```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=Tr
```

```
%cd /content/drive/MyDrive/BigDataHub/Differentiable_Render/
```

```
/content/drive/MyDrive/BigDataHub/Differentiable_Render
```

```
try:
  import datatools_bdh
except ModuleNotFoundError:
  !pip install git+https://github.com/sfu-bigdata/datatools-bdh.git
  import datatools_bdh
from datatools_bdh.ipython import *
```

```
Collecting git+https://github.com/sfu-bigdata/datatools-bdh.git
  Cloning https://github.com/sfu-bigdata/datatools-bdh.git to /tmp/pip-req-build-t7geafby
  Running command git clone -q https://github.com/sfu-bigdata/datatools-bdh.git /tmp/pip-req-build-t7geafby
Building wheels for collected packages: datatools-bdh
  Building wheel for datatools-bdh (setup.py) ... done
  Created wheel for datatools-bdh: filename=datatools_bdh-0.1-py3-none-any.whl size=33526 sha256=7b4b5a6ffc0b07aacce1290bef0
  Stored in directory: /tmp/pip-ephem-wheel-cache-r_hafj_6/wheels/dd/f7/e6/04bb4fb69b7d39aa1c649649a211bfdff04186aaa3defc6d9
Successfully built datatools-bdh
Installing collected packages: datatools-bdh
Successfully installed datatools-bdh-0.1
```

```
import pandas as pd
from datatools_bdh import _get_resource_path
from IPython.display import Markdown
import matplotlib.pyplot as plt
import seaborn as sns

# !wget https://raw.githubusercontent.com/sfu-bigdata/range-driver/public/range_driver/dict_utils.py
from dict_utils import *

base_dir = '/content/drive/MyDrive/BigDataHub/Differentiable_Render/conf/'
ymal_path = "training_config.yml"
conf = yload(open(base_dir + ymal_path))
```

```
plt.style.use(_get_resource_path('report.mplstyle'))
def displaymd(mdstr):
    display(Markdown(mdstr))
```

```
data_path = conf.train.output_result_dir

train_name = conf.analysis.train_loss_profile
val_name = conf.analysis.val_loss_profile
train_net_out_name = conf.analysis.train_output_profile
val_net_out_name = conf.analysis.val_output_profile
```

load the loss and output profiles

```
#data_path = "../data"

suffix = ""
#val_loss = pd.read_csv(f"{data_path}/val_loss_detail_pdepoch50{suffix}.csv")
val_loss = pd.read_csv(f"{data_path}{val_name}")
val_loss.set_index(['epoch','batch_id','index'])
# there are some irregular rows in this val loss .csv, we split them out below
total_rows = val_loss.isna().any(axis=1)
if any(total_rows):
    val_loss_totals = val_loss[total_rows]
    val_loss = val_loss[~total_rows]
if  val_loss_totals.iloc[0,0][0] == '[':
    val_loss_totals.iloc[:,-3:] = val_loss_totals['epoch'].str[1:-1].str.split(expand=True)
    val_loss_totals.loc[:,'epoch'] = val_loss.loc[val_loss_totals.index-1, 'epoch']
```

```

#train_loss = pd.read_csv(f"{data_path}/train_loss_detail_pdepoch50{suffix}.csv")
train_loss = pd.read_csv(f"{data_path}/{train_name}")
```

```
val_net_output = pd.read_csv(f"{data_path}/{val_net_out_name}")
train_net_output = pd.read_csv(f"{data_path}/{train_net_out_name}")

num_index = train_net_output['index'].unique().shape[0] # batch_size
train_net_output['ex_id'] = train_net_output['batch_id']*num_index+train_net_output['index']
val_net_output['ex_id'] = val_net_output['batch_id']*num_index+val_net_output['index']

```

```
train_net_output.shape
val_net_output.shape
```

```
(5000, 14)
```

```
import numpy as np
def col_to_int(col):
    return lambda x: pd.to_numeric(x[col]).astype(int)

train_loss = train_loss.assign(epoch=col_to_int('epoch'),
                  batch_id=col_to_int('batch_id'),
                  index=col_to_int('index'))
val_loss = val_loss.assign(epoch=col_to_int('epoch'),
                  batch_id=col_to_int('batch_id'),
                  index=col_to_int('index'))
```

```
train_loss.shape[0]/725
val_loss.shape[0]/250
```

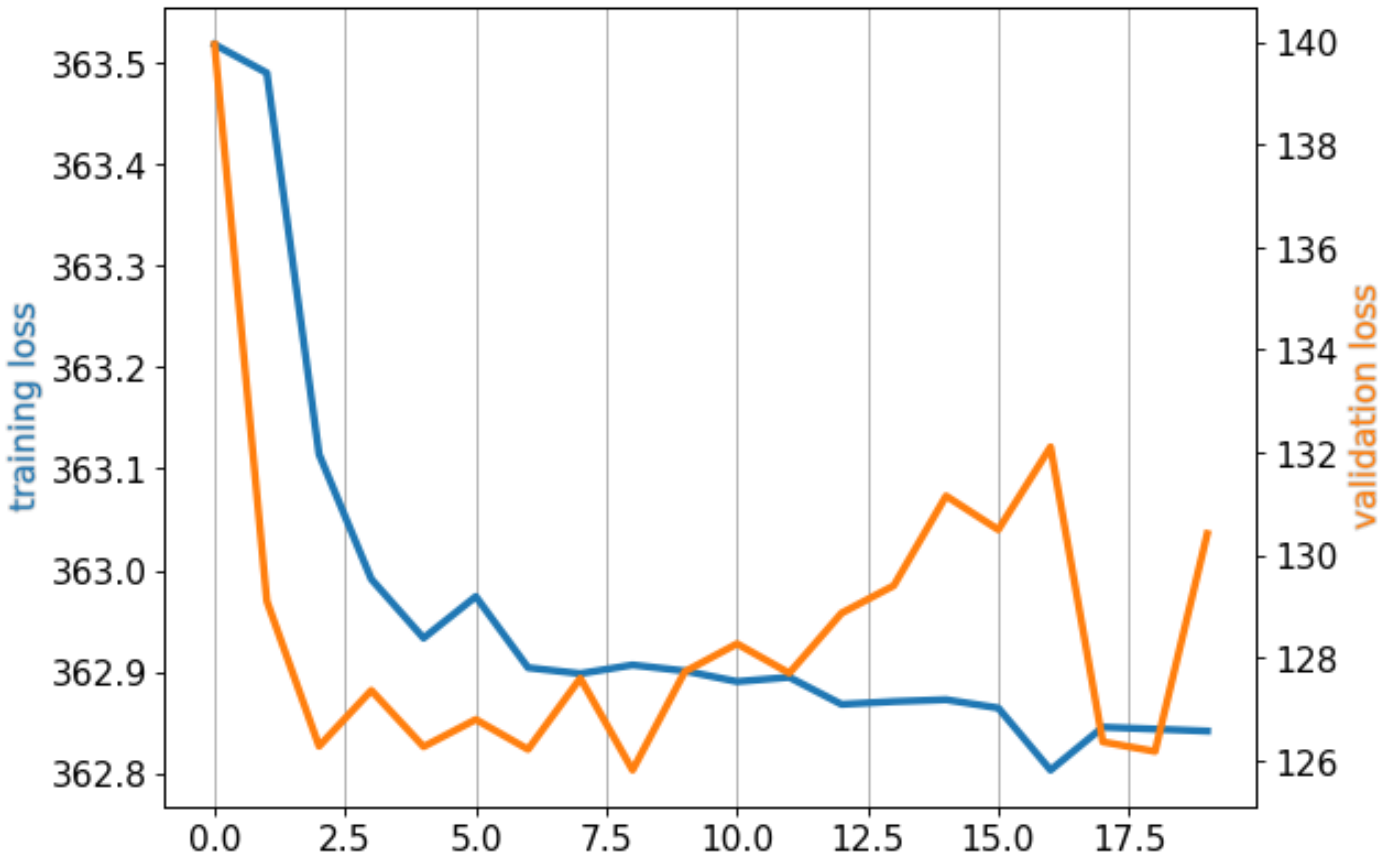
```
20.0
```

```
# calculate the index
num_index = train_loss['index'].unique().shape[0] # batch_size
train_loss['ex_id'] = train_loss['batch_id']*num_index+train_loss['index']
val_loss['ex_id'] = val_loss['batch_id']*num_index+val_loss['index']
```

```
# dataset size
train_dsize = train_loss[train_loss['epoch'] ==0 ].shape[0]
val_dsize = val_loss[val_loss['epoch'] ==0 ].shape[0]
```

```
1
```

```
import matplotlib.patheffects as pe
color_cycle_plt = plt.rcParams['axes.prop_cycle'].by_key()['color']
peff = [pe.withStroke(linewidth=1.5, foreground="lightgray")]

#ax = plt.su(figsize=(16,12))
plt.figure(figsize=(8, 6), dpi=80)
ax = plt.gca()
train_loss.groupby('epoch').mean().loc[:,"t1":].sum(axis=1).plot(ax=ax)
# twin object for two different y-axis on the sample plot
ax.set_ylabel("training loss",
              color=color_cycle_plt[0],
              fontsize=14,
              path_effects=peff)
ax2=ax.twinx()
# make a plot with different y-axis using second axis object
#ax2.plot(gapminder_us.year, gapminder_us["gdpPercap"],color="blue",marker="o")
ax2.set_ylabel("validation loss",
              color=color_cycle_plt[1],
              fontsize=14,
              path_effects=peff)
val_loss.groupby('epoch').mean().loc[:,"t1":].sum(axis=1).plot(c=color_cycle_plt[1], ax=ax2)
ax.xaxis.grid(True)
ax2.set_axisbelow(True)
```



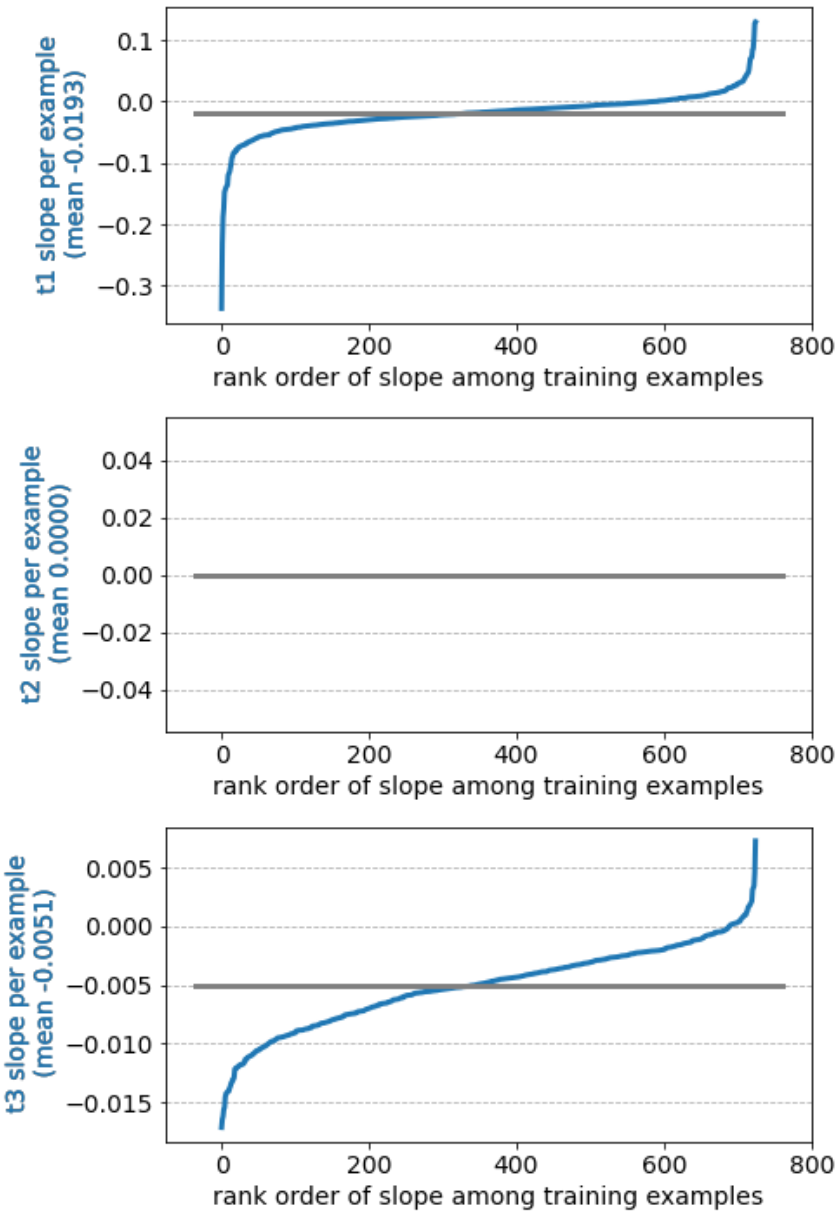

```
1
2
3 from scipy import stats
4
5 plt.rc('text', usetex=False)
6 peff = [pe.withStroke(linewidth=.2, foreground="black")]
7
8 def plot_target_field(target_field, sample_data, **kwargs):
9     lmdf = (sample_data
10             .groupby('ex_id')
11             .apply(lambda gn: stats.linregress(gn['epoch'], gn[target_field]))
12             .apply(pd.Series)
13            )
14     lmdf.columns = "slope intercept r_value p_value std_err".split()
15     mean_slope = lmdf['slope'].mean()
16     plt.plot(lmdf['slope'].sort_values().values, linewidth=3, **kwargs)
17     plt.plot(plt.gca().get_xlim(),
18              [mean_slope]*2,
19              color='gray',
20              **kwargs)
21     plt.ylabel(f'{target_field} slope per example\n(mean {mean_slope:.4f})',
22                color=color_cycle_plt[0],
23                fontsize=14,
24                path_effects=peff)
25     plt.xlabel('rank')
26     plt.gca().yaxis.grid(True, linestyle='dashed')
27     return lmdf
28
29 #plt.figure(figsize=(6,4))
30 fig, axs = plt.subplots(3,2, figsize=(16, 12))
31 displaymd("""# Distribution of loss term slope across training and validation examples
32 For each training or validation example, the three loss terms t1, t2, t3 are each fitted to a linear model w.r.t. training time
33 A negative slope indicates that the respective loss term for an example improves (i.e. gets smaller with increasing epochs).
34
35 We are sorting the slopes across all examples to get an idea of their distribution around the average slope across all examples
36 which is indicated by a gray horizontal line in each plot.
37 """)
38 plt.subplots_adjust(wspace=.4, hspace=.3)
39 vlmdfs = None
40 tlmdfs = None
41 for tid, term in enumerate(['t1', 't2', 't3']):
42     #sample_data = choose_N_examples(val_loss, N=10, field='ex_id')
43     ax=axs[tid][1] # right axis in row tid
44     plt.sca(ax)
45     vlmdf = plot_target_field(term, val_loss)
46     vlmdf['term'] = term
47     vlmdfs = pd.concat([vlmdfs, vlmdf])
48     plt.xlabel('rank order of slope among validation examples')
49     sample_data = (train_loss.groupby('ex_id')
50                                  .sample(3)
51                                  .reset_index(drop=False))
52     ax=axs[tid][0] # left axis in row tid
53     plt.sca(ax)
54     tlmdf = plot_target_field(term, train_loss)
55     tlmdf['term'] = term
```

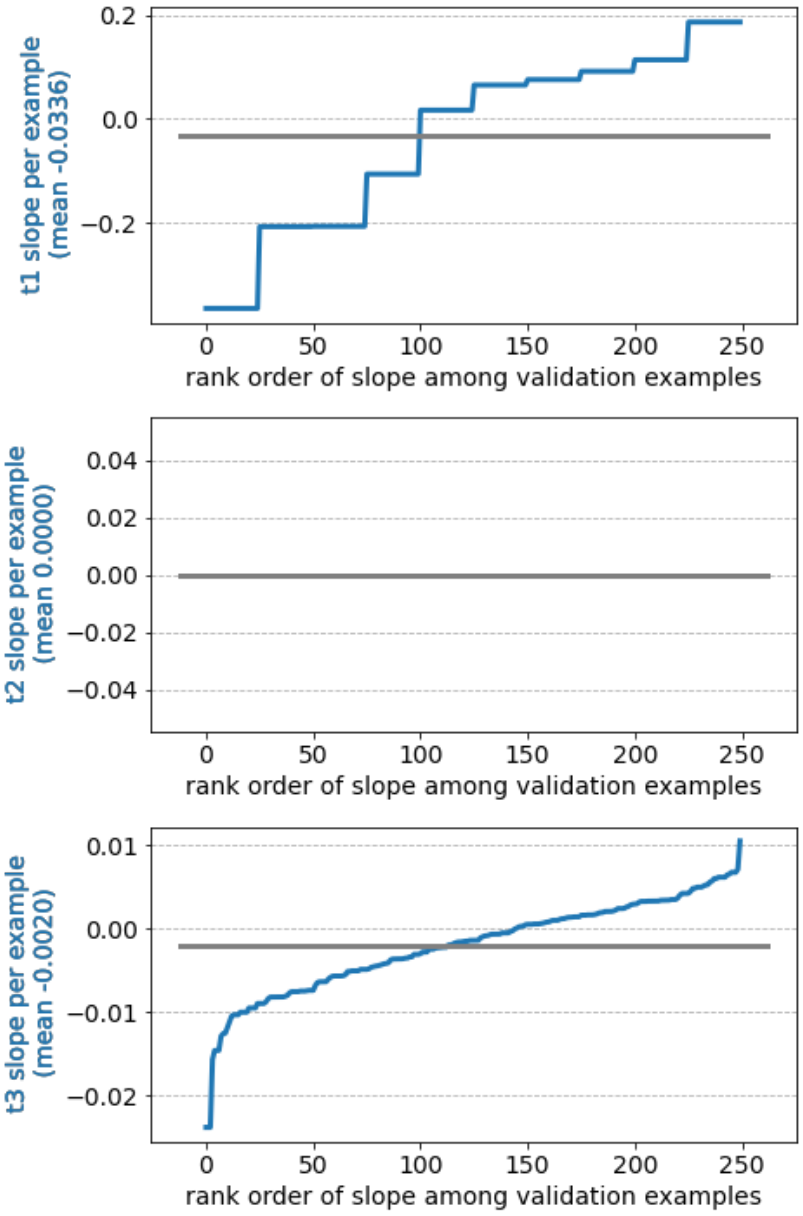
```
    tlmdf['term'] = term
    tlmdfs = pd.concat([tlmdfs, tlmdf])
    plt.xlabel('rank order of slope among training examples');

```

Distribution of loss term slope across training and validation examples

For each training or validation example, the three loss terms t1, t2, t3 are each fitted to a linear model w.r.t. training time given by the number of epochs. A negative slope indicates that the respective loss term for an example improves (i.e. gets smaller with increasing epochs).

We are sorting the slopes across all examples to get an idea of their distribution around the average slope across all examples (average slope of total loss), which is indicated by a gray horizontal line in each plot.





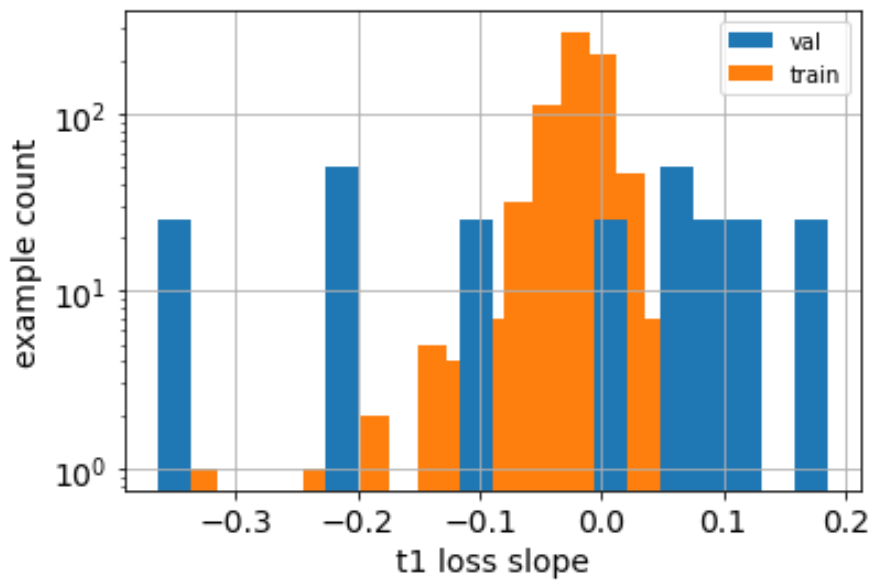

```
def add_ex_info(lmdfs, df):
    if 'batch_id' not in lmdfs:
        ex_ids = df[['batch_id','index','ex_id']].drop_duplicates()
        #display(ex_ids)
        lmdfs = lmdfs.join(ex_ids)#.sort_values(by='slope', ascending=False)
        return lmdfs.set_index(['batch_id','index']).sort_index()
    return lmdfs
vlmdfs = add_ex_info(vlmdfs, val_loss)
tlmdfs = add_ex_info(tlmdfs, train_loss)

#.plot(style='.', alpha=.5)
#sns.heatmap(data=vlmdfs, x='batch_id', y='index', c='slope')
```

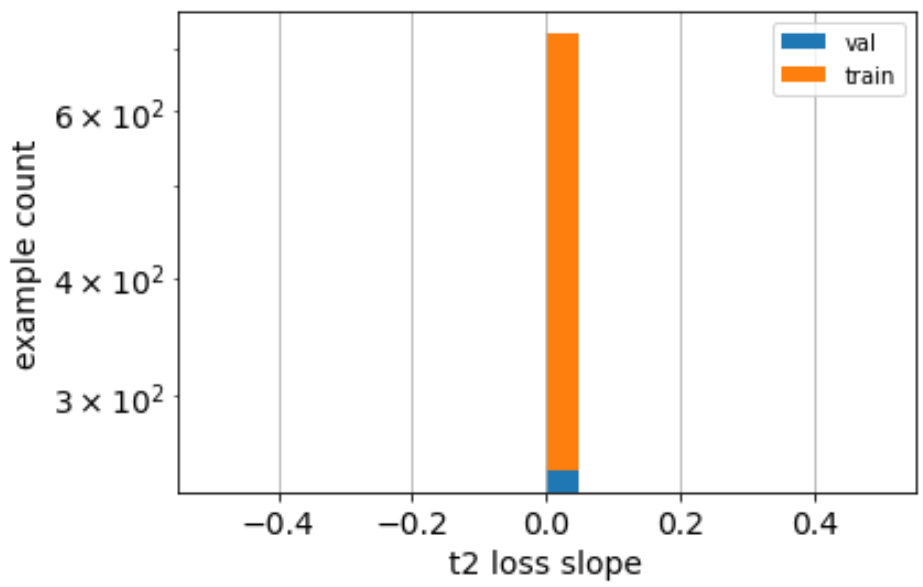
```
def floor_factor(value, factor):
    return np.floor(value/factor)*factor

def ceil_factor(value, factor):
    return np.ceil(value/factor)*factor

def grid_ticks(ser, step=.05):
    return np.arange(floor_factor(ser.min(), step),
                     ceil_factor(ser.max(), step),
                     step)

```

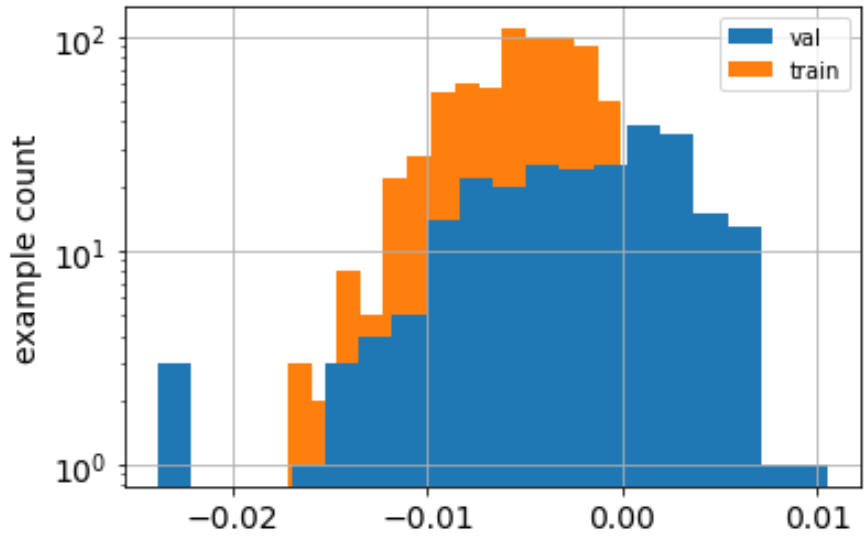
```
do_show_scatterslopes = False
if do_show_scatterslopes:
    tlmdfs['slope'].plot(style='.', alpha=.1)
    # make a grid of .05 spaced ticks
    plt.gca().set_yticks(grid_ticks(tlmdfs['slope']))
    plt.grid('minor')
```

```
displaymd("# Comparing loss slope distribution between training and validation")
hist_ps = dict(bins=20, log=True)
for field in ['t1', 't2', 't3']:
    (vlmdfs[(vlmdfs['term']==field).values]
      .sort_index()['slope']
      .hist(**hist_ps)
    )
    (tlmdfs[(tlmdfs['term']==field).values]
      .sort_index()['slope']
      .hist(**hist_ps, zorder=0)
    )
    plt.yscale('log')
    plt.legend(['val', 'train'])
    plt.xlabel(f"{field} loss slope")
    plt.ylabel("example count");
    plt.show()
    #sns.heatmap(data=vlmdfs, x='batch_id', y='index', c='slope')
```

Comparing loss slope distribution between training and validation

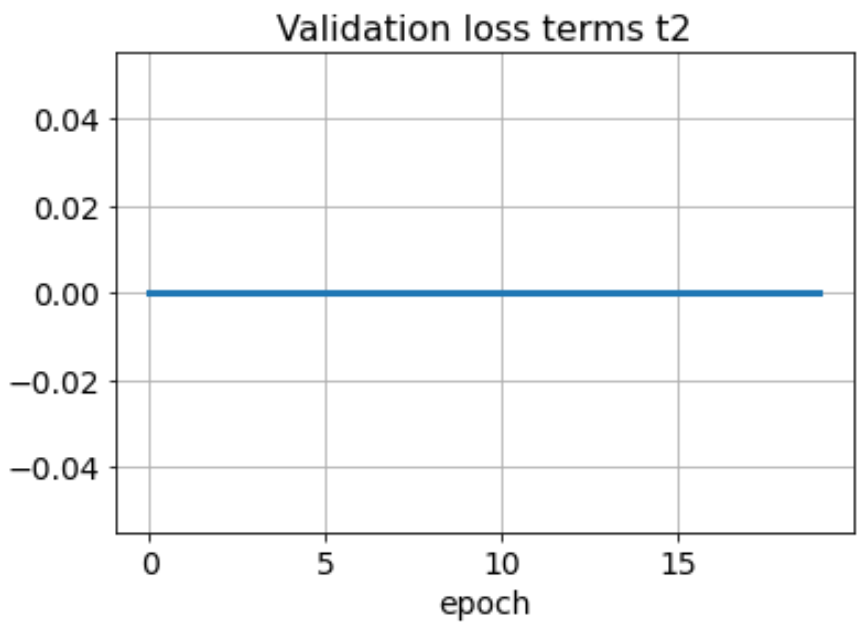







```
import matplotlib.pyplot as plt
target_term = "t2"
```

```
if False:
    trainval = "Training"
    dat = train_loss
else:
    trainval = "Validation"
    dat = val_loss
# taking the mean across: batch_id, index
(dat.groupby('epoch').mean().loc[:,target_term]
 #.sum(axis=1)
 .plot())
plt.title(f"{trainval} loss terms {target_term}")
plt.grid(True)
```




```
#vlmdfs.sort_values('slope', ascending=False).index[:10]

# (val_loss
#  .set_index(['batch_id','index'])
# .join(vlmdfs.sort_values('slope', ascending=False).iloc[:10]))

#tlmdfs.sort_values('slope', ascending=False).iloc[:10]

```

```
## read the image from drive (here I use sythetic image from  Boeing)
## option:  appy segmentation mask on image
import cv2
import os
import numpy as np

# acquire target img
def gen_target(synthetic_base_dir, img_name, seg_name):
    #read RGB image
    img_p = os.path.join(synthetic_base_dir, 'render_imgs', img_name)
    seg_p = os.path.join(synthetic_base_dir, 'seg_imgs', seg_name)

    target_img = cv2.imread(img_p)
    # print(target_img)

    #read segmentation img
    seg_img = cv2.imread(seg_p)

    # print(seg_img.shape)
    # print(np.unique(seg_img))
    # print(target_img.shape)

    ## apply segmentation mask on image (depends on your needs)
    target_img = target_img * (seg_img/255.0)
    #plt.imshow(seg_img.astype(np.float))
    #plt.imshow(target_img/255.0)
    #plt.show()

    return target_img
```

```
def visualize_picked_images(ex_index_list, mode):

   seq_file = 'metadata_train_sequence.txt' if mode=="Training" else ('metadata_val_sequence.txt' if mode =="Validation" else
   print("\n================================")
   print(f"mode: {mode}  seq_file:{seq_file}  ex_index_list: {ex_index_list} " )
```

```

  top_sweeps = []
  for ex_id in ex_index_list:
    # retrive sweep from meta file
    print(f"\n---------------------- sample index: {ex_id} --------------------------")
    visual_sweep(seq_file, ex_id)
    #get network output of last epoch
    dis_of_lastepoch_val = val_net_output.iloc[-val_dsize:,:]
    dis_of_lastepoch_train = train_net_output.iloc[-train_dsize:,:]
    #print distance of souce images in sweep
    #print("=====", distance_of_lastepoch[distance_of_lastepoch['ex_id'] == ex_id].loc[:,'d0':'d8'].to_numpy())
    if mode == "Validation":
      d = dis_of_lastepoch_val[dis_of_lastepoch_val['ex_id'] == ex_id].loc[:,'d0':'d8'].to_numpy()[0].tolist()
    elif mode == "Training":
      d = dis_of_lastepoch_train[dis_of_lastepoch_train['ex_id'] == ex_id].loc[:,'d0':'d8'].to_numpy()[0].tolist()

    display_d = list(map(lambda x:round(x, 4), d))
    print(f"distance: {display_d}")
    print(f"deviation strategy: {strategy_pose[ex_id%25]}" )
    #top_sweeps.append(sweep_names)

  return top_sweeps


```

```
def visual_sweep(seq_file, ex_id):
    df_st = pd.read_csv(dataset_dir + seq_file, header=None)
    df_st[21] = None
    df_st.loc[:,12:21] = df_st.loc[:,11:20].values
    df_st.loc[:,10:11] = df_st.loc[:,10].str.split(expand=True).values

    #get the sample
    print("=================", ex_id, dataset_dir + seq_file)
    print(df_st.shape)

    sample_info = df_st.iloc[ex_id,:].to_numpy()
    #print("sample info :", sample_info)

    tg_img_name = sample_info[0]
    sweep_names = sample_info[1:11]
    #print(f"sweep name: {sweep_names}")

    s_img_p = dataset_dir + "train_seq/"
    t_base_dir = dataset_dir + "Blender_images"

    #get target image
    target_img = gen_target(t_base_dir, tg_img_name, tg_img_name)
    target_img = cv2.resize(target_img,(400,300))

```

```
    plt.imshow(target_img[:,:,::-1]/255.0)
    plt.show()
    plt.figure(figsize=(25,15))
    for idx, s_img_name in enumerate(sweep_names):
      # get source image:
      #print(s_img_p + s_img_name)
      s_img = cv2.imread(s_img_p + s_img_name)
      # generate difference img between t and s
      diff_img = np.abs(target_img - s_img)/255.0

      # display images sweep
      plt.subplot(1,10,idx+1)
      plt.axis('off')
      plt.imshow(diff_img)
    plt.show()


```

```
def plot_lines(gdf, groupvar='epoch', fields=['t1','t2'],
               c=None, nsteps=100,
               group_agg='mean',
```

```
                cmap='turbo', decorate=True,
                kind='scatter',
                **kwargs):
    if not groupvar is None:
        gdf = gdf.groupby(groupvar).mean()
    gdf = gdf.reindex(np.arange(0,gdf.index.max(),1/nsteps)).interpolate('linear')
    if c is None:
        c = gdf.index
    data_args = dict(data=gdf, x=fields[0], y=fields[1])
    if kind == 'scatter':
        plt.scatter(
            **data_args,
            c=c, s=3, cmap=cmap,
            **kwargs)
    elif kind == 'line':
        sns.lineplot(
            data=gdf, x=groupvar, y=fields[0],
            #**data_args,
#           c=c, s=3, cmap=cmap,
            **kwargs)
    else:
        sns.kdeplot(
            **data_args,
            **kwargs)
    if decorate:
        plt.xlabel(fields[0])
        plt.ylabel(fields[1])
        cbar = plt.colorbar()
        if not groupvar is None:
            cbar.set_label(groupvar)
            cbar.set_alpha(1)
            cbar.draw_all()
        plt.grid()


plt.figure(figsize=(6,5))

def plot_ex_loss_term(mode, sel_ex, loss_df, fields=['t2','t3'], ):
    sel_ex_ids = sel_ex['ex_id']
    #visualize_picked_images(sel_ex_ids)
    #print("sel_ex =======, ")
    #print( sel_ex['ex_id'].to_numpy())

    val_loss_sample = loss_df.set_index('ex_id').loc[sel_ex_ids]
    scatter_ps = dict(kind='scatter')
    line_ps = dict(kind='line', decorate=False, alpha=.6)
    kde_ps = dict(kind='kde', fill=True, cmap='mako', decorate=False)
    plot_ps = line_ps
    #k, gdf = 0, val_loss_sample
    display(sel_ex[['slope','term','ex_id']])
    for k, (gn, gdf) in enumerate(val_loss_sample.groupby('ex_id')):
        if not 'decorate' in plot_ps:
            add_arg = dict(decorate=(k==0))
        else:
            add_arg = {}
        plot_lines(gdf,
                   fields=fields,
                   **plot_ps,
                   **add_arg,
                  )
    plt.grid(True)
    plt.show()

    #visualization
    visualize_picked_images(sel_ex['ex_id'].to_numpy(), mode)

def show_top_bottom(field, lmdfs, loss_df, mode, ntop=10 ):
    fields = [field, 't3']
    displaymd(f"### Top {ntop} examples for {field}")
    #ex_ids = val_loss[['batch_id','index','ex_id']].drop_duplicates()
    slmdfs = lmdfs.loc[lmdfs['term']==field].sort_values(by='slope', ascending=True)
    sel_ex = slmdfs.iloc[:ntop]
    plot_ex_loss_term(mode, sel_ex, loss_df=loss_df,  fields=fields)
    plt.show()

```

```
    displaymd(f"### Bottom {ntop} examples for {field}")
    sel_ex = slmdfs.iloc[-ntop:]
    plot_ex_loss_term(mode, sel_ex, loss_df=loss_df, fields=fields, )
    plt.show()
```

```
<Figure size 432x360 with 0 Axes>
```

Top and bottom perfomance of cases analysis

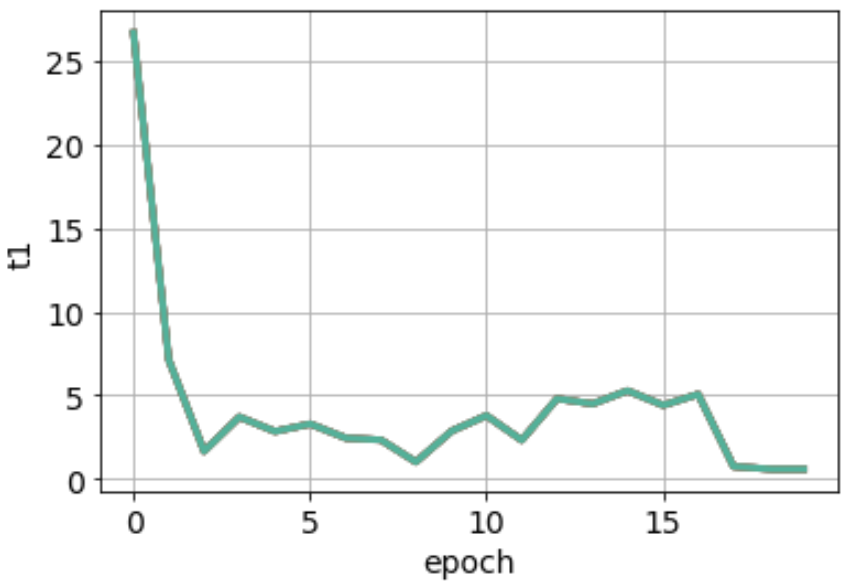
```
#----------------------
dataset_dir = conf.dataset.dataset_dir

displaymd("# Loss change per example across epochs")

# for mode, (lmdfs, loss_df) in {'Training': (tlmdfs, train_loss),
#                                'Validation': (vlmdfs, val_loss)}.items():

for mode, (lmdfs, loss_df) in {'Validation': (vlmdfs, val_loss)}.items():

    displaymd(f"## {mode} performance across epochs")
    show_top_bottom('t1', lmdfs, loss_df, mode)
    show_top_bottom('t2', lmdfs, loss_df, mode)
    show_top_bottom('t3', lmdfs, loss_df, mode)
```

Loss change per example across epochs

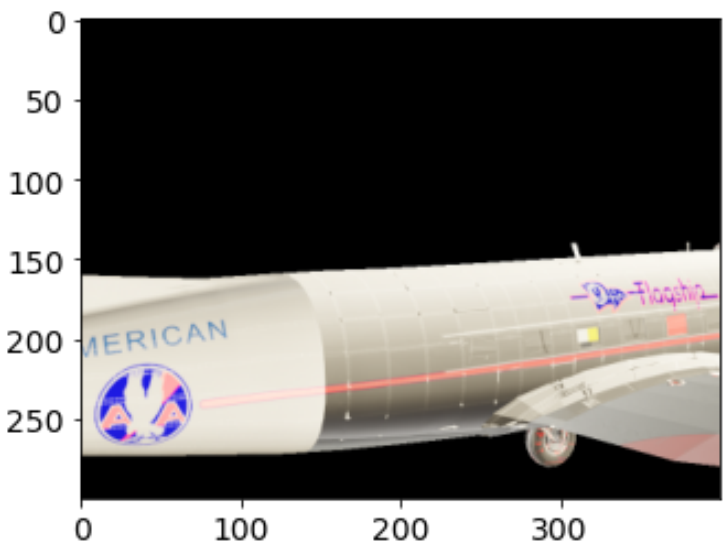
Validation performance across epochs

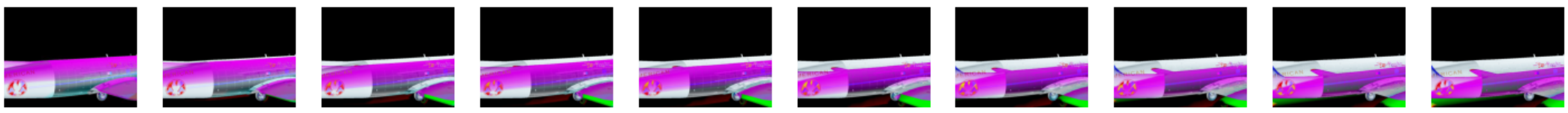
Top 10 examples for t1

batch_id	index	slope	term	ex_id
44	**0**	-0.364344	t1	176
49	**1**	-0.364344	t1	197
	0	-0.364344	t1	196
48	**3**	-0.364344	t1	195
	2	-0.364344	t1	194
	1	-0.364344	t1	193
	0	-0.364344	t1	192
47	**3**	-0.364344	t1	191
	2	-0.364344	t1	190
	1	-0.364344	t1	189




```
==================================
mode: Validation  seq_file:metadata_val_sequence.txt  ex_index_list: [176 197 196 195 194 193 192 191 190 189]

----------------------- sample index: 176 ---------------------------
================= 176 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

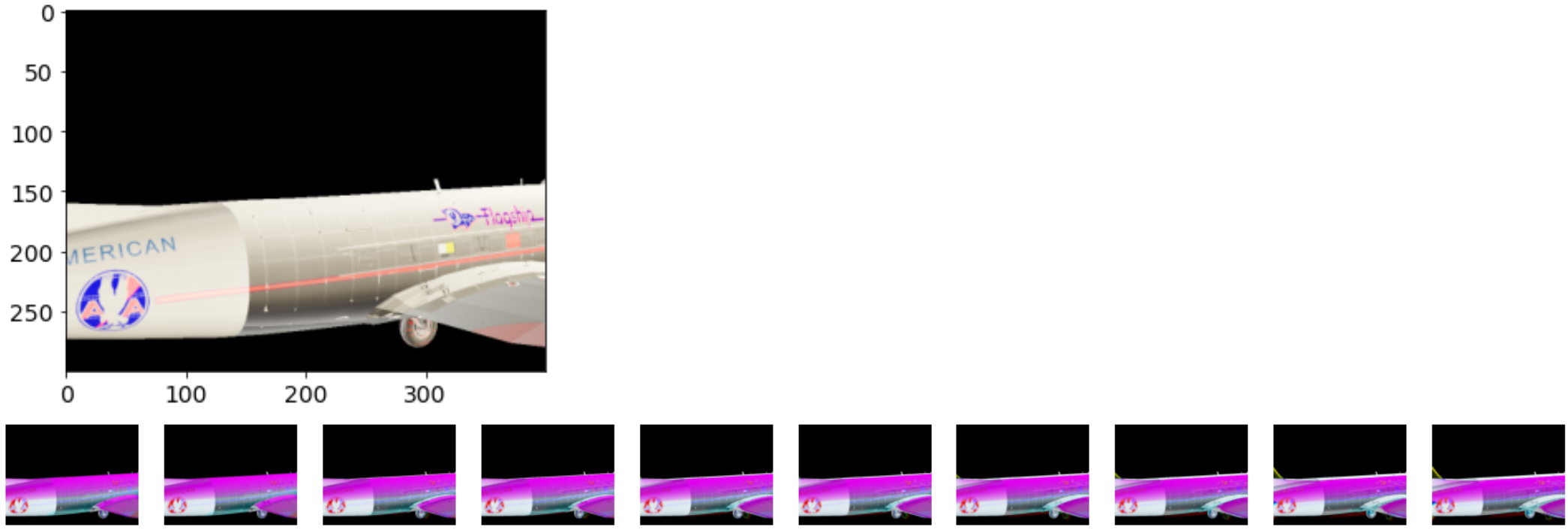




```
distance: [3.2969, 15.4515, 22.2906, 22.8493, 24.7313, 26.0385, 25.4434, 24.5321, 21.3761]
deviation strategy: [1, 0.2, 0.2, 0, 0, 0]

----------------------- sample index: 197 ---------------------------
================= 197 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

distance: [3.2969, 2.6577, 2.6153, 2.7307, 2.7626, 2.7626, 2.9926, 3.6791, 5.6197]
deviation strategy: [0, 0, 0, 0, 0, -1]

---------- sample index: 196 ----------
========== 196 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

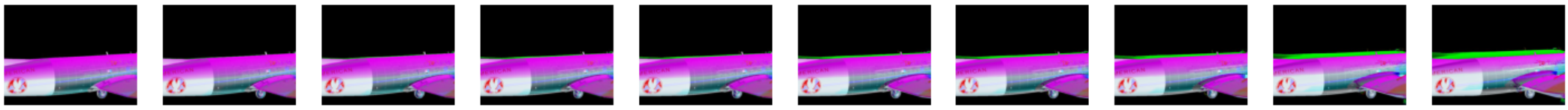


distance: [3.2969, 8.9341, 13.5769, 16.3363, 20.6305, 21.3377, 22.0537, 22.3471, 22.9813]
deviation strategy: [0, 0, 0, 0.2, 0.2, 1]

---------- sample index: 195 ----------
========== 195 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

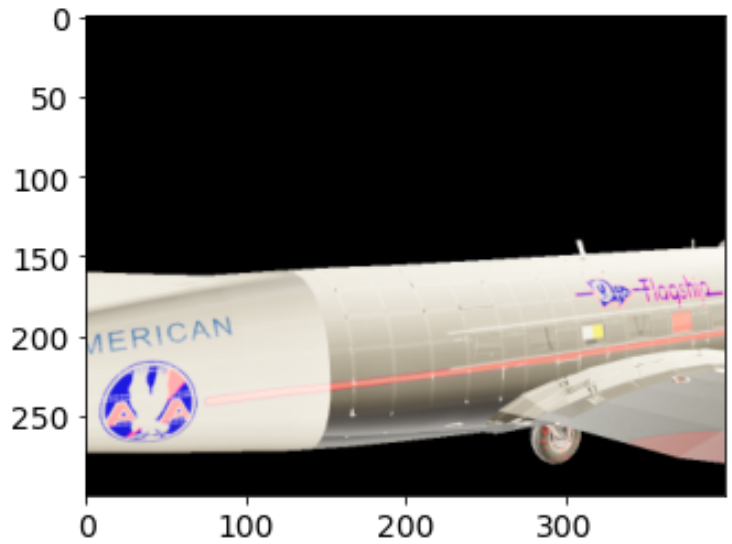
distance: [3.2969, 3.3306, 7.2806, 7.9442, 12.2747, 12.2747, 15.6149, 17.2958, 20.1994]
deviation strategy: [0, 0, 0, 0, 0, 1]

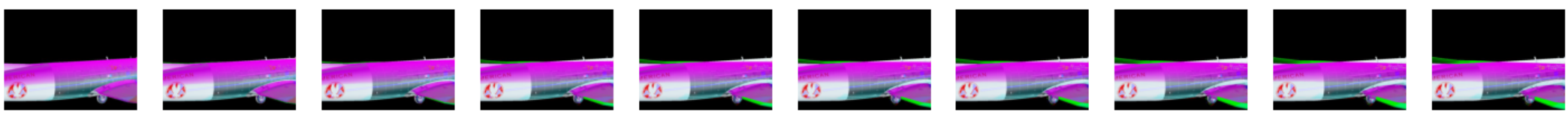
---------- sample index: 194 ----------
========== 194 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



distance: [3.2969, 2.6017, 3.1601, 4.1768, 4.7819, 5.9971, 6.2573, 7.1543, 11.9685]
deviation strategy: [0, 0, 0, -0.2, -1, -0.2]

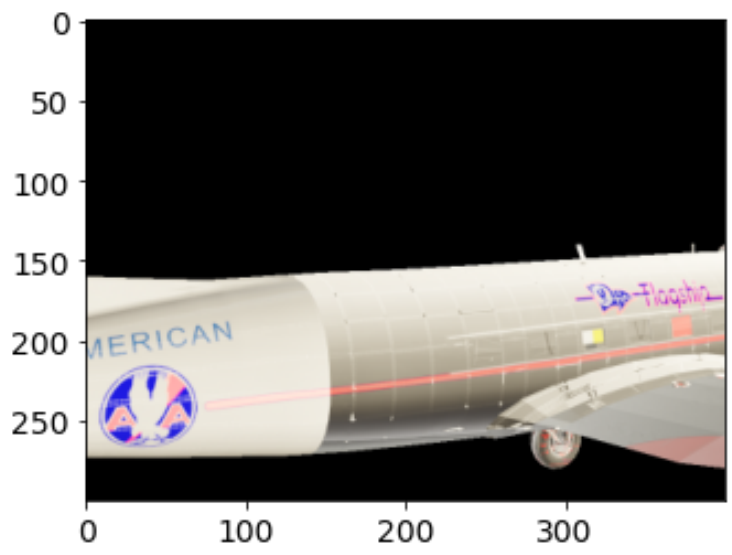
---------- sample index: 193 ----------
========== 193 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

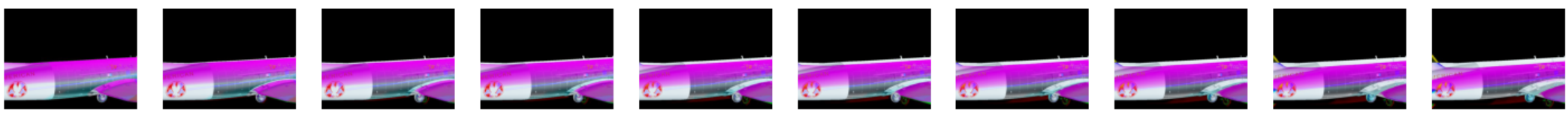




distance: [3.2969, 2.8168, 3.0633, 4.1279, 4.3665, 4.5993, 4.89, 5.1737, 5.0956]
deviation strategy: [0, 0, 0, 0, -1, 0]

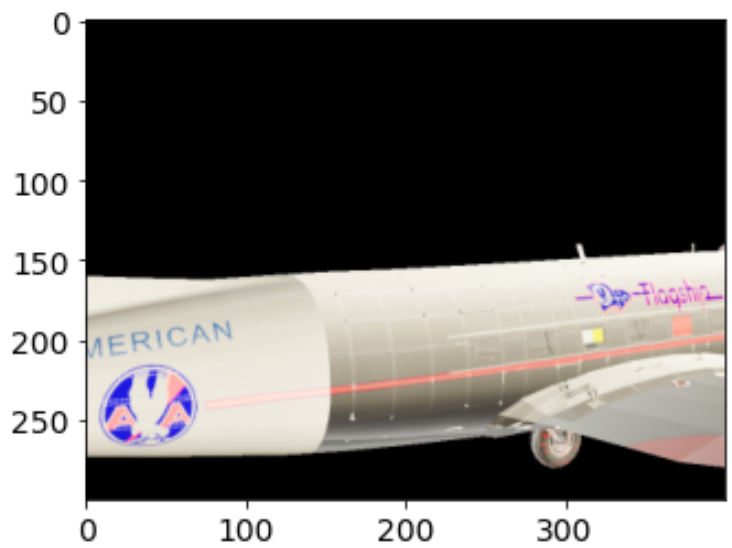
---------- sample index: 192 ----------
========== 192 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

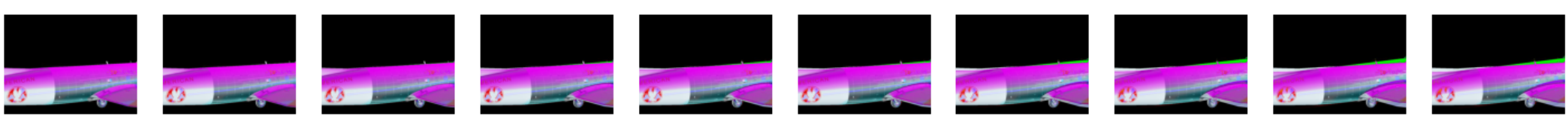




distance: [3.2969, 8.9341, 13.5769, 16.3363, 20.6305, 21.3377, 22.0537, 22.3471, 22.9813]
deviation strategy: [0, 0, 0, 0.2, 1, 0.2]

---------- sample index: 191 ----------
========== 191 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

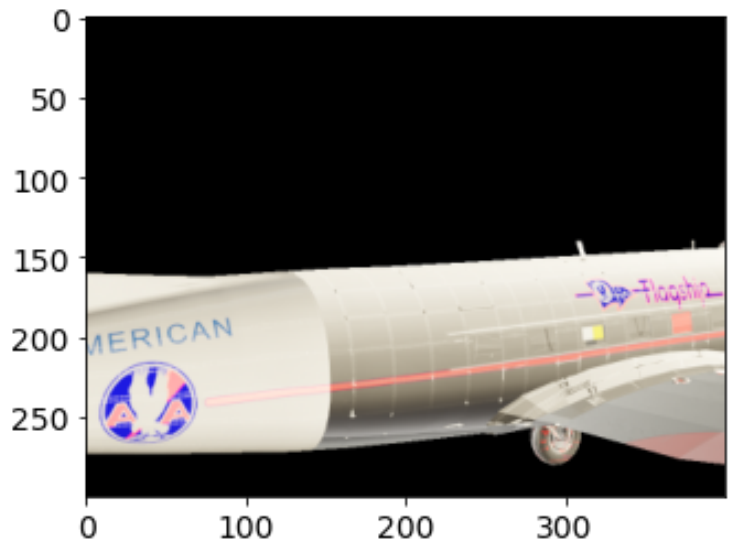


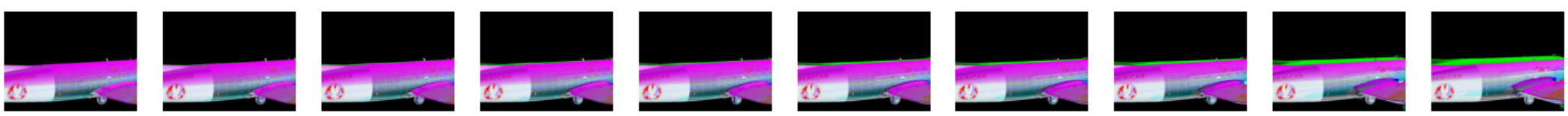


distance: [3.2969, 3.4053, 3.821, 5.0023, 5.6554, 5.6554, 7.804, 8.8261, 9.2714]
deviation strategy: [0, 0, 0, 0, 1, 0]

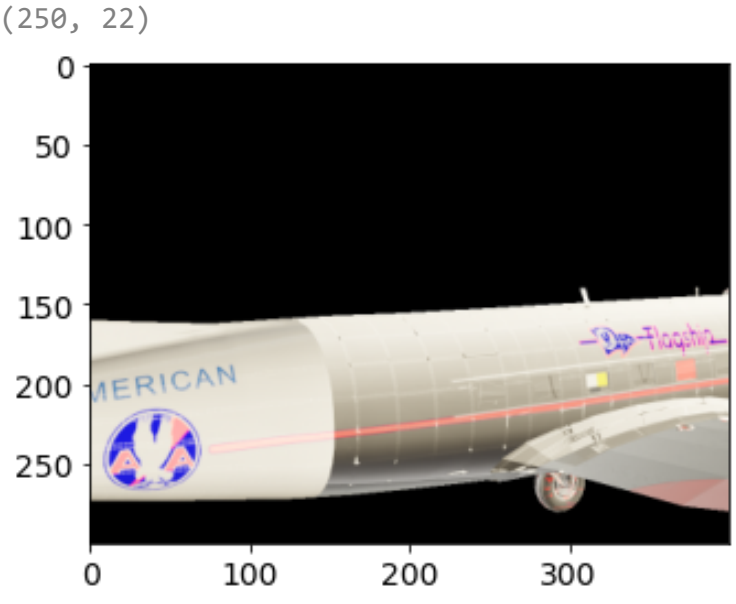
---------- sample index: 190 ----------

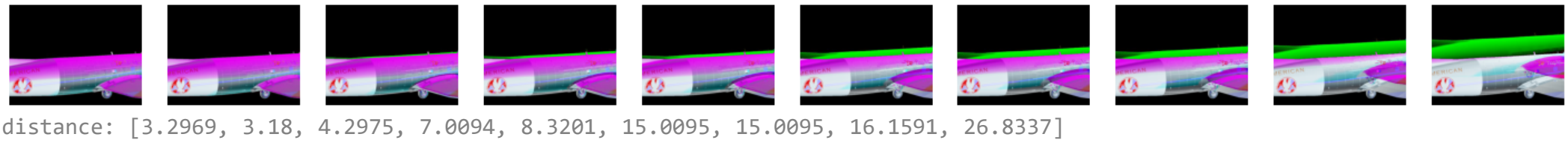
```
================= 190 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





```
distance: [3.2969, 2.6017, 3.1601, 4.1768, 4.7819, 5.9971, 6.2573, 7.1543, 11.9685]
deviation strategy: [0, 0, 0, -1, -0.2, -0.2]

---------------------- sample index: 189 ----------------------------
================= 189 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

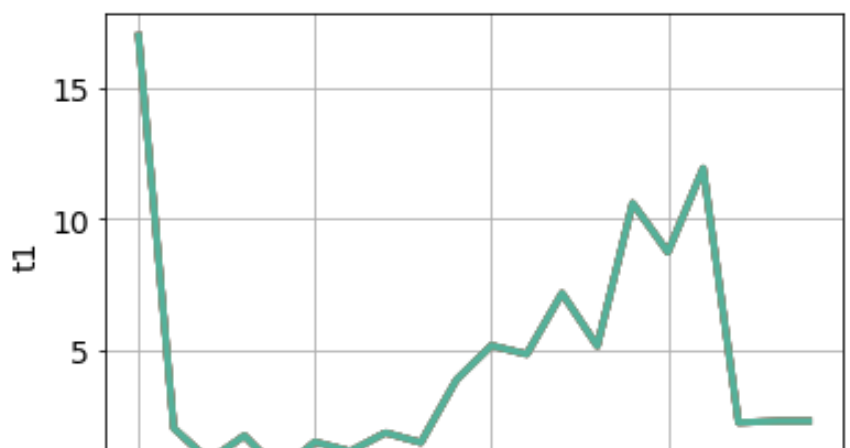


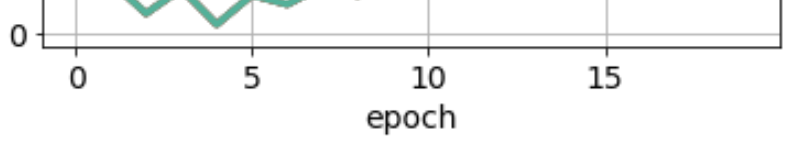


```
distance: [3.2969, 3.18, 4.2975, 7.0094, 8.3201, 15.0095, 15.0095, 16.1591, 26.8337]
deviation strategy: [0, 0, 0, -1, 0, 0]
```

Bottom 10 examples for t1

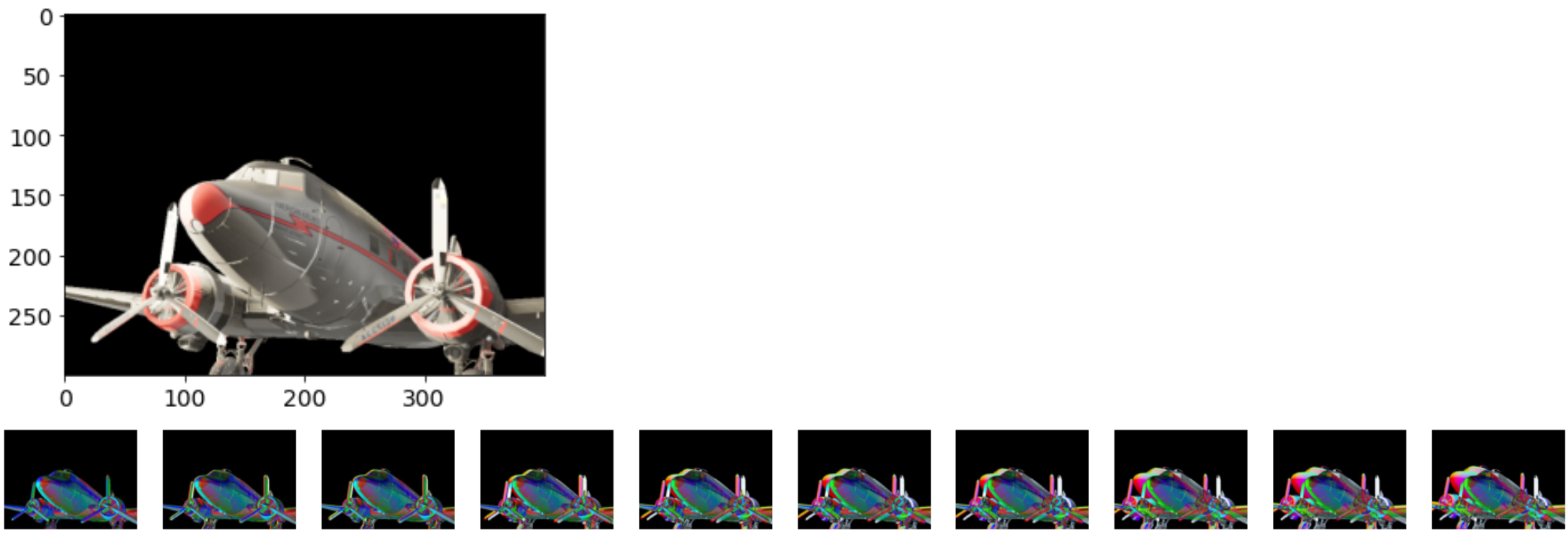
		slope	term	ex_id
batch_id	index			
16	**1**	0.186121	t1	65
	2	0.186121	t1	66
	3	0.186121	t1	67
17	**0**	0.186121	t1	68
	1	0.186121	t1	69
	2	0.186121	t1	70
	3	0.186121	t1	71
18	**0**	0.186121	t1	72
	2	0.186121	t1	74
12	**2**	0.186121	t1	50





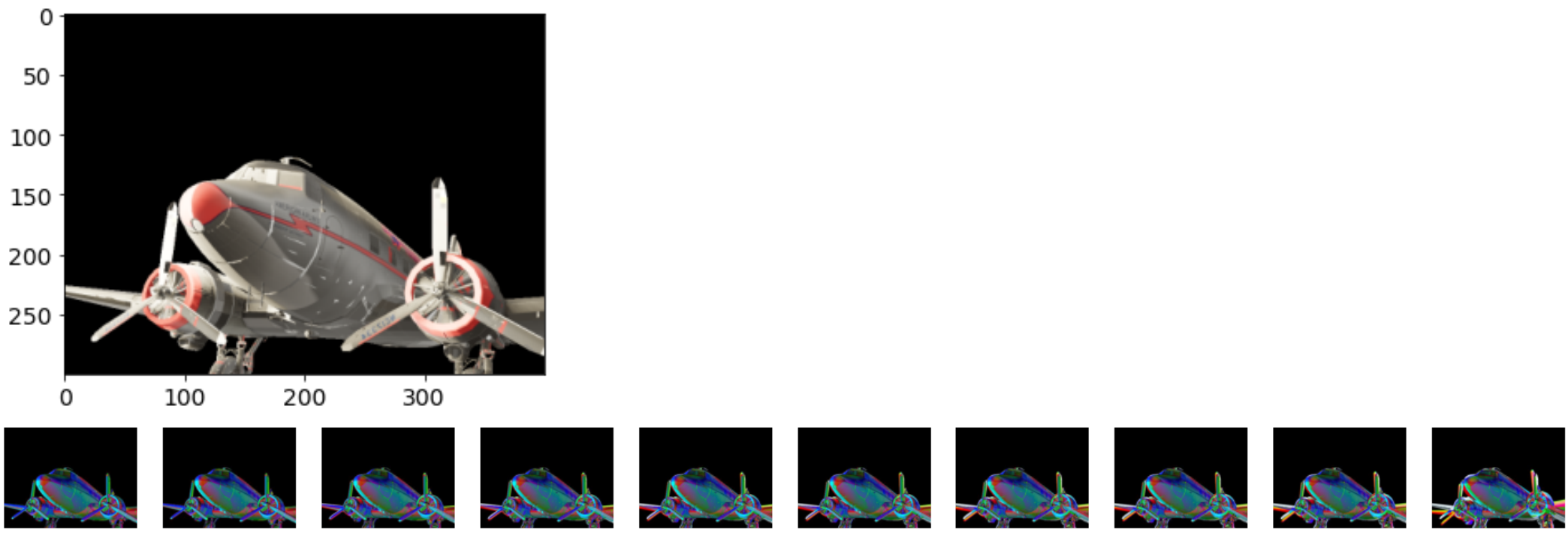
==========
mode: Validation seq_file:metadata_val_sequence.txt ex_index_list: [65 66 67 68 69 70 71 72 74 50]

---------- sample index: 65 ----------
========== 65 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



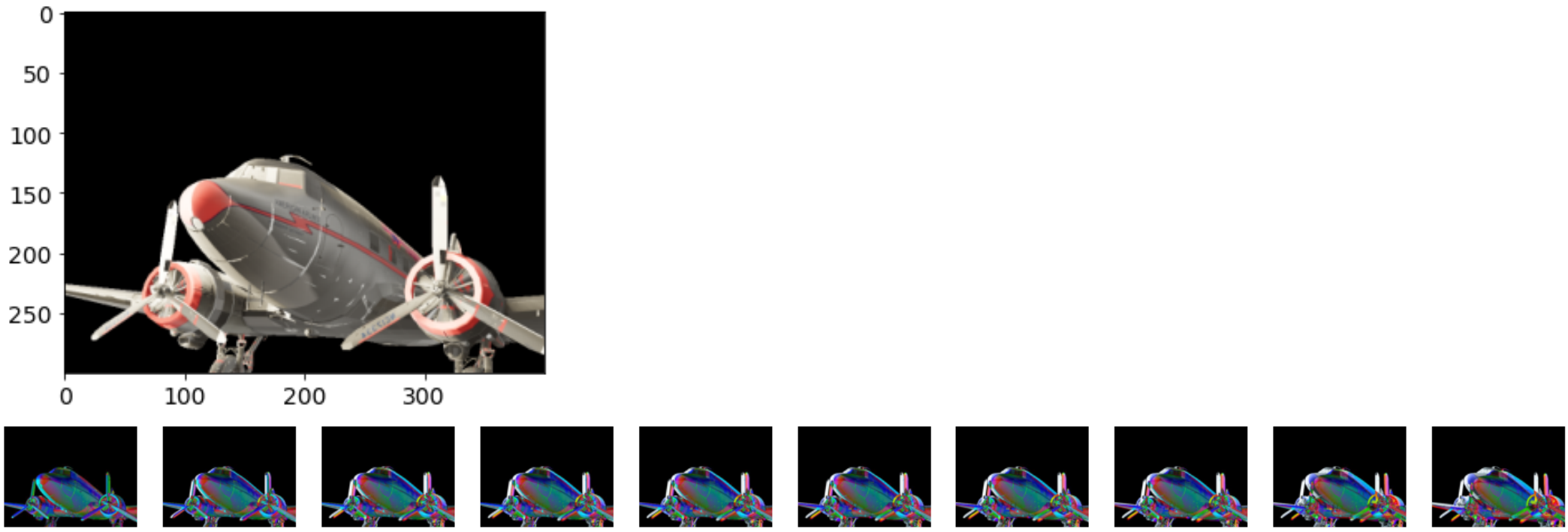
distance: [8.9701, 9.4521, 9.9999, 11.2485, 14.5238, 16.789, 17.0388, 25.8287, 29.1694]
deviation strategy: [0, 0, 0, -1, -0.2, -0.2]

---------- sample index: 66 ----------
========== 66 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



distance: [8.9701, 8.634, 8.7994, 8.5204, 8.7458, 8.7318, 8.9455, 9.0732, 8.5615]
deviation strategy: [0, 0, 0, 0, 1, 0]

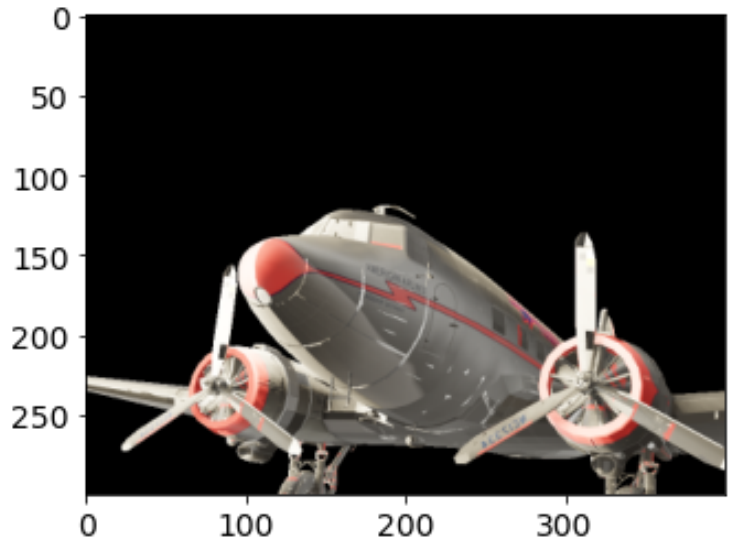
---------- sample index: 67 ----------
========== 67 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

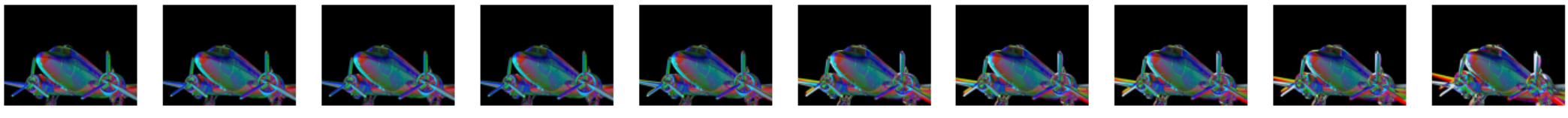


distance: [8.9701, 8.9808, 9.7468, 10.0817, 10.1696, 10.909, 11.5753, 12.5763, 17.3267]
deviation strategy: [0, 0, 0, 0.2, 1, 0.2]

---------- sample index: 68 ----------

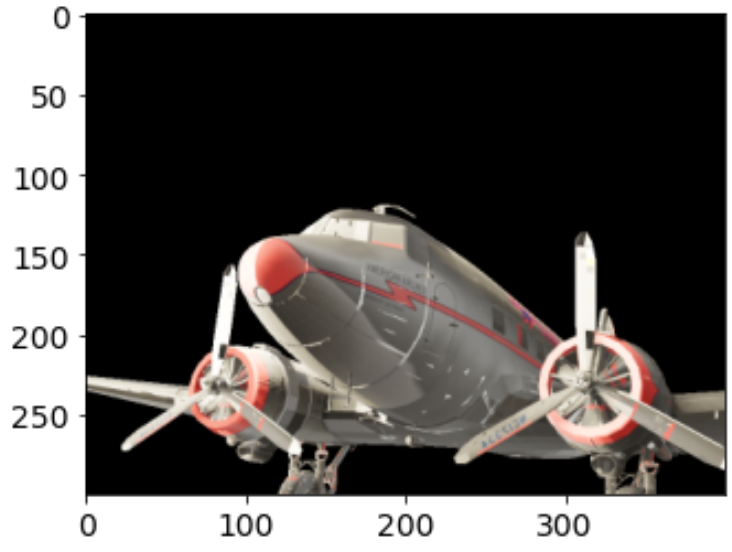
================= 68 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

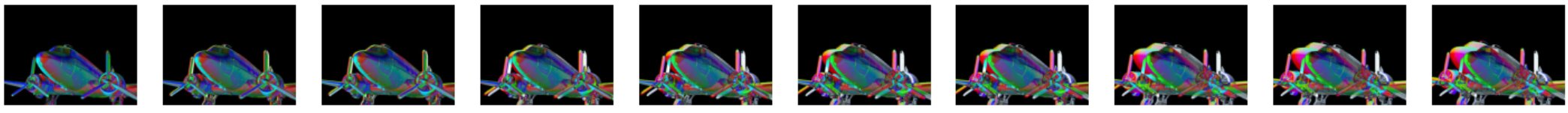




distance: [8.9701, 9.0249, 9.181, 9.1992, 8.8978, 9.0002, 9.2349, 8.9426, 8.9755]
deviation strategy: [0, 0, 0, 0, -1, 0]

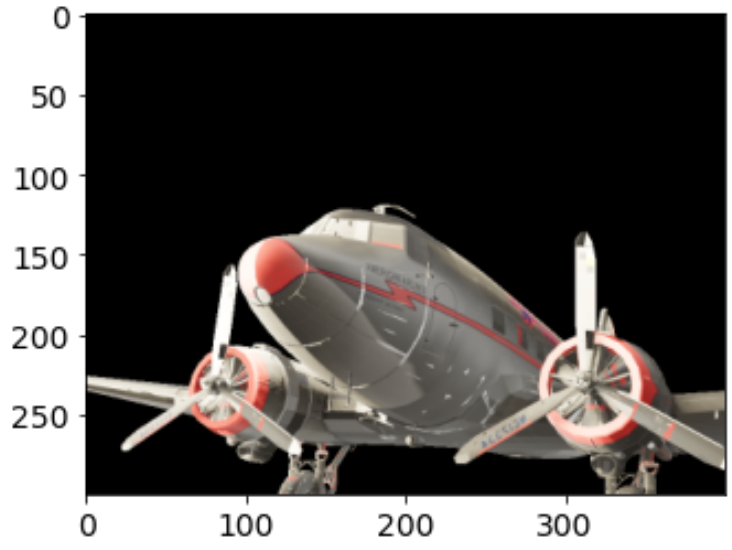
---------- sample index: 69 ----------
================= 69 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

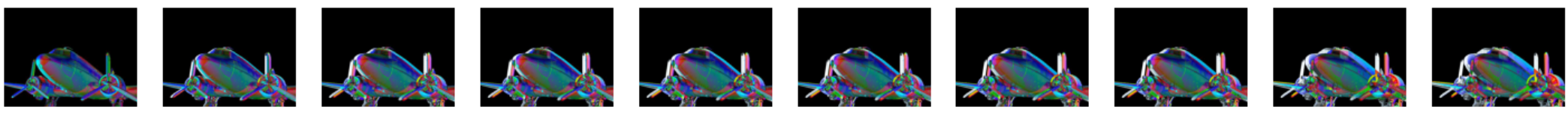




distance: [8.9701, 9.4521, 9.9999, 11.2485, 14.5238, 16.789, 17.0388, 25.8287, 29.1694]
deviation strategy: [0, 0, 0, -0.2, -1, -0.2]

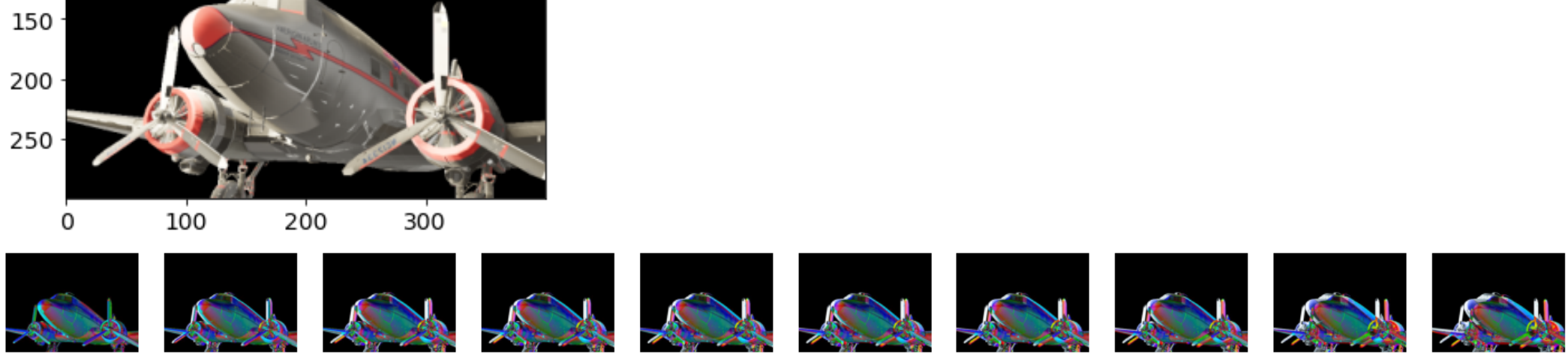
---------- sample index: 70 ----------
================= 70 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



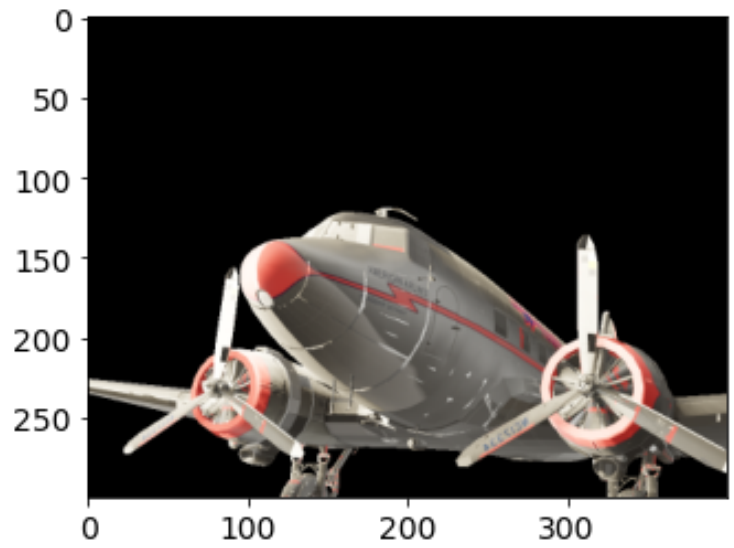


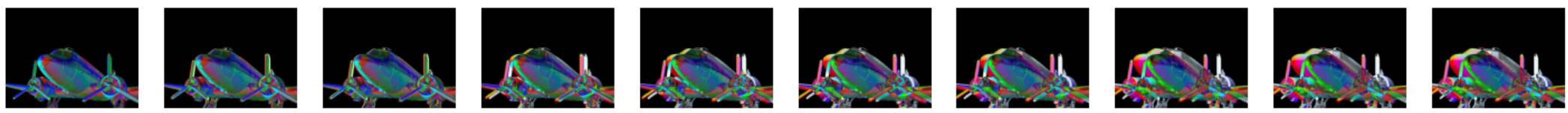
distance: [8.9701, 8.98, 10.0514, 9.9741, 9.5782, 9.7671, 9.2486, 9.6879, 13.387]
deviation strategy: [0, 0, 0, 0, 0, 1]

---------- sample index: 71 ----------
================= 71 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

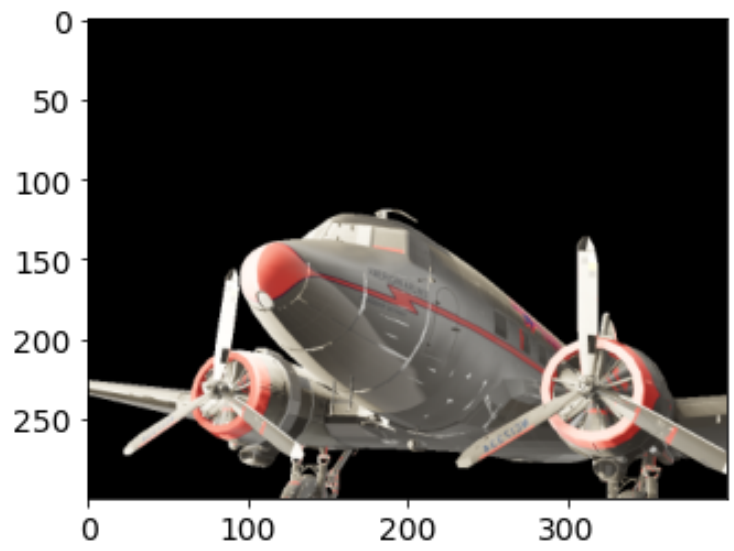


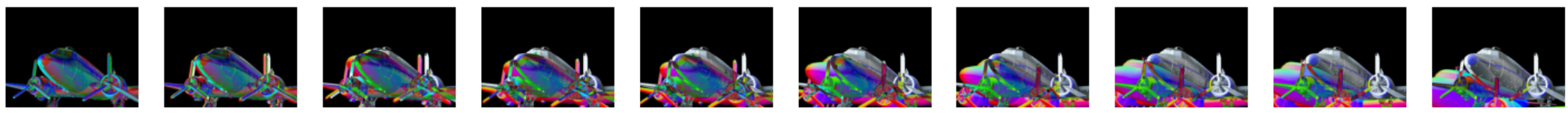
```
distance: [8.9701, 8.9808, 9.7468, 10.0817, 10.1696, 10.909, 11.5753, 12.5763, 17.3267]
deviation strategy: [0, 0, 0, 0.2, 0.2, 1]

---------------------- sample index: 72 ----------------------------
================= 72 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```



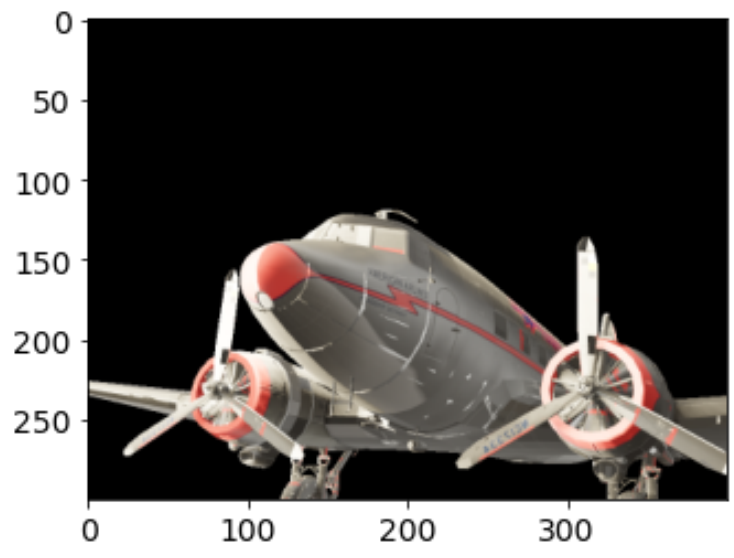


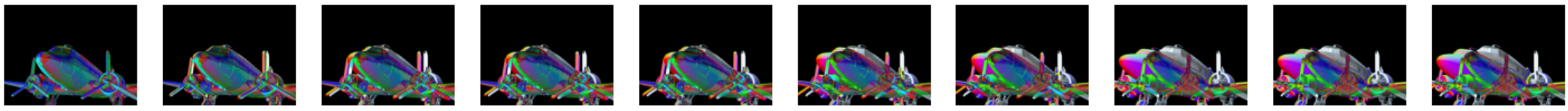
```
distance: [8.9701, 9.5053, 9.5405, 11.6466, 12.4322, 14.8985, 15.2624, 21.3582, 23.1164]
deviation strategy: [0, 0, 0, 0, 0, -1]

---------------------- sample index: 74 ----------------------------
================= 74 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





```
distance: [8.9701, 8.7915, 12.893, 18.9297, 20.4271, 25.7674, 31.2939, 35.6951, 36.3422]
deviation strategy: [1, 1, 1, 0, 0, 0]

---------------------- sample index: 50 ----------------------------
================= 50 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

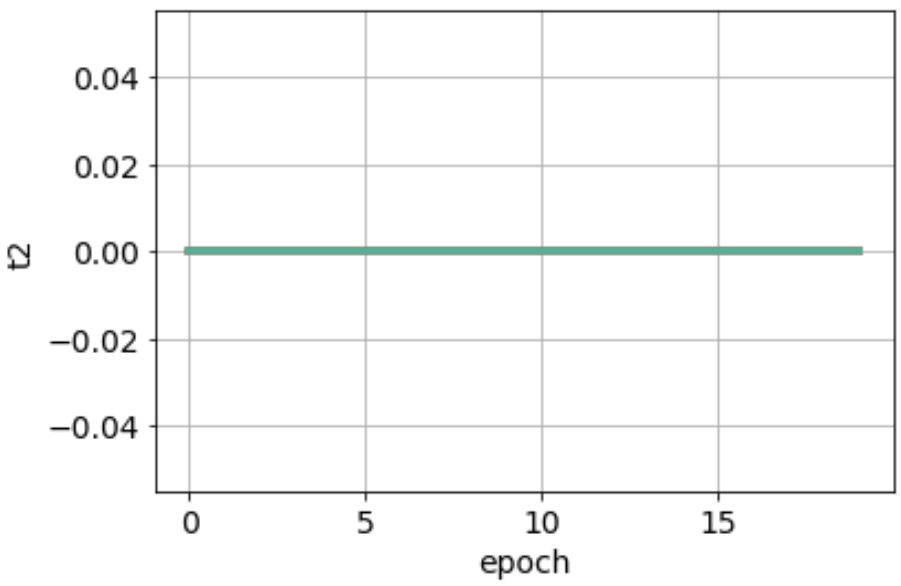




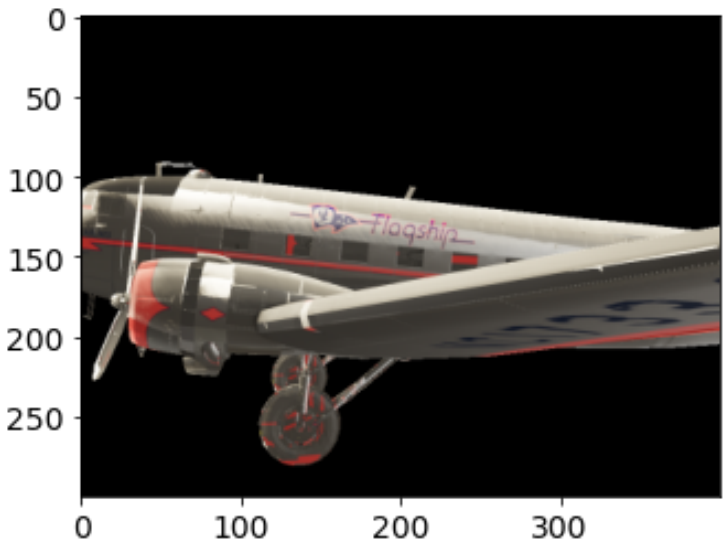
```
distance: [8.9701, 9.7566, 12.379, 12.3002, 12.8421, 17.7469, 18.3011, 25.0953, 25.0953]
deviation strategy: [1, 0, 0, 0, 0, 0]
```

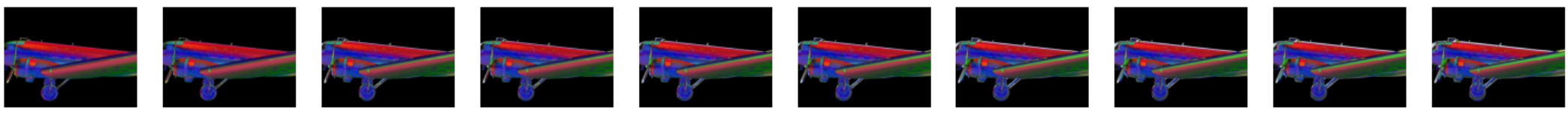
Top 10 examples for t2

batch_id	index	slope	term	ex_id
0	**0**	0.0	t2	0
39	**2**	0.0	t2	158
	3	0.0	t2	159
40	**0**	0.0	t2	160
	1	0.0	t2	161
	2	0.0	t2	162
	3	0.0	t2	163
41	**0**	0.0	t2	164
	1	0.0	t2	165
	2	0.0	t2	166

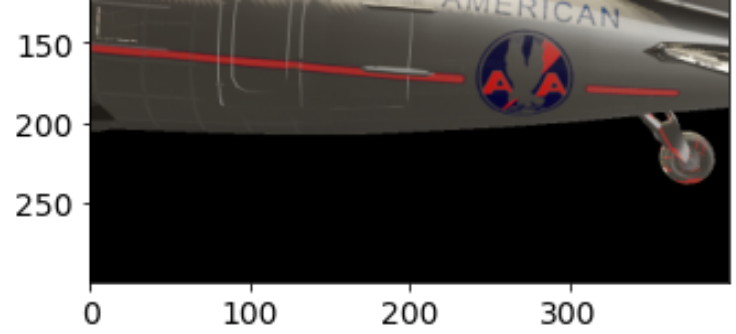


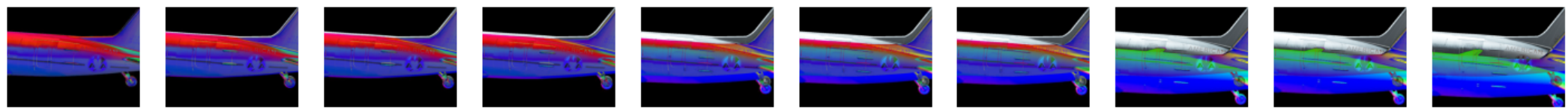

```
==================================
mode: Validation  seq_file:metadata_val_sequence.txt  ex_index_list: [  0 158 159 160 161 162 163 164 165 166]

----------------------- sample index: 0 ---------------------------
================= 0 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





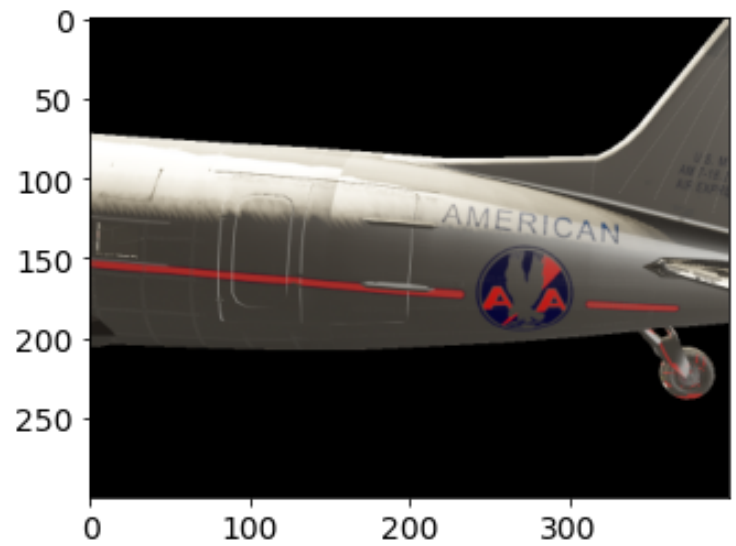
```
distance: [4.4313, 4.2737, 4.6018, 4.5923, 4.7794, 4.8696, 4.7472, 4.9388, 5.279]
deviation strategy: [1, 0, 0, 0, 0, 0]

----------------------- sample index: 158 ---------------------------
================= 158 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

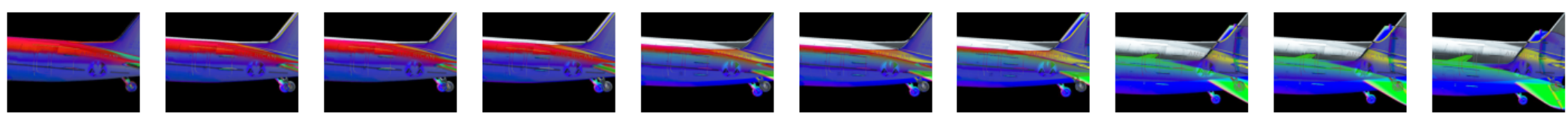




distance: [2.7835, 1.7815, 4.3381, 9.1352, 23.3128, 23.8482, 25.3206, 34.2048, 34.2048]
deviation strategy: [0, 0, 1, 0, 0, 0]

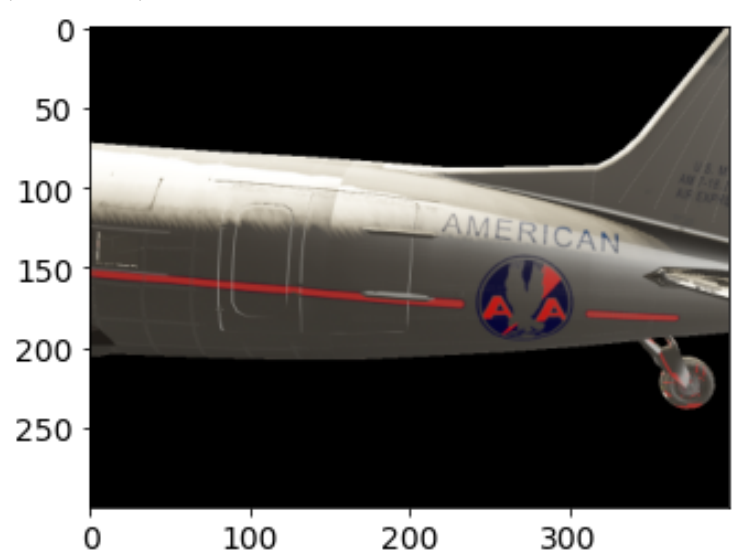
---------- sample index: 159 ----------
========== 159 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

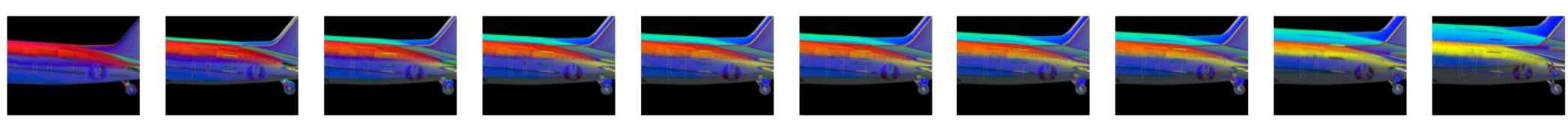




distance: [2.7835, 4.2458, 6.9853, 13.1842, 24.3567, 26.1725, 28.8151, 35.1757, 35.5796]
deviation strategy: [0.2, 0.2, 1, 0, 0, 0]

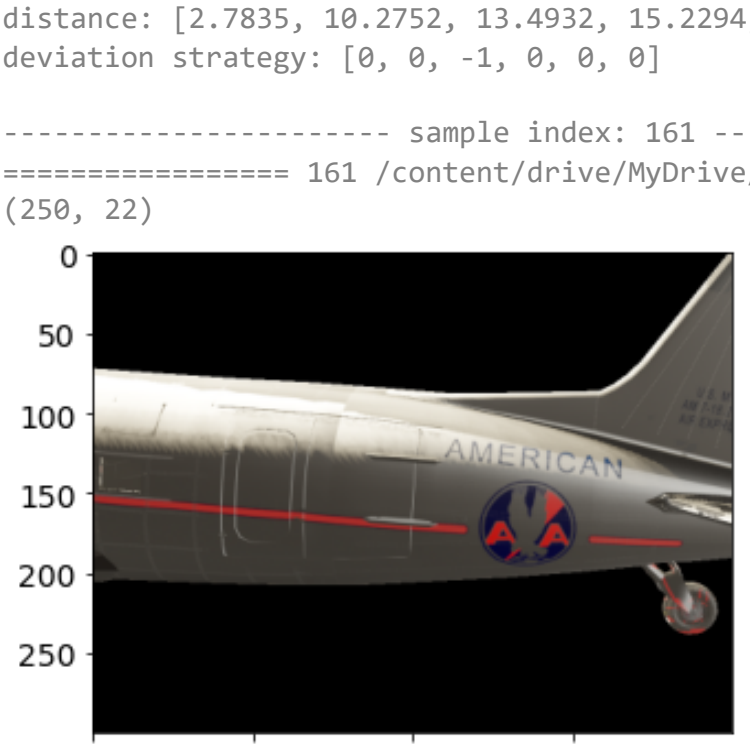
---------- sample index: 160 ----------
========== 160 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

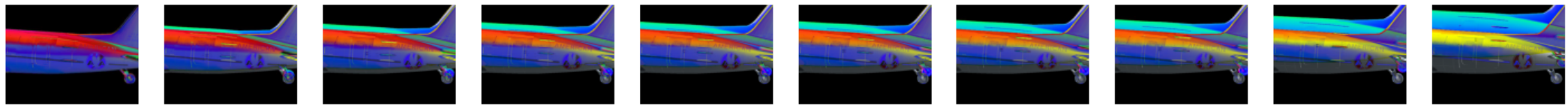




distance: [2.7835, 10.2752, 13.4932, 15.2294, 15.9098, 16.1073, 16.5643, 16.6824, 18.8499]
deviation strategy: [0, 0, -1, 0, 0, 0]

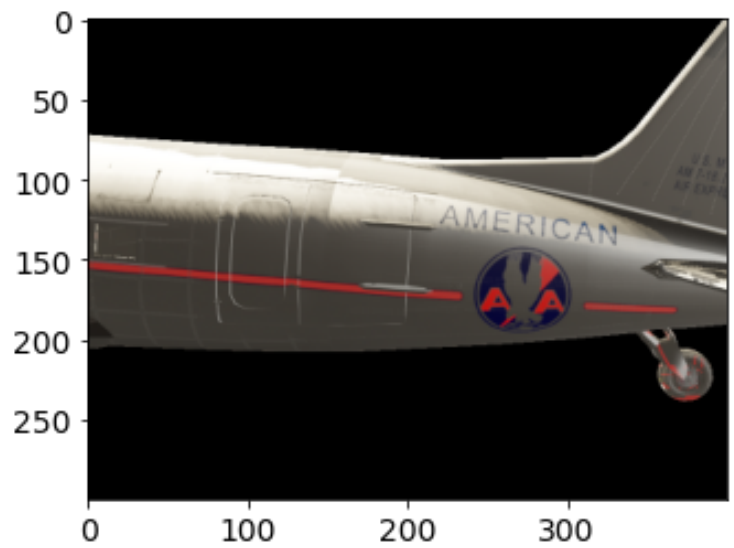
---------- sample index: 161 ----------
========== 161 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

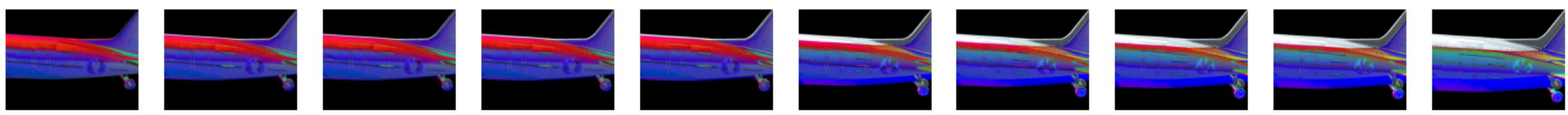




distance: [2.7835, 10.358, 13.8229, 15.3754, 16.1732, 16.1431, 16.7824, 17.1235, 19.5056]
deviation strategy: [-0.2, -0.2, -1, 0, 0, 0]

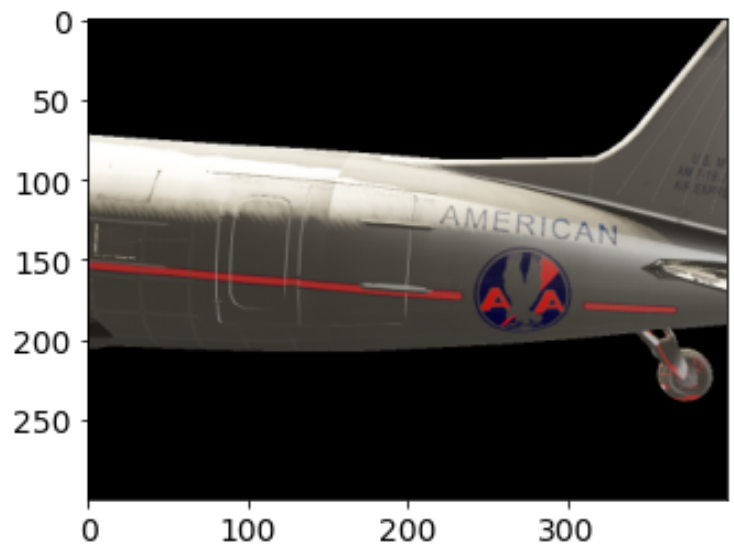
---------- sample index: 162 ----------
========== 162 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

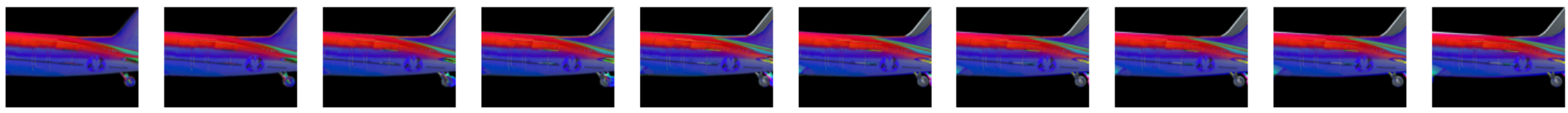




distance: [2.7835, 1.7272, 1.7329, 2.435, 3.9163, 21.5811, 23.9015, 26.0324, 27.3218]
deviation strategy: [0, 0, 0, 1, 0, 0]

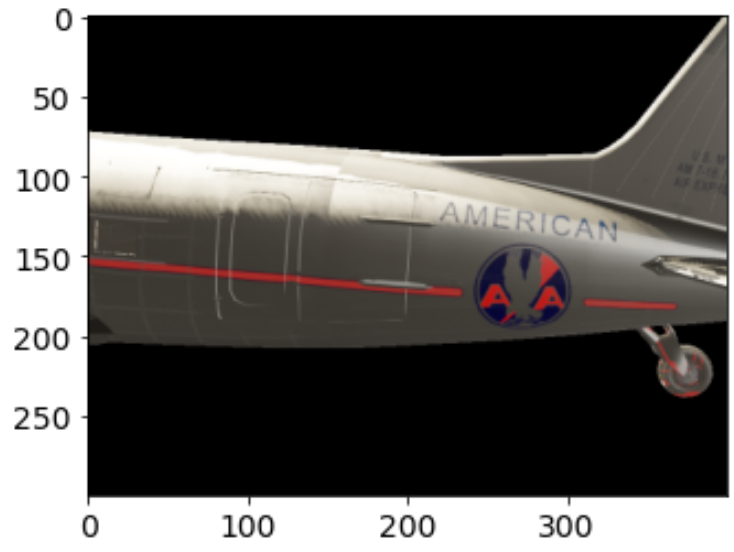
---------- sample index: 163 ----------
========== 163 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

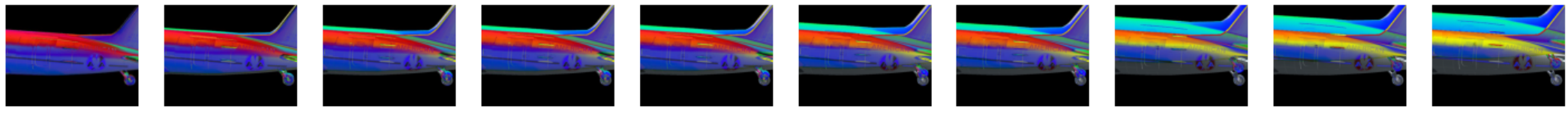




distance: [2.7835, 2.6896, 3.3801, 3.6005, 4.622, 3.6825, 3.5904, 3.2595, 3.3598]
deviation strategy: [0, 0, 0, 1, 0.2, 0.2]

---------- sample index: 164 ----------
========== 164 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

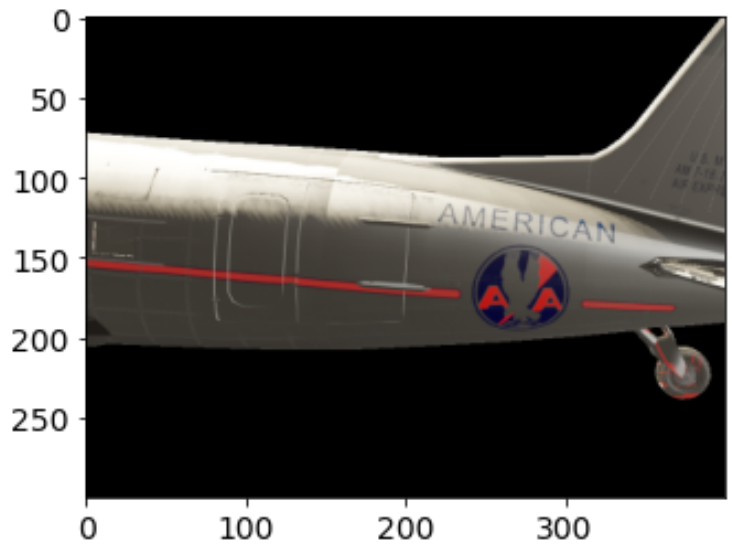


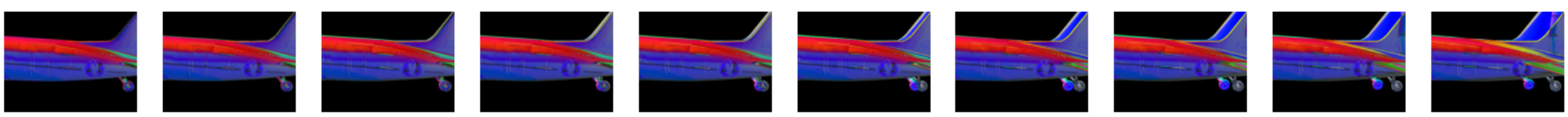


distance: [2.7835, 6.6444, 11.25, 12.1333, 12.6092, 15.2194, 15.2194, 18.8322, 19.5813]
deviation strategy: [0, 0, 0, -1, 0, 0]

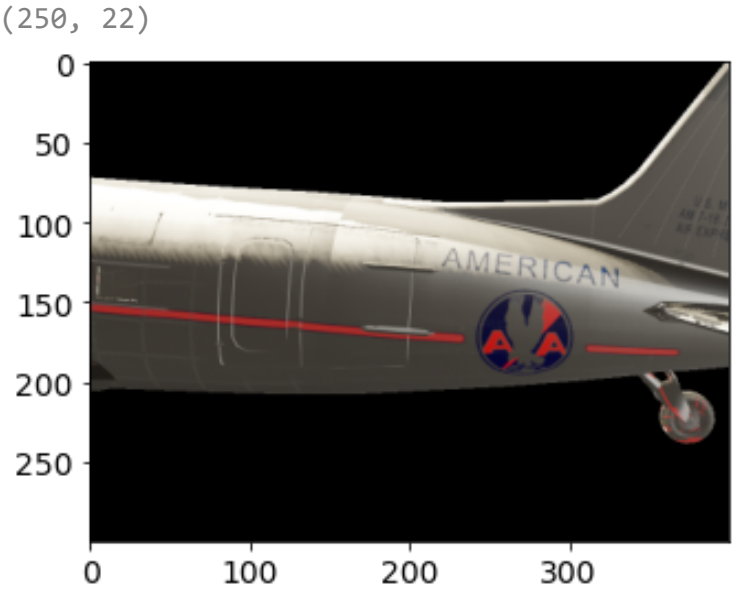
---------- sample index: 165 ----------

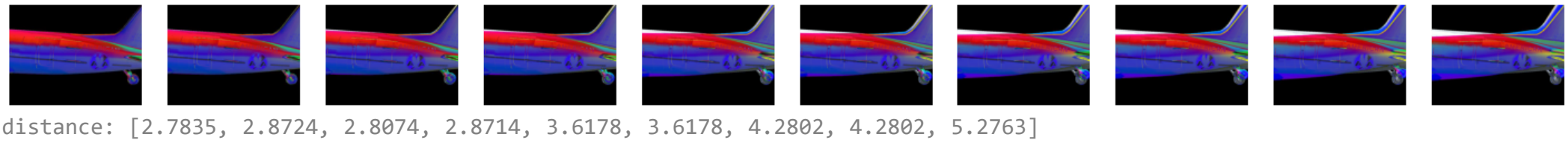
```
================= 165 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





```
distance: [2.7835, 3.1811, 3.6643, 3.5168, 3.5539, 3.3448, 3.0754, 4.7734, 5.1081]
deviation strategy: [0, 0, 0, -1, -0.2, -0.2]

---------------------- sample index: 166 ---------------------------
================= 166 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

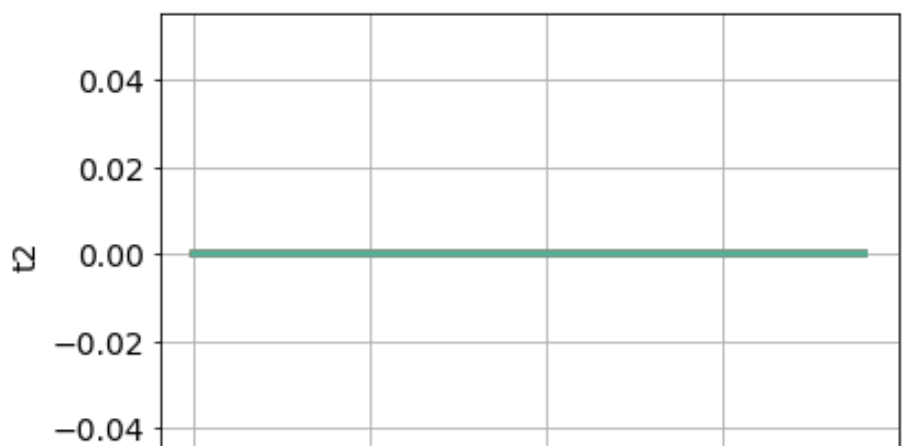


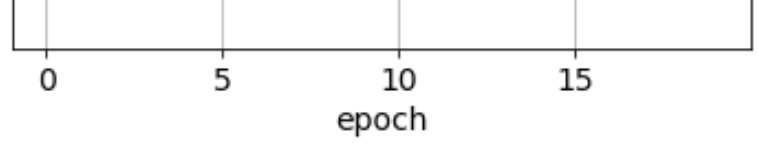


```
distance: [2.7835, 2.8724, 2.8074, 2.8714, 3.6178, 3.6178, 4.2802, 4.2802, 5.2763]
deviation strategy: [0, 0, 0, 0, 1, 0]
```

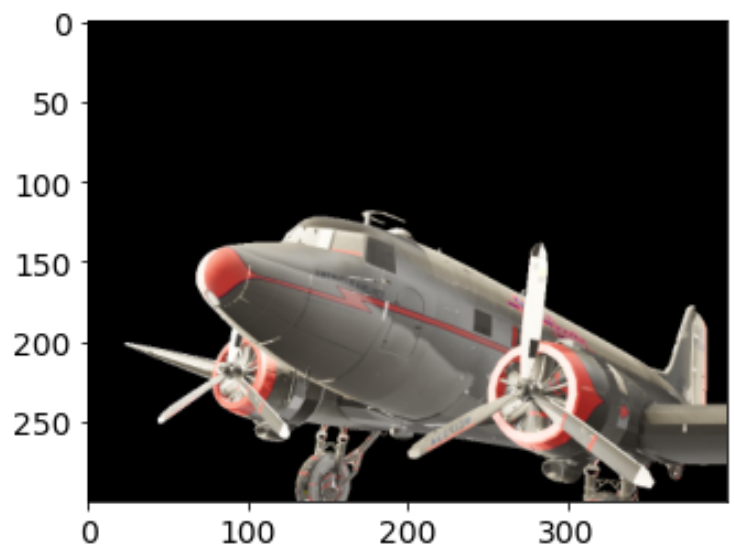
Bottom 10 examples for t2

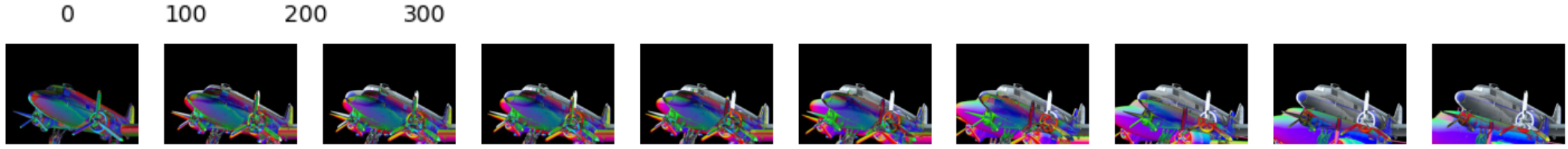
		slope	term	ex_id
batch_id	index			
21	**0**	0.0	t2	84
	1	0.0	t2	85
	2	0.0	t2	86
	3	0.0	t2	87
22	**0**	0.0	t2	88
	1	0.0	t2	89
	2	0.0	t2	90
	3	0.0	t2	91
19	**2**	0.0	t2	78
62	**1**	0.0	t2	249



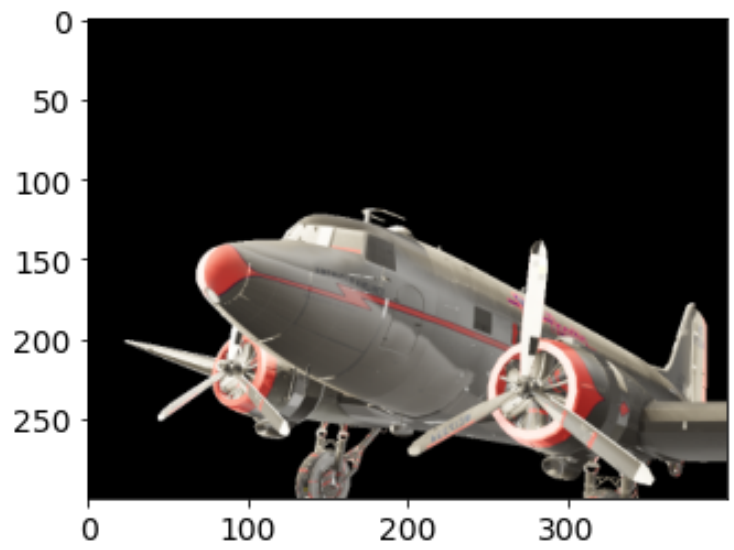


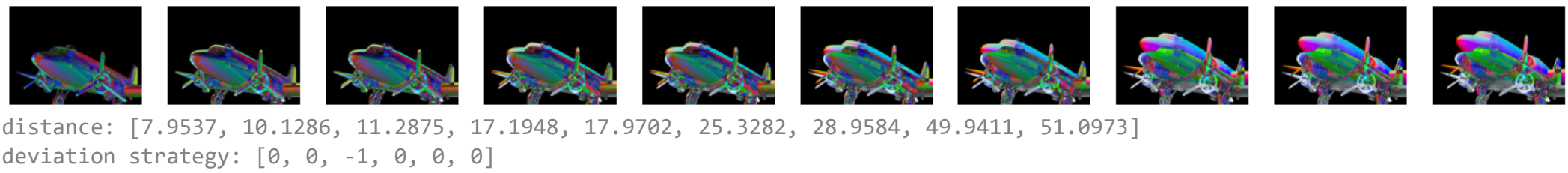

```
==================================
mode: Validation  seq_file:metadata_val_sequence.txt  ex_index_list: [ 84  85  86  87  88  89  90  91  78 249]

---------------------- sample index: 84 --------------------------
================= 84 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```



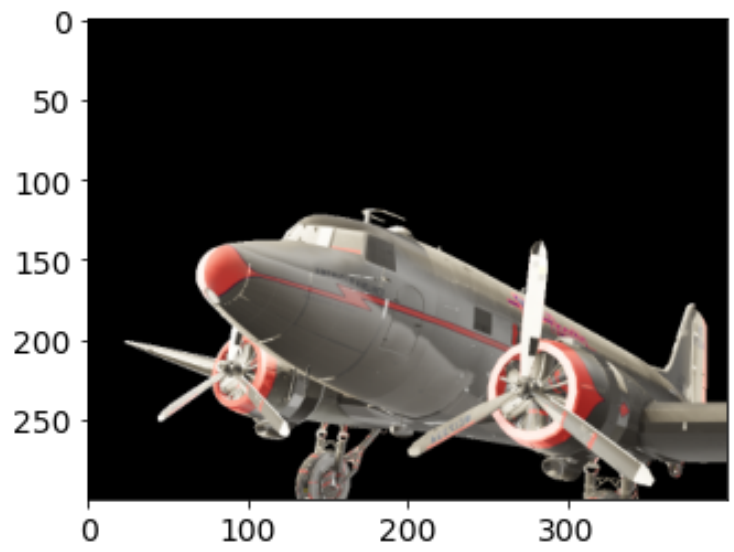


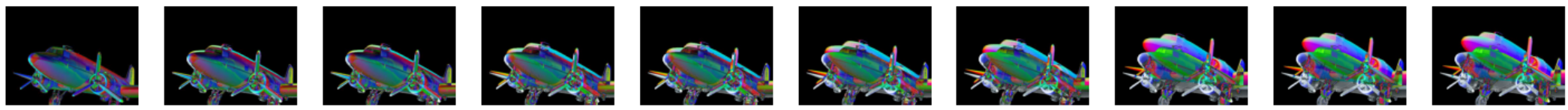
```
distance: [7.9537, 11.3489, 16.2355, 15.5197, 17.9712, 19.3302, 23.3051, 33.6556, 35.655]
deviation strategy: [0.2, 0.2, 1, 0, 0, 0]

---------------------- sample index: 85 --------------------------
================= 85 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





```
distance: [7.9537, 10.1286, 11.2875, 17.1948, 17.9702, 25.3282, 28.9584, 49.9411, 51.0973]
deviation strategy: [0, 0, -1, 0, 0, 0]

---------------------- sample index: 86 --------------------------
================= 86 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





```
distance: [7.9537, 9.1631, 9.0001, 13.4873, 14.2305, 22.6421, 25.8067, 46.6696, 48.9828]
deviation strategy: [-0.2, -0.2, -1, 0, 0, 0]

---------------------- sample index: 87 --------------------------
```

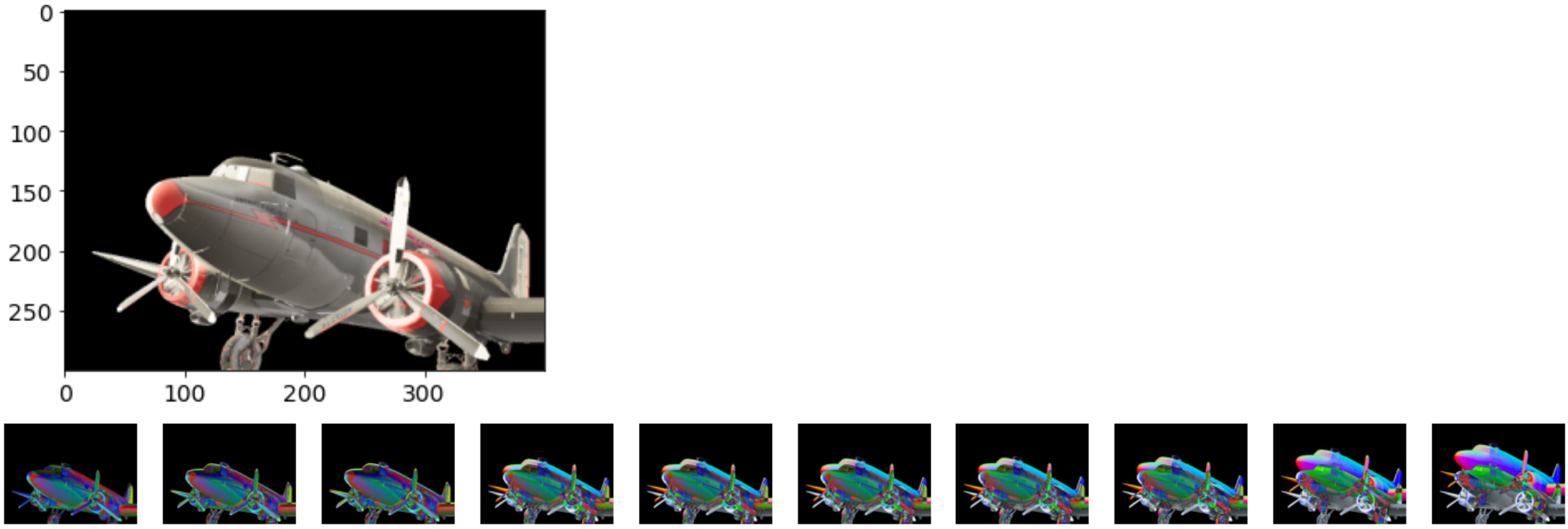
================= 87 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

distance: [7.9537, 8.9914, 9.8306, 11.04, 15.8296, 20.9703, 38.9206, 40.9043, 40.8223]
deviation strategy: [0, 0, 0, 1, 0, 0]

---------------------- sample index: 88 ----------------------------
================= 88 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

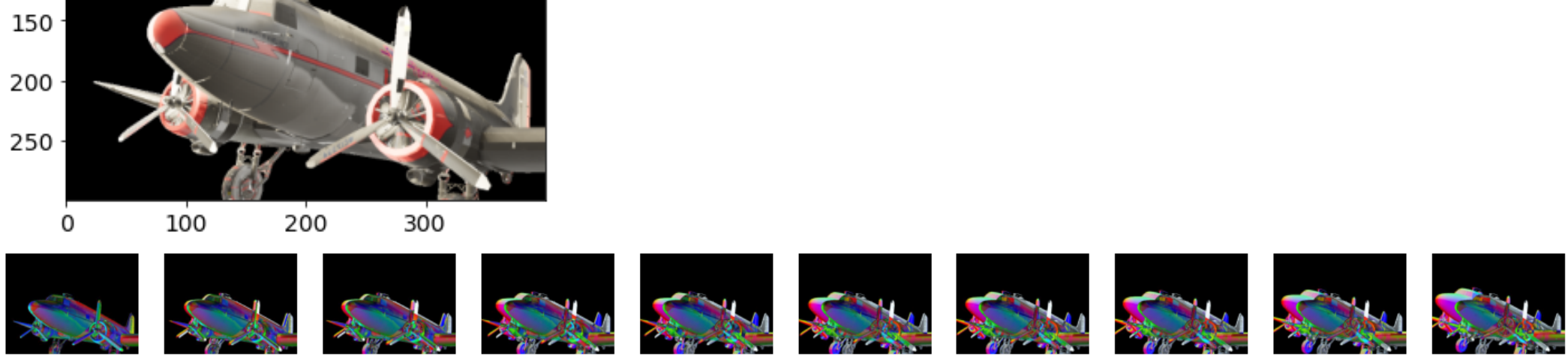
distance: [7.9537, 7.684, 6.8351, 8.1887, 9.9451, 11.3372, 16.6574, 18.5097, 19.8573]
deviation strategy: [0, 0, 0, 1, 0.2, 0.2]

---------------------- sample index: 89 ----------------------------
================= 89 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

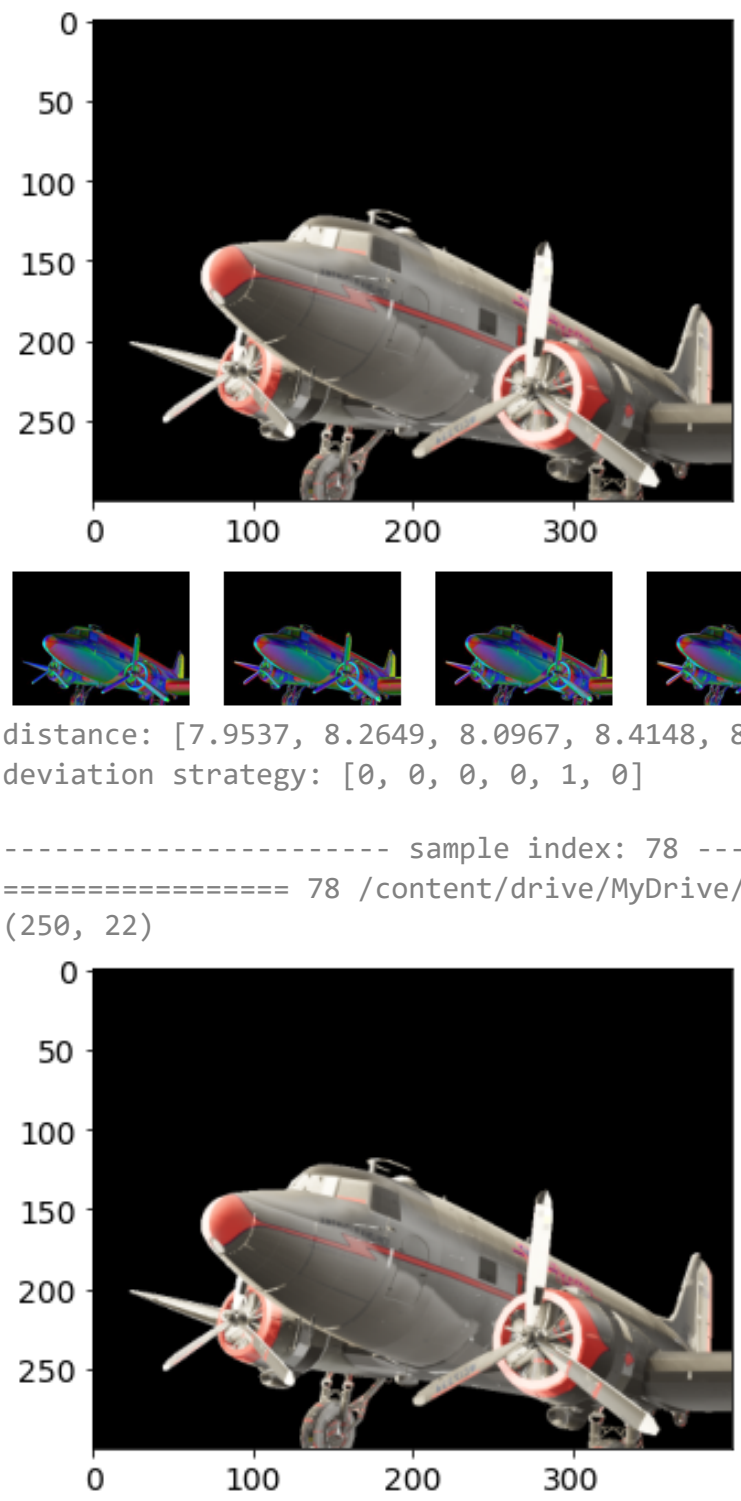


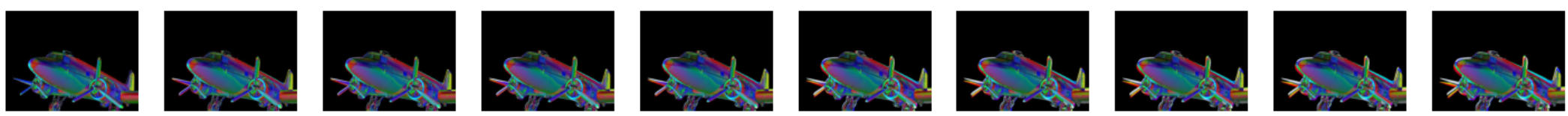
distance: [7.9537, 9.4608, 10.1732, 22.5296, 22.7321, 24.387, 26.0167, 28.8009, 51.2231]
deviation strategy: [0, 0, 0, -1, 0, 0]

---------------------- sample index: 90 ----------------------------
================= 90 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

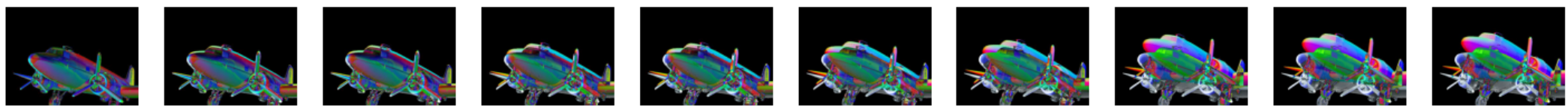


```
distance: [7.9537, 10.8811, 12.0369, 17.3601, 19.5378, 23.0688, 24.2378, 25.2713, 29.2626]
deviation strategy: [0, 0, 0, -1, -0.2, -0.2]

---------------------- sample index: 91 ----------------------------
================= 91 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

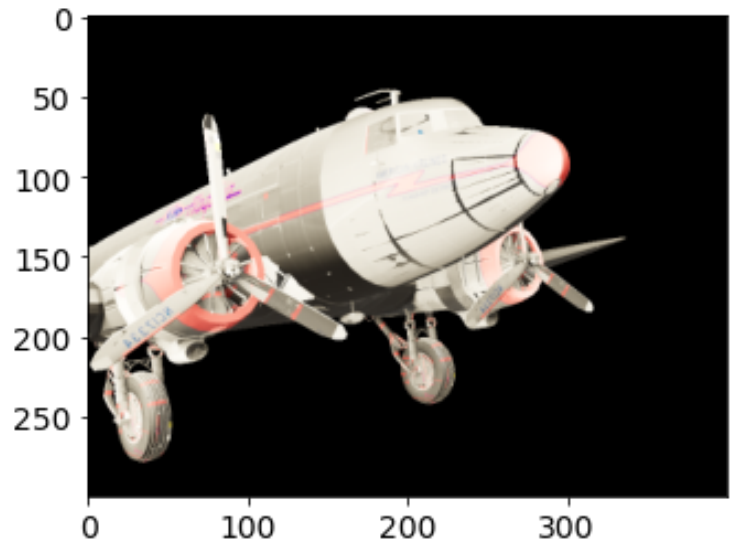


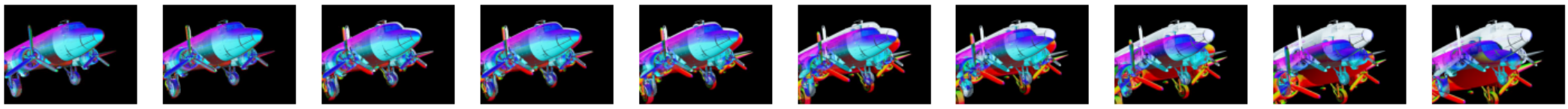


```
distance: [7.9537, 8.2649, 8.0967, 8.4148, 8.3005, 8.4319, 8.14, 7.7011, 7.7536]
deviation strategy: [0, 0, 0, 0, 1, 0]

---------------------- sample index: 78 ----------------------------
================= 78 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

```
distance: [7.9537, 9.1631, 9.0001, 13.4873, 14.2305, 22.6421, 25.8067, 46.6696, 48.9828]
deviation strategy: [-1, -0.2, -0.2, 0, 0, 0]

---------------------- sample index: 249 ----------------------------
================= 249 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

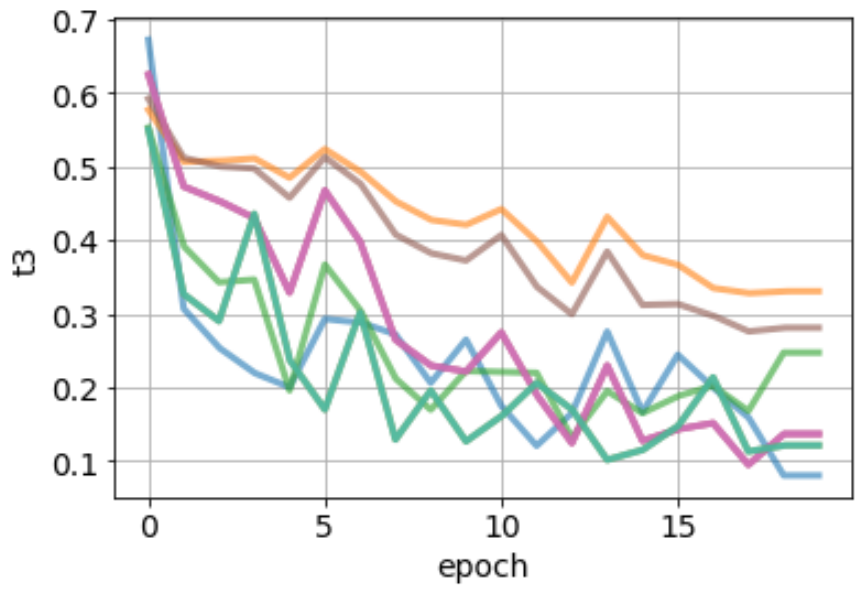




distance: [5.3699, 5.1047, 5.9724, 6.4179, 9.314, 21.4777, 23.8757, 43.3226, 56.2134]
deviation strategy: [1, 1, 1, 0, 0, 0]

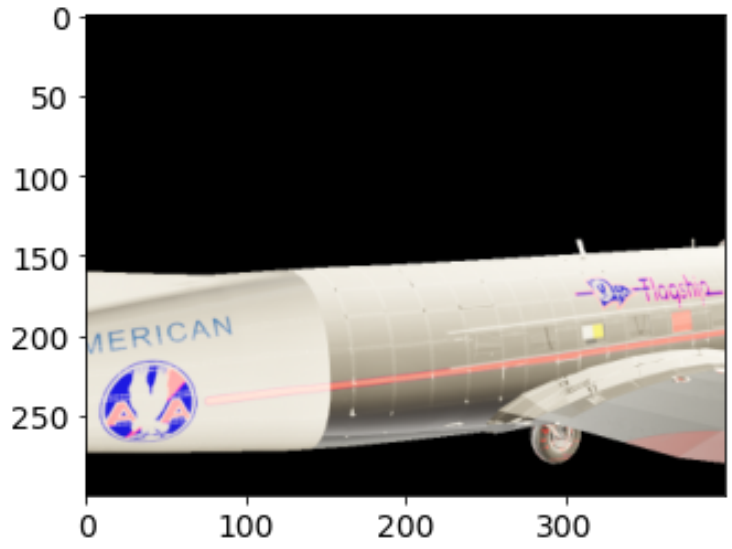
Top 10 examples for t3

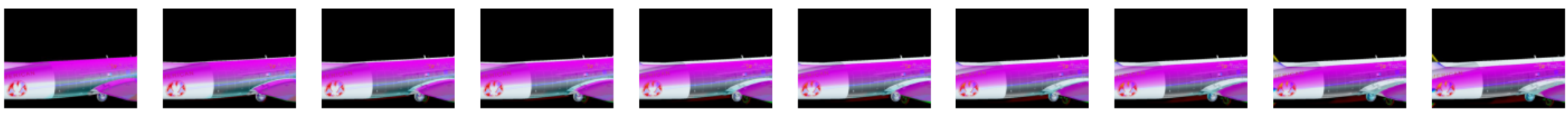
batch_id	index	slope	term	ex_id
48	**0**	-0.023838	t3	192
49	**0**	-0.023838	t3	196
47	**0**	-0.023838	t3	188
48	**3**	-0.015691	t3	195
61	**2**	-0.014602	t3	246
60	**2**	-0.014602	t3	242
59	**2**	-0.014602	t3	238
24	**1**	-0.012892	t3	97
45	**3**	-0.012583	t3	183
44	**3**	-0.012523	t3	179



==========
mode: Validation seq_file:metadata_val_sequence.txt ex_index_list: [192 196 188 195 246 242 238 97 183 179]

---------- sample index: 192 ----------
========== 192 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)





distance: [3.2969, 8.9341, 13.5769, 16.3363, 20.6305, 21.3377, 22.0537, 22.3471, 22.9813]
deviation strategy: [0, 0, 0, 0.2, 1, 0.2]

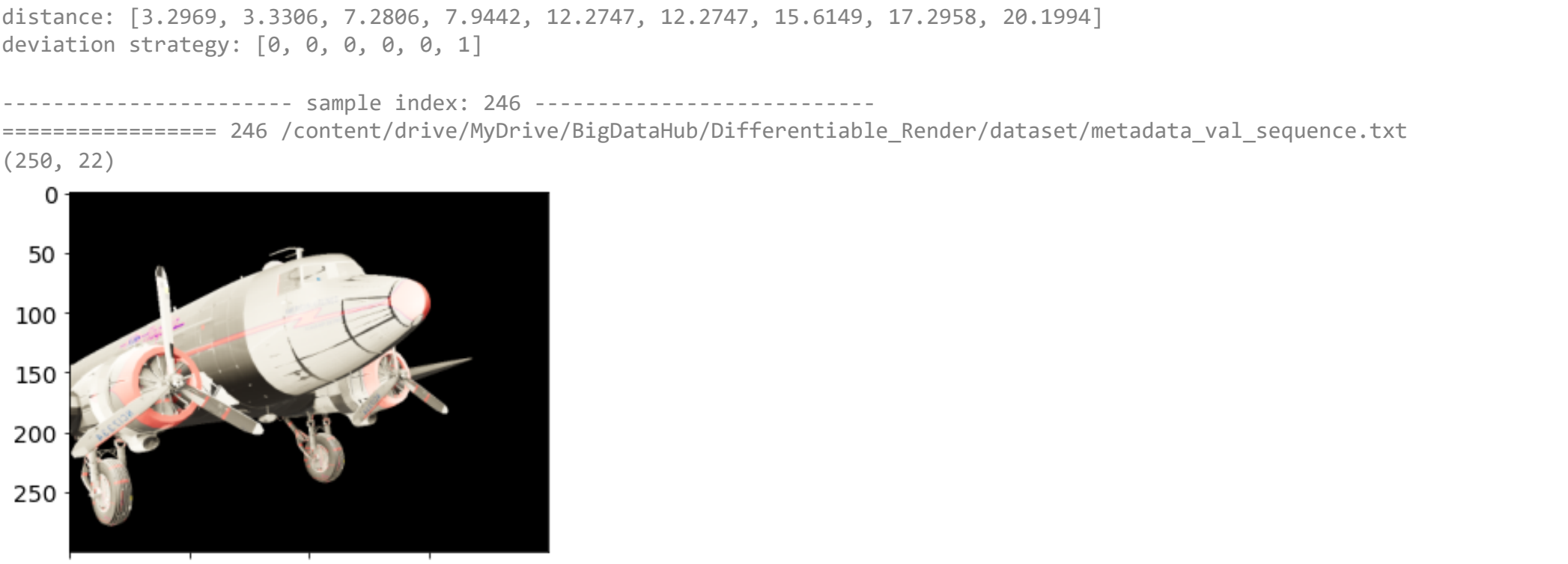
---------- sample index: 196 ----------
========== 196 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

distance: [3.2969, 8.9341, 13.5769, 16.3363, 20.6305, 21.3377, 22.0537, 22.3471, 22.9813]
deviation strategy: [0, 0, 0, 0.2, 0.2, 1]

---------- sample index: 188 ----------
========== 188 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

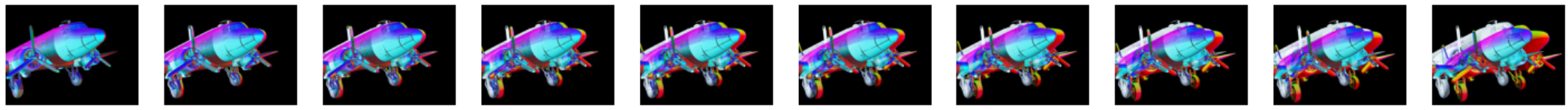
distance: [3.2969, 8.9341, 13.5769, 16.3363, 20.6305, 21.3377, 22.0537, 22.3471, 22.9813]
deviation strategy: [0, 0, 0, 1, 0.2, 0.2]

---------- sample index: 195 ----------
========== 195 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

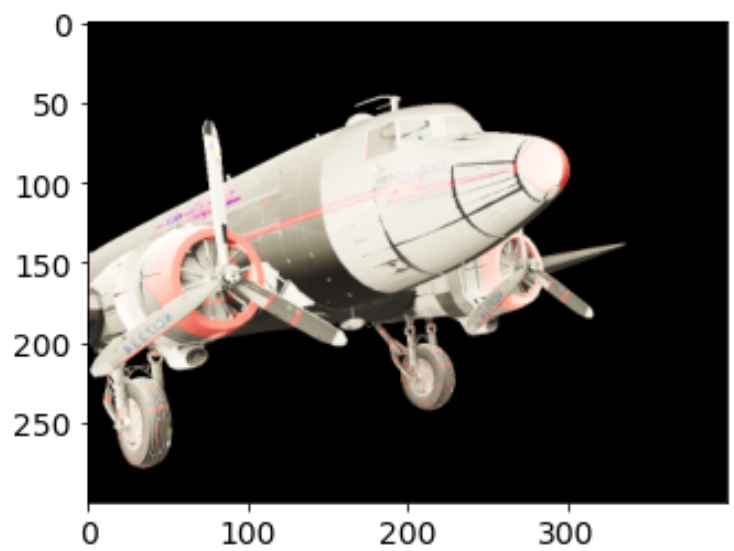


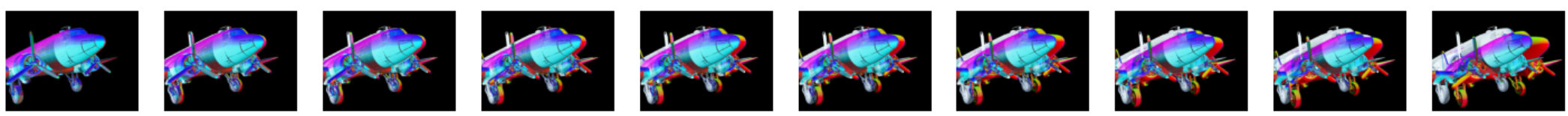
distance: [3.2969, 3.3306, 7.2806, 7.9442, 12.2747, 12.2747, 15.6149, 17.2958, 20.1994]
deviation strategy: [0, 0, 0, 0, 0, 1]

---------- sample index: 246 ----------
========== 246 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

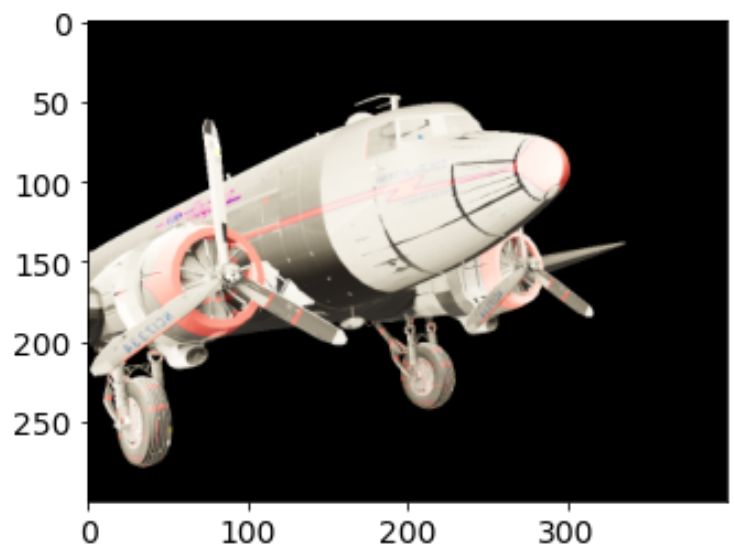


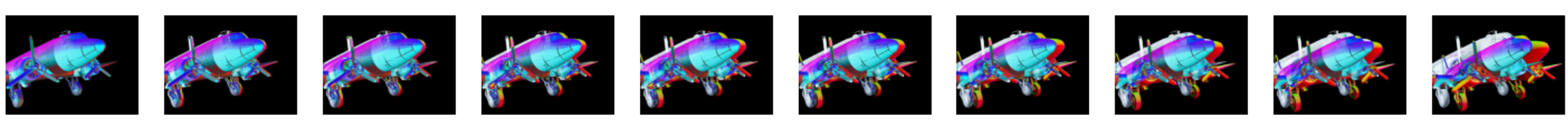
```
distance: [5.3699, 6.1917, 7.8801, 12.0246, 14.3194, 16.6883, 22.2821, 29.0536, 35.1446]
deviation strategy: [0, 0, 0, 0.2, 0.2, 1]

---------------------- sample index: 242 ---------------------------
================= 242 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```



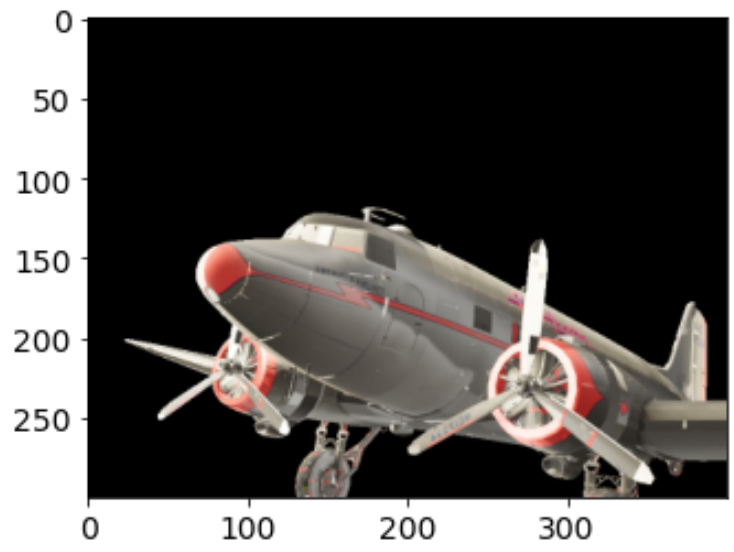


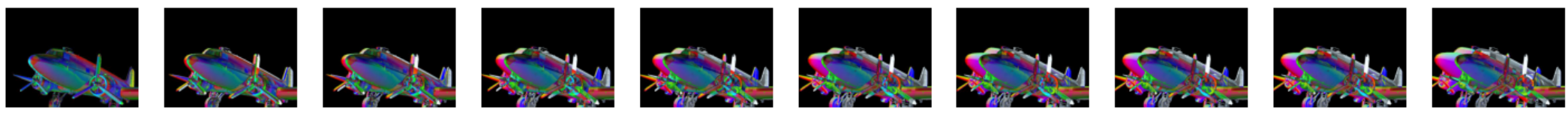
```
distance: [5.3699, 6.1917, 7.8801, 12.0246, 14.3194, 16.6883, 22.2821, 29.0536, 35.1446]
deviation strategy: [0, 0, 0, 0.2, 1, 0.2]

---------------------- sample index: 238 ---------------------------
================= 238 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





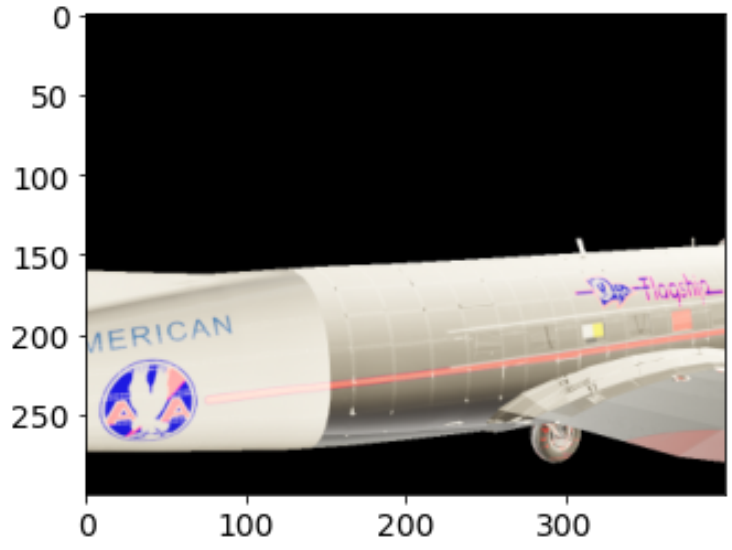
```
distance: [5.3699, 6.1917, 7.8801, 12.0246, 14.3194, 16.6883, 22.2821, 29.0536, 35.1446]
deviation strategy: [0, 0, 0, 1, 0.2, 0.2]

---------------------- sample index: 97 ---------------------------
================= 97 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

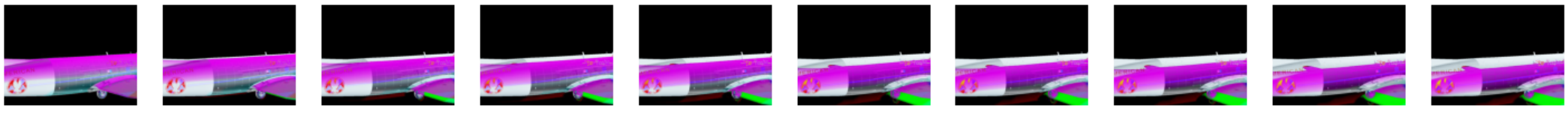




```
distance: [7.9537, 10.0103, 11.504, 15.7238, 16.691, 19.2407, 20.4337, 21.2695, 21.7848]
deviation strategy: [0, 0, 0, 0, 0, -1]

---------------------- sample index: 183 ---------------------------
```

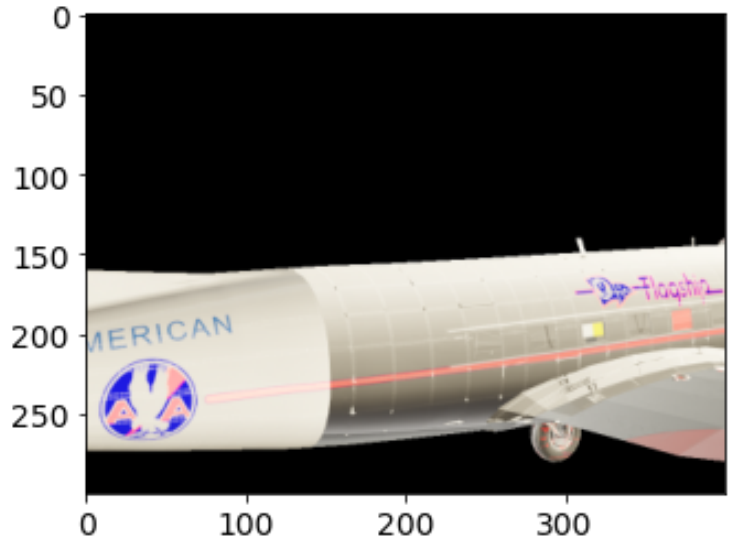
========== 183 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

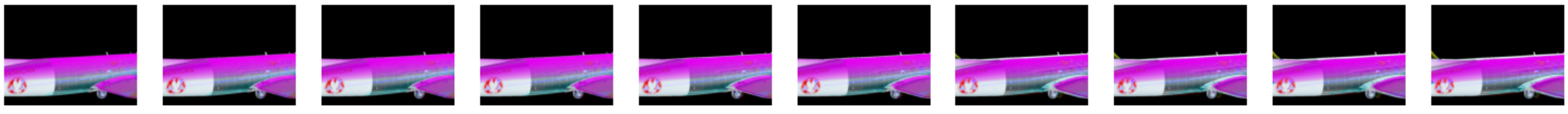




distance: [3.2969, 12.5127, 22.055, 22.2219, 23.2135, 25.8074, 25.7788, 25.784, 25.395]
deviation strategy: [0, 0, 1, 0, 0, 0]

---------- sample index: 179 ----------
========== 179 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

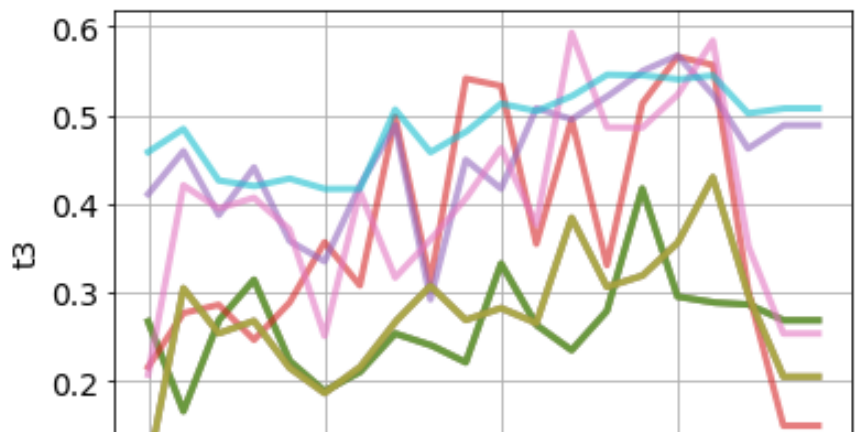


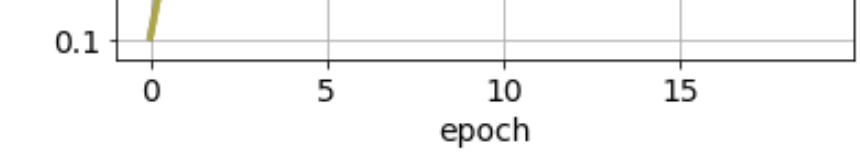


distance: [3.2969, 4.7162, 4.9728, 5.2094, 7.5131, 8.1136, 15.6399, 16.2337, 17.1846]
deviation strategy: [0, 1, 0, 0, 0, 0]

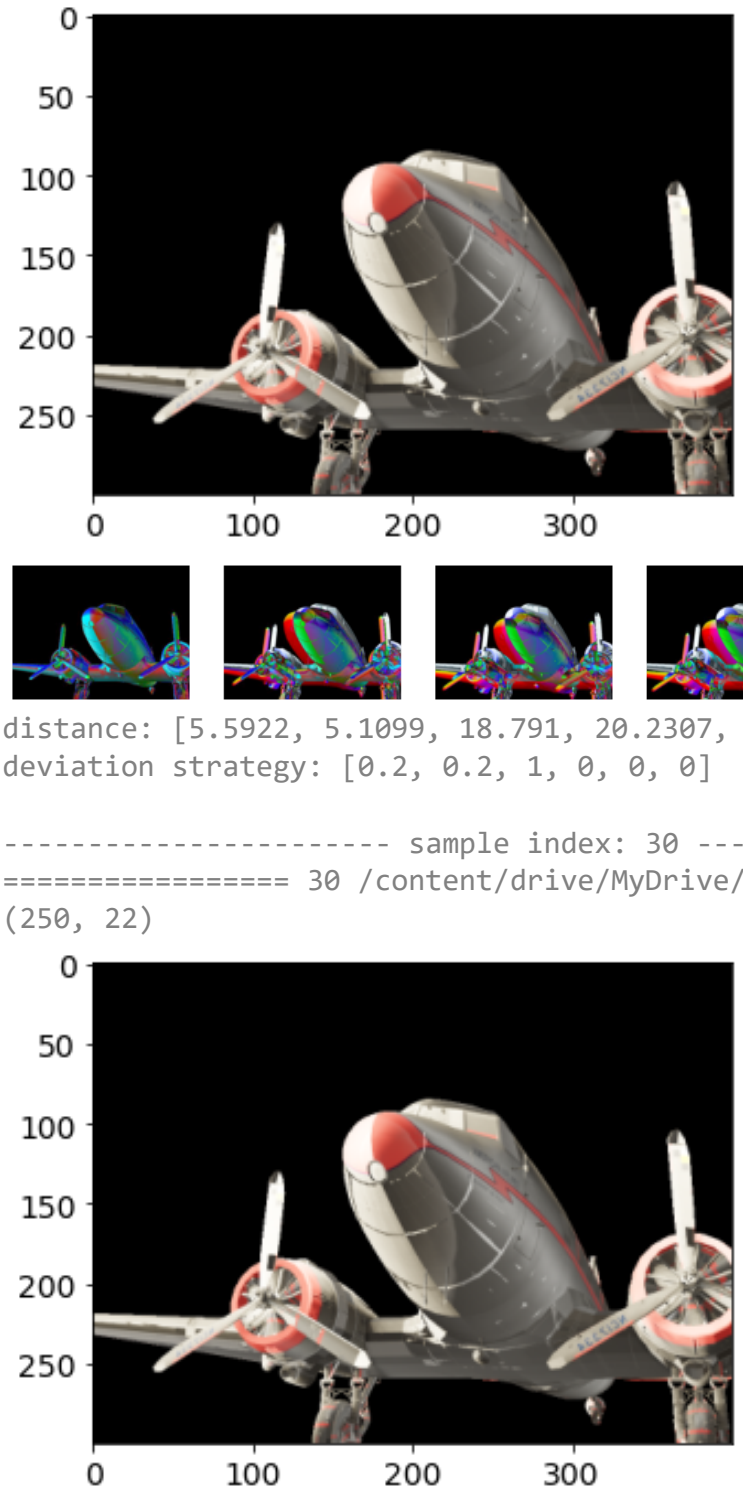
Bottom 10 examples for t3

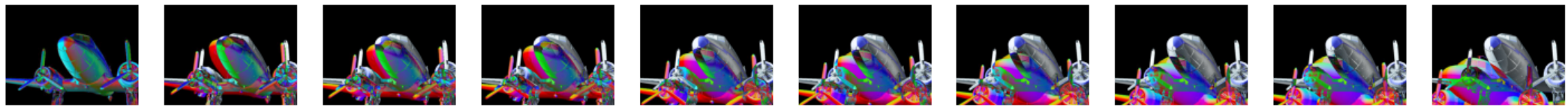
batch_id	index	slope	term	ex_id
8	2	0.006194	t3	34
7	2	0.006194	t3	30
6	2	0.006194	t3	26
54	0	0.006469	t3	216
45	1	0.006556	t3	181
46	2	0.006779	t3	186
44	2	0.006779	t3	178
45	2	0.006779	t3	182
38	2	0.007158	t3	154
30	0	0.010539	t3	120



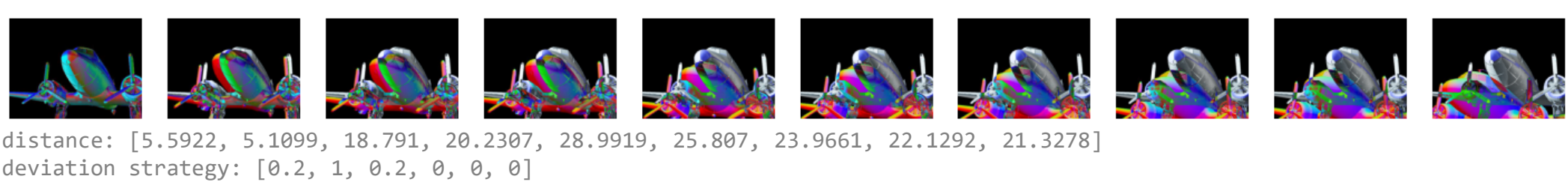



```
==================================
mode: Validation  seq_file:metadata_val_sequence.txt  ex_index_list: [ 34  30  26 216 181 186 178 182 154 120]

----------------------- sample index: 34 ---------------------------
================= 34 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

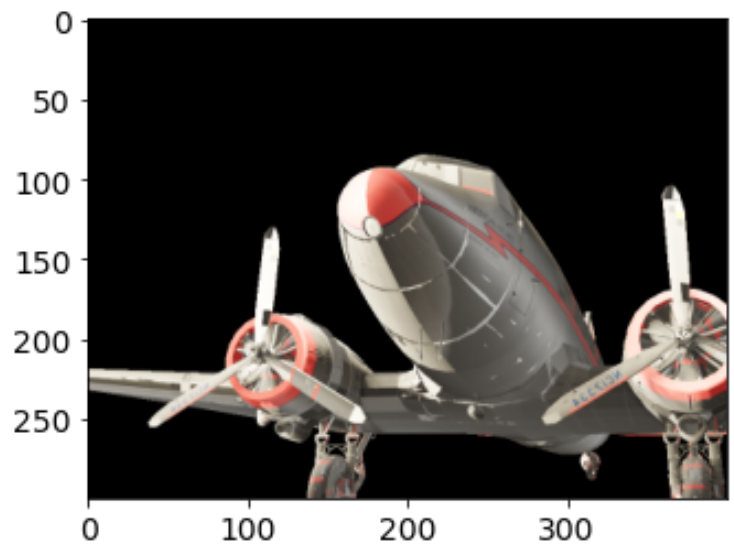


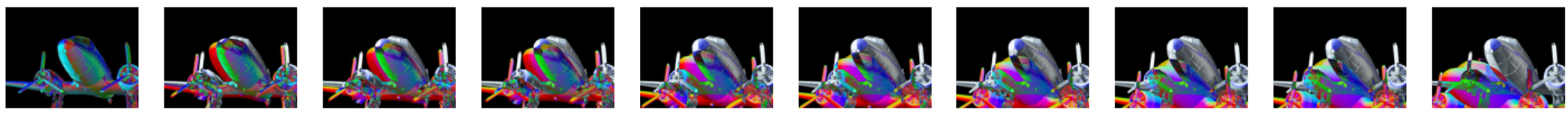


```
distance: [5.5922, 5.1099, 18.791, 20.2307, 28.9919, 25.807, 23.9661, 22.1292, 21.3278]
deviation strategy: [0.2, 0.2, 1, 0, 0, 0]

----------------------- sample index: 30 ---------------------------
================= 30 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```

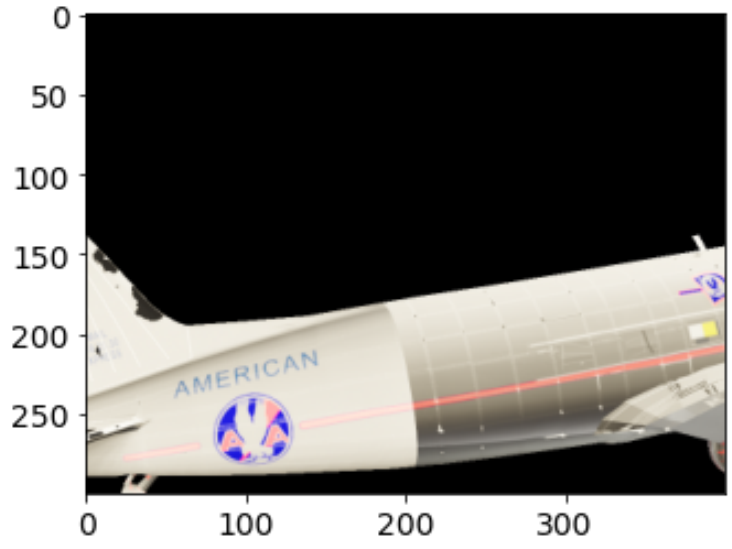
```
distance: [5.5922, 5.1099, 18.791, 20.2307, 28.9919, 25.807, 23.9661, 22.1292, 21.3278]
deviation strategy: [0.2, 1, 0.2, 0, 0, 0]

----------------------- sample index: 26 ---------------------------
================= 26 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)
```





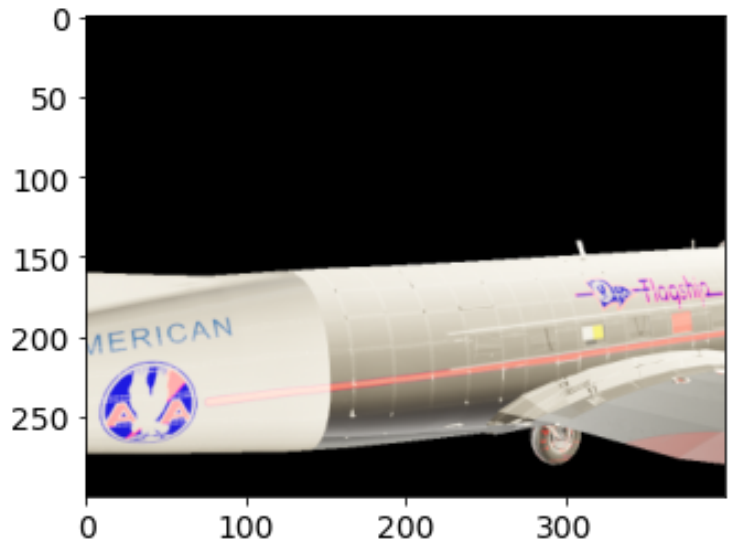
```
distance: [5.5922, 5.1099, 18.791, 20.2307, 28.9919, 25.807, 23.9661, 22.1292, 21.3278]
deviation strategy: [1, 0.2, 0.2, 0, 0, 0]
```

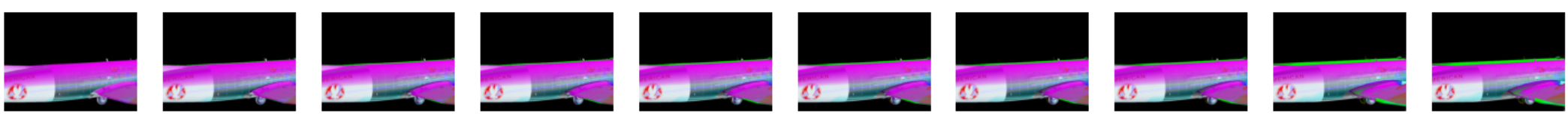
---------- sample index: 216 ----------
========== 216 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



distance: [7.1626, 7.022, 6.8386, 6.0545, 5.2895, 5.0789, 4.7951, 4.6881, 5.2568]
deviation strategy: [0, 0, 0, 0, 1, 0]

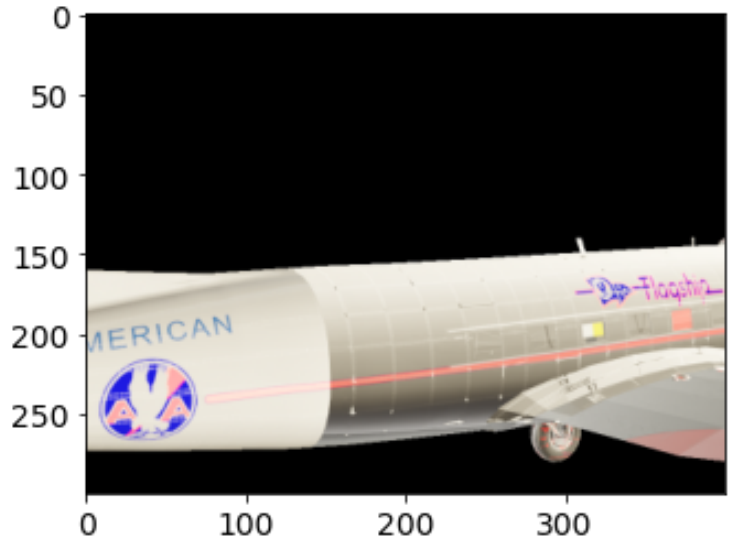
---------- sample index: 181 ----------
========== 181 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

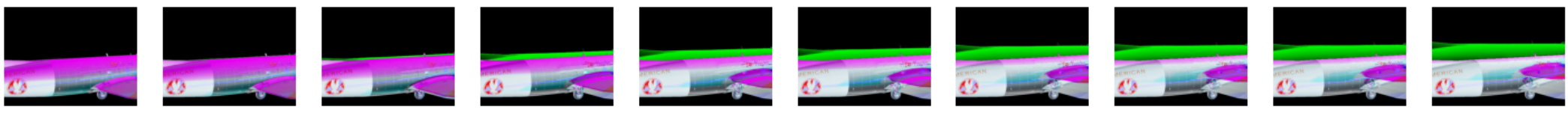




distance: [3.2969, 2.9992, 3.7672, 3.9992, 5.0531, 5.1772, 5.1772, 5.6123, 8.2853]
deviation strategy: [0, -1, 0, 0, 0, 0]

---------- sample index: 186 ----------
========== 186 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



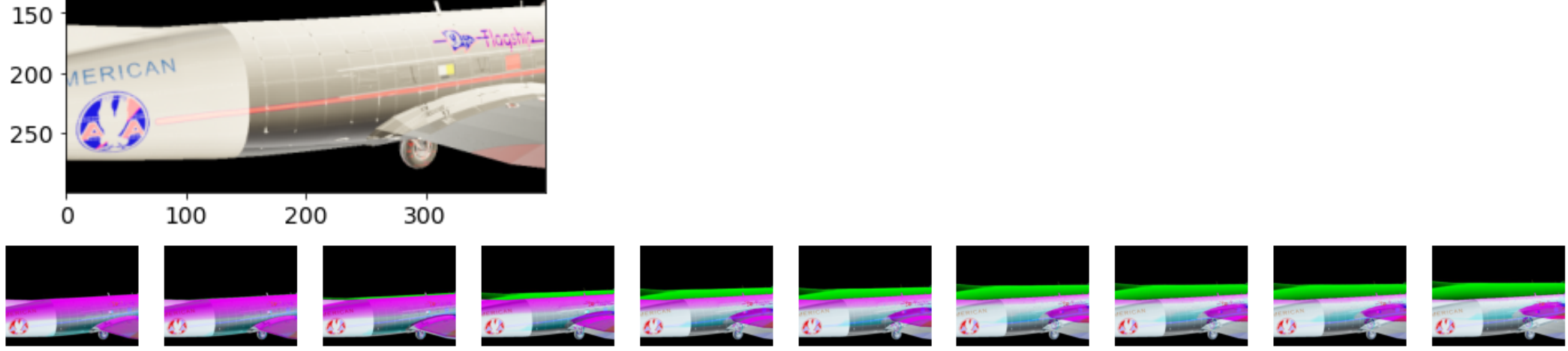


distance: [3.2969, 2.6557, 5.7136, 13.7494, 21.8477, 25.298, 25.2734, 26.2043, 25.1215]
deviation strategy: [-0.2, -0.2, -1, 0, 0, 0]

---------- sample index: 178 ----------
========== 178 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)

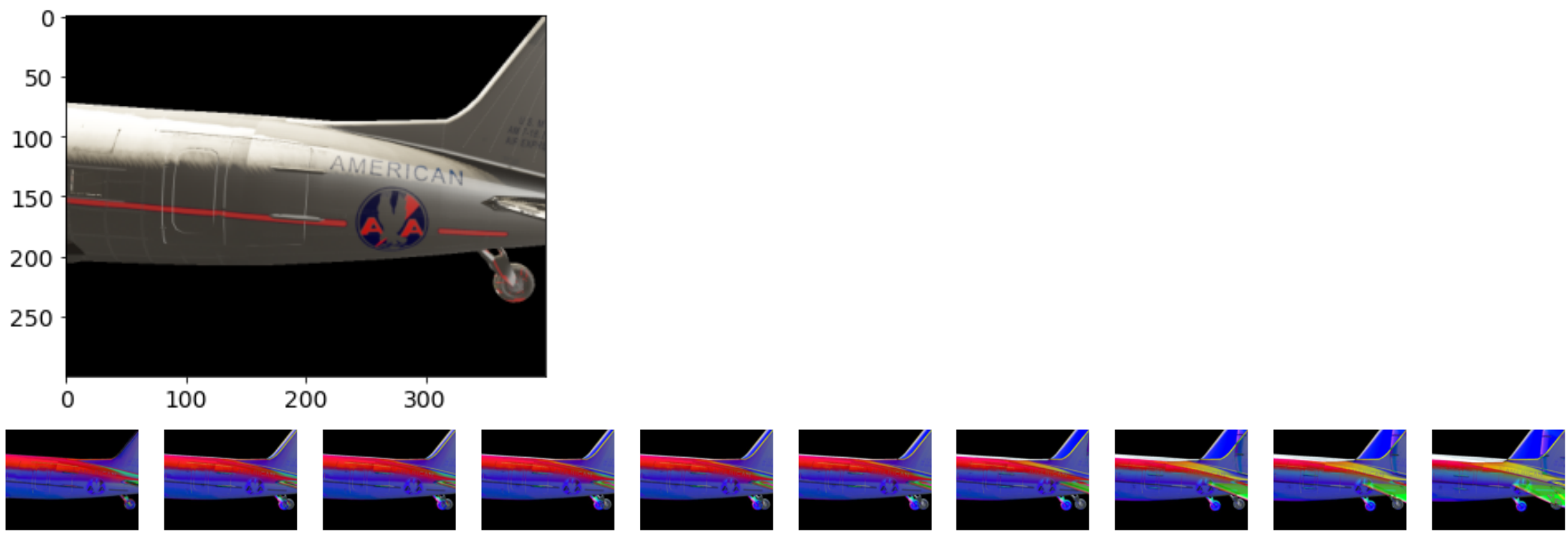
distance: [3.2969, 2.6557, 5.7136, 13.7494, 21.8477, 25.298, 25.2734, 26.2043, 25.1215]
deviation strategy: [-1, -0.2, -0.2, 0, 0, 0]

---------- sample index: 182 ----------
========== 182 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



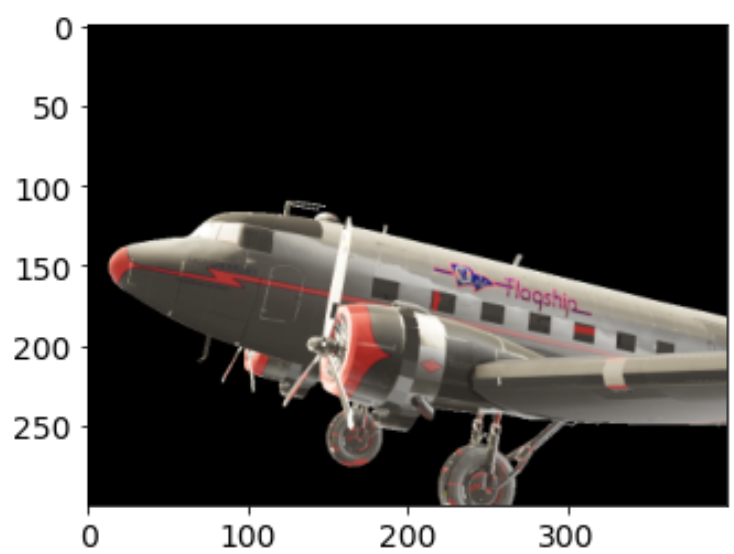
distance: [3.2969, 2.6557, 5.7136, 13.7494, 21.8477, 25.298, 25.2734, 26.2043, 25.1215]
deviation strategy: [-0.2, -1, -0.2, 0, 0, 0]

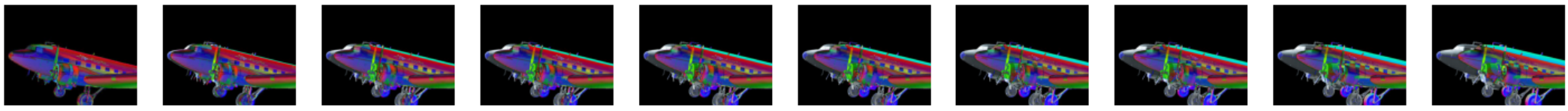
---------- sample index: 154 ----------
========== 154 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)



distance: [2.7835, 1.9453, 1.8179, 1.7626, 1.7184, 1.826, 4.8028, 9.2259, 11.7338]
deviation strategy: [0, 1, 0, 0, 0, 0]

---------- sample index: 120 ----------
========== 120 /content/drive/MyDrive/BigDataHub/Differentiable_Render/dataset/metadata_val_sequence.txt
(250, 22)





```
distance: [6.3851, 4.268, 3.0061, 3.1633, 4.7866, 5.1642, 8.2361, 8.1943, 11.0508]
deviation strategy: [0, 0, 0, 0, 0, 1]
```

HTML image embedded in colab

```
1 import pandas as pd
2 import numpy as np
3 from IPython.display import display, HTML
```

```
1 from datatools_bdh.data_uri import bytes_to_uri
2 def embeding_html_imgs(img_names):
3   # convert image name list to Dataframe HTML
```

```
#  img_name:  list of image names   e.g.  img_names = ["image_a.jpg","image_b.jpg"]
# return value: dataframe, which can display image list as HTML in jupyternotbook
  num = len(img_names)
  img_names = np.array(img_names).reshape(-1,len(img_names))
  df = pd.DataFrame(img_names)


  # could import this from datatools_bdh.all
  def html_img(ipath, widthpx=120):
      return f'<img width="{widthpx}px" src="{ipath}"/>'

  def make_file_uri(fname):
      try:
        return bytes_to_uri(open(fname,'rb').read())
      except:
        print(f"{fname} not found")
        return ";"

  #df = df.applymap(lambda c: f"<b>{c}</b>")
  df = df.applymap(lambda c: html_img(make_file_uri(c),widthpx=100))
  df.columns = ['']*df.shape[1]
  df.index.name = None
  df.index = ['']*df.shape[0]
  display(HTML(df.to_html(escape=False)))
  return df

img_names = ["image_a.jpg","image_b.jpg"]
df = embeding_html_imgs(img_names)
#HTML(df.to_html(escape=False))
```