The Massive Function

The single formulation is given by:

$$\mathcal{L}(\boldsymbol{\psi}, \mathbf{v}) = \sum_{i=1}^{n} \frac{\int_{V_i} L(x,y,z)\,dV}{\frac{4}{3}\pi R(v_i)^3} - \lambda \cdot \sum_{i=1}^{n} R(v_i) - \begin{cases} \frac{\lambda}{R(5) - R\left(\sum_{i=1}^{n} v_i\right)} & \text{if } \sum_{i=1}^{n} v_i > 0.1 \\ 0 & \text{otherwise} \end{cases}$$

where radius of the VTA sphere is defined as:

$$R(v_i) = \begin{cases} \sqrt{\frac{v_i - 0.1}{0.22}}, & \text{if } v_i > 0.1 \\ 0, & \text{otherwise} \end{cases}$$

and the sphere's mask is defined voxels within the stimulation sphere's radius:

$$S(\psi_i, v_i) = \begin{cases} 1, & \text{if } (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \leq R(v_i)^2 \\ 0, & \text{otherwise} \end{cases}$$

and the integration of sphere values is empirically the dot product of the sphere and Nifti:

$$\int_{V_i} L(x,y,z)\,dV \approx \boldsymbol{S}(\boldsymbol{\psi}_i, v_i)^\top \boldsymbol{L}$$

The Gradient Is Found with the Partial Difference Quotient:
partial difference quotient for the gain with respect to $\mathbf{v}$ at index i is defined as:

$$\frac{\partial \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{\partial v_i} \approx \frac{\mathcal{L}(\boldsymbol{\psi}, \mathbf{v} + h\mathbf{e}_i) - \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{h}$$

Gradient Ascent with Adam Finds the Solution:

$$v_i^{(k+1)} = \text{clip}\left[v_i^{(k)} + \alpha \frac{\hat{m}_i^{(k)}}{\sqrt{\hat{v}_i^{(k)}} + \epsilon} \right]$$

where:

$$\hat{m}_i^{(k)} = \frac{\beta_1 m_i^{(k)} + (1 - \beta_1) \nabla_{v_i} \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{1 - \beta_1^{k+1}}, \quad \hat{v}_i^{(k)} = \frac{\beta_2 v_i^{(k)} + (1 - \beta_2) \left(\nabla_{v_i} \mathcal{L}(\boldsymbol{\psi}, \mathbf{v}) \right)^2}{1 - \beta_2^{k+1}}$$

—

Key Elements: Definitions

- $\boldsymbol{S}(\boldsymbol{\psi}_i, v_i)$: Sphere mask for the i-th contact.
- $\boldsymbol{L}$: Landscape values representing magnitudes across a grid of points.
- $r(v_i) = \max\left(0, \sqrt{\frac{v_i - 0.1}{0.22}}\right)$: Radius of the sphere for contact v_i, ensuring $v_i > 0.1$ to compute nonzero radii.
- λ: Regularization coefficient penalizing excessive stimulation.
- Gradient clipping:

  $$\text{clip}\,[v_i] = \max(0, \min(v_i, 5))$$

  ensures v_i values remain within the allowable range (0 to 5).
- $\beta_1, \beta_2, \alpha, \epsilon$: Hyperparameters for Adam optimization:

– β_1: Momentum decay rate.
– β_2: Variance decay rate.
– α: Learning rate.
– ϵ: Small constant to prevent division by zero.

—

Description

This function integrates:

1. Target Function: Summing contributions of high-value regions across sphere masks, normalized by sphere volume. This forces the contact to stimulate high values, but not expand across the brain.

2. Penalty Functions:

   - Per-contact penalty: Regularizes activation of individual contacts. This suppresses each contact, resulting in it shutting off unless it adds marginal benefit.
   - All-contacts penalty: Applies a blocking constraint to prevent total stimulation exceeding the safety limit of 5mA.

3. Optimization:

   - Gradient ascent with adaptive learning (Adam).
   - Gradient clipping to stabilize updates.

4. Stop Conditions: Convergence determined by gradient magnitude (L1 norm) and capped iterations.

—

Component-Based Breakdown of the Optimization Framework

Next, we'll go over each component in a more understandable approach. If you've already got it, don't read further. We are going to talk about the mathematical framework for optimizing stimulation parameters $\boldsymbol{\psi}$ and $\mathbf{v}$. We will expand each part of the loss function into its fundamental components and combine them into the final, simplified formulation.

—

Component 1: Radius Function $R(v_i)$

The radius $R(v_i)$ determines the size of the sphere affected by a contact's stimulation amplitude v_i:

$$R(v_i) = \begin{cases} \sqrt{\dfrac{v_i - 0.1}{0.22}}, & \text{if } v_i > 0.1 \\ 0, & \text{otherwise} \end{cases}$$

Purpose:

- Links stimulation amplitude to the physical size of the region of influence.
- Prevents stimulation for small or zero amplitudes ($v_i \leq 0.1$).

—

Component 2: Sphere Mask Function $S(\psi_i, v_i)$

The sphere mask $S(\psi_i, v_i)$ identifies the region of influence around each electrode at position $\psi_i = (x_0, y_0, z_0)$, compared to all voxels in the Nifti Landscape $\mathrm{L}(\psi_k) = (x_k, y_k, z_k)$

$$S(\psi_i, v_i; \psi_k) = \begin{cases} 1, & \text{if } (x_k - x_0)^2 + (y_k - y_0)^2 + (z_k - z_0)^2 \leq R(v_i)^2 \quad \forall k \in \{1, \ldots, K\} \\ 0, & \text{otherwise} \end{cases}$$

where:

K	is the total number of voxels in the Nifti landscape,
$\psi_k = (x_k, y_k, z_k)$	are the coordinates of the k-th voxel,
$\psi_i = (x_0, y_0, z_0)$	are the sphere center coordinates for electrode i, and
$R(v_i)$	is the radius of the sphere defined by the amplitude v_i.

Purpose:

- Specifies which points in space are within the sphere defined by $R(v_i)$.
- Encodes spatial information of each sphere's center ψ_i and radius $R(v_i)$.
- Promotes density of positive values, avoiding uninhibited expansion of radius.

—

Component 3: Target Function $T(\boldsymbol{\psi}, \mathbf{v})$

The target function measures the effectiveness of stimulation by integrating the empirical landscape $L(x, y, z)$ within the sphere:

$$T(\boldsymbol{\psi}, \mathbf{v}) = \sum_{i=1}^{n} \frac{\int_{V_i} L(x, y, z)\, dV}{\frac{4}{3}\pi R(v_i)^3} = \sum_{i=1}^{n} \frac{\boldsymbol{S}(\psi_i, v_i)^\top \boldsymbol{L}}{\frac{4}{3}\pi R(v_i)^3}$$

Purpose:

- Evaluates stimulation effectiveness by weighting the empirical Nifti landscape values $\boldsymbol{L}$ within the sphere.
- Normalizes by the volume of the sphere to ensure scale invariance.
- The integral is equivalent to a dot product between mask and values $\boldsymbol{S}^\top \boldsymbol{L}$.

—

Component 4: Penalty Terms P_1 and P_2

The penalties P_1 and P_2 ensure safe and efficient stimulation:

$$P_1 = \lambda \cdot \sum_{i=1}^{n} R(v_i)$$

$$P_2 = \begin{cases} \frac{\lambda}{R(5) - R\left(\sum_{i=1}^{n} v_i\right)}, & \text{if } \sum_{i=1}^{n} v_i > 0.1 \\ 0, & \text{otherwise} \end{cases}$$

Purpose:

- P_1: Penalizes large spheres to discourage excessive stimulation.
- P_2: Implements a blocking constraint to prevent unsafe total stimulation amplitudes ($\sum v_i > 5$).

—

Component 5: Loss Function $\mathcal{L}(\boldsymbol{\psi}, \mathbf{v})$

The loss function combines the target function T and penalties P_1, P_2:

$$\mathcal{L}(\boldsymbol{\psi}, \mathbf{v}) = T(\boldsymbol{\psi}, \mathbf{v}) - P_1(\mathbf{v}) - P_2(\mathbf{v})$$

Substituting T, P_1, P_2:

$$\mathcal{L}(\boldsymbol{\psi}, \mathbf{v}) = \sum_{i=1}^{n} \frac{\boldsymbol{S}(\psi_i, v_i)^\top \boldsymbol{L}}{\frac{4}{3}\pi R(v_i)^3} - \lambda \cdot \sum_{i=1}^{n} R(v_i) - \begin{cases} \frac{\lambda}{R(5) - R\left(\sum_{i=1}^{n} v_i\right)}, & \text{if } \sum_{i=1}^{n} v_i > 0.1 \\ 0, & \text{otherwise} \end{cases}$$

—

Component 6: Difference Quotient

The difference quotient approximates the partial derivative of $\mathcal{L}$ with respect to v_i:

$$\frac{\partial \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{\partial v_i} \approx \frac{\mathcal{L}(\boldsymbol{\psi}, \mathbf{v} + h) - \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{h}$$

Purpose:

- Approximates gradients numerically for optimization.
- Evaluates the change in $\mathcal{L}$ when a small perturbation h is applied to v_i.

—

Component 7: Gradient Ascent with ADAM Optimization

The optimization step for v_i using ADAM is:

$$v_i^{(k+1)} = \mathrm{v}_i^{(k)} + \alpha \frac{\hat{m}_i^{(k)}}{\sqrt{\hat{v}_i^{(k)}} + \epsilon}$$

Where:

$$\hat{m}_i^{(k)} = \frac{\beta_1 m_i^{(k)} + (1 - \beta_1) \nabla_{v_i} \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{1 - \beta_1^{k+1}}, \quad \hat{v}_i^{(k)} = \frac{\beta_2 v_i^{(k)} + (1 - \beta_2) \left(\nabla_{v_i} \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})\right)^2}{1 - \beta_2^{k+1}}$$

Purpose:

- Dynamically adjusts step sizes using moment estimates m_i, v_i.
- Prevents destabilizing oscillation, which our gradients are sensitive to.
- Prevents gradient explosion or vanishing.

—

Summary: The Simplified Loss Function

The simplified formulation is:

$$\mathcal{L}(\boldsymbol{\psi}, \mathbf{v}) = \sum_{i=1}^{n} \frac{\boldsymbol{S}(\psi_i, v_i)^\top \boldsymbol{L}}{\frac{4}{3}\pi R(v_i)^3} - \lambda \cdot \sum_{i=1}^{n} R(v_i) - \begin{cases} \frac{\lambda}{R(5) - R\left(\sum_{i=1}^{n} v_i\right)}, & \text{if } \sum_{i=1}^{n} v_i > 0.1 \\ 0, & \text{otherwise} \end{cases}$$

This equation integrates all components R, S, T, P_1, P_2 into a single mathematical representation.

Optimization Loop for Loss Function $\mathcal{L}(\boldsymbol{\psi}, \mathbf{v})$

The optimization process iteratively updates the stimulation amplitudes $\mathbf{v}$ using gradient ascent with ADAM optimization. The steps are as follows:

Step 1: Compute the Gradient Vector

The gradient of the loss function with respect to each amplitude v_i is computed using the partial difference quotient:

$$\frac{\partial \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{\partial v_i} \approx \frac{\mathcal{L}(\boldsymbol{\psi}, \mathbf{v} + h) - \mathcal{L}(\boldsymbol{\psi}, \mathbf{v})}{h}$$

Step 2: Apply Gradient Clipping

The gradient vector $\nabla_{\mathbf{v}} \mathcal{L}$ is clipped to prevent instability:

$$\nabla_{\text{clipped}} = \text{clip}\left(\nabla_{\mathbf{v}}, -\lambda_{\max}, \lambda_{\max}\right)$$

Step 3: Perform ADAM Optimization

ADAM updates the amplitudes based on moment estimates:

$$v_i^{(k+1)} = \mathrm{v}_i^{(k)} + \alpha \frac{\hat{m}_i^{(k)}}{\sqrt{\hat{v}_i^{(k)}} + \epsilon}$$

where:

$$\hat{m}_i^{(k)} = \frac{\beta_1 m_i^{(k)} + (1 - \beta_1)\nabla_{\text{clipped}}}{1 - \beta_1^{k+1}}, \quad \hat{v}_i^{(k)} = \frac{\beta_2 v_i^{(k)} + (1 - \beta_2)(\nabla_{\text{clipped}})^2}{1 - \beta_2^{k+1}}$$

Step 4: Enforce Constraints on v

No negative amperages allowed, so apply constraints:

$$v_i \geq 0$$

Step 5: Check Stopping Conditions

Stop the optimization if:

- The gradient's L1 norm is below a threshold:

$$\|\nabla_{\mathbf{v}} \mathcal{L}\|_1 < \tau$$

- The maximum number of iterations is reached (iterations = 50).

Overall Loop

Repeat the following steps until convergence or stopping criteria are met:

$$\textbf{Loop:} \begin{cases} \text{1. Compute Gradient} \\ \text{2. Apply Gradient Clipping} \\ \text{3. Perform ADAM Update} \\ \text{4. Enforce Constraints} \\ \text{5. Check Stopping Conditions} \end{cases}$$

Step 6: Final Step

Once the loop has concluded, return the optimal amperages at each contact:

$$\mathbf{v}_{\text{optimal}} = \{v_1, v_2, \ldots, v_n\}$$

And that's all there is to it.
– Calvin