5/4

Summary of possible heatmaps produced for single-image when human prediction per image are known.

When a neural network predicts a **realism score** for an image using DNN feature maps, **in an experimental context where the image's true realism score is known**, we can analyze which feature maps help or hurt the prediction for a specific image.

We define:

- y_true: the ground-truth realism score
- y_pred: the model's predicted realism score
- y_pred_(-i): the prediction when **feature map i is removed**

From this, we compute:

- **Residual error before removal**: residual = |y_pred - y_true|
- **Residual error after removal**: residual_(-i) = |y_pred_(-i) - y_true|
- **Δerror = residual_(-i) - residual** → how prediction error changes when we remove a feature map. **Positive values indicate an important feature**, because its removal reduces prediction performance.
- **Δy = y_pred_(-i) - y_pred** → how the predicted realism score changes when we remove the feature map. **Positive values indicate the feature map contained "strange" content** because its removal increases realism.

📊 Interpretation Matrix

We can categorize each feature map based on:

- Whether it **helps or hurts the prediction** (Δerror)
- Whether it **pushes realism up or down** (Δy)

Δerror (change in residual)	**Δy (change in predicted realism)**	**Meaning**
> 0 (error ↑)	**< 0** (realism ↓)	✅ *Helpful* feature map that **increases realism**
> 0 (error ↑)	**> 0** (realism ↑)	✅ *Helpful* feature map that **decreases realism**
< 0 (error ↓)	**< 0** (realism ↓)	🚫 *Harmful* feature map that **increases realism**
< 0 (error ↓)	**> 0** (realism ↑)	🚫 *Harmful* feature map that **decreases realism**

□ Two Ways to Use This Information

1. Explaining Human Judgements of Realism (Conceptual Insight)

If your goal is to understand **human judgments**, focus on:

- Feature maps where **Δerror > 0** (removing them hurts the prediction)
- These are **important** feature maps (from human-oriented AI perspective): they provide meaningful information related to realism

You can further group them by whether they:

- Increase realism (Δy < 0)
- Decrease realism (Δy > 0)

This tells you which parts of the image **boost or suppress realism** in a useful way. They suggest this might be information that is relevant to explaining human judgments.

2. An Alternative Heatmap Approach: **Explaining Model Behavior (Debugging Insight)**

If your goal is to understand **how the model works** (or fails; i.e, explainable AI; XAI), also look at:

- Feature maps where **Δerror < 0** (removal improves prediction)
- These are **harmful or misleading** features—possibly reflecting noise, bias, or overfitting

They help you understand **why the model made a mistake** or **what information it's misusing**.

✅ So What Should You Show in Your Visualization?

A **good compromise**:

1. **Primary map**: Highlight *only important*, **helpful features**. This gives a **conceptual explanation of realism as perceived by humans**. We can use
    1. Red = "This region contributes positively to perceived realism"
    2. Blue = "This region contributes negatively to perceived realism"
2. **Secondary layer** (optional): Gray out or outline *harmful* or *misleading* feature maps to show where the model got confused. Helps with **model interpretability**.

Conceptual notes

There are two potentially different goals for this sub-project: **explaining model behavior** (Explainable AI) vs. **explaining human behavior** (human-aligned AI).

🎯 Two Types of Explainability

Goal	**Ground Truth Needed?**	**Key Signal**	**What It Explains**
Explainable AI *(model-behavior attribution)*	❌ No	**Δy** (change in prediction)	What the **model** focuses on to make a prediction
Human-Aligned AI *(human-behavior attribution)*	✅ Yes (y_true needed)	**Δerror** (change in residual error)	What **factors** are **actually informative** for predicting human realism judgments

---

✅ Model-Behavior Attribution (Explainable AI)

- **What it shows**: Which feature maps push the **model's prediction** of realism up or down.

- **Signal used**:
  → **Δy = y_pred_(-i) - y_pred**
- **What it's good for**:
  - Auditing the model
  - Understanding what parts of the input influence predictions
  - Generating saliency or attention maps (e.g., for user interfaces). NOTE: it would be interesting to compare the saliency maps produced by this method to attention maps produced e..g, by TranSalNet. GitHub - LJOVO/TranSalNet: TranSalNet: Towards perceptually relevant visual saliency prediction. Neurocomputing (2022)
- **Limitations**:
  - Does **not** tell you whether the model is *right* (with respect to ground truth)
  - Can highlight features that are *influential* but *not helpful*

☑ Human-Oriented Attribution (Human Behavior Explanation)

- **What it shows**: Which feature maps **actually encode information** predictive of **human realism ratings**
- **Signal used**:
  → **Δerror = |y_pred_(-i) - y_true| - |y_pred - y_true|**
- **What it's good for**:
  - Discovering perceptually relevant visual features
  - Explaining what makes an image realistic from a **human-centered** perspective
  - Interpreting the semantics of the representation
- **Limitations**:
  - Requires **ground truth**
  - Can't be used on new, unlabeled data

□ Analogy

Think of the model as a **student trying to guess a human rating**:

- **Δy** tells you what the student *believes* is important—it's the **student's reasoning**
- **Δerror** tells you what information actually **helps the student get closer to the human answer**—it's the **true learning**

🔎 To understand **the model**, use **Δy**
□ To understand **the concept of realism**, use **Δerror**

🚦Visual Interpretation

Attribution Type	**Color Map Meaning**	**Requires y_true?**
Model-Behavior (Δy)	Red = pushes realism ↑, Blue = pushes realism ↓	✖ No
Human-Behavior (Δerror)	Green = helpful info, Red = harmful info	☑ Yes

You can combine both in visualizations:

- Use **hue** to encode Δy (direction of influence)
- Use **saturation/brightness** to reflect Δerror (strength and helpfulness)

Icaro, note that this current exercise requires y_true, i.e, the human judgment per image. In previous weeks, where we learned a pruning from the entire dataset (and showed it generalizes) we found subsets of important features at the dataset level, which generalized to new data and allowed us to create heatmaps for new images (ones we don't have human judgments). In this context we can use the method described here, where we use saturation to reflect the importance of each heatmap for prediction ("AIS" score) and red/blue to indicate whether its removal pushes realism up or down.