TBG_ferromagnetism_figures

May 24, 2021

Table of Contents

1 Introduction

This Jupyter notebook loads data and generates figures from:

Emergent ferromagnetism near three-quarters filling in twisted bilayer graphene

Authors: Aaron L. Sharpe, Eli J. Fox, Arthur W. Barnard, Joe Finney, Kenji Watanabe, Takashi Taniguchi, M. A. Kastner, David Goldhaber-Gordon

https://arxiv.org/abs/1901.03520

The notebook has been tested with Python version 3.6.7 and Jupyter notebook server version 5.5.0.

2 Initialization

[1]:
```
# Setup

import numpy as np

import matplotlib
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.colors import LogNorm
from matplotlib.ticker import MultipleLocator, FormatStrFormatter,␣
↪ScalarFormatter
```

```
%matplotlib inline

from scipy.optimize import curve_fit

import sys
sys.path.insert(0, 'scripts')
from myTerrain import *
from dataStructure import *
from dataImport import *

myColdhot = coldHot(60,10)

from cycler import cycler

import json
```

[2]:
```
# Set constants

e = 1.602E-19 # Elementary charge in coulombs
h = 6.62607E-34 # Planck's constant in J*s
Ctop = 6.63E-04 # Top gate capacitance in F/m^2
Rq = h/e**2 # Resistance quantum
phib = h/e # Flux quantum
numsq = 2.15 # Number of squares between voltage terminals on the Hall bar
nsat = 3.370 # Density (in 10^12 cm^(-2)) corresponding to full filling of the␣
↪mini-Brillouin zone
```

[3]:
```
# Set default plotting parameters

plt.rcParams['axes.prop_cycle'] = cycler(
    color=['#E24A33','#348ABD','#988ED5','#777777',
           '#FBC15E','#8EBA42','#FFB5B8'])

plt.rcParams['axes.linewidth'] = 1
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['xtick.top'] = 'True'
plt.rcParams['xtick.major.size'] = 5
plt.rcParams['xtick.major.width'] = 1
plt.rcParams['xtick.minor.size'] = 2.5
plt.rcParams['xtick.minor.width'] = 1

plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['ytick.right'] = 'True'
plt.rcParams['ytick.major.size'] = 5
plt.rcParams['ytick.major.width'] = 1
plt.rcParams['ytick.minor.size'] = 2.5
plt.rcParams['ytick.minor.width'] = 1
```

```python
plt.rcParams['lines.linewidth'] = 1.5
plt.rcParams['font.size'] = 12
plt.rcParams['axes.labelsize']=12
plt.rcParams['font.family'] = 'Helvetica'
plt.rcParams['mathtext.fontset'] = 'custom'
plt.rcParams['mathtext.rm']='Helvetica'
plt.rcParams['mathtext.it']='Helvetica:italic'
plt.rcParams['mathtext.cal']='Helvetica:italic'
plt.rcParams['mathtext.bf']='Helvetica:bold'
```

3 Load data

[4]:
```python
def import_data(file):
    with open(file,'r') as f:
        json_load = json.load(f)
        storekey = []
        for key in json_load.keys():
            if type(json_load[key]) == list:
                json_load[key] = np.asarray(json_load[key])

    return json_load

fig1a = import_data('data/fig1a.json')

fig1b = import_data('data/fig1b.json')

fig2a = import_data('data/fig2a.json')

fig2b = import_data('data/fig2b.json')

fig2cd_figs5ab = import_data('data/fig2cd_figs5ab.json')

fig3 = import_data('data/fig3.json')

fig4 = import_data('data/fig4.json')

figs2 = import_data('data/figs2.json')

figs3 = import_data('data/figs3.json')

figs4 = import_data('data/figs4.json')

figs5cd_figs7ab = import_data('data/figs5cd_figs7ab.json')
```

```
figs6a_old = import_data('data/figs6a_old.json')

figs6b_old = import_data('data/figs6b_old.json')

figs6a = import_data('data/figs6a.json')

figs6b = import_data('data/figs6b.json')

figs8 = import_data('data/figs8.json')

figs9a = import_data('data/figs9a.json')

figs9b = import_data('data/figs9b.json')

figs10a = import_data('data/figs10a.json')
```

4 Figure 1

Figure caption:

Correlated states in near-magic-angle TBG. (A) Longitudinal resistance R_{xx} of the TBG device (measured between contacts separated by 2.15 squares) as a function of carrier density n (shown on the top axis) and perpendicular displacement field D (left axis), which are tuned by the top- and back-gate voltages, at 2.1 K. n is mapped to a filling factor relative to the superlattice density n_s, corresponding to four electrons per moiré unit cell, shown on the bottom axis. (Inset) Optical micrograph of the completed device. The scale bar is 5 μm. (B) Line cut of R_{xx} with respect to n taken at $D/\epsilon_0 = -0.22$ V/nm showing the resistance peaks at full filling of the superlattice, and additional peaks likely corresponding to correlated states emerging at intermediate fillings.

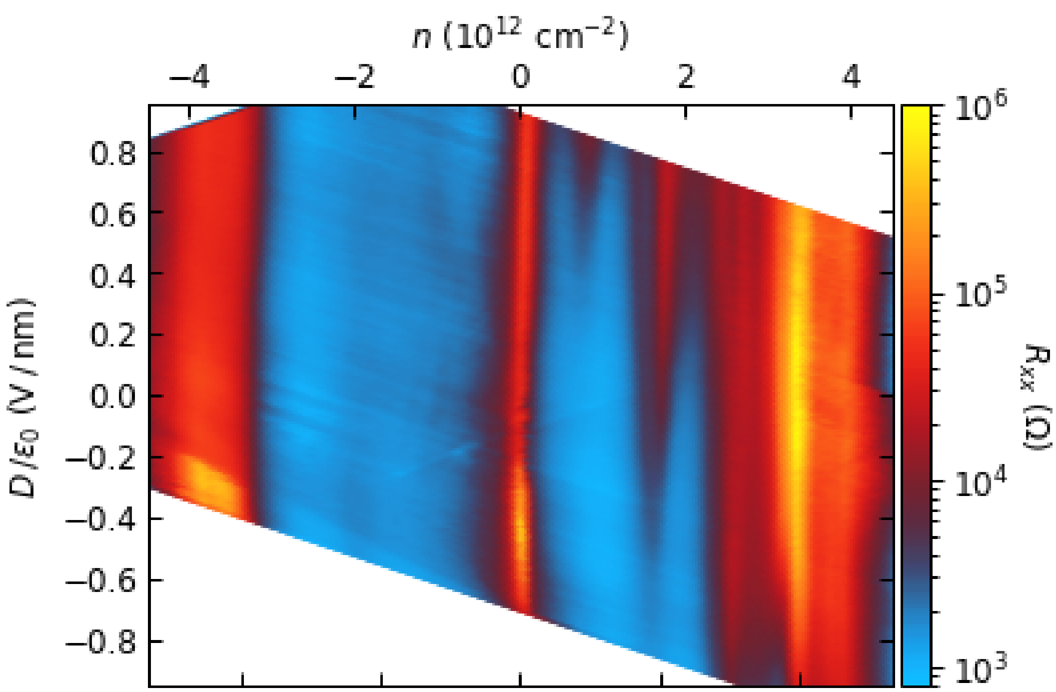
(Inset to A is not included in this notebook.)

4.1 Fig. 1A

[5]:
```
sweep = fig1a

fig, ax = plt.subplots(figsize = (5.9525,4))

plot = plt.pcolormesh(sweep['n']/nsat,sweep['D'],sweep['rxxt'],norm=LogNorm(),
                      shading='gouraud',linewidth=0,rasterized=True);

plt.xlim(-4.5/nsat,4.5/nsat)
ax.tick_params(labelbottom=False)

plt.ylim(-0.95,0.95)
plt.ylabel(r'$D\,/\epsilon_0\ (\mathrm{V\,/\,nm})$')

plot.set_cmap(myColdhot)
cbar = plt.colorbar(pad = 0.01)
```

```
cbar.ax.get_yaxis().labelpad = 15
cbar.set_label(r'$R_{xx}\ (\mathrm{\Omega})$', rotation=270)
cbar.ax.tick_params(which = 'major',direction='out')
cbar.ax.tick_params(which = 'minor',direction='out')
plt.clim(8E2,1E6)
cbar.set_ticks([1E3,1E4,1E5,1E6])
minortick = np.concatenate((np.linspace(8E2,9E2,2),np.linspace(2E3,9E3,8),
                            np.linspace(2E4,9E4,8),np.linspace(2E5,9E5,8)))
minorticklocs = (np.log(minortick)-np.log(8E2))/(np.log(1E6)-np.log(8E2))
cbar.ax.yaxis.set_ticks(minorticklocs, minor=True)

ax1twin = ax.twiny()
ax1twin.plot(range(100), np.ones(100)) # Create a dummy plot
ax1twin.cla()
plt.xlabel(r'$n\ (\mathrm{10^{12}\ cm^{-2}})$')
plt.xlim(-4.5,4.5)
majorLocator = MultipleLocator(0.2)
ax.yaxis.set_major_locator(majorLocator);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig1a.pdf',bbox_inches='tight',dpi=300)
```

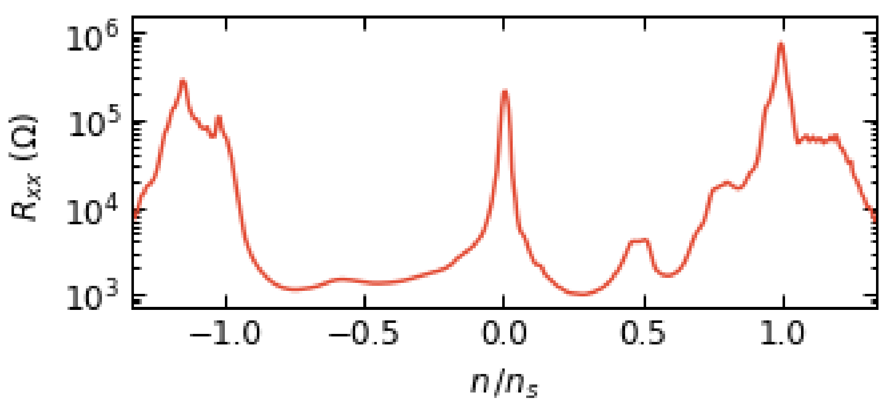
```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



4.2 Fig. 1B

[6]:
```
sweep = fig1b

fig,ax = plt.subplots(figsize = (5,2))
plt.semilogy(sweep['n']/nsat,sweep['rxxt'],color='C0',
             label = r'$D = -0.22\  \mathrm{V/nm}$')

plt.xlabel(r'$n\,/n_s$')
plt.ylabel(r'$R_{xx}\ (\mathrm{\Omega})$')

plt.xlim(-4.5/nsat,4.5/nsat)
plt.ylim(7E2,1.5E6);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig1b.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



5 Figure 2

Figure caption:

Emergent ferromagnetism near three-quarters filling. (A) Magnetic field dependence of the longitudinal resistance R_{xx} (upper panel) and Hall resistance R_{yx} (lower panel) with $n/n_s = 0.746$ and $D/\epsilon_0 = -0.62$ V/nm at 30 mK, demonstrating a hysteretic anomalous Hall effect resulting from emergent magnetic order. The solid and dashed lines correspond to measurements taken while sweeping the magnetic field B up and down, respectively. (B) Zero-field anomalous Hall resistance R_{yx}^{AH} (red) and ordinary Hall slope R_{H} (blue) as a function of n/n_s for $D/\epsilon_0 \approx -0.6$ V/nm. R_{yx}^{AH} is peaked sharply with a maximum around $n/n_s = 0.758$, coincident with R_H changing sign. These parameters are extracted from line fits of R_{yx} versus B on the upward and downward sweeping traces in a region where the B-dependence appears dominated by the ordinary Hall effect. The error bars reflect fitting parameter uncertainty along with the effect of varying the fitting window, and are omitted when smaller than the marker. (C) Temperature dependence of R_{yx} versus B at $D/\epsilon_0 = -0.62$ V/nm and $n/n_s = 0.746$ between 46 mK and 5.0 K, showing the hysteresis loop closing with increasing temperature. Successive curves are offset vertically by 20 k for clarity. (D) Coercive field and anomalous Hall resistance (extracted using the same fitting procedure as above) plotted as a function of temperature from the same data partially shown in (C). Data in Fig. 2 were taken during a separate cooldown from that of the data in the rest of the figures, but show representative behavior (see Supplementary Materials).

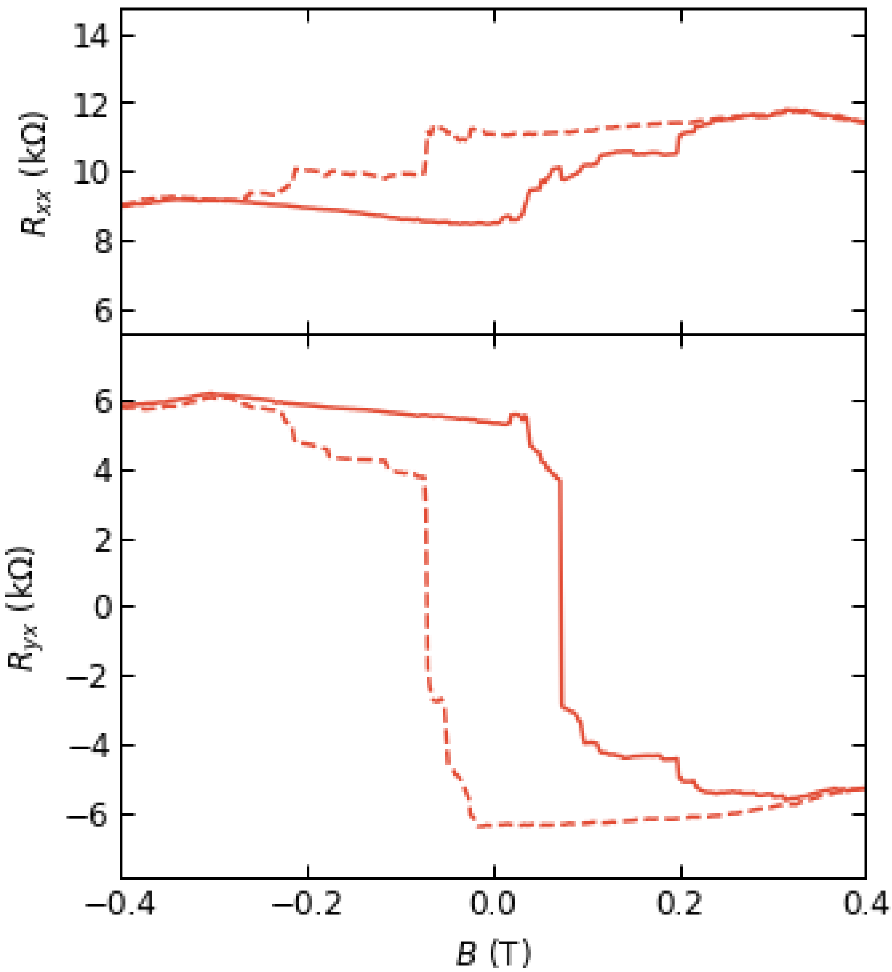
5.1 Fig. 2A

[7]:

```python
sweep = fig2a

fig = plt.figure(figsize = (5,6))
ax1 = fig.add_subplot(8,1,(1,3))
ax2 = fig.add_subplot(8,1,(4,8))
fig.subplots_adjust(hspace=0,wspace=0.4)

###############################
# upper panel -- Rxx
ax1.plot(sweep['B'],sweep['down_rxxt']/1000,color='C0',linestyle='--')
ax1.plot(sweep['B'],sweep['up_rxxt']/1000,color='C0',label=r'$D=-1.31\ \mathrm{V/nm}$')

ax1.set_xlim(-0.4,0.4)
ax1.set_ylim(10-15.8*3/10,10+15.8*3/10)
ax1.set_ylabel(r'$R_{xx}\ (\mathrm{k\Omega})$')

ax1.tick_params(labelbottom=False)

##############################
# lower panel -- Ryx
ax2.plot(sweep['B'],sweep['down_ryxl']/1000,color='C0',linestyle='--')
ax2.plot(sweep['B'],sweep['up_ryxl']/1000,color='C0',label=r'$D=-1.31\ \mathrm{V/nm}$')

majorLocator = MultipleLocator(2)
ax2.yaxis.set_major_locator(majorLocator)

ax2.set_xlim(-0.4,0.4)
ax2.set_xlabel(r'$B\ (\mathrm{T})$')
ax2.set_ylim(-7.9,7.9)
ax2.set_ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$');
```

```
# Uncomment to save a pdf of the figure:
# plt.savefig('fig2a.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

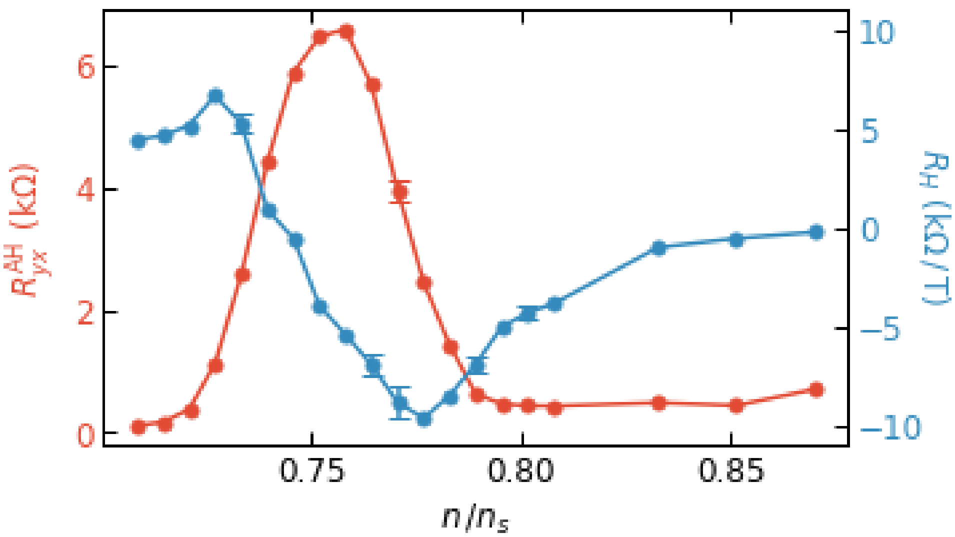


5.2 Fig. 2B

[8]:
```
sweep = fig2b

fig,ax = plt.subplots(figsize = (5,3))

###########################
# Plot anomalous Hall resistance on left axis
```

```
# Make error bar array, setting values to NaN for errors smaller than the␣
↪marker size
errbarAH = np.full(20,np.nan)
errbarAH[10] = sweep['R_AH_sigma'][10]/1000

plt.errorbar(sweep['nu'],sweep['R_AH']/1000,yerr=errbarAH,marker='o',
             markersize=5,capsize=4);

plt.xlabel(r'$n\,/n_s$')
plt.yticks(color='C0')
plt.ylabel(r'$R_{yx}^{\mathrm{AH}}\ (\mathrm{k\Omega})$',color='C0')

###########################
# Plot ordinary Hall coefficient on right axis

# Make error bar array, setting values to NaN for errors smaller than the␣
↪marker size
errbarH = np.full(20,np.nan)
for i in [4,9,10,13,15]:
    errbarH[i] = sweep['R_H_sigma'][i]/1000

ax2t=plt.twinx()
plt.errorbar(sweep['nu'],sweep['R_H']/1000,yerr=errbarH,marker='o',
             markersize=5,capsize=4,color='C1');

plt.axis('tight')
plt.xlabel(r'$n\,/n_s$')
plt.ylim(-11,11)
plt.yticks(color='C1')
plt.ylabel(r'$R_H\ (\mathrm{k\Omega\,/
↪\,T})$',color='C1',rotation=270,labelpad=10);


# Uncomment to save a pdf of the figure:
# plt.savefig('fig2b.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

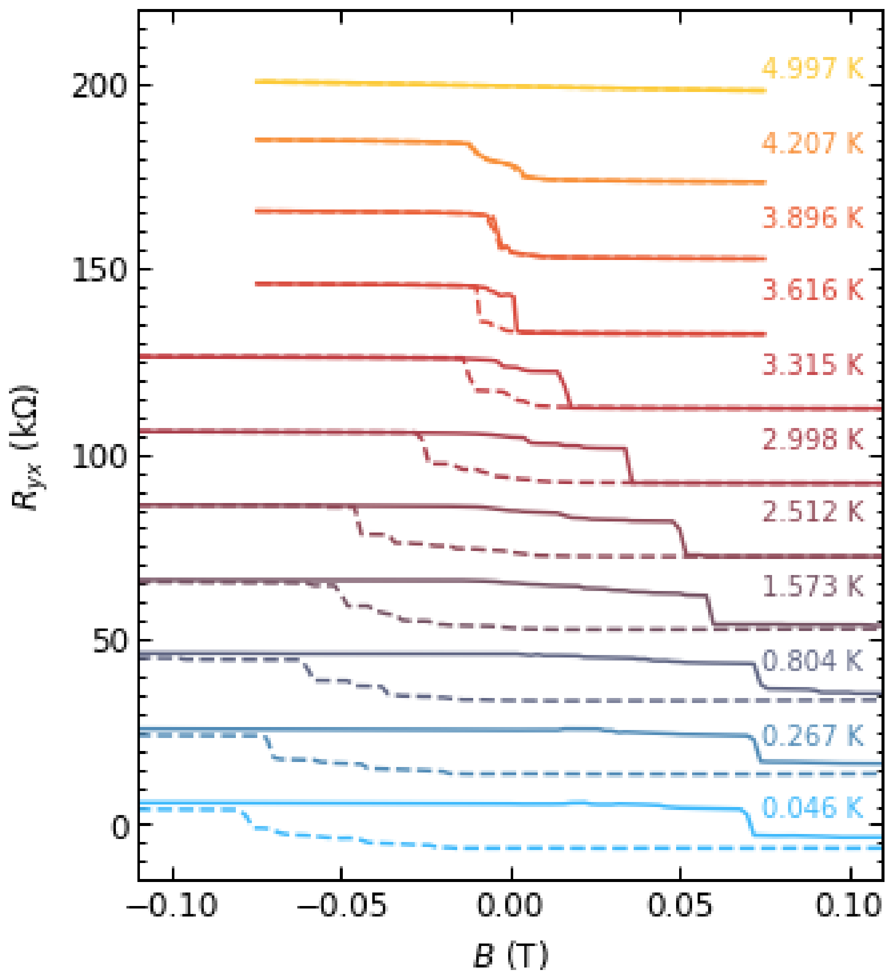


5.3 Fig. 2C

[9]:
```
sweep = fig2cd_figs5ab

fig,ax = plt.subplots(figsize = (5,6))

vertyx = 20 # Offset between sweeps (in kohm)

cmap = coldHot(105,80)
skip = 3 # Controls how many sweeps are plotted
temps = range(np.shape(sweep['B'])[1])
ind = cmap.N/len(temps[::skip])

j = 0
for i in temps[::skip]:
    fig = plt.plot(sweep['B'][:,i],sweep['up_ryxl'][:,i]/
↪1000+vertyx*j,color=cmap((1+ind*j)/255),
                label='%.3f K' % np.ma.masked_invalid(np.
↪asarray(sweep['down_temp'])[:,i]).mean(0))
    plt.plot(sweep['B'][:,i],sweep['down_ryxl'][:,i]/
↪1000+vertyx*j,color=cmap((1+ind*j)/255),linestyle='--')
    ax.text(0.975, 0.095+0.085*j,'%.3f K' % np.ma.
↪masked_invalid(sweep['down_temp'][:,i]).mean(0),
            transform=ax.transAxes,fontsize=10, va='top',␣
↪ha='right',color=cmap((1+ind*j)/255))
    j=j+1

plt.axis('tight')
```

```
plt.xlim(-0.11,0.11)
plt.ylim(-15,220)

plt.xlabel(r'$B\ (\mathrm{T})$')
plt.ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$')
handles, labels = ax1.get_legend_handles_labels()


majorLocator = MultipleLocator(0.05)
ax.xaxis.set_major_locator(majorLocator)

minorLocator = MultipleLocator(5)
ax.yaxis.set_minor_locator(minorLocator);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig2c.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



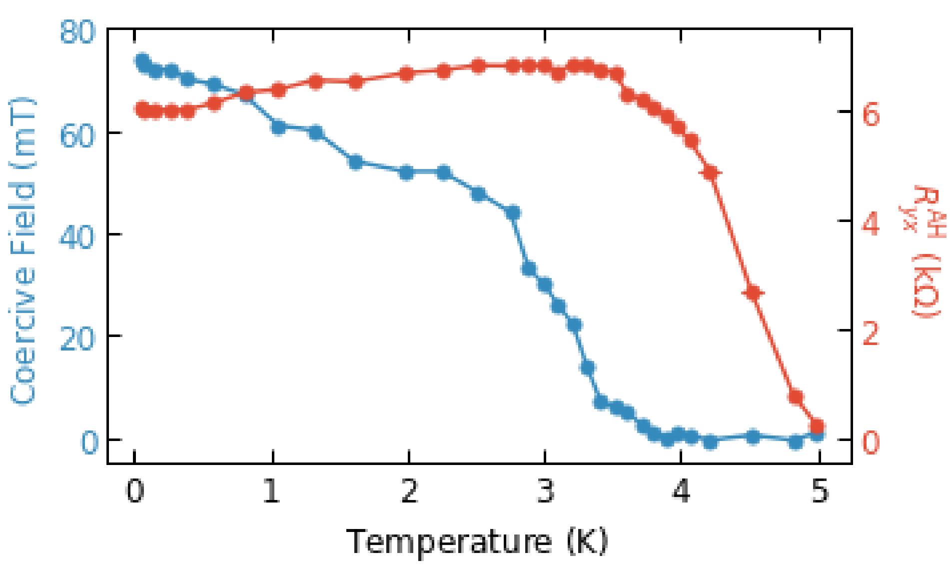
5.4 Fig. 2D

[10]:

```
sweep = fig2cd_figs5ab
fig,ax = plt.subplots(figsize = (5,3))

################################
# Plot coercive field on left axis

plt.plot(sweep['down_temp'][0,:-1],1000*sweep['Bcorctemp'][:
 ↪-1],'-o',markersize=5,color='C1')
plt.axis('tight')
plt.xlabel(r'$\mathrm{Temperature\ (K)}$')
plt.ylim(-5,80)
plt.yticks(color='C1')
plt.ylabel(r'$\mathrm{Coercive\ Field}\ (\mathrm{mT})$',color='C1')
```

```
###########################
# Plot anomalous Hall resistance on right axis

# Make error bar array, setting values to NaN for errors smaller than the␣
↪marker size
errbar = np.full(32,np.nan)
for i in [27,28]:
    errbar[i] = sweep['R_AH_sigma'][i]/1000

ax2t=plt.twinx()
ax2t.set_zorder(10)
plt.errorbar(sweep['down_temp'][0,:-1],sweep['R_AH'][:-1]/1000,yerr=errbar[:
↪-1],marker='o',
             markersize=5,capsize=4,color='C0');


plt.ylim(-0.46875,7.5)
plt.yticks(color='C0')
plt.ylabel(r'$R_{yx}^{\mathrm{AH}}\␣
↪(\mathrm{k\Omega})$',color='C0',rotation=270,labelpad=25);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig2d.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



6 Figure 3

Figure caption:

Nonlocal resistances providing evidence of chiral edge states. (A, B) Three- and four-terminal nonlocal resistances $R_{54,14}$ (A) and $R_{54,12}$ (B), measured at 2.1 K with $D/\epsilon_0 = -0.22$ V/nm, are shown in the upper and lower panels, respectively. For $n/n_s = 0.725$ (blue) away from the peak in AH resistance R_{yx}^{AH}, the nonlocal resistances are consistent with diffusive bulk transport. However, with $n/n_s = 0.749$ (red) in the magnetic regime where R_{yx}^{AH} is maximal, large, hysteretic nonlocal resistances suggest chiral edge states are present. Insets: Schematics of the respective measurement configurations. Green arrows in the upper inset represent the apparent edge state chirality for positive magnetization, whereas in the lower inset they reflect negative magnetization.

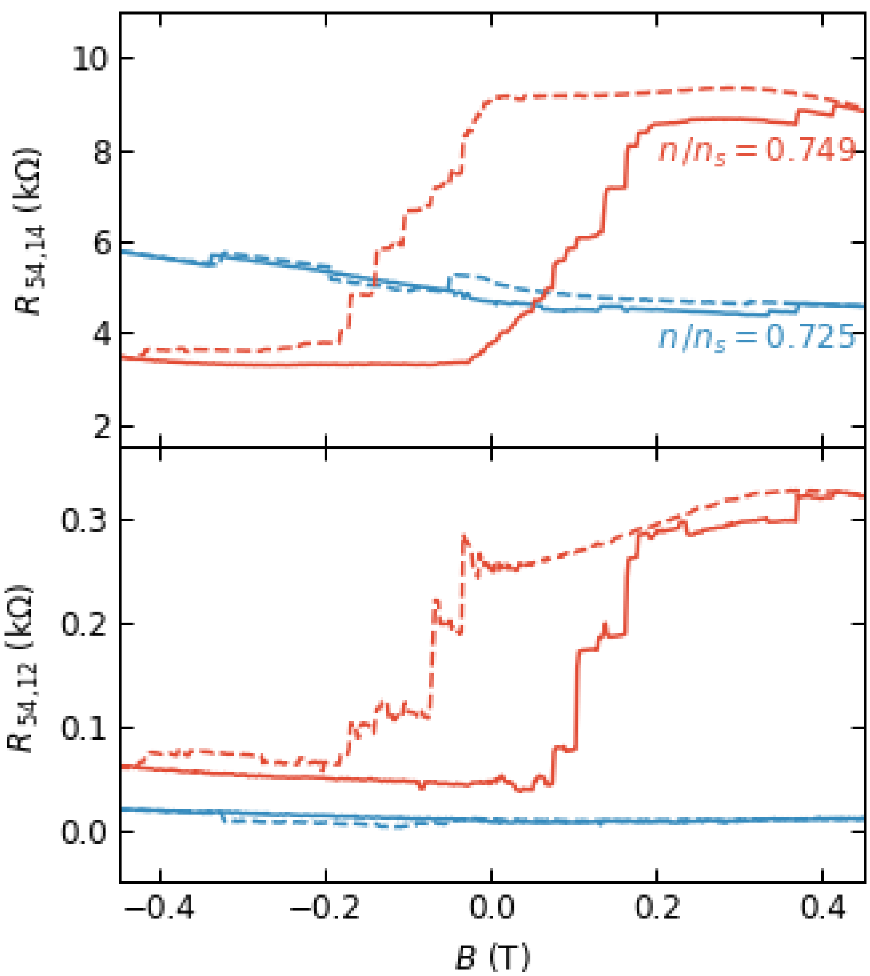
(Insets are not included in this notebook.)

[11]:
```
sweep = fig3

fig = plt.figure(figsize = (5,6))
ax1 = fig.add_subplot(2,2,(1,2))
ax2 = fig.add_subplot(2,2,(3,4))
fig.subplots_adjust(hspace=0,wspace=0.4)

###########################
# Upper panel -- R_{54,14}

colors = ['C1','C0']

for i in [0,1]:
    ax1.plot(sweep['B'][:,i],sweep['up_r1'][:,i]/1000,color=colors[i])
    ax1.plot(sweep['B'][:,i],sweep['down_r1'][:,i]/
 ↪1000,color=colors[i],linestyle='--')

xmag=0.45
ax1.set_xlim(-xmag,xmag)
ax1.set_ylim(1.5,11)
ax1.set_ylabel(r'$R\,_{54,\!14}\ (\mathrm{k\Omega})$')
txt1 = ax1.text(0.2,3.6,r'$n\,/n_s=0.725$',color=colors[0])
txt2 = ax1.text(0.2,7.7,r'$n\,/n_s=0.749$',color=colors[1])



###########################
# Lower panel -- R_{54,12}

for i in [0,1]:
    ax2.plot(sweep['B'][:,i],sweep['up_r12'][:,i]/1000,color=colors[i])
    ax2.plot(sweep['B'][:,i],sweep['down_r12'][:,i]/
 ↪1000,color=colors[i],linestyle='--')
```

```
ax1.tick_params(labelbottom=False)
ax2.set_xlim(-xmag,xmag)
ax2.set_xlabel(r'$B\ (\mathrm{T})$')
ax2.set_ylim(-0.05,0.37);
ax2.set_ylabel(r'$R\,_{54,\!12}\ (\mathrm{k\Omega})$')

# Uncomment to save a pdf of the figure:
# plt.savefig('fig3.pdf',bbox_inches='tight')
```

[11]: Text(0,0.5,'$R\\,_{54,\\!12}\\ (\\mathrm{k\\Omega})$')

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



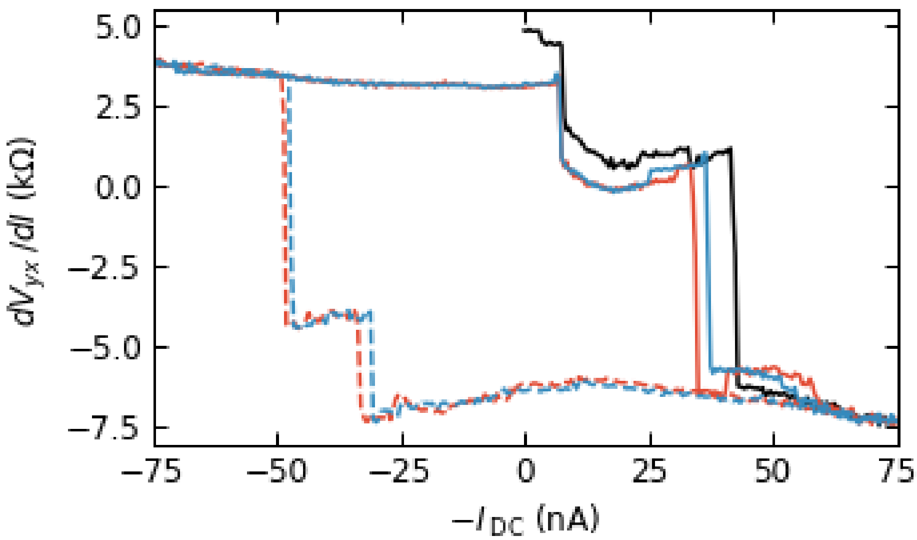
7 Figure 4

Figure caption:

Current-driven switching of the magnetization. Differential Hall resistance dV_{yx}/dI measured with a 5 nA AC bias as a function of an applied DC current I_{DC} at 2.1 K with $D/\epsilon_0 = -0.22$ V/nm and $n/n_s = 0.749$. After magnetizing the sample in a -500 mT field and returning to $B = 0$, I_{DC} was swept from 0 to -50 nA (black trace), resulting in dV_{yx}/dI changing sign. Two successive loops in I_{DC} between ± 50 nA demonstrate reversible and repeatable switching of the differential Hall resistance (red and blue, with solid and dashed traces corresponding to opposite sweep directions). Note that dV_{yx}/dI is plotted against $-I_{\mathrm{DC}}$ for better comparison with magnetic field hysteresis loops.

[12]:
```
sweep = fig4

sweepdirs = ['d1','u1','d2','u2','d3']
col = ['k','C0','C0','C1','C1']
linesty = ['-','--','-','--','-']
i = 0

fig, ax = plt.subplots(figsize = (5,3))
for sweepdir in sweepdirs:
    plt.plot(-sweep[sweepdir+'_Idc']*1E9,sweep[sweepdir+'_ryxl'][:,1]/
 ↪1000,color = col[i],linestyle=linesty[i])
    i=i+1

plt.axis('tight')
plt.xlabel(r'$-I\,_{\mathrm{DC}}\ (\mathrm{nA})$')
plt.ylabel(r'$dV_{yx}\,/dI\ (\mathrm{k\Omega})$')
plt.xlim(-75,75);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig4.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



8 Figure S2

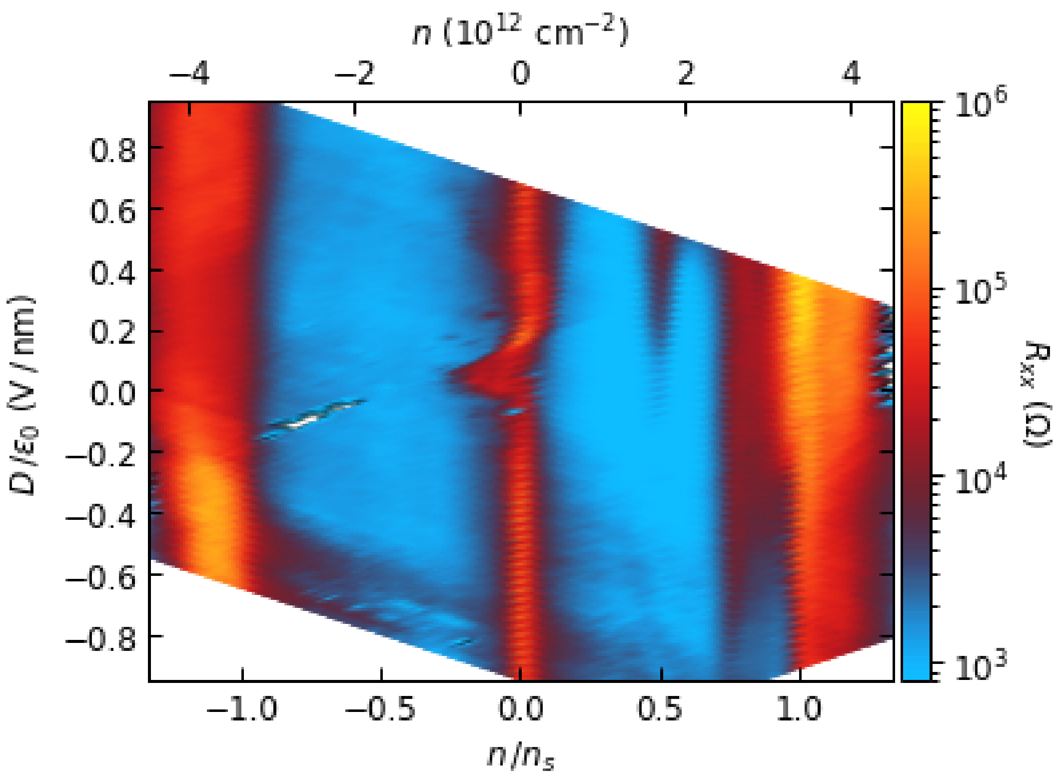
Figure caption:

Displacement field dependence from separate cooldown. Longitudinal resistance R_{xx} of the TBG device (measured between contacts separated by 2.15 squares at 40 mK) as a function of carrier density n (shown on the top axis), filling factor relative to the superlattice density n_s (bottom axis), and the applied perpendicular displacement field D (left axis). This cooldown was separate from either of those described in the main text.

[13]:
```
sweep = figs2

fig, ax = plt.subplots(figsize = (5.9525,4))
plot = plt.pcolormesh(sweep['nu'],sweep['D'],sweep['rxxt'],norm=LogNorm(),
                      shading='gouraud',linewidth=0,rasterized=True);

plt.xlim(-4.5/nsat,4.5/nsat)
plt.xlabel(r'$n\,/n_s$')
plt.ylim(-0.95,0.95)
plt.ylabel(r'$D\,/\epsilon_0\ (\mathrm{V\,/\,nm})$')

plot.set_cmap(myColdhot)
cbar = plt.colorbar(pad = 0.01)
cbar.ax.get_yaxis().labelpad = 15
cbar.set_label(r'$R_{xx}\ (\mathrm{\Omega})$', rotation=270)
cbar.ax.tick_params(which = 'major',direction='out')
cbar.ax.tick_params(which = 'minor',direction='out')
plt.clim(8E2,1E6)
```

```
cbar.set_ticks([1E3,1E4,1E5,1E6])
minortick = np.concatenate((np.linspace(8E2,9E2,2),np.linspace(2E3,9E3,8),
                            np.linspace(2E4,9E4,8),np.linspace(2E5,9E5,8)))
minorticklocs = (np.log(minortick)-np.log(8E2))/(np.log(1E6)-np.log(8E2))
cbar.ax.yaxis.set_ticks(minorticklocs, minor=True)

ax1twin = ax.twiny()
ax1twin.plot(range(100), np.ones(100)) # Create a dummy plot
ax1twin.cla()
plt.xlabel(r'$n\ (\mathrm{10^{12}\ cm^{-2}})$')
plt.xlim(-4.5,4.5)
majorLocator = MultipleLocator(0.2)
ax.yaxis.set_major_locator(majorLocator);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S2.pdf',bbox_inches='tight',dpi=300)
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



9 Figure S3

Figure caption:

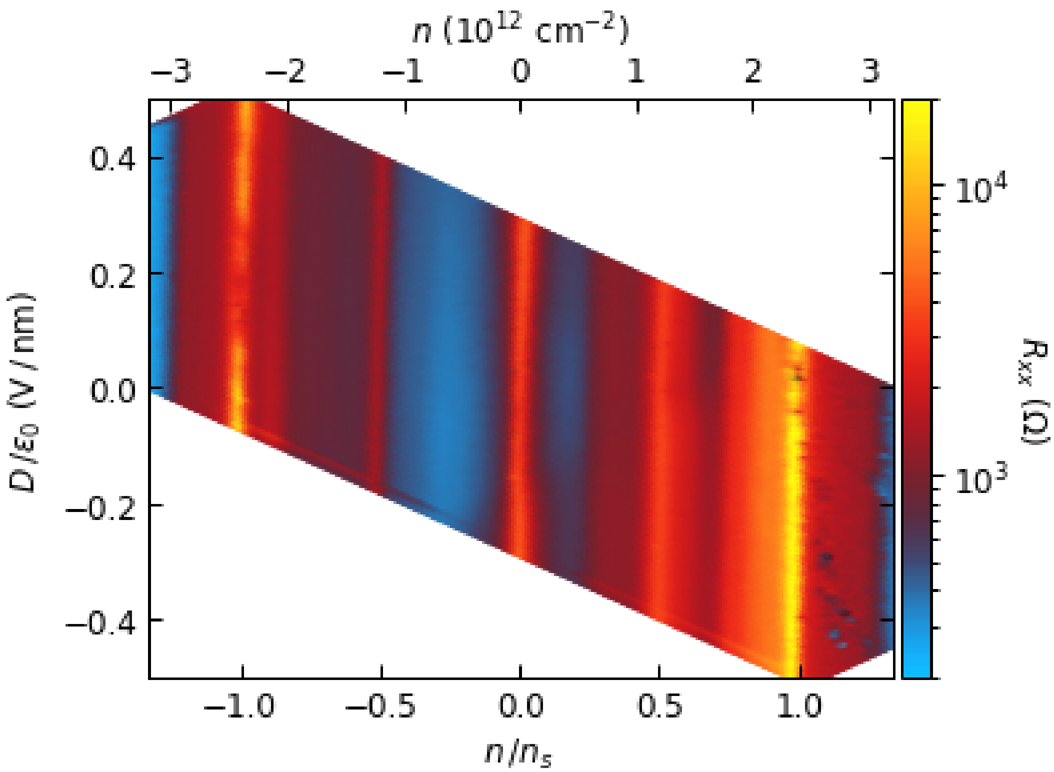
Displacement field dependence of TBG device misaligned to hBN. (A) An optical micrograph of the completed device. The scale bar is 20 μm. (B) Longitudinal resistance R_{xx} of the misaligned TBG device (measured between contacts separated by 1.25 squares at 1.5 K) as a function of carrier density n (shown on the top axis, or as a filling factor relative to the superlattice density shown on the bottom axis) and perpendicular displacement field D.

(Micrograph (A) is not included in this notebook.)

[14]:
```
sweep = figs3

nsat_tbg2=2.4

fig, ax = plt.subplots(figsize = (5.9525,4))
plot = plt.pcolormesh(sweep['nu'],sweep['D'],sweep['rxx78'],norm=LogNorm(),
                      shading='gouraud',linewidth=0,rasterized=True);

plt.xlim(-3.2/nsat_tbg2,3.2/nsat_tbg2)
plt.xlabel(r'$n\,/n_s$')

plt.ylim(-0.5,0.5)
plt.ylabel(r'$D\,/\epsilon_0\ (\mathrm{V\,/\,nm})$')

plot.set_cmap(myColdhot)
cbar = plt.colorbar(pad = 0.01)
cbar.ax.get_yaxis().labelpad = 15
cbar.set_label(r'$R_{xx}\ (\mathrm{\Omega})$', rotation=270)
cbar.ax.tick_params(which = 'major',direction='out')
cbar.ax.tick_params(which = 'minor',direction='out')
plt.clim(2E2,2E4)

cbar.set_ticks([1E3,1E4])
minortick = np.concatenate((np.linspace(2E2,9E2,8),np.
 ↪linspace(2E3,9E3,8),[2E4]))
minorticklocs = (np.log(minortick)-np.log(2E2))/(np.log(2E4)-np.log(2E2))
cbar.ax.yaxis.set_ticks(minorticklocs, minor=True)

ax1twin = ax.twiny()
ax1twin.plot(range(100), np.ones(100)) # Create a dummy plot
ax1twin.cla()
plt.xlabel(r'$n\ (\mathrm{10^{12}\ cm^{-2}})$')
plt.xlim(-3.2,3.2)
majorLocator = MultipleLocator(0.2)
ax.yaxis.set_major_locator(majorLocator);
```

```
# Uncomment to save a pdf of the figure:
# plt.savefig('fig_3.pdf',bbox_inches='tight',dpi=300)
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



10 Figure S4

Figure caption:

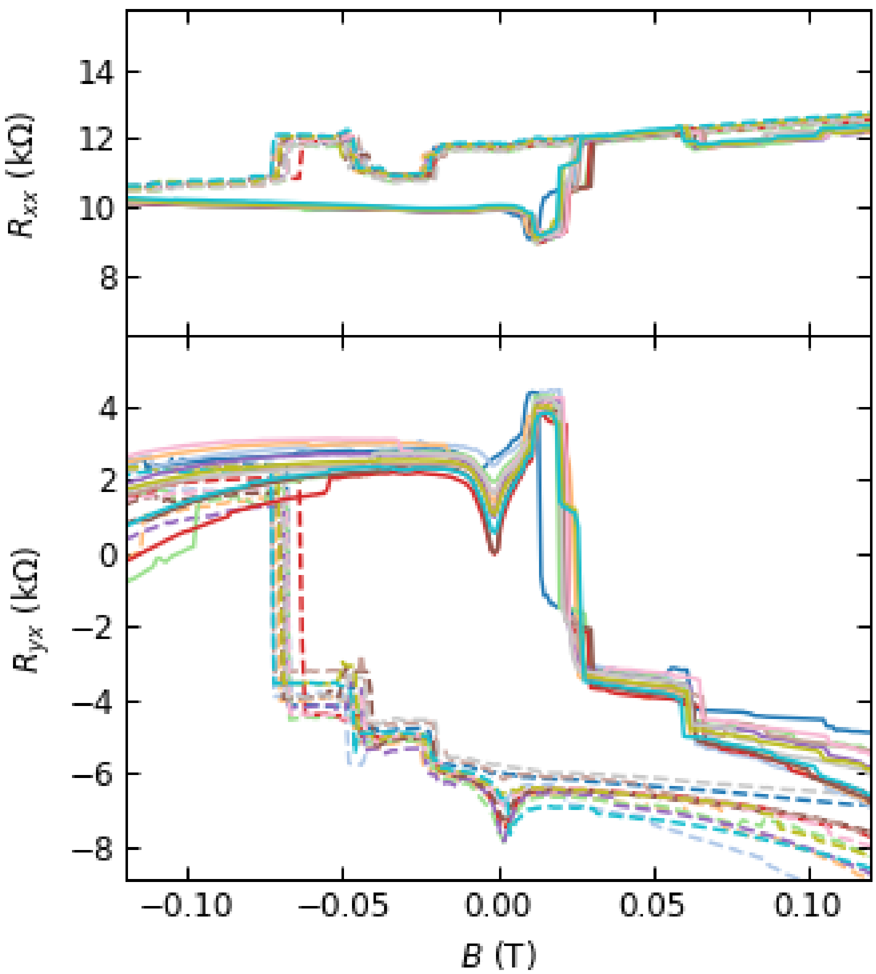
Repeated hysteresis loops. Longitudinal resistance R_{xx} (top panel) and Hall resistance R_{yx} (bottom panel) are shown as a function of magnetic field for twelve consecutive loops of the field between ± 250 mT for $n/n_s = 0.758$ and $D/\epsilon_0 = 0$ V/nm (in the same cooldown as that of Fig. S2). The solid and dashed lines correspond to measurements taken while sweeping the magnetic field B up and down, respectively.

[15]:
```
sweep = figs4

fig = plt.figure(figsize = (5,6))
ax1 = fig.add_subplot(8,1,(1,3))
ax2 = fig.add_subplot(8,1,(4,8))
```

```
fig.subplots_adjust(hspace=0,wspace=0.4)

###################################################################
# Upper panel -- Rxx

col = plt.get_cmap('tab20')
for i in range(np.shape(sweep['B'])[1]):
    ax1.plot(sweep['B'][:,i],sweep['down_rxxt'][:,i]/1000,linestyle='--',color␣
 ↪= col(i/12))

for i in range(np.shape(sweep['B'])[1]):
    ax1.plot(sweep['B'][:,i],sweep['up_rxxt'][:,i]/1000,color=col(i/12))

ax1.set_xlim(-0.12,0.12)
ax1.set_ylim(11-15.8*3/10,11+15.8*3/10)
ax1.set_ylabel(r'$R_{xx}\ (\mathrm{k\Omega})$',labelpad=10)

###################################################################
# Lower panel -- Ryx

for i in range(np.shape(sweep['B'])[1]):
    ax2.plot(sweep['B'][:,i],sweep['down_ryxl'][:,i]/1000,linestyle='--',color␣
 ↪= col(i/12))

for i in range(np.shape(sweep['B'])[1]):
    ax2.plot(sweep['B'][:,i],sweep['up_ryxl'][:,i]/1000,color = col(i/12))

ax2.set_xlim(-0.12,0.12)
ax2.set_xlabel(r'$B\ (\mathrm{T})$')
ax2.set_ylim(-8.9,5.9);
ax2.set_ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$')



# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S4.pdf',bbox_inches='tight')
```

[15]:
```
Text(0,0.5,'$R_{yx}\\ (\\mathrm{k\\Omega})$')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



11 Figure S5

Figure caption:

Behavior of the conductivity tensor. (A) The longitudinal conductivity σ_{xx} is plotted parametrically against the Hall conductivity σ_{yx} for a series of measurements at different temperatures with the density fixed at $n/n_s = 0.746$ and $D/\epsilon_0 = -0.62$ V/nm (shown in Fig.~2C of the main text). All conductivity values in this figure have been extracted from resistance measurements taken at 50 mT when sweeping the applied field downward from a value larger than the coercive field (so the sample has been magnetized by an upward field). The resistivity is derived from the measurements by assuming a homogeneous sample, and the conductivities are given by $\sigma_{xx} = \rho_{xx}/\left(\rho_{xx}^2 + \rho_{yx}^2\right)$ and $\sigma_{xy} = \rho_{yx}/\left(\rho_{xx}^2 + \rho_{yx}^2\right)$. The relationship between σ_{xy} and σ_{xx} is not consistent with an extrinsic AH effect resulting from skew scattering. (B). Arrhenius plot of σ_{xx} on a log scale versus $1/T$, with the same data shown in (A). The blue curve shows a fit of the data for $4.9\text{ K} > T > 1.9\text{ K}$ (the points shown in blue) to a model of activated conductivity with an additional, temperature-independent conduction channel (data points shown in red are excluded from the fit), yielding an estimated activation scale of $T_0 = 44$ K. (Inset) σ_{xx} is plotted on a linear scale against temperature. (C) σ_{xx} is plotted parametrically against σ_{xy} for a series of measurements at different densities at $T = 2.1$ K, with $D/\epsilon_0 = -0.22$ V/nm. These data were obtained during the same cooldown as that of the data shown in Figs.~1, 3, and 4 of the main text. Again, the behavior appears inconsistent with skew scattering. Moreover, it is qualitatively similar to the density dependence of the conductivity in a magnetic topological insulator approaching a QAH effect shown in Ref. (Checkelsky, 2014). (D) σ_{xx} as a function of n/n_s, from the same data as in (C), showing the emergence of a dip in σ_{xx} around $n/n_s = 3/4$ consistent with the approach to a Chern insulator state.

11.1 Calulating ρ and σ from temperature series for figures A and B

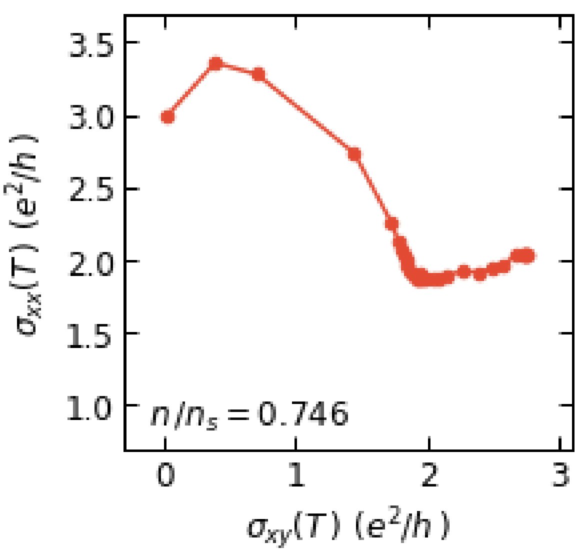
[16]:
```
sweep = fig2cd_figs5ab

# Set variables for subfigure size: (xsz,ysz)
xsz = 3
ysz = 3

# Generate arrays for resistivity, conductivity, and temperature
rhoxxstore = []
rhoyxstore = []
sigmaxxstore = []
sigmaxystore = []
tempstore = []

for i in range(np.shape(sweep['B'])[1]):
    down_rxx = sweep['down_rxxt'][~np.isnan(sweep['down_rxxt'][:,i]),i]
    up_rxx = sweep['up_rxxt'][~np.isnan(sweep['up_rxxt'][:,i]),i]
    down_ryx = sweep['down_ryxl'][~np.isnan(sweep['down_ryxl'][:,i]),i]
    up_ryx = sweep['up_ryxl'][~np.isnan(sweep['up_ryxl'][:,i]),i]
    down_B = sweep['B'][~np.isnan(sweep['B'][:,i]),i]

    # Calculate symmetrized longitudinal resistivity
    down_rhoxx = (down_rxx+up_rxx[::-1])/(2*numsq)
    up_rhoxx = (down_rxx[::-1]+up_rxx)/(2*numsq)

    temp = sweep['down_temp'][~np.isnan(sweep['down_temp'][:,i]),i].mean(0)

    # Calculate antisymmetrized Hall resistivity
    down_rhoyx = (down_ryx-up_ryx[::-1])/(2.)
    up_rhoyx = -(down_ryx[::-1]-up_ryx)/(2.)

    # Calculate conductivity in e^2/h from resistivities
    down_sigmaxx = down_rhoxx/(down_rhoxx**2+down_rhoyx**2)*Rq
    up_sigmaxx = up_rhoxx/(up_rhoxx**2+up_rhoyx**2)*Rq
    down_sigmaxy = down_rhoyx/(down_rhoxx**2+down_rhoyx**2)*Rq
    up_sigmaxy = up_rhoyx/(up_rhoxx**2+up_rhoyx**2)*Rq
```

```
    # Store symmetrized/antisymmetrized values at 50 mT on downward field sweep
    index = np.argmin(np.abs(down_B-0.05))
    rhoxxstore.append(down_rhoxx[index])
    rhoyxstore.append(down_rhoyx[index])
    sigmaxxstore.append(down_sigmaxx[index])
    sigmaxystore.append(down_sigmaxy[index])
    tempstore.append(temp)
```

11.1.1 Fig. S5A

[17]:
```
fig,ax1 = plt.subplots(figsize = (xsz,ysz))
plt.plot(-np.asarray(sigmaxystore),sigmaxxstore,'.-',markersize=9)
plt.xlim(-0.3,3.1)
plt.ylim(0.7,3.7)
plt.xlabel(r'$\sigma_{xy}(T\,)\ ({e^2/h}\,)$')#,fontsize=36)
plt.ylabel(r'$\sigma_{xx}(T\,)\ ({e^2/h}\,)$')#,fontsize=36)
txt = ax1.text(0.05,0.05,r'$n\,/n_s=0.746$',transform=ax1.transAxes);

# Uncomment to save a pdf of the figure:
# plt.savefig('figs_sig_xxvxy_t.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



11.1.2 Fig. S5B

[18]:
```
# Fitting conductivity as a function of temperature

# Model
def func(x,T0,A,B):
    return A*np.exp(-T0*x)+B

indlo= 12 # Low temperature cutoff for fit, ~2.5 K
indhi = 30 # High temperature cutoff for fit, ~4.9 K

popt, pcov = curve_fit(func, [1/t for t in tempstore[indlo:indhi]],␣
↪sigmaxxstore[indlo:indhi],
                       sigma=[y for y in sigmaxxstore[indlo:indhi]])
print('Fitting parameters and error estimates, for sigma_xx = A*exp(-T0/T)+B')
print('T0, A, B')
print(popt)
print(np.sqrt(np.diag(pcov)))


#########################
# Arrhenius plot of conductivity (log scale) vs. 1/T

fig,ax1 = plt.subplots(figsize = (xsz,ysz))
plt.semilogy([1/t for t in tempstore[indhi:]],sigmaxxstore[indhi:],'.
↪',markersize=9)
plt.semilogy([1/t for t in tempstore[indlo:indhi]],sigmaxxstore[indlo:indhi],'.
↪',markersize=9)
plt.semilogy([1/t for t in tempstore[:indlo]],sigmaxxstore[:indlo],'.
↪',markersize=9,color='C0')

xval = np.linspace(0,10,1000)
plt.plot(xval,func(xval, *popt), color='C1')


plt.xlim(0.12,0.47)
plt.ylim(1.7,3.9)
plt.xlabel(r'$1/T\ (\mathrm{K^{-1}})$')
plt.ylabel(r'$\sigma_{xx}\ ({e^2/h}\,)$ (log)')
txt = ax1.text(0.05,0.05,r'$n\,/n_s=0.746$',transform=ax1.transAxes)
ax1.yaxis.set_minor_formatter(ScalarFormatter())
ax1.yaxis.set_minor_formatter(matplotlib.ticker.FormatStrFormatter("%d"));

# Uncomment to save a pdf of the figure:
# plt.savefig('figs_sig_arr_fit.pdf',bbox_inches='tight')
```

```
Fitting parameters and error estimates, for sigma_xx = A*exp(-T0/T)+B
T0, A, B
```

```
[4.32418590e+01 1.13827289e+04 1.86632685e+00]
[1.27493082e+00 3.10249954e+03 6.80353039e-03]
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

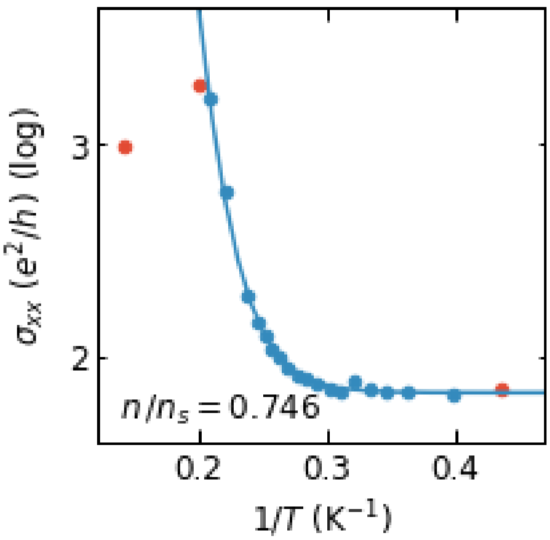
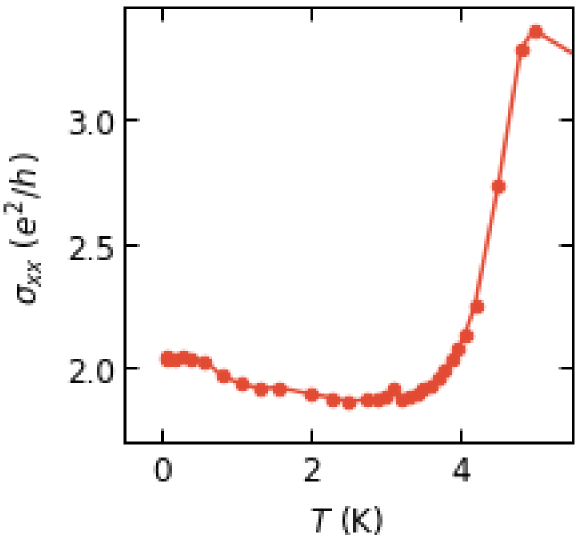


Fig. S5B, inset

[19]:
```
fig,ax1 = plt.subplots(figsize = (xsz,ysz))
plt.plot(tempstore,sigmaxxstore,'.-',markersize=9)

plt.xlim(-0.5,5.5)
plt.xlabel(r'$T\ (\mathrm{K})$')
plt.ylim(1.7,3.45)
plt.ylabel(r'$\sigma_{xx}\ ({e^2/h}\,)$')
ax1.yaxis.set_minor_formatter(ScalarFormatter())
ax1.yaxis.set_minor_formatter(matplotlib.ticker.FormatStrFormatter("%d"));

# Uncomment to save a pdf of the figure:
# plt.savefig('figs_sig_arr.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



11.2 Calulating ρ and σ from density series for figures C and D

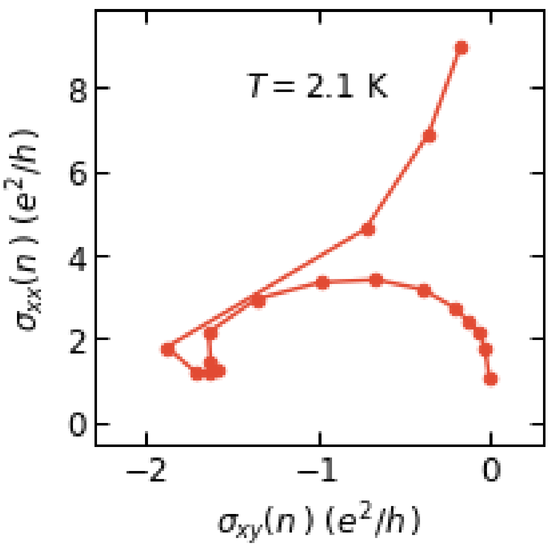
[20]:
```
sweep = figs5cd_figs7ab

# Set variables for subfigure size: (xsz,ysz)
xsz = 3
ysz = 3

# Generate arrays for resistivity, conductivity, and density
rhoxxstore = []
rhoyxstore = []
sigmaxxstore = []
sigmaxystore = []
densstore = []

for i in range(0,np.shape(sweep['B'])[1]):
    down_rxx = sweep['down_rxxt'][~np.isnan(sweep['down_rxxt'][:,i]),i]
    up_rxx = sweep['up_rxxt'][~np.isnan(sweep['up_rxxt'][:,i]),i]
    down_ryx = sweep['down_ryxl'][~np.isnan(sweep['down_ryxl'][:,i]),i]
    up_ryx = sweep['up_ryxl'][~np.isnan(sweep['up_ryxl'][:,i]),i]
    down_B = sweep['B'][~np.isnan(sweep['B'][:,i]),i]

    # Calculate symmetrized longitudinal resistivity
    down_rhoxx = (down_rxx+up_rxx[::-1])/(2*numsq)
    up_rhoxx = (down_rxx[::-1]+up_rxx)/(2*numsq)

    # Calculate antisymmetrized Hall resistivity
    down_rhoyx = (down_ryx-up_ryx[::-1])/(2.)
```

```
    up_rhoyx = -(down_ryx[::-1]-up_ryx)/(2.)

    # Calculate conductivity in e^2/h from resistivities
    down_sigmaxx = down_rhoxx/(down_rhoxx**2+down_rhoyx**2)*Rq
    up_sigmaxx = up_rhoxx/(up_rhoxx**2+up_rhoyx**2)*Rq
    down_sigmaxy = down_rhoyx/(down_rhoxx**2+down_rhoyx**2)*Rq
    up_sigmaxy = up_rhoyx/(up_rhoxx**2+up_rhoyx**2)*Rq

    # Store symmetrized/antisymmetrized values at 50 mT on downward field sweep
    index = np.argmin(np.abs(down_B-0.05))
    rhoxxstore.append(down_rhoxx[index])
    rhoyxstore.append(down_rhoyx[index])
    sigmaxxstore.append(down_sigmaxx[index])
    sigmaxystore.append(down_sigmaxy[index])
    densstore.append(sweep['n'][i]/nsat)
```

11.2.1 Fig. S5C

```
[21]: fig,ax1 = plt.subplots(figsize = (xsz,ysz))
plt.plot(sigmaxystore,sigmaxxstore,'.-',markersize=9)
plt.xlim(-2.3,0.3)
plt.ylim(-0.5,9.9)
plt.xlabel(r'$\sigma_{xy}(n\,)\ ({e^2/h}\,)$')
plt.ylabel(r'$\sigma_{xx}(n\,)\ ({e^2/h}\,)$')
txt = ax1.text(0.5,0.8,r'$T = 2.1\␣
↪\mathrm{K}$',horizontalalignment='center',transform=ax1.transAxes);

# Uncomment to save a pdf of the figure:
# plt.savefig('figs_sig_xxvxy_n.pdf',bbox_inches='tight')
```

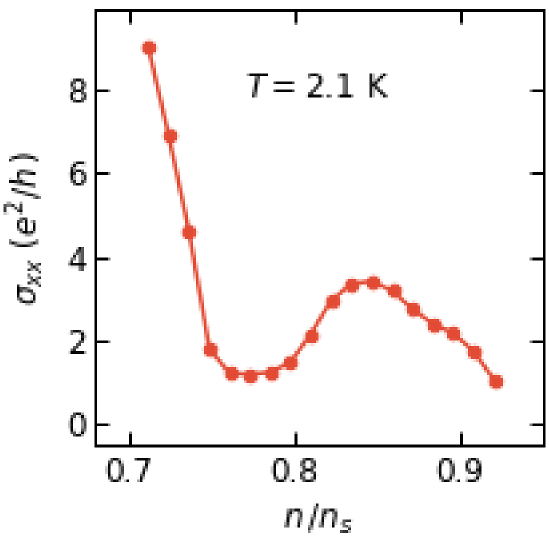
```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



11.2.2 Fig. S5D

[22]:
```
fig,ax1 = plt.subplots(figsize = (xsz,ysz))
plt.plot(densstore,sigmaxxstore,'.-',markersize=9)
plt.xlim(0.68,0.95)
plt.ylim(-0.5,9.9)
plt.xlabel(r'$n\,/n_s$')#,fontsize=36)
plt.ylabel(r'$\sigma_{xx}\ ({e^2/h}\,)$')#,fontsize=36)
txt = ax1.text(0.5,0.8,r'$T = 2.1\
 ↪\mathrm{K}$',horizontalalignment='center',transform=ax1.transAxes);

# Uncomment to save a pdf of the figure:
# plt.savefig('figs_sig_xxvn.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



12 Figure S6 (version 1)

This version of the figure was included in the original version of the manuscript posted to arXiv. It has been replaced by an updated version using different data, included below.

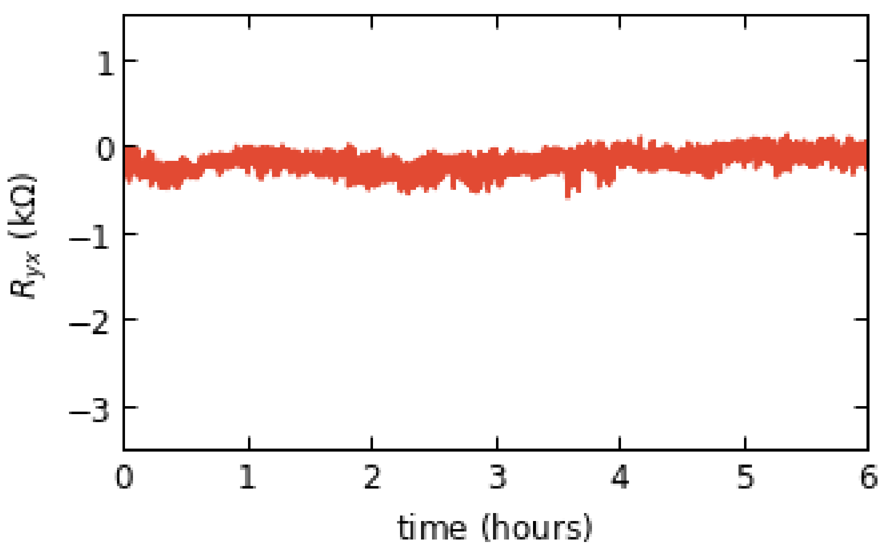
Figure caption:

Temporal stability of the magnetization. (A) Hall resistance R_{yx} at $n/n_s = 0.746$ and $D/\epsilon_0 = -0.62$ V/nm as a function of time over the course of 6 hours in zero field, after first magnetizing the sample by applying -250 mT and then returning the field to 0 T. (B) A full hysteresis loop taken prior to the measurement shown in (A) is displayed in red. The blue trace shows the behavior of R_{yx} as the field is swept from 0 to 250 mT following the measurement in (A). A clear anomalous Hall jump in the blue trace is comparable to those in the continuous red loop, indicating that the magnetization was stable through the 6 hour pause.

12.1 Fig. S6A

[23]:
```
sweep = figs6a_old

fig = plt.figure(figsize = (5,3))
plt.plot(sweep['time']/3600.,sweep['ryxl']/1000,color='C0')
plt.axis('tight')
plt.xlim(0,6)
plt.ylim(-3.5,1.5)
plt.xlabel(r'$\mathrm{time}\ (\mathrm{hours})$')
plt.ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$');

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S6a.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

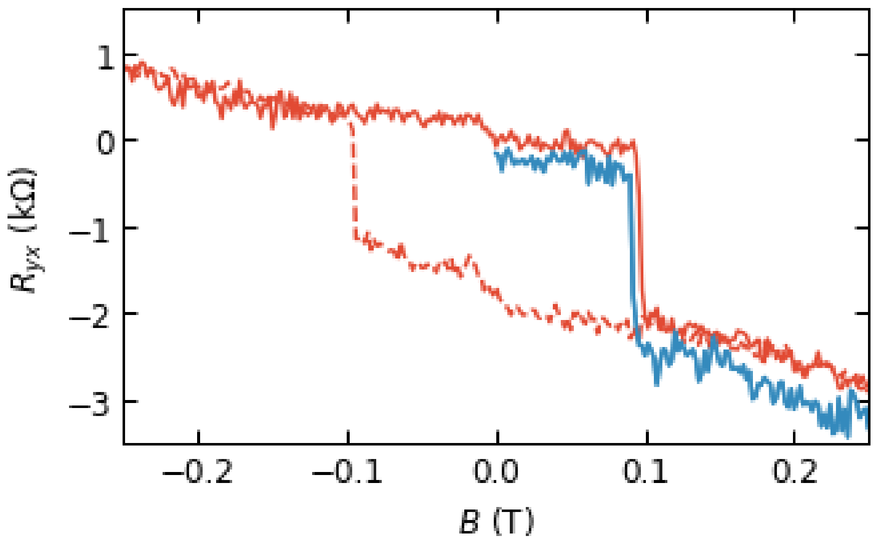


12.2 Fig. S6B

[24]:
```
sweep = figs6b_old

fig = plt.figure(figsize = (5,3))
plt.plot(sweep['prior_B'],sweep['prior_ryxld']/1000,color='C0',linestyle='--')
plt.plot(sweep['prior_B'],sweep['prior_ryxlu']/1000,color='C0')

plt.plot(sweep['post_B'],sweep['post_ryxl']/1000,color='C1')
plt.xlabel(r'$B\ (\mathrm{T})$')
plt.ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$')
```

```
plt.ylim(-3.5,1.5)
plt.xlim(-0.25,0.25);

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S6b.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



13 Figure S6 (updated, version 2)

Figure caption:

Temporal stability of the magnetization. (A) Hall resistance R_{yx} at $n/n_s = 0.746$ and $D/\epsilon_0 = -0.52$ V/nm as a function of time over the course of 6 hours in zero field, after first magnetizing the sample by applying -500 mT and then returning the field to 0 T. (B) A full hysteresis loop taken prior to the measurement shown in (A) is displayed in red. The blue trace shows the behavior of R_{yx} as the field is swept from 0 to 500 mT following the measurement in (A). A clear anomalous Hall jump in the blue trace is comparable to those in the continuous red loop, indicating that the magnetization was stable through the 6~hour pause.

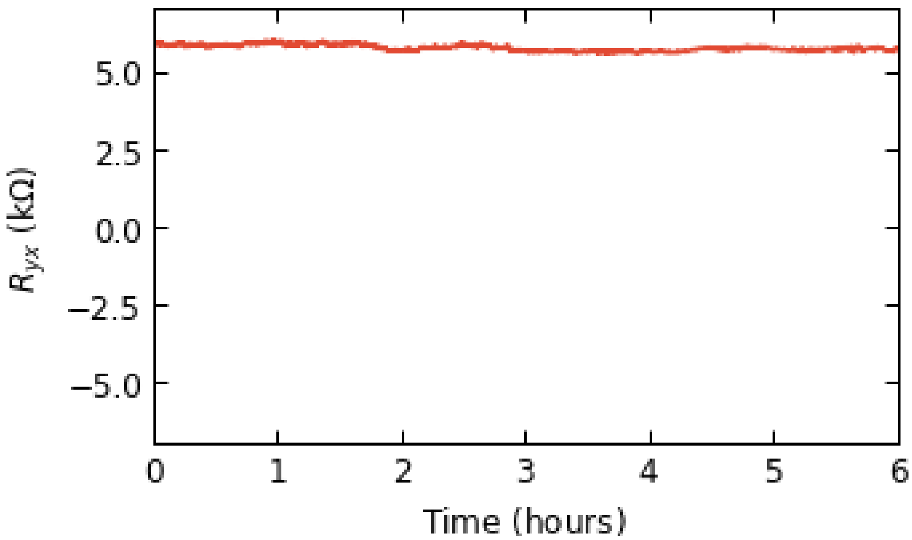
13.1 Fig. S6A

[25]:

```
sweep = figs6a

fig = plt.figure(figsize = (5,3))
plt.plot(sweep['time']/3600.,sweep['ryxl']/1000,color='C0')
plt.axis('tight')
```

```
plt.xlim(0,6)
plt.ylim(-7,7)
plt.xlabel(r'$\mathrm{Time}\ (\mathrm{hours})$')
plt.ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$');

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S6a.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

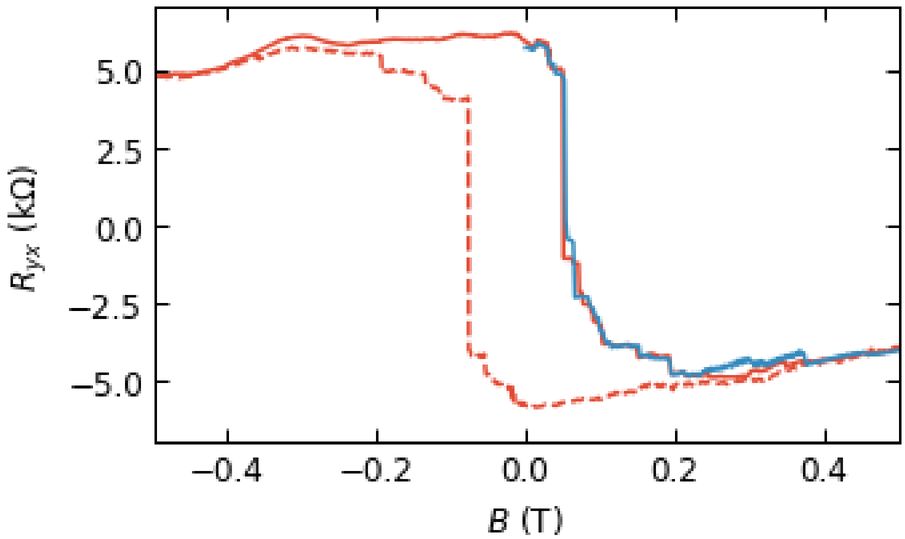


13.2 Fig. S6B

[26]:
```
sweep = figs6b

fig = plt.figure(figsize = (5,3))
plt.plot(sweep['prior_Bu'],sweep['prior_ryxlu']/1000,color='C0')
plt.plot(sweep['prior_Bd'],sweep['prior_ryxld']/1000,color='C0',linestyle='--')

plt.plot(sweep['post_B'],sweep['post_ryxl']/1000,color='C1')

plt.xlabel(r'$B\ (\mathrm{T})$')
plt.ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$')
plt.ylim(-7,7)
plt.xlim(-0.5,0.5);

# Uncomment to save a pdf of the figure:
```

```
# plt.savefig('fig_S6b.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



14 Figure S7

Figure caption:

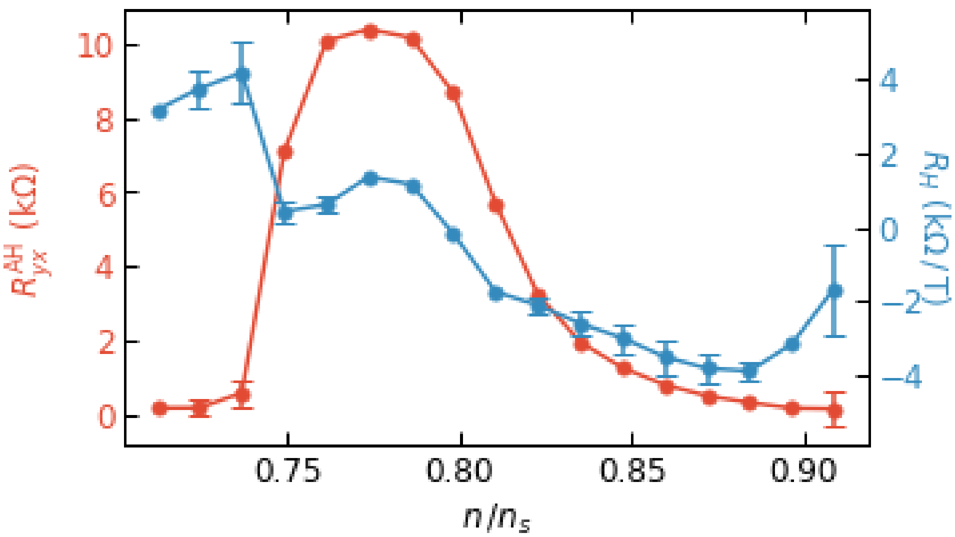
Density dependence near 3/4 with fixed displacement field at 2.1 K. (A) Zero-field anomalous Hall resistance R_{yx}^{AH} (red) and ordinary Hall slope R_{H} (blue) as a function of n/n_s while maintaining a constant displacement field $D/\epsilon_0 = -0.22$ V/nm. R_{yx}^{AH} is peaked at $n/n_s = 0.774$, close to the position of the peak at 0.758 in Fig. 2B of the main text and again coincident with a sign change in R_{H}. The full width at half maximum is slightly increased, at 0.07 instead of 0.04. (B) Magnetic field dependence of the longitudinal resistance R_{xx} (upper panel) and Hall resistance R_{yx} (lower panel) at $n/n_s = 0.774$, the largest hysteresis loop of the series shown in (A), with $R_{yx}^{\mathrm{AH}} = 10.4$ k .

14.1 Fig. S7A

[27]:
```
sweep = figs5cd_figs7ab

fig,ax = plt.subplots(figsize = (5,3))

###########################
# Plot anomalous Hall resistance on left axis

# Make error bar array, setting values to NaN for errors smaller than the␣
↪marker size
```

```python
errbarAH = np.array(sweep['R_AH_sigma'])/1000
for i in [0,3,4,5,6,7,8,9,10,11,12,13,14,15]:
    errbarAH[i]=np.nan

plt.errorbar(sweep['n'][:-1]/nsat,sweep['R_AH'][:-1]/1000,yerr=errbarAH[:-1],
             marker='o',markersize=5,capsize=4);
plt.xlabel(r'$n\,/n_s$')
plt.yticks(color='C0')

plt.ylabel(r'$R_{yx}^{\mathrm{AH}}\ (\mathrm{k\Omega})$',color='C0')

##########################
# Plot ordinary Hall coefficient on right axis

# Make error bar array, setting values to NaN for errors smaller than the␣
↪marker size
errbarH = np.array(sweep['R_H_sigma'])/1000
for i in [0,5,6,7,8,15]:
    errbarH[i]=np.nan

ax2t=plt.twinx()
plt.errorbar(sweep['n'][:-1]/nsat,sweep['R_H'][:-1]/1000,yerr=errbarH[:-1],
             marker='o',markersize=5,capsize=4,color='C1');
plt.axis('tight')
plt.ylim(-5.9,5.9)
plt.xlabel(r'$n\,/n_s$')
plt.yticks(color='C1')
plt.ylabel(r'$R_H\ (\mathrm{k\Omega\,/
↪\,T})$',color='C1',rotation=270,labelpad=10);


# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S7a.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

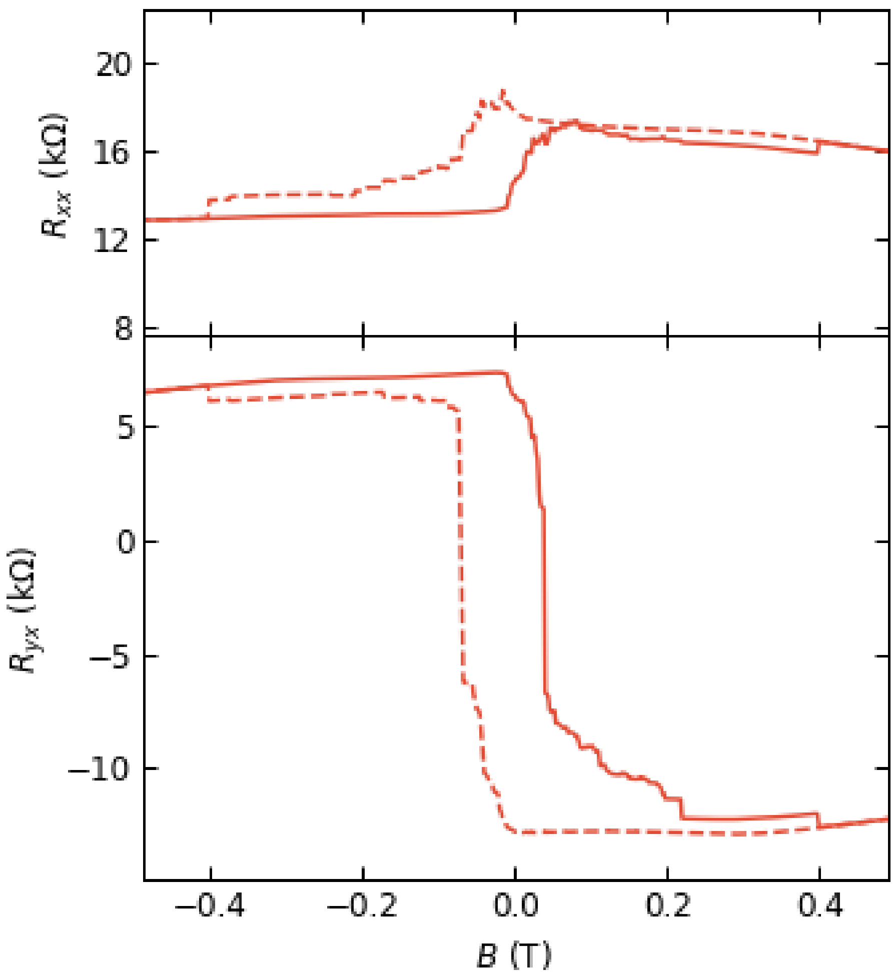


14.2 Fig. S7B

[28]:
```
sweep = figs5cd_figs7ab

fig = plt.figure(figsize = (5,6))
ax1 = fig.add_subplot(8,1,(1,3))
ax2 = fig.add_subplot(8,1,(4,8))
fig.subplots_adjust(hspace=0,wspace=0.4)

####################################################################
# upper panel -- Rxx

i = 4
ax1.plot(sweep['B'][:,i],sweep['down_rxxt'][:,i]/1000,color='C0',linestyle='--')
ax1.plot(sweep['B'][:,i],sweep['up_rxxt'][:,i]/
 ↪1000,color='C0',label=r'$R_{xx,\mathrm{top}}$')
ax1.set_xlim(-0.49,0.49)
ax1.set_ylabel(r'$R_{xx}\ (\mathrm{k\Omega})$')
ax1.set_ylim(15-24.8*3/10,15+24.8*3/10)
majorLocator = MultipleLocator(4)
ax1.yaxis.set_major_locator(majorLocator)


####################################################################
# lower pannel -- Ryx

ax2.plot(sweep['B'][:,i],sweep['down_ryxl'][:,i]/1000,color='C0',linestyle='--')
```

```
ax2.plot(sweep['B'][:,i],sweep['up_ryxl'][:,i]/1000,color='C0',label=r'$D=-1.
↪31\ \mathrm{V/nm}$')
ax2.set_xlim(-0.49,0.49)
ax2.set_ylim(-14.9,8.9)

ax2.set_xlabel(r'$B\ (\mathrm{T})$')
ax2.set_ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$');


# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S7b.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



15 Figure S8

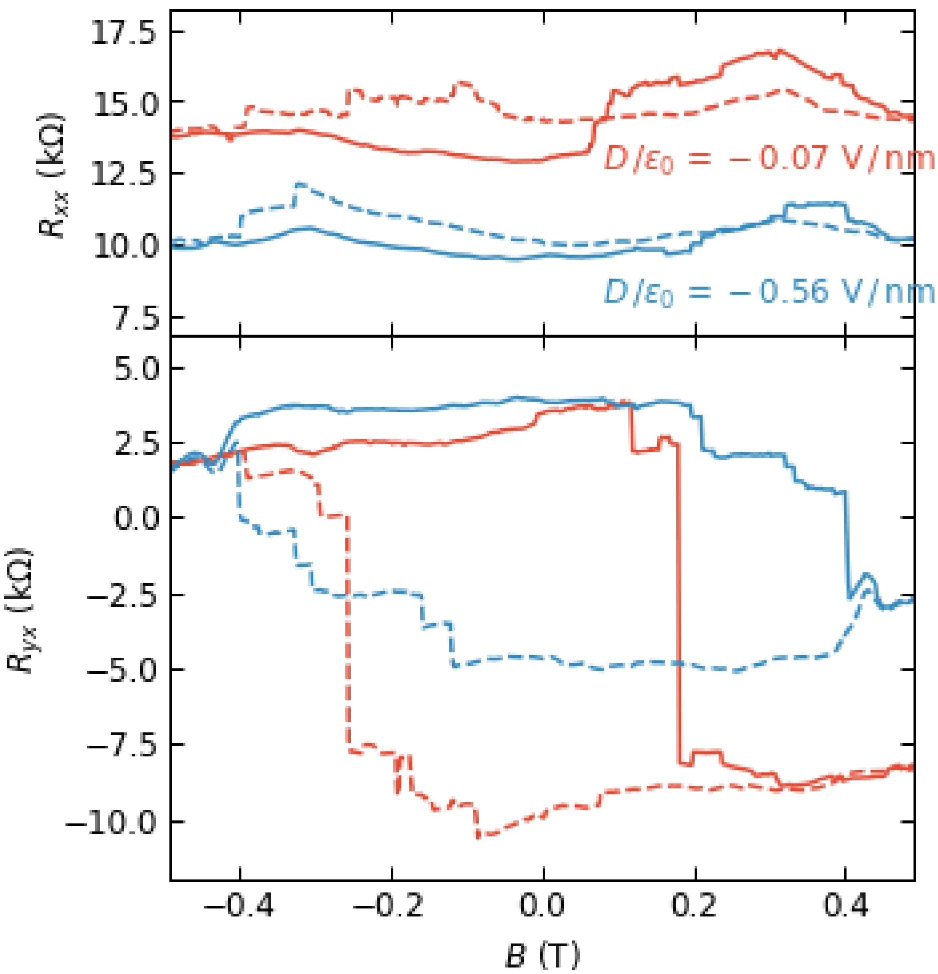
Figure caption:

Displacement field dependence of hysteresis loops. Longitudinal resistance R_{xx} (upper panel) and Hall resistance R_{yx} (lower panel) at $n/n_s = 0.749$ for two different displacement fields as labeled in the figure. Although tuning the displacement field from a large negative field to near zero causes a slight change in the longitudinal resistance and the hysteresis loop structure, the TBG magnetic field dependence remains hysteretic.

[29]:
```
sweep = figs8

fig = plt.figure(figsize = (5,6))
ax1 = fig.add_subplot(8,1,(1,3))
ax2 = fig.add_subplot(8,1,(4,8))
fig.subplots_adjust(hspace=0,wspace=0.4)

##################################################################
# upper panel -- Rxx

# D/\epsilon_0 = -0.07 V/nm
ax1.plot(sweep['D1_d_B'],sweep['D1_d_rxxt']/1000,color='C0',linestyle='--')
ax1.plot(sweep['D1_u_B'],sweep['D1_u_rxxt']/1000,color='C0')

# D/\epsilon_0 = -0.56 V/nm
ax1.plot(sweep['D2_d_B'],sweep['D2_d_rxxt']/1000,color='C1',linestyle='--')
ax1.plot(sweep['D2_u_B'],sweep['D2_u_rxxt']/1000,color='C1')
ax1.set_xlim(-0.49,0.49)
ax1.set_ylim(12.5-19*3/10,12.5+19*3/10)

ax1.set_ylabel(r'$R_{xx}\ (\mathrm{k\Omega})$')

txt1 = ax1.text(0.08,12.5,r'$D\,/\epsilon_0\,=-0.07\ \mathrm{V\,/
↪\,nm}$',color='C0')
txt2 = ax1.text(0.08,7.9,r'$D\,/\epsilon_0\,=-0.56\ \mathrm{V\,/
↪\,nm}$',color='C1')


##################################################################
# lower panel -- Ryx

# D/\epsilon_0 = -0.07 V/nm
ax2.plot(sweep['D1_d_B'],sweep['D1_d_ryxl']/1000,color='C0',linestyle='--')
ax2.plot(sweep['D1_u_B'],sweep['D1_u_ryxl']/1000,color='C0')

# D/\epsilon_0 = -0.56 V/nm
ax2.plot(sweep['D2_d_B'],sweep['D2_d_ryxl']/1000,color='C1',linestyle='--')
ax2.plot(sweep['D2_u_B'],sweep['D2_u_ryxl']/1000,color='C1')
```

```
ax2.set_xlim(-0.49,0.49)
ax2.set_ylim(-12,6) # 24.8

ax2.set_xlabel(r'$B\ (\mathrm{T})$')
ax2.set_ylabel(r'$R_{yx}\ (\mathrm{k\Omega})$');

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S8.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



16 Figure S9

Figure caption:

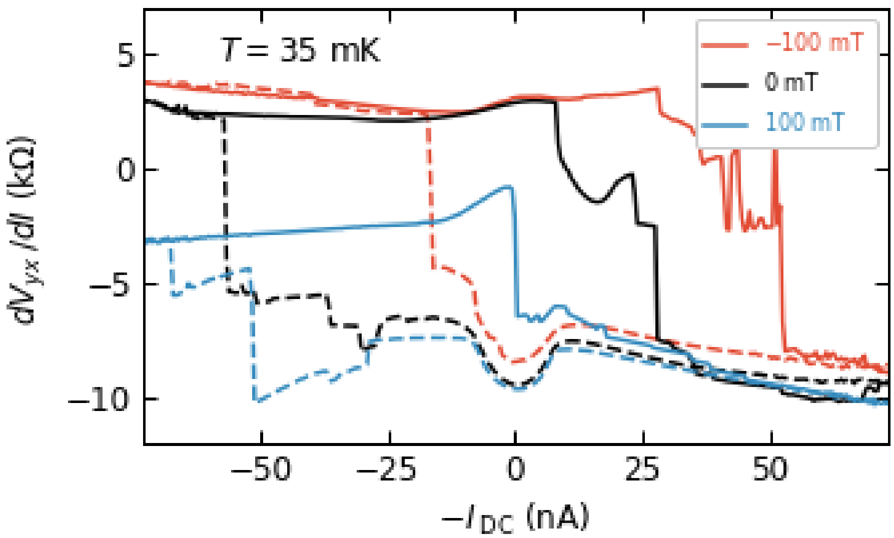
Current-driven switching in nonzero magnetic field, and characterization of the transition. (A) Hysteresis loops of the differential Hall resistance dV_{yx}/dI with respect to DC current (plotted as $-I_{\mathrm{DC}}$ as in Fig.~4 of the main text) at three different static magnetic fields after the sample was magnetized at 500 mT. These data were taken at 35 mK with $n/n_s = 0.749$ and $D/\epsilon_0 = -0.22$ V/nm during the same cooldown as for the data of Fig.~4. (B) Transition rate of the apparent magnetization switching at a fixed current I_{DC} after magnetizing the sample with a -75 nA current (at $T = 2.1$ K and zero field). The transition appears to be a memoryless process.

16.1 Fig. S9A

[30]:
```
sweep = figs9a

fields = ['m100mt','0mt','100mt']
sweepdirs = ['_d2','_u2']

pltcolors = ['C0','C0','k','k','C1','C1']
pltlinestyle = ['-','--','-','--','-','--']
labels = [r'$-100\ \mathrm{mT}$','',r'$0\ \mathrm{mT}$','',r'$100\␣
↪\mathrm{mT}$','']



fig,ax = plt.subplots(figsize = (5,3))
i = 0
for field in fields:
    for sweepdir in sweepdirs:
        plt.
↪plot(-1E9*sweep[field+sweepdir+'_Idc'],sweep[field+sweepdir+'_ryxl']/1000,
                  color=pltcolors[i],linestyle=pltlinestyle[i],label=labels[i])
        i = i+1

plt.axis('tight')
plt.xlim(-73,73)
plt.ylim(-12,7)
plt.xlabel(r'$-I\,_{\mathrm{DC}}\ (\mathrm{nA})$')
plt.ylabel(r'$dV_{yx}\,/dI\ (\mathrm{k\Omega})$')

leg = ax.legend(fontsize=8,framealpha=1,loc = 'upper right',markerscale=0)

colind=0
for text in leg.get_texts():
    text.set_color(pltcolors[colind])
    colind=colind+2
```

```
txt = ax.text(0.1,0.88,r'$T = 35\␣
↪\mathrm{mK}$',horizontalalignment='left',transform=ax.transAxes)

# Uncomment to save a pdf of the figure:
# plt.savefig('figs_dcfield.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```

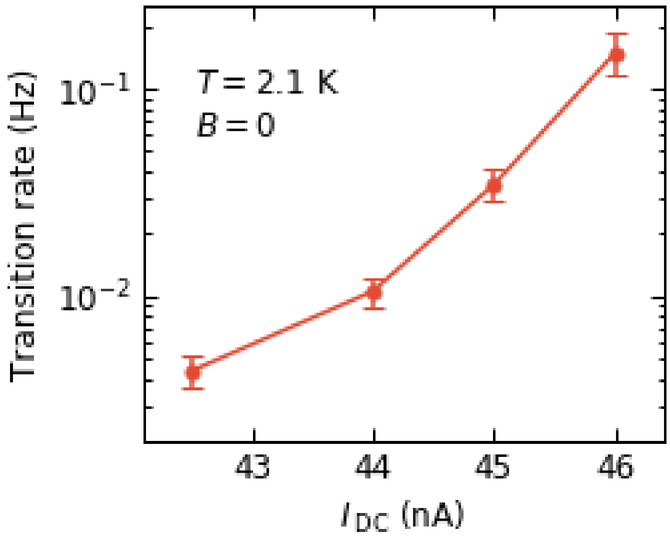


16.2 Fig. S9B

[31]:
```
sweep = figs9b

fig,ax = plt.subplots(figsize = (3.5,3))

ax.set_yscale("log")
plt.errorbar(1E9*sweep['tr_curr'],sweep['tr_rate'],yerr=sweep['tr_err'],
             marker='o',markersize=5,capsize=4);
plt.xlabel(r'$I\,_{\mathrm{DC}}\ (\mathrm{nA})$')
plt.ylabel('Transition rate (Hz)')
plt.xlim(42.1,46.4)
plt.ylim(2e-3,2.5e-1)
txt = ax.text(0.1,0.8,r'$T = 2.1\␣
↪\mathrm{K}$',horizontalalignment='left',transform=ax.transAxes)
txt1 = ax.text(0.1,0.7,r'$B = 0$',horizontalalignment='left',transform=ax.
↪transAxes)
```

```
# Uncomment to save a pdf of the figure:
# plt.savefig('figs_dctrans.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



17 Figure S10

Figure caption:

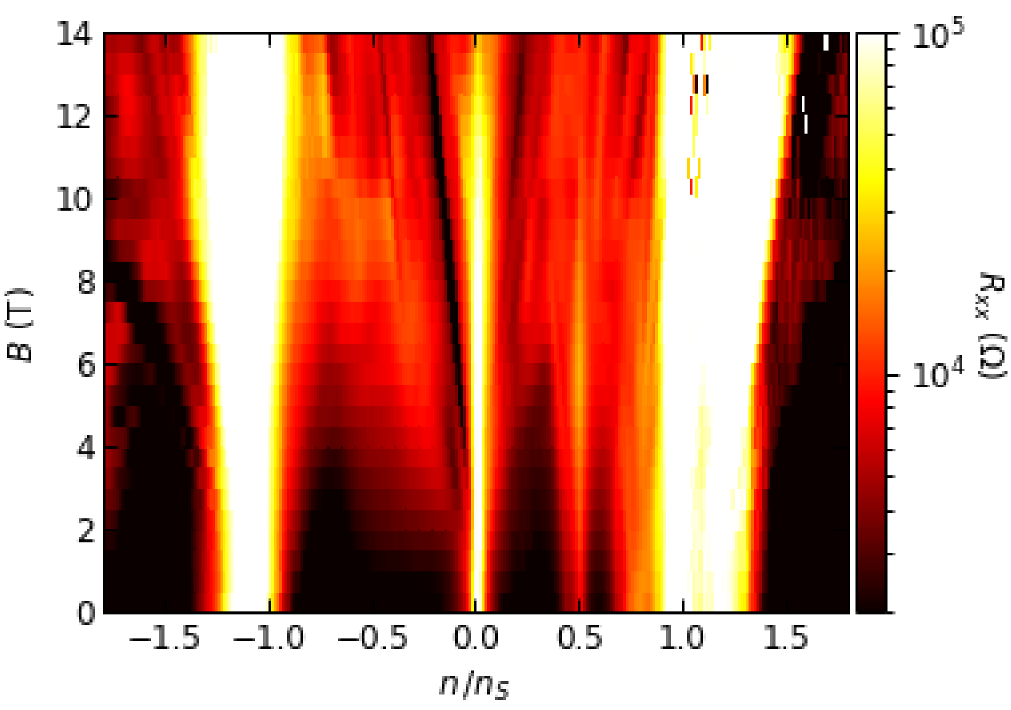
Quantum oscillations of TBG at fixed displacement field. (A) Landau fan diagram of the longitudinal resistance R_{xx} taken at 2.1 K for a fixed displacement field $D/\epsilon_0 = 0$ V/nm. Emerging from the CNP, we observe the Landau levels $\nu = \pm 2, \pm 4$. We further observe Landau levels from $n/n_s = 1/2$ of $\nu = 2, 4$, from $n/n_s = 3/4$ of $\nu = 1, 4$, and the sequence from $n/n_s = -1$ of $\nu = -8, -12, -16, -20$.
(B) Schematic of the Landau levels observed in (A).

17.1 Fig. S10A

[32]:
```
sweep = figs10a

fig, ax = plt.subplots(figsize = (5.9525,4))
fig = plt.pcolormesh(sweep['n']/nsat,sweep['B'],sweep['rxxt'],
                     norm=LogNorm(),linewidth=0,rasterized=True);

cbar = plt.colorbar(pad = 0.01)
cbar.ax.get_yaxis().labelpad = 15
```

```
cbar.ax.tick_params(which = 'major',direction='out')
cbar.ax.tick_params(which = 'minor',direction='out')

plt.clim(2E3,1E5);
cbar.set_ticks([1E4,1E5])
minortick = np.concatenate((np.linspace(2E3,9E3,8),np.linspace(2E4,9E4,8)))
minorticklocs = (np.log(minortick)-np.log(2E3))/(np.log(1E5)-np.log(2E3))
cbar.ax.yaxis.set_ticks(minorticklocs, minor=True)
cbar.set_label(r'$R_{xx}\ (\mathrm{\Omega})$', rotation=270)
cbar.ax.tick_params(which = 'major',direction='out')


plt.ylabel(r'$B\ (\mathrm{T})$')
plt.xlabel(r'$n\,/n_S$')
plt.xlim(-1.8,1.8)
plt.ylim(0,14)
fig.set_cmap('hot')

majorLocator = MultipleLocator(0.5)
ax.xaxis.set_major_locator(majorLocator)

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S10a.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



17.2 Fig. S10B

[33]:
```
fig, ax = plt.subplots(figsize = (5.,4.))
nx = np.linspace(0,10,1000)

# CNP
col = 'C0'
offset = 0.0
nx = np.linspace(0,0.25*nsat,1000)
for c in [2,4]:
    plt.plot((nx+offset)/nsat,phib*nx*1E16/c,color=col)
    plt.plot((nx+offset)/nsat,-phib*nx*1E16/c,color=col)
    plt.plot((-nx+offset)/nsat,phib*nx*1E16/c,color=col)
    plt.plot((-nx+offset)/nsat,-phib*nx*1E16/c,color=col)

# -1*n_s

col = 'C3'
offset = -nsat
nx = np.linspace(0,0.5*nsat,1000)
for c in [8,12,16,20]:
    plt.plot((-nx+offset)/nsat,phib*nx*1E16/c,color=col)
    plt.plot((-nx+offset)/nsat,-phib*nx*1E16/c,color=col)

# 1/2*n_s
offset = 0.5*nsat
nx = np.linspace(0,0.25*nsat,1000)
col = 'C1'
for c in [2,4]:
    plt.plot((nx+offset)/nsat,phib*nx*1E16/c,color=col)
    plt.plot((nx+offset)/nsat,-phib*nx*1E16/c,color=col)

# 3/4*n_s
offset = 0.75*nsat
nx = np.linspace(0,0.25*nsat,1000)
col = 'C2'

for c in [1,4]:
    plt.plot((nx+offset)/nsat,phib*nx*1E16/c,color=col)
    plt.plot((nx+offset)/nsat,-phib*nx*1E16/c,color=col)


plt.xlim(-1.8,1.8)
plt.ylim(0,14)
```

```
plt.ylabel(r'$B\ (\mathrm{T})$')
plt.xlabel(r'$n\,/n_S$')

majorLocator = MultipleLocator(0.5)
ax.xaxis.set_major_locator(majorLocator)

# Uncomment to save a pdf of the figure:
# plt.savefig('fig_S9b.pdf',bbox_inches='tight')
```

```
C:\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1328: UserWarning:
findfont: Font family ['Helvetica'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```