Center-Surround Model Evolution

From Classical Klindt to Dedicated ON/OFF Mixed

Overview

1. **Problem:** Modeling retinal ganglion cell receptive fields
2. **Challenge:** Capturing spatial structure + polarity (ON/OFF)
3. **Solution:** Progressive model refinement

Model Architecture Overview

```
┌──────────────┐     ┌─────────────────┐     ┌────────────┐
│    Image     │ --> │  Core (Conv)    │ --> │  Readout   │ --> Neural Response
│    Input     │     │  (Kernels)      │     │  (Masks)   │
└──────────────┘     └─────────────────┘     └────────────┘
```

Key Components:

- **Core:** Convolutional kernels extract features
- **Readout:** Spatial masks pool features for each neuron

Model 1: Classical Klindt

Architecture

```
┌──────────────┐     ┌─────────────────┐     ┌─────────────┐
│    Image     │ --> │  4 Kernels      │ --> │  1 Shared   │ --> Output
│              │     │  (no polarity)  │     │   Mask      │
└──────────────┘     └─────────────────┘     └─────────────┘
```

Properties

- **Kernels:** 4 unconstrained
- **Masks:** 1 shared across all channels
- **Parameters:** Minimal

Limitation

Model 2: Per-Channel Masks

Architecture

```
┌─────────────┐     ┌─────────────────┐     ┌────────────────────┐
│    Image    │ --> │   4 Kernels     │ --> │  4 Spatial Masks   │ --> Output
│             │     │                 │     │  (1 per channel)   │
└─────────────┘     └─────────────────┘     └────────────────────┘
```

Properties

- **Kernels:** 4 unconstrained
- **Masks:** 4 (one per kernel channel)
- **Parameters:** 4x more masks

Limitation

Model 3: N-Masks (Surround Model)

Architecture

```
┌──────────────┐     ┌─────────────────┐     ┌──────────────────────┐
│    Image     │ --> │   4 Kernels     │ --> │   2 Spatial Masks    │ --> Output
│              │     │                 │     │  (Center + Surround) │
└──────────────┘     └─────────────────┘     └──────────────────────┘
```

Properties

- **Kernels:** 4 unconstrained
- **Masks:** N configurable (typically 2)
- **Each mask:** Pools from ALL channels

Use Case

Model 4: ON/OFF Model

Key Innovation: Explicit Polarity Constraints

```
                                                  ┌──────────────────────┐
                       ┌─────────────────────┐    │  ON Mask             │
                       │  ON Kernels (2)     │--> │  (pools ON only)     │──┐
                       │  [positive wts]     │    └──────────────────────┘  │
┌──────────────┐       │                     │                              │
│    Image     │  -->  │                     │                              ├--> Output
│              │       │  OFF Kernels (2)    │    ┌──────────────────────┐  │
└──────────────┘       │  [negative wts]     │--> │  OFF Mask            │──┘
                       └─────────────────────┘    │  (pools OFF only)    │
                                                  └──────────────────────┘
```

Model 4: ON/OFF Model (cont.)

Properties

- **ON Kernels:** Constrained positive weights (detect light increments)
- **OFF Kernels:** Constrained negative weights (detect light decrements)
- **ON Mask:** Pools exclusively from ON kernels
- **OFF Mask:** Pools exclusively from OFF kernels

Benefits

- Explicit polarity separation
- Biologically interpretable
- Clear visualization of cell type

Model 5: ON/OFF Mixed (Shared Kernels)

Architecture

```
                                             ┌─────────────────────┐
                    ┌────────────────────┐   │  ON Mask            │──┐
                    │  ON Kernels (2)    │-->│  (all ON kernels)   │  │
 ┌─────────────┐    │  [positive wts]    │   ├─────────────────────┤  │
 │   Image     │ -->│                    │-->│  Mixed Mask         │──┼--> Output
 │             │    │  OFF Kernels (2)   │   │  (1 ON + 1 OFF)     │  │
 └─────────────┘    │  [negative wts]    │-->├─────────────────────┤  │
                    └────────────────────┘   │  OFF Mask           │──┘
                                             │  (all OFF kernels)  │
                                             └─────────────────────┘
```

Model 5: ON/OFF Mixed (cont.)

Properties

- **Kernels:** 4 (2 ON + 2 OFF) - **SHARED**
- **Masks:** 3 (ON, OFF, Mixed)
- **Mixed mask:** Uses kernel 0 from ON + kernel 0 from OFF

Problem: Conflicting Optimization

```
ON Kernel 0 must satisfy:
  ├── ON mask: "be a good ON detector for pure ON cells"
  └── Mixed mask: "be a good ON component for ON-OFF cells"

These may require DIFFERENT spatial structures!
```

The Sharing Problem Visualized

Shared Architecture (Model 5)

```
Kernel 0 (ON) ─────┬────> ON Mask (all cells using ON)
                   │
                   └────> Mixed Mask (ON-OFF cells)

Conflicting gradients during training!
```

Result

- Kernels develop unexpected center-surround structures
- Neither ON nor Mixed pathway gets optimal features
- Compromised performance

Model 6: Dedicated ON/OFF Mixed

Solution: Dedicated Kernels per Pathway

```
                                                     ┌──────────────────────┐
                     ┌─────────────────────┐         │   ON Mask            │ ──┐
                     │   ON Kernels (0,1)  │ -->     │   (kernels 0,1 ONLY) │   │
                     │   [positive wts]    │         └──────────────────────┘   │
┌──────────────┐     ├─────────────────────┤         ┌──────────────────────┐   │
│    Image     │ --> │   OFF Kernels (2,3) │ -->     │   OFF Mask           │ ──┼--> Output
│              │     │   [negative wts]    │         │   (kernels 2,3 ONLY) │   │
└──────────────┘     ├─────────────────────┤         └──────────────────────┘   │
                     │   Mixed Kernels(4,5)│                                     │
                     │   [1 pos + 1 neg]   │ -->     ┌──────────────────────┐   │
                     └─────────────────────┘         │   Mixed Mask         │ ──┘
                                                     │   (kernels 4,5 ONLY) │
                                                     └──────────────────────┘
```

Model 6: Dedicated ON/OFF Mixed (cont.)

Kernel Allocation

Kernel Index	Polarity	Dedicated To
0, 1	Positive (ON)	ON Mask only
2, 3	Negative (OFF)	OFF Mask only
4	Positive (ON-like)	Mixed Mask only
5	Negative (OFF-like)	Mixed Mask only

Key Insight

- **No kernel serves multiple masters**
- Each pathway optimizes its own features

Architecture Comparison

Model	Kernels	Masks	Polarity	Kernel Sharing
Classical	4	1	None	N/A
Per-Channel	4	4	None	N/A
N-Masks	4	N	None	All shared
ON/OFF	4	2	Explicit	Pathway-dedicated
ON/OFF Mixed	4	3	Explicit	Mixed shares
Dedicated	**6**	**3**	**Explicit**	**None**

Parameter Comparison

Model 5 (Shared): 4 kernels

```
ON pathway:    kernels 0,1 → ON mask
OFF pathway:   kernels 2,3 → OFF mask
Mixed pathway: kernel 0,2  → Mixed mask (SHARED!)
```

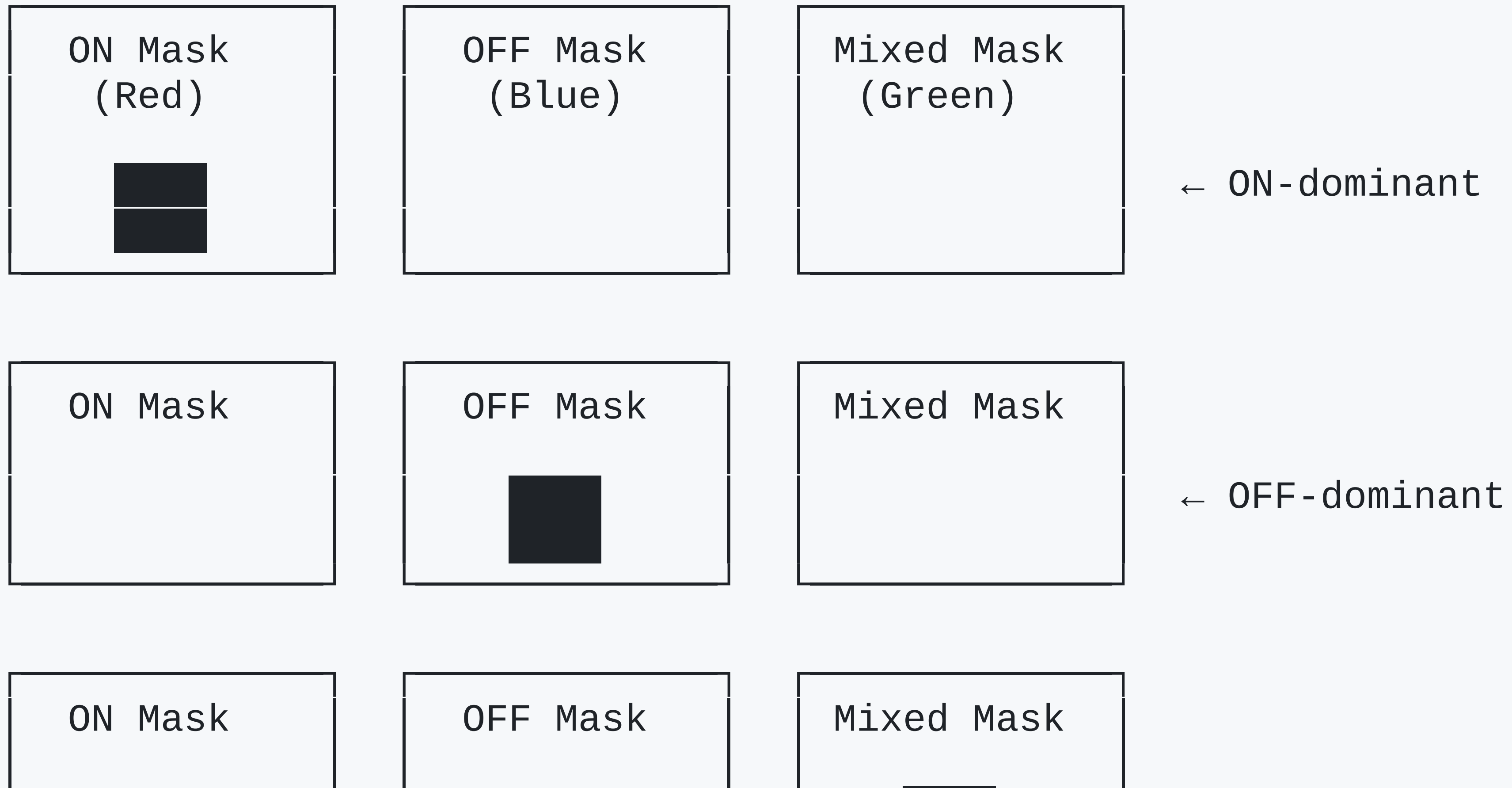
Model 6 (Dedicated): 6 kernels

```
ON pathway:    kernels 0,1 → ON mask (exclusive)
OFF pathway:   kernels 2,3 → OFF mask (exclusive)
Mixed pathway: kernels 4,5 → Mixed mask (exclusive)
```

Trade-off: +2 kernels for clean optimization

Visualization: Spatial Masks

Three Mask Types per Neuron



RGB Overlay Visualization

Combined View: R=ON, G=Mixed, B=OFF

```
┌──────────────────────────────────────┐
│                                      │
│        Pure Red = ON only            │
│        Pure Blue = OFF only          │
│        Pure Green = Mixed only       │
│        Yellow = ON + Mixed           │
│        Cyan = OFF + Mixed            │
│        White = All three             │
│                                      │
└──────────────────────────────────────┘
```

Shows spatial alignment of different pathways

Cell Type Classification

Based on Dominant Pathway

```
# Compute pathway strengths
on_strength = sum(ON_mask × ON_weights)
off_strength = sum(OFF_mask × OFF_weights)
mixed_strength = sum(Mixed_mask × Mixed_weights)

# Classify
if on_strength > threshold and others < threshold:
    cell_type = "ON-dominant"
elif off_strength > threshold and others < threshold:
    cell_type = "OFF-dominant"
elif mixed_strength > threshold and others < threshold:
    cell_type = "Mixed-dominant"
else:
    cell_type = "Multi-pathway"
```

Design Evolution Summary

```
Classical Klindt
      │
      ▼ "Need spatial separation"
Per-Channel Masks
      │
      ▼ "Too many parameters"
N-Masks (Surround)
      │
      ▼ "Need polarity constraints"
ON/OFF Model
      │
      ▼ "Can't model ON-OFF cells"
ON/OFF Mixed (Shared)
      │
      ▼ "Conflicting optimization"
Dedicated ON/OFF Mixed  ✓
```

Key Takeaways

1. **Biological Constraints Help**
   - Explicit ON/OFF polarity improves interpretability
   - Matches known retinal circuitry
2. **Dedicated Resources Prevent Conflicts**
   - Shared kernels create optimization pressure
   - Dedicated kernels allow clean specialization
3. **Trade-off: Parameters vs. Optimization**
   - 6 kernels > 4 kernels
   - But cleaner training dynamics
4. **Visualization Matters**

Future Directions

1. **Temporal Dynamics**
   - Add temporal kernels for motion sensitivity
2. **Hierarchical Models**
   - Stack multiple layers for complex features
3. **Biological Validation**
   - Compare learned kernels to measured RFs
   - Validate cell type classifications
4. **Different Cell Types**
   - Direction-selective cells
   - Contrast-invariant cells

Thank You

Code Repository Structure

```
center_surround/
├── models/
│   ├── klindt.py           # Classical model
│   └── klindtSurround.py   # All variants
├── training/
│   └── search.py           # Hyperparameter search
├── utils/
│   └── visualization.py    # Plotting functions
└── scripts/
    ├── klindtSurround/
    ├── klindtONOFF/
    ├── klindtONOFFMixed/
    └── klindtDedicatedONOFFMixed/
```

Appendix: Kernel Constraints

ON Kernels (Positive)

```
# During forward pass
conv.weight[:n_on].data = torch.abs(conv.weight[:n_on].data)
```

OFF Kernels (Negative)

```
# During forward pass
conv.weight[n_on:n_on+n_off].data = -torch.abs(conv.weight[n_on:n_on+n_off].data)
```

Mixed Kernels

```
# First half: positive, Second half: negative
conv.weight[mixed_start:mixed_start+n_mixed_on].data = torch.abs(...)
conv.weight[mixed_start+n_mixed_on:].data = -torch.abs(...)
```

Appendix: Hyperparameters

Tuned Parameters (Optuna)

- `smoothness_reg`: 1e-6 to 1e-1 (log scale)
- `weights_reg`: 1e-5 to 1e-1 (log scale)
- `mask_reg`: 1e-5 to 1e-1 (log scale)
- `learning_rate`: 1e-3 to 1e-1 (log scale)

Fixed Parameters

- `kernel_size`: 24×24
- `dropout_rate`: 0.2
- `batch_norm`: True

Appendix: Loss Function

```
# Poisson negative log-likelihood
loss = PoissonNLLLoss(predicted, target)

# Regularization
reg = (smoothness_reg × kernel_smoothness +
       mask_reg × L1(mask_weights) +
       weights_reg × L1(readout_weights))

# Total
total_loss = loss + reg
```

Evaluation Metric

- Pearson correlation between predicted and actual responses
- Averaged across all neurons