analysisv4

August 18, 2025

1 HybridNet Evaluation

Version: 2.1.0 **Date:** Mon Aug 18th 2025

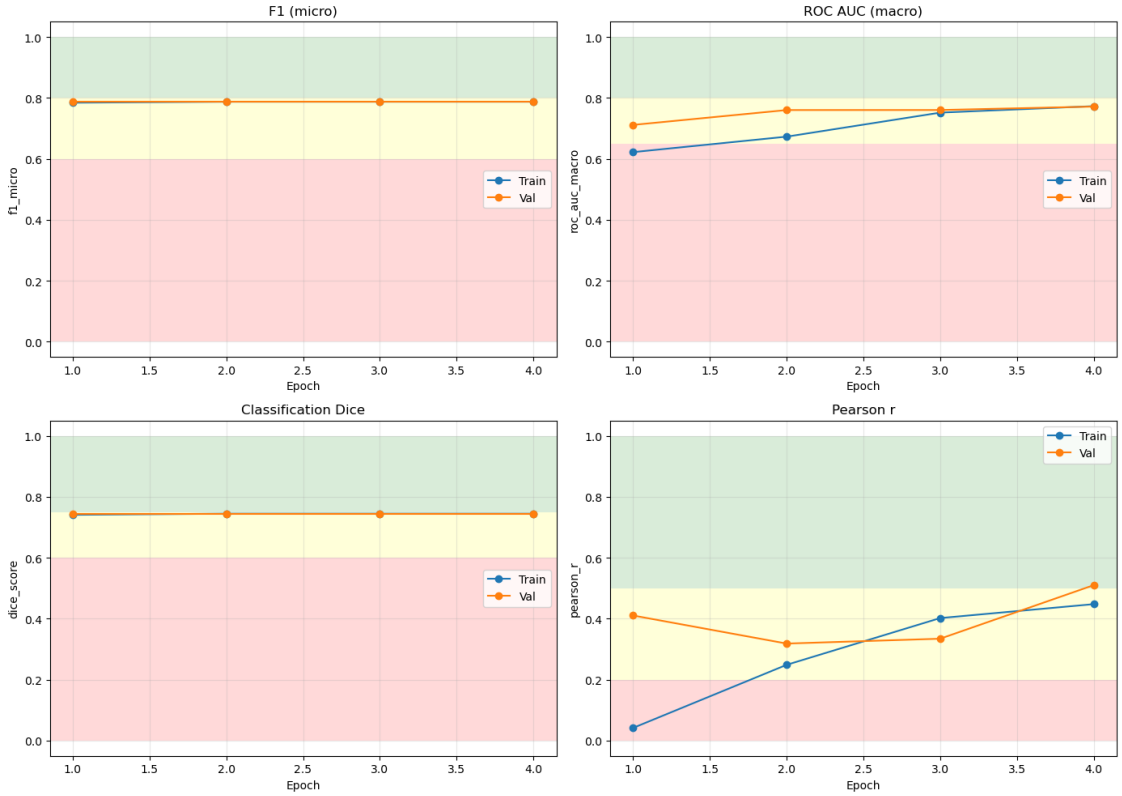
[6]:
```
import json
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

TRAIN = r'/Users/jbalkovec/Desktop/DR/experiments/model_logs/
 ↪hybrid_train_history.csv'
VAL = r'/Users/jbalkovec/Desktop/DR/experiments/model_logs/hybrid_val_history.
 ↪csv'
JSON_FILE = r'/Users/jbalkovec/Desktop/DR/experiments/model_logs/
 ↪hybrid_per_class_and_thresh.json'

BANDS = {
    "f1_micro":      [(0.0, 0.60, "red"), (0.60, 0.80, "yellow"), (0.80, 1.00,␣
 ↪"green")],
    "roc_auc_macro": [(0.0, 0.65, "red"), (0.65, 0.80, "yellow"), (0.80, 1.00,␣
 ↪"green")],
    "dice_score":    [(0.0, 0.60, "red"), (0.60, 0.75, "yellow"), (0.75, 1.00,␣
 ↪"green")],
    "pearson_r":     [(0.0, 0.20, "red"), (0.20, 0.50, "yellow"), (0.50, 1.00,␣
 ↪"green")],
}

def plot_with_bands(ax, df_train, df_val, metric, title=None):
    # background bands
    for low, high, color in BANDS[metric]:
        ax.axhspan(low, high, facecolor=color, alpha=0.15)
    # plot train vs val
    ax.plot(df_train["epoch"], df_train[metric], marker="o", label="Train")
    ax.plot(df_val["epoch"],   df_val[metric],   marker="o", label="Val")
    ax.set_xlabel("Epoch")
    ax.set_ylabel(metric)
    ax.set_title(title or metric)
```

```
    ax.grid(True, alpha=0.3)
    ax.legend()
```

[5]:
```
train_df = pd.read_csv(TRAIN)
val_df   = pd.read_csv(VAL)
with open(JSON_FILE, "r") as f:
    j = json.load(f)
```

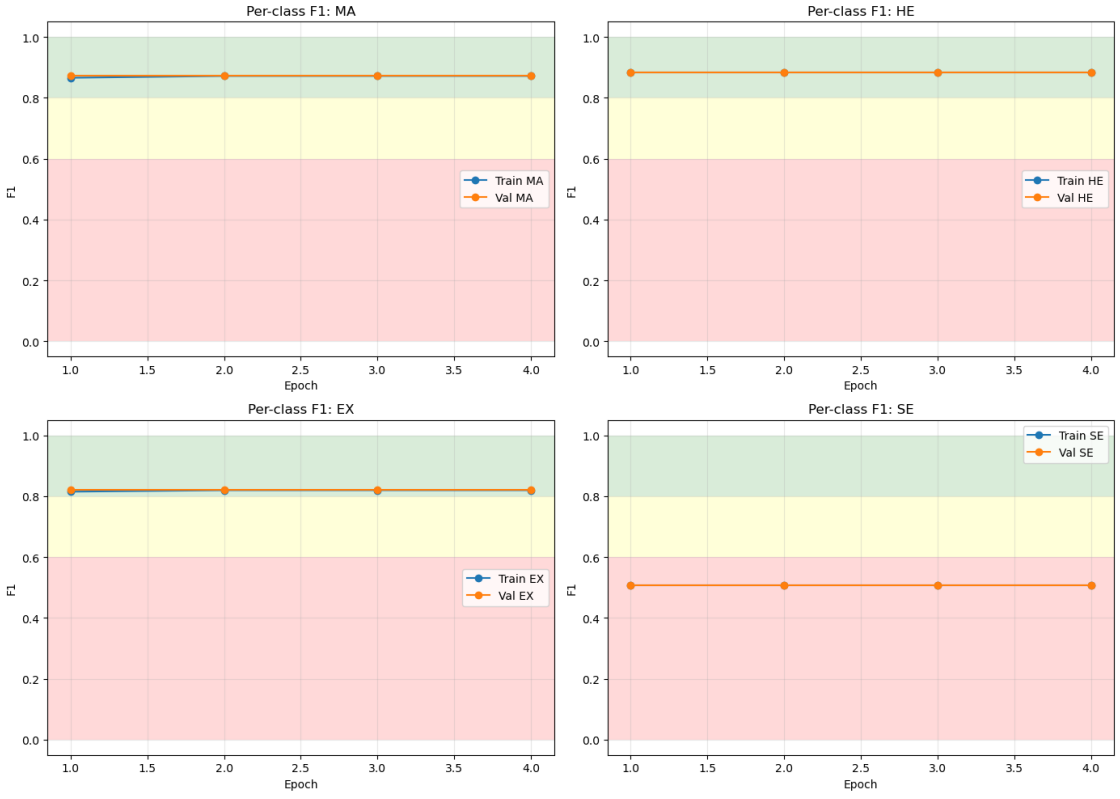
1.1 Core Metrics

[7]:
```
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
metrics = ["f1_micro", "roc_auc_macro", "dice_score", "pearson_r"]
titles  = ["F1 (micro)", "ROC AUC (macro)", "Classification Dice", "Pearson r"]

for ax, m, t in zip(axes.ravel(), metrics, titles):
    if m in train_df and m in val_df:
        plot_with_bands(ax, train_df, val_df, m, t)
plt.tight_layout()
plt.show()
```



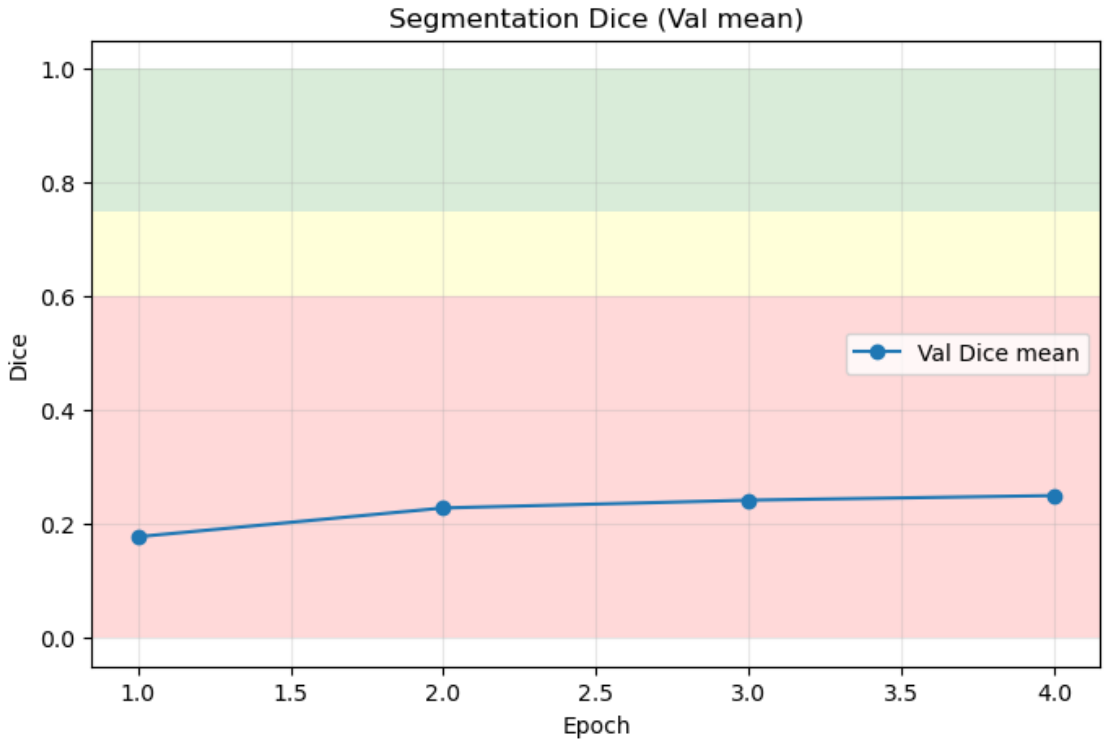
1.2 Per Class F1

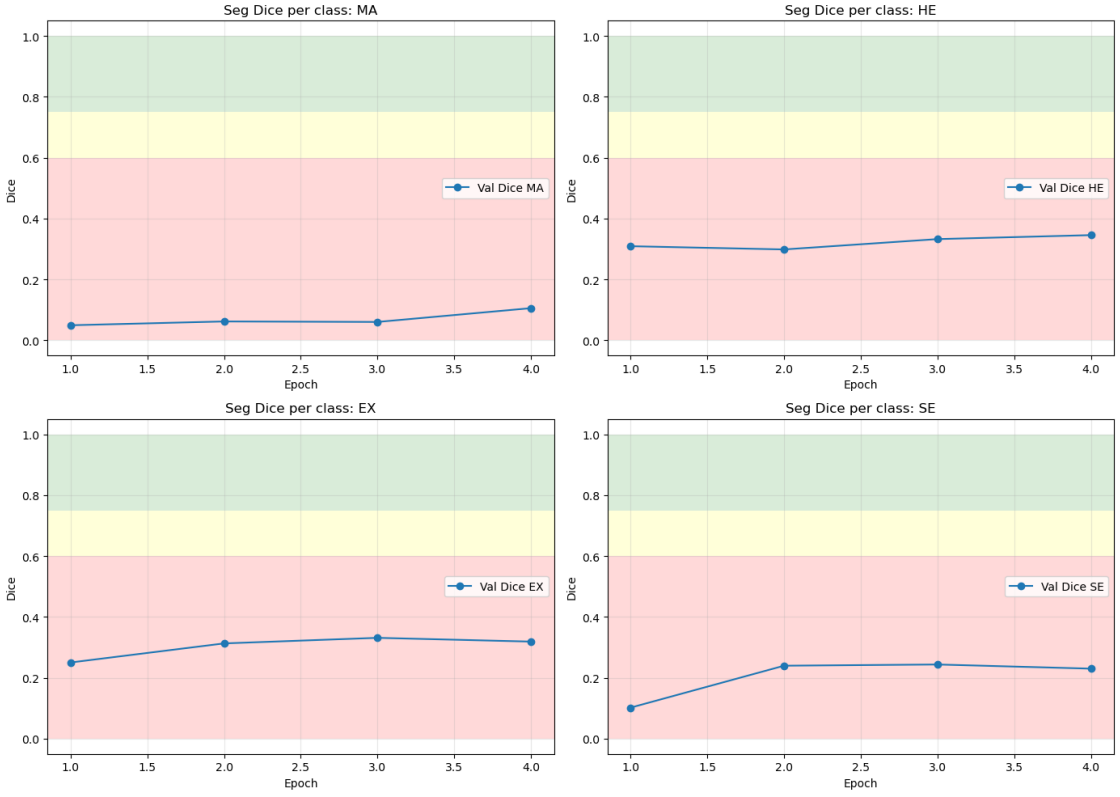
[9]:
```
train_f1 = np.array(j["train_f1"], dtype=float)
val_f1   = np.array(j["val_f1"],   dtype=float)
epochs   = np.arange(1, train_f1.shape[0] + 1)
lesions  = ["MA", "HE", "EX", "SE"]

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
for i, lesion in enumerate(lesions):
    ax = axes[i // 2, i % 2]
    for low, high, color in BANDS["f1_micro"]:
        ax.axhspan(low, high, facecolor=color, alpha=0.15)
    ax.plot(epochs, train_f1[:, i], marker="o", label=f"Train {lesion}")
    ax.plot(epochs, val_f1[:, i],   marker="o", label=f"Val {lesion}")
    ax.set_title(f"Per-class F1: {lesion}")
    ax.set_xlabel("Epoch")
    ax.set_ylabel("F1")
    ax.grid(True, alpha=0.3)
    ax.legend()
plt.tight_layout()
plt.show()
```



1.3 Segmentation Dice

[10]:
```
val_dice_mean = np.array(j.get("val_seg_dice_mean", []), dtype=float)
val_dice_pc   = np.array(j.get("val_seg_dice_per_class", []), dtype=float)
if val_dice_mean.size > 0:
    fig, ax = plt.subplots(figsize=(8, 5))
    for low, high, color in BANDS["dice_score"]:
        ax.axhspan(low, high, facecolor=color, alpha=0.15)
    ax.plot(epochs, val_dice_mean, marker="o", label="Val Dice mean")
    ax.set_title("Segmentation Dice (Val mean)")
    ax.set_xlabel("Epoch")
    ax.set_ylabel("Dice")
    ax.grid(True, alpha=0.3)
    ax.legend()
    plt.show()

    if val_dice_pc.ndim == 2 and val_dice_pc.shape[1] == 4:
        fig, axes = plt.subplots(2, 2, figsize=(14, 10))
        for i, lesion in enumerate(lesions):
            ax = axes[i // 2, i % 2]
            for low, high, color in BANDS["dice_score"]:
                ax.axhspan(low, high, facecolor=color, alpha=0.15)
            ax.plot(epochs, val_dice_pc[:, i], marker="o", label=f"Val Dice␣
 ↪{lesion}")
            ax.set_title(f"Seg Dice per class: {lesion}")
            ax.set_xlabel("Epoch")
            ax.set_ylabel("Dice")
            ax.grid(True, alpha=0.3)
            ax.legend()
        plt.tight_layout()
        plt.show()
```


Segmentation Dice (Val mean)
Dice
Epoch
Val Dice mean
1.0
0.8
0.6
0.4
0.2
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Seg Dice per class: MA
Val Dice MA
Seg Dice per class: HE
Val Dice HE
Seg Dice per class: EX
Val Dice EX
Seg Dice per class: SE
Val Dice SE
Dice
Epoch