brainstorming for things to try:

- 1 solid analysis for presentation
- https://github.com/ohsono/stats414/tree/main

(1) feature engineering (Nils / Rohan)

- interest categories
- removing unimportant features & analyzing for model results
- data pruning (don't see the 1/3 clickthrough rate stated in lecture)
    - finding better user removal threshold
    - Removing uninformative features

(2) generate 2 different synthetic datasets (Lucy / Setara)

- ChatGPT (more recent version)
    - prompt engineering (casual vs formatted requests, etc)
- realtabformer and multi table generation (uses chat 2)**
    - https://github.com/worldbank/REaLTabFormer
- CTGAN
    - https://github.com/sdv-dev/CTGAN

post Wednesday

(3) framework for evaluating quality of synthetic data

- statistical distributions comparison
- lasso on real vs. synthetic data
- we need to do fidelity and utilization based off the slides

(4) test synthetic + real vs. real classifiers and compare different metrics - try 2 classifiers ?

- logistic regression, xgboost, stacked etc.

framework based off what guest lecturer said:

- prune based off the task id and user id and this should be 6 subgroups
- afterwards can do preprocessing, then running xgboost, random forest, etc to find unimportant features and remove those (for this I fear we have to do it on the full pruned dataset and each subset for the synthetic data)
- create synthetic data and do this on each of the subgroups for computational purposes
- do evaluation to see which synthesizer is better
- then choose one and run logistic regression/xg boost
- then can compare metrics and to midterm metrics

```
Start coding or generate with AI.
```

(1) Load Data

```
!pip install -q joblib
!pip install -q realtabformer
!pip install -q ctgan
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 kB 4.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 74.3/74.3 kB 7.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 89.4 MB/s eta 0:00:00
```

```
import numpy as np
import kagglehub
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
import xgboost as xgb
import os
import joblib
```

```
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import roc_auc_score, log_loss
from sklearn.model_selection import train_test_split
from realtabformer import REaLTabFormer
from ctgan import CTGAN
from ctgan import load_demo
```

Run once then rely on cached data afterwards!

Dataset link: https://www.kaggle.com/datasets/xiaojiu1414/digix-global-ai-challenge?resource=download

```
# Download latest version
path = kagglehub.dataset_download("xiaojiu1414/digix-global-ai-challenge")

print("Path to dataset files:", path)
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/xiaojiu1414/digix-global-ai-challenge?dataset_versio
100%|██████████| 879M/879M [00:40<00:00, 22.6MB/s]Extracting files...

Path to dataset files: /root/.cache/kagglehub/datasets/xiaojiu1414/digix-global-ai-challenge/versions/1
```

```
# List test files
os.listdir(os.path.join(path, "test"))
```

```
['test_data_ads.csv', 'test_data_feeds.csv']
```

```
# List training files
os.listdir(os.path.join(path, "train"))
```

```
['train_data_ads.csv', 'train_data_feeds.csv']
```

```
train_feeds = pd.read_csv(os.path.join(path, "train", "train_data_feeds.csv"))
train_ads = pd.read_csv(os.path.join(path, "train", "train_data_ads.csv"))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-2701221684.py in <cell line: 0>()
----> 1 train_feeds = pd.read_csv(os.path.join(path, "train", "train_data_feeds.csv"))
      2 train_ads = pd.read_csv(os.path.join(path, "train", "train_data_ads.csv"))

NameError: name 'path' is not defined
```

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

```
Mounted at /content/drive
```

```
joblib.dump(train_feeds, '/content/drive/MyDrive/train_feeds.joblib')
joblib.dump(train_ads, '/content/drive/MyDrive/train_ads.joblib')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-295855377.py in <cell line: 0>()
----> 1 joblib.dump(train_feeds, '/content/drive/MyDrive/train_feeds.joblib')
      2 joblib.dump(train_ads, '/content/drive/MyDrive/train_ads.joblib')

NameError: name 'train_feeds' is not defined
```

Uncaching Data

Using joblib to decrease compile time. It's fast and robust with large data and uses disk-caching to avoid reloading data. Storing a copy in your drive then copying it over to /content/ during session for speed.

Note: Caching the dataset thru joblib means any changes made after this instance will not be reflected in subsequent reboots of session

Resources: https://joblib.readthedocs.io/en/stable/

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

```
Mounted at /content/drive
```

```
!cp /content/drive/MyDrive/train_feeds.joblib /content/
!cp /content/drive/MyDrive/train_ads.joblib /content/
```

```
train_feeds = joblib.load('/content/train_feeds.joblib')
train_ads = joblib.load('/content/train_ads.joblib')
```

(1.1) Data Descriptions

Ads Data Description - A.K.A. "Source Domain"

Variable	Chinese Description	English Translation
label		User ID
user_id		User ID
age		Age
gender		Gender
residence		Permanent residence (province)
city		Permanent residence (city ID).
city_rank		Permanent residence (city level).
series_dev		Device series
series_group		Device series group
emui_dev	emui	EMUI version number
device_name		Phone model used by the user
device_size		Size of the user's phone
net_type		Network status when the behavior occurred
task_id		Unique identifier of the ad task
adv_id	id	Material ID corresponding to the ad task
creat_type_cd	id	Creative type ID of the material
adv_prim_id	id	Advertiser ID corresponding to the ad task
inter_type_cd		Interaction type of the material in the ad task
slot_id	id	Ad placement ID
site_id	id	Media ID
spread_app_id	id	Application ID associated with the ad task
hispace_app_tags		Tags of the application associated with the ad task
app_second_class		Secondary category of the application associated with the ad task
app_score	app	App score
ad_click_list_001	id	List of ad task IDs clicked by the user
ad_click_list_002	id	List of advertiser IDs for ads clicked by the user
ad_click_list_003		List of recommended apps from ads clicked by the user
ad_close_list_001		List of ad task IDs closed by the user
ad_close_list_002		List of advertiser IDs for ads closed by the user
ad_close_list_003		List of recommended apps from ads closed by the user
pt_d		
u_newsCatInterestsST		User's short-term interest category preferences Timestamp
u_feedLifeCycle		User engagement on news feeds
u_refreshTimes		Average number of valid news feed updates per day
log_id	id	Sample ID

Feeds Data Description - A.K.A "Target Domain"

Variable	Chinese Description	English Translation
u_userId		User ID
u_phonePrice		Price of a user's device
u_browserLifeCycle		User engagement on browser
u_browserMode		Browser service type

Variable	Chinese Description		English Translation
u_feedLifeCycle			User engagement on news feeds
u_refreshTimes			Average number of valid news feed updates per day
u_newsCatInterests			Liked news feed categories based on the user's click behavior
u_newsCatDislike			Negative feedback category preferences in news feed content
u_newsCatInterestsST			User's short-term interest category preferences
u_click_ca2_news			Click sequence of article categories by the user
i_docId	docid		Article doc ID
i_s_sourceId	sourceid		Source ID of the article
i_regionEntity	id		Regional entity ID of the article
i_cat	id		Article category ID
i_entities	id		Entity word IDs in the article
i_dislikeTimes			Number of negative feedbacks on the article
i_upTimes			Number of likes on the article
I_dtype			Display type of the article
e_ch			Channel
e_m			Device model where the event originated
e_po			Position (ranking)
e_pl			Location visited
e_rn			Feed refresh count (Nth refresh)
e_section			Type of news feed scene
e_et			Timestamp
label	-1	1	Whether the user clicked (–1: No, 1: Yes)
cilLabel	-1	1	Whether the user liked (–1: No, 1: Yes)
pro			Article reading progress

(2) Data Pruning

```
train_ads.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7675517 entries, 0 to 7675516
Data columns (total 35 columns):
 #   Column                Dtype
---  ------                -----
 0   log_id                int64
 1   label                 int64
 2   user_id               int64
 3   age                   int64
 4   gender                int64
 5   residence             int64
 6   city                  int64
 7   city_rank             int64
 8   series_dev            int64
 9   series_group          int64
 10  emui_dev              int64
 11  device_name           int64
 12  device_size           int64
 13  net_type              int64
 14  task_id               int64
 15  adv_id                int64
 16  creat_type_cd         int64
 17  adv_prim_id           int64
 18  inter_type_cd         int64
 19  slot_id               int64
 20  site_id               int64
 21  spread_app_id         int64
 22  hispace_app_tags      int64
 23  app_second_class      int64
 24  app_score             float64
 25  ad_click_list_v001    object
 26  ad_click_list_v002    object
 27  ad_click_list_v003    object
 28  ad_close_list_v001    object
 29  ad_close_list_v002    object
 30  ad_close_list_v003    object
 31  pt_d                  datetime64[ns]
 32  u_newsCatInterestsST  object
 33  u_refreshTimes        int64
 34  u_feedLifeCycle       int64
dtypes: datetime64[ns](1), float64(1), int64(26), object(7)
memory usage: 2.0+ GB
```

```
# initially the distribution is
train_ads['label'].value_counts() / train_ads.shape[0]
```

label	count
0	0.984478
1	0.015522

dtype: float64

```
# Convert pt_d to datetime for completeness
train_ads["pt_d"] = pd.to_datetime(train_ads["pt_d"], format="%Y%m%d%H%M")
train_ads = train_ads.sort_values("pt_d").reset_index(drop=True)

# STEP 2: Find the top 6 task_id subgroups (D1–D6)
print("STEP 2: Building D1–D6 (top 6 task_ids, then user-level pruning)")

task_id_counts = train_ads["task_id"].value_counts()
top_6_task_ids = task_id_counts.head(6).index.tolist()

print("\nTop 6 task_ids by frequency:")
for i, t in enumerate(top_6_task_ids, 1):
    print(f"  D{i}: task_id={t}  ({task_id_counts[t]:,} rows)")

D_sets = {}          # pruned D1–D6 stored here
D_stats = []         # for a little summary table

for i, t_id in enumerate(top_6_task_ids, 1):
    print(f"\n==================== D{i}: task_id = {t_id} ====================")

    # ---- 2a. subset for this task_id (original subgroup) ----
    df_t = train_ads[train_ads["task_id"] == t_id].copy()
    ctr_before = df_t["label"].mean()

    print(f"Rows before pruning: {len(df_t):,}")
    print(f"CTR before pruning:  {ctr_before:.4f} ({100*ctr_before:.2f}%)")

    # ---- 2b. keep only users who have EVER clicked in THIS subgroup ----
    users_clicked = (
        df_t.groupby("user_id")["label"].any()
        .pipe(lambda s: s[s])           # keep True
        .index
    )

    df_pruned = df_t[df_t["user_id"].isin(users_clicked)].copy()
    ctr_after = df_pruned["label"].mean()

    print(f"Rows after pruning:  {len(df_pruned):,}")
    print(f"CTR after pruning:   {ctr_after:.4f} ({100*ctr_after:.2f}%)")

    # store
    D_sets[f"D{i}"] = df_pruned
    D_stats.append({
        "Dataset": f"D{i}",
        "task_id": t_id,
        "rows_before": len(df_t),
        "rows_after": len(df_pruned),
        "ctr_before": ctr_before,
        "ctr_after": ctr_after,
    })

# Optional: see a small summary like the slide's table
D_stats_df = pd.DataFrame(D_stats)
print("\nSummary of D1–D6:")
print(D_stats_df[["Dataset","task_id","rows_before","rows_after",
                  "ctr_before","ctr_after"]])

# STEP 3: Combine D1–D6 if you want ONE pruned dataset for modeling
print("STEP 3: Combine pruned D1–D6 into one dataset (for modeling)")

ads_pruned_all = pd.concat(D_sets.values(), ignore_index=True)

combined_ctr = ads_pruned_all["label"].mean()
```

```
print(f"Rows in combined pruned dataset: {len(ads_pruned_all):,}")
print(f"Users in combined pruned dataset: {ads_pruned_all['user_id'].nunique():,}")
print(f"CTR in combined pruned dataset:  {combined_ctr:.4f} ({100*combined_ctr:.2f}%)")

# From here on, use ads_pruned_all for feature engineering:
pruned_train_ads = ads_pruned_all.copy()
```

```
STEP 2: Building D1–D6 (top 6 task_ids, then user-level pruning)

Top 6 task_ids by frequency:
  D1: task_id=22100  (154,812 rows)
  D2: task_id=14584  (126,367 rows)
  D3: task_id=34382  (122,094 rows)
  D4: task_id=34975  (92,381 rows)
  D5: task_id=31941  (77,470 rows)
  D6: task_id=31996  (69,691 rows)

==================== D1: task_id = 22100 ====================
Rows before pruning: 154,812
CTR before pruning:  0.0018 (0.18%)
Rows after pruning:  4,628
CTR after pruning:   0.0607 (6.07%)

==================== D2: task_id = 14584 ====================
Rows before pruning: 126,367
CTR before pruning:  0.0148 (1.48%)
Rows after pruning:  29,359
CTR after pruning:   0.0637 (6.37%)

==================== D3: task_id = 34382 ====================
Rows before pruning: 122,094
CTR before pruning:  0.0016 (0.16%)
Rows after pruning:  2,724
CTR after pruning:   0.0712 (7.12%)

==================== D4: task_id = 34975 ====================
Rows before pruning: 92,381
CTR before pruning:  0.0019 (0.19%)
Rows after pruning:  3,880
CTR after pruning:   0.0459 (4.59%)

==================== D5: task_id = 31941 ====================
Rows before pruning: 77,470
CTR before pruning:  0.0322 (3.22%)
Rows after pruning:  12,677
CTR after pruning:   0.1966 (19.66%)

==================== D6: task_id = 31996 ====================
Rows before pruning: 69,691
CTR before pruning:  0.0025 (0.25%)
Rows after pruning:  1,555
CTR after pruning:   0.1132 (11.32%)

Summary of D1–D6:
  Dataset  task_id  rows_before  rows_after  ctr_before  ctr_after
0      D1    22100       154812        4628    0.001815   0.060717
1      D2    14584       126367       29359    0.014790   0.063660
2      D3    34382       122094        2724    0.001589   0.071219
3      D4    34975        92381        3880    0.001927   0.045876
4      D5    31941        77470       12677    0.032167   0.196576
5      D6    31996        69691        1555    0.002525   0.113183
STEP 3: Combine pruned D1–D6 into one dataset (for modeling)
Rows in combined pruned dataset: 54,823
Users in combined pruned dataset: 4,114
CTR in combined pruned dataset:  0.0947 (9.47%)
```

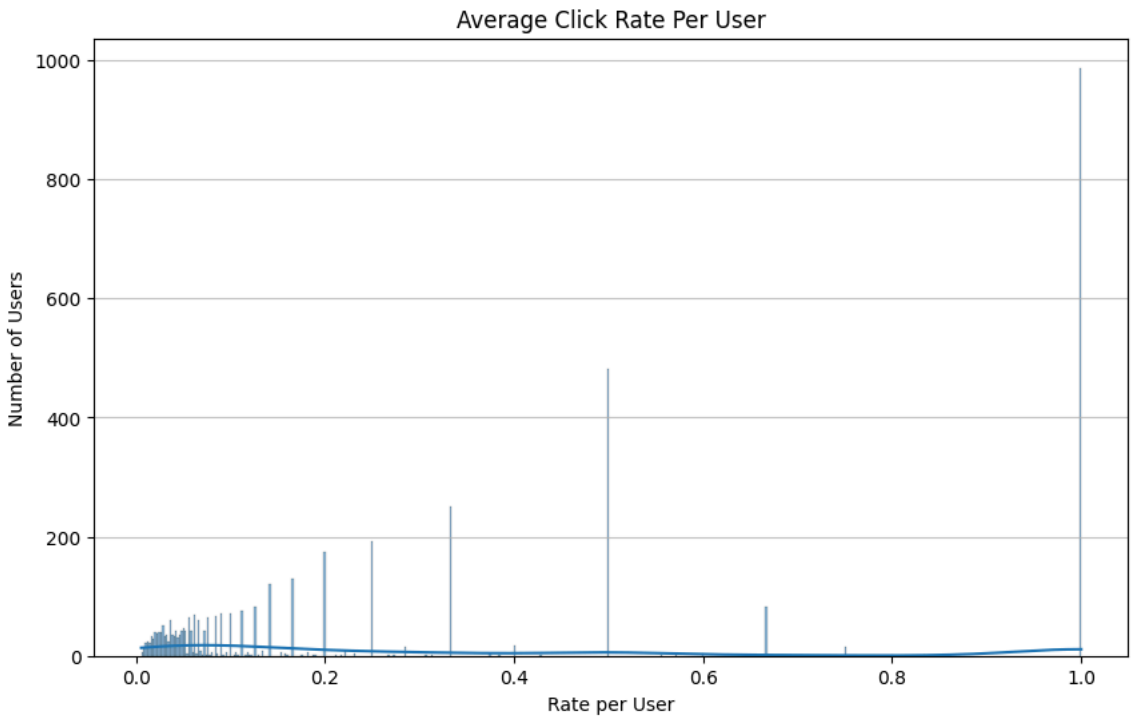
```
# Calculate number of online instances per user
instances_per_user = pruned_train_ads.groupby(['user_id']).size()

# Calculate the sum of 'label' (clicks) for each user
clicks_per_user = pruned_train_ads.groupby(['user_id'])['label'].sum()

# Average click rate for each user
average_click_rate = clicks_per_user / instances_per_user
```

```
results = []
thresholds = np.linspace(0.01, 0.1, num = 10)

for thresh in thresholds:
  higher_click_rate_users = average_click_rate[average_click_rate > thresh] # Corrected filter
  higher_click_rate_users_indices = higher_click_rate_users.index
```

```
  higher_pruned_train_ads = pruned_train_ads[pruned_train_ads['user_id'].isin(higher_click_rate_users_indices)]

  label_counts = higher_pruned_train_ads['label'].value_counts(normalize=True)

  no_click_proportion = label_counts.get(0, 0) # Get proportion of 0, default to 0 if not present
  click_proportion = label_counts.get(1, 0)   # Get proportion of 1, default to 0 if not present
  num_users_left = len(higher_click_rate_users_indices)

  results.append({
      'threshold': thresh,
      'no-click (0)': no_click_proportion,
      'click (1)': click_proportion,
      'num users left': num_users_left
  })

threshold_df = pd.DataFrame(results)
display(threshold_df)
```

	threshold	no-click (0)	click (1)	num users left
0	0.01	0.900960	0.099040	4093
1	0.02	0.879922	0.120078	3950
2	0.03	0.855660	0.144340	3757
3	0.04	0.825749	0.174251	3539
4	0.05	0.794105	0.205895	3342
5	0.06	0.766017	0.233983	3190
6	0.07	0.734753	0.265247	3044
7	0.08	0.705078	0.294922	2926
8	0.09	0.686050	0.313950	2855
9	0.10	0.647123	0.352877	2705

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting the distribution of clicks per user
plt.figure(figsize=(10, 6))
sns.histplot(average_click_rate, bins=500, kde=True)
plt.title('Average Click Rate Per User')
plt.xlabel('Rate per User')
plt.ylabel('Number of Users')
plt.grid(axis='y', alpha=0.75)
plt.show()
```



Questions on Additional Pruning Methodology

- How representative are instance users (1-3/5/10) of ad clickthrough rate predciton?
- Should we exclude users past a certain individual click rate (.03/.05/.1) as only a click over many instances may be indicative of an accidental click? .03 Threshold elminiates ~50% of users.
- Should we exclude the roughly 80 users whose only appearance one the web page led to a ad click?

```
pruned_train_ads.head()
```

	log_id	label	user_id	age	gender	residence	city	city_rank	series_dev	series_group	...	ad_click_list
0	1131711	0	247105	3	2	11	169	2	16	5	...	12092^21493^26644^14606
1	1131737	0	247105	3	2	11	169	2	16	5	...	12092^21493^26644^14606
2	1131714	0	247105	3	2	11	169	2	16	5	...	12092^21493^26644^14606
3	1140732	0	208905	8	2	42	410	2	21	4	...	28403^20581^18365^13230
4	1140771	0	208905	8	2	42	410	2	21	4	...	28403^20581^18365^13230

5 rows × 35 columns

```
pruned_train_ads.shape
```

```
(54823, 35)
```

```
pruned_train_ads['label'].value_counts() / pruned_train_ads.shape[0]
```

label	count
0	0.905332
1	0.094668

dtype: float64

(3) Data Processing

type casting to decrease memory requirements

```
# source: https://medium.com/pythoneers/optimising-data-storage-in-python-through-efficient-datatype-casting-3df6c4b
# modifies dataframe in place
def reduce_mem_usage(df, verbose=True):
    numerics = ['int16', 'int32', 'int64', 'float16', 'float64']
    start_mem = df.memory_usage().sum() / 1024**2
    for col in df.columns:
        col_type = df[col].dtypes
        if col_type in numerics:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                # float32: default in pytorch
                if c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)

    end_mem = df.memory_usage().sum() / 1024**2

    print('Memory usage before optimization is: {:.2f} MB'.format(start_mem))
    print('Memory usage after optimization is: {:.2f} MB'.format(end_mem))
    print('Decreased by {:.1f}%'.format(100 * (start_mem - end_mem) / start_mem))

    return df
```

steps:

- drop irrelevant variables
- parse timestamp pt_d into new features (did in pruning)
- create new features: length of cat news interests and ads clicked/closed, remove the concatenated features
- turn features into categorical i.e. gender, age

functions for data processing

**Is add_list_count_features inducing a lot of information loss by removing ad content types, etc. that could be indicative of higher likelihood to click ads? E.g. People who enjoy online shopping --> more likely to click ads?

Right now, all that the list count features are measuring is whether more or less interests lead to greater/fewer clicks/closes**

```
# purpose: take the lists with ^, remove those features, and instead create new features for the lengths of those li
def add_list_count_features(df, list_features):
    """
    Converts caret-separated list columns into count features,
    aggregates click/close means, drops original list columns,
    and returns the updated dataframe.
    """
```

```
    # ---- Helper function ----
    def count_caret_entries(series):
        return series.fillna("").apply(
            lambda x: len([t for t in str(x).split("^") if t])
        )

    # ---- Step 1: Create *_count columns ----
    for col in list_features:
        df[f"{col}_count"] = count_caret_entries(df[col])

    # ---- Step 2: Drop original list-like columns ----
    df = df.drop(columns=list_features)

    # ---- Step 3: Aggregate click/close features ----
    click_cols = [
        "ad_click_list_v001_count",
        "ad_click_list_v002_count",
        "ad_click_list_v003_count"
    ]
    close_cols = [
        "ad_close_list_v001_count",
        "ad_close_list_v002_count",
        "ad_close_list_v003_count"
    ]

    df["click_count_mean"] = df[click_cols].mean(axis=1)
    df["close_count_mean"] = df[close_cols].mean(axis=1)

    # Drop individual count columns (keep only means)
    df = df.drop(columns=click_cols + close_cols)

    return df
```

doing the processing

```
pruned_train_ads.shape
```

```
(54823, 35)
```

```
# Target column
target = 'label'

# Categorical features (integer codes)
categorical_features = [
    'gender','age','residence','city','city_rank','series_dev','series_group',
    'emui_dev','device_name','device_size','net_type',
    'creat_type_cd','adv_prim_id','inter_type_cd','slot_id',
    'spread_app_id','hispace_app_tags','app_second_class','u_feedLifeCycle'
]

# List-like string features: we convert to simple counts
list_features = [
    'ad_click_list_v001','ad_click_list_v002','ad_click_list_v003',
    'ad_close_list_v001','ad_close_list_v002','ad_close_list_v003',
    'u_newsCatInterestsST'
]

#calling function
train_ads_new = add_list_count_features(pruned_train_ads, list_features)

#removing irrelevant variables, aka ids and pt_d which is now irrelevant
drop_cols = ['log_id','adv_id','task_id','site_id', 'user_id', 'pt_d']
train_ads_new = train_ads_new.drop(columns=drop_cols)
```

```
# Store a list of continuous numeric features
numeric_features = ['app_score','u_refreshTimes','click_count_mean','close_count_mean','u_newsCatInterestsST_count']
```

```
# standardizing #going to move this later since we only need it for logistic regression
#scaler = StandardScaler()
#train_ads_new[num_cols] = scaler.fit_transform(train_ads_new[num_cols])
```

```
# take the numerics and check if we can use a smaller data type
train_ads_new.info()
```

Show hidden output

```
reduce_mem_usage(train_ads_new)
```

```
Memory usage before optimization is: 10.46 MB
Memory usage after optimization is: 2.04 MB
Decreased by 80.5%
```

	label	age	gender	residence	city	city_rank	series_dev	series_group	emui_dev	device_name	...	slot_id	spr
0	0	3	2	11	169	2	16	5	28	240	...	16	
1	0	3	2	11	169	2	16	5	28	240	...	16	
2	0	3	2	11	169	2	16	5	28	240	...	16	
3	0	8	2	42	410	2	21	4	12	310	...	16	
4	0	8	2	42	410	2	21	4	12	310	...	16	
...	...	...	...	...	...	...	...	...	...	...	...	...	
54818	0	8	2	17	343	5	16	5	21	127	...	16	
54819	1	5	2	32	179	5	30	3	13	194	...	23	
54820	0	6	4	33	319	3	27	2	11	140	...	16	
54821	0	6	4	33	319	3	27	2	11	140	...	16	
54822	0	6	4	33	319	3	27	2	11	140	...	16	

54823 rows × 25 columns

(3.1) Recaching for Trimmed/Pruned Dataset

Again, Run Once Then Retrieve From Cache

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

Show hidden output

```
joblib.dump(train_ads_new, '/content/drive/MyDrive/train_ads_new.joblib')
```

Show hidden output

Uncaching Trimmed/Pruned Dataset

```
!cp /content/drive/MyDrive/train_ads_new.joblib /content/
```

```
train_ads_new = joblib.load('/content/train_ads_new.joblib')
```

```
train_ads_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54823 entries, 0 to 54822
Data columns (total 25 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   label                     54823 non-null  int64
 1   age                       54823 non-null  int64
 2   gender                    54823 non-null  int64
 3   residence                 54823 non-null  int64
 4   city                      54823 non-null  int64
 5   city_rank                 54823 non-null  int64
 6   series_dev                54823 non-null  int64
 7   series_group              54823 non-null  int64
 8   emui_dev                  54823 non-null  int64
 9   device_name               54823 non-null  int64
 10  device_size               54823 non-null  int64
 11  net_type                  54823 non-null  int64
 12  creat_type_cd             54823 non-null  int64
```

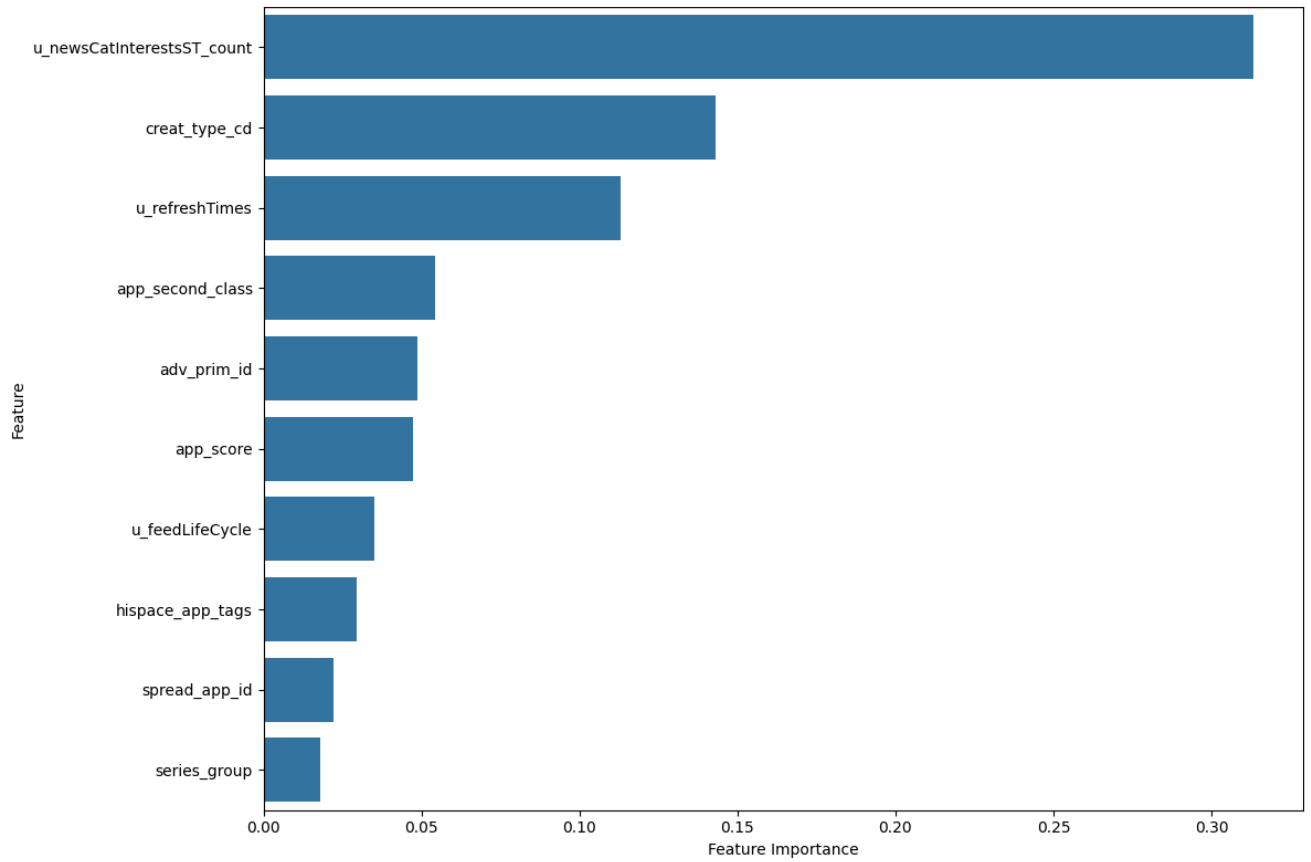
```
 13  adv_prim_id                54823 non-null  int64
 14  inter_type_cd              54823 non-null  int64
 15  slot_id                    54823 non-null  int64
 16  spread_app_id              54823 non-null  int64
 17  hispace_app_tags           54823 non-null  int64
 18  app_second_class           54823 non-null  int64
 19  app_score                  54823 non-null  float64
 20  u_refreshTimes             54823 non-null  int64
 21  u_feedLifeCycle            54823 non-null  int64
 22  u_newsCatInterestsST_count  54823 non-null  int64
 23  click_count_mean           54823 non-null  float64
 24  close_count_mean           54823 non-null  float64
dtypes: float64(3), int64(22)
memory usage: 10.5 MB
```

(3.2) Uninformative Features Testing

XGBClassifier

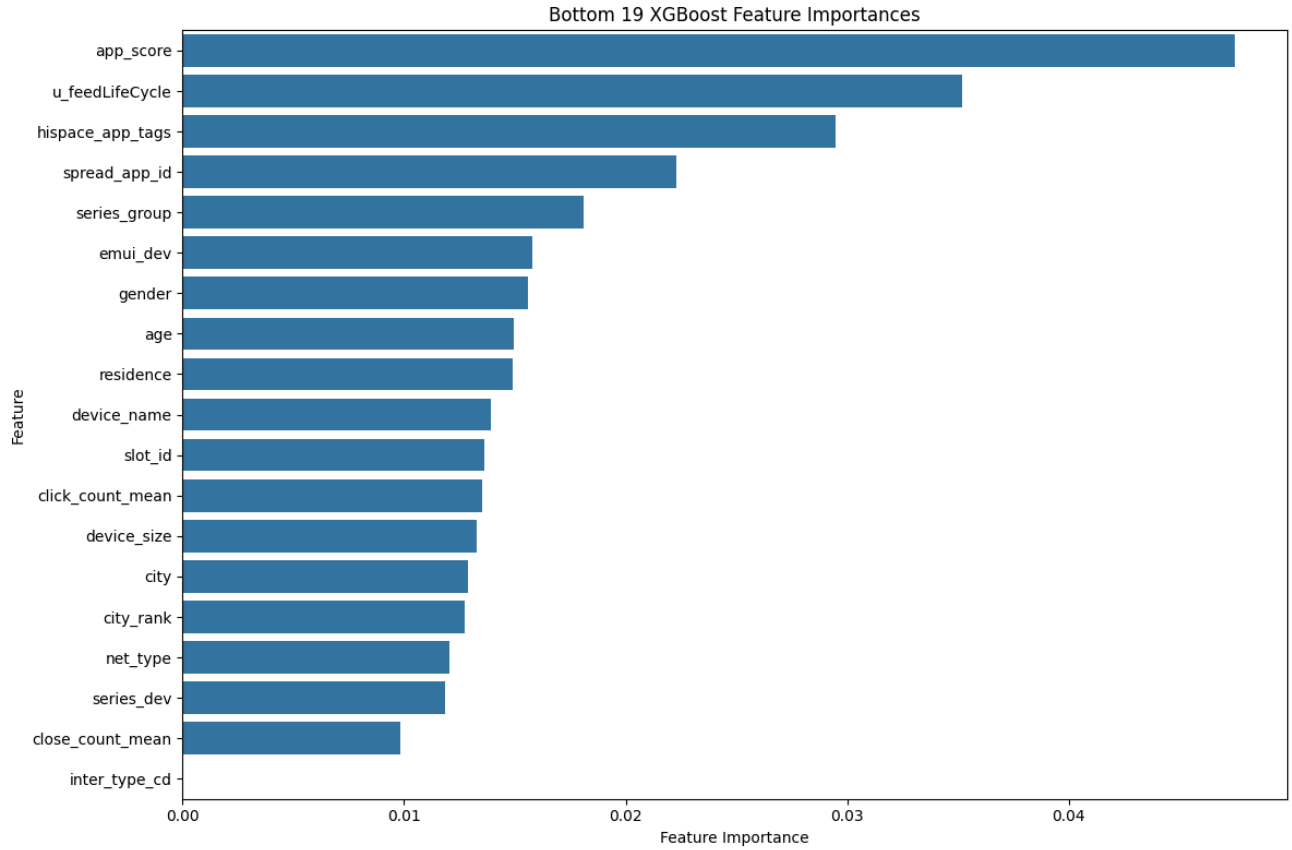
```
# For feature importance, no need for train/test split
X = train_ads_new[numeric_features + categorical_features]
y = train_ads_new[target]

# Training XGB Classifier
model = xgb.XGBClassifier(objective='binary:logistic',
                          eval_metric='logloss',
                          enable_categorical=True,
                          random_state=42)

model.fit(X, y)

# Sorting Feature Importance
feature_importances = pd.DataFrame({
    'feature': X.columns,
    'importance': model.feature_importances_
}).sort_values(by='importance', ascending=False)
```

```
plt.figure(figsize=(12, 8))
sns.barplot(x='importance', y='feature', data=feature_importances.head(10)) # Display top 20 featuresplt.title('Top
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```




```
# Plotting Feature Impotance

# Get bottom 19 least important features
bottom_features = feature_importances.tail(19)

# Print them clearly
print("\n Bottom 19 least important features:")
print(bottom_features.to_string(index=False))

# Plot them
plt.figure(figsize=(12, 8))
sns.barplot(x='importance', y='feature', data=bottom_features)
plt.title('Bottom 19 XGBoost Feature Importances')
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```

```
Bottom 19 least important features:
          feature  importance
        app_score    0.047495
    u_feedLifeCycle    0.035208
   hispace_app_tags    0.029457
      spread_app_id    0.022308
       series_group    0.018110
           emui_dev    0.015797
             gender    0.015607
                age    0.014960
          residence    0.014898
        device_name    0.013921
            slot_id    0.013603
   click_count_mean    0.013531
        device_size    0.013283
               city    0.012883
          city_rank    0.012736
           net_type    0.012056
         series_dev    0.011870
   close_count_mean    0.009809
      inter_type_cd    0.000000
```



remove anything that is 0 so inter_type_cd

also based on guest lecturer's slides we want to remove: adv_prim_id, spread_app_id, hispace_app_tags, app_second_class, app_score

(3.3) Removing Unimportant Feature

```
train_ads_new = train_ads_new.drop(columns=["inter_type_cd", "adv_prim_id", "spread_app_id", "hispace_app_tags", "ap
```

```
train_ads_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54823 entries, 0 to 54822
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   label                       54823 non-null  int64
 1   age                         54823 non-null  int64
 2   gender                      54823 non-null  int64
 3   residence                   54823 non-null  int64
 4   city                        54823 non-null  int64
 5   city_rank                   54823 non-null  int64
 6   series_dev                  54823 non-null  int64
 7   series_group                54823 non-null  int64
 8   emui_dev                    54823 non-null  int64
 9   device_name                 54823 non-null  int64
 10  device_size                 54823 non-null  int64
 11  net_type                    54823 non-null  int64
 12  creat_type_cd               54823 non-null  int64
 13  slot_id                     54823 non-null  int64
 14  u_refreshTimes              54823 non-null  int64
 15  u_feedLifeCycle             54823 non-null  int64
 16  u_newsCatInterestsST_count  54823 non-null  int64
 17  click_count_mean            54823 non-null  float64
 18  close_count_mean            54823 non-null  float64
dtypes: float64(2), int64(17)
memory usage: 7.9 MB
```

(3.4) Applying this preprocessing to all of the subsets for synthetic data generation

```
def preprocess_ads_subset(df):
    """
    Apply the same preprocessing you used for the combined dataset
    to a single D_i subset.
    """
    df_proc = df.copy()

    # convert list-like features to counts + add click/close means
    df_proc = add_list_count_features(df_proc, list_features)

    # drop IDs and datetime (after you've already used pt_d)
    df_proc = df_proc.drop(columns=drop_cols, errors='ignore')

    # if you want to drop inter_type_cd as well:
    df_proc = df_proc.drop(columns=["inter_type_cd", "adv_prim_id", "spread_app_id", "hispace_app_tags", "app_second

    # optional: memory optimization
    df_proc = reduce_mem_usage(df_proc, verbose=False)

    return df_proc
```

```
# D_sets["D1"], ..., D_sets["D6"] exist from pruning loop

D_processed = {}

for name, df_sub in D_sets.items():
    print(f"Preprocessing {name} ...")
    D_processed[name] = preprocess_ads_subset(df_sub)
    print(f"  {name} shape after preprocessing: {D_processed[name].shape}")
```

```
Preprocessing D1 ...
Memory usage before optimization is: 0.71 MB
Memory usage after optimization is: 0.16 MB
Decreased by 77.5%
  D1 shape after preprocessing: (4628, 19)
Preprocessing D2 ...
Memory usage before optimization is: 4.48 MB
Memory usage after optimization is: 1.01 MB
Decreased by 77.5%
  D2 shape after preprocessing: (29359, 19)
Preprocessing D3 ...
Memory usage before optimization is: 0.42 MB
Memory usage after optimization is: 0.09 MB
Decreased by 77.5%
  D3 shape after preprocessing: (2724, 19)
Preprocessing D4 ...
```

```
Memory usage before optimization is: 0.59 MB
Memory usage after optimization is: 0.13 MB
Decreased by 77.5%
  D4 shape after preprocessing: (3880, 19)
Preprocessing D5 ...
Memory usage before optimization is: 1.93 MB
Memory usage after optimization is: 0.44 MB
Decreased by 77.5%
  D5 shape after preprocessing: (12677, 19)
Preprocessing D6 ...
Memory usage before optimization is: 0.24 MB
Memory usage after optimization is: 0.05 MB
Decreased by 77.5%
  D6 shape after preprocessing: (1555, 19)
```

```
D_processed["D4"].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3880 entries, 1853 to 7664400
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   label                       3880 non-null   int8
 1   age                         3880 non-null   int8
 2   gender                      3880 non-null   int8
 3   residence                   3880 non-null   int8
 4   city                        3880 non-null   int16
 5   city_rank                   3880 non-null   int8
 6   series_dev                  3880 non-null   int8
 7   series_group                3880 non-null   int8
 8   emui_dev                    3880 non-null   int8
 9   device_name                 3880 non-null   int16
 10  device_size                 3880 non-null   int16
 11  net_type                    3880 non-null   int8
 12  creat_type_cd               3880 non-null   int8
 13  slot_id                     3880 non-null   int8
 14  u_refreshTimes              3880 non-null   int8
 15  u_feedLifeCycle             3880 non-null   int8
 16  u_newsCatInterestsST_count  3880 non-null   int8
 17  click_count_mean            3880 non-null   float32
 18  close_count_mean            3880 non-null   float32
dtypes: float32(2), int16(3), int8(14)
memory usage: 136.4 KB
```

saving this so we do not have to rerun

```
import joblib

# Save D_processed (all 6 subsets) to Google Drive
joblib.dump(D_processed, '/content/drive/MyDrive/D_processed.joblib')
print("✅ Saved D_processed to Google Drive")
print(f"Saved {len(D_processed)} subsets: {list(D_processed.keys())}")
```

```
✅ Saved D_processed to Google Drive
Saved 6 subsets: ['D1', 'D2', 'D3', 'D4', 'D5', 'D6']
```

```
#if session restarts can load back in like this
from google.colab import drive
import joblib

# Mount Google Drive
drive.mount('/content/drive')

# Copy from Drive to local (faster access)
!cp /content/drive/MyDrive/D_processed.joblib /content/

# Load it back
D_processed = joblib.load('/content/D_processed.joblib')

# Verify it loaded correctly
print(f"Loaded D_processed with {len(D_processed)} subsets")
print(f"Subsets: {list(D_processed.keys())}")

# Check shapes to confirm
for name, df in D_processed.items():
    print(f"  {name}: {df.shape}")
```

```
Mounted at /content/drive
Loaded D_processed with 6 subsets
```

```
Subsets: ['D1', 'D2', 'D3', 'D4', 'D5', 'D6']
  D1: (4628, 19)
  D2: (29359, 19)
  D3: (2724, 19)
  D4: (3880, 19)
  D5: (12677, 19)
  D6: (1555, 19)
```

(4) Data Generation

From the github - The model implements an optimal stopping criterion based on the synthetic data distribution when training a non-relational tabular model. The model will stop training when the synthetic data distribution is close to the real data distribution.

Make sure to set the epochs parameter to a large number to allow the model to fit the data better. The model will stop training when the optimal stopping criterion is met.

4.1 Editing bug in realtabformer package

```
pip install -U git+https://github.com/worldbank/REaLTabFormer.git
```

Show hidden output

```
!git clone https://github.com/worldbank/REaLTabFormer.git
```

```
Cloning into 'REaLTabFormer'...
remote: Enumerating objects: 1197, done.
remote: Counting objects: 100% (255/255), done.
remote: Compressing objects: 100% (99/99), done.
remote: Total 1197 (delta 167), reused 188 (delta 123), pack-reused 942 (from 1)
Receiving objects: 100% (1197/1197), 14.48 MiB | 17.85 MiB/s, done.
Resolving deltas: 100% (584/584), done.
```

Steps to fix it since I think you need to need to edit it every time the session restarts

1. run all chunks above
2. go to files -> content -> REaLTabFormer -> src -> realtabformer -> open realtabformber.py
3. scroll to find this code block

```
if _delta_mean_sensitivity_value < best_mean_sensitivity_value:
               best_mean_sensitivity_value = _delta_mean_sensitivity_value
               trainer.save_model(mean_closest_bdm_path.as_posix())
               trainer.state.save_to_json(
                   (mean_closest_bdm_path / "trainer_state.json").as_posix()
               )
```

4. then you want to add this code underneath it

```
           if not any(os.listdir(not_bdm_path.as_posix())):
             trainer.save_model(not_bdm_path.as_posix())
             trainer.state.save_to_json(
               (not_bdm_path / "trainer_state.json").as_posix()
             )
```

5. do command/control s to save it
6. now can run the other code blocks

```
import sys
sys.path.insert(0, "/content/REaLTabFormer/src")
import os
os.environ["WANDB_DISABLED"] = "true"
from realtabformer.realtabformer import REaLTabFormer
import shutil

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

```
DRIVE_MODELS_DIR = "/content/drive/MyDrive/realtab_models"
os.makedirs(DRIVE_MODELS_DIR, exist_ok=True)
```

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

Setara's version

```
def train_and_save_rtf(df, dataset_name, drive_models_dir=DRIVE_MODELS_DIR):
    """
    df           : preprocessed dataframe (one of your D1–D6 subsets)
    dataset_name : e.g. 'D1', 'D2', etc.
    """
    print(f"\n=== Training REaLTabFormer for {dataset_name} (rows={len(df):,}) ===")

    # 1. Instantiate model (tabular GPT-2)
    rtf_model = REaLTabFormer(
        model_type="tabular",
        gradient_accumulation_steps=4,
        logging_steps=100,
        epochs=75,  # ADDED: Max epochs limit (model may stop earlier)
        batch_size=128
    )

    # 2. Fit on this subset
    rtf_model.fit(
        df,
        num_bootstrap=3,   # REDUCED from 10 to 3 (cuts training time roughly in half)
        frac=0.1,           # fraction per bootstrap
        full_sensitivity=False,
        n_critic_stop=1,
        n_critic=1,
    )

    print("Finished training.")

    # 3. Save model locally
    local_model_dir = f"/content/rtf_{dataset_name}"
    rtf_model.save(local_model_dir)
    print(f"Saved local model to: {local_model_dir}")

    # 4. Zip and copy to Google Drive
    zip_base = f"/content/rtf_{dataset_name}"      # base name for archive
    zip_path = shutil.make_archive(zip_base, "zip", root_dir=local_model_dir)  # FIXED: added root_dir parameter
    drive_zip_path = os.path.join(drive_models_dir,
                                  os.path.basename(zip_path))
    shutil.move(zip_path, drive_zip_path)

    print(f"Zipped model and saved to Drive: {drive_zip_path}")
    return local_model_dir, drive_zip_path

model_dirs = {}
for name in ["D2", "D3"]:  # Only D1, D2, D3 #already ran D1 so hopefully it saved
    if name in D_processed:  # Safety check
        df_sub = D_processed[name]
        local_dir, drive_zip = train_and_save_rtf(df_sub, name)
        model_dirs[name] = {"local_dir": local_dir, "drive_zip": drive_zip}
    else:
        print(f"Warning: {name} not found in D_processed")

print("\n✅ Finished training and saving REaLTabFormer models for D1–D3.")
```

```
=== Training REaLTabFormer for D2 (rows=29,359) ===
Computing the sensitivity threshold...
Using parallel computation!!!
ate (0.6267584045778126) in                    the data. This will not give a reliable early stopping condition. Con
  warnings.warn(
```

Bootstrap round: 100% 3/3 [00:00<00:00, 106.66it/s]

```
Sensitivity threshold summary:
count    3.000000
mean     0.270889
std      0.004903
min      0.266204
25%      0.268342
50%      0.270479
75%      0.273232
max      0.275984
dtype: float64
Sensitivity threshold: 0.27543376652769486 qt_max: 0.05
```

Map: 100% 29359/29359 [00:17<00:00, 1710.88 examples/s]

```
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
`loss_type=None` was set in the config but it is unrecognized. Using the default loss: `ForCausalLMLoss`.
```

[58/58 00:10, Epoch 1/1]

Step	Training Loss

8832/? [00:56<00:00, 155.74it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 1,                    sensitivity_threshold: 0.27543376652769486,                    val_sensitiv
```

[116/116 00:11, Epoch 2/2]

Step	Training Loss
100	0.456000

8832/? [00:57<00:00, 157.16it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 2,                    sensitivity_threshold: 0.27543376652769486,                    val_sensitiv
```

[174/174 00:10, Epoch 3/3]

Step	Training Loss

8832/? [00:56<00:00, 155.70it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 3,                    sensitivity_threshold: 0.27543376652769486,                    val_sensitiv
```

[232/232 00:10, Epoch 4/4]

Step	Training Loss
200	0.385600

8832/? [00:56<00:00, 155.42it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 4,                    sensitivity_threshold: 0.27543376652769486,                    val_sensitiv
```

[290/290 00:10, Epoch 5/5]

Step	Training Loss

8832/? [00:57<00:00, 152.32it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 5,                    sensitivity_threshold: 0.27543376652769486,                    val_sensitiv
```

[348/348 00:11, Epoch 6/6]

Step	Training Loss
300	0.349000

8832/? [00:57<00:00, 153.83it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 6,                    sensitivity_threshold: 0.27543376652769486,                    val_sensitiv
```

[406/406 00:11, Epoch 7/7]

Step	Training Loss
400	0.323300

8832/? [00:57<00:00, 152.98it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 7,                    sensitivity_threshold: 0.27543376652769486,                        val_sensitiv
```

[464/464 00:10, Epoch 8/8]

Step	Training Loss

8832/? [00:57<00:00, 154.08it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 8,                    sensitivity_threshold: 0.27543376652769486,                        val_sensitiv
```

[522/522 00:11, Epoch 9/9]

Step	Training Loss
500	0.294200

8832/? [00:58<00:00, 150.61it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 9,                    sensitivity_threshold: 0.27543376652769486,                        val_sensitiv
```

[580/580 00:10, Epoch 10/10]

Step	Training Loss

8832/? [00:59<00:00, 151.36it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 10,                     sensitivity_threshold: 0.27543376652769486,                         val_sensiti
```

[638/638 00:11, Epoch 11/11]

Step	Training Loss
600	0.277500

8832/? [00:58<00:00, 152.70it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 11,                     sensitivity_threshold: 0.27543376652769486,                         val_sensiti
```

[696/696 00:10, Epoch 12/12]

Step	Training Loss

8832/? [00:58<00:00, 153.49it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 12,                     sensitivity_threshold: 0.27543376652769486,                         val_sensiti
```

[754/754 00:11, Epoch 13/13]

Step	Training Loss
700	0.265400

8832/? [00:58<00:00, 149.41it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 13,                     sensitivity_threshold: 0.27543376652769486,                         val_sensiti
```

[812/812 00:10, Epoch 14/14]

Step	Training Loss
800	0.261200

8832/? [00:59<00:00, 150.11it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 14,                     sensitivity_threshold: 0.27543376652769486,                         val_sensiti
```

[870/870 00:10, Epoch 15/15]

Step	Training Loss

8832/? [01:01<00:00, 146.12it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 15,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[928/928 00:11, Epoch 16/16]

Step	Training Loss
900	0.255200

8832/? [01:06<00:00, 132.62it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 16,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[986/986 00:10, Epoch 17/17]

Step	Training Loss

8832/? [01:04<00:00, 134.39it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 17,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1044/1044 00:11, Epoch 18/18]

Step	Training Loss
1000	0.249600

8832/? [01:02<00:00, 140.54it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 18,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1102/1102 00:11, Epoch 19/19]

Step	Training Loss
1100	0.247700

8832/? [01:00<00:00, 147.01it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 19,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1160/1160 00:10, Epoch 20/20]

Step	Training Loss

8832/? [01:01<00:00, 147.02it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 20,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1218/1218 00:11, Epoch 21/21]

Step	Training Loss
1200	0.244200

8832/? [00:59<00:00, 150.17it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 21,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1276/1276 00:10, Epoch 22/22]

Step	Training Loss

8832/? [01:00<00:00, 146.60it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 22,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1334/1334 00:11, Epoch 23/23]

Step	Training Loss
1300	0.241500

8832/? [01:00<00:00, 146.86it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 23,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1392/1392 00:10, Epoch 24/24]

Step	Training Loss

8832/? [01:00<00:00, 145.99it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 24,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1450/1450 00:11, Epoch 25/25]

Step	Training Loss
1400	0.240000

8832/? [01:00<00:00, 146.79it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 25,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1508/1508 00:11, Epoch 26/26]

Step	Training Loss
1500	0.237900

8832/? [01:01<00:00, 143.58it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 26,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1566/1566 00:10, Epoch 27/27]

Step	Training Loss

8832/? [01:00<00:00, 148.39it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 27,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1624/1624 00:11, Epoch 28/28]

Step	Training Loss
1600	0.236100

8832/? [01:00<00:00, 145.56it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 28,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1682/1682 00:10, Epoch 29/29]

Step	Training Loss

8832/? [01:00<00:00, 145.34it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 29,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1740/1740 00:11, Epoch 30/30]

Step	Training Loss
1700	0.233200

8832/? [01:00<00:00, 144.79it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 30,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1798/1798 00:10, Epoch 31/31]

Step	Training Loss

8832/? [01:00<00:00, 145.86it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 31,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1856/1856 00:12, Epoch 32/32]

Step	Training Loss
1800	0.231400

8832/? [01:00<00:00, 144.82it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 32,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1914/1914 00:11, Epoch 33/33]

Step	Training Loss
1900	0.231900

8832/? [01:01<00:00, 144.13it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 33,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[1972/1972 00:10, Epoch 34/34]

Step	Training Loss

8832/? [01:01<00:00, 144.43it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 34,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2030/2030 00:11, Epoch 35/35]

Step	Training Loss
2000	0.230000

8832/? [01:02<00:00, 143.14it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 35,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2088/2088 00:10, Epoch 36/36]

Step	Training Loss

8832/? [01:01<00:00, 141.98it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 36,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2146/2146 00:11, Epoch 37/37]

Step	Training Loss
2100	0.228400

8832/? [01:01<00:00, 143.45it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 37,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2204/2204 00:11, Epoch 38/38]

Step	Training Loss
2200	0.229000

8832/? [01:01<00:00, 143.95it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 38,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2262/2262 00:10, Epoch 39/39]

Step	Training Loss

8832/? [01:01<00:00, 142.11it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 39,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2320/2320 00:11, Epoch 40/40]

Step	Training Loss

2300	0.227600

8832/? [01:01<00:00, 142.48it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 40,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2378/2378 00:10, Epoch 41/41]

Step	Training Loss

8832/? [01:01<00:00, 142.53it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 41,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2436/2436 00:11, Epoch 42/42]

Step	Training Loss
2400	0.226300

8832/? [01:02<00:00, 140.48it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 42,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2494/2494 00:10, Epoch 43/43]

Step	Training Loss

8832/? [01:02<00:00, 141.04it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 43,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2552/2552 00:11, Epoch 44/44]

Step	Training Loss
2500	0.223800

8832/? [01:02<00:00, 142.25it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 44,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2610/2610 00:11, Epoch 45/45]

Step	Training Loss
2600	0.225500

8832/? [01:04<00:00, 138.27it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 45,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2668/2668 00:10, Epoch 46/46]

Step	Training Loss

8832/? [01:03<00:00, 140.37it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 46,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2726/2726 00:11, Epoch 47/47]

Step	Training Loss
2700	0.225100

8832/? [01:02<00:00, 141.02it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 47,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2784/2784 00:10, Epoch 48/48]

Step	Training Loss

8832/? [01:03<00:00, 139.23it/s]

8832/? [01:03<00:00, 139.23it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 48,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2842/2842 00:11, Epoch 49/49]

Step	Training Loss
2800	0.223700

8832/? [01:03<00:00, 140.33it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 49,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2900/2900 00:10, Epoch 50/50]

Step	Training Loss
2900	0.224300

8832/? [01:03<00:00, 141.18it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 50,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[2958/2958 00:10, Epoch 51/51]

Step	Training Loss

8832/? [01:03<00:00, 140.11it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 51,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3016/3016 00:11, Epoch 52/52]

Step	Training Loss
3000	0.223100

8832/? [01:06<00:00, 134.20it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 52,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3074/3074 00:10, Epoch 53/53]

Step	Training Loss

8832/? [01:10<00:00, 125.11it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 53,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3132/3132 00:11, Epoch 54/54]

Step	Training Loss
3100	0.221600

8832/? [01:09<00:00, 125.58it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 54,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3190/3190 00:11, Epoch 55/55]

Step	Training Loss

8832/? [01:06<00:00, 130.07it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 55,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3248/3248 00:11, Epoch 56/56]

Step	Training Loss
3200	0.221300

8832/? [01:05<00:00, 135.18it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
```

```
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 56,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3306/3306 00:11, Epoch 57/57]

Step	Training Loss
3300	0.221600

8832/? [01:05<00:00, 136.18it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 57,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3364/3364 00:11, Epoch 58/58]

Step	Training Loss

8832/? [01:06<00:00, 133.54it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 58,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3422/3422 00:11, Epoch 59/59]

Step	Training Loss
3400	0.220400

8832/? [01:04<00:00, 138.15it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 59,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3480/3480 00:10, Epoch 60/60]

Step	Training Loss

8832/? [01:03<00:00, 138.72it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 60,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3538/3538 00:11, Epoch 61/61]

Step	Training Loss
3500	0.220300

8832/? [01:05<00:00, 132.65it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 61,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3596/3596 00:10, Epoch 62/62]

Step	Training Loss

8832/? [01:04<00:00, 136.35it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 62,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3654/3654 00:11, Epoch 63/63]

Step	Training Loss
3600	0.219700

8832/? [01:04<00:00, 137.14it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 63,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[3712/3712 00:11, Epoch 64/64]

Step	Training Loss
3700	0.219600

8832/? [01:04<00:00, 138.45it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
```

Critic round: 64, sensitivity_threshold: 0.27543766527694866, val_sensiti

[3770/3770 00:10, Epoch 65/65]

Step	Training Loss

8832/? [01:04<00:00, 138.36it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 65, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[3828/3828 00:11, Epoch 66/66]

Step	Training Loss
3800	0.218800

8832/? [01:05<00:00, 137.08it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 66, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[3886/3886 00:10, Epoch 67/67]

Step	Training Loss

8832/? [01:04<00:00, 134.07it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 67, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[3944/3944 00:12, Epoch 68/68]

Step	Training Loss
3900	0.218000

8832/? [01:05<00:00, 136.19it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 68, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[4002/4002 00:11, Epoch 69/69]

Step	Training Loss
4000	0.218600

8832/? [01:04<00:00, 136.23it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 69, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[4060/4060 00:10, Epoch 70/70]

Step	Training Loss

8832/? [01:05<00:00, 137.91it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 70, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[4118/4118 00:11, Epoch 71/71]

Step	Training Loss
4100	0.218000

8832/? [01:05<00:00, 136.17it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 71, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[4176/4176 00:11, Epoch 72/72]

Step	Training Loss

8832/? [01:05<00:00, 136.29it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 72, sensitivity_threshold: 0.27543376652769486, val_sensiti
```

[4234/4234 00:11, Epoch 73/73]

Step	Training Loss

4200	0.217800

8832/? [01:04<00:00, 132.43it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 73,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[4292/4292 00:10, Epoch 74/74]

Step	Training Loss

8832/? [01:05<00:00, 133.76it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 74,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
```

[4350/4350 00:11, Epoch 75/75]

Step	Training Loss
4300	0.217100

8832/? [01:06<00:00, 134.30it/s]

```
Generated 0 invalid samples out of total 8832 samples generated. Sampling efficiency is: 100.0000%
Critic round: 75,                    sensitivity_threshold: 0.27543376652769486,                        val_sensiti
Finished training.
Copying artefacts from: best-disc-model
Copying artefacts from: mean-best-disc-model
Copying artefacts from: not-best-disc-model
Copying artefacts from: last-epoch-model
Saved local model to: /content/rtf_D2
Zipped model and saved to Drive: /content/drive/MyDrive/realtab_models/rtf_D2.zip

=== Training REaLTabFormer for D3 (rows=2,724) ===
Computing the sensitivity threshold...
Using parallel computation!!!
ate (0.8667400881057269) in                    the data. This will not give a reliable early stopping condition. Con
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/realtabformer/realtabformer.py:597: UserWarning: qt_interval adjusted from 10
  warnings.warn(
```

Bootstrap round: 100% 3/3 [00:00<00:00, 145.52it/s]

```
Sensitivity threshold summary:
count    3.000000
mean     0.501523
std      0.040585
min      0.466420
25%      0.479304
50%      0.492188
75%      0.519075
max      0.545962
dtype: float64
Sensitivity threshold: 0.5405849816476346 qt_max: 0.05
```

Map: 100% 2724/2724 [00:01<00:00, 1869.38 examples/s]

```
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
```

[6/6 00:02, Epoch 1/1]

Step	Training Loss

896/? [00:05<00:00, 169.17it/s]

```
Generated 0 invalid samples out of total 896 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 1,                     sensitivity_threshold: 0.5405849816476346,                         val_sensitivi
```

[12/12 00:02, Epoch 2/2]

Step	Training Loss

896/? [00:05<00:00, 171.55it/s]

```
Generated 0 invalid samples out of total 896 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 2,                     sensitivity_threshold: 0.5405849816476346,                         val_sensitivi
```

[18/18 00:02, Epoch 3/3]

Step	Training Loss

896/? [00:05<00:00, 169.63it/s]

```
Generated 0 invalid samples out of total 896 samples generated. Sampling efficiency is: 100.0000%
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to
There were missing keys in the checkpoint model loaded: ['lm_head.weight'].
Critic round: 3,                     sensitivity_threshold: 0.5405849816476346,                         val_sensitivi
```

[24/24 00:02, Epoch 4/4]