Title: Designing a Quantum Transducer With Genetic Algorithms and Electron Transport Calculations

Speakers: Kevin Ryczko

Collection: Machine Learning for Quantum Design

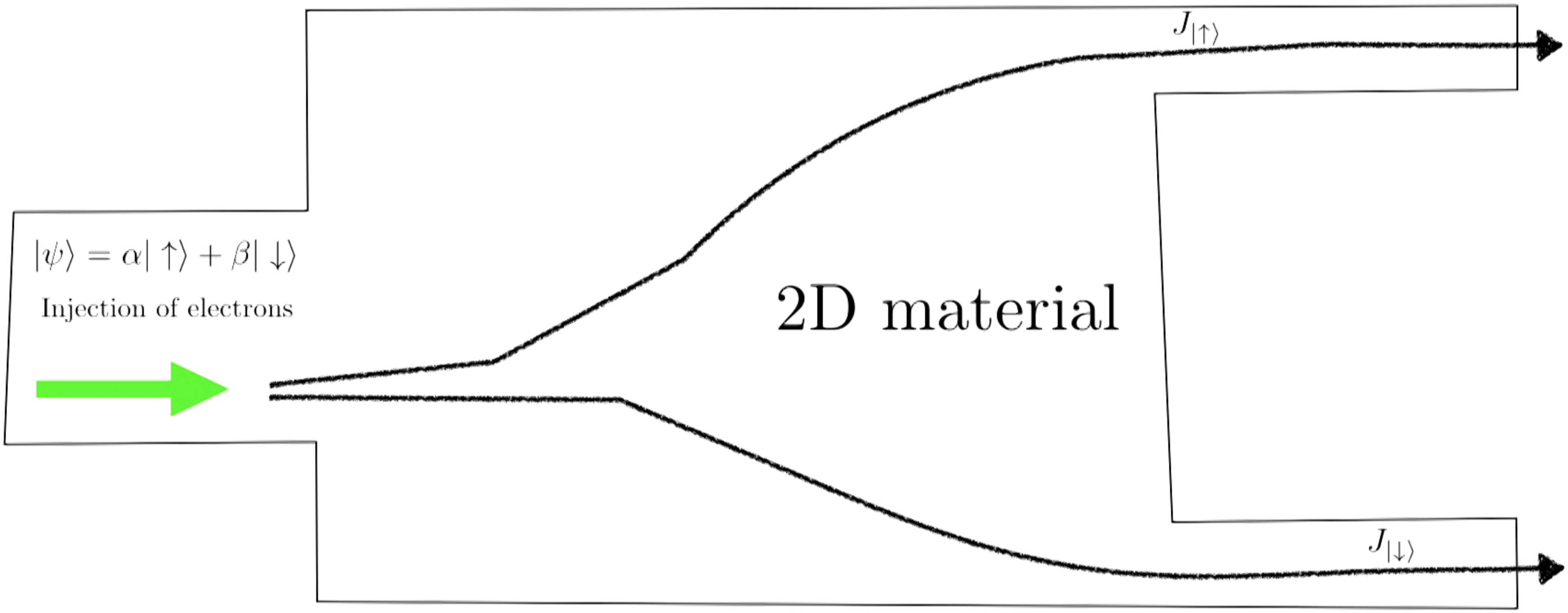
Date: July 08, 2019 - 2:30 PM

URL: http://pirsa.org/19070021

Abstract: The fields of quantum information and quantum computation are reliant on creating and maintaining low-dimensional quantum states. In two-dimensional hexagonal materials, one can describe a two-dimensional quantum state with electron quasi-momentum. This description, often referred to as valleytronics allows one to define a two-state vector labelled by k and k', which correspond to symmetric valleys in the conduction band. In this work, we present an algorithm that allows one to construct a nanoscale device that topologically separates k and k' current. Our algorithm incorporates electron transport calculations, artificial neural networks, and genetic algorithms to find structures that optimize a custom objective function. Our first result is that when modifying the on-site energies via doping with simple shapes the genetic algorithm is able to find structures that are able to topologically separate the valley currents with approximately 90% purity. We then introduce an arbitrary shape generator via a policy defined by an artificial neural network to modify the on-site energies of the nanoribbons. We study the dynamics of the genetic algorithms for both cases. Lastly, we then attempt to physically motivate the solutions by mapping the high dimensional search space to a lower dimensional one that can be better understood.

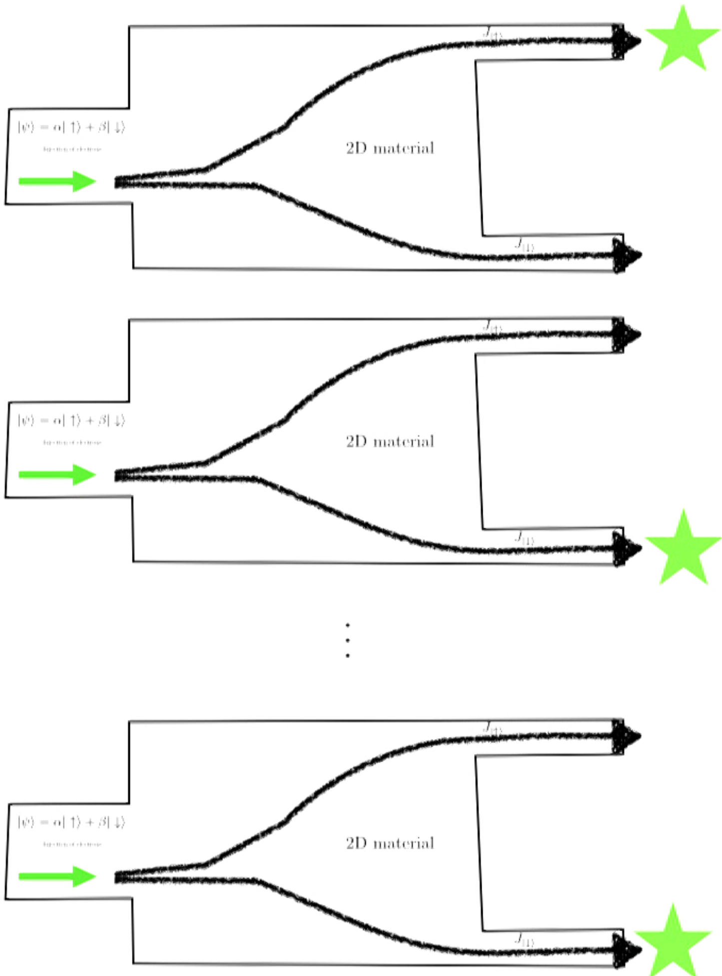
Designing a Quantum Transducer with Genetic Algorithms and Electron Transport Calculations

Kevin Ryczko, Pierre Darancet, Isaac Tamblyn

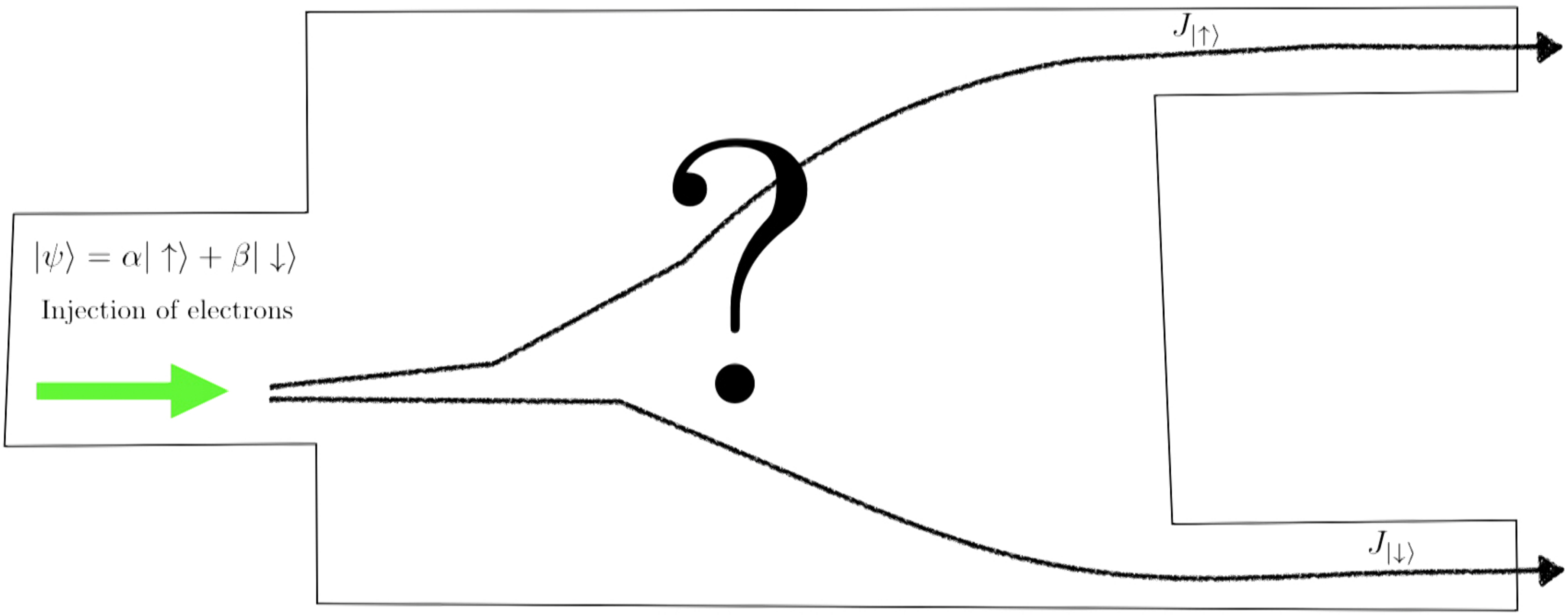


$$\mathbf{J}(\mathbf{r},t) = \frac{i\hbar}{2m}\left(\psi\nabla\psi^* - \psi^*\nabla\psi\right)$$


PERIMETER
PI
INSTITUT



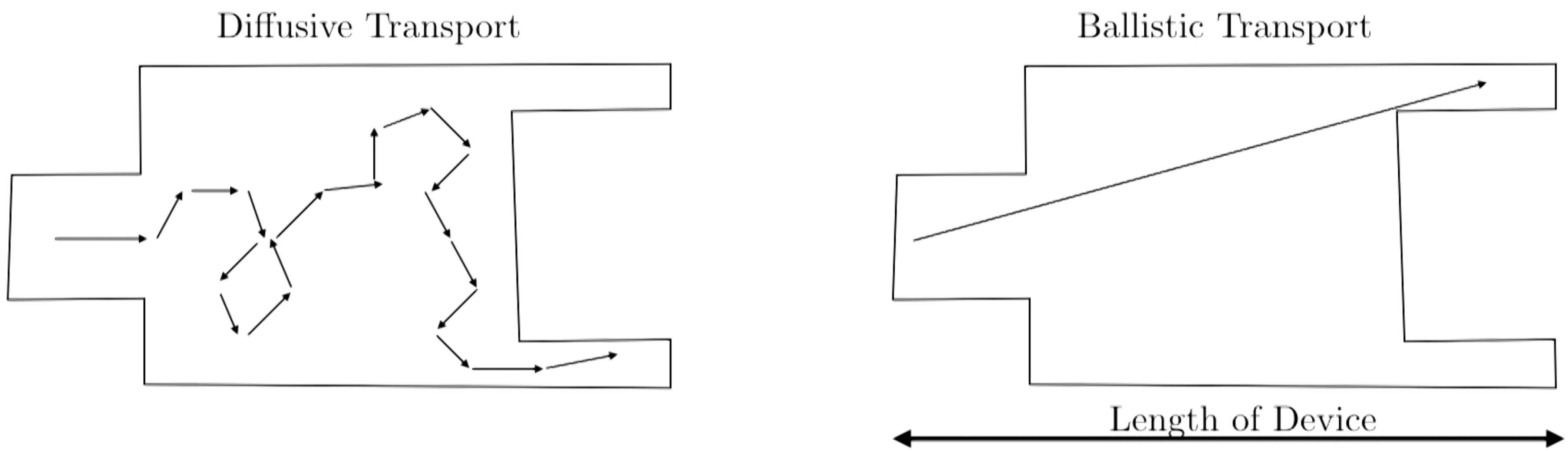
$$|\alpha|^2 = \frac{1}{3}, \quad |\beta|^2 = \frac{2}{3}$$



$$\mathbf{J}(\mathbf{r},t) = \frac{i\hbar}{2m}\left(\psi\nabla\psi^* - \psi^*\nabla\psi\right)$$

How do we calculate currents in the devices?

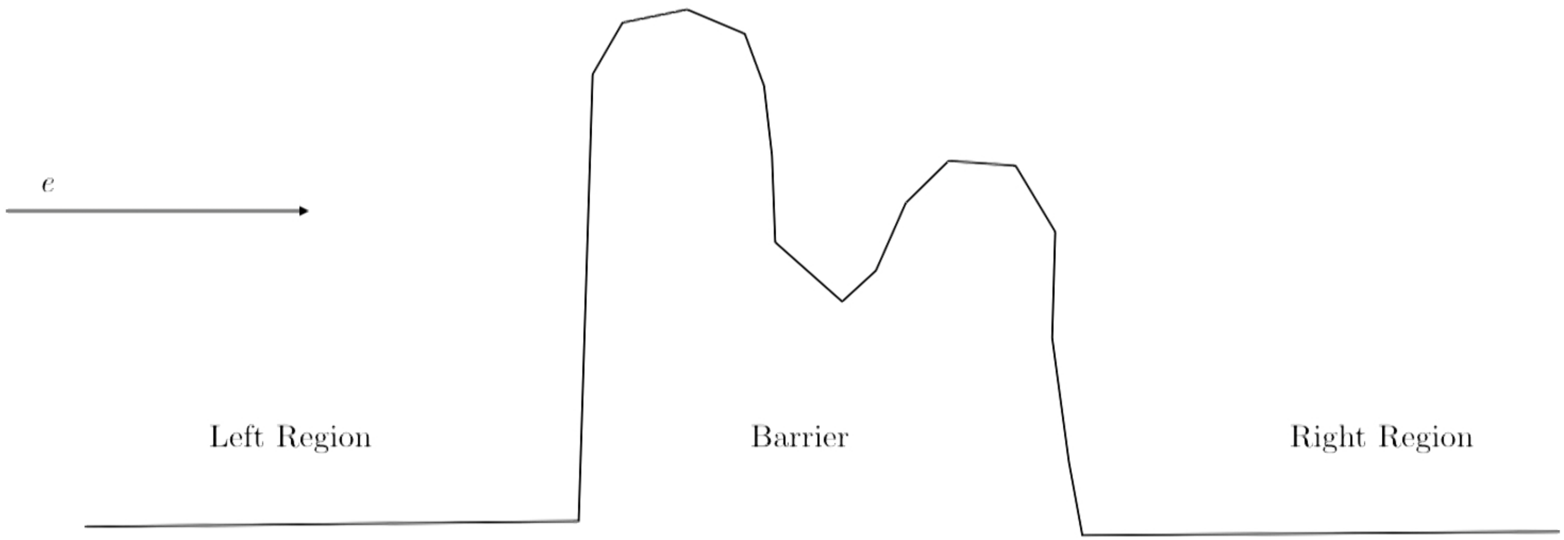
What Kind of Transport?



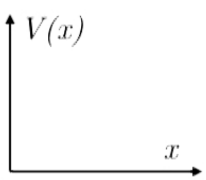
Type of Transport depends on 3 length scales:

1. The Fermi Wavelength (35 nm)
2. The Mean Free Path (30 μm)
3. Phase-Relaxation Length (30 μm) for high mobility semiconductors

Electron transport

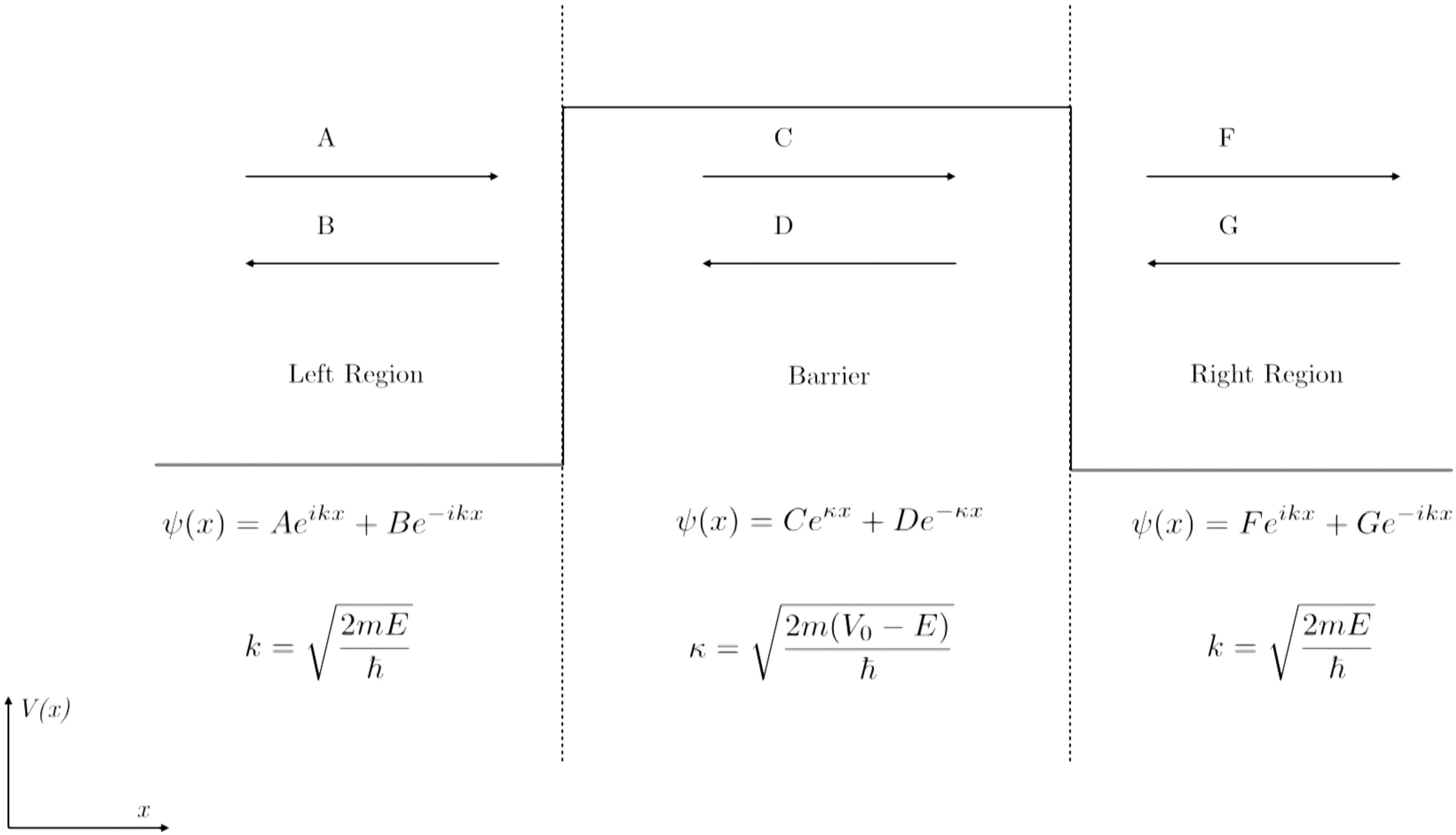


$$I_T = I_{\text{in}} T$$



The current comes from transmission through the barrier

Electron transport



$$\psi(x) = Ae^{ikx} + Be^{-ikx} \qquad \psi(x) = Ce^{\kappa x} + De^{-\kappa x} \qquad \psi(x) = Fe^{ikx} + Ge^{-ikx}$$

$$k = \sqrt{\frac{2mE}{\hbar}} \qquad \kappa = \sqrt{\frac{2m(V_0 - E)}{\hbar}} \qquad k = \sqrt{\frac{2mE}{\hbar}}$$

Tight Binding in 1D

$$\psi(x) = \sum_i c_i |\phi_i(x - x_i)\rangle \qquad\qquad H = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x)$$

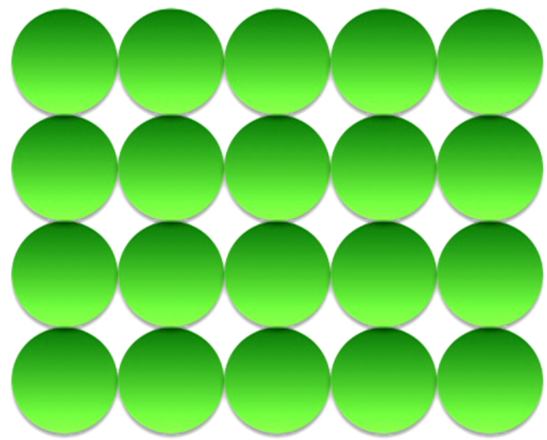
$$\left(\sum_j \langle \phi_j(x - x_j)|\right) H \left(\sum_i c_i |\phi_i(x - x_i)\rangle\right) = E \left(\sum_j \langle \phi_j(x - x_j)|\right) \left(\sum_i c_i |\phi_i(x - x_i)\rangle\rangle\right)$$

$$\mathbf{H}_{ji} = \int dx\ \phi_j^*(x - x_j) H \phi_i(x - x_i) = \begin{cases} h & \text{if } i = j \\ -t & \text{if } i \pm 1 = j \end{cases} \qquad\qquad \mathbf{S}_{ji} = \int dx\ \phi_j^*(x - x_j)\phi_i(x - x_i) = \delta_{ji}$$

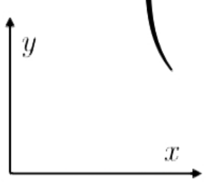
$$\xrightarrow{\qquad x \qquad}$$

Tight Binding and Electron Transport in 2D

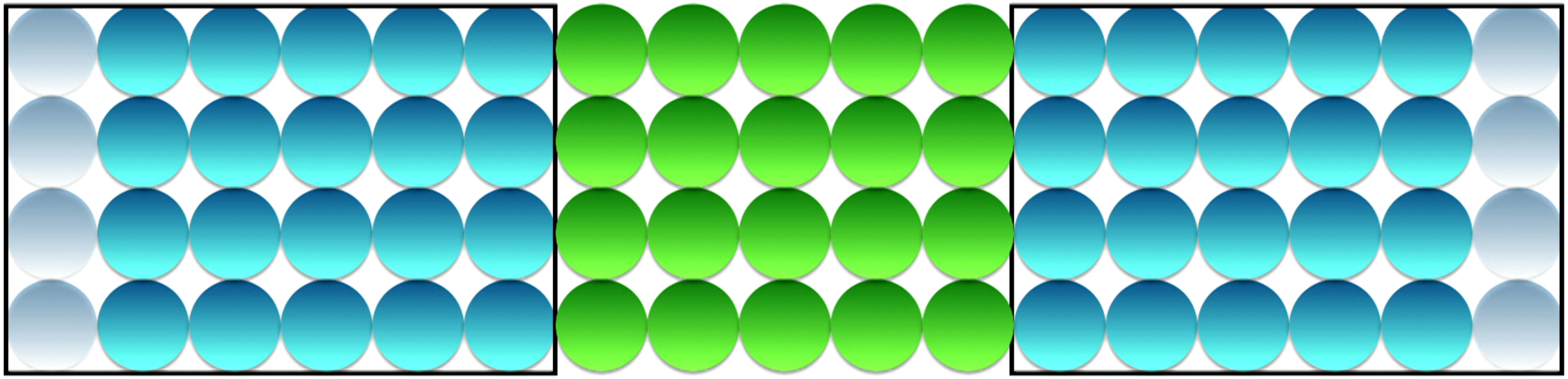
$$\boldsymbol{\psi}_j = \boldsymbol{c}_j |\phi(\boldsymbol{x}_j)\rangle$$



$$\begin{pmatrix} \boldsymbol{H_0} - E\boldsymbol{I} & \boldsymbol{T_0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{T_0^\dagger} & \boldsymbol{H_1} - E\boldsymbol{I} & \boldsymbol{T_1} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T_1^\dagger} & \boldsymbol{H_2} - E\boldsymbol{I} & \boldsymbol{T_2} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{T_{N-1}^\dagger} & \boldsymbol{H_N} - E\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_N \end{pmatrix} = \boldsymbol{0}$$



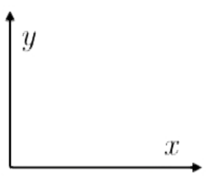
Tight Binding and Electron Transport in 2D



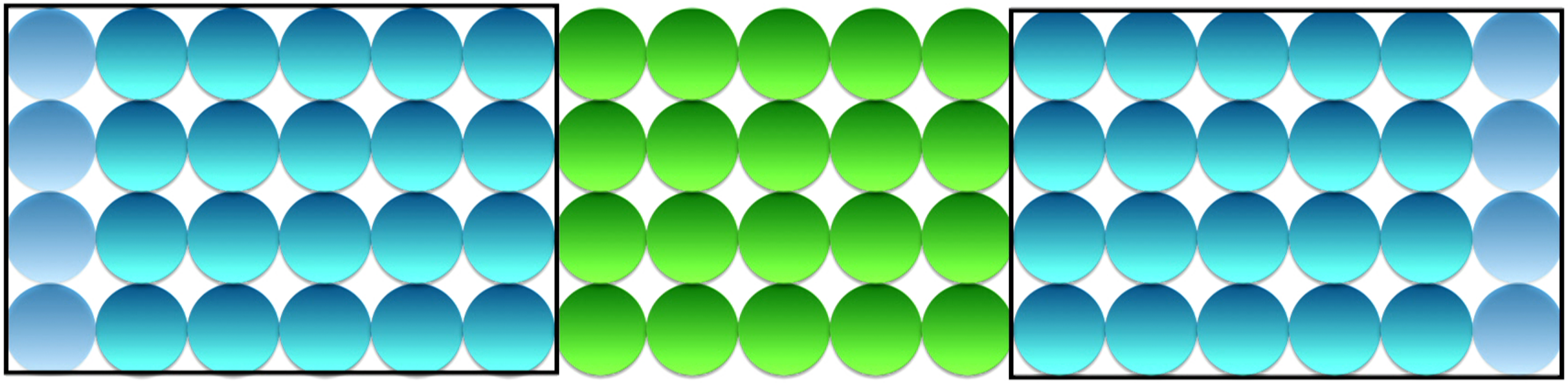
$$\boldsymbol{T}^{\dagger}_{\boldsymbol{n-1}}\boldsymbol{c}_{\boldsymbol{n-1}} + (\boldsymbol{H}_{\boldsymbol{n}} - E\boldsymbol{I})\boldsymbol{c}_{\boldsymbol{n}} + \boldsymbol{T}_{n}\boldsymbol{c}_{\boldsymbol{n+1}} = 0$$

$$\boldsymbol{c}_{\boldsymbol{n-1}} \equiv \boldsymbol{c}, \quad \boldsymbol{c}_{\boldsymbol{n}} = \boldsymbol{c}\lambda, \quad \boldsymbol{c}_{\boldsymbol{n+1}} = \boldsymbol{c}\lambda^2$$

$$\boldsymbol{T}^{\dagger}_{\boldsymbol{n-1}}\boldsymbol{c} + (\boldsymbol{H}_{\boldsymbol{n}} - E\boldsymbol{I})\boldsymbol{c}\lambda + \boldsymbol{T}_{n}\boldsymbol{c}\lambda^2 = 0$$



Tight Binding and Electron Transport in 2D



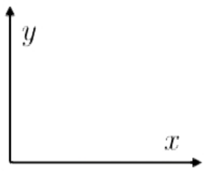
$$\boldsymbol{\zeta} = \boldsymbol{c}\lambda$$

$$\begin{pmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{T}^{\dagger}_{\boldsymbol{n-1}} & (\boldsymbol{H_n} - E\boldsymbol{I}) \end{pmatrix} + \lambda \begin{pmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{T}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{\zeta} \end{pmatrix} = \mathbf{0}$$

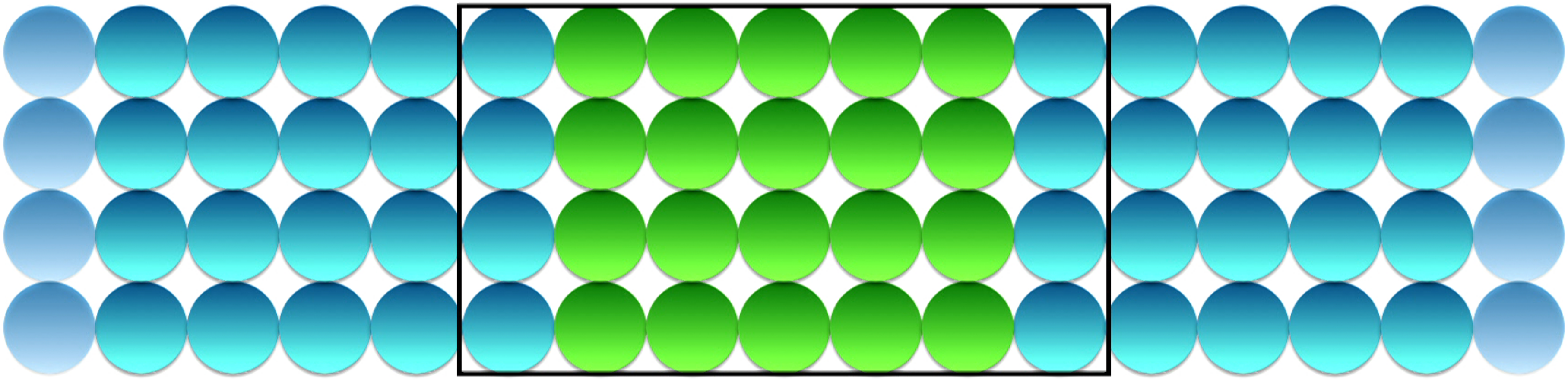
$$\boldsymbol{v}^{\dagger}_{\boldsymbol{k}} \tilde{\boldsymbol{v}}_{\boldsymbol{l}} = \tilde{\boldsymbol{v}}^{\dagger}_{\boldsymbol{k}} \boldsymbol{v}_{\boldsymbol{l}} = \delta_{k,l}$$

$$\lambda_k$$

$$\boldsymbol{\Gamma}^n = \sum_{k=1}^{N} \lambda_k^n \boldsymbol{v}_{\boldsymbol{k}} \tilde{\boldsymbol{v}}^{\dagger}_{\boldsymbol{k}}$$

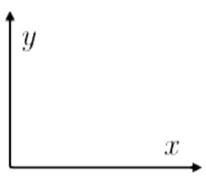


Tight Binding and Electron Transport in 2D

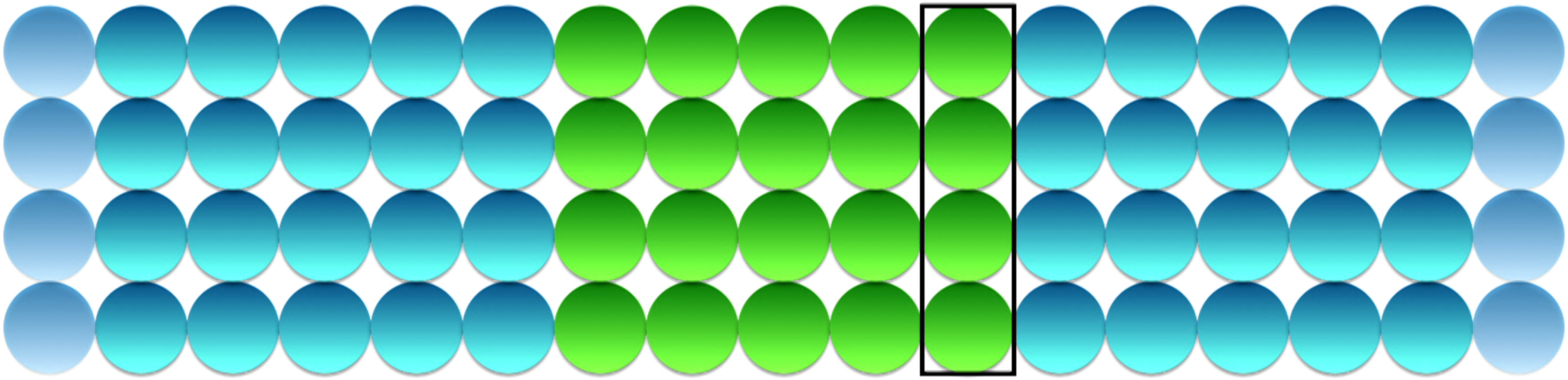


$$\begin{pmatrix} \boldsymbol{\Gamma}_{\leftarrow}^{-1}\boldsymbol{T}_{-1}^{\dagger} + \boldsymbol{H}_0 - E\boldsymbol{I} & \boldsymbol{T}_0 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{T}_0^{\dagger} & \boldsymbol{H}_1 - E\boldsymbol{I} & \boldsymbol{T}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{T}_1^{\dagger} & \boldsymbol{H}_2 - E\boldsymbol{I} & \boldsymbol{T}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \boldsymbol{T}_{N-1}^{\dagger} & \boldsymbol{H}_N - E\boldsymbol{I} + \boldsymbol{\Gamma}_{\rightarrow}\boldsymbol{T}_N \end{pmatrix}$$

$$\times \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_N \end{pmatrix} = \begin{pmatrix} (\boldsymbol{\Gamma}_{\rightarrow}^{-1} - \boldsymbol{\Gamma}_{\leftarrow}^{-1})\boldsymbol{T}_{-1}^{\dagger}\boldsymbol{c}_{0,\rightarrow} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

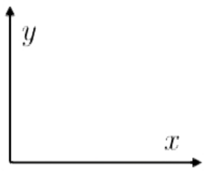


Tight Binding and Electron Transport in 2D



$$\boldsymbol{c_N} = \sum_{k,\rightarrow=1}^{N} t_k \boldsymbol{v_k}$$

$$t_{\ell,k} = \tilde{\boldsymbol{v}}_{\boldsymbol{\ell}}^{\dagger} \boldsymbol{c_N}.$$

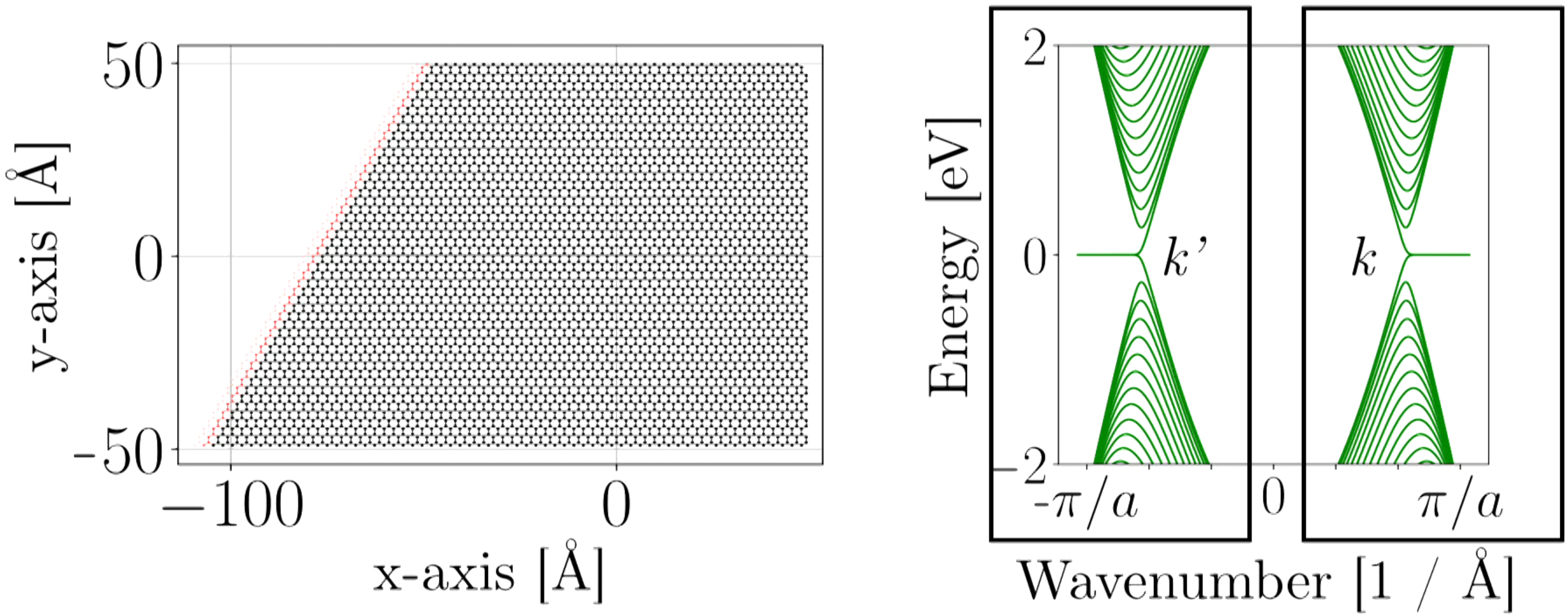


kwant Python Package

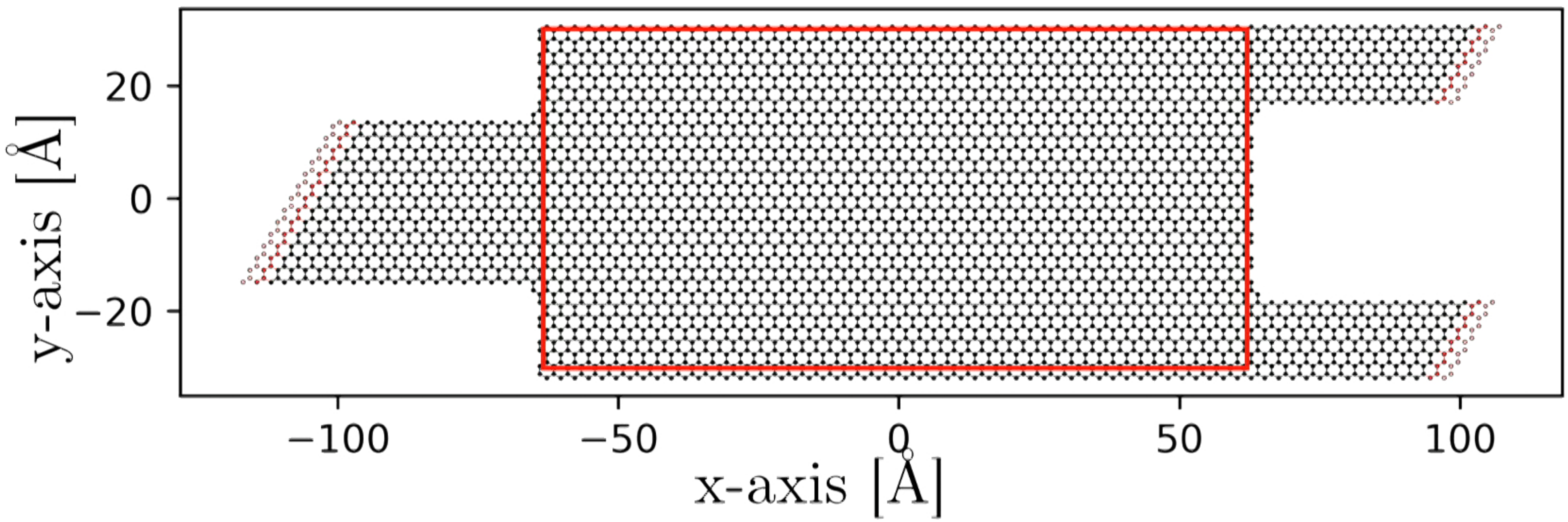
- Allows one to construct and solve for the transmission in a small number of lines of Python
- Well documented
- Growing community with a good mailing list
- Faster than other codes written in FORTRAN and C/C++
- Free!!

Tight Binding for Graphene

$$\mathbf{H}_{ji} = \int dx\ \phi_j^*(x - x_j) H \phi_i(x - x_i) = \begin{cases} 0 \text{ eV} & \text{if } i = j \\ -2.7 \text{ eV} & \text{if } i \pm 1 = j \end{cases}$$



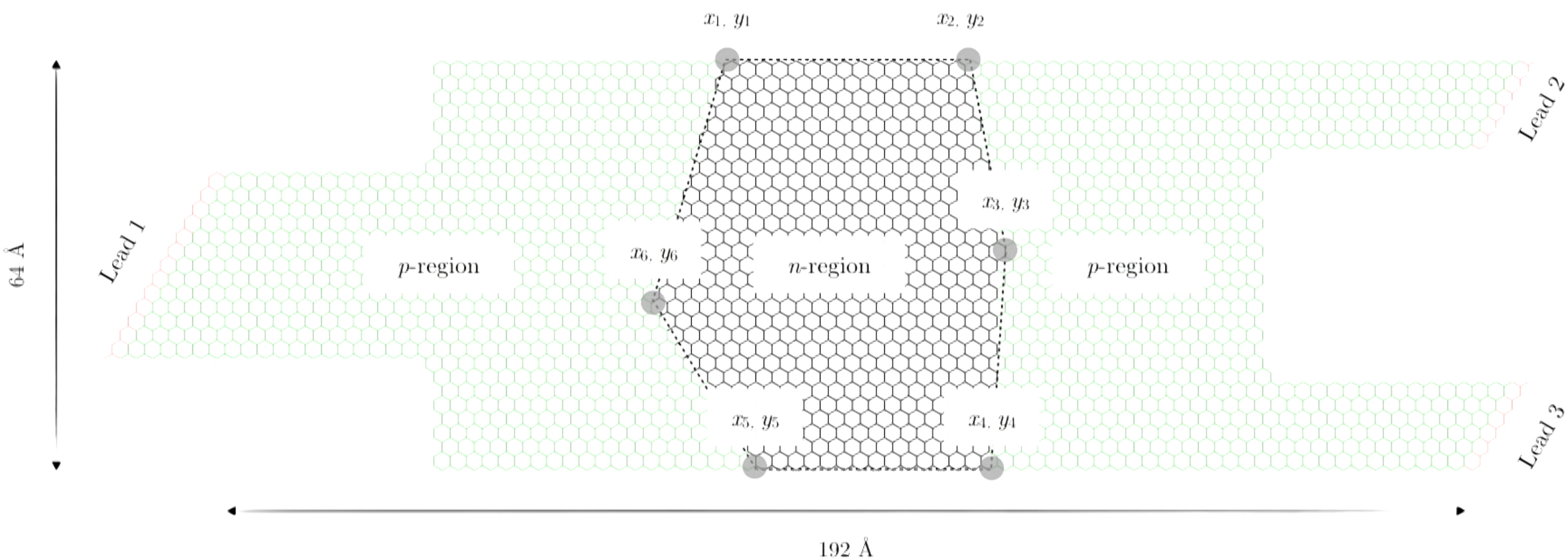
Design Space



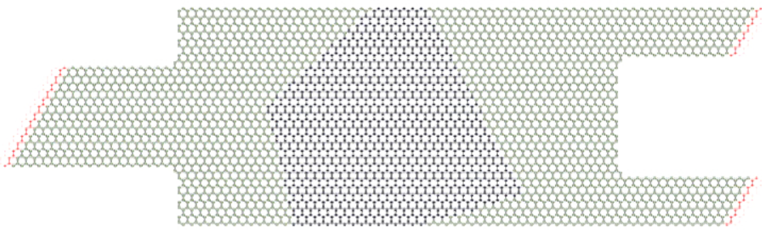
- Apply a gate voltage across (or doping) the device as demonstrated in [1], modifying the on-site energies of the atoms!
- But what atoms do we select? With 4000 orbitals (as shown above) there are 2^{4000} possibilities!

[1] Rycerz A, Tworzydło J, Beenakker CW. Valley filter and valley valve in graphene. Nature Physics. 2007 Mar;3(3):172.

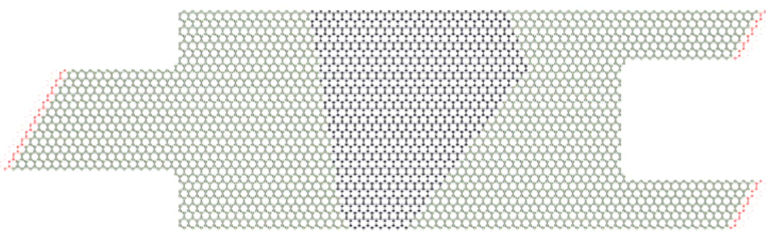
Simple Strategy - Drawing Polygons



Simple Strategy - Drawing Polygons

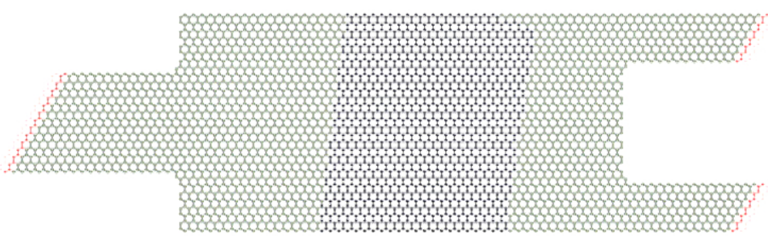


$$\begin{pmatrix} x_1, y_1 \\ x_2, y_2 \\ x_3, y_3 \\ x_4, y_4 \\ x_5, y_5 \\ x_6, y_6 \end{pmatrix}^{(0)} \longrightarrow \text{Valley Currents for device 0}$$



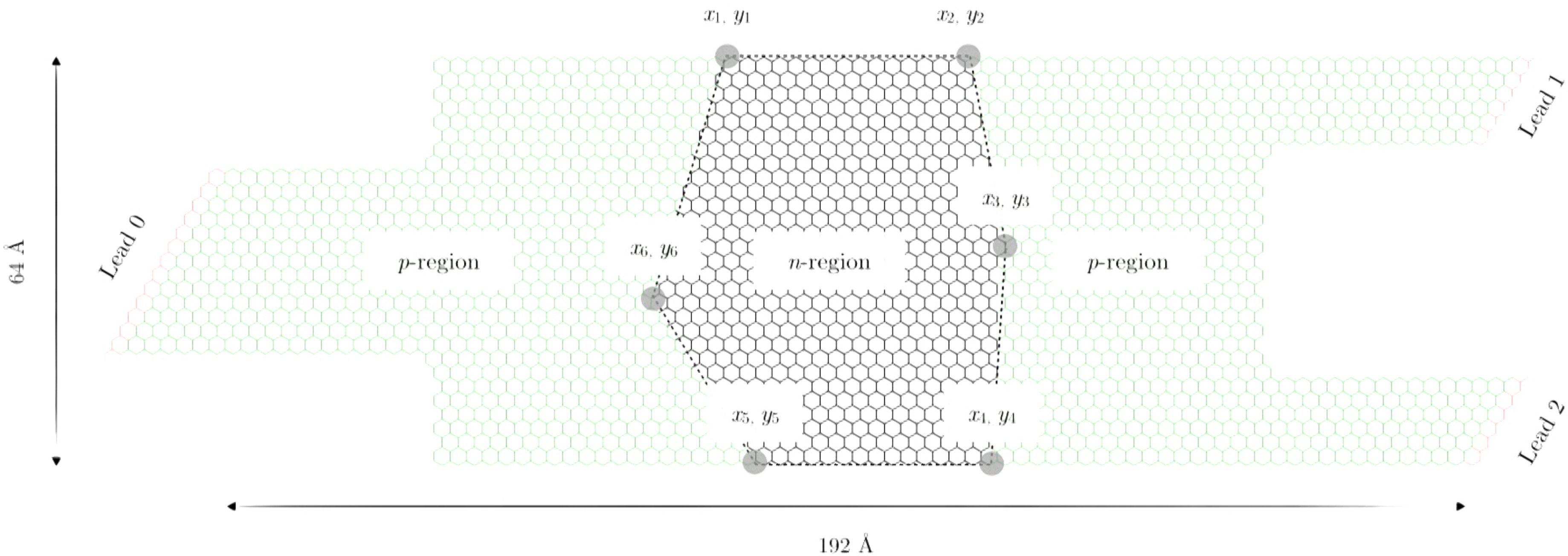
$$\begin{pmatrix} x_1, y_1 \\ x_2, y_2 \\ x_3, y_3 \\ x_4, y_4 \\ x_5, y_5 \\ x_6, y_6 \end{pmatrix}^{(1)} \longrightarrow \text{Valley Currents for device 1}$$

$$\vdots$$



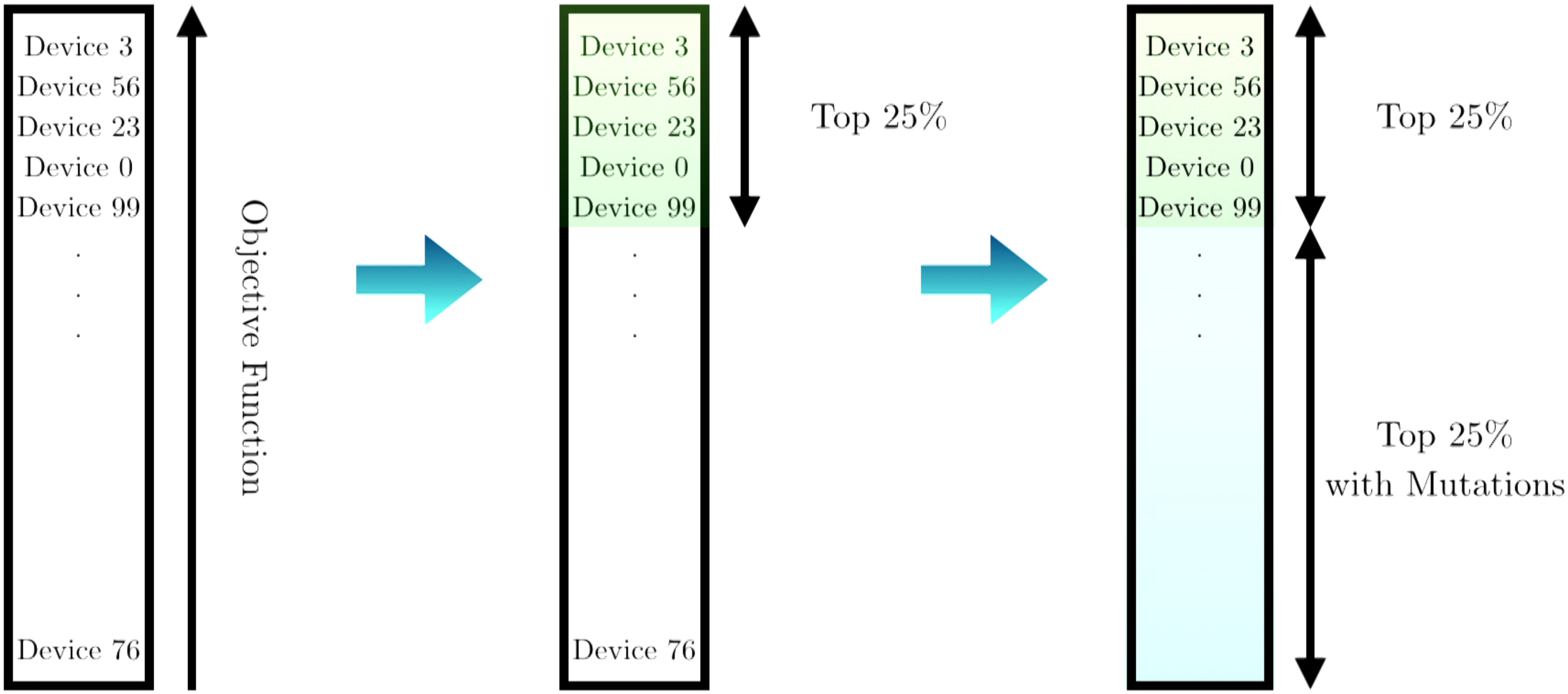
$$\begin{pmatrix} x_1, y_1 \\ x_2, y_2 \\ x_3, y_3 \\ x_4, y_4 \\ x_5, y_5 \\ x_6, y_6 \end{pmatrix}^{(\mathrm{N})} \longrightarrow \text{Valley Currents for device N}$$

Simple Strategy - Drawing Polygons

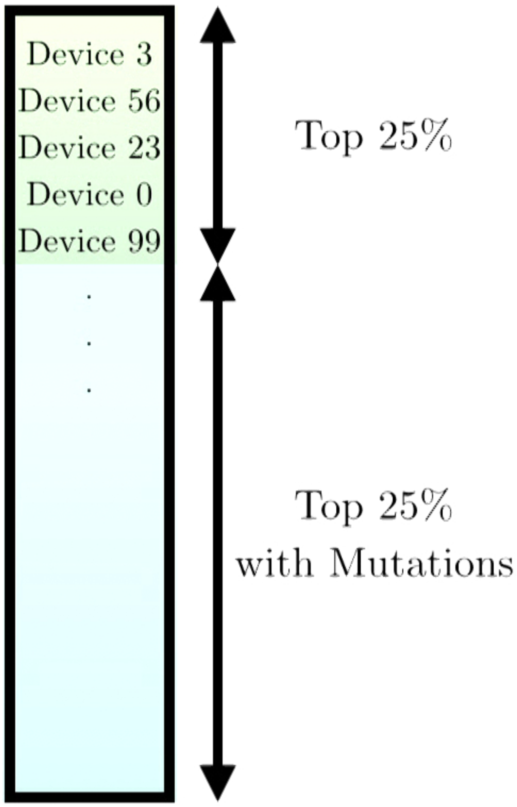


$$F(I_{k,1}, I_{k',1}, I_{k,2}, I_{k',2}) = \frac{I_{k',1}}{I_{k,1} + I_{k',1}} + \frac{I_{k,2}}{I_{k,2} + I_{k',2}} + (I_{k',1} + I_{k,2})$$

Simple Strategy - Drawing Polygons



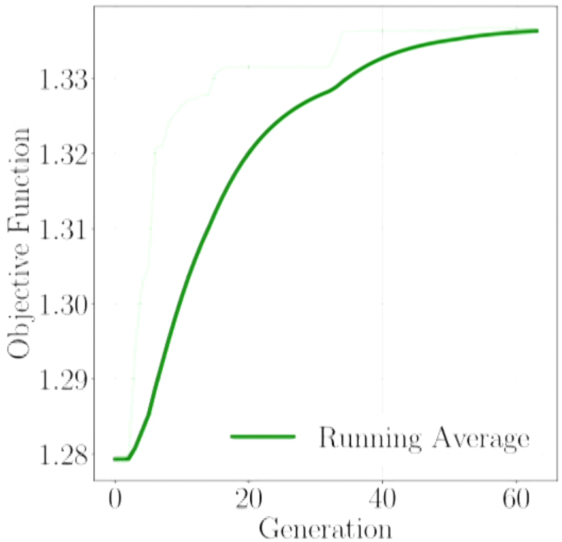
Simple Strategy - Drawing Polygons



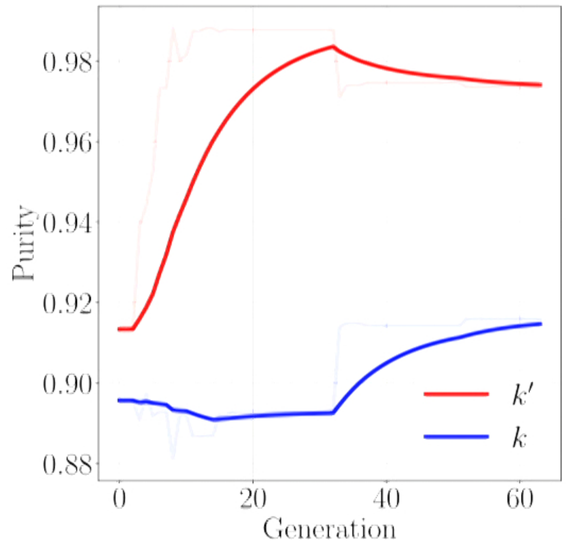
$$\begin{pmatrix} x_1, y_1 \\ x_2, y_2 \\ x_3, y_3 \\ x_4, y_4 \\ x_5, y_5 \\ x_6, y_6 \end{pmatrix} = \begin{pmatrix} x_1, y_1 \\ x_2, y_2 \\ x_3, y_3 \\ x_4, y_4 \\ x_5, y_5 \\ x_6, y_6 \end{pmatrix} + \alpha \begin{pmatrix} u_{11}, u_{12} \\ u_{21}, u_{22} \\ u_{31}, u_{32} \\ u_{41}, u_{42} \\ u_{51}, u_{52} \\ u_{61}, u_{62} \end{pmatrix}$$

$$\alpha = 0.1, \quad u_{ij} \in [-1, 1]$$

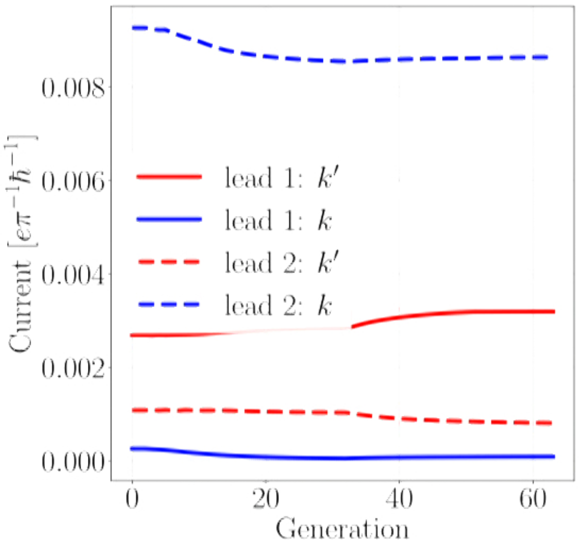
Results



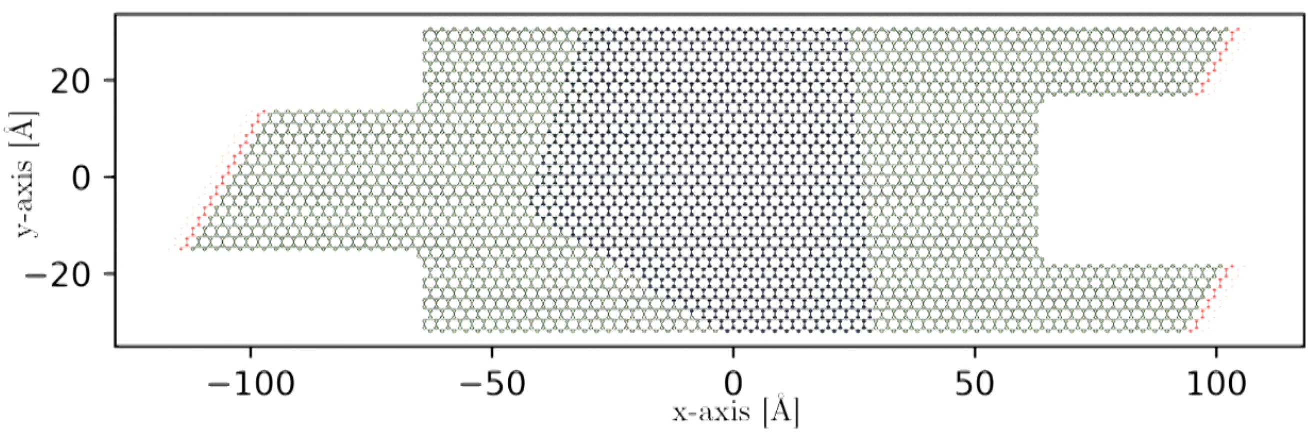
Objective Function
1.28
1.29
1.30
1.31
1.32
1.33
Running Average
0
20
40
60
Generation



Purity
0.88
0.90
0.92
0.94
0.96
0.98
k'
k
0
20
40
60
Generation

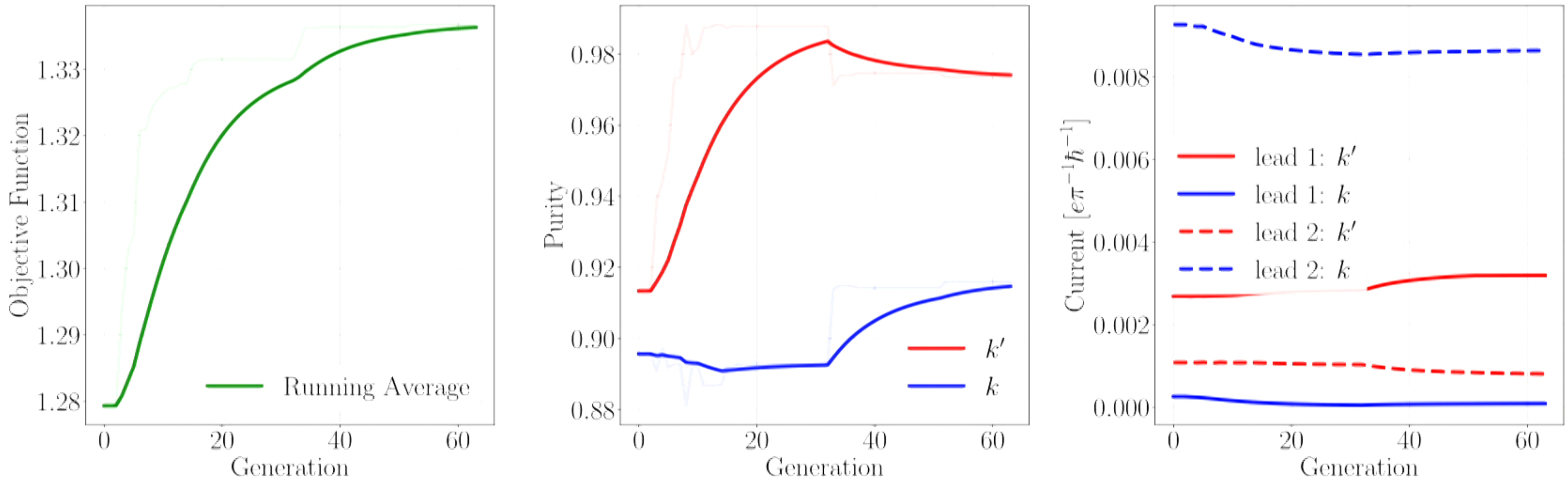


Current $[e\pi^{-1}\hbar^{-1}]$
0.000
0.002
0.004
0.006
0.008
lead 1: k'
lead 1: k
lead 2: k'
lead 2: k
0
20
40
60
Generation

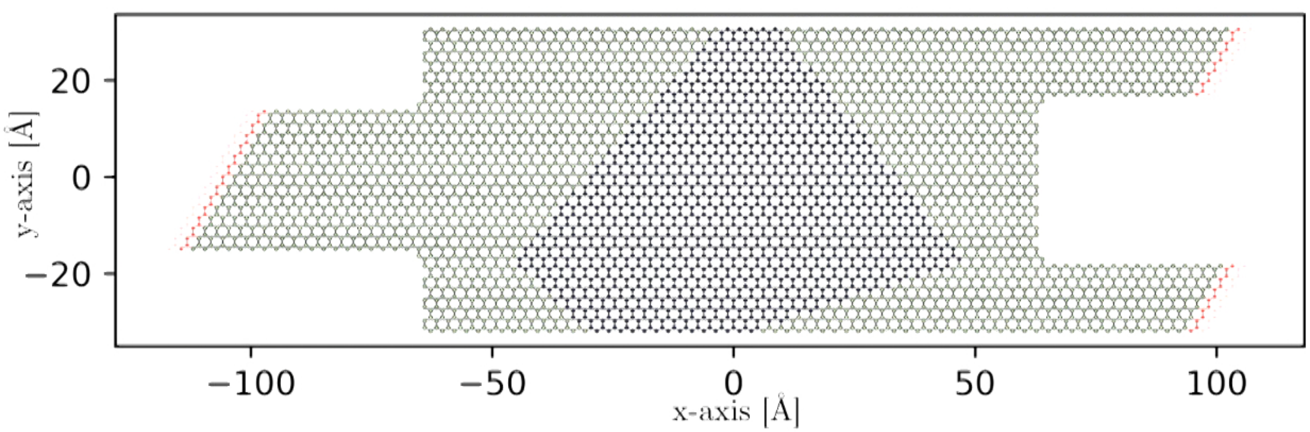


y-axis [Å]
20
0
−20
−100
−50
0
50
100
x-axis [Å]

Results

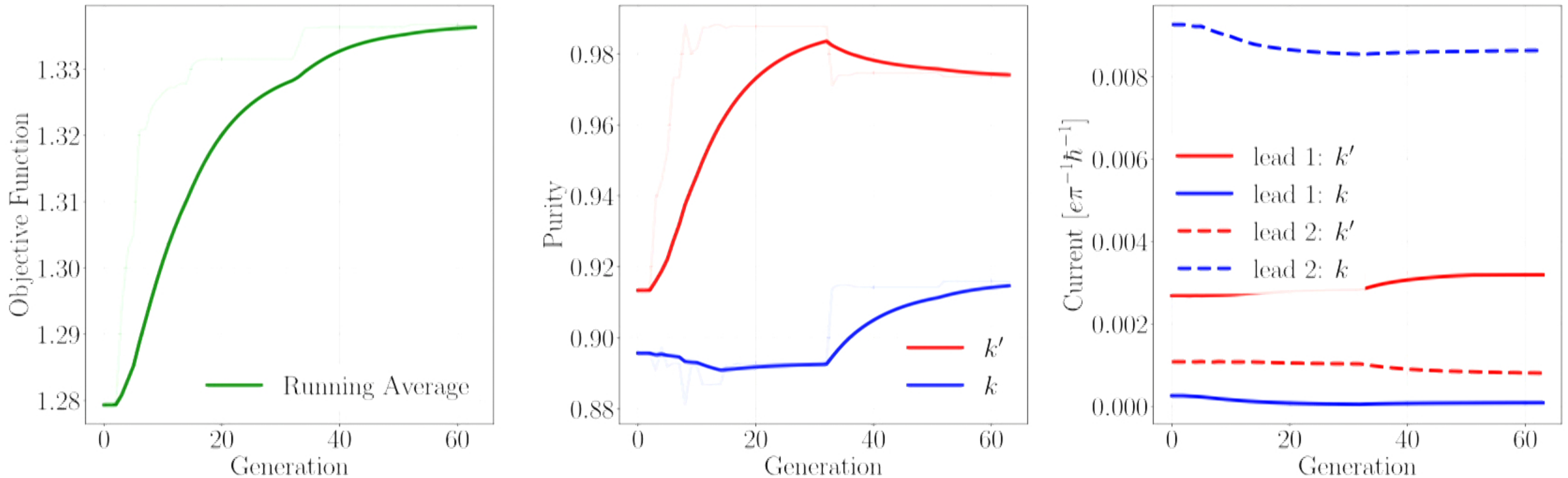


Objective Function
Running Average
Generation
Purity
k'
k
Current $[e\pi^{-1}\hbar^{-1}]$
lead 1: k'
lead 1: k
lead 2: k'
lead 2: k

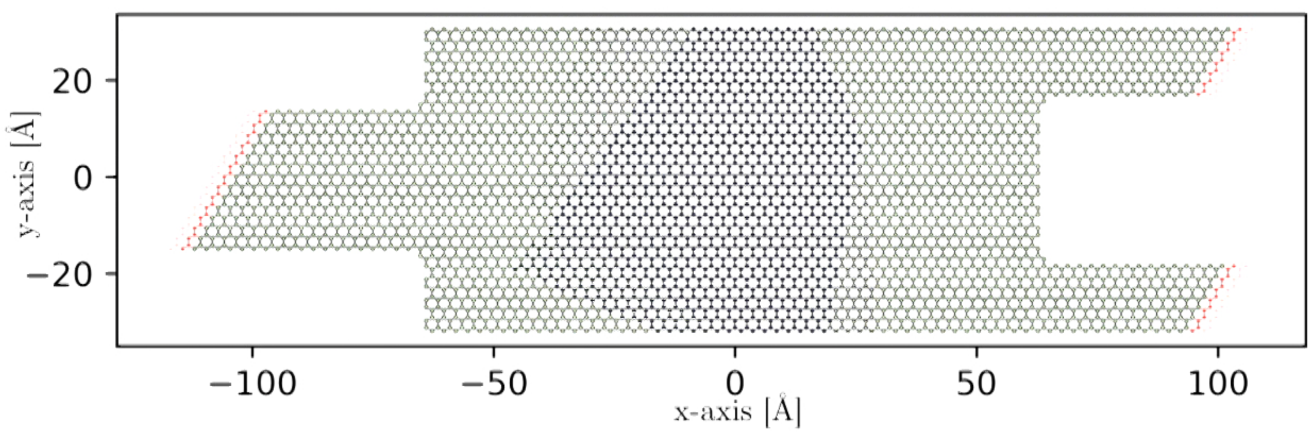


y-axis [Å]
x-axis [Å]

Results

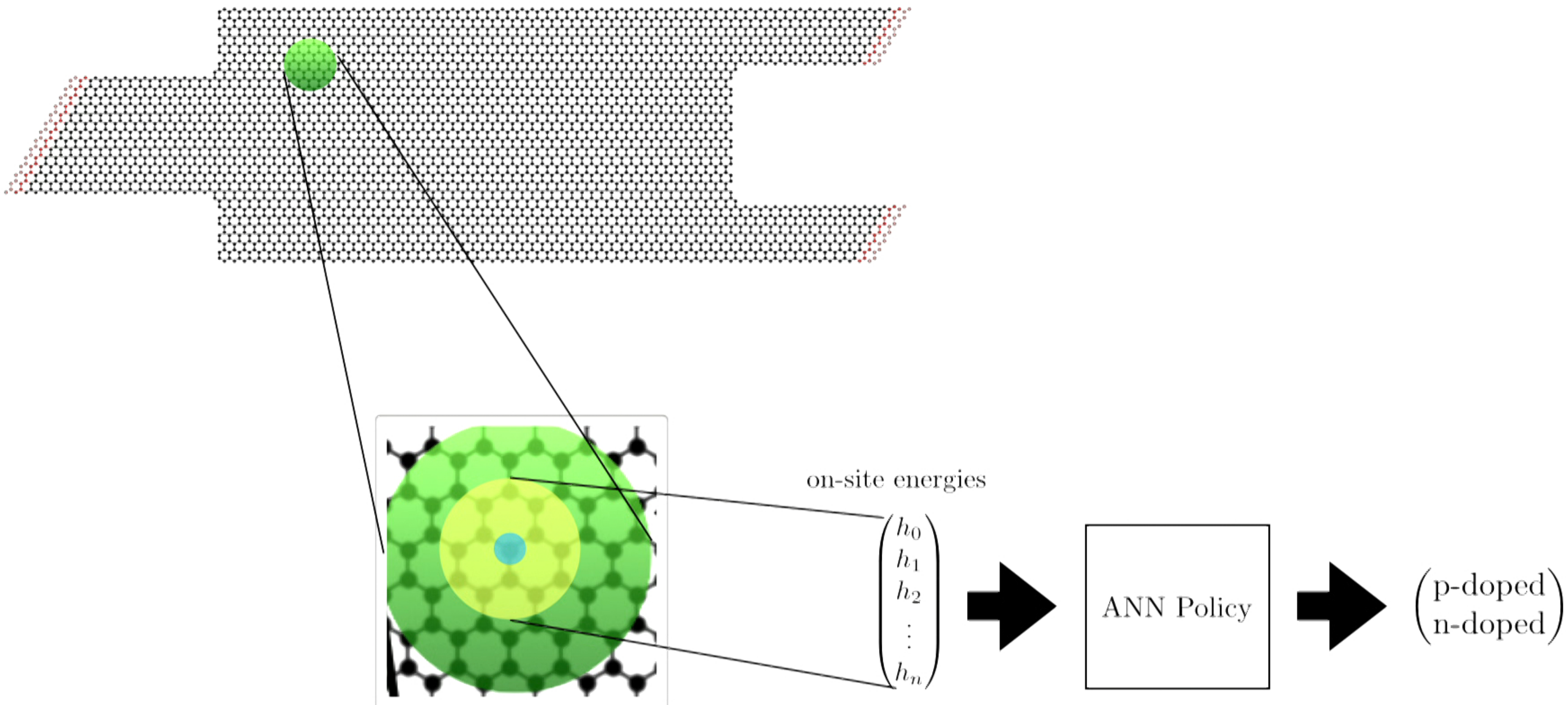


1.33
1.32
1.31
1.30
1.29
1.28
Objective Function
Running Average
0 20 40 60
Generation
0.98
0.96
0.94
0.92
0.90
0.88
Purity
k'
k
0 20 40 60
Generation
0.008
0.006
0.004
0.002
0.000
Current $[e\pi^{-1}\hbar^{-1}]$
lead 1: k'
lead 1: k
lead 2: k'
lead 2: k
0 20 40 60
Generation

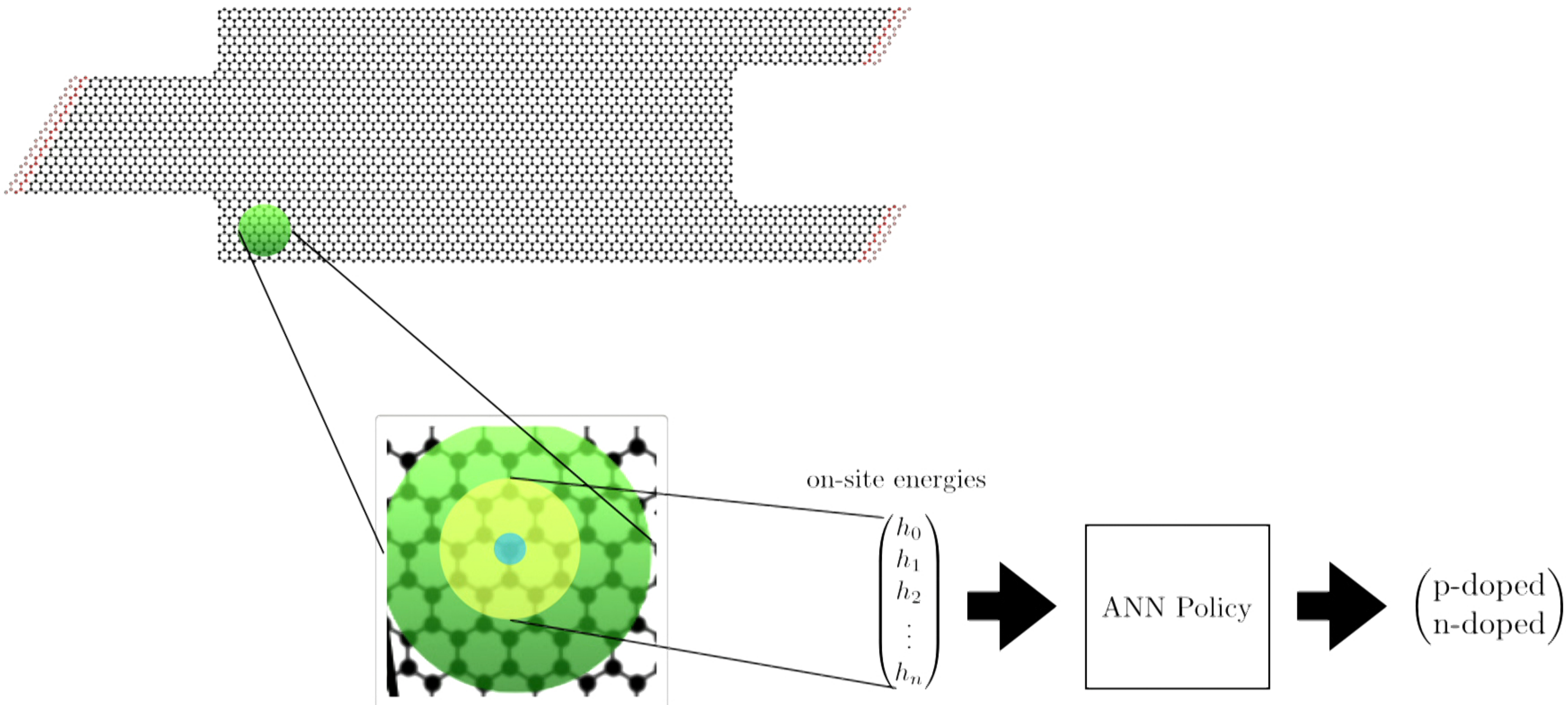


20
0
−20
y-axis [Å]
−100 −50 0 50 100
x-axis [Å]

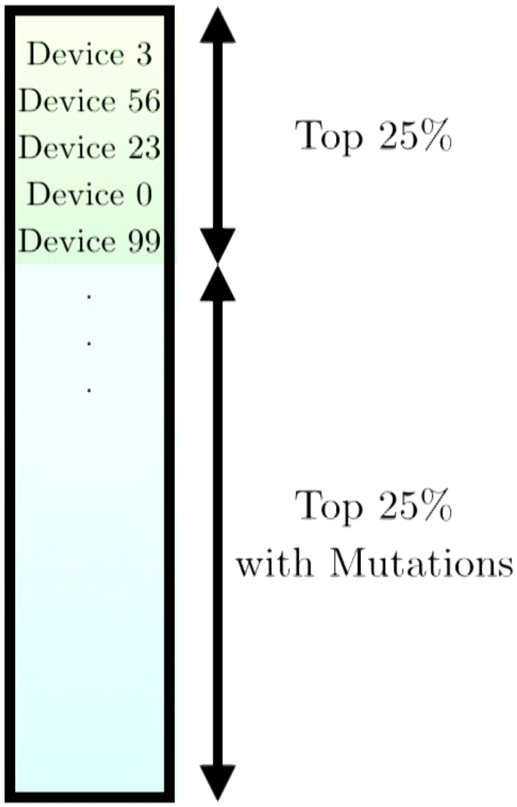
More Sophisticated Strategy - ANN Policy



More Sophisticated Strategy - ANN Policy



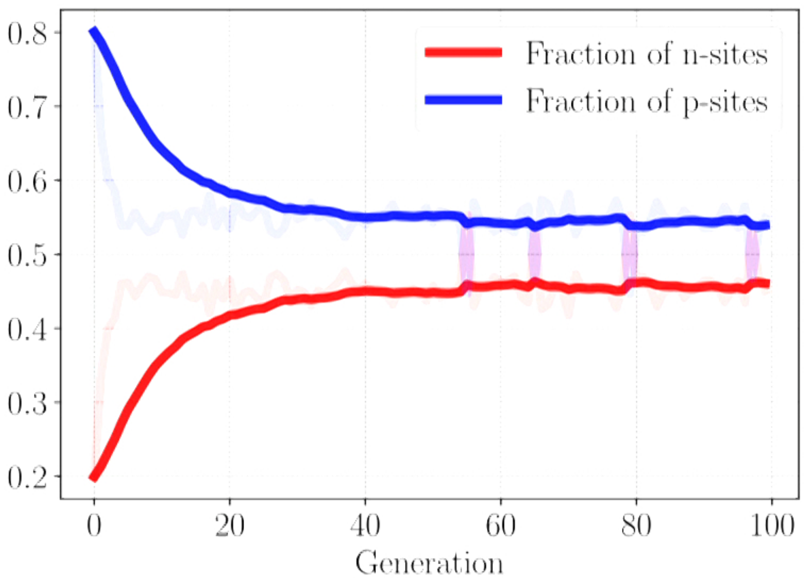
More Sophisticated Strategy - ANN Policy

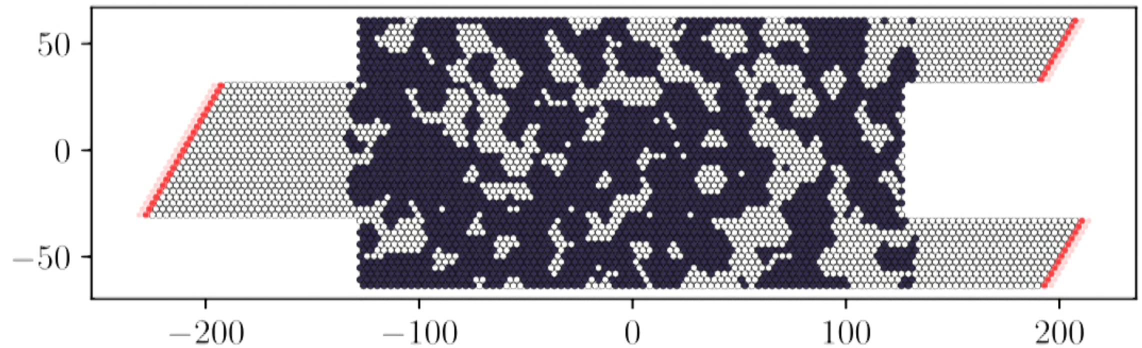


$$\begin{pmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ w_{N1} & w_{N2} & \dots & w_{NN} \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ w_{N1} & w_{N2} & \dots & w_{NN} \end{pmatrix} + \alpha \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ u_{21} & u_{22} & \dots & u_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ u_{N1} & u_{N2} & \dots & u_{NN} \end{pmatrix}$$

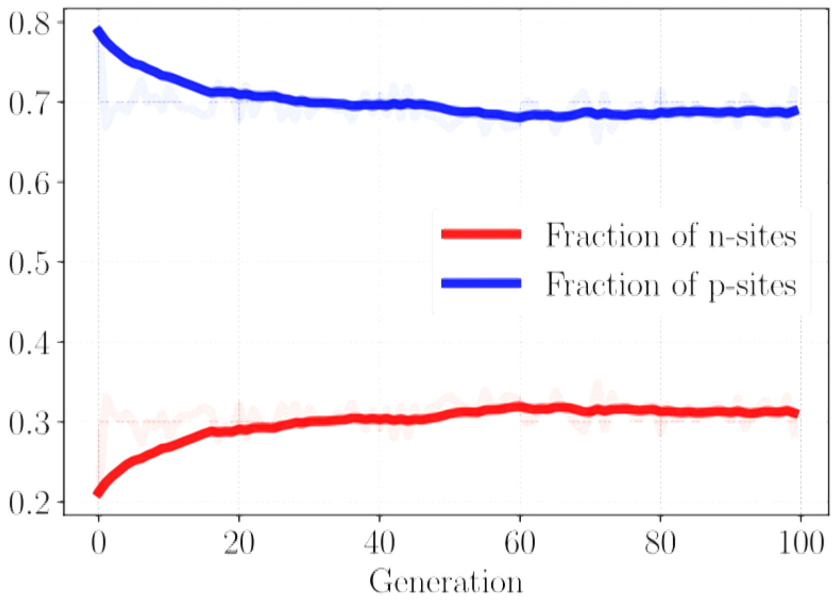
$$\alpha = 0.1, \quad u_{ij} \in [-1, 1]$$

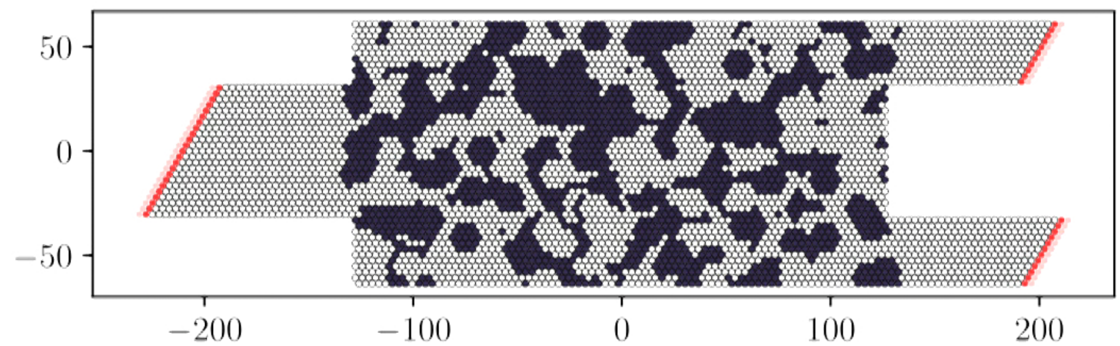
More Sophisticated Strategy - Testing ANN Policy



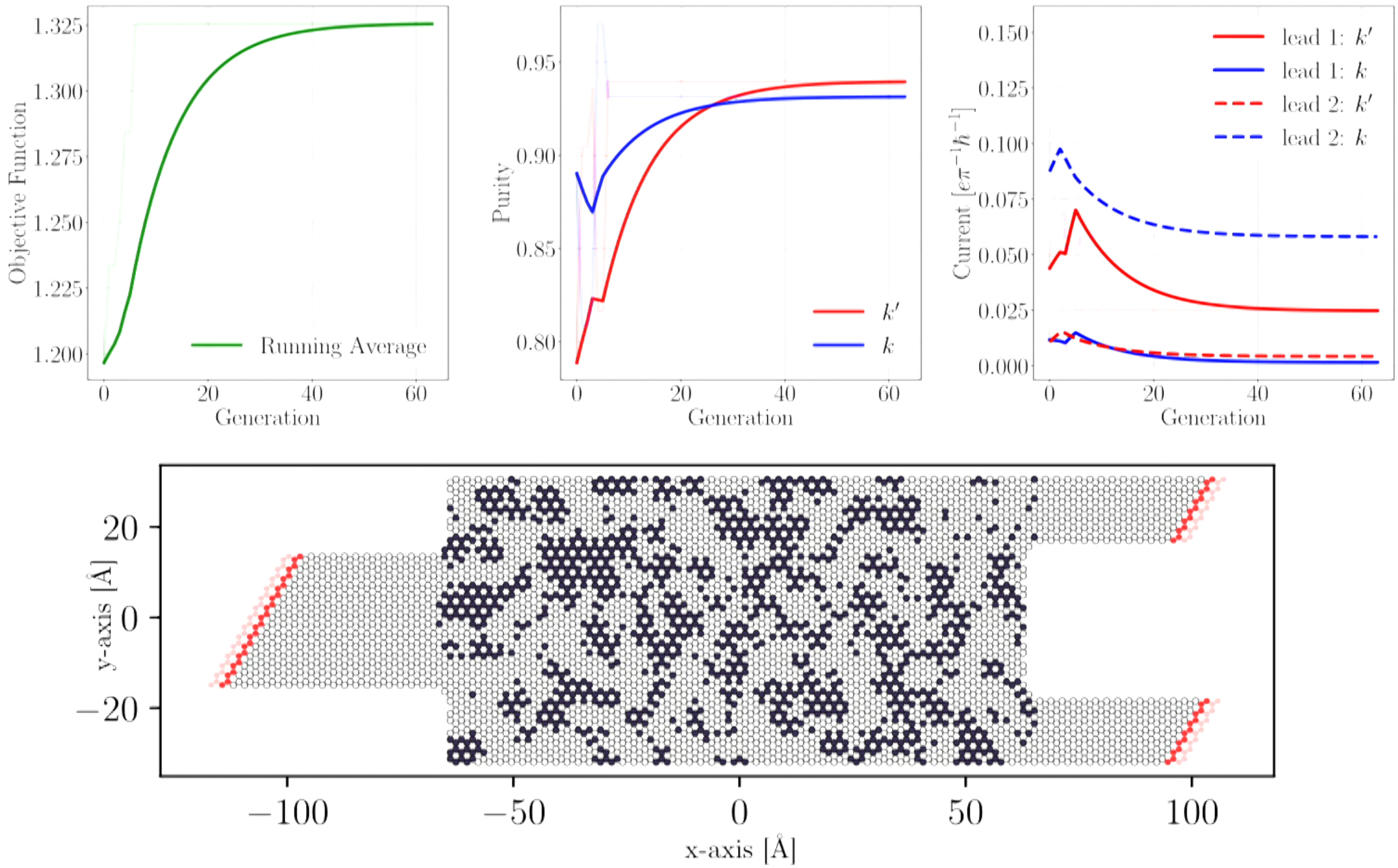


More Sophisticated Strategy - Testing ANN Policy



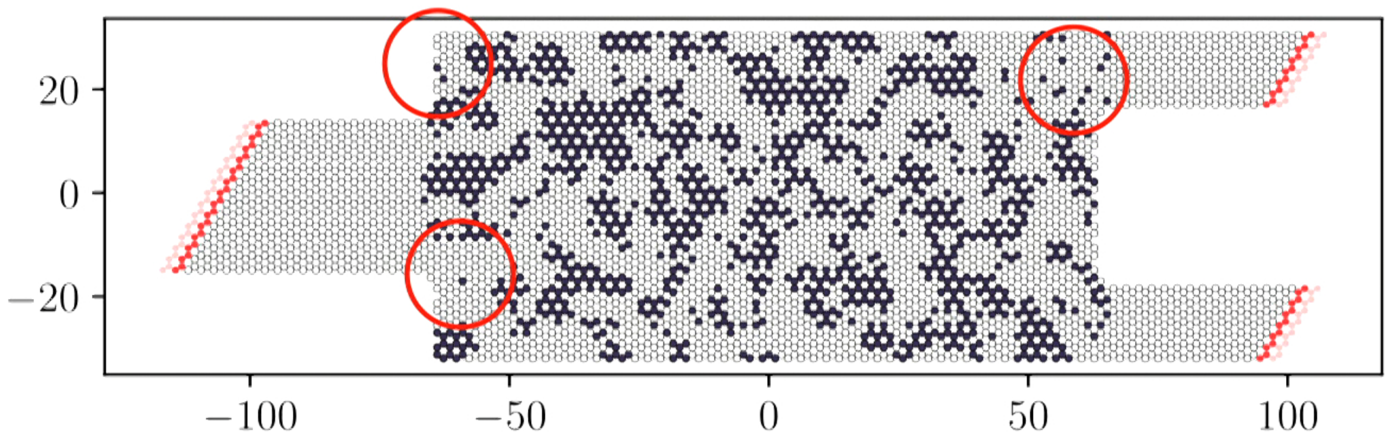


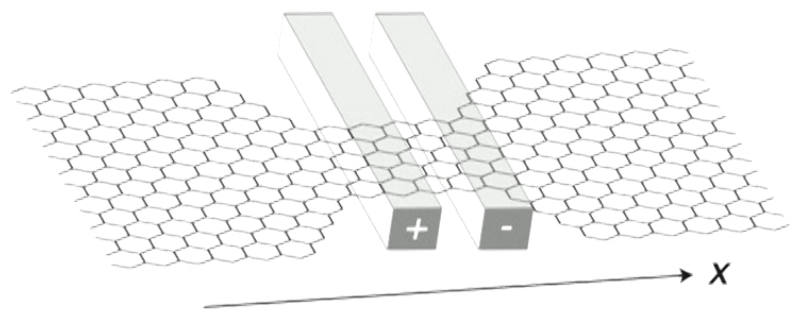
Results



1.325
1.300
1.275
1.250
1.225
1.200
Objective Function
Running Average
0
20
40
60
Generation
0.95
0.90
0.85
0.80
Purity
k'
k
Generation
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Current $[e\pi^{-1}\hbar^{-1}]$
lead 1: k'
lead 1: k
lead 2: k'
lead 2: k
Generation
20
0
-20
y-axis [Å]
-100
-50
0
50
100
x-axis [Å]

Experimental Realizability

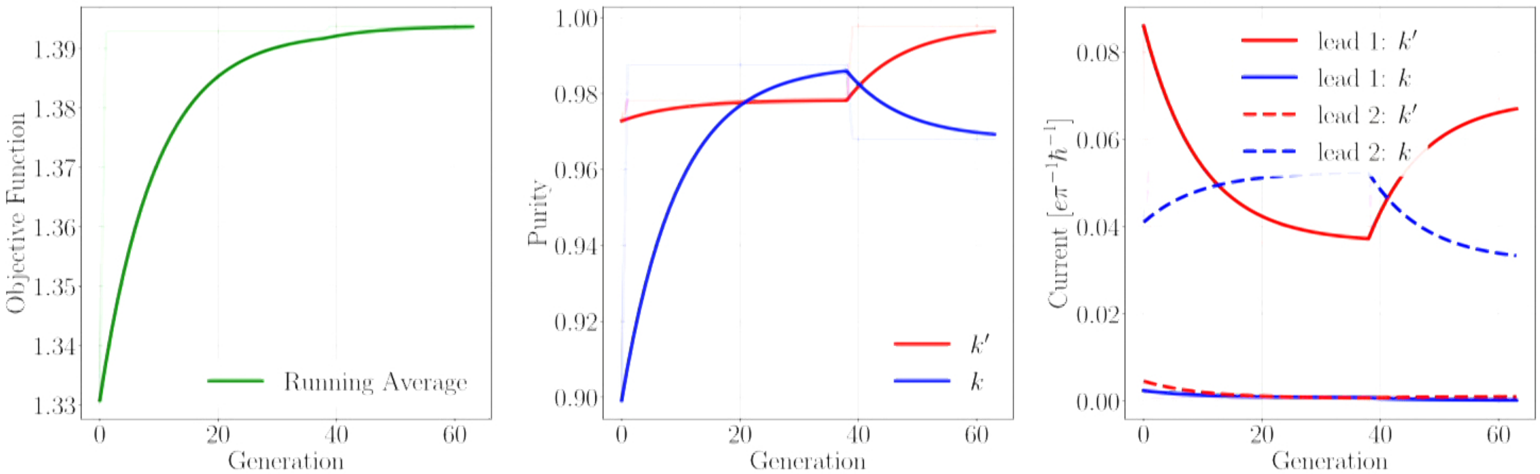


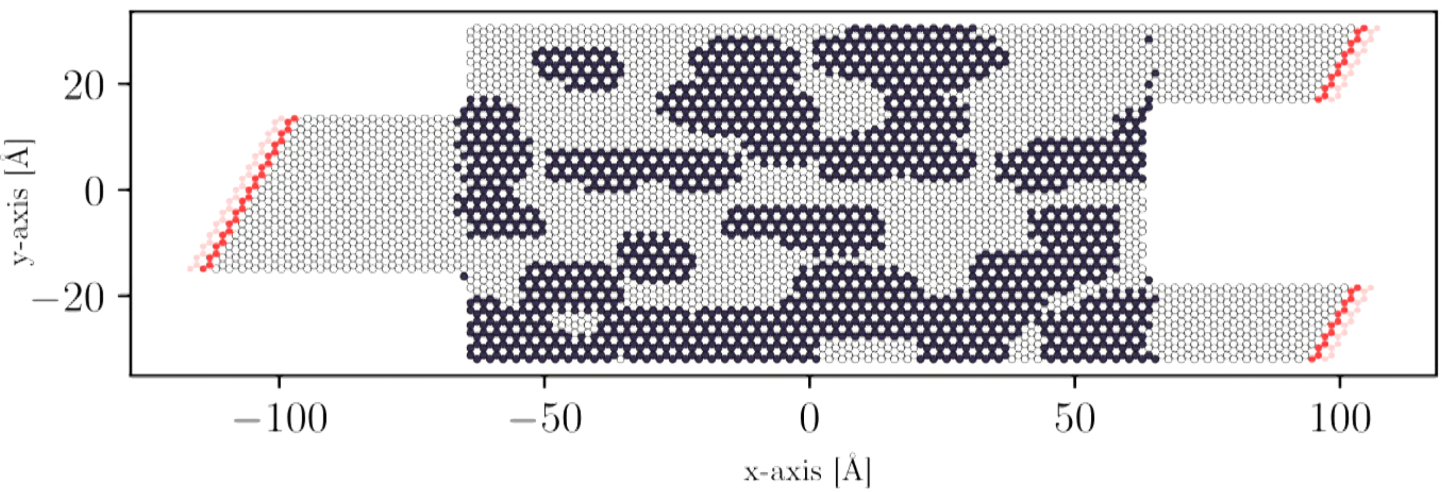


Taken from [1]

[1] Rycerz A, Tworzydło J, Beenakker CW. Valley filter and valley valve in graphene. Nature Physics. 2007 Mar;3(3):172.

Results - With Image Filtering





Summary

- Treat calculating ballistic current as a "particle passing through a finite barrier" problem
- Able to search for structures by pairing transport calculations with GAs
- The polygon strategy for splitting valley currents yields high purities, but gets stuck in a local minima of optimizing one valley current and produces devices with low currents
- The ANN policy for splitting valley currents yields high purities, rids the local minima, and has an order of magnitude higher currents