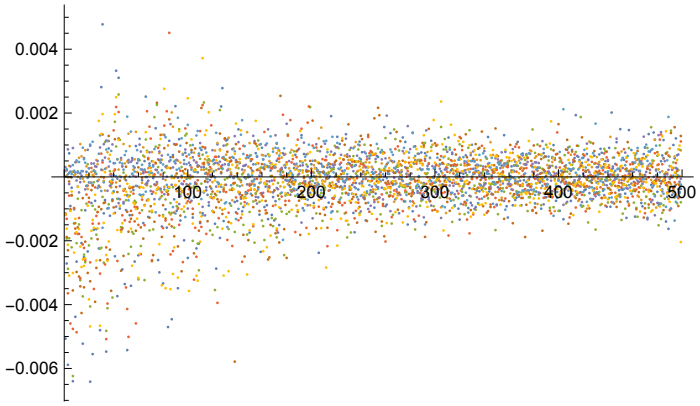



**500 trials e = 1 II**

Out[ ]=

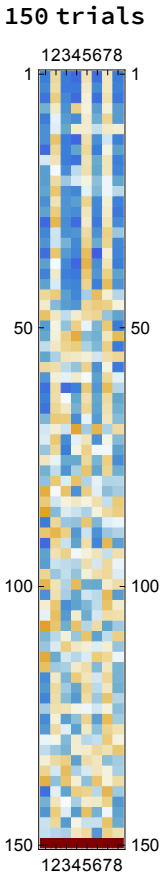



Out[ ]=



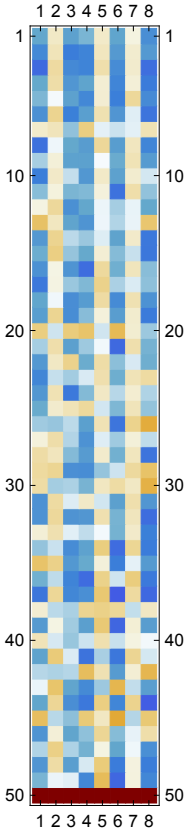

**ChangeinExtractablework // MatrixPlot**

Out[378]=



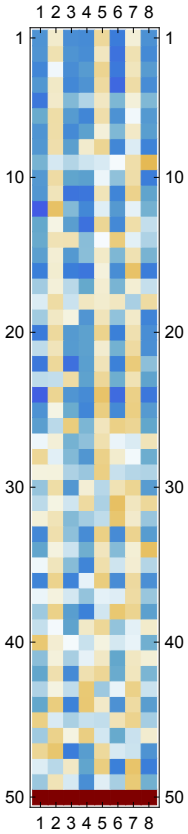
**50 trials e = 1**

Out[314]=



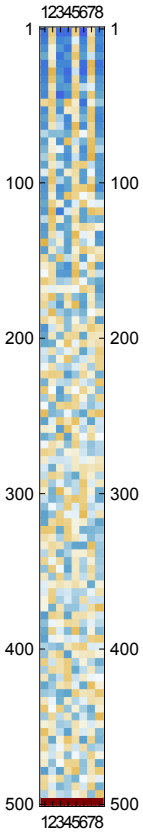

**50 trials e = 1**

Out[274]=



```
500 trials E = 1
```

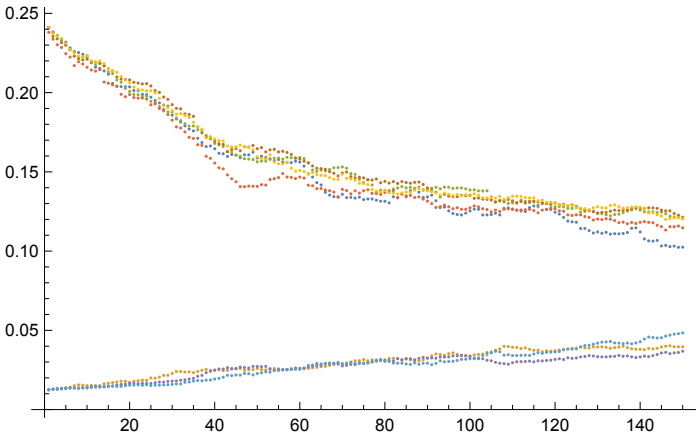
Out[ ]=




```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

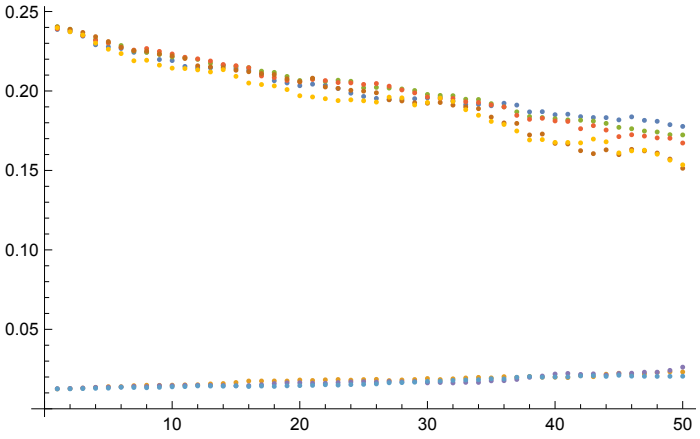
```
150 trials
```

Out[379]=



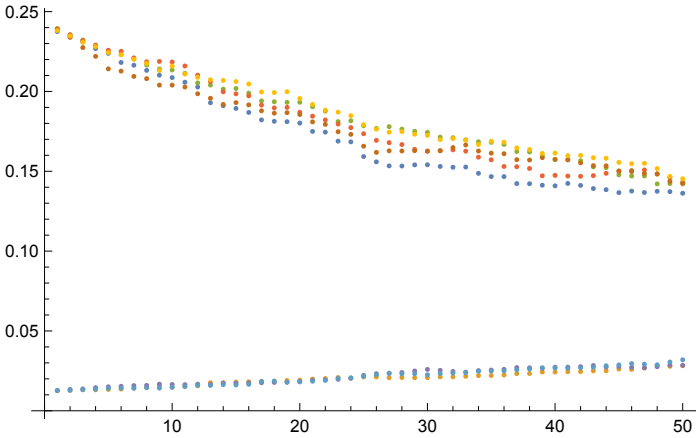
**50 trials e = 1**

Out[315]=



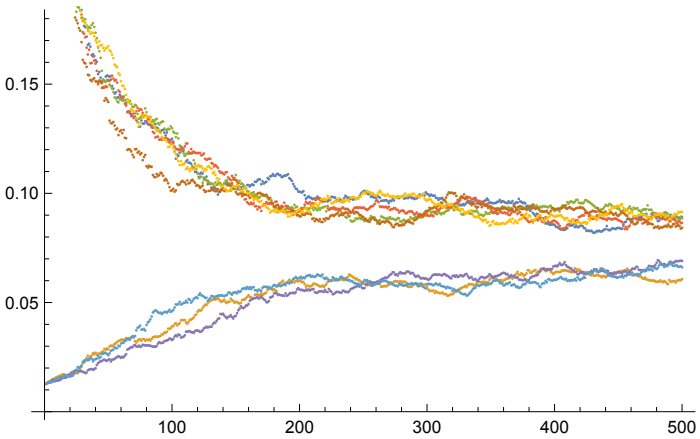

**50 trials e = 1**

Out[275]=

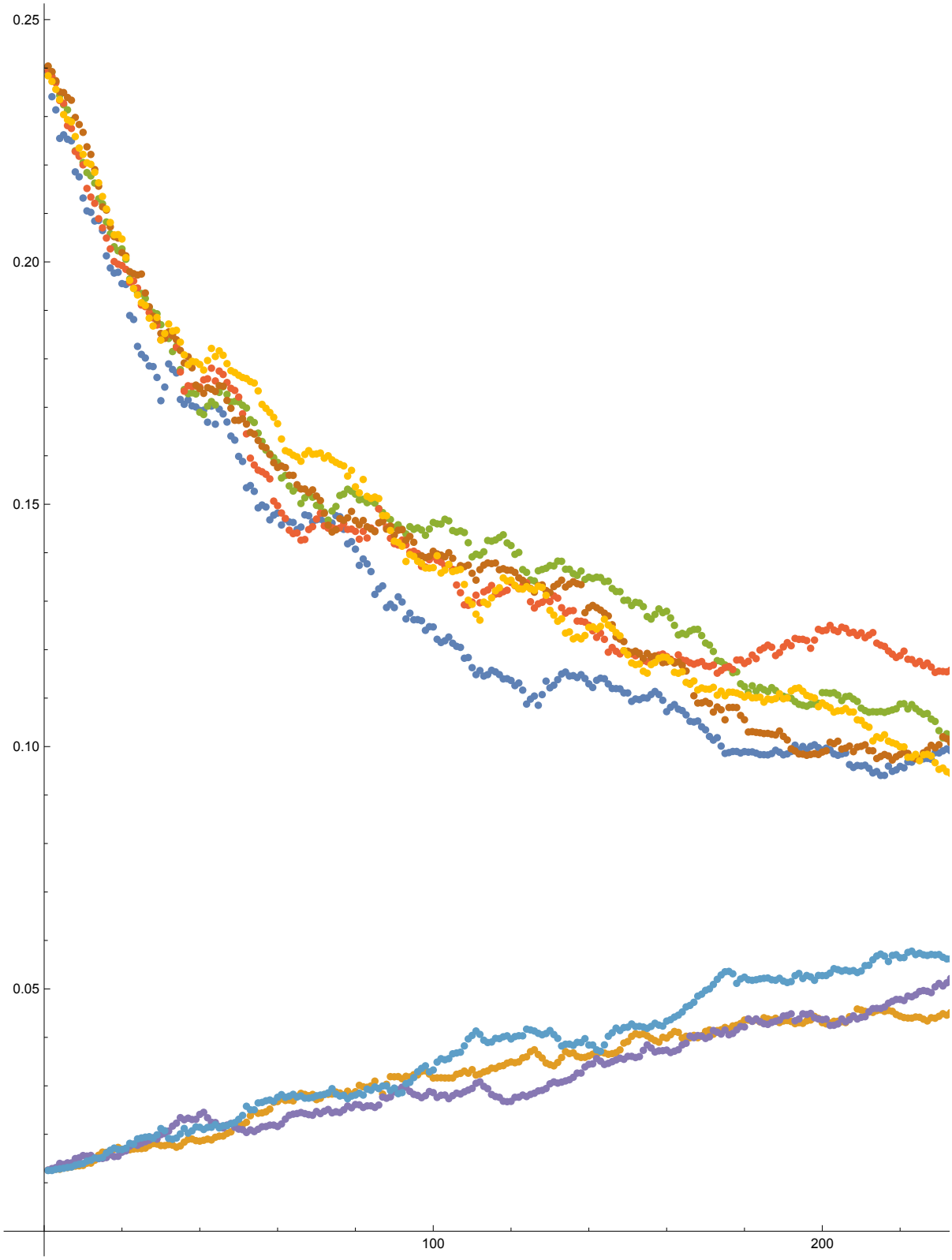



**500 trials II e = 1**

Out[ ]=

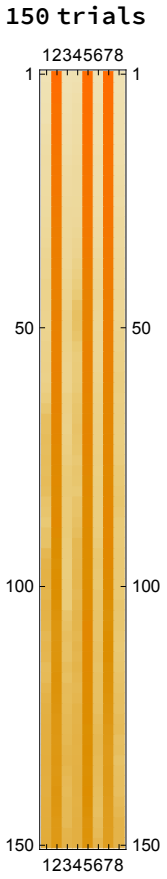


Out[•]=



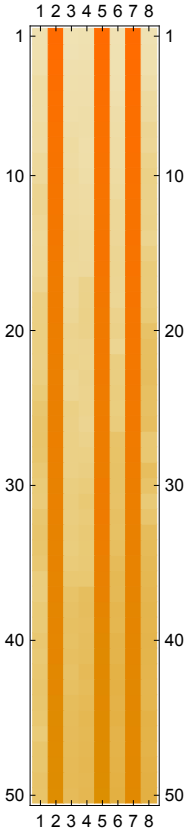

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

Out[380]=



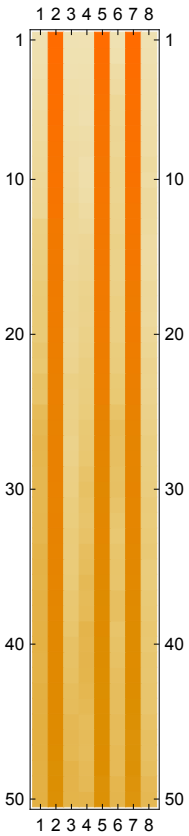
50 trials e = 1

Out[316]=



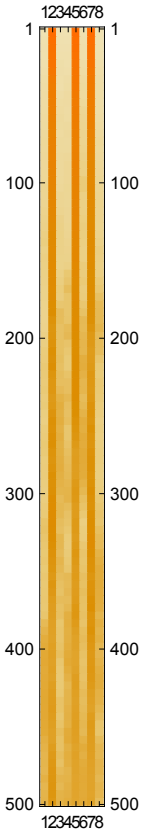

50 trials e = 1

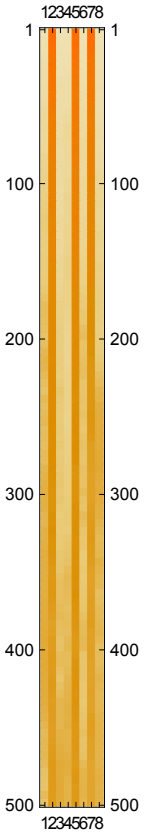
Out[276]=



**500 trials II e = 1**

*Out[ ]=*

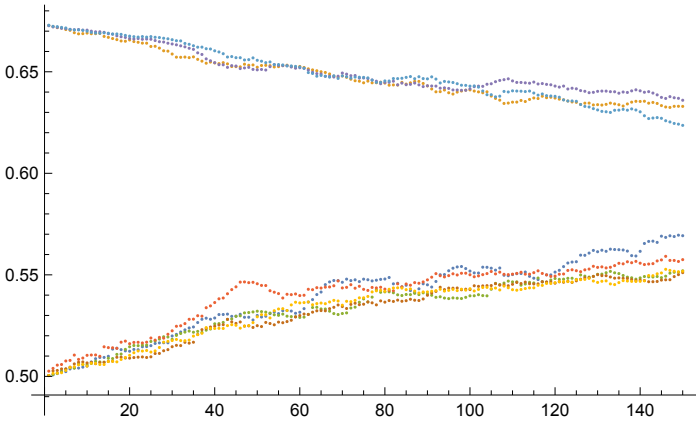


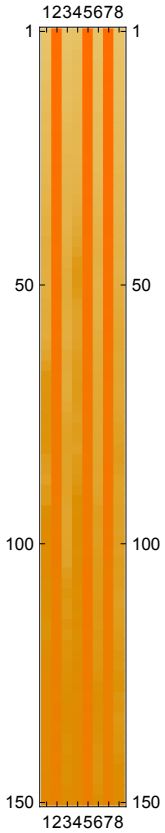
```
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot
```

150 trials

Out[381]=

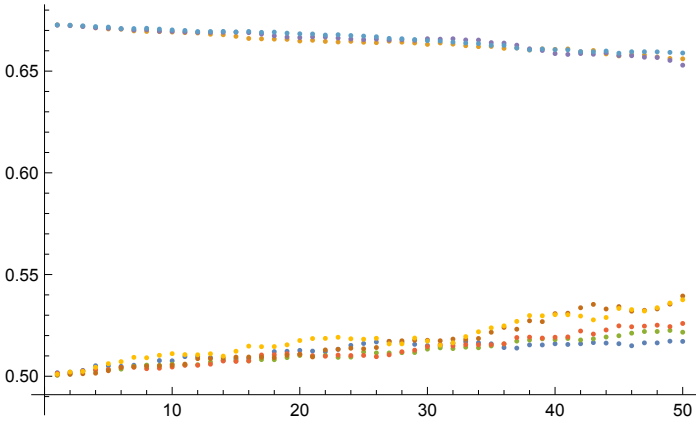



Out[382]=

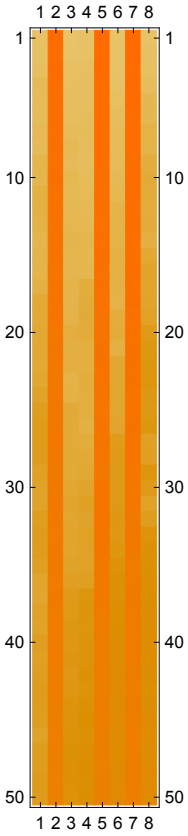


**50 trials e = 1**

Out[317]=

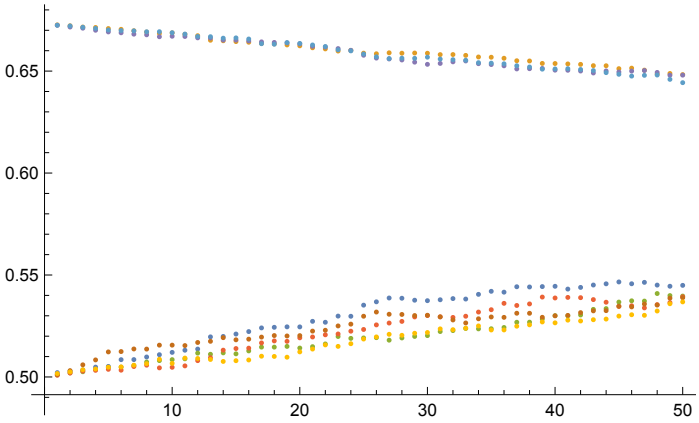



Out[318]=

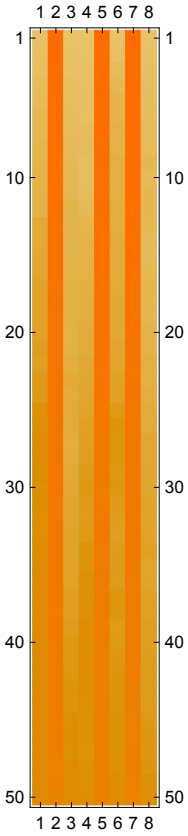


50 trials e = 1

Out[277]=




Out[278]=



# 8-qubits in a line with two qubit nearest neighbour interaction

```
In[•]:= Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
         RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
         AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
        RandomUnitary :=
         Module[{SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary},
          SubsysInteraction = RandomHamiltonian[1, 2];
          Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^6]];
          Hi2 =
           KroneckerProduct[IdentityMatrix[2^2], SubsysInteraction, IdentityMatrix[2^4]];
          Hi3 = KroneckerProduct[IdentityMatrix[2^4],
            SubsysInteraction, IdentityMatrix[2^2]];
          Hi4 = KroneckerProduct[IdentityMatrix[2^6], SubsysInteraction];
          interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4) .1]]
         ]

In[•]:= PossibleOrders = {
            ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8|⟩,
            <|Q1 → 8, Q2 → 1, Q3 → 2, Q4 → 3, Q5 → 4, Q6 → 5, Q7 → 6, Q8 → 7|>
           };

        ρi = NThermalQBit[{.2, .2, .2, .2, .4, .2, .2, .2}];
        ρ = ρi;
        baseOrder = ρ[qbitIDs];
        interactionUnitary = RandomUnitary;
        DMS = Table[
            RO  = RandomChoice[PossibleOrders];
            Ui = MakeDM[interactionUnitary, RO];
            ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
            Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];

In[•]:= AmbientTemps = Table[
            Association[Table[
              qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
            ,
            {P, DMS}];

In[•]:= AmbientPops = Table[
            Association[Table[
              qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
            {Temps, AmbientTemps}];
```

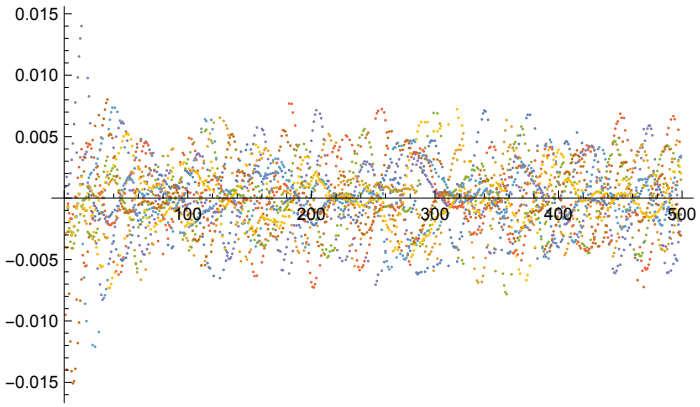
```
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
```

In[•]:=
```
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
```

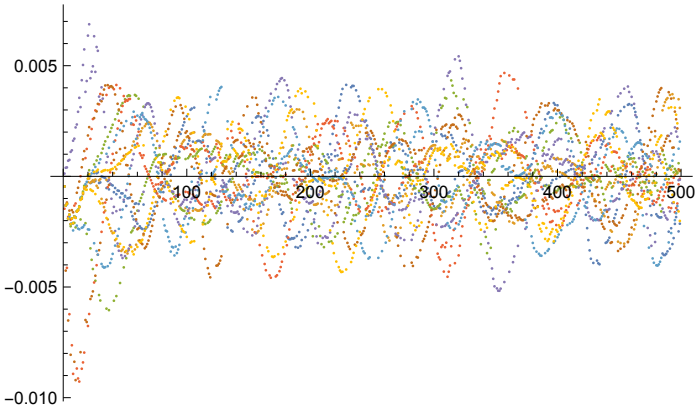
```
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]
```

500 trials

Out[•]=

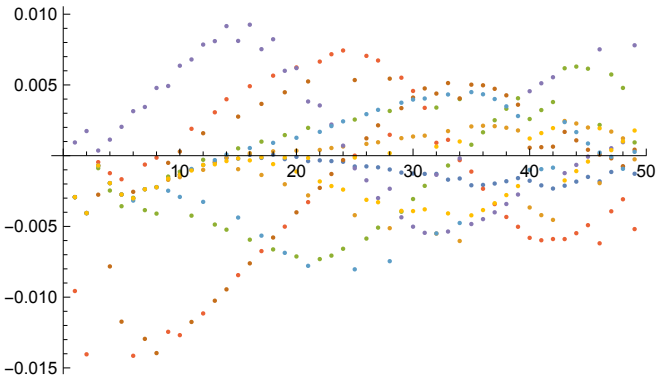


## 500 trials




```
50 trials e = 1
```

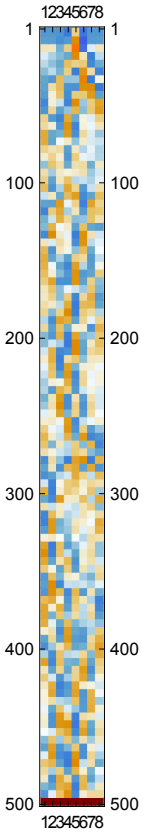
Out[ ]=




```
ChangeinExtractablework // MatrixPlot
```

500 trials e = 1

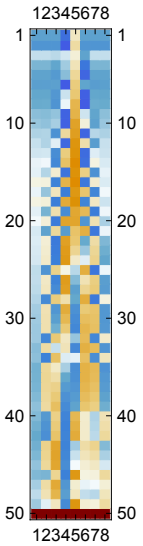
Out[ ]=



## 500 trials


```
50 trials e = 1
```

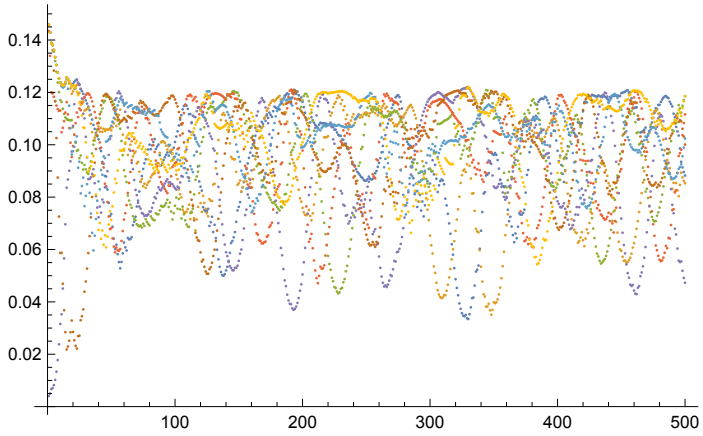
Out[ ]=



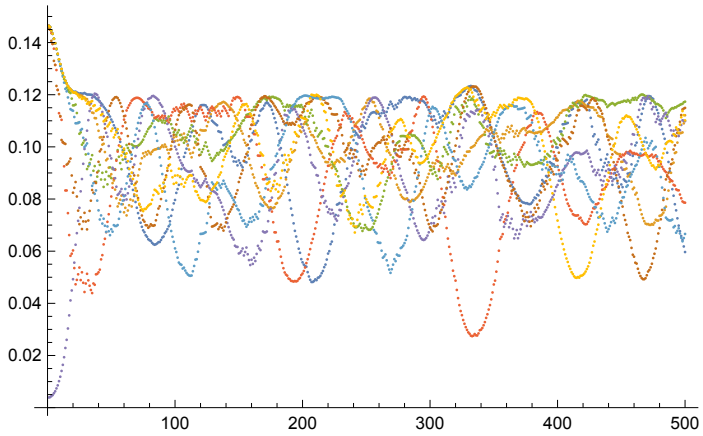

```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

**500 trials e = 1**

Out[ ]=

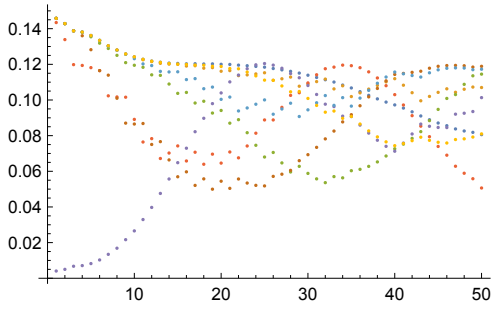



## 500 trials




**50 trials e = 1**

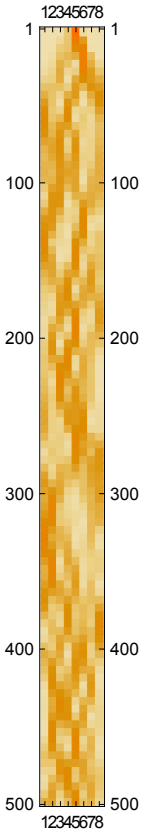
Out[ ]=




```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

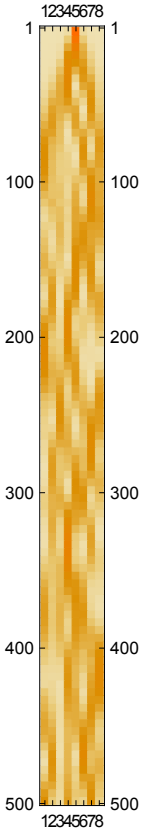
**500 trials e = 1**

Out[ ]=

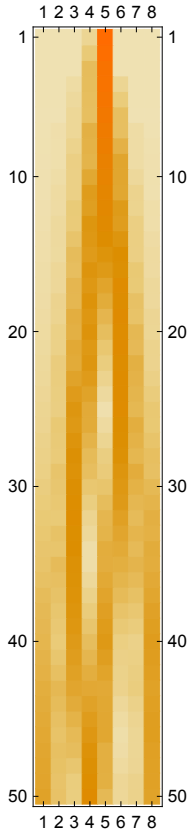


## 500 trials

Out[ ]=




```
50 trials e = 1
```




```
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot
```

# 8-qubits in a line with two qubit nearest neighbour interaction (multiple temp variation)

```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
 RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomUnitary :=
 Module[{SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[1, 2];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^6]];
  Hi2 =
   KroneckerProduct[IdentityMatrix[2^2], SubsysInteraction, IdentityMatrix[2^4]];
  Hi3 = KroneckerProduct[IdentityMatrix[2^4],
    SubsysInteraction, IdentityMatrix[2^2]];
  Hi4 = KroneckerProduct[IdentityMatrix[2^6], SubsysInteraction];
  interactionUnitary = SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4) .1]]
```

```
 ]
PossibleOrders = {
   ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8|⟩,
   <|Q1 → 8, Q2 → 1, Q3 → 2, Q4 → 3, Q5 → 4, Q6 → 5, Q7 → 6, Q8 → 7|>
  };
ρi = NThermalQBit[{.2, .3, .2, .2, .4, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   RO  = RandomChoice[PossibleOrders];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];

ListPlot[Transpose[ChangeinExtractablework],
```

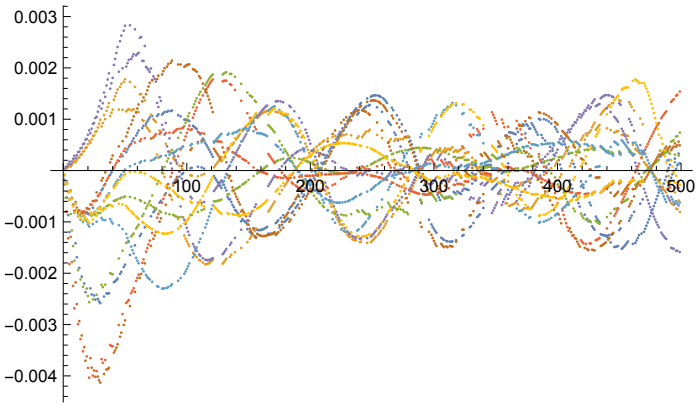
```
 AxesLabel → Automatic, PlotRange → All]
```

```
ChangeinExtractablework // MatrixPlot
```

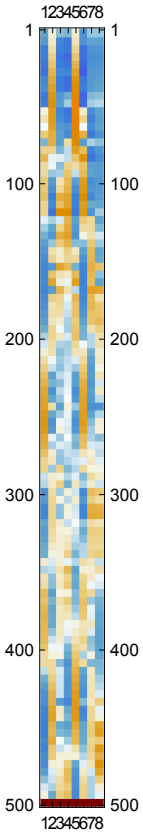
```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot
```

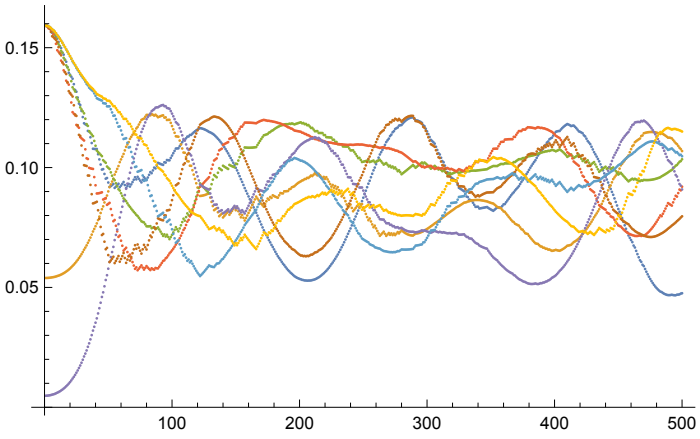
Out[ ]=



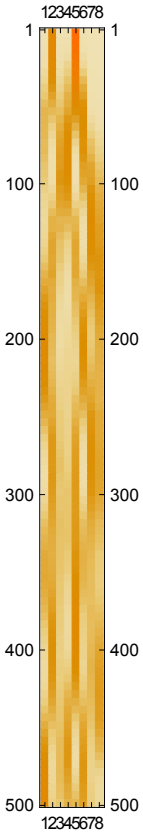
Out[ ]=


12345678
1
100
200
300
400
500
12345678

Out[ ]=

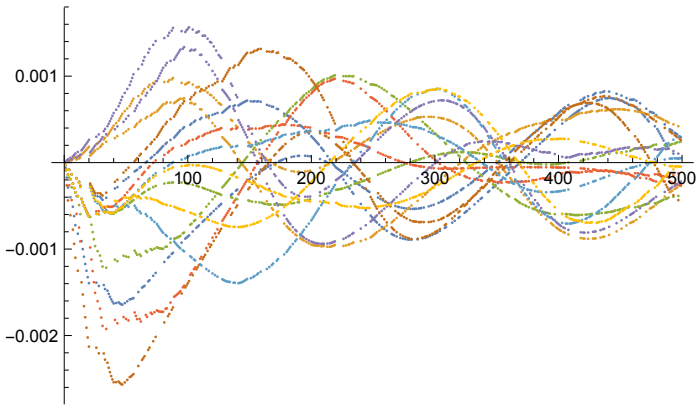

0.15
0.10
0.05
100
200
300
400
500

Out[ ]=

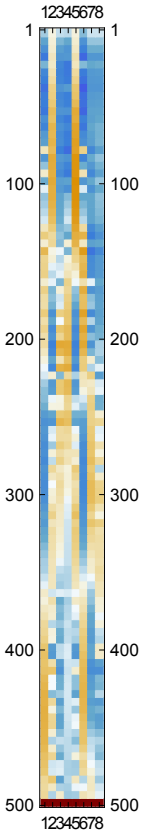



**500 trials**

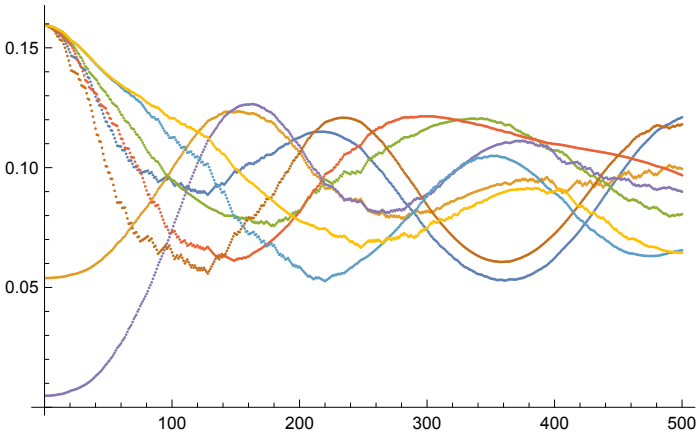
Out[ ]=



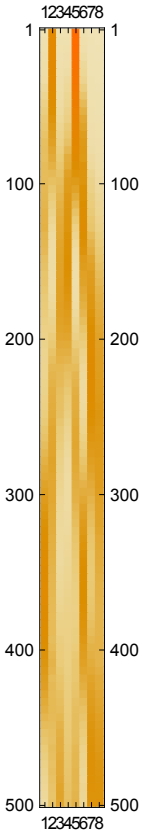
Out[ ]=


12345678
1
100
200
300
400
500
12345678

Out[ ]=


0.15
0.10
0.05
100
200
300
400
500

Out[ ]=




# 8 – Qubits on a lattice

# Unitary translationally invariant (same everywhere at every time step)

## Four qubit pockets (single temp variation)

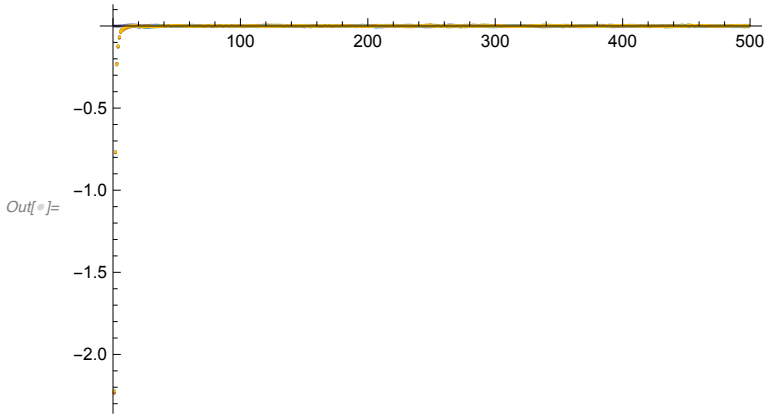
```
In[•]:= Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
         RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
         AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
        RandomOrder[] := Module[{items, vals},
          items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8};
          vals = RandomSample[Table[i, {i, 8}]];
          Association[Table[items[[i]] -> vals[[i]], {i, 8}]]
         ]
        RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, interactionUnitary},
          SubsysInteraction = RandomHamiltonian[1, 4];
          Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^4]];
          Hi2 = KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction];
          interactionUnitary = SparseArray[MatrixExp[I (Hi1 + Hi2) .1]]
         ]

In[•]:= PossibleOrders = {
            <|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8|>,
            <|Q1 → 5, Q2 → 6, Q3 → 3, Q4 → 4, Q5 → 1, Q6 → 2, Q7 → 7, Q8 → 8|>
          };

        ρi = NThermalQBit[{.2, .2, .2, .2, .5, .2, .2, .2}];
        ρ = ρi;
        baseOrder = ρ[qbitIDs];
        interactionUnitary = RandomUnitary;
        DMS = Table[
            (*RO = RandomOrder[];*)
            RO = RandomChoice[PossibleOrders];
            Ui = MakeDM[interactionUnitary, RO];
            ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
            Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];

In[•]:= AmbientTemps = Table[
            Association[Table[
              qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
            ,
            {P, DMS}];

In[•]:= AmbientPops = Table[
            Association[Table[
              qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
            {Temps, AmbientTemps}];
```

```
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
```

In[•]:=
```
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];

ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]
```

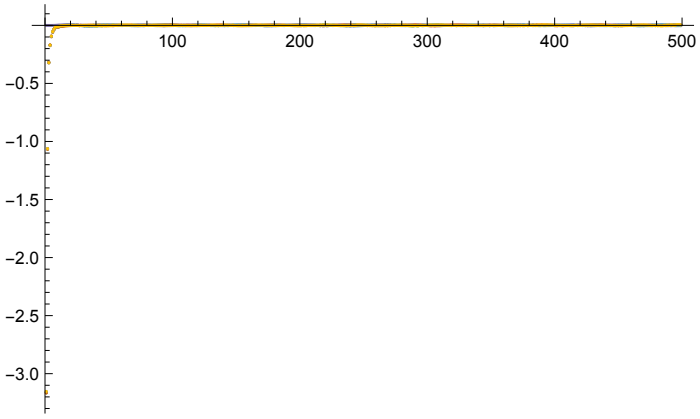
500 tirals

Out[•]=



```
500 trials
```

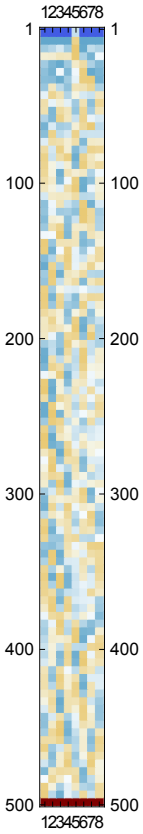
Out[ ]=




```
ChangeinExtractablework // MatrixPlot
```

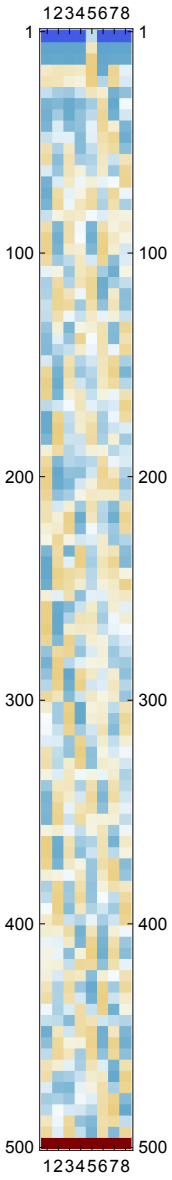
```
500 trials
```

Out[ ]=



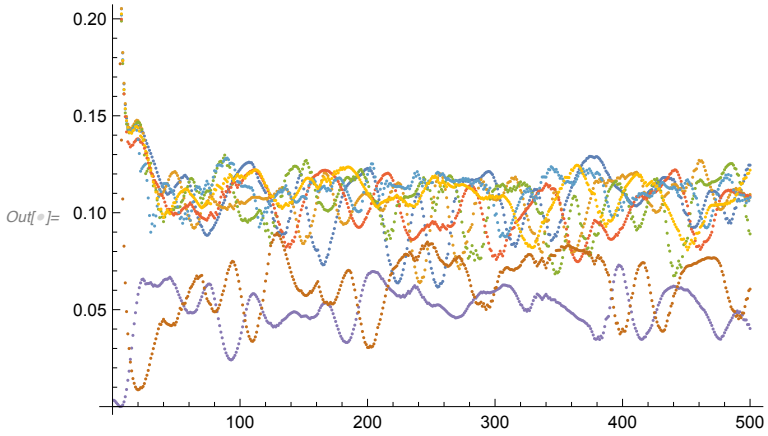
**500 trials**

Out[ ]=

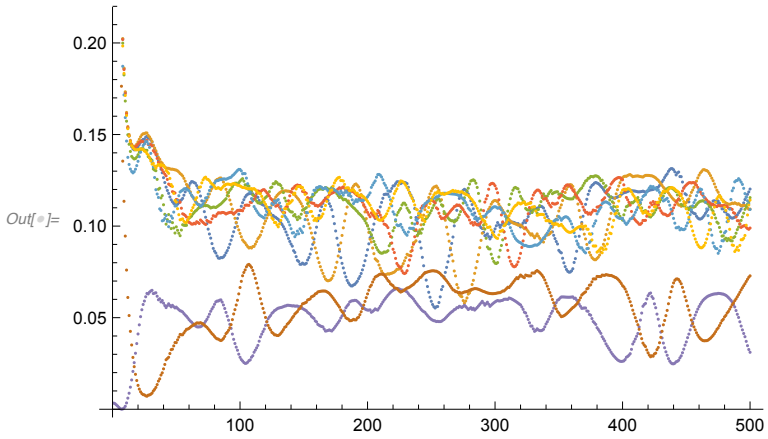



```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

```
500 trials
```



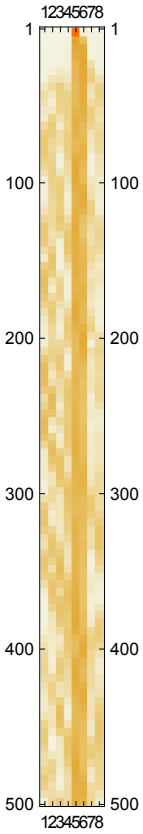

```
500 trials
```



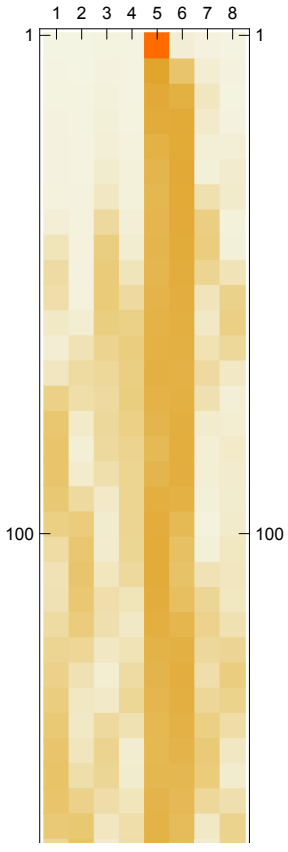

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

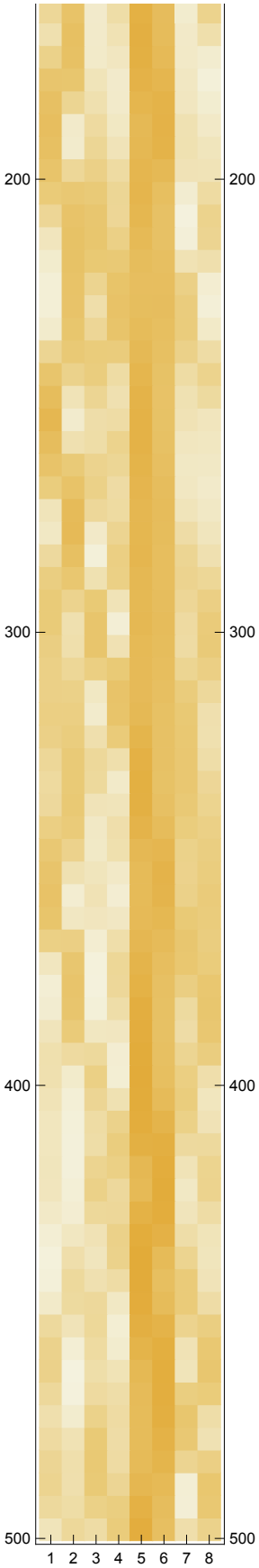
**500 trials**

Out[ ]=




**500 trials**





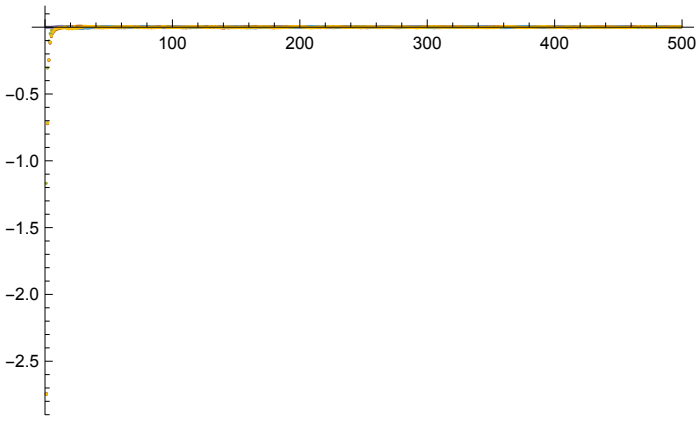

```
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot
```

## Four qubit pockets (multiple temp variation)

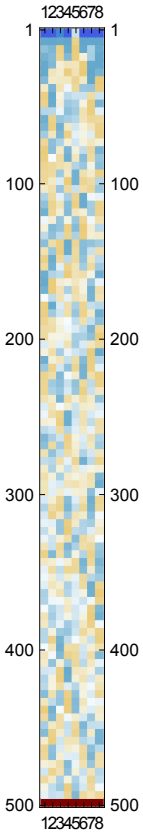
```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
 RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8};
  vals = RandomSample[Table[i, {i, 8}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 8}]]
 ]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[1, 4];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^4]];
  Hi2 = KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction];
  interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2) .1]]
 ]
PossibleOrders = {
   ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8|⟩,
   <|Q1 → 5, Q2 → 6, Q3 → 3, Q4 → 4, Q5 → 1, Q6 → 2, Q7 → 7, Q8 → 8|>
  };
ρi = NThermalQBit[{.2, .2, .3, .2, .5, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   (*RO = RandomOrder[];*)
   RO  = RandomChoice[PossibleOrders];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
```

```
        ],
       {qbit, Keys[P[qbitIDs]]}],
      {t, Length[DMS]}]
   ]];

Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot

ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]

ChangeinExtractablework // MatrixPlot

ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]

Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot

500 trials
```

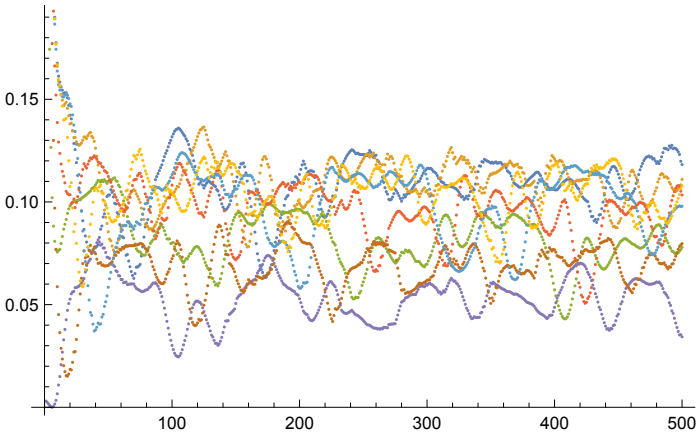
Out[ ]=



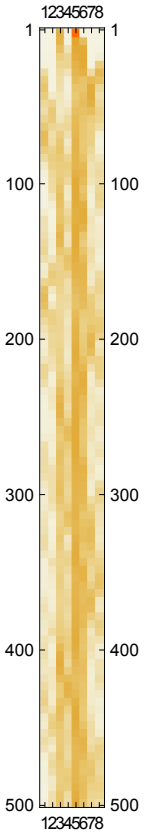
Out[ ]=


12345678
1
100
200
300
400
500
12345678

Out[ ]=

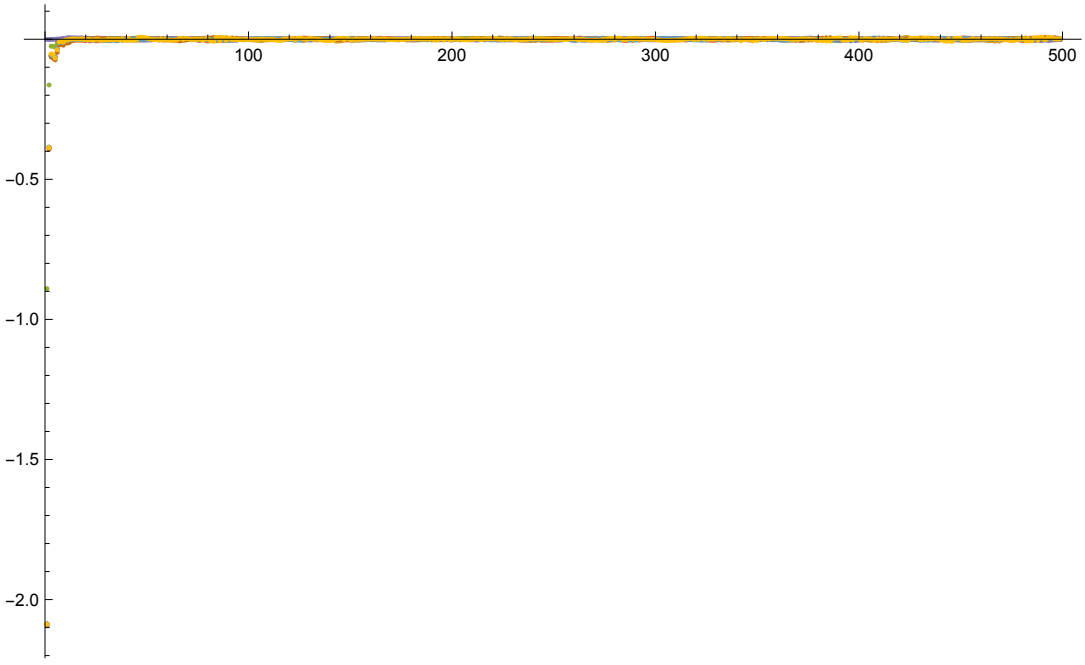

0.15
0.10
0.05
100
200
300
400
500

Out[ ]=

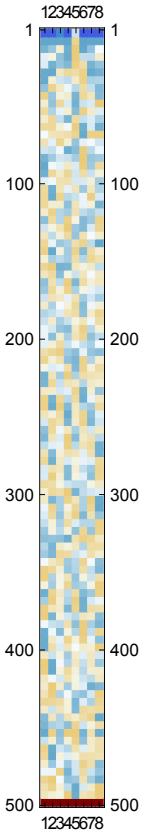



500 trials

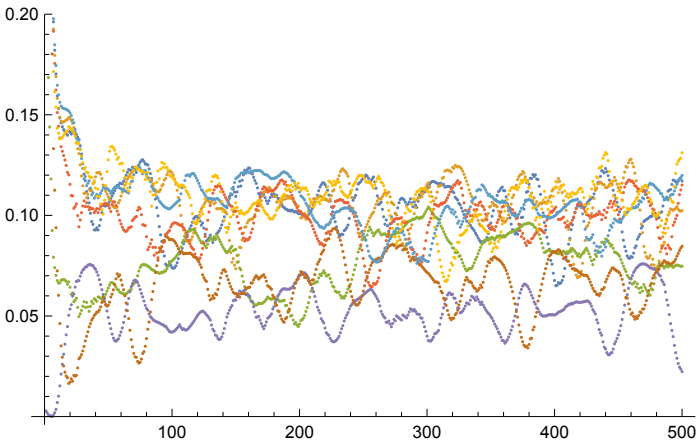
Out[ ]=



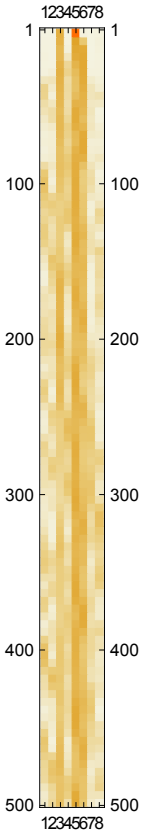
Out[ ]=


12345678
1
100
200
300
400
500
12345678

Out[ ]=


0.20
0.15
0.10
0.05
100
200
300
400
500

Out[ ]=




# Random Unitary (different unitaries in each pocket at every time step)

## Four qubit pockets (single temp variation)

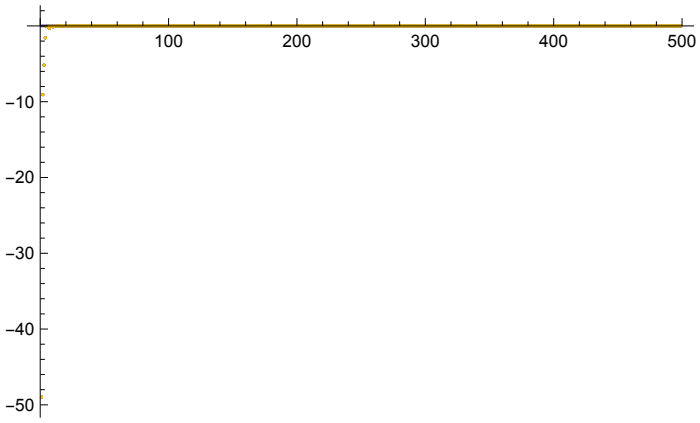
In[ ]:=
```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary, RandomUnitary,
 PossibleOrders, ρi, DMS, AmbientPops, AmbientTemps, SubsysInteraction1,
 SubsysInteraction2, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8};
  vals = RandomSample[Table[i, {i, 8}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 8}]]
 ]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, interactionUnitary},
  SubsysInteraction1 = RandomHamiltonian[1, 4];
  SubsysInteraction2 = RandomHamiltonian[3, 4];
  Hi1 = KroneckerProduct[SubsysInteraction1, IdentityMatrix[2^4]];
  Hi2 = KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction2];
  interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2) .1]]
 ]
```

```
In[•]:= PossibleOrders = {
          ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8|⟩,
          <|Q1 → 5, Q2 → 6, Q3 → 3, Q4 → 4, Q5 → 1, Q6 → 2, Q7 → 7, Q8 → 8|>
        };

      ρi = NThermalQBit[{.2, .2, .2, .2, .5, .2, .2, .2}];
      ρ = ρi;
      baseOrder = ρ[qbitIDs];
      interactionUnitary = RandomUnitary;
      DMS = Table[
          (*RO = RandomOrder[];*)
          RO  = RandomChoice[PossibleOrders];
          Ui = MakeDM[interactionUnitary, RO];
          ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
          Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];

In[•]:= AmbientTemps = Table[
          Association[Table[
            qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
          ,
          {P, DMS}];

In[•]:= AmbientPops = Table[
          Association[Table[
            qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
          {Temps, AmbientTemps}];

      ExtractableWorkOverTime = N[Module[{P},
           Table[
            P = DMS[[t]];
            Table[
             AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
                 ThermalQBit[AmbientPops[[t]][qbit]]
              ],
             {qbit, Keys[P[qbitIDs]]}],
            {t, Length[DMS]}]
          ]];

      Clear[EntropyOfQubits]
      EntropyOfQubits = N[Module[{P},
           Table[
            P = DMS[[t]];
            Table[
             VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
             {qbit, Keys[P[qbitIDs]]}],
            {t, Length[DMS]}]
          ]];
      ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
      EntropyOfQubits // MatrixPlot
```

In[•]:=
```
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];

ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]
```

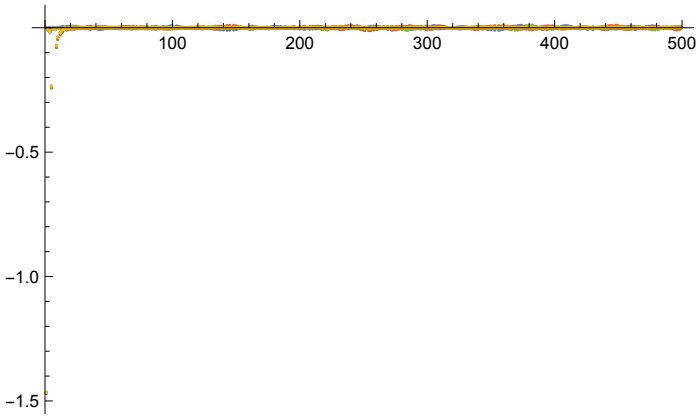
500 trials

Out[•]=




500 trials

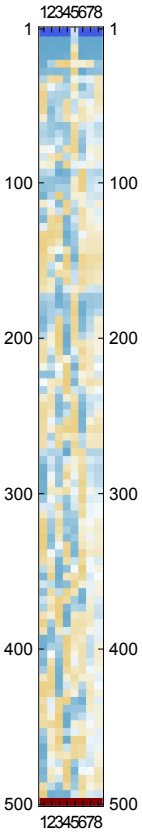
Out[•]=




```
ChangeinExtractablework // MatrixPlot
```

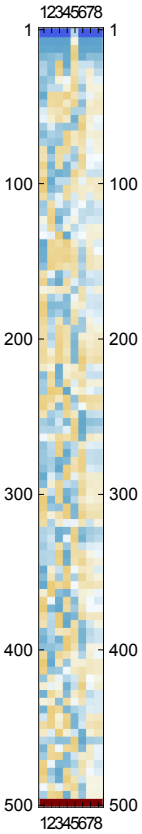
**500 trials**

Out[ ]=




**500 trials**

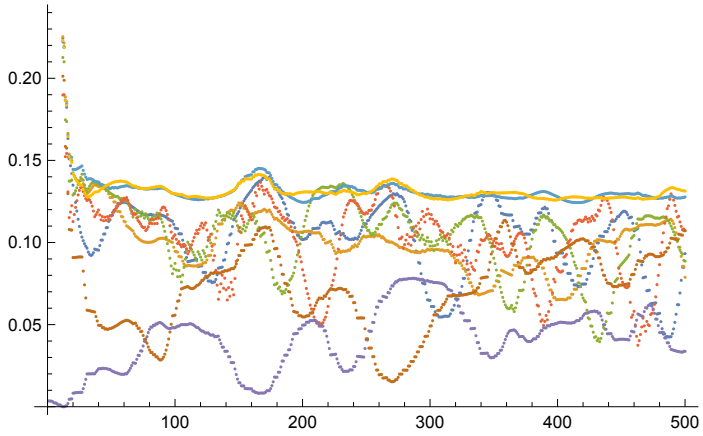
Out[ ]=



```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

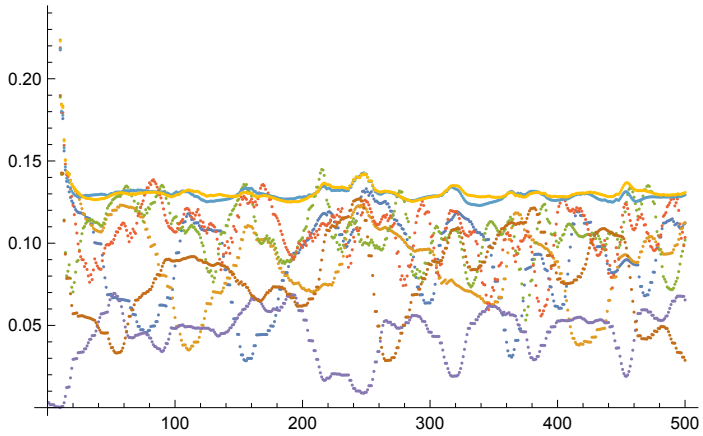
500 trials

Out[ ]=




500 trials

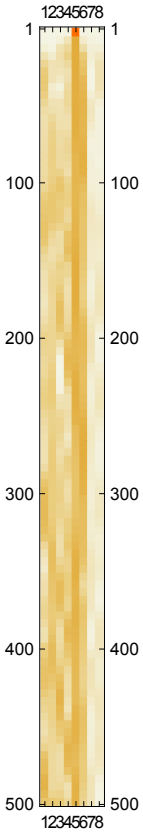
Out[ ]=




```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

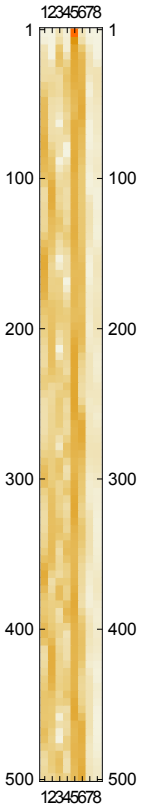
**500 trials**

Out[ ]=




**500 trials**

Out[ ]=

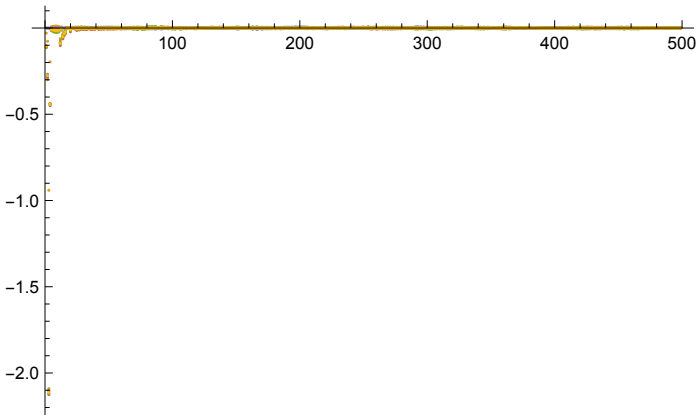


## Four qubit pockets (multiple temp variation)

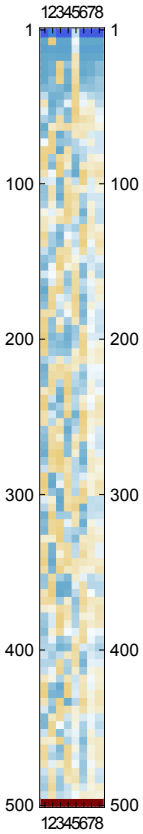
```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary, RandomUnitary,
 PossibleOrders, SubsysInteraction1, SubsysInteraction2, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8};
  vals = RandomSample[Table[i, {i, 8}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 8}]]
 ]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, interactionUnitary},
  SubsysInteraction1 = RandomHamiltonian[1, 4];
  SubsysInteraction2 = RandomHamiltonian[3, 4];
  Hi1 = KroneckerProduct[SubsysInteraction1, IdentityMatrix[2^4]];
  Hi2 = KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction2];
  interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2) .1]]
 ]
PossibleOrders = {
   ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8|⟩,
   <|Q1 → 5, Q2 → 6, Q3 → 3, Q4 → 4, Q5 → 1, Q6 → 2, Q7 → 7, Q8 → 8|>
  };
ρi = NThermalQBit[{.2, .3, .2, .2, .5, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   (*RO = RandomOrder[];*)
   RO  = RandomChoice[PossibleOrders];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
```

```
        ],
       {qbit, Keys[P[qbitIDs]]}],
      {t, Length[DMS]}]
   ]];

Clear[EntropyOfQubits]
EntropyOfQubits = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      VNEntropy[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]]],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ListPlot[Transpose[EntropyOfQubits], AxesLabel → Automatic, PlotRange → All]
EntropyOfQubits // MatrixPlot

ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]

ChangeinExtractablework // MatrixPlot

ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]

Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot

500 trials
```

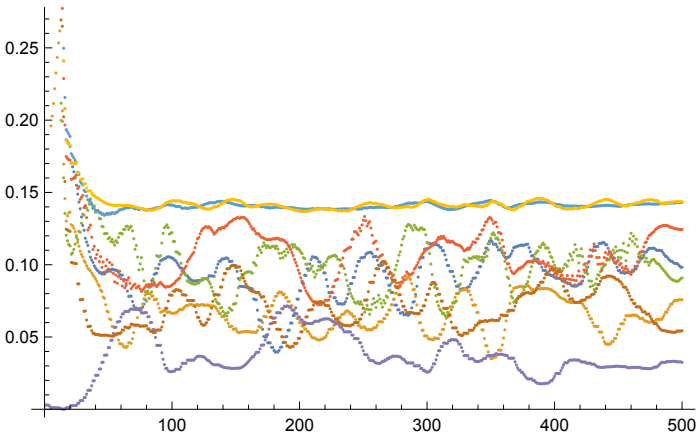
Out[ ]=



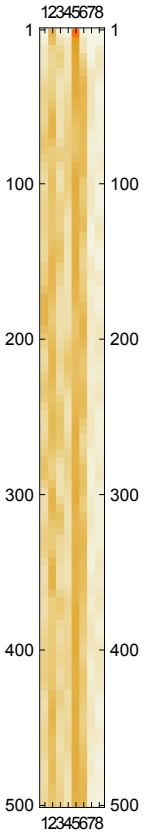
Out[ ]=


12345678
1
100
200
300
400
500
12345678

Out[ ]=

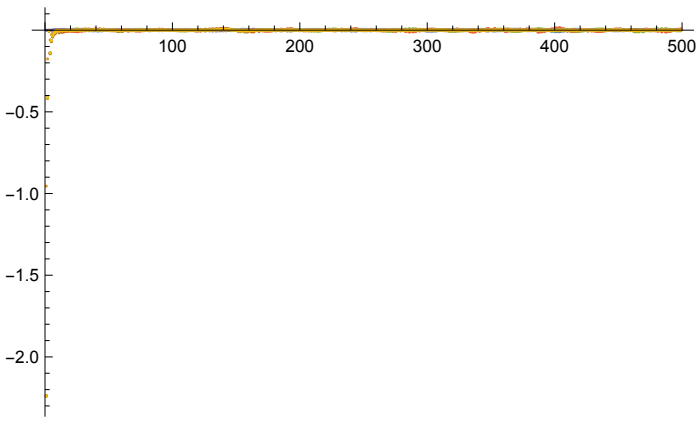

0.25
0.20
0.15
0.10
0.05
100
200
300
400
500

Out[ ]=

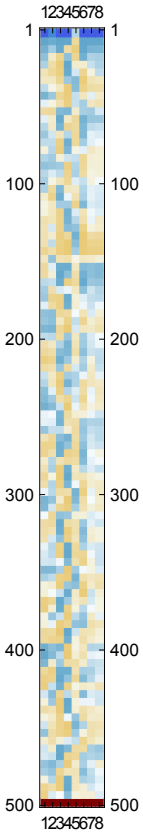



**500 trials**

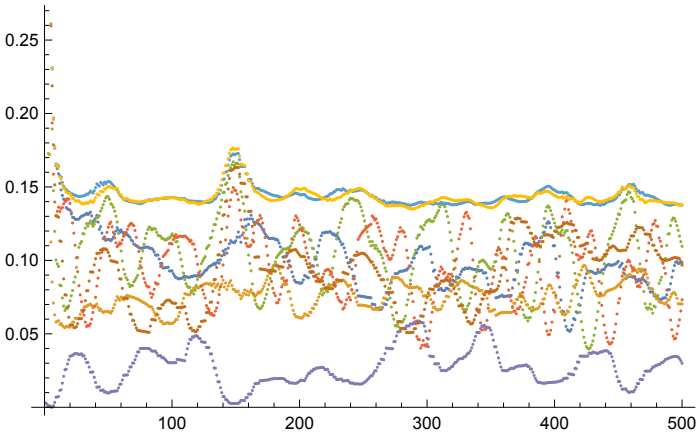
Out[ ]=



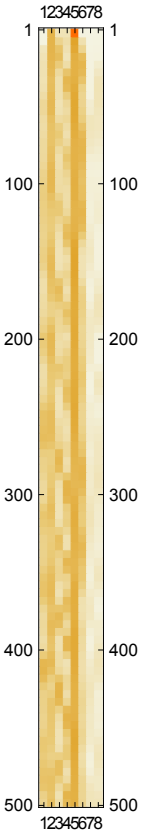
Out[ ]=


12345678
1
100
200
300
400
500
12345678

Out[ ]=


0.25
0.20
0.15
0.10
0.05
100
200
300
400
500

Out[•]=




# 12-Qubit gas model with one qubit temperature variation

In[•]:=

```
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12};
  vals = RandomSample[Table[i, {i, 12}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 12}]]
 ]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2, Hi3, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[2, 4];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^8]];
  Hi2 = KroneckerProduct[IdentityMatrix[2^8], SubsysInteraction];
  Hi3 =
   KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction, IdentityMatrix[2^4]];
  interactionUnitary = SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3) .1]]
 ]
Print[RandomUnitary]
```

SparseArray [ Specified elements: 97336
Dimensions: {4096, 4096} ]

Data not in notebook; Store now »

```
c \0
```

In[•]:=
```
ρi = NThermalQBit[{.2, .2, .2, .2, .2, .2, .4, .2, .2, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   RO = RandomOrder[];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 10}];
```

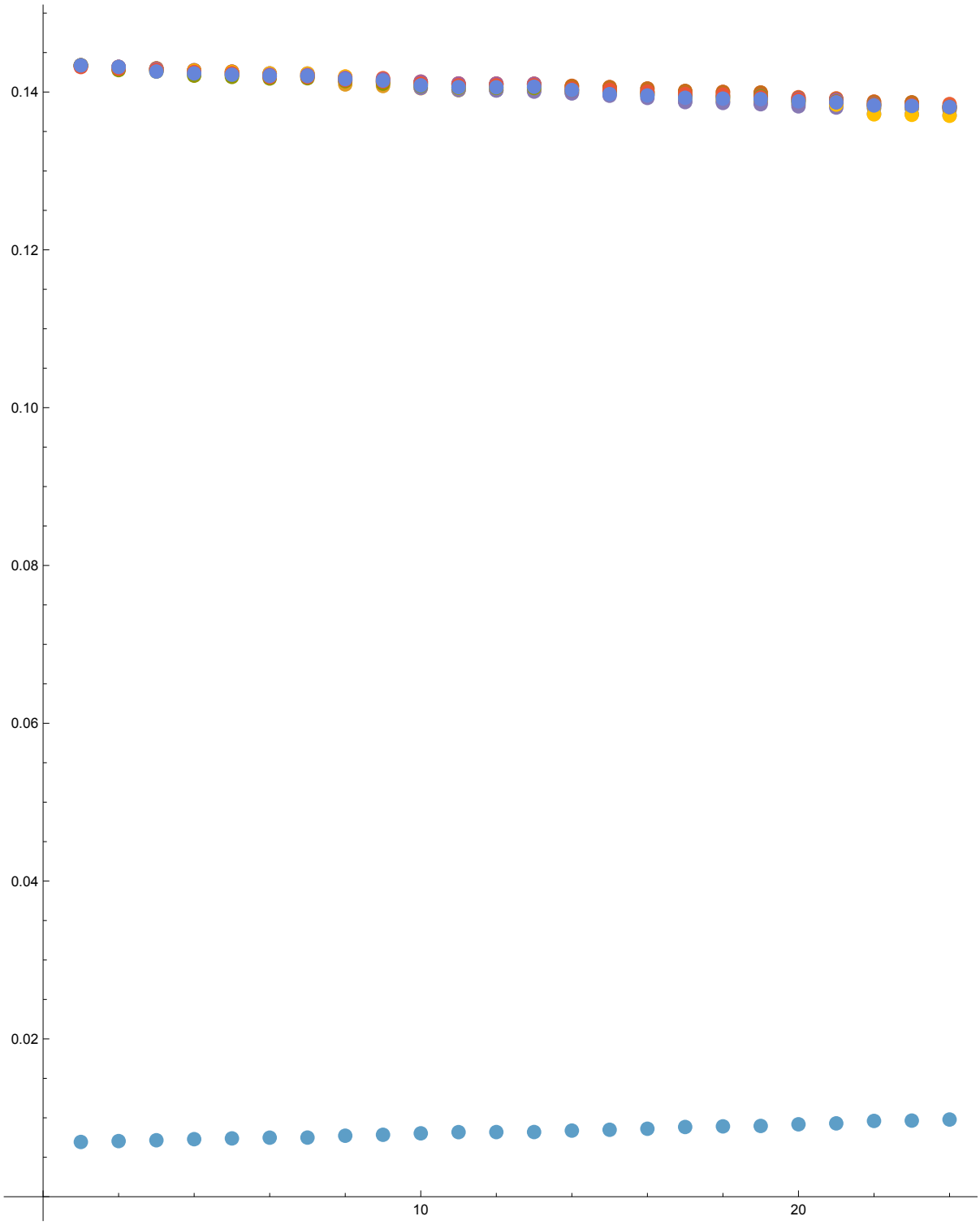
In[•]:=
```
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]],
   {P, DMS}];
```

In[•]:=
```
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
```

In[•]:=
```
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
```

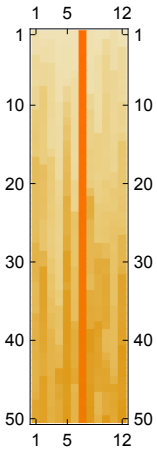
In[•]:=
```
ListPlot[Transpose[ExtractableWorkOverTime],
 AxesLabel → Automatic, PlotRange → All]
```

Out[•]=



In[ ]:=
```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

Out[ ]=




# 12 Qubits on a lattice

## Unitary translationally invariant (same everywhere at every time step)

### Two qubit pocket

### Four qubit pocket

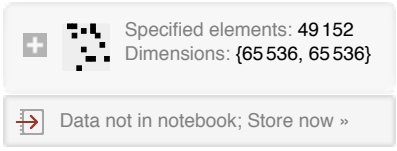
## Random unitary in each pocket

### Two qubit pocket

### Four qubit pocket

# 16-Qubit gas model with one qubit temperature variation

```
In[•]:= KroneckerProduct[RandomHamiltonian[1, 4], IdentityMatrix[2^12]]
```

```
Out[•]= SparseArray[ 

]
```

```
In[•]:= Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
  RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
  AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
 RandomOrder[] := Module[{items, vals},
   items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16};
   vals = RandomSample[Table[i, {i, 16}]];
   Association[Table[items[[i]] -> vals[[i]], {i, 16}]]
  ]
 RandomUnitary :=
  Module[{SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary},
   SubsysInteraction = RandomHamiltonian[1, 4];
   Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^12]];
   Hi2 =
    KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction, IdentityMatrix[2^8]];
   Hi3 = KroneckerProduct[IdentityMatrix[2^8],
     SubsysInteraction, IdentityMatrix[2^4]];
   Hi4 = KroneckerProduct[IdentityMatrix[2^12], SubsysInteraction];
   interactionUnitary = SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4) .1]]
  ]
 RandomUnitary
```

```
Out[•]= interactionUnitary
```

```
In[•]:= ρi = NThermalQBit[{.2, .2, .2, .2, .2, .2, .2, .2, .4, .2, .2, .2, .2, .2, .2, .2}];
 ρ = ρi;
 baseOrder = ρ[qbitIDs];
 interactionUnitary = RandomUnitary;
 DMS = Table[
     RO = RandomOrder[];
     Ui = MakeDM[interactionUnitary, RO];
     ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
     Round[ReOrder[ρ, baseOrder], 10^-15], {i, 10}];
```

```
In[•]:= AmbientTemps = Table[
          Association[Table[
            qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
          ,
          {P, DMS}];

In[•]:= AmbientPops = Table[
          Association[Table[
            qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
          {Temps, AmbientTemps}];

In[•]:= ExtractableWorkOverTime = N[Module[{P},
           Table[
            P = DMS[[t]];
            Table[
             AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
                ThermalQBit[AmbientPops[[t]][qbit]]
              ],
             {qbit, Keys[P[qbitIDs]]}],
            {t, Length[DMS]}]
          ]];
Out[•]= $Aborted
```

```
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]


ChangeinExtractablework // MatrixPlot



ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]


Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

Out[•]= $Aborted

Out[•]= ListPlot[Transpose[{}], AxesLabel → Automatic, PlotRange → All]

Out[•]=

Out[•]= ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]

Out[•]= $Aborted

# 16-Qubit gas model with multiple qubit

# temperature variation

```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
 RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16};
  vals = RandomSample[Table[i, {i, 16}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 16}]]
 ]
RandomUnitary :=
 Module[{SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[1, 4];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^12]];
  Hi2 =
   KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction, IdentityMatrix[2^8]];
  Hi3 = KroneckerProduct[IdentityMatrix[2^8],
    SubsysInteraction, IdentityMatrix[2^4]];
  Hi4 = KroneckerProduct[IdentityMatrix[2^12], SubsysInteraction];
  interactionUnitary =  SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4) .1]]
 ]
ρi = NThermalQBit[{.2, .4, .2, .2, .2, .2, .2, .2, .4, .2, .2, .2, .2, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   RO = RandomOrder[];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
```

```
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]


ChangeinExtractablework // MatrixPlot



ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]


Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

# 16-Qubit gas model with two qubit nearest neighbour interaction (one temp variation)

```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, Hi5, Hi6, Hi7, Hi8,
 interactionUnitary, RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2,
   Hi3, Hi4, Hi5, Hi6, Hi7, Hi8, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[1, 2];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^14]];
  Hi2 = KroneckerProduct[
    IdentityMatrix[2^2], SubsysInteraction, IdentityMatrix[2^12]];
  Hi3 = KroneckerProduct[IdentityMatrix[2^4],
    SubsysInteraction, IdentityMatrix[2^10]];
  Hi4 = KroneckerProduct[IdentityMatrix[2^6],
    SubsysInteraction, IdentityMatrix[2^8]];
  Hi5 = KroneckerProduct[IdentityMatrix[2^8],
    SubsysInteraction, IdentityMatrix[2^6]];
  Hi6 = KroneckerProduct[IdentityMatrix[2^10],
    SubsysInteraction, IdentityMatrix[2^4]];
```

```
  Hi7 = KroneckerProduct[IdentityMatrix[2^12],
    SubsysInteraction, IdentityMatrix[2^2]];
  Hi8 = KroneckerProduct[IdentityMatrix[2^14], SubsysInteraction];
  interactionUnitary =
   SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4 + Hi5 + Hi6 + Hi7 + Hi8) .1]]
 ]
PossibleOrders = {
   ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8, Q9 → 9,
    Q10 → 10, Q11 → 11, Q12 → 12, Q13 → 13, Q14 → 14, Q15 → 15, Q16 → 16|⟩,
   <|Q1 → 16, Q2 → 1, Q3 → 2, Q4 → 3, Q5 → 4, Q6 → 5, Q7 → 6, Q8 → 7, Q9 → 8,
    Q10 → 9, Q11 → 10, Q12 → 11, Q13 → 12, Q14 → 13, Q15 → 14, Q16 → 15|>
  };
ρi = NThermalQBit[{.2, .2, .2, .2, .2, .2, .2, .2, .4, .2, .2, .2, .2, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   RO = RandomOrder[];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]
```

```
ChangeinExtractablework // MatrixPlot
```

```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

# 16-Qubit gas model with two qubit nearest neighbour interaction (multiple temperature variation)

```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, Hi5, Hi6, Hi7, Hi8,
 interactionUnitary, RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomUnitary := Module[{SubsysInteraction, Hi1, Hi2,
   Hi3, Hi4, Hi5, Hi6, Hi7, Hi8, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[1, 2];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^14]];
  Hi2 = KroneckerProduct[
    IdentityMatrix[2^2], SubsysInteraction, IdentityMatrix[2^12]];
  Hi3 = KroneckerProduct[IdentityMatrix[2^4],
    SubsysInteraction, IdentityMatrix[2^10]];
  Hi4 = KroneckerProduct[IdentityMatrix[2^6],
    SubsysInteraction, IdentityMatrix[2^8]];
  Hi5 = KroneckerProduct[IdentityMatrix[2^8],
    SubsysInteraction, IdentityMatrix[2^6]];
  Hi6 = KroneckerProduct[IdentityMatrix[2^10],
    SubsysInteraction, IdentityMatrix[2^4]];
  Hi7 = KroneckerProduct[IdentityMatrix[2^12],
    SubsysInteraction, IdentityMatrix[2^2]];
  Hi8 = KroneckerProduct[IdentityMatrix[2^14], SubsysInteraction];
  interactionUnitary =
   SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4 + Hi5 + Hi6 + Hi7 + Hi8) .1]]
 ]
PossibleOrders = {
```

```
    ⟨|Q1 → 1, Q2 → 2, Q3 → 3, Q4 → 4, Q5 → 5, Q6 → 6, Q7 → 7, Q8 → 8, Q9 → 9,
     Q10 → 10, Q11 → 11, Q12 → 12, Q13 → 13, Q14 → 14, Q15 → 15, Q16 → 16|⟩,
    <|Q1 → 16, Q2 → 1, Q3 → 2, Q4 → 3, Q5 → 4, Q6 → 5, Q7 → 6, Q8 → 7, Q9 → 8,
     Q10 → 9, Q11 → 10, Q12 → 11, Q13 → 12, Q14 → 13, Q15 → 14, Q16 → 15|>
   };
ρi = NThermalQBit[{.2, .2, .2, .2, .4, .2, .2, .2, .4, .2, .2, .2, .2, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   RO = RandomOrder[];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]


ChangeinExtractablework // MatrixPlot



ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

# 16-Qubits on a lattice

```
Clear[SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary,
 RandomUnitary, PossibleOrders, ρi, DMS, AmbientPops,
 AmbientTemps, ExtractableWorkOverTime, ChangeinExtractablework]
RandomOrder[] := Module[{items, vals},
  items = {Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16};
  vals = RandomSample[Table[i, {i, 16}]];
  Association[Table[items[[i]] -> vals[[i]], {i, 16}]]
 ]
RandomUnitary :=
 Module[{SubsysInteraction, Hi1, Hi2, Hi3, Hi4, interactionUnitary},
  SubsysInteraction = RandomHamiltonian[1, 4];
  Hi1 = KroneckerProduct[SubsysInteraction, IdentityMatrix[2^12]];
  Hi2 =
   KroneckerProduct[IdentityMatrix[2^4], SubsysInteraction, IdentityMatrix[2^8]];
  Hi3 = KroneckerProduct[IdentityMatrix[2^8],
    SubsysInteraction, IdentityMatrix[2^4]];
  Hi4 = KroneckerProduct[IdentityMatrix[2^12], SubsysInteraction];
  interactionUnitary = SparseArray[MatrixExp[I (Hi1 + Hi2 + Hi3 + Hi4) .1]]
 ]
PossibleOrders = {
   ⟨|Q1 → 1, Q2 → 2, Q3 → 13, Q4 → 14, Q5 → 3, Q6 → 4, Q7 → 15, Q8 → 16,
    Q9 → 6, Q10 → 7, Q11 → 10, Q12 → 11, Q13 → 5, Q14 → 9, Q15 → 8, Q16 → 12|⟩,
   <|Q1 → 1, Q2 → 4, Q3 → 13, Q4 → 16, Q5 → 2, Q6 → 3, Q7 → 4, Q8 → 15,
    Q9 → 5, Q10 → 6, Q11 → 9, Q12 → 10, Q13 → 7, Q14 → 8, Q15 → 11, Q16 → 12|>,
   ⟨|Q1 → 1, Q2 → 4, Q3 → 13, Q4 → 16, Q5 → 2, Q6 → 3, Q7 → 4, Q8 → 15,
    Q9 → 5, Q10 → 8, Q11 → 9, Q12 → 12, Q13 → 6, Q14 → 7, Q15 → 10, Q16 → 11|⟩,
   <|Q1 → 1, Q2 → 4, Q3 → 5, Q4 → 8, Q5 → 2, Q6 → 3, Q7 → 6, Q8 → 7, Q9 → 9,
    Q10 → 12, Q11 → 13, Q12 → 16, Q13 → 10, Q14 → 11, Q15 → 14, Q16 → 15|>,
   ⟨|Q1 → 1, Q2 → 2, Q3 → 13, Q4 → 14, Q5 → 3, Q6 → 4, Q7 → 15, Q8 → 16, Q9 → 5,
    Q10 → 6, Q11 → 9, Q12 → 10, Q13 → 7, Q14 → 8, Q15 → 11, Q16 → 12|⟩,
   <|Q1 → 1, Q2 → 4, Q3 → 13, Q4 → 16, Q5 → 2, Q6 → 3, Q7 → 6, Q8 → 7, Q9 → 5,
    Q10 → 9, Q11 → 8, Q12 → 12, Q13 → 10, Q14 → 11, Q15 → 14, Q16 → 15|>,
   ⟨|Q1 → 3, Q2 → 4, Q3 → 15, Q4 → 16, Q5 → 1, Q6 → 2, Q7 → 5, Q8 → 6, Q9 → 9,
    Q10 → 10, Q11 → 13, Q12 → 14, Q13 → 7, Q14 → 8, Q15 → 11, Q16 → 12|⟩,
   <|Q1 → 1, Q2 → 4, Q3 → 5, Q4 → 8, Q5 → 2, Q6 → 3, Q7 → 6, Q8 → 7, Q9 → 9,
    Q10 → 10, Q11 → 13, Q12 → 14, Q13 → 11, Q14 → 12, Q15 → 15, Q16 → 16|>,
```

```
    ⟨|Q1 → 1, Q2 → 2, Q3 → 13, Q4 → 14, Q5 → 5, Q6 → 6, Q7 → 9, Q8 → 10, Q9 → 11,
     Q10 → 12, Q11 → 15, Q12 → 16, Q13 → 3, Q14 → 4, Q15 → 7, Q16 → 8|⟩,
    <|Q1 → 2, Q2 → 3, Q3 → 14, Q4 → 15, Q5 → 1, Q6 → 4, Q7 → 5, Q8 → 8, Q9 → 9,
     Q10 → 12, Q11 → 13, Q12 → 16, Q13 → 6, Q14 → 7, Q15 → 10, Q16 → 11|>,
    ⟨|Q1 → 1, Q2 → 2, Q3 → 5, Q4 → 6, Q5 → 3, Q6 → 4, Q7 → 7, Q8 → 8, Q9 → 9,
     Q10 → 10, Q11 → 13, Q12 → 14, Q13 → 11, Q14 → 12, Q15 → 15, Q16 → 16|⟩,
    <|Q1 → 1, Q2 → 2, Q3 → 5, Q4 → 6, Q5 → 3, Q6 → 4, Q7 → 7, Q8 → 8, Q9 → 9,
     Q10 → 12, Q11 → 13, Q12 → 16, Q13 → 10, Q14 → 11, Q15 → 14, Q16 → 15|>
  };
ρi = NThermalQBit[{.2, .2, .2, .2, .2, .2, .2, .2, .4, .2, .2, .2, .2, .2, .2, .2}];
ρ = ρi;
baseOrder = ρ[qbitIDs];
interactionUnitary = RandomUnitary;
DMS = Table[
   RO = RandomOrder[];
   Ui = MakeDM[interactionUnitary, RO];
   ρ = Ui.ReOrder[ρ, RO].Transpose[Ui];
   Round[ReOrder[ρ, baseOrder], 10^-15], {i, 150}];
AmbientTemps = Table[
   Association[Table[
     qbit -> T[PTR[P, {qbit}]], {qbit, Keys[P[qbitIDs]]}]]
   ,
   {P, DMS}];
AmbientPops = Table[
   Association[Table[
     qbit -> PopFromTemp[Temps[qbit][[2]]], {qbit, Keys[Temps]}]],
   {Temps, AmbientTemps}];
ExtractableWorkOverTime = N[Module[{P},
    Table[
     P = DMS[[t]];
     Table[
      AmbientTemps[[t]][qbit] × D[PTR[P, DeleteCases[Keys[P[qbitIDs]], qbit]] ||
         ThermalQBit[AmbientPops[[t]][qbit]]
       ],
      {qbit, Keys[P[qbitIDs]]}],
     {t, Length[DMS]}]
   ]];
ChangeinExtractablework =
  Table[ExtractableWorkOverTime[[i + 1]] - ExtractableWorkOverTime[[i]],
   {i, Length[ExtractableWorkOverTime]}];
ListPlot[Transpose[ChangeinExtractablework],
 AxesLabel → Automatic, PlotRange → All]


ChangeinExtractablework // MatrixPlot
```

```
ListPlot[Transpose[ExtractableWorkOverTime], AxesLabel → Automatic]
```

```
Table[
  Table[
   N[Temp[PartialTrace[img, DeleteCases[Keys[img[qbitIDs]], Q]]], 5],
   {Q, Keys[img[qbitIDs]]}], {img, DMS}] // MatrixPlot
```

# Unitary translationally invariant (same everywhere at every time step)

## Two qubit pocket

## Four qubit pocket

In[•]:=
```
LatticePoints = {{q1, 1, 2, 5, 6}, {q2, 2, 3, 6, 7}, {q3, 3, 4, 7, 8},
    {q4, 4, 1, 8, 5}, {q5, 5, 6, 9, 10}, {q6, 6, 7, 10, 11}, {q7, 7, 8, 11, 12},
    {q8, 8, 5, 12, 9}, {q9, 9, 10, 13, 14}, {q10, 10, 11, 14, 15},
    {q11, 11, 12, 15, 16}, {q12, 12, 9, 16, 13}, {q13, 13, 14, 1, 2},
    {q14, 14, 15, 2, 3}, {q15, 15, 15, 3, 4}, {q16, 16, 13, 4, 1}};
x = RandomChoice[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}]
Print[x]
val = Mod[x, 4]
```

Out[•]=
```
4
```

```
4
```

Out[•]=
```
0
```

In[•]:=
```
Clear[Points, Sets]
```

In[•]:=

```
x = RandomChoice[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}];
Clear[Points, Sets]
For[Points = 1;
 Sets = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
 Print[x], Points < 4, Points++, If[Mod[x, 4] == 0,
  Sets = DeleteCases[DeleteCases[DeleteCases[DeleteCases[DeleteCases[
         DeleteCases[DeleteCases[DeleteCases[DeleteCases[Sets, Mod[x, 16]],
            Mod[x - 1, 16]], Mod[x - 3, 16]], Mod[x - 4, 16]], Mod[x + 4, 16]],
        Mod[x + 3, 16]], Mod[x - 5, 16]], Mod[x - 7, 16]], Mod[x + 1, 16]];
  p2 = RandomChoice[Sets];
  Print[p2], Null];
 If[Mod[x, 4] == 1, Sets = DeleteCases[DeleteCases[DeleteCases[DeleteCases[
        DeleteCases[DeleteCases[DeleteCases[DeleteCases[DeleteCases[Sets, Mod[x,
              16]], Mod[x - 1, 16]], Mod[x - 3, 16]], Mod[x - 4, 16]], Mod[x + 4, 16]],
        Mod[x + 3, 16]], Mod[x + 5, 16]], Mod[x + 7, 16]], Mod[x + 1, 16]];
  p2 = RandomChoice[Sets];
  Print[p2], Null];
 If[Mod[x, 4] == 2 || 3, Sets = DeleteCases[DeleteCases[DeleteCases[DeleteCases[
        DeleteCases[DeleteCases[DeleteCases[DeleteCases[DeleteCases[Sets, Mod[x,
              16]], Mod[x - 1, 16]], Mod[x - 3, 16]], Mod[x - 4, 16]], Mod[x + 4, 16]],
        Mod[x + 3, 16]], Mod[x - 5, 16]], Mod[x + 5, 16]], Mod[x + 1, 16]];
  p2 = RandomChoice[Sets];
  Print[p2], Null];
 x = p2;
 Clear[p2]]
```

```
6
4
14
16
```

```
If[val == 0,
 p2 = RandomChoice[DeleteCases[DeleteCases[DeleteCases[DeleteCases[DeleteCases[
        DeleteCases[DeleteCases[DeleteCases[DeleteCases[{1, 2, 3, 4, 5, 6, 7,
             8, 9, 10, 11, 12, 13, 14, 15, 16}, Mod[x, 16]], Mod[x - 1, 16]],
           Mod[x - 3, 16]], Mod[x - 4, 16]], Mod[x + 4, 16]], Mod[x + 3, 16]],
      Mod[x - 5, 16]], Mod[x - 7, 16]], Mod[x + 1, 16]]];
 Print[p2], Null]
Clear[p2]
If[Mod[x, 4] == 1,
 p2 = RandomChoice[DeleteCases[DeleteCases[DeleteCases[DeleteCases[
       DeleteCases[DeleteCases[DeleteCases[DeleteCases[DeleteCases[
            {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}, Mod[x, 16]],
          Mod[x - 1, 16]], Mod[x - 3, 16]], Mod[x - 4, 16]], Mod[x + 4, 16]],
       Mod[x + 3, 16]], Mod[x + 5, 16]], Mod[x + 7, 16]], Mod[x + 1, 16]]];
 Print[p2], Null]
Clear[p2]
```

Out[•]= 8

8

Out[•]= 0

2

# Random unitary in each pocket

## Two qubit pocket

## Four qubit pocket