

Ferromagnetism in Bernal stacked bilayer graphene

Khanh Pham

2021 Physics REU, University of California, Santa Barbara and Department of Physics, UC Berkeley

Faculty Advisor: Andrea Young

Department of Physics, University of California at Santa Barbara

(Dated: September 29, 2021)

It has been recently experimentally observed that rhombohedral (ABC) trilayer graphene, exhibits ferromagnetism[1] and superconductivity[2] at certain applied electric field strengths and carrier densities. It is unknown, however, when and why exactly Bernal (AB) stacked bilayer graphene becomes ferromagnetic but not superconductive when applying electric fields and varying carrier densities. This is strange because AB bilayer graphene and ABC trilayer graphene have similar band structure and density of states, meaning that their electronic properties should be similar as well. We numerically calculate the regions in which AB stacked bilayer graphene becomes ferromagnetic and compare the results to experimental observations. AB bilayer graphene is much more stable and easier to produce than ABC trilayer graphene, and if observed to be superconducting, would be a more suitable material for use. These results imply that transistor like devices using graphene can be designed to easily switch between being an insulator, superconductor, or a ferromagnet just by changing the applied gate voltage to the device.

I. INTRODUCTION

A. Graphene

Graphene is a two dimensional material made of carbon atoms arranged in a honeycomb lattice as shown below in Fig. 1. It is characterized by two lattice vectors, $\mathbf{a}_1 = \frac{a}{2}(3, \sqrt{3})$ and $\mathbf{a}_2 = \frac{a}{2}(3, -\sqrt{3})$ where $a = 2.46 \text{ \AA}$ is the lattice constant. The atoms of graphene are given labels of A and B atoms. The nearest neighbor vectors are given as $\delta_1 = \frac{a}{2}(1, \sqrt{3})$, $\delta_2 = \frac{a}{2}(1, -\sqrt{3})$, and $\delta_3 = -d(1, 0)$. The Brillouin zone of graphene also turns out to be hexagonal, with special points being the corners of the Brillouin zone, labelled as $K = \frac{2\pi}{3\sqrt{3}a}(\sqrt{3}, 1)$ and $K' = \frac{2\pi}{3\sqrt{3}a}(\sqrt{3}, -1)$ points in k -space. Graphene has many interesting properties, including being a rather good electrical and thermal conductor. It is also one of the lightest, strongest, and most flexible materials discovered. Graphene can be manipulated into other interesting systems, like carbon nanotubes by rolling up a graphene monolayer, or by stacking it upon itself enough times for it to become the three dimensional material, graphite.

In AB stacked bilayer graphene, a second layer of graphene is situated directly above the first and shifted until its B atoms are in the center of the hexagons of the first layer. The first two layers of ABC stacked trilayer graphene are identical to those of AB stacked bilayer graphene. However, the third layer has shifted over another set of atoms so that its B atoms are in the center of the hexagons of the second layer. It turns out that among the most common variations of bilayer graphene,

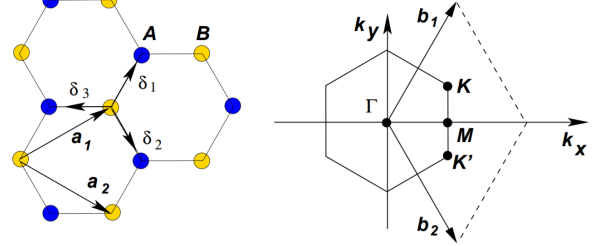


FIG. 1: Left: Monolayer graphene structure with lattice vectors and nearest neighbor vectors. Right: First Brillouin zone of Graphene with labelled K and K' points. This comes from Figure 2 from [3].

AA and AB, that AB stacked bilayer graphene is much more stable. AA stacked bilayer graphene has the second layer's atom perfectly align with the first layer's atom, which causes the carbon atoms to feel a stronger repulsive force. It has also been experimentally observed that ABC trilayer graphene is quite unstable[4]. If found to be superconducting, the stability of AB bilayer graphene makes it a much better candidate for future physical applications.

B. Tight binding model

We assume a tight-binding model for the electrons in graphene where the electrons can hop between nearest

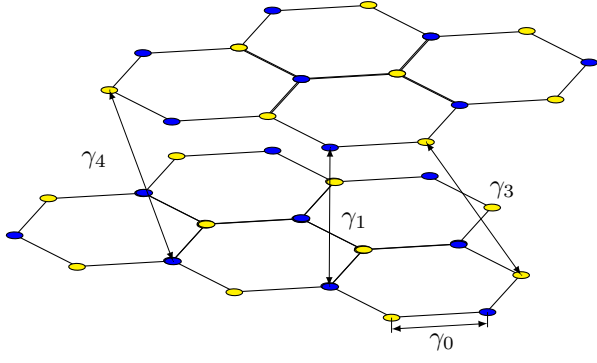


FIG. 2: structure of AB bilayer graphene

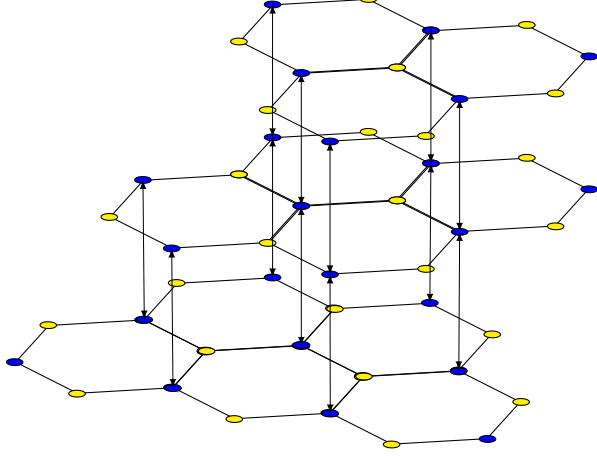


FIG. 3: structure of ABC trilayer graphene

neighbor atoms, and the Hamiltonian can be written as

$$\begin{aligned} \mathcal{H}_{t.b} = & -\gamma_0 \left(\sum_{\langle i,j \rangle, m} \hat{a}_{m,i}^\dagger \hat{b}_{m,j} + \hat{b}_{m,i}^\dagger \hat{a}_{m,j} \right) \\ & - \gamma_1 \left(\sum_j \hat{a}_{1,j}^\dagger \hat{a}_{2,j} + \hat{a}_{2,j}^\dagger \hat{a}_{1,j} \right) \\ & - \gamma_3 \left(\sum_{\langle i,j \rangle} \hat{b}_{1,i}^\dagger \hat{b}_{2,j} + \hat{b}_{2,j}^\dagger \hat{b}_{1,i} \right) \\ & - \gamma_4 \left(\sum_{\langle i,j \rangle} \hat{a}_{1,i}^\dagger \hat{b}_{2,j} + \hat{a}_{2,i}^\dagger \hat{b}_{1,j} + \hat{b}_{1,j}^\dagger \hat{a}_{2,i} + \hat{b}_{2,j}^\dagger \hat{a}_{1,i} \right) \end{aligned} \quad (1)$$

where the operators $\hat{a}_{m,i}$ ($\hat{a}_{m,i}^\dagger$) refers to annihilation (creation) of an electron at site i on layer m , and similarly for $\hat{b}_{m,i}$. The γ_i are the different energies required for hopping, which are $\gamma_0 = 3160$ meV, $\gamma_1 = 381$ meV, $\gamma_3 = 380$ meV, and $\gamma_4 = -140$ meV[5]. There are some additional terms that can be added, such as an applied interlayer potential between the two layers or additional energies coming from dimer sites. In solving for a usable form of

the Hamiltonian, one obtains a Hamiltonian that can be written in matrix form as

$$\mathcal{H}_{bi,k} = \begin{bmatrix} -\Delta/2 & -\gamma_0 f(\mathbf{k}) & -\gamma_4 f(\mathbf{k}) & -\gamma_3 f(\mathbf{k})^* \\ -\gamma_0 f(\mathbf{k})^* & -\Delta/2 + \Delta' & \gamma_1 & -\gamma_4 f(\mathbf{k}) \\ -\gamma_4 f(\mathbf{k})^* & \gamma_1 & \Delta/2 + \Delta' & -\gamma_0 f(\mathbf{k}) \\ -\gamma_3 f(\mathbf{k}) & -\gamma_4 f(\mathbf{k})^* & -\gamma_0 f(\mathbf{k})^* & \Delta/2 \end{bmatrix} \quad (2)$$

where $f(\mathbf{k}) = e^{-ik_x a} \left[1 + 2e^{i3k_x a/2} \cos\left(\frac{\sqrt{3}}{2} k_y a\right) \right]$ and the used basis is $[\Psi_{a,1}, \Psi_{b,1}, \Psi_{a,2}, \Psi_{b,2}]$. The Δ parameter is the strength of an interlayer potential, usually in the range of tens of meV, and Δ' is a term resulting from energy at dimer sites, where $\Delta' = 22$ meV[5]. However, it becomes possible to Taylor expand the Hamiltonian around the K and K' points, into the form

$$\mathcal{H}'_k = \begin{bmatrix} -\Delta/2 & v_0 \pi^\dagger & v_4 \pi^\dagger & v_3 \pi^\dagger \\ v_0 \pi & -\Delta/2 + \Delta' & \gamma_1 & v_4 \pi^\dagger \\ v_4 \pi & \gamma_1 & \Delta/2 + \Delta' & v_0 \pi^\dagger \\ v_3 \pi^\dagger & v_4 \pi & v_0 \pi & \Delta/2 \end{bmatrix}, \quad (3)$$

where $\pi = \xi k_x + i k_y$ and $\xi = +1$ at the K point and $\xi = -1$ at the K' point. The new coefficients v_i are what are known as the Fermi velocities, $v_i = \frac{\sqrt{3} a \gamma_i}{2\hbar}$.

C. Ferromagnetism

We explain the origin of ferromagnetism in materials. Ferromagnetism is the result the Coulomb interaction and quantum mechanics. For the charge carriers in materials, electrons and holes are fermions. As a result, the many body wave function for the system must be antisymmetric under the exchange operator, i.e., $\hat{P}(\Psi_{M.B}) = -(\Psi_{M.B})$. For a simple system of 2 electrons, we consider their spatial wavefunction and their spin wavefunction. Supposing that the electrons are in states ψ_1 and ψ_2 , then their wave function is either of the form

$$\Psi_C(\vec{x}_1, \vec{x}_2) = \frac{1}{\sqrt{2}} (\psi_1(\vec{x}_1) \psi_2(\vec{x}_2) + \psi_2(\vec{x}_1) \psi_1(\vec{x}_2)) \cdot \Phi_{A.S.}$$

or

$$\Psi_F(\vec{x}_1, \vec{x}_2) = \frac{1}{\sqrt{2}} (\psi_1(\vec{x}_1) \psi_2(\vec{x}_2) - \psi_2(\vec{x}_1) \psi_1(\vec{x}_2)) \cdot \Phi_S,$$

where here $\Phi_{A.S}$ is an antisymmetric spin wavefunction and Φ_S is a symmetric spin wavefunction. In the case of an antisymmetric wavefunction, we have that $\Phi_{A.S} = \frac{1}{\sqrt{2}} (|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$, meaning that the electrons must have opposite spins. In the case of a symmetric wavefunction, the symmetric spin wavefunction is either $\Phi_S = |\uparrow\uparrow\rangle, |\downarrow\downarrow\rangle$, or $\frac{1}{\sqrt{2}} (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)$. We note here that the two electrons can share the same spin if their spin wavefunction is symmetric.

In the case of Ψ_C , it turns out that the electrons are on average closer together. If we account for the Coulomb interaction, classical electromagnetism states that the potential energy between two charges is proportional to the square of the inverse distance between them. Thus, electrons with a symmetric spatial wavefunction are on average closer together, causing the energy resulting from the Coulomb energy to be higher. If the spatial wavefunction is antisymmetric, the electrons are on average farther apart, causing the Coulomb energy to be lower.

Then, ferromagnetism occurs when the energy due to Coulomb interaction becomes more significant than the system's kinetic energy. In the case that the Coulomb energy must be minimized due to it being larger than the kinetic energy, the system's spatial wavefunction is antisymmetric so as to maximize the distance between electrons. Due to the spatial wavefunction being antisymmetric, the spin wavefunction has to be symmetric, allowing for the system to have all its spins align, resulting in ferromagnetism in the system.

II. CALCULATIONS

A. Band structure

The band structure of the bilayer graphene system can be computed by calculating the eigenvalues of $\mathcal{H}_{\mathbf{k}}$ or $\mathcal{H}'_{\mathbf{k}}$ for various points in k -space. In particular, we calculate the band structure of the system along the line $k_y = \frac{1}{\sqrt{3}}k_x$, which gives the band structure along the $K' - \Gamma - K - M$ points. We are only interested in the system at low energies, which happens to be near the K and K' points. This gives an electron the property of belonging to one of four flavors, depending on whether it is up or down spin and whether it is in the K or K' valley.

Increasing the strength of the interlayer potential induces a larger bandgap in the inner bands of the band structure.

B. Carrier Density and Density of states

Using the previously calculated band structure, we can then compute the density of states by first calculating the carrier density via

$$n(\mu) = \frac{1}{A \cdot (2\pi)^2} \int \int \frac{1}{\exp\left(\frac{\epsilon(\vec{k}) - \mu}{k_B T}\right) + 1} dk_x dk_y, \quad (4)$$

where A is the area of a unit cell of the graphene lattice, ϵ is a band structure energy at some point in k space, and μ is the Fermi energy we can attempt to vary. This

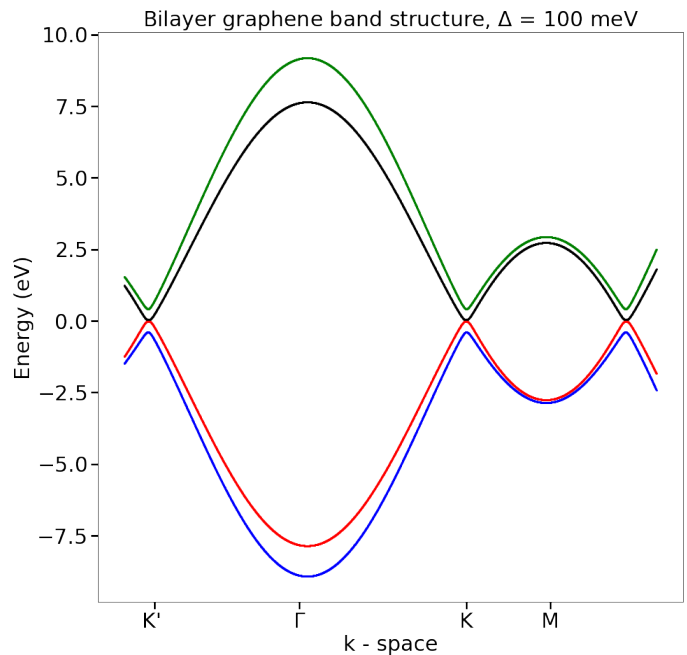


FIG. 4: Bilayer graphene band structure at interlayer potential $\Delta = 100$ meV

comes from standard statistical mechanics as the average density is given by $\langle n \rangle = \frac{1}{A} \sum_{\vec{k}} \frac{1}{\exp\left(\frac{\epsilon(\vec{k}) - \mu}{k_B T}\right) + 1}$, where the sum over $\vec{k}$ should be done over the whole Brillouin zone. We generally take the temperature to be $T = 0$ K, which just causes $\frac{1}{\exp\left(\frac{\epsilon(\vec{k}) - \mu}{k_B T}\right) + 1} = 1$ when $\epsilon(\vec{k}) < \mu$ and $\frac{1}{\exp\left(\frac{\epsilon(\vec{k}) - \mu}{k_B T}\right) + 1} = 0$ when $\epsilon(\vec{k}) > \mu$. Given the density, we can then calculate the density of states via

$$\rho(\mu) = \frac{dn}{d\mu}. \quad (5)$$

With the density of states calculation, we obtain the plot in Figure 5. As seen in the density of states plot, $\rho(\mu) = 0$ for $-50 \text{ meV} \leq \mu \leq 50 \text{ meV}$, which was caused by the applied interlayer potential. This is also evident in the band structure plot, making it not possible for any carriers to have an energy between -50 meV and 50 meV.

C. Stoner ferromagnetism

The tight binding model does not account for electron-electron interactions within graphene. To account for this, we use a Stoner model that gives the grand potential of the system per area as

$$\frac{\Phi}{A} = \sum_{\alpha} E_0(n_{\alpha}) + V_{\text{int}} + \mu \sum_{\alpha} n_{\alpha} \quad (6)$$

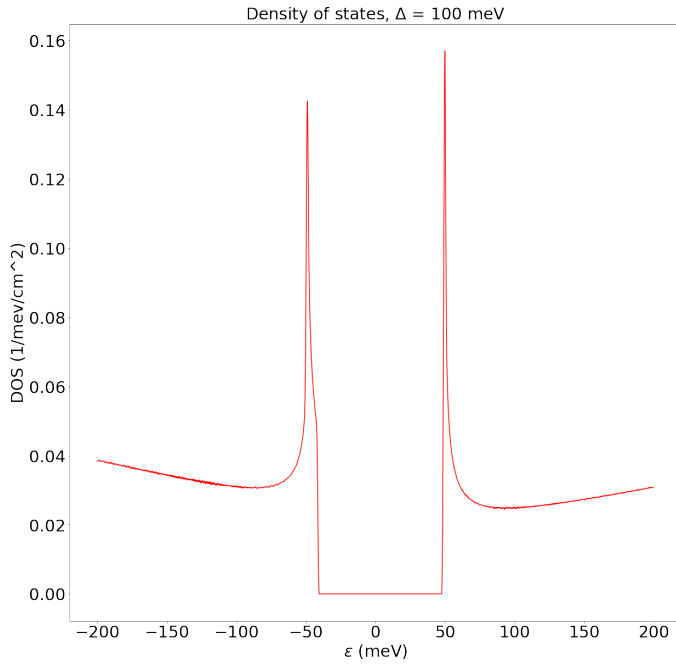


FIG. 5: Bilayer graphene density of states at interlayer potential $\Delta = 100$ meV

where

$$V_{\text{int}} = \frac{UA_{\text{u.c.}}}{2} \sum_{\alpha \neq \beta} n(\alpha)n_{\beta} + JA_{\text{u.c.}}(n_1 - n_3)(n_2 - n_4). \quad (7)$$

We note that $n_{\alpha} = \int_0^{\mu} \rho(\epsilon)d\epsilon$ is the electron density for a particular flavor, α . Then $E_0(\mu_{\alpha}) = \int_0^{\mu_{\alpha}} \epsilon \rho(\epsilon)d\epsilon$ is the kinetic energy of a flavor and μ is some fixed chemical potential. The flavors are labelled accordingly by $1 = \{K, \uparrow\}$, $2 = \{K', \uparrow\}$, $3 = \{K, \downarrow\}$, and $4 = \{K', \downarrow\}$. The term V_{int} represents an interaction energy between different flavors, with U being an interaction strength between different flavors, J being an interaction specific to total polarization differences, and $A_{\text{u.c.}}$ being the area of a unit cell of bilayer graphene. We have implicitly assumed that electrons of different flavors all interact with each other equally. According to the Stoner criterion for ferromagnetism, ferromagnetism occurs whenever $U\rho(\epsilon) > 1$. Thus, U is essentially the strength of the Coulomb interaction between electrons of different flavors, with ferromagnetism occurring whenever the Coulomb energy V_{int} becomes dominant over the kinetic energy. As U is increased, more ferromagnetic regions should show up. The J parameter is included for the fact that the system seems to polarize the spins a lot more strongly than the valleys. We note that J really only affects the $2x$ phase. Larger J causes the $2x$ phase to be suppressed, while more negative J cause the $2x$ phase to be more common.

In practice, we compute our grand potential as a

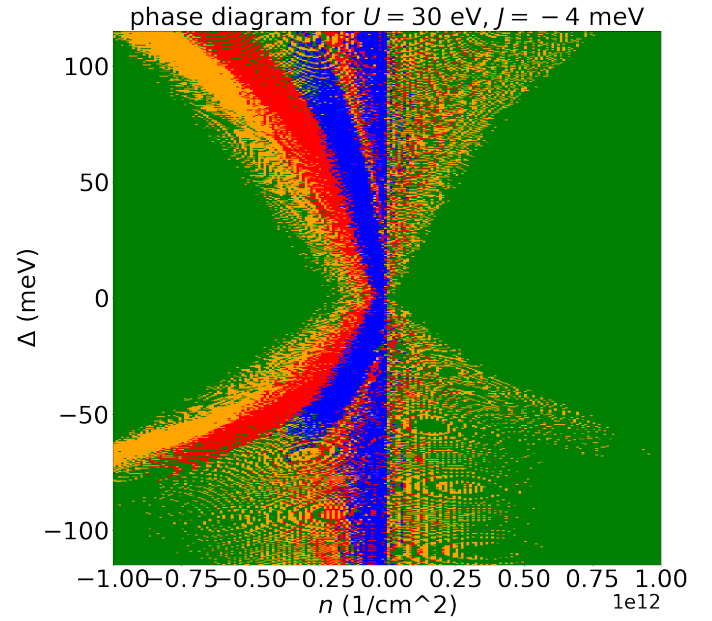


FIG. 6: Simulated bilayer graphene phase diagram at fixed $U = 30$ eV. The color map is given as $\{\text{blue, red, yellow, green} = 1, 2, 3, 4\}$.

function of the interlayer potential Δ , and carrier density, n . Thus, it becomes necessary to compute the kinetic energy as a function of carrier density instead, $E_0(n_{\alpha}) = \int_0^{n_{\alpha}} \epsilon(n)dn$.

While U and J are possible to vary, they are in practice rather difficult to vary experimentally. It is instead much easier to vary carrier density and the interlayer potential. Thus, we plot $\frac{\Phi}{A}$ as a color plot as a function of (n_T, Δ) , where $n_T = n_1 + n_2 + n_3 + n_4$ is the total electron density giving plots similar to that in Figure 6. The colors represents how many flavors are being occupied, with the blue and red regions being the ferromagnetic regions, yellow being less ferromagnetic, and green being essentially paramagnetic.

III. DISCUSSION

A. Experimental Observations

Professor Young's group has done some experimental work and the data obtained is given in Figure 8. They measured the resistance of AB bilayer graphene as a function of the carrier density and the applied electric field. Resistance can be mapped to the minimum energy state of the system, which is why the resistance plot can give a phase diagram of the system. Within the data, there are some regions of specific resistances that are believed to be where the AB bilayer graphene system occupies $n\pi$

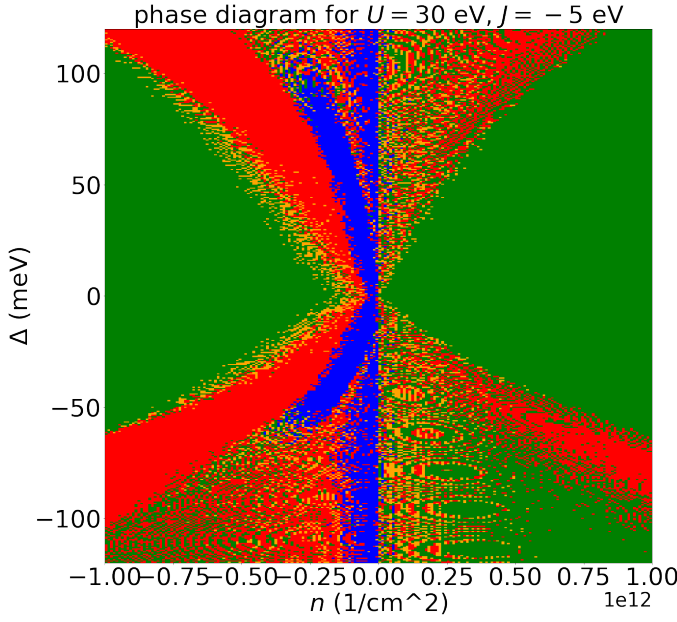


FIG. 7: Another phase diagram, but with $J = -5$ eV. The $2x$ phase has essentially suppressed the $3x$ phase.

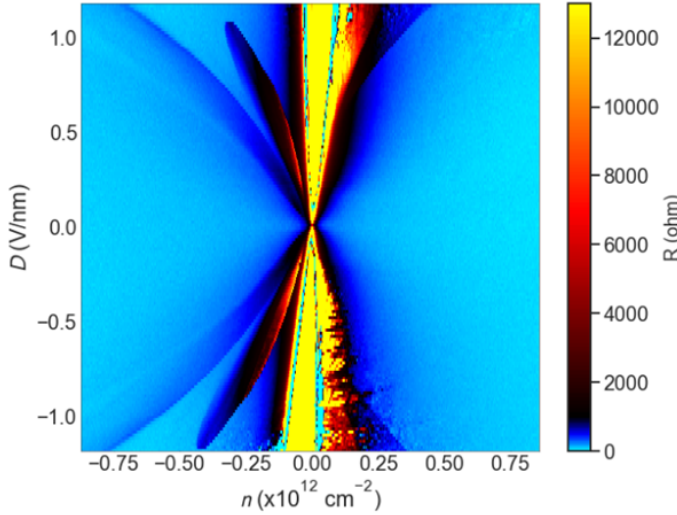


FIG. 8: Resistance of AB bilayer graphene as a function of electric field and carrier density

flavors, where n is the number of flavors being occupied. The most ferromagnetic regions are those labelled by the $2x$ and $1x$ regions.

B. Conclusion

We see that our simulated results and the experimental data do seem to corroborate each other. Specifically, our calculations also include a butterfly wing $1x$ region as well. Our calculations also captures the behavior of the

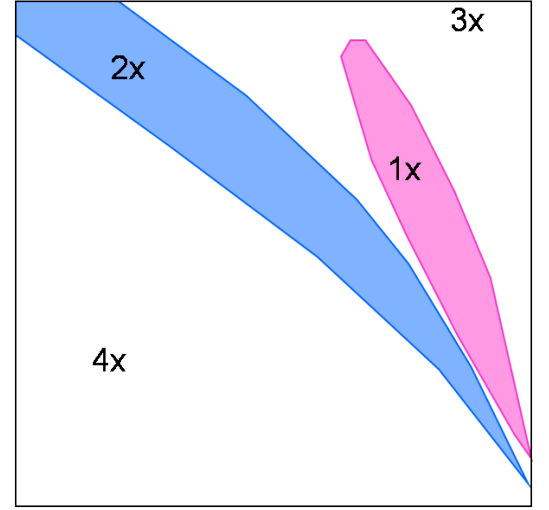


FIG. 9: Labelled phases in experimental data for positive electric field and negative carrier density

$2x$ butterfly wing region fairly well. The results of both experiment and computation indicate that AB bilayer graphene becomes ferromagnetic in the $1x$ and $2x$ regions along some defined curve in the electric field and carrier density plane. Changing the simulation parameters U and J leads to more accurate phase diagrams compared to experiment, where it seems the likelihood of polarizing the spins is rather quite strong, with $|\frac{U}{J}| \approx \frac{1}{6}$.

C. Further work

Our simulation currently still has quite a bit of noise, with a lot of phase transitions between the different nx regions. This is not really observed experimentally and is likely due to the energies between various phases being very close together near the boundaries. In order to resolve this, it becomes necessary to make the energy calculations more accurate. It is difficult to increase the accuracy of our simulation without also greatly increasing its run time, whether through increasing the number of k points sampled to calculate the density of states, or by making the integration more accurate.

Another thing that can be investigated is how the Fermi surface changes at the tip of the butterfly wing. The Fermi surface is the surface separating occupied and unoccupied energy states of the system, and understanding more about this can give greater intuition into understanding why phase transitions occur where they do. A more sophisticated method to account for the Coulomb interaction than the Stoner model is to perform a Hartree - Fock analysis. Using $\mathcal{H}_{bi,k}$ in (2), we add to it a Hamil-

tonian that accounts for the Coulomb interaction[6],

$$\mathcal{H}_C = \frac{1}{2} \sum_{\alpha, \beta, i, j} \int d^2\mathbf{r} d^2\mathbf{r}' U(|\mathbf{r} - \mathbf{r}'|) \psi_{i\alpha}^\dagger(\mathbf{r}) \psi_{j\beta}^\dagger(\mathbf{r}') \psi_{j\beta}(\mathbf{r}') \psi_{i\alpha}(\mathbf{r}) \quad (8)$$

where $\alpha, \beta = 1, \dots, 4$ refers to the possible flavors, $i, j = 1, \dots, 4$ run over the two bilayer sublattice and layers, $\psi_{j\alpha}^\dagger(\mathbf{r})$ creates an electron at $\mathbf{r}$ with the appropriate j location and α flavor. The interaction term is

$$U(\mathbf{r}) = \int \frac{d^2q}{(2\pi)^2} e^{i\mathbf{q}\cdot\mathbf{r}} \cdot \frac{2\pi z^2 \tanh(qd)}{\epsilon q}, \quad (9)$$

where z is the electron charge, and d and ϵ are experimental terms, with d being the distance between metallic gates and the bilayer system and ϵ being some dielectric constant.

IV. ACKNOWLEDGMENTS

I would like to thank my faculty advisor, Andrea Young, for guiding me throughout the REU program. Additionally, I would like to thank Andrea's graduate student, Haoxin Zhou, for teaching me about his code and letting me adapt some of it to this project. I would also like to thank the director of the REU program, Sathya Guruswamy for organising many professional and social events. Finally, I would like to thank the National Science Foundation for funding the program. This work was possible by NSF REU grant PHY-1852574.

V. APPENDIX

Python code

In general, the used Hamiltonians are too complicated to analytically obtain useful expressions for electron density or density of states. Thus, it becomes necessary to numerically calculate the desired quantities.

```
import numpy as np
import matplotlib.pyplot as plt
import math
from matplotlib.pyplot import figure
from numpy import matrix
from numpy import linalg as LA
from math import log, exp, cos
from math import e
from scipy.interpolate import interp1d
import pandas as pd
import pandas as pd
import numba
from numba import jit
```

This imports all of the libraries we will use. Any code with "#haoxin" is modified from Haoxin Zhou's code on ABC trilayer graphene, written originally in Matlab. This first function computes the eigenvalues of the Hamiltonian in Eq. 3 at every point in the specified grid in the $k_x - k_y$ plane.

```
@jit(nopython=True)#haoxin
def dispersion(numx, numy, kxmin, kxmax,
kymin, kymax, d1, ksi):
```

```
g0 = 3160
g1 = 381
g3 = 380
g4 = -140
d2 = 22
```

```
h0 = np.array([[-d1/2, 0, 0, 0],
[0, -d1/2 + d2, g1, 0],
[0, 0, d1/2 + d2, 0],
[0, 0, 0, d1/2]])
```

```
h0 = h0+np.transpose(h0)
```

```
hx = (3**0.5)/2*ksi*np.array([
[0, g0, -g4, g3],
[0, 0, 0, -g4],
[0, 0, 0, g0],
[0, 0, 0, 0]])
hx = hx+np.transpose(hx)
```

```
hy = (3**0.5)/2*1j*np.array([
[0, -g0, g4, -g3],
[0, 0, 0, g4],
[0, 0, 0, -g0],
[0, 0, 0, 0]])
hy = hy+np.conjugate(np.transpose(hy))
```

```
kkx = np.linspace(kxmin, kxmax, numx)
kky = np.linspace(kymin, kymax, numy)
```

```
results = np.zeros((4, numx, numy))
for ii in range(0, numx):
    for jj in range(0, numy):
        hh = h0+kkx[ii]*hx+kky[jj]*hy
        eigenvalues
        = np.sort(np.real(LA.eig(hh)[0]))
        for i in range(0, 4):
            results[i, ii, jj]
            = eigenvalues[i]
    return results
```

The next code computes the $n(\mu)$ curve mentioned ear-

lier.

```
@jit(nopython=True, parallel = True)#haoxin
def fermidirac(E, mu, T):
```

```
    kb = 8.617333262145e-2
    return np.divide(1,np.exp(
        np.divide((E-mu),(kb*T)))+1)
```

```
@jit(nopython=True, parallel = True)#haoxin
def density(kx_mesh, ky_mesh, EE0, mu, T):
```

```
    dkx = kx_mesh[0][1] - kx_mesh[0][0]
    dky = ky_mesh[1][0] - ky_mesh[0][0]
    ne_mesh = fermidirac(EE0,mu,T)
    return (np.sum(ne_mesh)
        *dkx*dky/((2*np.pi)**2))
```

#haoxin

```
def n_list_gen(T, mu_range,num_mu, numx,
    numy, kx_range, ky_range, d1, ksi):
    uu = d1
    kxx = np.linspace(kx_range[0],
        kx_range[1],numx)
    kky = np.linspace(ky_range[0],
        ky_range[1],numy)
    [kx_mesh,ky_mesh] = np.meshgrid(kxx,kky)
    dkx = kx_mesh[0][1] - kx_mesh[0][0]
    dky = ky_mesh[1][0] - ky_mesh[0][0]

    mmu = np.linspace(mu_range[0],
        mu_range[1], num_mu)
    nne = np.zeros(num_mu)

    EE = dispersion(numx, numy, kx_range[0],
        kx_range[1], ky_range[0], ky_range[1], uu,
        ksi)

    eev = EE[1, :, :]
    eec = EE[2, :, :]

    for ii in range(0,num_mu):
        nne[ii] = density(
            kx_mesh, ky_mesh, eec, mmu[ii], T)
        + density(
            kx_mesh, ky_mesh, eev, mmu[ii], T)
    dmu = (mmu[1]-mmu[0])/1e3
    dos = np.diff(nne)/dmu*4.0
    dos_list = list(dos)

    cnp = int(round(np.mean(
        dos_list.index(min(dos_list)))))

    nne0 = nne[cnp]
```

```
    return (nne-nne0)/((2.46e-10)**2)*(1e-4)
```

Using the `n_list_gen` function, we generate many $\mu(n)$ curves at various Δ . We can then use our grand potential function, which gives us the number of flavors that are occupied in order to achieve the minimum energy at that particular Δ and n .

```
@jit(nopython=True)
def minimum_system_densities(E, n_list,
    n, U, J):
```

```
    n_zero = find_nearest(n_list, 0)
```

```
    if n >= 0:
```

```
        n_list = n_list[n_zero:]
        E = E[n_zero:]
```

```
    else:
```

```
        n_list = n_list[:n_zero+1]
        E = E[:n_zero+1]
```

```
    n_up_limit_1 = find_nearest(n_list, n)
    n_up_limit_2 = find_nearest(n_list, n/2)
    n_up_limit_3 = find_nearest(n_list, n/3)
    n_up_limit_4 = find_nearest(n_list, n/4)
```

```
    A = (3**0.5)*((2.46e-8)**2)/2
```

```
    if n >= 0:
```

```
        E_1 = E[:n_up_limit_1]
        n_list_1 = n_list[:n_up_limit_1]
        E_2 = E[:n_up_limit_2]
        n_list_2 = n_list[:n_up_limit_2]
        E_3 = E[:n_up_limit_3]
        n_list_3 = n_list[:n_up_limit_3]
        E_4 = E[:n_up_limit_4]
        n_list_4 = n_list[:n_up_limit_4]
```

```
        E_1f = np.trapz(E_1, n_list_1)
```

```
        E_2f = np.trapz(E_2, n_list_2)
        E_2f = 2*E_2f + U*A*(n**2)/8
        + J*A*(n**2)/4
```

```
        E_3f = np.trapz(E_3, n_list_3)
        E_3f = 3*E_3f + 3*U*A*(n**2)/18
```

```
        E_4f = np.trapz(E_4, n_list_4)
        E_4f = 4*E_4f + 6*U*A*(n**2)/32
```

```
    else:
```

```
        E_1 = E[n_up_limit_1:]
        n_list_1 = n_list[n_up_limit_1:]
```

```

E_2 = E[n_up_limit_2:]
n_list_2 = n_list[n_up_limit_2:]
E_3 = E[n_up_limit_3:]
n_list_3 = n_list[n_up_limit_3:]
E_4 = E[n_up_limit_4:]
n_list_4 = n_list[n_up_limit_4:]

E_1f = np.trapz(E_1, n_list_1)
E_1f = (-1)*E_1f

E_2f = np.trapz(E_2, n_list_2)
E_2f = -2*E_2f + U*A*(n**2)/8
        + J*A*(n**2)/4

E_3f = np.trapz(E_3, n_list_3)
E_3f = -3*E_3f + 3*U*A*(n**2)/18

E_4f = np.trapz(E_4, n_list_4)
E_4f = -4*E_4f + 6*U*A*(n**2)/32

energy_list = [E_1f, E_2f, E_3f, E_4f]
minimum = min(energy_list)
if minimum == E_1f:
    result = 1
elif minimum == E_2f:
    result = 2
elif minimum == E_3f:
    result = 3
else:
    result = 4

return result

```

An example of the code to generate a phase diagram is given below:

```

from matplotlib.colors
import ListedColormap

col_dict={1:"blue",
          2:"red",
          3:"orange",
          4:"green"}

cm = ListedColormap([col_dict[x]
                    for x in col_dict.keys()])

n_list_5 = np.linspace(-1e+12, 1e+12, 800)
meV_list = np.linspace(-120,120,150)
N,meV = np.meshgrid(n_list_3, meV_list)

T = 0.0
mu_range = [-200, 200]
num_mu = 4000

```

```

ksi = -1
num_ne_list = 600
kx_range = [-0.5, 0.5]
ky_range = [-0.5, 0.5]
numx = 3000
numy = 3000
mmu = np.linspace(mu_range[0],
mu_range[1], num_mu)
list_of_lists = []
new_mmu = np.linspace(mu_range[0],
mu_range[1], 64000)
for i in range(0,75):
    start_time = time.time()
    n_list_3 = np.linspace(-1e+12, 1e+12, 800)
    list_values = []
    n_list = n_list_gen(T, mu_range,
num_mu, numx, numy,
kx_range, ky_range, meV_list[i], ksi)
    interpolated_n = interp1d(mmu, n_list,
kind='cubic')
    new_n_list = interpolated_n(new_mmu)
    new_n_list = list(new_n_list)
    for j in range(0,800):
        list_values.append(
            minimum_system_densities(new_mmu,
new_n_list, n_list_3[j],
30000,-5000))
    list_of_lists.append(list_values)
z_2d = np.array(list_of_lists)

fig_2 = plt.figure(figsize=(20,15))
plt.imshow(z_2d, extent=(n_list_3[0],
n_list_3[-1], meV_list[-1],meV_list[0]),
interpolation='nearest',
cmap=cm, aspect='auto')
plt.xlabel('$n$ (1/cm2)', fontsize = 20)
plt.ylabel(r'$\Delta$ (meV)', fontsize = 20)
plt.title('Bilayer graphene phase diagram for
$U = 30$, $J = -5$ eV', fontsize = 20)
plt.xticks(fontsize = 20)
plt.yticks(fontsize = 20)
plt.gca().invert_yaxis()

```

-
- [1] Zhao, H. *et al.* Half and quarter metals in rhombohedral trilayer graphene. arXiv:2104.00653v3 [cond-mat.mes-hall] (2021).
 - [2] Zhao, H. *et al.* Superconductivity in rhombohedral trilayer graphene. arXiv:2106.07640v1 [cond-mat.mes-hall] (2021).
 - [3] Castro Neto A H, Guinea F, Peres N M R, Novoselov K S and Geim A K 2009 *Rev. Mod. Phys.* **81** 109.
 - [4] Sugawara, K., Yamamura, N., Matsuda, K. *et al.* Selective fabrication of free-standing ABA and ABC trilayer graphene with/without Dirac-cone energy bands. *NPG Asia Mater* 10, e466 (2018).
 - [5] Edward McCann and Mikito Koshino 2013 *Rep. Prog. Phys.* 76 056503.
 - [6] Tommaso Cea and Francisco Guinea. Band structure and insulating states driven by Coulomb interaction in twisted bilayer graphene. *Phys. Rev. B*, 102(4):045107, July 2020.