# deep-tropical_contraction_main

September 3, 2019

In [1]:
```
from scipy.io import netcdf, loadmat
import numpy as np
import os
from scipy import interpolate
import glob
from netCDF4 import Dataset
from scipy import interp
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
from numpy import polyfit
import numpy.matlib
from matplotlib.patches import Rectangle
from itcz4 import itcz
from scipy.stats.stats import pearsonr
from scipy.stats import linregress
%matplotlib inline
import copy
# from matplotlib import rcParams
# rcParams['font.family'] = 'sans-serif'
# rcParams['font.sans-serif'] = ['Arial']
import cmaps
import matplotlib
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from matplotlib.colors import ListedColormap
plt.rcParams["font.family"] = "Times New Roman"
#plt.rcParams["font.weight"] = "bold"
#plt.rc('text', usetex=True)
plt.rc('font', family='serif')
plt.rcParams['xtick.major.pad'] = 10
plt.rcParams['ytick.major.pad'] = 10
```

In [2]:
```
##### get land mask
f = netcdf.netcdf_file('/Volumes/Lacie/Downloads/orog_fx_CCSM4_amip_r0i0p0.regrid.nc',
orog = f.variables['orog'][::]
mask=copy.deepcopy(orog)
```

```
mask[orog>0]=np.nan
mask[orog==0]=1

f = netcdf.netcdf_file('/Volumes/Lacie/Downloads/orog_fx_CCSM4_amip_r0i0p0.regrid.hdeg
orog = f.variables['orog'][::]
maskhi=copy.deepcopy(orog)
maskhi[orog>0]=np.nan
maskhi[orog==0]=1


def cmap(name):
    colors ={"BlueDarkRed18":[[36.0,0.0,216.0],[24.0,28.0,247.0],[40.0,87.0,255.0],[61
"BlueDarkRed15":[[36.0,0.0,216.0],[24.0,28.0,247.0],[40.0,87.0,255.0],[61.0,135.0,255.
"BlueDarkRed13":[[36.0,0.0,216.0],[24.0,28.0,247.0],[40.0,87.0,255.0],[61.0,135.0,255.
"BlueDarkRed19":[[61.0,135.0,255.0],[86.0,176.0,255.0],[117.0,211.0,255.0],[153.0,234.
"BlueDarkRed19_r":[[255.0,61.0,61.0],[255.0,120.0,86.0],[255.0,172.0,117.0],[255.0,214
"BlueDarkRed9":[[117.0,211.0,255.0],[153.0,234.0,255.0],[188.0,249.0,255.0],[255.0,255
    data = np.array(colors[name])
    data = data / np.max(data)
    cmap = ListedColormap(data, name=name)
    return cmap
cmap_rdbu=cmap('BlueDarkRed13')
cmap_rdbu_r = ListedColormap(cmap_rdbu.colors[::-1])
cmap_jet=plt.cm.get_cmap('jet')
cmap_jet.set_under('w')

def plt_mpsi(var,lat,plev,mon):
    plt.contourf(lat,plev/100.,var[mon,::]/(10**10),[-5,-3,-0.5,-0.2,0.2,0.5,3,5],cmap=
    plt.colorbar()
    plt.gca().invert_yaxis()
    plt.xlim(-40,40)
    plt.xticks(np.arange(-50,55,5))
    plt.grid()

#first average and then compute boundary
def get_hadley_boundary_(var,lat):
    lat_n_hi = np.arange(10,44.2,0.2)
    lat_s_hi = np.arange(-10,-44.,-0.2)
    inds=len(lat)//2
    lat_n = lat[inds:]
    lat_s = lat[inds::-1]
    lat_nb=np.zeros(12)
    lat_sb=np.zeros(12)
    for mon in range(12):
        var_n = np.squeeze(np.mean(var[mon,10:13,inds:],0))
        var_s = np.squeeze(np.mean(var[mon,10:13,inds::-1],0))
        f = interpolate.interp1d(lat_n,var_n,'linear')
        var_n_hi=f(lat_n_hi)
```

```
        f = interpolate.interp1d(lat_s,var_s,'linear')
        var_s_hi=f(lat_s_hi)
        lat_nb[mon] = get_sign_change_n(var_n_hi,lat_n_hi)
        lat_sb[mon] = get_sign_change_p(var_s_hi,lat_s_hi)
    return lat_sb,lat_nb
#compute hadley boundary at each level between 400 hPa and 600hPa and then avergae
def get_hadley_boundary(var,lat):
    lat_n_hi = np.arange(10,44.2,0.2)
    lat_s_hi = np.arange(-10,-44.,-0.2)
    inds=len(lat)//2
    lat_n = lat[inds:]
    lat_s = lat[inds::-1]
    lat_nb=np.zeros(12)
    lat_sb=np.zeros(12)
    for mon in range(12):
        tmp1 = []
        tmp2 = []
        for level in [10,11,12]:
            var_n = var[mon,level,inds:]
            var_s = var[mon,level,inds::-1]
            f = interpolate.interp1d(lat_n,var_n,'linear')
            var_n_hi=f(lat_n_hi)
            tmp1.append(get_sign_change_n(var_n_hi,lat_n_hi))
            f = interpolate.interp1d(lat_s,var_s,'linear')
            var_s_hi=f(lat_s_hi)
            tmp2.append(get_sign_change_p(var_s_hi,lat_s_hi))
        lat_nb[mon] = np.mean(tmp1)
        lat_sb[mon] = np.mean(tmp2)
    return lat_sb,lat_nb


def get_hadley_boundary1(var,lat):
    lat_n_hi = np.arange(10,44.2,0.2)
    lat_s_hi = np.arange(-10,-44.,-0.2)
    inds=len(lat)//2
    lat_n = lat[inds:]
    lat_s = lat[inds::-1]
    lat_nb=np.zeros(12)
    lat_sb=np.zeros(12)
    for mon in range(12):
        var_n = np.squeeze(np.mean(var[mon,17:20,inds:],0))
        var_s = np.squeeze(np.mean(var[mon,17:20,inds::-1],0))
        f = interpolate.interp1d(lat_n,var_n,'linear')
        var_n_hi=f(lat_n_hi)
        f = interpolate.interp1d(lat_s,var_s,'linear')
        var_s_hi=f(lat_s_hi)
        lat_nb[mon] = get_sign_change_n(var_n_hi,lat_n_hi)
        lat_sb[mon] = get_sign_change_p(var_s_hi,lat_s_hi)
```

```
    return lat_sb,lat_nb

def get_annual_hadley_boundary(var,lat):
    lat_n_hi = np.arange(15,40.2,0.2)
    lat_s_hi = np.arange(-15,-40.,-0.2)
    inds=len(lat)//2
    lat_n = lat[inds:]
    lat_s = lat[inds::-1]
    var_n = np.squeeze(np.mean(var[10:13,inds:],0))
    var_s = np.squeeze(np.mean(var[10:13,inds::-1],0))
    f = interpolate.interp1d(lat_n,var_n,'cubic')
    var_n_hi=f(lat_n_hi)
    f = interpolate.interp1d(lat_s,var_s,'cubic')
    var_s_hi=f(lat_s_hi)
    lat_nb = get_sign_change_n(var_n_hi,lat_n_hi)
    lat_sb = get_sign_change_p(var_s_hi,lat_s_hi)
    return lat_sb,lat_nb

def get_pe_boundary(var,lat):
    lat_n_hi = np.arange(20,48.2,0.2)
    lat_s_hi = np.arange(-20,-48.,-0.2)
    inds=len(lat)//2
    lat_n = lat[inds:]
    lat_s = lat[inds::-1]
    lat_nb=np.zeros(12)
    lat_sb=np.zeros(12)
    for mon in range(12):
        var_n = np.squeeze(var[mon,inds:])
        var_s = np.squeeze(var[mon,inds::-1])
        f = interpolate.interp1d(lat_n,running_mean(var_n,5),'linear')
        var_n_hi=f(lat_n_hi)
        f = interpolate.interp1d(lat_s,running_mean(var_s,5),'linear')
        var_s_hi=f(lat_s_hi)
        lat_nb[mon] = get_sign_change_p(var_n_hi,lat_n_hi)
        lat_sb[mon] = get_sign_change_p(var_s_hi,lat_s_hi)
    return lat_sb,lat_nb

def get_sign_change_n(var,lat):
    for i in range(len(var)-1):
        if var[i]>=0 and var[i+1]<=0:
            return lat[i]
    return 40.

def get_sign_change_p(var,lat):
    for i in range(len(var)-1):
        if var[i]<=0 and var[i+1]>=0:
            return lat[i]
    return -38.
```

```
def running_mean(var,window):
    var_=np.zeros(len(var))
    for i in range(len(var)):
        if i<=window//2 or i>=len(var)-window//2-2:
            var_[i]=var[i]
        else:
            var_[i]=np.mean(var[i-window//2:i+window//2+1])
    return var_

def get_max_lat(var,lat):
    lat_hi= np.arange(20,50,0.1)
    time_hi = np.arange(0,12.1,0.1)
    time = np.arange(-0.5,13.5,1)
    var_=np.zeros((14,np.size(var,1)))
    var_[0,:]=var[-1,:]
    var_[-1,:]=var[0,:]
    var_[1:-1,:]=var
    f=interpolate.interp2d(lat,time,var_,'cubic')
    var_tmp=f(lat_hi,time_hi)
    lat_max = lat_hi[np.argmax(var_tmp,1)]
    return time_hi, lat_max

def get_extend(var):
    var_=np.zeros((14,np.size(var,1)))
    var_[0,:]=var[-1,:]
    var_[-1,:]=var[0,:]
    var_[1:-1,:]=var
    return var_

def get_min_loc(var):
    latf = np.arange(-10,15.2,0.2)
    month = np.arange(1,13,1)
    f=interpolate.interp2d(lat[51:71], month, var[:,51:71], kind='cubic')
    varf = f(latf,month)
    var_tomay = varf[:5,:50]
    lat_sh = latf[:50]
    var_frommay = varf[4:,50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmin(var_tomay,1)]
    lat2=lat_nh[np.argmin(var_frommay,1)]
    return lat1,lat2

def get_min_loc2(var):
    latf = np.arange(-10,15.2,0.2)
    month = np.arange(1,13,1)
    f=interpolate.interp2d(lat_hi[140:221], month, var[:,140:221], kind='cubic')
    varf = f(latf,month)
```

```
    var_tomay = varf[:5,:50]
    lat_sh = latf[:50]
    var_frommay = varf[4:,50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmin(var_tomay,1)]
    lat2=lat_nh[np.argmin(var_frommay,1)]
    return lat1,lat2

def get_min_loc_season(var):
    var_jja = np.mean(var[5:8,:],0)
    var_djf = (var[0,:]+var[1,:]+var[11,:])/3.
    latn_hi = np.arange(5,15.2,0.2)
    lats_hi = np.arange(-15,-5,0.2)
    f=interpolate.interp1d(lat, var_jja, kind='cubic')
    var_jja = f(latn_hi)
    itcz_n = latn_hi[np.argmin(var_jja)]
    f=interpolate.interp1d(lat, var_djf, kind='cubic')
    var_djf = f(lats_hi)
    itcz_s = latn_hi[np.argmin(var_djf)]
    return itcz_s,itcz_n


def get_min_loc1(var):
    latf = np.arange(-10,15.2,0.2)
    month = np.arange(1,13,1)
    f=interpolate.interp2d(lat[51:71], month, var[:,51:71], kind='cubic')
    varf = f(latf,month)
    var_tomay = varf[:4,:50]
    lat_sh = latf[:50]
    var_frommay = varf[4:,50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmin(var_tomay,1)]
    lat2=lat_nh[np.argmin(var_frommay,1)]
    return lat1,lat2

def get_max_loc(var):
    latf = np.arange(-10,15.2,0.2)
    month = np.arange(1,13,1)
    f=interpolate.interp2d(lat[51:71], month, var[:,51:71], kind='cubic')
    varf = f(latf,month)
    var_tomay = varf[:5,:50]
    lat_sh = latf[:50]
    var_frommay = varf[4:,50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmax(var_tomay,1)]
    lat2=lat_nh[np.argmax(var_frommay,1)]
    return lat1,lat2
```

```
def get_array(start,end,lev):
    a=np.arange(start,end,lev)

    return list(a[:len(a)//2])+list(a[len(a)//2+1:])

def get_annual_range(var):

    latf = np.arange(-10,15.2,0.2)
    f=interpolate.interp1d(lat[51:71], var[51:71], kind='cubic')
    varf = f(latf)
    var_sh = varf[:40]
    lat_sh = latf[:40]
    var_nh = varf[50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmin(var_sh)]
    lat2=lat_nh[np.argmin(var_nh)]
    return lat2-lat1

def get_annual_max(var):
    latf = np.arange(-10,15.2,0.2)
    f=interpolate.interp1d(lat[51:71], var[51:71], kind='cubic')
    varf = f(latf)
    var_sh = varf[:40]
    lat_sh = latf[:40]
    var_nh = varf[50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmin(var_sh)]
    lat2=lat_nh[np.argmin(var_nh)]
    return lat1, lat2

def get_annual_max2(var):
    latf = np.arange(-10,15.2,0.2)
    f=interpolate.interp1d(lat_hi[140:221], var[140:221], kind='cubic')
    varf = f(latf)
    var_sh = varf[:40]
    lat_sh = latf[:40]
    var_nh = varf[50:]
    lat_nh = latf[50:]
    lat1=lat_sh[np.argmin(var_sh)]
    lat2=lat_nh[np.argmin(var_nh)]
    return lat1, lat2

def get_gcm(models,varname,ind=[5,6]):

    ind1=ind[0];ind2=ind[1]

    if varname == 'wap':
        var_hist =0
```

```
        var_rcp =0
        for name in models:
            os.chdir('/Volumes/Zhou/data/cmip5/historical/wap/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('wap_'+name+'.month.regrid.nc', 'r')
            var_hist_ = np.squeeze(np.mean(f.variables['wap'][ind1:ind2,5,:,:]*864.,0))
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('wap_Amon_'+name+'_rcp85_r1i1p1_208001-210012.month
            var_rcp_ = np.squeeze(np.mean(f.variables['wap'][ind1:ind2,5,:,:]*864.,0))
            var_hist = var_hist + var_hist_/len(models)
            var_rcp = var_rcp + var_rcp_/len(models)

    if varname == 'ua':
        var_hist =0
        var_rcp =0
        i=0
        for name in models:
            os.chdir('/Volumes/Zhou/data/cmip5/historical/ua/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('ua_'+name+'.month.regrid.nc', 'r')
            var_hist_ = np.squeeze(np.mean(np.mean(f.variables['ua'][ind1:ind2,8:9,:,:]
            var_hist = var_hist + var_hist_/len(models)
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ua/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('ua_Amon_'+name+'_rcp85_r1i1p1_208001-210012.month.
            var_rcp_ = np.squeeze(np.mean(np.mean(f.variables['ua'][ind1:ind2,8:9,:,:]
            var_rcp = var_rcp + var_rcp_/len(models)
    if varname == 'pr':
        var_hist =0
        var_rcp =0
        for name in models:
            os.chdir('/Volumes/Zhou/data/cmip5/historical/pr/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('pr_'+name+'.month.regrid.nc', 'r')
            var_hist_ = np.squeeze(np.mean(f.variables['pr'][ind1:ind2,:,:],0))
            var_hist = var_hist + var_hist_/len(models)
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/pr/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('pr_Amon_'+name+'_rcp85_r1i1p1_208001-210012.month.
            var_rcp_ = np.squeeze(np.mean(f.variables['pr'][ind1:ind2,:,:],0))
            var_rcp = var_rcp + var_rcp_/len(models)
    return var_hist,var_rcp



def get_gcm1(models,varname,ind=[5,6]):

    ind1=ind[0];ind2=ind[1]

    if varname == 'wap':
        var_hist =0
        var_rcp =0
        var_hist__ = np.zeros((len(models),121,144))
```

```
            var_rcp__ = np.zeros((len(models),121,144))
            i=0
            for name in models:
                os.chdir('/Volumes/Zhou/data/cmip5/historical/wap/'+name+'/r1i1p1/')
                f = netcdf.netcdf_file('wap_'+name+'.month.regrid.nc', 'r')
                var_hist_ = np.squeeze(np.mean(f.variables['wap'][ind1:ind2,5,:,:]*864.,0))
                os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap/'+name+'/r1i1p1/')
                f = netcdf.netcdf_file('wap_Amon_'+name+'_rcp85_r1i1p1_208001-210012.month
                var_rcp_ = np.squeeze(np.mean(f.variables['wap'][ind1:ind2,5,:,:]*864.,0))
                var_hist = var_hist + var_hist_/len(models)
                var_rcp = var_rcp + var_rcp_/len(models)
                var_rcp__[i,::] = var_rcp_
                var_hist__[i,::] = var_hist_
                i+=1

        if varname == 'ua':
            var_hist =0
            var_rcp =0
            var_hist__ = np.zeros((len(models),121,144))
            var_rcp__ = np.zeros((len(models),121,144))
            i=0
            for name in models:
                os.chdir('/Volumes/Zhou/data/cmip5/historical/ua/'+name+'/r1i1p1/')
                f = netcdf.netcdf_file('ua_'+name+'.month.regrid.nc', 'r')
                var_hist_ = np.squeeze(np.mean(np.mean(f.variables['ua'][ind1:ind2,8:9,:,:]
                var_hist = var_hist + var_hist_/len(models)
                var_hist__[i,::] = var_hist_
                os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ua/'+name+'/r1i1p1/')
                f = netcdf.netcdf_file('ua_Amon_'+name+'_rcp85_r1i1p1_208001-210012.month.
                var_rcp_ = np.squeeze(np.mean(np.mean(f.variables['ua'][ind1:ind2,8:9,:,:]
                var_rcp__[i,::] = var_rcp_
                var_rcp = var_rcp + var_rcp_/len(models)
                i+=1

        return var_hist,var_rcp,var_hist__,var_rcp__

    def get_gcm_amip(models,varname,ind=[5,6],exp='4k'):

        ind1=ind[0];ind2=ind[1]

        if varname == 'wap':
            var_hist =0
            var_rcp =0
            for name in models:
                os.chdir('/Volumes/Zhou/data/cmip5/amip/wap/'+name+'/r1i1p1/')
                f = netcdf.netcdf_file('wap_'+name+'.month.regrid.nc', 'r')
                var_hist_ = np.squeeze(np.mean(f.variables['wap'][ind1:ind2,5,:,:]*864.,0))
                os.chdir('/Volumes/Zhou/data/cmip5/amip'+exp+'/wap/'+name+'/r1i1p1/')
```

```
            f = netcdf.netcdf_file('wap_'+name+'.month.regrid.nc', 'r')
            var_rcp_ = np.squeeze(np.mean(f.variables['wap'][ind1:ind2,5,:,:]*864.,0))
            var_hist = var_hist + var_hist_/len(models)
            var_rcp = var_rcp + var_rcp_/len(models)

    if varname == 'ua':
        var_hist =0
        var_rcp =0
        for name in models:
            os.chdir('/Volumes/Zhou/data/cmip5/amip/ua/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('ua_'+name+'.month.regrid.nc', 'r')
            var_hist_ = np.squeeze(np.mean(np.mean(f.variables['ua'][ind1:ind2,8:9,:,:]
            var_hist = var_hist + var_hist_/len(models)
            os.chdir('/Volumes/Zhou/data/cmip5/amip'+exp+'/ua/'+name+'/r1i1p1/')
            f = netcdf.netcdf_file('ua_'+name+'.month.regrid.nc', 'r')
            var_rcp_ = np.squeeze(np.mean(np.mean(f.variables['ua'][ind1:ind2,8:9,:,:]
            var_rcp = var_rcp + var_rcp_/len(models)
    return var_hist,var_rcp


def get_latmax_tp(var0,lat0):
    var = var0[184:210,:]
    lat = lat0[184:210]
    latmax_nh = [np.nan]*144
    for i in range(144):
        latmax_nh[i] = lat[np.argmin(var[:,i])]

    var = var0[150:176,:]
    lat = lat0[150:176]
    latmax_sh = [np.nan]*144
    for i in range(144):
        latmax_sh[i] = lat[np.argmin(var[:,i])]
    return latmax_nh,latmax_sh

def get_latwt_tp(var0,lat0):
    var = var0[181:206,:]
    var[var>0] = 0.
    lat = lat0[181:206]
    latmax_nh = [np.nan]*np.size(var0,1)
    for i in range(np.size(var0,1)):
        latmax_nh[i] = np.sum(var[:,i]*lat)/np.sum(var[:,i])
    var = var0[150:180,:]
    var[var>0] = 0.
    lat = lat0[150:180]
    latmax_sh = [np.nan]*np.size(var0,1)
    for i in range(np.size(var0,1)):
        latmax_sh[i] = np.sum(var[:,i]*lat)/np.sum(var[:,i])
    return latmax_nh,latmax_sh
```

```
def plot_scatter(models_hist,var1,var2):
    for i in range(len(models_hist)):
        plt.text(var1[i],var2[i],models_hist[i])
    plt.xlim([min(var1),max(var1)])
    plt.ylim([min(var2),max(var2)])


def get_jet_lat(models):

    ua_hist =0
    ua_rcp =0
    N=np.size(models)
    jet_loc_hist = np.zeros((N,12))
    jet_loc_rcp = np.zeros((N,12))
    lon1=48;lon2=65; #100-->180
    #lon1=30;lon2=65; #100-->180
    lat1=225;lat2=267; #0-->60
    i=0
    for name in models:
        os.chdir('/Volumes/Zhou/data/cmip5/historical/ua/'+name+'/r1i1p1/')
        f = netcdf.netcdf_file('ua_'+name+'.month.regrid.hi.nc', 'r')
        lat = f.variables['lat'][:]
        lat_r = lat[lat1:lat2]
        ua_hist_ = np.squeeze(f.variables['ua'][:,7,lat1:lat2,lon1:lon2])
        os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ua/'+name+'/r1i1p1/')
        f = netcdf.netcdf_file('ua_Amon_'+name+'_rcp85_r1i1p1_208001-210012.month.regri
        ua_rcp_ = np.squeeze(f.variables['ua'][:,7,lat1:lat2,lon1:lon2])

        ua_hist = ua_hist + ua_hist_/len(models)
        ua_rcp = ua_rcp + ua_rcp_/len(models)
        #print lat_r[np.argmax(ua_hist_,axis=1)][6,:]
        #print lat_r[np.argmax(ua_rcp_,axis=1)][6,:]
        jet_loc_hist[i,:] = np.mean(lat_r[np.argmax(ua_hist_,axis=1)],1)
        jet_loc_rcp[i,:] = np.mean(lat_r[np.argmax(ua_rcp_,axis=1)],1)


        #jet_loc_hist[i,:] = lat_r[np.argmax(np.mean(ua_hist_,2),axis=1)]
        #jet_loc_rcp[i,:] = lat_r[np.argmax(np.mean(ua_rcp_,2),axis=1)]



        i=i+1
```

```
            return jet_loc_hist,jet_loc_rcp
```

```
/anaconda3/lib/python3.7/site-packages/scipy/io/netcdf.py:317: RuntimeWarning: Cannot close a r
  ), category=RuntimeWarning)
```

In [57]:
```
months = ['J','F','M','A','M','J','J','A','S','O','N','D']
#get land mask
f = netcdf.netcdf_file('/Volumes/Lacie/Downloads/orog_fx_CCSM4_amip_r0i0p0.regrid.nc'
orog = f.variables['orog'][::]
mask=copy.deepcopy(orog)
mask[orog>0]=np.nan
mask[orog==0]=1
# zoanl mean amip
os.chdir('/Volumes/Zhou/data/cmip5/amip/wap')
f = netcdf.netcdf_file('wap.month.regrid.mm.2d.nc', 'r')
lat = f.variables['lat'][:]
omega_p500_2d_amip = np.squeeze(f.variables['wap'][:,5,:]*864.)

omega_p500_2d_am_amip = np.mean(omega_p500_2d_amip,0)
os.chdir('/Volumes/Zhou/data/cmip5/amip4K/wap')
f = netcdf.netcdf_file('wap.month.regrid.mm.2d.nc', 'r')
omega_p500_2d_amip4k = np.squeeze(f.variables['wap'][:,5,:]*864.)
omega_p500_2d_am_amip4k = np.mean(omega_p500_2d_amip4k,0)
os.chdir('/Volumes/Zhou/data/cmip5/amipFuture/wap')
f = netcdf.netcdf_file('wap.month.regrid.mm.2d.nc', 'r')
omega_p500_2d_amipfuture = np.squeeze(f.variables['wap'][:,5,:]*864.)
omega_p500_2d_am_amipfuture = np.mean(omega_p500_2d_amipfuture,0)

os.chdir('/Volumes/Zhou/data/cmip5/amip/tas')
f = netcdf.netcdf_file('tas.month.regrid.mm.nc', 'r')
ts_amip = f.variables['tas'][::]
ts_2d_amip = np.nanmean(ts_amip*mask[None,:,:],2)

os.chdir('/Volumes/Zhou/data/cmip5/amipFuture/tas')
f = netcdf.netcdf_file('tas.month.regrid.mm.nc', 'r')
ts_amip = f.variables['tas'][::]
ts_2d_amipfuture = np.nanmean(ts_amip*mask[None,:,:],2)

aa=ts_2d_amip.T-np.nanmean(ts_2d_amip[:,40:80],1)[None,:]
bb=ts_2d_amipfuture.T-np.nanmean(ts_2d_amipfuture[:,40:80],1)[None,:]
cc=bb-aa;
dtp_warm_futurediff = np.mean(cc[57:64,:]);

[lat1_omega_amip,lat2_omega_amip]=get_min_loc(omega_p500_2d_amip)
[lat1_omega_amip4k,lat2_omega_amip4k]=get_min_loc(omega_p500_2d_amip4k)
[lat1_omega_amipfuture,lat2_omega_amipfuture]=get_min_loc(omega_p500_2d_amipfuture)
```

```
srang1_amip = np.std(np.array(list(lat1_omega_amip)+list(lat2_omega_amip)))
srang1_amip4k = np.std(np.array(list(lat1_omega_amip4k)+list(lat2_omega_amip4k)))
srang1_amipfuture = np.std(np.array(list(lat1_omega_amipfuture)+list(lat2_omega_amipfu
srang1_4kdiff = srang1_amip4k - srang1_amip
srang1_futurediff = srang1_amipfuture - srang1_amip

arang_amip = get_annual_range(omega_p500_2d_am_amip)
arang_amip4k = get_annual_range(omega_p500_2d_am_amip4k)
arang_amipfuture = get_annual_range(omega_p500_2d_am_amipfuture)
arang_4kdiff = arang_amip4k - arang_amip
arang_futurediff = arang_amipfuture - arang_amip

# zonal mean historical
os.chdir('/Volumes/Zhou/data/cmip5/historical/pr')
f = netcdf.netcdf_file('pr.month.regrid.mm.2d.nc', 'r')
pr_2d_hist = f.variables['pr'][:,:]*86400.
pr_2d_annual_hist = np.squeeze(np.mean(pr_2d_hist,0))
pr_2d_summer_hist = np.squeeze(np.mean(pr_2d_hist[5:7,:],0))
pr_2d_winter_hist = np.squeeze(np.mean(pr_2d_hist[0:2,:],0))
lat = f.variables['lat'][:]
os.chdir('/Volumes/Zhou/data/cmip5/historical/thetae_p925')
f = netcdf.netcdf_file('thetae.month.regrid.mm.2d.nc', 'r')
thetae_2d_hist = f.variables['thetae'][:,:]
# os.chdir('/Volumes/Zhou/data/cmip5/historical/Fnet')
# f = netcdf.netcdf_file('Fnet.month.regrid.mm.2d.nc', 'r')
# Fnet_2d_hist = f.variables['Fnet'][:,:]
os.chdir('/Volumes/Zhou/data/cmip5/historical/wap')
f = netcdf.netcdf_file('wap.month.regrid.mm.2d.nc', 'r')
plev = f.variables['plev'][:]
omega_p500_2d_hist = np.squeeze(f.variables['wap'][:,5,:]*864.)
omega_p500_2d_annual_hist = np.squeeze(np.mean(f.variables['wap'][:,5,:]*864.,0))
omega_2d_annual_hist = np.squeeze(np.mean(f.variables['wap'][:,:,:]*864.,0))
omega_2d_summer_hist = np.squeeze(np.mean(f.variables['wap'][5:7,:,:]*864.,0))
omega_2d_winter_hist = np.squeeze(np.mean(f.variables['wap'][0:2,:,:]*864.,0))
os.chdir('/Volumes/Zhou/data/cmip5/historical/ua')
f = netcdf.netcdf_file('ua.month.regrid.mm.2d.nc', 'r')
ua_2d_p500_hist = f.variables['ua'][:,8,:]
ua_2d_annual_hist = np.squeeze(np.mean(f.variables['ua'][:,:,:],0))
ua_2d_summer_hist = np.squeeze(np.mean(f.variables['ua'][5:7,:,:],0))
ua_2d_winter_hist = np.squeeze(np.mean(f.variables['ua'][0:2,:,:],0))


# lat-lon historical
os.chdir('/Volumes/Zhou/data/cmip5/historical/pr')
f = netcdf.netcdf_file('pr.month.regrid.mm.nc', 'r')
pr_hist = f.variables['pr'][:,:]*86400.
lat = f.variables['lat'][:]
pr_summer_hist = np.mean(pr_hist[5:7,::],0)
```

```
lon = f.variables['lon'][:]
lon_cmip5 = f.variables['lon'][:]
lat_cmip5 = f.variables['lat'][:]
os.chdir('/Volumes/Zhou/data/cmip5/historical/hfls')
f = netcdf.netcdf_file('hfls.month.regrid.mm.nc', 'r')
evap_hist = f.variables['hfls'][:,:]*86400./2500000.
evap_summer_hist = np.mean(evap_hist[5:7,::],0)
evap_2d_hist = np.mean(evap_hist,2)
evap_2d_summer_hist = np.mean(evap_2d_hist[5:7,:],0)
evap_2d_winter_hist = np.mean(evap_2d_hist[0:2,:],0)
os.chdir('/Volumes/Zhou/data/cmip5/historical/ts')
f = netcdf.netcdf_file('ts.month.regrid.mm.nc', 'r')
ts_hist = f.variables['ts'][:,:,:]
ts_2d_hist = np.nanmean(ts_hist*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/historical/tas')
f = netcdf.netcdf_file('tas.month.regrid.mm.nc', 'r')
tas_hist = f.variables['tas'][:,:,:]
tas_2d_hist = np.nanmean(tas_hist*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/historical/huss')
f = netcdf.netcdf_file('huss.month.regrid.mm.nc', 'r')
huss_hist = f.variables['huss'][::]
huss_2d_hist = np.nanmean(huss_hist*mask[None,:,:],2)
hsfc_hist = tas_hist + huss_hist*2.5e6/1004.5
os.chdir('/Volumes/Zhou/data/cmip5/historical/ta')
f = netcdf.netcdf_file('ta.month.regrid.mm.nc', 'r')
ta_hist = f.variables['ta'][:,1,:,:]
ta_2d_hist = np.nanmean(ta_hist*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/historical/hus')
f = netcdf.netcdf_file('hus.month.regrid.mm.nc', 'r')
hus_hist = f.variables['hus'][:,1,:,:]
hus_2d_hist = np.nanmean(hus_hist*mask[None,:,:],2)
h_hist = ta_hist + hus_hist*2.5e6/1004.5

os.chdir('/Volumes/Zhou/data/cmip5/historical/ta')
f = netcdf.netcdf_file('ta.month.regrid.mm.nc', 'r')
ta_hist = f.variables['ta'][:,:,:,:]
ta_2d_hist = np.mean(ta_hist,3)
ta_2d_annual_hist = np.squeeze(np.mean(ta_2d_hist[:,:,:],0))
ta_2d_summer_hist = np.squeeze(np.mean(ta_2d_hist[5:7,:,:],0))
ta_2d_winter_hist = np.squeeze(np.mean(ta_2d_hist[0:2,:,:],0))

os.chdir('/Volumes/Zhou/data/cmip5/historical/thetae_p925')
f = netcdf.netcdf_file('thetae.month.regrid.mm.nc', 'r')
thetae_hist = f.variables['thetae'][:,:,:]
thetae_2d_hist = np.nanmean(thetae_hist*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/historical/wap')
f = netcdf.netcdf_file('wap.month.regrid.mm.nc', 'r')
omega_p500_hist = np.squeeze(f.variables['wap'][:,5,:,:]*864.)
```

```
omega_p500_asia_hist = np.squeeze(np.mean(f.variables['wap'][:,5,:,47:62]*864.,2))
omega_p500_atlantic_hist = np.squeeze(np.mean(f.variables['wap'][:,5,:,105:118]*864.,
omega_p500_summer_hist = np.squeeze(np.mean(f.variables['wap'][5:7,5,:,:]*864.,0))
omega_p500_winter_hist = np.squeeze(np.mean(f.variables['wap'][0:2,5,:,:]*864.,0))

f = netcdf.netcdf_file('wap.month.regrid.nc', 'r')
omega_p500_summer_hist_ = np.squeeze(np.mean(f.variables['wap'][:,5:7,5,:,:]*864.,1))


os.chdir('/Volumes/Zhou/data/cmip5/historical/ua')
f = netcdf.netcdf_file('ua.month.regrid.mm.nc', 'r')
ua_p500_hist = np.squeeze(f.variables['ua'][:,8,:,:])
ua_p500_asia_hist = np.squeeze(np.mean(f.variables['ua'][:,8,:,47:62],2))
ua_p500_atlantic_hist = np.squeeze(np.mean(f.variables['ua'][:,8,:,105:118],2))
ua_p500_summer_hist = np.squeeze(np.mean(f.variables['ua'][5:7,8,:,:],0))
ua_p500_winter_hist = np.squeeze(np.mean(f.variables['ua'][0:2,8,:,:],0))
f = netcdf.netcdf_file('ua.month.regrid.nc', 'r')
ua_p500_hist_ = np.squeeze(f.variables['ua'][:,:,8,:,:])
ua_p500_asia_hist_ = np.squeeze(np.mean(f.variables['ua'][:,:,8,:,47:62],3))
ua_p500_summer_hist_ = np.squeeze(np.mean(f.variables['ua'][:,5:7,8,:,:],1))

# os.chdir('/Volumes/Zhou/data/cmip5/historical/Fnet')
# f = netcdf.netcdf_file('Fnet.month.regrid.mm.nc', 'r')
# Fnet_hist = np.squeeze(f.variables['Fnet'][:,:,:])

#2d rcp
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/pr')
f = netcdf.netcdf_file('pr_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.2d.nc', 'r
pr_2d_rcp = f.variables['pr'][::]*86400.
pr_2d_annual_rcp = np.squeeze(np.mean(pr_2d_rcp,0))
pr_2d_summer_rcp = np.squeeze(np.mean(pr_2d_rcp[5:7,:],0))
pr_2d_winter_rcp = np.squeeze(np.mean(pr_2d_rcp[0:2,:],0))
lat = f.variables['lat'][:]
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/thetae_p925')
f = netcdf.netcdf_file('thetae_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.2d.nc'
thetae_2d_rcp = f.variables['thetae'][::]
# os.chdir('/Volumes/Zhou/data/cmip5/rcp85/Fnet')
# f = netcdf.netcdf_file('Fnet_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.2d.nc'
# Fnet_2d_rcp = f.variables['Fnet'][::]
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap')
f = netcdf.netcdf_file('wap_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.2d.nc', '
omega_p500_2d_rcp = np.squeeze(f.variables['wap'][:,5,:]*864.)
omega_p500_2d_annual_rcp = np.squeeze(np.mean(f.variables['wap'][:,5,:]*864.,0))
omega_2d_annual_rcp = np.squeeze(np.mean(f.variables['wap'][:,:,:]*864.,0))
omega_2d_summer_rcp = np.squeeze(np.mean(f.variables['wap'][5:7,:,:]*864.,0))
omega_2d_winter_rcp = np.squeeze(np.mean(f.variables['wap'][0:2,:,:]*864.,0))
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ua')
f = netcdf.netcdf_file('ua_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.2d.nc', 'r
```

```
ua_2d_p500_rcp = f.variables['ua'][:,8,:]
ua_2d_annual_rcp = np.squeeze(np.mean(f.variables['ua'][:,:,:],0))
ua_2d_summer_rcp = np.squeeze(np.mean(f.variables['ua'][5:7,:,:],0))
ua_2d_winter_rcp = np.squeeze(np.mean(f.variables['ua'][0:2,:,:],0))



#lat-lon rcp
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/pr')
f = netcdf.netcdf_file('pr_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
pr_rcp = f.variables['pr'][::]*86400.
pr_summer_rcp = np.mean(pr_rcp[5:7,::],0)
pr_winter_rcp = np.mean(pr_rcp[0:2,::],0)
lon = f.variables['lon'][:]

os.chdir('/Volumes/Zhou/data/cmip5/rcp85/hfls')
f = netcdf.netcdf_file('hfls_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
evap_rcp = f.variables['hfls'][:,:]*86400./2500000.
evap_summer_rcp = np.mean(evap_rcp[5:7,::],0)
evap_2d_rcp = np.mean(evap_hist,2)
evap_2d_summer_rcp = np.mean(evap_2d_rcp[5:7,:],0)
evap_2d_winter_rcp = np.mean(evap_2d_rcp[0:2,:],0)
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ts')
f = netcdf.netcdf_file('ts_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
ts_rcp = f.variables['ts'][::]
ts_2d_rcp = np.nanmean(ts_rcp*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/tas')
f = netcdf.netcdf_file('tas_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
tas_rcp = f.variables['tas'][::]
tas_2d_rcp = np.nanmean(tas_rcp*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/huss')
f = netcdf.netcdf_file('huss_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
huss_rcp = f.variables['huss'][::]
huss_2d_rcp = np.nanmean(huss_rcp*mask[None,:,:],2)
hsfc_rcp = tas_rcp + huss_rcp*2.5e6/1004.5

os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ta')
f = netcdf.netcdf_file('ta_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
ta_rcp = f.variables['ta'][:,1,:,:]
ta_2d_rcp = np.nanmean(ta_rcp*mask[None,:,:],2)
os.chdir('/Volumes/Zhou/data/cmip5/rcp85/hus')
f = netcdf.netcdf_file('hus_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
hus_rcp = f.variables['hus'][:,1,:,:]
hus_2d_rcp = np.nanmean(hus_rcp*mask[None,:,:],2)
h_rcp = ta_rcp + hus_rcp*2.5e6/1004.5


os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ta')
```

```
f = netcdf.netcdf_file('ta_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
ta_rcp = f.variables['ta'][:,:,:,:]
ta_2d_rcp = np.mean(ta_rcp,3)
ta_2d_annual_rcp = np.squeeze(np.mean(ta_2d_rcp[:,:,:],0))
ta_2d_summer_rcp = np.squeeze(np.mean(ta_2d_rcp[5:7,:,:],0))
ta_2d_winter_rcp = np.squeeze(np.mean(ta_2d_rcp[0:2,:,:],0))

os.chdir('/Volumes/Zhou/data/cmip5/rcp85/thetae_p925')
f = netcdf.netcdf_file('thetae_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', '
thetae_rcp = f.variables['thetae'][::]
thetae_2d_rcp = np.nanmean(thetae_rcp*mask[None,:,:],2)

os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap')
f = netcdf.netcdf_file('wap_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
omega_p500_rcp = np.squeeze(f.variables['wap'][:,5,:,:]*864.)
omega_p500_asia_rcp = np.squeeze(np.mean(f.variables['wap'][:,5,:,47:62]*864.,2))
omega_p500_atlantic_rcp = np.squeeze(np.mean(f.variables['wap'][:,5,:,105:118]*864.,2
omega_p500_summer_rcp = np.squeeze(np.mean(f.variables['wap'][5:7,5,:,:]*864.,0))
omega_p500_winter_rcp = np.squeeze(np.mean(f.variables['wap'][0:2,5,:,:]*864.,0))
f = netcdf.netcdf_file('wap_Amon_rcp85_r1i1p1_208001-210012.month.regrid.nc', 'r')
omega_p500_summer_rcp_ = np.squeeze(np.mean(f.variables['wap'][:,5:7,5,:,:]*864.,1))

os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ua')
f = netcdf.netcdf_file('ua_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', 'r')
ua_p500_rcp = np.squeeze(f.variables['ua'][:,8,:,:])
ua_p500_asia_rcp = np.squeeze(np.mean(f.variables['ua'][:,8,:,47:62],2))
ua_p500_atlantic_rcp = np.squeeze(np.mean(f.variables['ua'][:,8,:,105:118],2))
ua_p500_summer_rcp = np.squeeze(np.mean(f.variables['ua'][5:7,8,:,:],0))
ua_p500_winter_rcp = np.squeeze(np.mean(f.variables['ua'][0:2,8,:,:],0))

f = netcdf.netcdf_file('ua_Amon_rcp85_r1i1p1_208001-210012.month.regrid.nc', 'r')
ua_2p500_rcp_ = np.squeeze(f.variables['ua'][:,:,8,:,:])
ua_p500_asia_rcp_ = np.squeeze(np.mean(f.variables['ua'][:,:,8,:,47:62],3))
ua_p500_summer_rcp_ = np.squeeze(np.mean(f.variables['ua'][:,5:7,8,:,:],1))


# os.chdir('/Volumes/Zhou/data/cmip5/rcp85/Fnet')
# f = netcdf.netcdf_file('Fnet_Amon_rcp85_r1i1p1_208001-210012.month.regrid.mm.nc', '
# Fnet_rcp = f.variables['Fnet'][::]

os.chdir('/Volumes/Zhou/data/cmip5/amipFuture/ts/CCSM4/r1i1p1/')
f = netcdf.netcdf_file('ts_CCSM4.month.regrid.nc', 'r')
ts_amipfuture = f.variables['ts'][::]
ts_2d_amipfuture = np.nanmean(ts_amipfuture*mask[None,:,:],2)
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:27: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:32: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:104: RuntimeWarning: Mean of empt
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:108: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:112: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:117: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:121: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:135: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:212: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:216: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:220: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:226: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:230: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:245: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:278: RuntimeWarning: Mean of empty
```

In [63]:
```
models_hist=['ACCESS1-0','ACCESS1-3','bcc-csm1-1','BNU-ESM','CanESM2','CCSM4','CESM1-C
             'CSIRO-Mk3-6-0','GFDL-CM3','GISS-E2-R','HadGEM2-ES','inmcm4','IPSL-CM5A-LR','I
             'MIROC5','MIROC-ESM','MPI-ESM-LR','MRI-CGCM3','NorESM1-M']
lat_len = 361
lon_len = 144
lev_len = 12
omega_p500_hist = np.zeros((len(models_hist),12,lat_len,lon_len))
omega_p500_rcp85 = np.zeros((len(models_hist),12,lat_len,lon_len))


for i in range(len(models_hist)):
    model = models_hist[i]

    os.chdir('/Volumes/Zhou/data/cmip5/historical/wap/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.hdeg.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    omega_p500_hist[i,::] = np.squeeze(f.variables['wap'][:,5,:,:]*864.)

    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.hdeg.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    omega_p500_rcp85[i,::] = np.squeeze(f.variables['wap'][:,5,:,:]*864.)
    lat_hi = f.variables['lat'][:]

    lon_hi = f.variables['lon'][:]
```

In [64]:
```
models_hist_=['ACCESS1-0','bcc-csm1-1','BNU-ESM','CanESM2','CCSM4','CESM1-CAM5','CNRM-
             'CSIRO-Mk3-6-0','GFDL-CM3','GISS-E2-R','HadGEM2-ES','inmcm4','IPSL-CM5A-LR',\
             'MIROC5','MIROC-ESM','MPI-ESM-LR','MRI-CGCM3','NorESM1-M']

dmse_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
mseu_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
msel_hist= np.zeros((len(models_hist_),12,lat_len,lon_len))
Fnet_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
```

```
rad_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
hfs_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
swrad_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
lwrad_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
lwradtoa_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
rlus_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
rlutcs_hist  = np.zeros((len(models_hist_),12,lat_len,lon_len))
lwradcldtoa_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
thetae_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))
ts_hist = np.zeros((len(models_hist_),12,lat_len,lon_len))


dmse_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
mseu_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
msel_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
Fnet_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
rad_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
hfs_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
swrad_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
lwrad_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
lwradtoa_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
rlus_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
rlutcs_rcp85  = np.zeros((len(models_hist_),12,lat_len,lon_len))
lwradcldtoa_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
thetae_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
ts_rcp85 = np.zeros((len(models_hist_),12,lat_len,lon_len))
for i in range(len(models_hist_)):
    model = models_hist_[i]

    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/thetae_p925/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.hdeg.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    thetae_rcp85[i,::]= np.squeeze(f.variables['thetae'][::])
    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ts/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.hdeg.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    ts_rcp85[i,::]= np.squeeze(f.variables['ts'][::])

    os.chdir('/Volumes/Zhou/data/cmip5/historical/Fnet/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.hdeg.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    Fnet_hist[i,::]= np.squeeze(f.variables['Fnet'][::])
    lwradtoa_hist[i,::]= -np.squeeze(f.variables['rlut'][::])
    os.chdir('/Volumes/Zhou/data/cmip5/historical/rlutcs/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.hdeg.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    rlutcs_hist[i,::]= np.squeeze(f.variables['rlutcs'][::])
    lwradcldtoa_hist[i,::]= lwradtoa_hist[i,::] + rlutcs_hist[i,::]
```

```
            os.chdir('/Volumes/Zhou/data/cmip5/historical/mse/'+model+'/r1i1p1/')
            filename = glob.glob('*regrid.hdeg.hi.nc')[0]
            f = netcdf.netcdf_file(filename, 'r')
            mse=f.variables['mse'][::]-0.
            mse[np.abs(mse)>2000.]=np.nan
            dmse_hist[i,::] = np.squeeze(np.mean(mse[:,20:27,:,:],1)-np.mean(mse[:,1:8,:,:],1
            mseu_hist[i,::] = np.squeeze(np.mean(mse[:,20:27,:,:],1))
            msel_hist[i,::] = np.squeeze(np.mean(mse[:,1:8,:,:],1))

            os.chdir('/Volumes/Zhou/data/cmip5/historical/thetae_p925/'+model+'/r1i1p1/')
            filename = glob.glob('*regrid.hdeg.nc')[0]
            f = netcdf.netcdf_file(filename, 'r')
            thetae_hist[i,::]= np.squeeze(f.variables['thetae'][::])
            os.chdir('/Volumes/Zhou/data/cmip5/historical/ts/'+model+'/r1i1p1/')
            filename = glob.glob('*regrid.hdeg.nc')[0]
            f = netcdf.netcdf_file(filename, 'r')
            ts_hist[i,::]= np.squeeze(f.variables['ts'][::])

            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/Fnet/'+model+'/r1i1p1/')
            filename = glob.glob('*regrid.hdeg.nc')[0]
            f = netcdf.netcdf_file(filename, 'r')
            Fnet_rcp85[i,::]= np.squeeze(f.variables['Fnet'][::])
            lwradtoa_rcp85[i,::]= -np.squeeze(f.variables['rlut'][::])
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/rlutcs/'+model+'/r1i1p1/')
            filename = glob.glob('*regrid.hdeg.nc')[0]
            f = netcdf.netcdf_file(filename, 'r')
            rlutcs_rcp85[i,::]= np.squeeze(f.variables['rlutcs'][::])
            lwradcldtoa_rcp85[i,::]= lwradtoa_rcp85[i,::] + rlutcs_rcp85[i,::]
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/mse/'+model+'/r1i1p1/')
            filename = glob.glob('*regrid.hdeg.hi.nc')[0]
            f = netcdf.netcdf_file(filename, 'r')
            mse = f.variables['mse'][::]-0.
            plev_hi = f.variables['plev'][:]

            lat_hi = f.variables['lat'][:]

            lon_hi = f.variables['lon'][:]
            mse[np.abs(mse)>2000.]=np.nan
            dmse_rcp85[i,::] = np.squeeze(np.mean(mse[:,21:28,:,:],1)-np.mean(mse[:,1:8,:,:],1
            mseu_rcp85[i,::] = np.squeeze(np.mean(mse[:,21:28,:,:],1))
            msel_rcp85[i,::] = np.squeeze(np.mean(mse[:,1:8,:,:],1))
```

In [70]:
```
fig=plt.figure(figsize=(16,11))
widths = [2, 1.8,0.6 ]
heights = [1, 1]
spec = fig.add_gridspec(ncols=3, nrows=2, width_ratios=widths,height_ratios=heights)
```

```
ax=fig.add_subplot(spec[0,0])
maphi.drawcoastlines(linewidth=0.6)
plt.yticks([-10,-5,0,5,10],['10$\mathrm{^o}$S','5$\mathrm{^o}$S','0','5$\mathrm{^o}$N
plt.xticks([60,120,180,240,300],['60$\mathrm{^o}$E','120$\mathrm{^o}$E','180$\mathrm{^
maphi.fillcontinents()
tmp1=np.mean(omega_p500_hist[:,5:8,::],1)
bb=np.mean(omega_p500_hist[:,[11,0,1],::],1)
tmp1[:,:181,:] = bb[:,:181,:]
latmax1_nh,latmax1_sh = get_latwt_tp(np.mean(tmp1,0),lat_hi)
tmp1[:,180,:] = np.nan
tmp2=np.mean(omega_p500_rcp85[:,5:8,::],1)
bb=np.mean(omega_p500_rcp85[:,[11,0,1],::],1)
tmp2[:,:181,:] = bb[:,:181,:]
latmax2_nh,latmax2_sh = get_latwt_tp(np.mean(tmp2,0),lat_hi)
tmp2[:,180,:] = np.nan
ind1 = np.argmax(lon>60)
ind2 = np.argmax(lon>280)
plt.plot(lon[ind1:ind2],latmax1_nh[ind1:ind2],'k--',linewidth=2.5)
plt.plot(lon[ind1:ind2],latmax2_nh[ind1:ind2],'m--',linewidth=2.5)
ind3 = np.argmax(lon>320)
ind4 = np.argmax(lon>350)
plt.plot(lon[ind3:ind4],latmax1_nh[ind3:ind4],'k--',linewidth=2.5)
plt.plot(lon[ind3:ind4],latmax2_nh[ind3:ind4],'m--',linewidth=2.5)
ind5 = np.argmax(lon>50)
ind6 = np.argmax(lon>200)
plt.plot(lon[ind5:ind6],latmax1_sh[ind5:ind6],'k--',linewidth=2.5)
plt.plot(lon[ind5:ind6],latmax2_sh[ind5:ind6],'m--',linewidth=2.5)


aa = tmp2-tmp1;
bb=np.sum(aa>0,0)
#CS1=maphi.contourf(x_hi,y_hi,np.mean(tmp2-tmp1,0),[-16,-12,-8,-4,4,8,12,16],cmap='bw
CS1=maphi.contourf(x_hi,y_hi,np.mean(tmp2-tmp1,0),get_array(-28,35,7),cmap='bwr',exte
maphi.contour(x_hi,y_hi,-np.mean(tmp1,0),np.arange(35,76,24),colors='k',alpha=0.7,zor
#maphi.contour(x_hi,y_hi,np.mean(tmp1,0),[-30],colors='k')
aa = tmp2-tmp1;
bb=np.sum(tmp2-tmp1>0,0)-len(models_hist_)/2
bb[180,:]=0
inv1 = 3; inv2=3
maphi.scatter(x_hi[::inv1,::inv2][np.abs(bb[::inv1,::inv2])>5],y_hi[::inv1,::inv2][np
# plt.plot(lon[ind1:ind2],latmax1_nh[ind1:ind2],'k-',linewidth=2.5)
# plt.plot(lon[ind3:ind4],latmax1_nh[ind3:ind4],'k-',linewidth=2.5)
# plt.plot(lon[ind5:ind6],latmax1_sh[ind5:ind6],'k-',linewidth=2.5)
cb=plt.colorbar(CS1,orientation='horizontal',shrink=0.75,extendfrac=0)
cb.ax.tick_params(labelsize=12.5);#cb.set_label('[ hPa day$^{-1}$ ]',fontsize=14,labe
ax.set_aspect('auto')
plt.tick_params(labelsize=12.5,direction='in',length=5)
plt.text(-0.18, 1.14,'a',fontsize=22,transform = ax.transAxes)
```

```
plt.title('Seasonal Mean',fontsize=17.5,loc='center')
#plt.text(-0.4,0.5,'RCP8.5 $\minus$ HIST',rotation=90,fontsize=22,horizontalalignment=
#plt.title(r'$\mathregular{\Delta \omega_{500}}$',fontsize=22,loc='left')
plt.grid(linewidth='0.8')
plt.text(-0.28, 0.5,'$\mathregular{\omega_{500}}$',rotation=90,fontsize=21,horizontala
plt.text(-0.21, 0.56,'(HIST vs RCP8.5)',rotation=90,fontsize=17.5,horizontalalignment=

#plt.text(-0.2, 0.882,'Jun-Jul-Aug',rotation=90,color='r',fontsize=13.3,transform = ax
#plt.text(-0.2, 0.422,'Dec-Jan-Feb',rotation=90,color='r',fontsize=13.3,transform = ax

ax=fig.add_subplot(spec[0,1])
time = np.arange(1,13,1)
omega_p500_hist[np.abs(omega_p500_hist)>1000.]=np.nan
omega_p500_rcp85[np.abs(omega_p500_rcp85)>1000.]=np.nan
omega_p500_2d_hist = np.mean(np.nanmean(omega_p500_hist,3),0)
omega_p500_2d_rcp = np.mean(np.nanmean(omega_p500_rcp85,3),0)
[lat_omega1,lat_omega2]=get_min_loc2(omega_p500_2d_hist)
[lat_omega3,lat_omega4]=get_min_loc2(omega_p500_2d_rcp)
time=np.arange(0,14,1)
aa=omega_p500_2d_rcp-omega_p500_2d_hist
bb=get_extend(aa)
CS1=plt.contourf(time,lat_hi,bb.T,[-8,-6,-4,-2,2,4,6,8],cmap='bwr')
plt.contour(time,lat_hi,-get_extend(omega_p500_2d_hist).T,[22],linewidths=1.5,colors=
plt.contour(time,lat_hi,-get_extend(omega_p500_2d_rcp).T,[22],linewidths=1.5,colors='r
plt.ylim([-12,12]);plt.xlim([0.5,12.5])
plt.plot(range(1,5),lat_omega1[0:-1],'k--',linewidth=2.5)
plt.plot(range(6,12),lat_omega2[1:-1],'k--',linewidth=2.5)
plt.plot(range(1,5),lat_omega3[0:-1],'m--',linewidth=2.5)
plt.plot(range(6,12),lat_omega4[1:-1],'m--',linewidth=2.5)
plt.plot([0.5,12.5],[0,0],'k--',linewidth=1,dashes=(10, 5))
cc=np.mean(omega_p500_rcp85,3)-np.mean(omega_p500_hist,3)
dd=np.sum(cc-0.5>0,axis=0)-9
time1=np.arange(1,13,1)
critic = 6
lat2,time2 = np.meshgrid(lat_hi,time1)
inv1=1;inv2=3
plt.scatter(time2[::inv1,::inv2][np.abs(dd[::inv1,::inv2])>6],lat2[::inv1,::inv2][np.a
plt.yticks([-10,-5,0,5,10],['10$\mathrm{^o}$S','5$\mathrm{^o}$S','0','5$\mathrm{^o}$N
plt.tick_params(direction='in',labelsize=12.5,length=5)
plt.ylim([-12,12]);plt.xlim([0.5,12.5])
cb=plt.colorbar(CS1,orientation='horizontal',shrink=0.78,extendfrac=0.1)
cb.ax.tick_params(labelsize=12.5)
#cb.set_label('[ hPa day$^{-1}$ ]',fontsize=15)
plt.text(-0.2, 1.14,'b',fontsize=22,transform = ax.transAxes)
plt.title(r'Zonal Mean',loc='center',fontsize=17.5)
plt.xticks(np.arange(1,13,1),months,fontsize=11.8)
#plt.title(r'$\mathregular{\Delta \omega_{500}}$',loc='left',fontsize=22,pad=8)
```

```
ax09=fig.add_subplot(spec[0,2])
lat_hi1 = np.arange(-12,12.1,0.1)
f=interpolate.interp1d(lat_hi,np.nanmean(omega_p500_2d_hist,0),'cubic')
lati1,lati2=get_annual_max2(np.nanmean(omega_p500_2d_hist,0))
CS=plt.plot(f(lat_hi1),lat_hi1,color='k',linestyle='-',linewidth=2.5)
plt.plot(0,lati1,'>',markersize=18,color='k');
plt.plot(0,lati2,'>',markersize=18,color='k');
f=interpolate.interp1d(lat_hi,np.nanmean(omega_p500_2d_rcp,0),'cubic')
lati1,lati2=get_annual_max2(np.nanmean(omega_p500_2d_rcp,0))
CS=plt.plot(f(lat_hi1),lat_hi1,'m-',linewidth=2.5)
plt.plot(0,lati1,'>',markersize=18,color='m');
plt.plot(0,lati2,'>',markersize=18,color='m');
plt.ylim([-12,12]);plt.xlim([-25,0])
plt.xticks([-20,-10,0]);plt.yticks([-10,-5,0,5,10],[])
ax09.spines['bottom'].set_position('center')
ax09.spines['right'].set_color('none')
ax09.spines['top'].set_color('none')
ax09.xaxis.set_ticks_position('bottom')
ax09.yaxis.set_ticks_position('left')
plt.tick_params(direction='in',labelsize=12.5,length=6)
plt.gca().invert_xaxis()
plt.text(-0.36,1.05,'Annual Zonal Mean',fontsize=17.5,transform = ax09.transAxes)
plt.text(-0.55, 1.14,'c',fontsize=22,transform = ax09.transAxes)
plt.text(0., -.266,'[hPa day$^{-1}$]',fontsize=15.7,transform = ax09.transAxes)


ax=fig.add_subplot(spec[1,0])
maphi.drawcoastlines(linewidth=0.6)
plt.yticks([-10,-5,0,5,10],['10$\mathrm{^o}$S','5$\mathrm{^o}$S','0','5$\mathrm{^o}$N
plt.xticks([60,120,180,240,300],['60$\mathrm{^o}$E','120$\mathrm{^o}$E','180$\mathrm{^
maphi.fillcontinents()
tmp1=np.mean(ts_hist[:,5:8,::],1)
bb=np.mean(ts_hist[:,[11,0,1],::],1)
tmp1[:,:181,:] = bb[:,:181,:]

aa = np.mean(np.mean(ts_hist[:,5:8,::],1),0)
aa[np.isnan(maskhi)]=np.nan
dd=np.nanmean(aa[140:221,:],0)
ee = aa - dd[None,:]

# aa[np.isnan(maskhi)]=np.nan
# dd = np.nanmean(aa[140:221,:])
# ee = aa-dd
latmax1_nh,latmax1_sh_ = get_latwt_tp(-ee,lat_hi)

aa = np.mean(np.mean(ts_hist[:,[11,0,1],::],1),0)
aa[np.isnan(maskhi)]=np.nan
dd=np.nanmean(aa[140:221,:],0)
```

```
ee = aa - dd[None,:]

# aa[np.isnan(maskhi)]=np.nan
# dd = np.nanmean(aa[140:221,:])
# ee = aa-dd
latmax1_nh_,latmax1_sh = get_latwt_tp(-ee,lat_hi)

tmp1[:,180,:] = np.nan
tmp2=np.mean(ts_rcp85[:,5:8,::],1)
bb=np.mean(ts_rcp85[:,[11,0,1],::],1)
tmp2[:,:181,:] = bb[:,:181,:]

aa = np.mean(np.mean(ts_rcp85[:,5:8,::],1),0)
aa[np.isnan(maskhi)]=np.nan
dd=np.nanmean(aa[140:221,:],0)
ee = aa - dd[None,:]

# aa[np.isnan(maskhi)]=np.nan
# dd = np.nanmean(aa[140:221,:])
# ee = aa-dd
latmax2_nh,latmax2_sh_ = get_latwt_tp(-ee,lat_hi)

aa = np.mean(np.mean(ts_rcp85[:,[11,0,1],::],1),0)
aa[np.isnan(maskhi)]=np.nan
dd=np.nanmean(aa[140:221,:],0)
ee = aa - dd[None,:]

# aa[np.isnan(maskhi)]=np.nan
# dd = np.nanmean(aa[140:221,:])
# ee = aa-dd

latmax2_nh_,latmax2_sh = get_latwt_tp(-ee,lat_hi)

tmp2[:,180,:] = np.nan
ind1 = np.argmax(lon>60)
ind2 = np.argmax(lon>280)

ind3 = np.argmax(lon>90)
ind4 = np.argmax(lon>130)

latmax1_nh[ind3:ind4] = (latmax1_nh[ind3:ind4]-np.mean(latmax1_nh[ind3:ind4]))*0.2+np
latmax2_nh[ind3:ind4] = (latmax2_nh[ind3:ind4]-np.mean(latmax2_nh[ind3:ind4]))*0.2+np
plt.plot(lon[ind1:ind2],np.array(latmax1_nh[ind1:ind2])+0.05,'k--',linewidth=2.5)
plt.plot(lon[ind1:ind2],np.array(latmax2_nh[ind1:ind2])-0.05,'m--',linewidth=2.5)
ind3 = np.argmax(lon>320)
ind4 = np.argmax(lon>350)
plt.plot(lon[ind3:ind4],latmax1_nh[ind3:ind4],'k--',linewidth=2.5)
plt.plot(lon[ind3:ind4],latmax2_nh[ind3:ind4],'m--',linewidth=2.5)
```

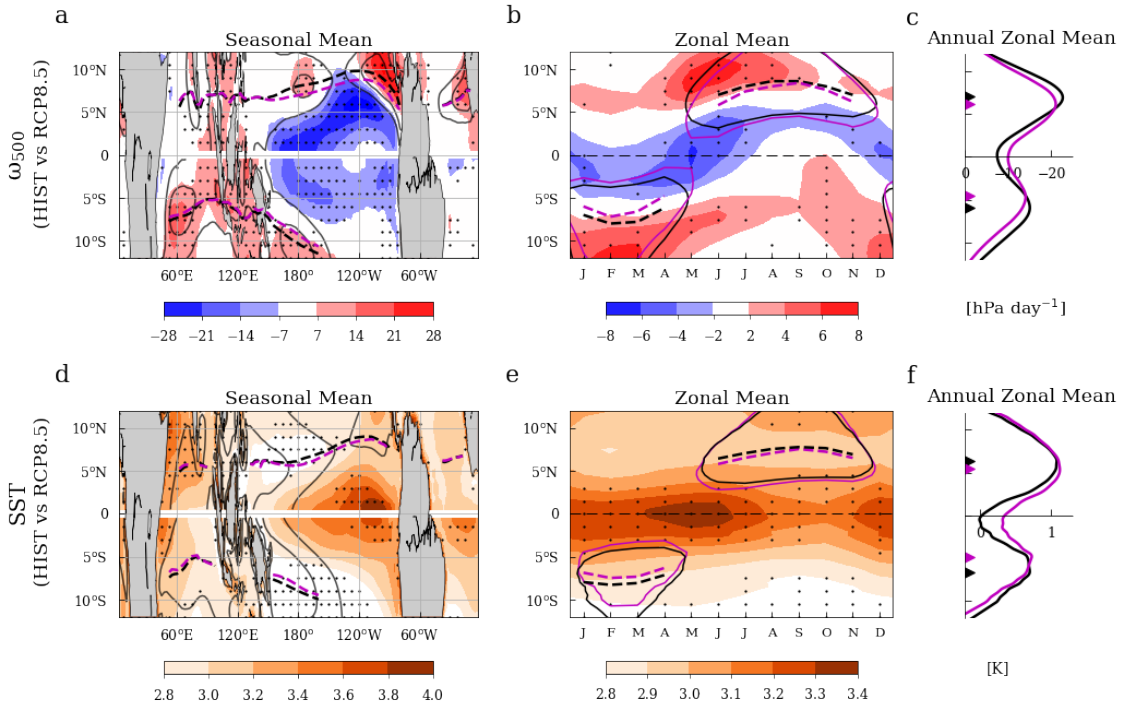
```
ind5 = np.argmax(lon>50)
ind6 = np.argmax(lon>200)
plt.plot(lon[ind5:ind6],latmax1_sh[ind5:ind6],'k--',linewidth=2.5)
plt.plot(lon[ind5:ind6],latmax2_sh[ind5:ind6],'m--',linewidth=2.5)
aa = tmp2-tmp1;
bb=np.sum(aa>0,0)
#CS1=maphi.contourf(x_hi,y_hi,np.mean(tmp2-tmp1,0),[-16,-12,-8,-4,4,8,12,16],cmap='bw
CS1=maphi.contourf(x_hi,y_hi,np.mean(tmp2-tmp1,0),np.arange(2.8,4.0,0.2),cmap='Oranges
maphi.contour(x_hi,y_hi,np.mean(tmp1,0),np.arange(301.2,303.5,1.0),colors='k',alpha=0

aa = tmp2-tmp1;

aa[np.abs(aa)>1000]=np.nan
aa[np.tile(np.isnan(maskhi_),[len(models_hist_),1,1])]=np.nan
cc = np.nanmean(np.nanmean(aa[:,181:200,:],2),1)
dd = np.nanmean(np.nanmean(aa[:,160:180,:],2),1)
ee = aa -0.
ee[:,181:,:]=cc[:,np.newaxis,np.newaxis]
ee[:,:180,:]=dd[:,np.newaxis,np.newaxis]
bb=np.sum(aa-ee>0,0)-9
bb[180,:]=0
critic=5

bb[180,:]=0
inv1 = 3; inv2=3
maphi.scatter(x_hi[::inv1,::inv2][np.abs(bb[::inv1,::inv2])>6],y_hi[::inv1,::inv2][np
# plt.plot(lon[ind1:ind2],latmax1_nh[ind1:ind2],'k-',linewidth=2.5)
# plt.plot(lon[ind3:ind4],latmax1_nh[ind3:ind4],'k-',linewidth=2.5)
# plt.plot(lon[ind5:ind6],latmax1_sh[ind5:ind6],'k-',linewidth=2.5)
cb=plt.colorbar(CS1,orientation='horizontal',shrink=0.75,extendfrac=0)
cb.ax.tick_params(labelsize=12.5);#cb.set_label('[ hPa day$^{-1}$ ]',fontsize=14,labe
ax.set_aspect('auto')
plt.tick_params(labelsize=12.5,direction='in',length=5)
plt.text(-0.18, 1.14,'d',fontsize=22,transform = ax.transAxes)
plt.title('Seasonal Mean',fontsize=17.5,loc='center')
#plt.text(-0.4,0.5,'RCP8.5 $\minus$ HIST',rotation=90,fontsize=22,horizontalalignment=
#plt.title(r'$\mathregular{\Delta \omega_{500}}$',fontsize=22,loc='left')
plt.grid(linewidth='0.8')
plt.text(-0.28, 0.56,'SST',rotation=90,fontsize=20,horizontalalignment='center',verti
plt.text(-0.21, 0.56,'(HIST vs RCP8.5)',rotation=90,fontsize=17.5,horizontalalignment=

ax=fig.add_subplot(spec[1,1])
time = np.arange(1,13,1)
# ts_hist[np.abs(ts_hist)>1000.]=np.nan
# ts_rcp85[np.abs(ts_rcp85)>1000.]=np.nan
ts_2d_hist = np.mean(np.nanmean(ts_hist*maskhi[None,None,:,:],3),0)
ts_2d_rcp = np.mean(np.nanmean(ts_rcp85*maskhi[None,None,:,:],3),0)
time=np.arange(0,14,1)
```

```
aa=ts_2d_rcp-ts_2d_hist
bb=get_extend(aa)
CS1=plt.contourf(time,lat_hi,bb.T,np.arange(2.8,3.45,0.1),cmap='Oranges')
cc = ts_2d_rcp - np.mean(ts_2d_rcp[:,142:221],1)[:,None]
plt.contour(time,lat_hi,get_extend(cc).T,[1.23],linewidths=1.5,colors='m')
dd = ts_2d_hist - np.mean(ts_2d_hist[:,142:221],1)[:,None]
plt.contour(time,lat_hi,get_extend(dd).T,[1.2],linewidths=1.5,colors='k')

[lat_omega11,lat_omega21]=get_latwt_tp(-(cc-0.03).T,lat_hi)
[lat_omega31,lat_omega41]=get_latwt_tp(-dd.T,lat_hi)

lat_omega11[8]=lat_omega11[8]+0.2
plt.plot(range(1,5),lat_omega21[0:4],'m--',linewidth=2.5)
plt.plot(range(6,12),np.array(lat_omega11[5:11])-0.3,'m--',linewidth=2.5)
plt.plot(range(1,5),lat_omega41[0:4],'k--',linewidth=2.5)
plt.plot(range(6,12),lat_omega31[5:11],'k--',linewidth=2.5)
plt.plot([0.5,12.5],[0,0],'k--',linewidth=1,dashes=(10, 5))
cc=np.nanmean(ts_rcp85*maskhi[None,None,:,:],3)-np.nanmean(ts_hist*maskhi[None,None,:
dd = np.nanmean(cc[:,:,140:221],2)
ee = cc -0.
ee[:,:,:]=dd[:,:,np.newaxis]
ff=np.sum(cc-ee>0,axis=0)-9
time1=np.arange(1,13,1)
critic = 6
lat2,time2 = np.meshgrid(lat_hi,time1)
inv1=1;inv2=3
plt.scatter(time2[::inv1,::inv2][np.abs(ff[::inv1,::inv2])>6],lat2[::inv1,::inv2][np.a
plt.yticks([-10,-5,0,5,10],['10$\mathrm{^o}$S','5$\mathrm{^o}$S','0','5$\mathrm{^o}$N
plt.tick_params(direction='in',labelsize=12.5,length=5)
plt.ylim([-12,12]);plt.xlim([0.5,12.5])
plt.xticks(np.arange(1,13,1),months,fontsize=11.8)
cb=plt.colorbar(CS1,orientation='horizontal',shrink=0.78,extendfrac=0.1)
cb.ax.tick_params(labelsize=12.5)

#cb1.set_label('[m s$^{-1}$]',fontsize=14)

plt.text(-0.2, 1.14,'e',fontsize=22,transform = ax.transAxes)
plt.title(r'Zonal Mean',loc='center',fontsize=17.5)
#plt.title(r'$\mathregular{\Delta \omega_{500}}$',loc='left',fontsize=22,pad=8)
# axins1 = inset_axes(ax,width="50%", height="5%",loc='lower right')
# cb1=plt.colorbar(CS1,orientation='horizontal',shrink=0.75, cax=axins1, orientation=
# cb1.ax.tick_params(labelsize=10)

ax10=fig.add_subplot(spec[1,2])
lat_hi1 = np.arange(-19,19.1,0.1)
f=interpolate.interp1d(lat_hi[142:221],np.nanmean(ts_2d_hist[:,142:221],0),'cubic')
lati1,lati2=get_annual_max2(np.nanmean(-ts_2d_hist,0))
CS=plt.plot(f(lat_hi1)-np.nanmean(f(lat_hi1)),lat_hi1,color='k',linestyle='-',linewid
```

```
        plt.plot(-0.2,lati1,'>',markersize=18,color='k');
        plt.plot(-0.2,lati2,'>',markersize=18,color='k');
        f=interpolate.interp1d(lat_hi[142:221],np.nanmean(ts_2d_rcp[:,142:221],0),'cubic')
        lati1,lati2=get_annual_max2(np.nanmean(-ts_2d_rcp,0))
        CS=plt.plot(f(lat_hi1)-np.nanmean(f(lat_hi1)),lat_hi1,'m-',linewidth=2.5)
        plt.plot(-0.2,lati1,'>',markersize=18,color='m');
        plt.plot(-0.2,lati2,'>',markersize=18,color='m');
        plt.ylim([-12,12]);plt.xlim([-0.2,1.3])
        plt.yticks([-10,-5,0,5,10],[]);#plt.xticks([-20,-10,0]);
        ax10.spines['bottom'].set_position('center')
        ax10.spines['right'].set_color('none')
        ax10.spines['top'].set_color('none')
        ax10.xaxis.set_ticks_position('bottom')
        ax10.yaxis.set_ticks_position('left')
        plt.tick_params(direction='in',labelsize=12.5,length=6)

        plt.text(-0.36,1.05,'Annual Zonal Mean',fontsize=17.5,transform = ax10.transAxes)
        plt.text(-0.55, 1.14,'f',fontsize=22,transform = ax10.transAxes)
        plt.text(0.2, -.266,'[K]',fontsize=14,transform = ax10.transAxes)
        #plt.gca().invert_xaxis()




        plt.subplots_adjust(hspace=0.22,wspace=0.35)

        box = ax09.get_position()
        ax09.set_position([box.x0-0.016, box.y0+0.1, box.width, box.height*0.7])

        box = ax10.get_position()
        ax10.set_position([box.x0-0.016, box.y0+0.1, box.width, box.height*0.7])
        plt.savefig('/Users/wenyuzhou/fig1.eps',bbox_inches='tight',format='eps', dpi=400)
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:429: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:435: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:36: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:42: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:66: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:67: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:85: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:139: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:425: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:431: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:149: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:164: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:174: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:203: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:210: RuntimeWarning: invalid value
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:217: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:243: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:244: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:263: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:267: RuntimeWarning: invalid value
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:292: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:297: RuntimeWarning: Mean of empty
```




In [45]:
```
models_hist_=['ACCESS1-0','ACCESS1-3','bcc-csm1-1','BNU-ESM','CanESM2','CCSM4','CESM1-
              'CSIRO-Mk3-6-0','GFDL-CM3','GISS-E2-R','HadGEM2-ES','inmcm4','IPSL-CM5A-LR','I
              'MIROC5','MIROC-ESM','MPI-ESM-LR','MRI-CGCM3','NorESM1-M']
lat_len = 121
lon_len = 144
lev_len = 12
omega_p500_hist_ = np.zeros((len(models_hist_),12,lat_len,lon_len))
ts_hist_ = np.zeros((len(models_hist_),12,lat_len,lon_len))

omega_p500_rcp85_ = np.zeros((len(models_hist_),12,lat_len,lon_len))
ts_rcp85_ = np.zeros((len(models_hist_),12,lat_len,lon_len))


for i in range(len(models_hist_)):
    model = models_hist_[i]
```

```
    os.chdir('/Volumes/Zhou/data/cmip5/historical/wap/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    omega_p500_hist_[i,::] = np.squeeze(f.variables['wap'][:,5,:,:]*864.)
    os.chdir('/Volumes/Zhou/data/cmip5/historical/ts/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    ts_hist_[i,::]= np.squeeze(f.variables['ts'][::])

    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap/'+model+'/r1i1p1/')
    filename = glob.glob('*2080*regrid.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    omega_p500_rcp85_[i,::] = np.squeeze(f.variables['wap'][:,5,:,:]*864.)
    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ts/'+model+'/r1i1p1/')
    filename = glob.glob('*regrid.nc')[0]
    f = netcdf.netcdf_file(filename, 'r')
    ts_rcp85_[i,::]= np.squeeze(f.variables['ts'][::])
```

In [71]:
```
fig=plt.figure(figsize=(12,15))


lat2,time2 = np.meshgrid(lat,time1)

models_amip=['bcc-csm1-1','CanAM4','CCSM4','CNRM-CM5','IPSL-CM5A-LR','MIROC5','MPI-ESM
omega_p500_2d_amip_mm=0
omega_p500_2d_amip4k_mm=0
omega_p500_2d_amipfuture_mm=0
lat1_omega_amip=np.zeros(4)
lat2_omega_amip=np.zeros(8)
lat1_omega_amip4k=np.zeros(4)
lat2_omega_amip4k=np.zeros(8)
lat1_omega_amipfuture=np.zeros(4)
lat2_omega_amipfuture=np.zeros(8)
lat1_omega_amip_mm=np.zeros(4)
lat2_omega_amip_mm=np.zeros(8)
lat1_omega_amip4k_mm=np.zeros(4)
lat2_omega_amip4k_mm=np.zeros(8)
lat1_omega_amipfuture_mm=np.zeros(4)
lat2_omega_amipfuture_mm=np.zeros(8)

omega_p500_2d_amip_ = np.zeros((len(models_amip),12,121))
omega_p500_2d_amip4k_ = np.zeros((len(models_amip),12,121))
omega_p500_2d_amipfuture_ = np.zeros((len(models_amip),12,121))
i=0
for model in models_amip:
    os.chdir('/Volumes/Zhou/data/cmip5/amip/wap/'+model+'/r1i1p1/')

    f = netcdf.netcdf_file('wap_'+model+'.month.regrid.2d.nc', 'r')
```

```
        lat = f.variables['lat'][:]
        omega_p500_2d_amip = np.squeeze(f.variables['wap'][:,5,:]*864.)
        omega_p500_2d_am_amip = np.mean(omega_p500_2d_amip,0)
        omega_p500_2d_amip_[i,::] = omega_p500_2d_amip
        os.chdir('/Volumes/Zhou/data/cmip5/amip4k/wap/'+model+'/r1i1p1/')
        f = netcdf.netcdf_file('wap_'+model+'.month.regrid.2d.nc', 'r')
        omega_p500_2d_amip4k = np.squeeze(f.variables['wap'][:,5,:]*864.)
        omega_p500_2d_amip4k_[i,::] = omega_p500_2d_amip4k
        omega_p500_2d_am_amip4k = np.mean(omega_p500_2d_amip4k,0)
        os.chdir('/Volumes/Zhou/data/cmip5/amipfuture/wap/'+model+'/r1i1p1/')
        f = netcdf.netcdf_file('wap_'+model+'.month.regrid.2d.nc', 'r')
        omega_p500_2d_amipfuture = np.squeeze(f.variables['wap'][:,5,:]*864.)

        omega_p500_2d_amipfuture_[i,::] = omega_p500_2d_amipfuture
        omega_p500_2d_am_amipfuture = np.mean(omega_p500_2d_amipfuture,0)

        time = np.arange(1,13,1)
        lat_hi= np.arange(-15,15.2,0.2)
        f=interpolate.interp2d(lat,time,omega_p500_2d_amip,'cubic')
        omega_p500_2d_amip_hi = f(lat_hi,time)
        f=interpolate.interp2d(lat,time,omega_p500_2d_amip4k,'cubic')
        omega_p500_2d_amip4k_hi = f(lat_hi,time)
        f=interpolate.interp2d(lat,time,omega_p500_2d_amipfuture,'cubic')
        omega_p500_2d_amipfuture_hi = f(lat_hi,time)


        [lat1_omega_amip,lat2_omega_amip]=get_min_loc1(omega_p500_2d_amip)
        [lat1_omega_amip4k,lat2_omega_amip4k]=get_min_loc1(omega_p500_2d_amip4k)
        [lat1_omega_amipfuture,lat2_omega_amipfuture]=get_min_loc1(omega_p500_2d_amipfutu



        lat1_omega_amip_mm=lat1_omega_amip_mm+lat1_omega_amip/np.size(models_amip)
        lat2_omega_amip_mm=lat2_omega_amip_mm+lat2_omega_amip/np.size(models_amip)
        lat1_omega_amip4k_mm=lat1_omega_amip4k_mm+lat1_omega_amip4k/np.size(models_amip)
        lat2_omega_amip4k_mm=lat2_omega_amip4k_mm+lat2_omega_amip4k/np.size(models_amip)
        lat1_omega_amipfuture_mm=lat1_omega_amipfuture_mm+lat1_omega_amipfuture/np.size(mo
        lat2_omega_amipfuture_mm=lat2_omega_amipfuture_mm+lat2_omega_amipfuture/np.size(mo



        omega_p500_2d_amip_mm=omega_p500_2d_amip_mm+omega_p500_2d_amip/len(models_amip)
        omega_p500_2d_amip4k_mm=omega_p500_2d_amip4k_mm+omega_p500_2d_amip4k/len(models_am
        omega_p500_2d_amipfuture_mm=omega_p500_2d_amipfuture_mm+omega_p500_2d_amipfuture/l
        i=i+1

lat2_omega_amip_mm[6]=6.2
lat2_omega_amip4k_mm[6]=6.1
```

```
time = np.arange(0,14,1)
ax04=plt.subplot(323)
aa=omega_p500_2d_amip4k_mm-omega_p500_2d_amip_mm
bb=get_extend(aa)
CS1=plt.contourf(time,lat,bb.T,[-8,-6,-4,-2,2,4,6,8],cmap='bwr',extend='both')
plt.contour(time,lat,-get_extend(omega_p500_2d_amip_mm).T,[22],linewidths=2,colors='k
plt.contour(time,lat,-get_extend(omega_p500_2d_amip4k_mm).T,[22],linewidths=2,colors=
cc=omega_p500_2d_amip4k_-omega_p500_2d_amip_
dd=np.sum(cc>0,axis=0)-4
time1=np.arange(1,13,1)
critic = 2

plt.scatter(time2[np.abs(dd)>critic],lat2[np.abs(dd)>critic],s=7,marker='.',c='k')
plt.plot(range(1,4),lat1_omega_amip_mm[0:-1],'k--',linewidth=2.5)
plt.plot(range(5,12),lat2_omega_amip_mm[0:-1],'k--',linewidth=2.5)
plt.plot(range(1,4),lat1_omega_amip4k_mm[0:-1],'m--',linewidth=2.5)
plt.plot(range(5,12),lat2_omega_amip4k_mm[0:-1],'m--',linewidth=2.5)
plt.ylim([-12,12])
# cb04=plt.colorbar(CS,extendfrac=0)
# cb04.ax.tick_params(labelsize=16)
# cb04.set_label('[hPa day$^{-1}$]',fontsize=15)
#plt.grid()
plt.plot([0.5,12.5],[0,0],'k--',linewidth=1,dashes=(10, 5))
#plt.ylabel('Latitude',fontsize=18)
plt.xticks(np.arange(1,13,1),months,fontsize=12)
plt.yticks([-10,-5,0,5,10],['10$\mathrm{^o}$S','5$\mathrm{^o}$S','0','5$\mathrm{^o}$N
plt.tick_params(labelsize=15,direction='in',length=5)
plt.title('AMIP4K vs AMIP',loc='right',fontsize=18)
plt.title(r'$\mathregular{\Delta \omega_{500}}$',loc='left',fontsize=22,pad=8)
plt.text(-0.25, 1.08,'c',fontsize=24,transform = ax04.transAxes)
plt.xlim(0.5,12.5)
ax.set_aspect('auto')

ax010=plt.subplot(324)
lat_hi = np.arange(-12,12.1,0.1)
f=interpolate.interp1d(lat,np.nanmean(omega_p500_2d_amip_mm,0),'cubic')
lati1,lati2=get_annual_max(np.nanmean(omega_p500_2d_amip_mm,0))
CS=plt.plot(f(lat_hi),lat_hi,color='k',linestyle='-',linewidth=2.5)
plt.plot(0,lati1,'>',markersize=18,color='k');
plt.plot(0,lati2,'>',markersize=18,color='k');
f=interpolate.interp1d(lat,np.nanmean(omega_p500_2d_amip4k_mm,0),'cubic')
lati1,lati2=get_annual_max(np.nanmean(omega_p500_2d_amip4k_mm,0))
CS=plt.plot(f(lat_hi),lat_hi,'m-',linewidth=2.5)
plt.plot(0,lati1,'>',markersize=18,color='m');
plt.plot(0,lati2,'>',markersize=18,color='m');
plt.ylim([-12,12]);plt.xlim([-25,0])
plt.xticks([-20,-10,0]);plt.yticks([-10,-5,0,5,10],[])
ax010.spines['bottom'].set_position('center')
```

```
ax010.spines['right'].set_color('none')
ax010.spines['top'].set_color('none')
ax010.xaxis.set_ticks_position('bottom')
ax010.yaxis.set_ticks_position('left')
plt.tick_params(direction='in',labelsize=17,length=6)
plt.gca().invert_xaxis()
plt.text(-0.3, 1.08,'d',fontsize=24,transform = ax010.transAxes)

ax03=plt.subplot(321)
aa=omega_p500_2d_amipfuture_mm-omega_p500_2d_amip_mm
bb=get_extend(aa)
CS=plt.contourf(time,lat,bb.T,[-8,-6,-4,-2,2,4,6,8],cmap='bwr',extend='both')
plt.contour(time,lat,-get_extend(omega_p500_2d_amip_mm).T,[22],linewidths=1.5,colors=
plt.contour(time,lat,-get_extend(omega_p500_2d_amipfuture_mm).T,[22],linewidths=1.5,c
cc=omega_p500_2d_amipfuture_-omega_p500_2d_amip_
dd=np.sum(cc>0,axis=0)-4
time1=np.arange(1,13,1)
critic = 2
plt.scatter(time2[np.abs(dd)>critic],lat2[np.abs(dd)>critic],s=7,marker='.',c='k')

plt.plot(range(1,4),lat1_omega_amip_mm[0:-1],'k--',linewidth=2.5)
plt.plot(range(5,12),lat2_omega_amip_mm[0:-1],'k--',linewidth=2.5)
plt.plot(range(1,4),lat1_omega_amipfuture_mm[0:-1],'m--',linewidth=2.5)
plt.plot(range(5,12),lat2_omega_amipfuture_mm[0:-1],'m--',linewidth=2.5)
plt.ylim([-12,12])
# cb03=plt.colorbar(CS,extendfrac=0)
# cb03.ax.tick_params(labelsize=16)
# cb03.set_label('[hPa day$^{-1}$]',fontsize=15)
# #plt.grid()
plt.xticks(np.arange(1,13,1),months,fontsize=12)
plt.yticks([-10,-5,0,5,10],['10$\mathrm{^o}$S','5$\mathrm{^o}$S','0','5$\mathrm{^o}$N
plt.tick_params(labelsize=15,direction='in',length=5)
plt.title('AMIPFuture vs AMIP',loc='right',fontsize=19)
plt.title(r'$\mathregular{\Delta \omega_{500}}$',loc='left',fontsize=20,pad=8)
plt.text(-0.25, 1.08,'a',fontsize=24,transform = ax03.transAxes)
plt.xlim(0.5,12.5)
#plt.text(-0.2, 0.58,'$\mathregular{\omega_{500}}$',rotation=90,fontsize=22,transform
plt.plot([0.5,12.5],[0,0],'k--',linewidth=1,dashes=(10, 5))

ax009=plt.subplot(322)
lat_hi = np.arange(-12,12.1,0.1)
f=interpolate.interp1d(lat,np.nanmean(omega_p500_2d_amip_mm,0),'cubic')
lati1,lati2=get_annual_max(np.nanmean(omega_p500_2d_amip_mm,0))
CS=plt.plot(f(lat_hi),lat_hi,color='k',linestyle='-',linewidth=2.5)
plt.plot(0,lati1-0.45,'>',markersize=18,color='k');
plt.plot(0,lati2,'>',markersize=18,color='k');
f=interpolate.interp1d(lat,np.nanmean(omega_p500_2d_amipfuture_mm,0),'cubic')
lati1,lati2=get_annual_max(np.nanmean(omega_p500_2d_amipfuture_mm,0))
```

```
CS=plt.plot(f(lat_hi),lat_hi,'m-',linewidth=2.5)
plt.plot(0,lati1,'>',markersize=18,color='m');
plt.plot(0,lati2,'>',markersize=18,color='m');
plt.ylim([-12,12]);plt.xlim([-25,0])
plt.xticks([-20,-10,0]);plt.yticks([-10,-5,0,5,10],[])
ax009.spines['bottom'].set_position('center')
ax009.spines['right'].set_color('none')
ax009.spines['top'].set_color('none')
ax009.xaxis.set_ticks_position('bottom')
ax009.yaxis.set_ticks_position('left')
plt.tick_params(direction='in',labelsize=17,length=6)
plt.gca().invert_xaxis()
plt.text(0.1,-0.31,'[hPa day$^{-1}$]',fontsize=15.5,transform = ax009.transAxes)
plt.text(-0.3, 1.08,'b',fontsize=24,transform = ax009.transAxes)


models_hist=['ACCESS1-0','ACCESS1-3','bcc-csm1-1','BNU-ESM','CanESM2','CCSM4','CESM1-C
        'CSIRO-Mk3-6-0','GFDL-CM3','GISS-E2-R','HadGEM2-ES','inmcm4','IPSL-CM5A-LR','I
        'MIROC5','MIROC-ESM','MPI-ESM-LR','MRI-CGCM3','NorESM1-M']

lat_omega_1=np.zeros(5)
lat_omega_2=np.zeros(8)
lat_omega_3=np.zeros(5)
lat_omega_4=np.zeros(8)

lat_omega_1_all=np.zeros((len(models_hist),5))
lat_omega_2_all=np.zeros((len(models_hist),8))
lat_omega_3_all=np.zeros((len(models_hist),5))
lat_omega_4_all=np.zeros((len(models_hist),8))

lat_hist_djf = np.zeros(np.size(models_hist))
lat_rcp_djf = np.zeros(np.size(models_hist))
lat_hist_jja = np.zeros(np.size(models_hist))
lat_rcp_jja = np.zeros(np.size(models_hist))

srang_hist = np.zeros(np.size(models_hist))
srang_rcp = np.zeros(np.size(models_hist))
srang_diff = np.zeros(np.size(models_hist))
srang1_hist = np.zeros(np.size(models_hist))
srang1_rcp = np.zeros(np.size(models_hist))
srang1_diff = np.zeros(np.size(models_hist))
jj_lat_diff = np.zeros(np.size(models_hist))
arang_hist = np.zeros(np.size(models_hist))
arang_rcp = np.zeros(np.size(models_hist))
arang_diff = np.zeros(np.size(models_hist))
dtp_warm_hist = np.zeros(np.size(models_hist))
dtp_warm_diff = np.zeros(np.size(models_hist))
dtp_warm_diff_mjj = np.zeros(np.size(models_hist))
```

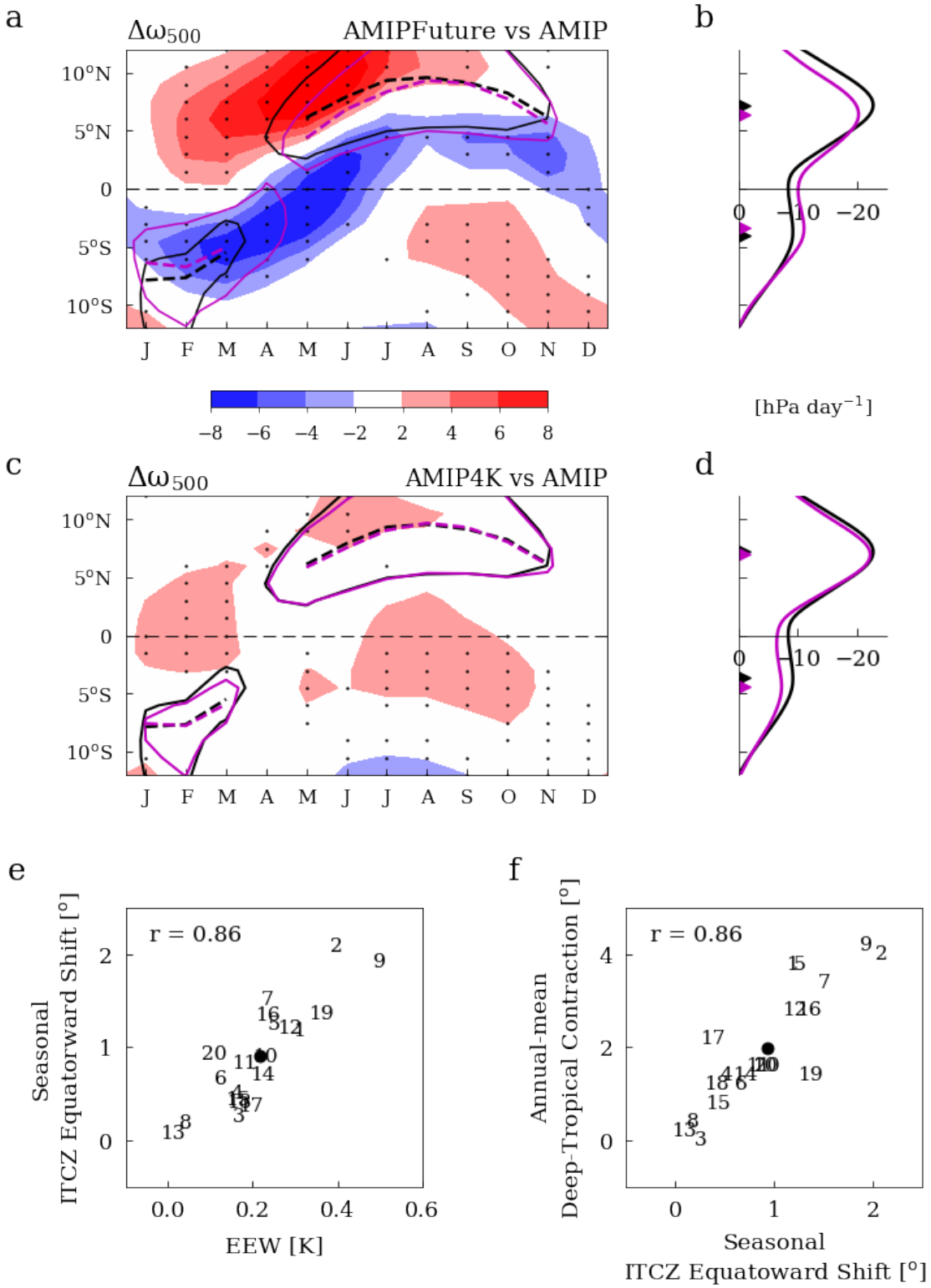
```
dtp_warm_diff_mjj_n = np.zeros(np.size(models_hist))
dtp_warm_diff_n = np.zeros((len(models_hist),12))
dtp_warm_diff_mjj_norm = np.zeros(np.size(models_hist))
dtp_warm_diff_w = np.zeros(np.size(models_hist))
dtp_warm_diff_amj_s = np.zeros(np.size(models_hist))
dnino_warm_diff = np.zeros(np.size(models_hist))

itcz_strength_hist = np.zeros(np.size(models_hist))
itcz_strength_rcp = np.zeros(np.size(models_hist))
itcz_strength_am_hist = np.zeros(np.size(models_hist))
itcz_strength_am_rcp = np.zeros(np.size(models_hist))

tp_warm = np.zeros(np.size(models_hist))
gb_warm_n = np.zeros(np.size(models_hist))
gb_warm_s = np.zeros(np.size(models_hist))
i=0
for model in models_hist:
    os.chdir('/Volumes/Zhou/data/cmip5/historical/wap/'+model+'/r1i1p1/')

    f = netcdf.netcdf_file('wap_'+model+'.month.regrid.2d.nc', 'r')
    lat = f.variables['lat'][:]
    omega_p500_2d_hist = np.squeeze(f.variables['wap'][:,5,:]*864.)
    omega_p500_2d_am_hist = np.mean(omega_p500_2d_hist,0)
    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/wap/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('wap_Amon_'+model+'_rcp85_r1i1p1_208001-210012.month.regri
    omega_p500_2d_rcp = np.squeeze(f.variables['wap'][:,5,:]*864.)
    omega_p500_2d_am_rcp = np.mean(omega_p500_2d_rcp,0)


    os.chdir('/Volumes/Zhou/data/cmip5/historical/ts/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('ts_'+model+'.month.regrid.nc', 'r')

    lon = f.variables['lon'][:]
    ts_hist = f.variables['ts'][::]
    ts_hist_nino = np.mean(ts_hist[5:7,57:64,64:100])
    ts_2d_hist = np.nanmean(ts_hist*mask[None,:,:],2)
    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/ts/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('ts_Amon_'+model+'_rcp85_r1i1p1_208001-210012.month.regrid
    ts_rcp = f.variables['ts'][::]
    ts_rcp_nino = np.mean(ts_rcp[5:7,57:64,64:100])
    ts_2d_rcp = np.nanmean(ts_rcp*mask[None,:,:],2)
    aa=ts_2d_hist.T-np.nanmean(ts_2d_hist[:,47:74],1)[None,:]
    aa1=ts_2d_hist.T-np.nanmean(ts_2d_hist[:,65:74],1)[None,:]
    dtp_warm_hist[i]=np.mean(aa[57:64,:]);
    bb=ts_2d_rcp.T-np.nanmean(ts_2d_rcp[:,47:74],1)[None,:]
    bb1=ts_2d_rcp.T-np.nanmean(ts_2d_rcp[:,65:74],1)[None,:]
    cc=bb-aa;
    cc1=bb1-aa1;
```

```
tp_warm[i] = np.mean(np.nanmean(ts_2d_rcp[:,47:74],1)-np.nanmean(ts_2d_hist[:,47:
gb_warm_n[i] = np.mean(np.nanmean(ts_2d_rcp[:,74:108],1)-np.nanmean(ts_2d_hist[:,
gb_warm_s[i] = np.mean(np.nanmean(ts_2d_rcp[:,12:55],1)-np.nanmean(ts_2d_hist[:,1
dtp_warm_diff[i] = np.mean(cc[57:64,:]);
dtp_warm_diff_mjj[i] = np.mean(cc[57:64,5:7]);
dtp_warm_diff_mjj_n[i] = np.mean(cc1[57:64,5:7]);
dtp_warm_diff_n[i,:] = np.mean(cc1[57:64,:],0);
dtp_warm_diff_mjj_norm[i] = np.mean(cc[57:64,5:7])/tp_warm[i]
dtp_warm_diff_amj_s[i] = np.mean(cc[57:64,5:7]);
dtp_warm_diff_w[i] = np.mean((cc[57:64,11]+cc[57:64,0]+cc[57:64,1])/3.);
dnino_warm_diff[i] = ts_rcp_nino - ts_hist_nino - tp_warm[i]
[lat1,lat2]=get_min_loc(omega_p500_2d_hist)
[lat_hist_djf[i],lat_hist_jja[i]]=get_min_loc_season(omega_p500_2d_hist)
srang_hist[i] = max(lat2)-min(lat1)
srang1_hist[i] = np.std(np.array(list(lat1)+list(lat2)))
[lat3,lat4]=get_min_loc(omega_p500_2d_rcp)
[lat_rcp_djf[i],lat_rcp_jja[i]]=get_min_loc_season(omega_p500_2d_rcp)
srang_rcp[i] = max(lat4)-min(lat3)
srang1_rcp[i] = np.std(np.array(list(lat3)+list(lat4)))

srang_diff[i] = srang_rcp[i] - srang_hist[i]
srang1_diff[i] = srang1_rcp[i] - srang1_hist[i]
jj_lat_diff[i]=np.mean(lat4[1:3])-np.mean(lat2[1:3])

arang_hist[i] = get_annual_range(omega_p500_2d_am_hist)
arang_rcp[i] = get_annual_range(omega_p500_2d_am_rcp)
arang_diff[i] = arang_rcp[i] - arang_hist[i]

time = np.arange(1,13,1)
lat_hi= np.arange(-15,15.2,0.2)
f=interpolate.interp2d(lat,time,omega_p500_2d_hist,'cubic')
omega_p500_2d_hist_hi = f(lat_hi,time)
itcz_strength_hist[i]=np.mean(np.min(omega_p500_2d_hist_hi,0))
f=interpolate.interp2d(lat,time,omega_p500_2d_rcp,'cubic')
omega_p500_2d_rcp_hi = f(lat_hi,time)
itcz_strength_rcp[i]=np.mean(np.min(omega_p500_2d_rcp_hi,0))


lat_nh_hi= np.arange(0,15.2,0.2)
lat_sh_hi= np.arange(-15,0,0.2)
f=interpolate.interp1d(lat,omega_p500_2d_am_hist,'cubic')
omega_p500_2d_hist_am_nh_hi = f(lat_nh_hi)
omega_p500_2d_hist_am_sh_hi = f(lat_sh_hi)
itcz_strength_am_hist[i]=(np.min(omega_p500_2d_hist_am_nh_hi)+np.min(omega_p500_2
f=interpolate.interp1d(lat,omega_p500_2d_am_rcp,'cubic')
omega_p500_2d_rcp_am_nh_hi = f(lat_nh_hi)
omega_p500_2d_rcp_am_sh_hi = f(lat_sh_hi)
itcz_strength_am_rcp[i]=(np.min(omega_p500_2d_rcp_am_nh_hi)+np.min(omega_p500_2d_
```

```
    lat_omega_1=lat_omega_1+lat1/np.size(models_hist)
    lat_omega_2=lat_omega_2+lat2/np.size(models_hist)
    lat_omega_3=lat_omega_3+lat3/np.size(models_hist)
    lat_omega_4=lat_omega_4+lat4/np.size(models_hist)
    lat_omega_1_all[i,:]=lat1
    lat_omega_2_all[i,:]=lat2
    lat_omega_3_all[i,:]=lat3
    lat_omega_4_all[i,:]=lat4


    i=i+1

itcz_strength_diff = (itcz_strength_rcp - itcz_strength_hist)/itcz_strength_hist
itcz_strength_am_diff = (itcz_strength_am_rcp - itcz_strength_am_hist)/itcz_strength_a

ax02=plt.subplot(326)
for i in range(np.size(models_hist)):
    exec('ax'+str(i)+'=plt.text(-srang1_diff[i],-arang_diff[i],str(i+1),horizontalalig
    #exec('ax'+str(i)+'=plt.text(dtp_warm_diff[i],-arang_diff[i],str(i+1),horizontala
ax20 = plt.scatter(np.mean(-srang1_diff[i]),np.mean(-arang_diff),s=80,c='k',marker='o
#ax21 = plt.plot(np.mean(-srang1_4kdiff),np.mean(-arang_4kdiff)+0.2,ms=10,c='b',marke
#ax22 = plt.plot(np.mean(-srang1_futurediff)+0.2,np.mean(-arang_futurediff),ms=10,c='
plt.xlim(-0.5,2.5)
plt.ylim(-1,5)
plt.yticks([0,2,4])
a=pearsonr(-srang1_diff,-arang_diff)[0]
plt.text(0.08,0.88,'r = '+str(np.floor(a*100)/100),fontsize=17,transform=ax02.transAx
plt.xlabel('Seasonal\nITCZ Equatoward Shift [$\mathrm{^o}$]',fontsize=17.5,labelpad=1
plt.ylabel('Annual-mean\nDeep-Tropical Contraction [$\mathrm{^o}$]',fontsize=17.5,labe
plt.tick_params(labelsize=17,direction='in',length=5)
plt.text(-0.4, 1.1,'f',fontsize=24,transform = ax02.transAxes)

ax01=plt.subplot(325)
for i in range(np.size(models_hist)):
    #exec('ax'+str(i)+'=plt.scatter(dtp_warm_diff[i],-srang1_diff[i],s=100,c=\'dimgre


    exec('ax'+str(i)+'=plt.text(dtp_warm_diff[i],-srang1_diff[i],str(i+1),horizontalal
ax20 = plt.scatter(np.mean(dtp_warm_diff),np.mean(-srang1_diff),s=80,c='k',marker='o'
#ax21 = plt.scatter(0,-np.mean(srang1_4kdiff),s=100,c='b',marker='o')
#ax22 = plt.scatter(0.39,-np.mean(srang1_futurediff)+0.2,s=100,c='r',marker='o')
plt.ylim(-0.5,2.5)
plt.xlim(-0.1,0.6)
a=pearsonr(dtp_warm_diff,-srang1_diff)[0]
#plt.legend([ax19,ax20,ax21,ax22],['CMIP5 Individuals','CMIP5 MME','AMIP 4K','AMIP Fu
#           handletextpad=0.0,frameon=False,loc='lower right',edgecolor='inherit',sca
```

```
        plt.text(0.08,0.88,'r = '+str(np.floor(a*100)/100),fontsize=17,transform=ax01.transAx
        plt.xlabel('EEW [K]',fontsize=17.5,labelpad=10)
        plt.ylabel('Seasonal\nITCZ Equatorward Shift [$\mathrm{^o}$]',fontsize=17.5,labelpad=
        plt.text(-0.4, 1.1,'e',fontsize=24,transform = ax01.transAxes)
        plt.yticks([0,1,2])
        plt.tick_params(labelsize=17,direction='in',length=5)

        cax = fig.add_axes([0.2, 0.665, 0.3, 0.012])
        cb1=plt.colorbar(CS1, cax=cax, orientation='horizontal',extendfrac=0)
        cb1.ax.tick_params(labelsize=14)
        #cb1.set_label('[hPa day$^{-1}$]',fontsize=15)

        fig.subplots_adjust(wspace=0.35,hspace=0.4)

        box = ax01.get_position()
        ax01.set_position([box.x0, box.y0-0.015, box.width*0.8, box.height])

        box = ax02.get_position()
        ax02.set_position([box.x0, box.y0-0.015, box.width*0.8, box.height])

        box = ax03.get_position()
        ax03.set_position([box.x0, box.y0+0.04, box.width*1.3, box.height])

        box = ax04.get_position()
        ax04.set_position([box.x0, box.y0, box.width*1.3, box.height])


        box = ax009.get_position()
        ax009.set_position([box.x0+0.1, box.y0+0.04, box.width*0.4, box.height])
        box = ax010.get_position()
        ax010.set_position([box.x0+0.1, box.y0, box.width*0.4, box.height])


        plt.savefig('/Users/wenyuzhou/fig2.eps', bbox_inches='tight',format='eps', dpi=400)
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:258: RuntimeWarning: Mean of empty
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:263: RuntimeWarning: Mean of empty
```

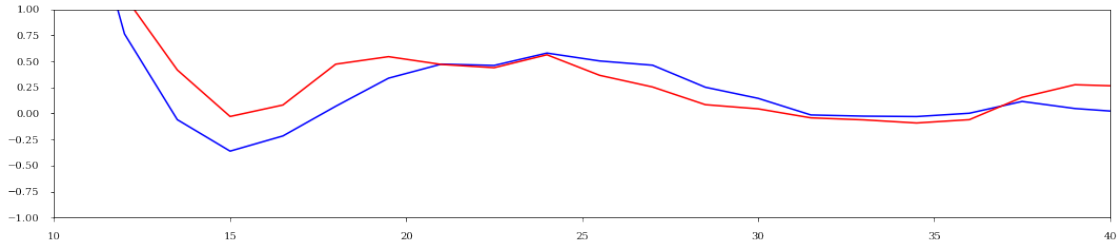



```
In [41]: Lv = 2.5*(10**6);
```

```
mpsi_hist = 0
mpsi_rcp = 0
hb_rcp_n_all = np.zeros((len(models_hist),12))
hb_rcp_s_all = np.zeros((len(models_hist),12))
hb_hist_n_all = np.zeros((len(models_hist),12))
hb_hist_s_all = np.zeros((len(models_hist),12))
pe_hist = 0
pe_rcp = 0
pe_rcp_n_all = np.zeros((len(models_hist),12))
pe_rcp_s_all = np.zeros((len(models_hist),12))
pe_hist_n_all = np.zeros((len(models_hist),12))
pe_hist_s_all = np.zeros((len(models_hist),12))


hb_ann_rcp_n_all = np.zeros(len(models_hist))
hb_ann_rcp_s_all = np.zeros(len(models_hist))
hb_ann_hist_n_all = np.zeros(len(models_hist))
hb_ann_hist_s_all = np.zeros(len(models_hist))


i=0
for model in models_hist:
    os.chdir('/Volumes/Zhou/data/cmip5/historical/mpsi/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('mpsi_'+model+'.month.regrid.nc', 'r')
    lat = f.variables['lat'][:]
    plev = f.variables['plev'][:]
    mpsi_hist_ = f.variables['mpsi'][:,:,:]#time,plev,lat
    mpsi_hist = mpsi_hist + mpsi_hist_/len(models_hist)
    os.chdir('/Volumes/Zhou/data/cmip5/rcp85/mpsi/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('mpsi_Amon_'+model+'_rcp85_r1i1p1_208001-210012.month.regri
    lat = f.variables['lat'][:]
    plev = f.variables['plev'][:]
    mpsi_rcp_ = f.variables['mpsi'][:,:,:]
    mpsi_rcp = mpsi_rcp + mpsi_rcp_/len(models_hist)

    [hb_hist_s_,hb_hist_n_] = get_hadley_boundary(mpsi_hist_,lat)
    [hb_rcp_s_,hb_rcp_n_] = get_hadley_boundary(mpsi_rcp_,lat)
    [hb_ann_hist_s_,hb_ann_hist_n_] = get_annual_hadley_boundary(np.mean(mpsi_hist_,0)
    [hb_ann_rcp_s_,hb_ann_rcp_n_] = get_annual_hadley_boundary(np.mean(mpsi_rcp_,0),la


    hb_hist_n_all[i,:] = hb_hist_n_
    hb_hist_s_all[i,:] = hb_hist_s_
    hb_rcp_n_all[i,:] = hb_rcp_n_
    hb_rcp_s_all[i,:] = hb_rcp_s_


    hb_ann_hist_n_all[i] = hb_ann_hist_n_
```

```
            hb_ann_hist_s_all[i] = hb_ann_hist_s_
            hb_ann_rcp_n_all[i] = hb_ann_rcp_n_
            hb_ann_rcp_s_all[i] = hb_ann_rcp_s_

            os.chdir('/Volumes/Zhou/data/cmip5/historical/pr/'+model+'/r1i1p1/')
            f = netcdf.netcdf_file('pr_'+model+'.month.regrid.nc', 'r')
            pr_hist_ = np.squeeze(np.mean(f.variables['pr'][:,:,:],2))
            os.chdir('/Volumes/Zhou/data/cmip5/historical/hfls/'+model+'/r1i1p1/')
            f = netcdf.netcdf_file('hfls_'+model+'.month.regrid.nc', 'r')
            evap_hist_ = np.squeeze(np.mean(f.variables['hfls'][:,:,:],2))/Lv
            pe_hist_ = pr_hist_ - evap_hist_
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/pr/'+model+'/r1i1p1/')
            f = netcdf.netcdf_file('pr_Amon_'+model+'_rcp85_r1i1p1_208001-210012.month.regrid
            pr_rcp_ = np.squeeze(np.mean(f.variables['pr'][:,:,:],2))
            os.chdir('/Volumes/Zhou/data/cmip5/rcp85/hfls/'+model+'/r1i1p1/')
            f = netcdf.netcdf_file('hfls_Amon_'+model+'_rcp85_r1i1p1_208001-210012.month.regr
            evap_rcp_ = np.squeeze(np.mean(f.variables['hfls'][:,:,:],2))/Lv
            pe_rcp_ = pr_rcp_ - evap_rcp_

            pe_hist = pe_hist + pe_hist_/len(models_hist)
            pe_rcp = pe_rcp + pe_rcp_/len(models_hist)
            [pe_hist_s_,pe_hist_n_] = get_pe_boundary(pe_hist_,lat)
            [pe_rcp_s_,pe_rcp_n_] = get_pe_boundary(pe_rcp_,lat)

            pe_hist_n_all[i,:] = pe_hist_n_
            pe_rcp_n_all[i,:] = pe_rcp_n_
            pe_hist_s_all[i,:] = pe_hist_s_
            pe_rcp_s_all[i,:] = pe_rcp_s_

            i=i+1
        [hb_hist_s,hb_hist_n] = get_hadley_boundary(mpsi_hist,lat)
        [hb_rcp_s,hb_rcp_n] = get_hadley_boundary(mpsi_rcp,lat)


        [pe_hist_s,pe_hist_n] = get_pe_boundary(pe_hist,lat)
        [pe_rcp_s,pe_rcp_n] = get_pe_boundary(pe_rcp,lat)
```

```
34.79999999999995
-38.0
29.799999999999965
35.39999999999995
```




In [38]:
```
models_amip=['IPSL-CM5A-LR','MRI-CGCM3','bcc-csm1-1','CanAM4','CCSM4','CNRM-CM5','MIR(
mpsi_amip = 0
mpsi_amip4k = 0
mpsi_amip4xco2 = 0
mpsi_amipfuture = 0
hb_ann_amip_s_all = np.zeros(len(models_amip))
hb_ann_amip_n_all = np.zeros(len(models_amip))
hb_ann_amip4k_s_all = np.zeros(len(models_amip))
hb_ann_amip4k_n_all = np.zeros(len(models_amip))
hb_ann_amip4xco2_s_all = np.zeros(len(models_amip))
hb_ann_amip4xco2_n_all = np.zeros(len(models_amip))
hb_ann_amipfuture_s_all = np.zeros(len(models_amip))
hb_ann_amipfuture_n_all = np.zeros(len(models_amip))



hb_amip_n_all = np.zeros((len(models_amip),12))
hb_amip_s_all = np.zeros((len(models_amip),12))
hb_amip4k_n_all = np.zeros((len(models_amip),12))
hb_amip4k_s_all = np.zeros((len(models_amip),12))
hb_amip4xco2_n_all = np.zeros((len(models_amip),12))
hb_amip4xco2_s_all = np.zeros((len(models_amip),12))
hb_amipfuture_n_all = np.zeros((len(models_amip),12))
hb_amipfuture_s_all = np.zeros((len(models_amip),12))


pe_amip_n_all = np.zeros((len(models_amip),12))
pe_amip_s_all = np.zeros((len(models_amip),12))
pe_amip4k_n_all = np.zeros((len(models_amip),12))
pe_amip4k_s_all = np.zeros((len(models_amip),12))
pe_amip4xco2_n_all = np.zeros((len(models_amip),12))
pe_amip4xco2_s_all = np.zeros((len(models_amip),12))
pe_amipfuture_n_all = np.zeros((len(models_amip),12))
pe_amipfuture_s_all = np.zeros((len(models_amip),12))


plt.figure()
```

```
i=0
for model in models_amip:
    os.chdir('/Volumes/Zhou/data/cmip5/amip/mpsi/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('mpsi_'+model+'.month.regrid.nc', 'r')
    lat = f.variables['lat'][:]
    plev = f.variables['plev'][:]
    mpsi_amip_ = f.variables['mpsi'][:,:,:]#time,plev,lat
    os.chdir('/Volumes/Zhou/data/cmip5/amip4k/mpsi/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('mpsi_'+model+'.month.regrid.nc', 'r')
    mpsi_amip4k_ = f.variables['mpsi'][:,:,:]#time,plev,lat
    os.chdir('/Volumes/Zhou/data/cmip5/amip4xco2/mpsi/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('mpsi_'+model+'.month.regrid.nc', 'r')
    mpsi_amip4xco2_ = f.variables['mpsi'][:,:,:]#time,plev,lat
    os.chdir('/Volumes/Zhou/data/cmip5/amipfuture/mpsi/'+model+'/r1i1p1/')
    f = netcdf.netcdf_file('mpsi_'+model+'.month.regrid.nc', 'r')
    mpsi_amipfuture_ = f.variables['mpsi'][:,:,:]#time,plev,lat

    mpsi_amip = mpsi_amip + mpsi_amip_/len(models_amip)
    mpsi_amip4k = mpsi_amip4k + mpsi_amip4k_/len(models_amip)
    mpsi_amip4xco2 = mpsi_amip4xco2 + mpsi_amip4xco2_/len(models_amip)
    mpsi_amipfuture = mpsi_amipfuture + mpsi_amipfuture_/len(models_amip)
    [hb_amip_s_,hb_amip_n_] = get_hadley_boundary(mpsi_amip_,lat)
    [hb_amip4k_s_,hb_amip4k_n_] = get_hadley_boundary(mpsi_amip4k_,lat)
    [hb_amip4xco2_s_,hb_amip4xco2_n_] = get_hadley_boundary(mpsi_amip4xco2_,lat)
    [hb_amipfuture_s_,hb_amipfuture_n_] = get_hadley_boundary(mpsi_amipfuture_,lat)

    [hb_ann_amip_s_,hb_ann_amip_n_] = get_annual_hadley_boundary(np.mean(mpsi_amip_,0
    [hb_ann_amip4k_s_,hb_ann_amip4k_n_] = get_annual_hadley_boundary(np.mean(mpsi_ami
    [hb_ann_amip4xco2_s_,hb_ann_amip4xco2_n_] = get_annual_hadley_boundary(np.mean(mp
    [hb_ann_amipfuture_s_,hb_ann_amipfuture_n_] = get_annual_hadley_boundary(np.mean(

    hb_amip_n_all[i,:] = hb_amip_n_
    hb_amip_s_all[i,:] = hb_amip_s_
    hb_amip4k_n_all[i,:] = hb_amip4k_n_
    hb_amip4k_s_all[i,:] = hb_amip4k_s_
    hb_amip4xco2_n_all[i,:] = hb_amip4xco2_n_
    hb_amip4xco2_s_all[i,:] = hb_amip4xco2_s_
    hb_amipfuture_n_all[i,:] = hb_amipfuture_n_
    hb_amipfuture_s_all[i,:] = hb_amipfuture_s_

    hb_ann_amip_n_all[i] = hb_ann_amip_n_
    hb_ann_amip_s_all[i] = hb_ann_amip_s_
    hb_ann_amip4k_n_all[i] = hb_ann_amip4k_n_
    hb_ann_amip4k_s_all[i] = hb_ann_amip4k_s_
    hb_ann_amip4xco2_n_all[i] = hb_ann_amip4xco2_n_
    hb_ann_amip4xco2_s_all[i] = hb_ann_amip4xco2_s_
    hb_ann_amipfuture_n_all[i] = hb_ann_amipfuture_n_
    hb_ann_amipfuture_s_all[i] = hb_ann_amipfuture_s_
```

```
            i=i+1

        [hb_amip_s,hb_amip_n] = get_hadley_boundary(mpsi_amip,lat)
        [hb_amip4k_s,hb_amip4k_n] = get_hadley_boundary(mpsi_amip4k,lat)
        [hb_amip4xco2_s,hb_amip4xco2_n] = get_hadley_boundary(mpsi_amip4xco2,lat)
        [hb_amipfuture_s,hb_amipfuture_n] = get_hadley_boundary(mpsi_amipfuture,lat)
```

```
<Figure size 432x288 with 0 Axes>
```

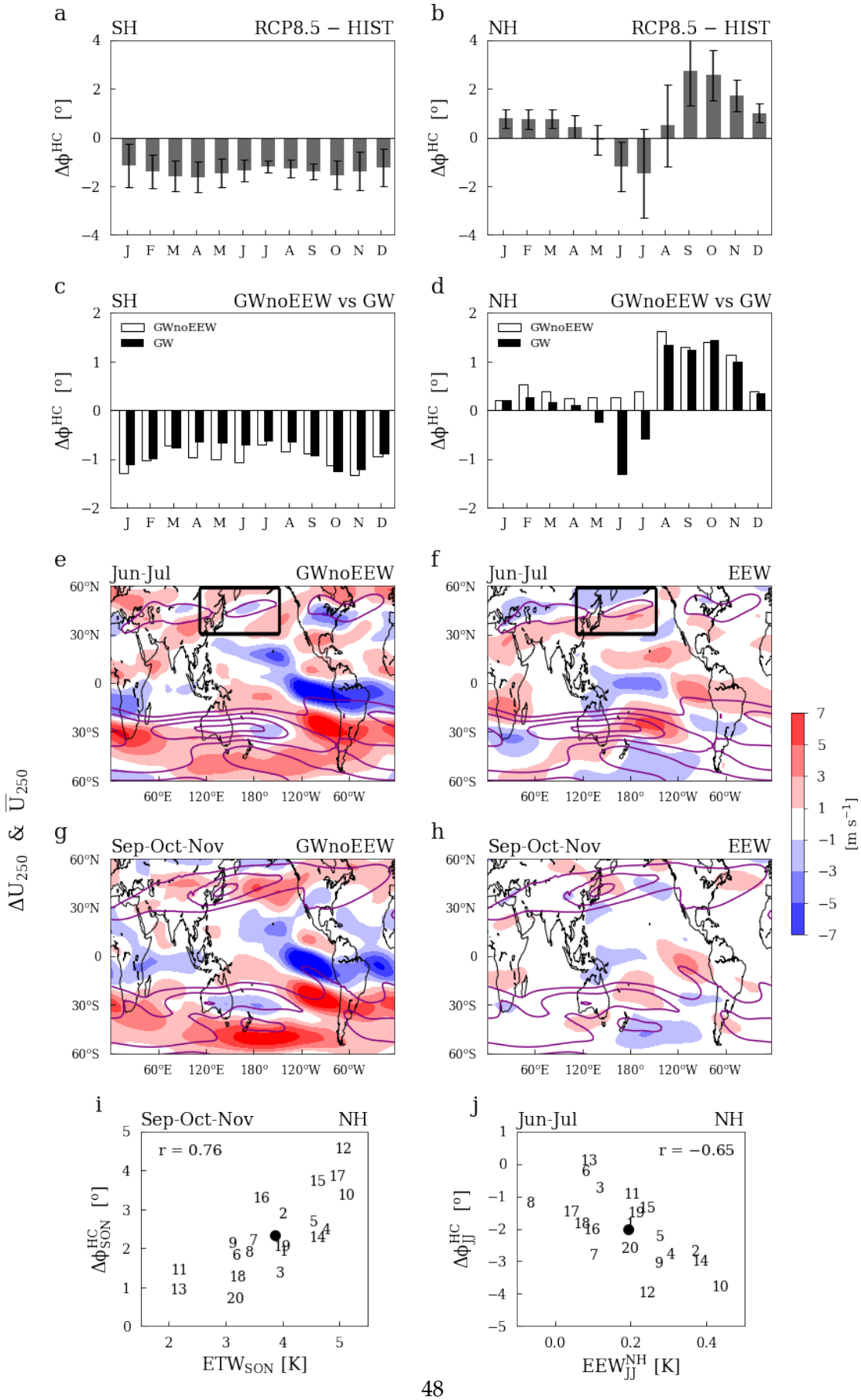
In [72]:
```
fig = plt.figure(figsize=(12,24))
ax=plt.subplot(522)
aa1=hb_rcp_n_all-hb_hist_n_all;
aa2=pe_rcp_n_all-pe_hist_n_all;
heights = np.mean((aa1+aa1)/2.,0)
yerr = np.std((aa1+aa1)/2.,0)
plt.bar(np.arange(0.6,12.6,1),heights,yerr=yerr,capsize=4,color='dimgrey',ecolor='k',w
plt.plot([-1,13],[0,0],'k-',linewidth=1)
plt.ylim(-4,4)
plt.xlim([-0.2,12.1])
plt.ylabel('$\mathregular{\Delta \phi^{HC}}$  [$\mathregular{^o}$]',fontsize=18)
plt.tick_params(labelsize=14,direction='in',length=5)
_=plt.xticks(np.arange(0.5,12.5,1),months,fontsize=13.5)
plt.title('NH',loc='left',fontsize=18)
plt.title('RCP8.5 $-$ HIST',loc='right',fontsize=18)
plt.text(-0.2, 1.1,'b',fontsize=22,transform = ax.transAxes)
ax=plt.subplot(521)
aa1=hb_rcp_s_all-hb_hist_s_all
aa2=pe_rcp_s_all-pe_hist_s_all
heights = np.mean((aa1+aa1)/2.,0)
yerr = np.std((aa1+aa1)/2.,0)
plt.bar(np.arange(0.6,12.6,1),heights,yerr=yerr,capsize=4,color='dimgrey',ecolor='k',w
plt.plot([-1,13],[0,0],'k-',linewidth=1)
plt.ylim(-4,4)
plt.xlim([-0.2,12.1])
plt.ylabel('$\mathregular{\Delta \phi^{HC}}$  [$\mathregular{^o}$]',fontsize=18)
plt.tick_params(labelsize=14,direction='in',length=5)
_=plt.xticks(np.arange(0.5,12.5,1),months,fontsize=13.5)
plt.title('SH',loc='left',fontsize=18)
plt.title('RCP8.5 $-$ HIST',loc='right',fontsize=18)
plt.text(-0.2, 1.1,'a',fontsize=22,transform = ax.transAxes)
ax=plt.subplot(524)
heightsaa = hb_am2gw1_n-hb_am2_n
heightsbb = hb_am2wtdepgw1_n-hb_am2_n

rects1=plt.bar(np.arange(0.35,12.35,1),heightsbb,capsize=3,color='grey',ecolor='dimgre
rects2=plt.bar(np.arange(0.65,12.65,1),heightsaa,capsize=3,color='k',ecolor='dimgrey'
```

```
plt.plot([-1,13],[0,0],'k-',linewidth=1)
plt.ylim(-2,2)
plt.xlim([-0.2,12.1])
plt.ylabel('$\mathregular{\Delta \phi^{HC}}$  [$\mathregular{^o}$]',fontsize=18)
plt.tick_params(labelsize=14,direction='in',length=5)
_=plt.xticks(np.arange(0.5,12.5,1),months,fontsize=13.5)
plt.title('NH',loc='left',fontsize=18)
plt.title('GWnoEEW vs GW',loc='right',fontsize=18)
plt.text(-0.2, 1.1,'d',fontsize=22,transform = ax.transAxes)

plt.legend((rects1[0], rects2[0]), ('GWnoEEW', 'GW'),loc='upper left',fontsize=11.7,f
ax=plt.subplot(523)
heightsaa = hb_am2gw_s-hb_am2_s
heightsbb = hb_am2wtdepgw_s-hb_am2_s
rects1=plt.bar(np.arange(0.35,12.35,1),heightsbb,capsize=3,color='grey',ecolor='dimgr
rects2=plt.bar(np.arange(0.65,12.65,1),heightsaa,capsize=3,color='k',ecolor='dimgrey'
plt.plot([-1,13],[0,0],'k-',linewidth=1)
plt.ylim(-2,2)
plt.xlim([-0.2,12.1])
plt.ylabel('$\mathregular{\Delta \phi^{HC}}$  [$\mathregular{^o}$]',fontsize=18)
plt.tick_params(labelsize=14,direction='in',length=5)
_=plt.xticks(np.arange(0.5,12.5,1),months,fontsize=13.5)
plt.title('SH',loc='left',fontsize=18)
plt.title('GWnoEEW vs GW',loc='right',fontsize=18)
plt.text(-0.2, 1.1,'c',fontsize=22,transform = ax.transAxes)

plt.legend((rects1[0], rects2[0]), ('GWnoEEW', 'GW'),loc='upper left',fontsize=11.7,f

lons, lats = np.meshgrid(lon_am2, lat_am2);
lat_south1=-60;lat_north1=60.;lon_west1=0.;lon_east1=357.5;lonr=0
map1=Basemap(projection='cyl',llcrnrlat=lat_south1,urcrnrlat=lat_north1,\
            llcrnrlon=lon_west1+lonr,urcrnrlon=lon_east1+lonr,resolution='c')
x, y = map1(lons, lats)
meridians1 = np.arange(60,420,60)

parallels1 = np.arange(-60,90,30)
levels_ua = [20,26,32,38,44,50]

levels_ua = [20,30,40,50]
levels = [-9,-7,-5,-3,-1.0,1.0,3,5,7,9]
ax=plt.subplot(525)
map1.drawcoastlines();#map1.fillcontinents()
plt.yticks([-60,-30,0,30,60],['60$\mathregular{^o}$S','30$\mathregular{^o}$S','0','30
plt.xticks([60,120,180,240,300],['60$\mathregular{^o}$E','120$\mathregular{^o}$E','18
map1.contour(x,y,ua_am2_summer,levels_ua,colors='purple',linestyle='-',linewidths=1.5
norm = matplotlib.colors.BoundaryNorm(levels,len(levels))
CS1=map1.contourf(x,y,ua_am2wtdepgw1_summer-ua_am2_summer,levels[1:-1],extend='both',
bb=np.zeros(np.shape(ua_am2_summer))
```

```
bb[60:74,45:85]=0.8
map1.contour(x,y,bb,colors='k',linewidths=1)
ax.set_aspect('auto')
plt.tick_params(labelsize=12,direction='in',length=5)
plt.title('Jun-Jul',loc='left',fontsize=18)
plt.title('GWnoEEW',loc='right',fontsize=18)
plt.text(-0.2, 1.1,'e',fontsize=22,transform = ax.transAxes)
ax=plt.subplot(527)
aa=np.zeros((90,144))
aa[66::,20:64]=0.1
map1.drawcoastlines();#map1.fillcontinents()
plt.yticks([-60,-30,0,30,60],['60$\mathregular{^o}$S','30$\mathregular{^o}$S','0','30
plt.xticks([60,120,180,240,300],['60$\mathregular{^o}$E','120$\mathregular{^o}$E','18
map1.contour(x,y,ua_am2_autumn,levels_ua,colors='purple',linestyle='-',linewidths=1.5
norm = matplotlib.colors.BoundaryNorm(levels,len(levels))
map1.contourf(x,y,ua_am2wtdepgw1_autumn-ua_am2_autumn+aa*24,levels[1:-1],extend='both
ax.set_aspect('auto')
plt.tick_params(labelsize=12,direction='in',length=5)
plt.text(-0.2, 1.1,'g',fontsize=22,transform = ax.transAxes)
plt.title('Sep-Oct-Nov',loc='left',fontsize=18)
plt.title('GWnoEEW',loc='right',fontsize=18)
ax=plt.subplot(526)
map1.drawcoastlines();#map1.fillcontinents()
plt.yticks([-60,-30,0,30,60],['60$\mathregular{^o}$S','30$\mathregular{^o}$S','0','30
plt.xticks([60,120,180,240,300],['60$\mathregular{^o}$E','120$\mathregular{^o}$E','18
map1.contour(x,y,ua_am2_summer,levels_ua,colors='purple',linestyle='-',linewidths=1.5
norm = matplotlib.colors.BoundaryNorm(levels,len(levels))
map1.contourf(x,y,ua_am2gw1_summer-ua_am2wtdepgw1_summer,levels[1:-1],extend='both',no
bb[60:74,45:85]=0.8
map1.contour(x,y,bb,colors='k',linewidths=1)
ax.set_aspect('auto')
plt.tick_params(labelsize=12,direction='in',length=5)
plt.title('Jun-Jul',loc='left',fontsize=18)
plt.title('EEW',loc='right',fontsize=18)
plt.text(-0.2, 1.1,'f',fontsize=22,transform = ax.transAxes)
ax=plt.subplot(528)
map1.drawcoastlines();#map1.fillcontinents()
plt.yticks([-60,-30,0,30,60],['60$\mathregular{^o}$S','30$\mathregular{^o}$S','0','30
plt.xticks([60,120,180,240,300],['60$\mathregular{^o}$E','120$\mathregular{^o}$E','18
map1.contour(x,y,ua_am2_autumn,levels_ua,colors='purple',linestyle='-',linewidths=1.5
norm = matplotlib.colors.BoundaryNorm(levels,len(levels))
bb=np.ones((90,144))
bb[32:55,84:120]=0.7
map1.contourf(x,y,(ua_am2gw1_autumn-ua_am2wtdepgw1_autumn)*bb,levels[1:-1],extend='bo
ax.set_aspect('auto')
plt.tick_params(labelsize=12,direction='in',length=5)
plt.title('Sep-Oct-Nov',loc='left',fontsize=18)
plt.title('EEW',loc='right',fontsize=18)
```

```
plt.text(-0.2, 1.1,'h',fontsize=22,transform = ax.transAxes)
plt.text(-1.65,1.1,'$\mathregular{\Delta U_{250}}$  &  $\mathregular{\overline{U}_{250

hadley_annual=(np.mean(hb_rcp_n_all,1)-np.mean(hb_hist_n_all,1))
hadley_s1=np.mean(hb_rcp_n_all[:,5:7],1)-np.mean(hb_hist_n_all[:,5:7],1)-(np.mean(hb_
hadley_a1=np.mean(hb_rcp_n_all[:,8:11],1)-np.mean(hb_hist_n_all[:,8:11],1)#-(np.mean(

hadley_s2=np.mean(hb_rcp_s_all[:,5:7],1)-np.mean(hb_hist_s_all[:,5:7],1)-(np.mean(hb_
hadley_a2=np.mean(hb_rcp_s_all[:,8:11],1)-np.mean(hb_hist_s_all[:,8:11],1)#-(np.mean(


ax01=plt.subplot(5,2,9)
for i in range(np.size(models_hist)):
    #exec('ax'+str(i)+'=plt.plot(gb_warm_n[i],hadley_a1[i],ms=10,c=\'dimgrey\',marker=
    exec('ax'+str(i)+'=plt.text(gb_warm_n[i],hadley_a1[i],str(i+1),horizontalalignment
ax20 = plt.plot(np.mean(gb_warm_n),np.mean(hadley_a1),ms=10,c='k',marker='o')
plt.xlim(1.5,5.5)
plt.ylim(0,5)
a=pearsonr(gb_warm_n,hadley_a1)[0]

plt.text(0.08,0.88,'r = '+str(np.floor(a*100)/100),fontsize=15.5,transform=ax01.transA
plt.xlabel('ETW$\mathregular{_{SON}}$ [K]',fontsize=18)
plt.ylabel('$\mathregular{\Delta \phi^{HC}_{SON}}$  [$\mathregular{^o}$]',fontsize=18
plt.tick_params(labelsize=14,direction='in',length=5)
plt.text(-0.2, 1.1,'i',fontsize=22,transform = ax01.transAxes)
plt.tick_params(labelsize=14,direction='in',length=5)
plt.title('Sep-Oct-Nov',loc='left',fontsize=18)
plt.title('NH',loc='right',fontsize=18)
ax02=plt.subplot(5,2,10)
dtp_warm_diff_mjj_n[5]=0.08
for i in range(np.size(models_hist)):
    #exec('ax'+str(i)+'=plt.plot(dtp_warm_diff_mjj_n[i],hadley_s1[i],ms=10,c=\'dimgrey
    exec('ax'+str(i)+'=plt.text(dtp_warm_diff_mjj_n[i],hadley_s1[i],str(i+1),horizonta
ax20 = plt.plot(np.mean(dtp_warm_diff_mjj_n),np.mean(hadley_s1),ms=10,c='k',marker='o
plt.xlim(-0.1,0.5)
plt.ylim(-5,1)
a=pearsonr(dtp_warm_diff_mjj_n,hadley_s1)[0]
plt.text(0.62,0.88,'r = $\minus$'+str(-np.floor(a*100)/100),fontsize=15.5,transform=ax
plt.xlabel('EEW$^\mathregular{NH}_\mathregular{JJ}$ [K]',fontsize=18)
plt.ylabel('$\mathregular{\Delta \phi^{HC}_{JJ}}$  [$\mathregular{^o}$]',fontsize=18,l
plt.tick_params(labelsize=14,direction='in',length=5)
plt.text(-0.2, 1.1,'j',fontsize=22,transform = ax02.transAxes)
plt.tick_params(labelsize=14,direction='in',length=5)
plt.title('Jun-Jul',loc='left',fontsize=18)
plt.title('NH',loc='right',fontsize=18)
cax = fig.add_axes([0.923, 0.355, 0.015, 0.13])
cb1=plt.colorbar(CS1, cax=cax, orientation='vertical',extendfrac=0)
cb1.ax.tick_params(labelsize=14)
```

```
cb1.set_label('[m s$^{-1}$]',fontsize=15)

plt.subplots_adjust(wspace=0.325,hspace=0.4)
box = ax02.get_position()
ax02.set_position([box.x0+0.035, box.y0, box.width*0.8, box.height])
box = ax01.get_position()
ax01.set_position([box.x0+0.035, box.y0, box.width*0.8, box.height])


plt.savefig('/Users/wenyuzhou/fig3.eps', bbox_inches='tight',format='eps', dpi=400)
```


a
SH
RCP8.5 − HIST
b
NH
RCP8.5 − HIST
c
SH
GWnoEEW vs GW
GWnoEEW
GW
d
NH
GWnoEEW vs GW
e
Jun-Jul
GWnoEEW
f
Jun-Jul
EEW
g
Sep-Oct-Nov
GWnoEEW
h
Sep-Oct-Nov
EEW
i
Sep-Oct-Nov
NH
r = 0.76
j
Jun-Jul
NH
r = −0.65
J F M A M J J A S O N D
60°N
30°N
0
30°S
60°S
60°E 120°E 180° 120°W 60°W
[m s⁻¹]

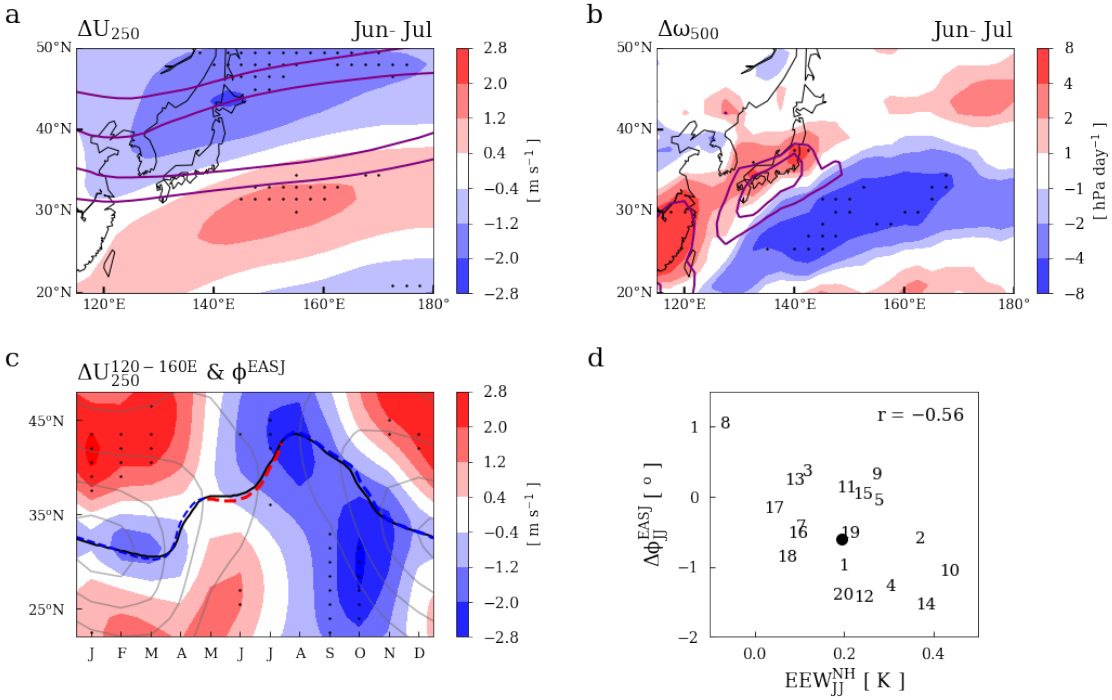
48

```
In [73]: fig=plt.figure(figsize=(17,10))
         ax06=plt.subplot(223)
         time = np.arange(-0.5,13.5,1)
         time_hi,lat_max_hist = get_max_lat(ua_p500_asia_hist[:,73:95],lat[73:95])
         time_hi,lat_max_rcp = get_max_lat(ua_p500_asia_rcp[:,73:95],lat[73:95])
         for i in np.arange(0,14,1):
             lat_max_rcp[i]=lat_max_rcp[i]+0.2
         aa=ua_p500_asia_rcp-ua_p500_asia_hist
         bb=get_extend(aa)
         CS=plt.contourf(time,lat,bb.T,[-3,-2,-1,1,2,3],cmap='bwr',extend='both')
         CS=plt.contourf(time,lat,bb.T,[-2.8,-2,-1.2,-0.4,0.4,1.2,2,2.8],cmap='bwr',extend='bo

         cc=ua_p500_asia_rcp_-ua_p500_asia_hist_
         dd=np.sum(cc>0,axis=0)-9
         dd[5,77:79]=16
         dd[6,88:91]=16

         dd[5,89]=16
         time1=np.arange(0.5,12.5,1)
         critic = 6
         lat2,time2 = np.meshgrid(lat,time1)
         plt.scatter(time2[np.abs(dd)>critic],lat2[np.abs(dd)>critic],s=7,marker='.',c='k')

         aa=ua_p500_asia_hist
         bb=get_extend(aa)
         plt.contour(time,lat,bb.T,[20,35,50,65],colors='dimgrey',alpha=0.5,linewidths=1.5)
         plt.plot(time_hi[:],lat_max_hist[:],'-',linewidth=2,color='k')
         #plt.plot(time_hi[:],lat_max_rcp[:],'--',linewidth=2,color='b')
         plt.plot(time_hi[0:44],lat_max_rcp[0:44],'--',linewidth=2,color='b')
         plt.plot(time_hi[70:],lat_max_rcp[70:],'--',linewidth=3,color='b')
         plt.plot(time_hi[44:70],lat_max_rcp[44:70],'--',linewidth=3,color='r')
         plt.ylim([22,48])
         cb06=plt.colorbar(CS,extendfrac=0)
         cb06.ax.tick_params(labelsize=14)
         cb06.set_label('[ m s$^{-1}$ ]',fontsize=14)
         ax06.set_aspect('auto')
         #plt.title('RCP8.5 $-$ HIST',loc='right',fontsize=17)
         plt.title('$\mathregular{\Delta U^{120-160E}_{250}}$ & $\mathregular{\phi^{EASJ}}$',lo
         plt.xlim([0,12])
         plt.xticks(np.arange(0,12,1)+0.5,months,fontsize=14.5)
         plt.yticks([25,35,45],['25$\mathregular{^o}$N','35$\mathregular{^o}$N','45$\mathregula
         plt.tick_params(labelsize=13,direction='in',length=5)
         plt.text(-0.2, 1.1,'c',fontsize=25,transform = ax06.transAxes)
```

```
ax03=plt.subplot(221)
lons, lats = np.meshgrid(lon_cmip5, lat_cmip5);
lat_south1=20.;lat_north1=50.;lon_west1=115.;lon_east1=180.;lonr=0
map1=Basemap(projection='cyl',llcrnrlat=lat_south1,urcrnrlat=lat_north1,\
            llcrnrlon=lon_west1+lonr,urcrnrlon=lon_east1+lonr,resolution='c')
x, y = map1(lons, lats)
parallels1 = [20,30,40,50];meridians1 = [120,140,160,180]
models1=['ACCESS1-0','ACCESS1-3','BNU-ESM','CanESM2','CCSM4','CESM1-CAM5','CNRM-CM5',\
        'GFDL-CM3','GISS-E2-R','inmcm4','IPSL-CM5A-LR','IPSL-CM5B-LR',\
        'MIROC5','MIROC-ESM','MRI-CGCM3','NorESM1-M']
models1=['ACCESS1-0','bcc-csm1-1','BNU-ESM','CanESM2','CCSM4','CESM1-CAM5','CNRM-CM5'
        'CSIRO-Mk3-6-0','GFDL-CM3','GISS-E2-R','HadGEM2-ES','inmcm4','IPSL-CM5A-LR',\
        'MIROC5','MIROC-ESM','MPI-ESM-LR','MRI-CGCM3','NorESM1-M']
[ua_hist,ua_rcp,ua_hist_,ua_rcp_]=get_gcm1(models1,'ua',ind=[5,7])
[wap_hist,wap_rcp,wap_hist_,wap_rcp_]=get_gcm1(models1,'wap',ind=[5,7])
[pr_hist,pr_rcp]=get_gcm(models1,'pr',ind=[5,7])
map1.drawcoastlines()
map1.drawparallels(parallels1,labels=[1,0,0,1],linewidth=2,fontsize=13,dashes=[3,900])
map1.drawmeridians(meridians1,labels=[1,0,0,1],linewidth=2,fontsize=13,dashes=[3,900])
map1.contour(x,y,ua_hist,np.arange(15.2,50,4),colors='purple',linestyle='-',linewidths
levels=[-4,-2.8,-2,-1.2,-0.4,0.4,1.2,2,2.8,4]
norm = matplotlib.colors.BoundaryNorm(levels,len(levels))
CS1=map1.contourf(x,y,ua_rcp-ua_hist+0.2,levels[1:-1],extend='both',norm=norm,cmap=plt
cc=ua_rcp_-ua_hist_
dd=np.sum(cc>0,axis=0)-9
ee = np.roll(dd,-5)
time1=np.arange(0.5,12.5,1)
critic = 5
map1.scatter(x[np.abs(ee)>critic],y[np.abs(ee)>critic],s=8,marker='.',c='k')
cb03=plt.colorbar(CS1,extendfrac=0)
cb03.ax.tick_params(labelsize=14)
cb03.set_label('[ m s$^{-1}$ ]',fontsize=14)
ax03.set_aspect('auto')
plt.title('$\mathregular{\Delta U_{250}}$',loc='left',fontsize=20,pad=10)
plt.title('Jun- Jul',loc='right',fontsize=20,pad=10)
plt.tick_params(labelsize=13,direction='in',length=5)
plt.text(-0.2, 1.1,'a',fontsize=25,transform = ax03.transAxes)

ax04=plt.subplot(222)
map1.drawcoastlines()
map1.drawparallels(parallels1,labels=[1,0,0,1],linewidth=2,fontsize=13,dashes=[3,900])
map1.drawmeridians(meridians1,labels=[1,0,0,1],linewidth=2,fontsize=13,dashes=[3,900])
map1.contour(x,y,-wap_hist,[30,40,50,60,70],colors='purple',linestyle='-',linewidths=
levels=[-18,-12,-6,-3.6,-1.2,1.2,3.6,6,12,18]
levels=[-12,-8,-4,-2,-1,1,2,4,8,12]
cmap=plt.cm.get_cmap('bwr',9)
norm = matplotlib.colors.BoundaryNorm(levels,len(levels))
CS1=map1.contourf(x,y,wap_rcp-wap_hist,levels[1:-1],extend='both',norm=norm,cmap=cmap)
```

```
cc=wap_rcp_-wap_hist_
dd=np.sum(cc>0,axis=0)-9
time1=np.arange(0.5,12.5,1)
critic = 5
map1.scatter(x[np.abs(dd)>critic],y[np.abs(dd)>critic],s=8,marker='.',c='k')


cb04=plt.colorbar(CS1,extendfrac=0)
cb04.ax.tick_params(labelsize=14)
cb04.set_label('[ hPa day$^{-1}$ ]',fontsize=14)
ax04.set_aspect('auto')
plt.title('$\mathregular{\Delta \omega_{500}}$',loc='left',fontsize=20,pad=10)
plt.title('Jun- Jul',loc='right',fontsize=20,pad=10)
plt.tick_params(labelsize=13,direction='in',length=5)
plt.text(-0.2, 1.1,'b',fontsize=25,transform = ax04.transAxes)


ax05=plt.subplot(224)
[jet_loc_hist,jet_loc_rcp] =get_jet_lat(models_hist)
jet_loc_hist_am = np.mean(jet_loc_hist,1)
jet_loc_rcp_am = np.mean(jet_loc_rcp,1)
jet_loc_diff_am = jet_loc_rcp_am - jet_loc_hist_am
jet_loc_diff_jj = np.mean(jet_loc_rcp[:,5:7],1) - np.mean(jet_loc_hist[:,5:7],1)
aa=jet_loc_diff_jj-jet_loc_diff_am
tmp1=[]
tmp2=[]
for i in range(np.size(models_hist)):
    if i==5:
        print(i)
    else:
        exec('ax'+str(i)+'=plt.text(dtp_warm_diff_mjj_n[i],aa[i],str(i+1),horizontala
        tmp1.append(dtp_warm_diff_mjj_n[i])
        tmp2.append(aa[i])
    plt.xlim(-0.1,0.5)
    plt.ylim(-2,1.5)
ax20 = plt.plot(np.mean(dtp_warm_diff_mjj_n),np.mean(aa),ms=10,c='k',marker='o')
plt.text(0.62,0.88,'r = $\minus$0.56',fontsize=16,transform=ax05.transAxes)
plt.xlabel('EEW$^\mathregular{NH}_\mathregular{{JJ}}$ [ K ]',fontsize=18)
plt.ylabel('$\mathregular{\Delta \phi^{EASJ}_{JJ}}$ [ $\mathregular{^o}$ ]',fontsize=
plt.tick_params(labelsize=14,direction='in',length=5)
plt.yticks([-2,-1,0,1])
#plt.xticks([0,0.1,0.2,0.3,0.4])
#plt.xlim([-0.06,0.285])
plt.text(-0.46, 1.1,'d',fontsize=25,transform = ax05.transAxes)
fig.subplots_adjust(hspace=0.4,wspace=0.3)
box = ax05.get_position()
ax05.set_position([box.x0+0.04, box.y0, box.width*0.6, box.height])
plt.savefig('/Users/wenyuzhou/fig4.eps', bbox_inches='tight',format='eps', dpi=400)
```

5


a
$\Delta U_{250}$
Jun- Jul
50°N
40°N
30°N
20°N
120°E
140°E
160°E
180°
2.8
2.0
1.2
0.4
−0.4
−1.2
−2.0
−2.8
[ m s$^{-1}$ ]
b
$\Delta\omega_{500}$
Jun- Jul
8
4
2
1
−1
−2
−4
−8
[ hPa day$^{-1}$ ]
c
$\Delta U_{250}^{120-160E}$ & $\phi^{EASJ}$
45°N
35°N
25°N
J F M A M J J A S O N D
d
r = −0.56
$\Delta\phi_{JJ}^{EASJ}$ [ ° ]
$EEW_{JJ}^{NH}$ [ K ]
0.0
0.2
0.4
1
0
−1
−2
8
13
3
9
11
15
5
17
16
7
19
2
18
1
10
20
12
4
14