1 Model

$$x_{n,t}^k = f_{\text{Update}}(f_{\text{Agg}}^C(\mathbf{N}_{n,t}^{k-1}), f_{\text{Agg}}^S(\mathbf{S}_{n,t}^{k-1}), f_{\text{Agg}}^T(x_{n,t-1}^{k-1}, \cdots, x_{n,t-d}^{k-1}), x_{n,t}^{k-1}) \quad (1)$$

Where

f_{Agg}^C is the county-timestep-layer aggregation function.
f_{Agg}^S is the subreddit-layer aggregation function.
f_{Agg}^T is the temporal aggregation function.

The aggregation functions operate on the preceeding iterations messages, from the $(k-1)^{\text{th}}$ step of updates. The temporal aspect of the data is encoded in the parameters to the temporal temporal aggregation function. In this formaulation, there is a single, large graph with a node for each county, at each timestep. The temporal parameters represent the connections between different temporal realizations of a given county. In a sense, each node has a directional connection to d nodes 'in the past'.

1.1 f_{Agg}^C

The parameters to this function represent the connections for a single county node n at a single timestep t. The set $\mathbf{N}_{n,t}^{k-1}$ contains the messages from the neighborhood of the county node n at the same timestep, from the preceeding round of updates.
A draft approach would be to use **symmetric normalization**, and sum the set of messages, each weighted by the factor $\frac{1}{\sqrt{\mathcal{N}_u * \mathcal{N}_v}}$

$$f_{\text{Agg}}^C(\mathbf{N}_{n,t}^{k-1}) = \sum_v^{\mathbf{N}} \left(\frac{x_{v,t}^{k-1}}{\sqrt{\mathcal{N}_v * \mathcal{N}_n}} \right)$$

This could be improved by the use of 'deep' methods, where a small parameterized function or network could be used to aggregate the messages.

1.2 f_{Agg}^S

The parameters to this function represent the connections between a single county-node n at a single timestep t and the set of 'static' subreddit nodes. The term static is used to refer to the fact that there is only a single node for a given subreddit, not a node for each timestep.
The set $\mathbf{S}_{n,t}^{k-1}$ represents the set of messages from all subreddits connected to the county at that timestep, from the preceeding iteration of the update function.
The message sent by a subreddit is simply a parameter holding its current 'embedding' in some low dimensional space, weighted by the edge weight (actual count value from the counties reddit activity for that timestep). For a subreddit s, its message to node n will be a function:

$$\mathbf{m}_{n,t,s}^k = x_s^{k-1} * g(e_{n,t,s})$$

Where $x_s^{k-1} \in \mathcal{R}^i$ is the state of the subreddit in the preceeding iteration of message passing, and i is the chosen dimension of the embedding space for subreddits. The function g weights this message by the value of the edge between the subreddit and the county node at that timestep, $e_{n,t,s}$
To aggregate these, the **Set Pooling** method would be used, where two MLPs are trained, with one operating directly on subreddit state values, and the second oeprating on the sum of these.

$$f_{\text{Agg}}^S(\mathbf{S}_{n,t}^{k-1}) = \text{MLP}_\theta \left(\sum_s^S \text{MLP}_\phi \left(\mathbf{m}_{n,t,s}^{k-1} \right) \right)$$

The weight values for the ϕ MLP would be shared across all subreddits.

1.3 $f_{\textbf{Agg}}^T$

This function encodes the temporal history of a specific node, at a specific timestep. The sequence $x_{n,t-1}^{k-1}, \cdots, x_{n,t-d}^{k-1}$ represents the d nodes associated with county n, at the d preceeding timesteps.
To handle the sequential nature of this data, the a draft aggregation function could be an LSTM model.

$$f_{\text{Agg}}^T(x_{n,t-1}^{k-1}, \cdots, x_{n,t-d}^{k-1}) = \text{LSTM}(x_{n,t-1}^{k-1}, \cdots, x_{n,t-d}^{k-1})$$