Ayna Assignment: Polygon Colorization with Conditional UNet

Hyperparameter Settings

Parameter	Value	Description
EPOCHS	150	Number of training epochs
LR	1e-3	Learning rate for AdamW optimizer
BASE_CH	32	Base number of channels in the UNet
COND_METHOD	"film"	Conditioning method used (others defined but not used in training)
BATCH_SIZE	32	Batch size for training and validation
LOSS	0.7 × L1 + 0.3 × MSE	Weighted combination of L1 and MSE losses for reconstruction

Rationale:

- L1 is used for sharpness and pixel-level accuracy.
- MSE is used for smoothness and stable convergence.
- FiLM (Feature-wise Linear Modulation) allows dynamic conditioning based on the color input.
- AdamW is chosen for better regularization than Adam.

Model Design

Architecture Overview

- Based on a **UNet** design with skip connections.
- Input:
  - RGB polygon outline image
  - Color conditioning input (both index and RGB)
- Output: Color-filled polygon image.

Conditioning Strategy

- Uses a FiLM-based conditioning , where:
  - A learnable embedding maps the color index to a modulation vector.
  - This is applied to the intermediate features of the UNet using scaling and bias.

Training Dynamics

Features

- Full training + validation loop implemented.
- Gradient scaling is used for faster/more stable training on GPU.
- Model checkpoint saved based on **best validation loss**.
- PSNR (Peak Signal-to-Noise Ratio) computed for visual quality estimation.

Metric Tracking

- Tracked via:
  - `train_loss`, `val_loss`
  - `mse`, `l1`, `psnr` for both training and validation
- Weights & Biases (wandb) integration for visual metric tracking (if enabled).

Experiment Results

Output Quality

- In most cases, the model successfully fills the correct polygon with the intended color.
- The output images have smooth edges and accurate boundaries due to the UNet and resizing.

Observed Failure Modes

- Slight color bleed or incomplete fills in low-contrast shapes.
- Uncommon shapes like nanogon showed less accurate fills — likely due to underrepresentation in training data.

Key Insights and Learnings

- **FiLM-based conditioning** was effective at integrating class-wise information (color) into the UNet pipeline.
- The `0.7 L1 + 0.3 MSE` loss gave a strong balance of edge-preserving sharpness and smooth region fill.
- Using `GradScaler` improved performance on GPU-enabled Colab sessions.
- wandb helped visualize training convergence and quickly identify performance drops.
- Adding more shape/color diversity or trying other conditioning methods like `concat_rgb` could further improve generalization.