

# The Sprawlter Graph Readability Metric: Combining Sprawl and Area-aware Clutter

Zipeng Liu, Takayuki Itoh, Jessica Q. Dawson, and Tamara Munzner *Senior Member, IEEE*

**Abstract**—Graph drawing readability metrics are routinely used to assess and create node-link layouts of network data. Existing readability metrics fall short in three ways. The many count-based metrics such as edge-edge or node-edge crossings simply provide integer counts, missing the opportunity to quantify the amount of overlap between items, which may vary in size, at a more fine-grained level. Current metrics focus solely on single-level topological structure, ignoring the possibility of multi-level structure such as large and thus highly salient metanodes. Most current metrics focus on the measurement of clutter in the form of crossings and overlaps, and do not take into account the trade-off between the clutter and the information sparsity of the drawing, which we refer to as sprawl. We propose an area-aware approach to clutter metrics that tracks the extent of geometric overlaps between node-node, node-edge, and edge-edge pairs in detail. It handles variable-size nodes and explicitly treats metanodes and leaf nodes uniformly. We call the combination of a sprawl metric and an area-aware clutter metric a *sprawlter* metric. We present an instantiation of the sprawlter metrics featuring a formal and thorough discussion of the crucial component, the penalty mapping function. We implement and validate our proposed metrics with extensive computational analysis of graph layouts, considering four layout algorithms and 56 layouts encompassing both real-world data and synthetic examples illustrating specific configurations of interest.

**Index Terms**—Graph drawing, graph drawing metrics, readability metrics, aesthetic criteria

## 1 INTRODUCTION

COMPUTING quality metrics is an important and popular quantitative approach to evaluate and generate visualizations [1], especially for node-link diagrams, an intuitive and effective visual representation for relationships between entities in a graph. Graph layout has been studied for many years, with major effort devoted to automatic generation of readable and faithful node-link graph layouts [2], [3]. One of the important aspects in making good layouts is to have good metrics to measure the readability or faithfulness, which are used by graph layout algorithms either explicitly or implicitly. In particular, we focus on readability metrics, also historically known as aesthetic criteria. Many readability metrics have been proposed in previous papers, such as edge-edge crossings, node-edge crossings, edge bends, and angular resolution. Many empirical studies have been conducted to evaluate their correspondence with human judgements [4], [5]. We note three problems with many previous readability metrics.

First, they report simple **integer counts** of detrimental events such as node-edge crossings. Although counts are straightforward to understand and to compute, they lack the precision to capture how badly an event hinders readability, such as how much an edge crosses a node as measured in length, or whether two edges cross with a glancing angle.

Second, they only compute on **single-level** structure, even in a multi-level graph layout, which are used in many domains [6], especially for large graphs. A multi-level graph combines a cluster hierarchy with a base graph, where

the original graph nodes comprise the lowest leaf level of the hierarchy, and metanodes (also known as clusters or subgraphs) at higher levels contain lower-level nodes. The metanodes are typically more salient than the leaf nodes because they are bigger in size, but this important multi-level structural information is not taken into account when measuring readability by the previous metrics.

Third, they are mainly **clutter-only** metrics that only penalize crossings and do not fully assess the layout quality. In addition to clutter, information density [7] (also known as data-link ratio [8] or space efficiency [9]) also matters.

We use the term geometric sprawl, or **sprawl** for short, to capture disparity between the size of small features and a potentially large canvas on which they are spread out. This measure is different from topological sparseness: sprawl considers geometric layout information such as coordinate positions and area. There is an important trade-off between clutter and sprawl: it is possible to eliminate all overlaps (especially for node-node and node-edge overlap) by simply either pushing the nodes away from each other or shrinking the node size until they are completely separated, but at the cost of tiny nodes and edges spread out on a huge canvas. The quantification of sprawl itself is not new; it is the inverse (reciprocal) of metrics recently called compactness by Kieffer et al. [10] and visualization coverage by Dunne et al. [11]. The idea of minimizing the metric of total area is longstanding [12]. The novelty of our work lies in the direct capture of this readability trade-off, and our emphasis that it constitutes the most central property of graph legibility.

We illustrate the three problems above in Fig. 1. All three example layouts show two-level synthetic graphs containing four colored metanodes drawn as minimum bounding circles of their constituent leaf nodes. The count does not reflect the fact that layout (b) is more cluttered than (a) in

• Z. Liu, T. Munzner, and J. Q. Dawson are with the University of British Columbia. E-mail: {zipeng | tmm | jqdawson}@cs.ubc.ca.

• T. Itoh is with Ochanomizu University. E-mail: itot@is.ocha.ac.jp.

Manuscript received October 4 2019; revised December 20, 2019.

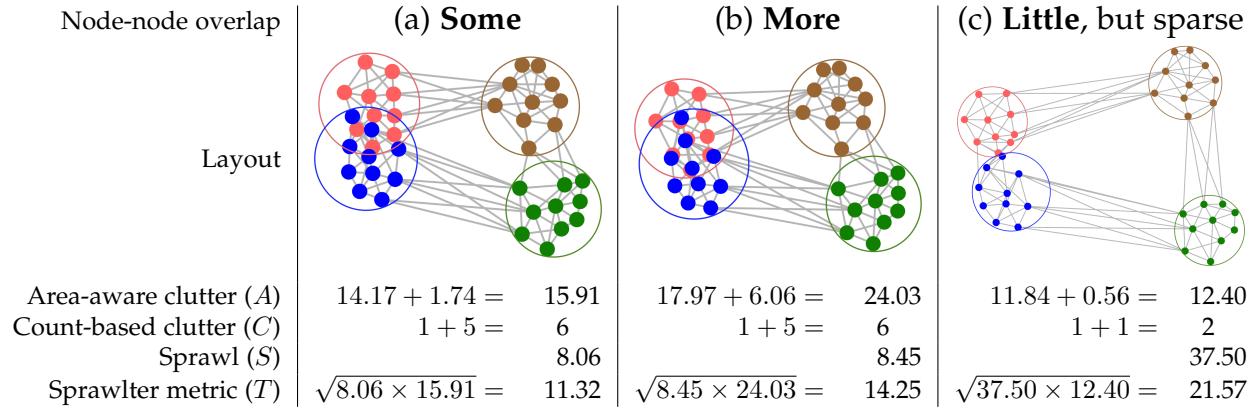


Fig. 1. Comparison of approaches to measuring node-node crossing in three synthetic 2-level graph layouts. (a) Some overlap. (b) More overlap. (c) More sprawl. Our sprawlter metric ( $T$ ) is the geometric mean of the sprawl ( $S$ ) and area-aware clutter metrics ( $A$ ). Clutter metrics ( $A$  and  $C$ ) are broken down into metanode vs. leaf node pairs. For all metrics, larger numbers are worse, indicating less readability. The increased overlap between leaf nodes and the metanode pair is captured by the area-aware metric ( $24.03 > 15.91$ ). The increased sprawl when nodes are shrunk in size and pushed away from each other to avoid overlap, resulting in a more sparse layout, is captured by the sprawl ( $37.50 > 8.45$ ). Their mean ( $T$ ) assesses the trade-off between clutter and sprawl in the geometric configurations.

terms of both leaf node overlaps and metanode overlaps, as there are exactly six crossings in both layouts (the integer count problem). Nodes in layout (c) are shrunk and spread out on a larger canvas resulting in fewer overlaps, but the layout suffers from poor space efficiency [9] due to sprawl.

To address the first two problems, we contribute an area-aware *approach* for three *families* of clutter metrics: node-node (NN), node-edge (NE), and edge-edge (EE) crossings. It measures the amount of geometric overlaps (NN, NE) or crossing angle (EE) in both single-level and multi-level graph layouts, then computes a penalty for each geometric measurement, and finally sums the individual penalties as a NN, NE, or EE clutter metric. We identify general requirements and make specific choices for the crucial step of mapping geometric measurements to penalties. To address the third problem, we contribute the **sprawlter metric** to capture the trade-off between sprawl and clutter. It computes the geometric mean of sprawl and area-aware clutter for each of the NN, NE, and EE families. We implement these new metrics, and validate their benefits by comparing quantitative computational results of our approaches to traditional count-based approaches and recent proposals [11], with respect to qualitative assessments of layout pictures.

## 2 RELATED WORK

We discuss related work on readability metrics and the evaluation of these metrics.

### 2.1 Graph drawing readability metrics

We identify two categories of graph drawing metrics: single-purpose and compound metrics. The former only quantify one single feature of the layout, while the latter explicitly combine or implicitly reflect multiple single-purpose ones.

#### 2.1.1 Single-purpose metrics

The single-purpose metrics most relevant to our own work focus on how clearly the drawn elements of a graph layout can be seen. There are many other types of metrics that are less relevant, including those related to graph topology

such as average degree of nodes or modularity, those related to faithfulness such as shape-based metrics [13], and those measuring specific aspects of perception and cognition such as symmetry [14], [15], edge bends [5], and edge continuity [16]. A recent paper uses many metrics to build a neural network to assess quality of graph layout, but does not focus on readability metrics [17].

Well over a dozen single-purpose readability metrics have been proposed, with some evaluated by controlled experiments with human subjects. Early work from Purchase formally documented several metrics [18], with many subsequent experiments that compare these and other metrics to human judgements [4], [15], [16]; Dunne and Shneiderman provide a survey of empirical studies on readability metrics [19]. We categorize these metrics by the three problems we note above: clutter, multi-level, and sprawl.

**Clutter metrics.** Clutter is an important and obvious readability obstacle. The well known clutter-related metrics are edge-edge, node-edge, and node-node crossings, which report discrete integer numbers of the crossing events. Some metrics report continuous real numbers, such as angular resolution at nodes [18], which measures the minimum angle of edges that are incident to an individual node, and angular resolution at edges, which measures the angles between crossing edges, or the total resolution of Argyriou et al. [20] that combines them.

The impact of crossing angles, namely the angle at which two edges cross, has been heavily studied. Although Ware et al. did not find significant relevance between human performance and average crossing angles in their multi-factor experiment using global graph layouts as stimuli [16], a later single-factor experiment using simple synthetic drawings conducted by Huang et al. showed that the crossing angles have a significant impact on response time [21]. Dunne et al. proposed another version of edge crossing angle metric, defined as the average deviation of the crossing angles from an ideal angle ( $70^\circ$ ) [11]. Their metric is 0 for the ideal crossing angle, which ignores the fact that there is still a detrimental crossing event that reduces readability (R2 in section 4). Our own edge-edge metric differs with an explicit

function that maps angles into penalties instead of using them directly, allowing us to clearly distinguish even ideal crossing angles from the non-crossing case.

**Multi-level metrics.** Some metrics measure geometric separability, namely how compactly leaf nodes lie within or are spread between metanodes, including edge length variability [22], [23], which is also applicable to single-level layouts. However, they do not recognize salience of clutter in a multi-level structure. The many topological separability metrics, such as modularity, conductance, coverage, performance, and so on [24], [25] do not measure geometric configurations. The only true geometric metrics in the literature that address the multi-level problem are *ambiguity measures* for community structure from Wang et al. [26] and the *group overlap* metric from Dunne et al. [11]. Wang’s approach focuses on the ambiguity caused by overlap of metanodes rather than readability, where they considered the geometric distribution of leaf nodes within metanodes rather than salience of clutter. It is later used in their follow-up papers to assess readability of clusters [27], [28]. Dunne’s proposal for group overlap does compare overlaps between a metanode and leaf nodes, but not between metanodes, which are more visually salient. Our approach features a uniform treatment of nodes and metanodes to capture the clutter in both single-level and multi-level graph layouts.

**Sprawl metrics.** Several simple sprawl-related metrics have been proposed, including total drawing area, the maximum edge length, sum of edge lengths, and the aspect ratio of the drawing canvas [12]. The multiplicative inverse of sprawl, the metric that we use, has previously been called compactness by Kieffer et al. [10] and visualization coverage by Dunne et al. [11]. We use the inverse quantity to emphasize the problematic nature of sparsity as analogous to clutter. Kieffer et al. found evidence of its importance in small user-generated graph layouts [10], and Dunne et al. proposed it as a stand-alone separate metric for NN. The novelty of our work lies in exploring the trade-off between sprawl and clutter, not introducing the sprawl metric itself.

### 2.1.2 Compound metrics

Individual single-purpose metrics can be combined into a compound metric. Such compound metrics are often implicitly incorporated into automatic layout algorithms, as with physics models that measure a graph layout with an artificial physics quantity, such as force (GEM [29]), energy (LinLog [22]), or stress (NEATO [30]), as if the nodes and edges are real-world physical objects. For example, force models typically encourage uniform edge lengths and clustering around high-degree nodes, and discourage node-node overlaps. Some proposals have been made to add other types of forces to explicitly incorporate more single-purpose metrics [31], [32]. A compound metric explicitly proposed in previous work is a weighted sum of multiple single-purpose metrics [33], but justifying the weights remains an open problem. Our sprawlter metric, where sprawl and clutter is combined, is also an example of an explicit compound metric. We emphasize the specific trade-off between clutter and sprawl as directly opposite quantities, rather than attempting to combine very disparate quantities of all possible metrics into a single compound quantity.

## 2.2 Evaluation of graph drawing metrics

Previously proposed metrics have been evaluated either with human subjects or mathematical models, with the goal of obtaining a deeper understanding of the nuances and ranking the metrics by perceptual or cognitive impact. One approach is to gather quantitative performance data or qualitative feedback on the stimuli graph layouts from human observers, where the stimuli can be generated by automatic computational algorithms or human manual labour. There are many papers from earlier years that followed this general approach. Typical examples are from Purchase et al. [5], [14], [34], [35], Ware et al. [16], Huang et al. [21], [36], Kobourov et al. [37]. Another approach is to gather user-generated layouts in order to understand their mental model of graph layouts. Van Ham and Rogowitz asked participants to position the nodes in a graph layout and investigated which metrics are useful for explaining their layouts [23]. Following this thread, Dwyer et al. compared user-generated layouts with automatic layouts [15]. A third approach focuses on quantitative models of layouts and relevant tasks. McGuffin et al. evaluated space efficiency of visual representations of trees mathematically and provided design guidelines accordingly [9]. Dawson et al. proposed and evaluated a predictive model of human behaviour tracing paths through a graph [4]. None of these evaluations provide direct guidance on how to address the three problems that we tackle in this work.

## 3 SPRAWLTER METRICS

We describe the computation of the sprawlter metrics in this section including the general idea, technical details, and how they address the three problems presented in the introduction. First, for each clutter event, the amount of overlap (area of node-node overlap, length of node-edge overlaps, and angle of edge-edge crossing) is measured. Then, the overlap measurement is transitioned into a penalty – an indication of the degree of clutter – with a penalty mapping function. We elaborate further on these penalty mapping functions in section 4. Finally, these penalties for each individual clutter event are summed up for node-node overlap (NN), node-edge overlap (NE), and edge-edge crossing (EE) respectively as the area-aware metrics. Meanwhile, sprawl is also computed based on geometric properties of the entire layout. The sprawlter metric is a geometric mean combining the sprawl and area-aware metric.

### 3.1 Graph layout definitions

We define a graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. The graph  $G$  can be either a single-level graph or a multi-level graph; in the latter case, we simply expand the definition of the nodes  $v \in V$  to be either leaf nodes or metanodes. Since we treat both of these cases in a completely uniform way, our algorithm does not need to further consider the hierarchy  $H$  that is part of a compound graph, except for a low-level implementation detail (when avoiding computation of overlaps between a metanode and its descendants). We do not consider metaedges, which are analogous to metanodes, but are not commonly used in current layout algorithms.

We denote the geometric measurements by functions of graph elements: the area of a node  $area(v)$ , overlapping area of two nodes  $area(v_1, v_2)$ , and similarly, length of an edge  $length(e)$ , diameter of a node  $length(v)$ , overlapping length of a node-edge pair  $length(v, e)$ . The crossing acute angle (in radians) of two edges is  $angle(e_1, e_2)$ , and its complement is  $c\_angle(e_1, e_2) = \pi/2 - angle(e_1, e_2)$ . When they just begin to touch, we consider them to intersect but with zero overlap:  $area(v_1, v_2) = 0$ , or  $length(v, e) = 0$ . When they do not intersect, the measurement function is undefined – it is **not** zero.

### 3.2 Measuring geometric overlap

We measure the overlapping area of a node pair, overlapping length of a node-edge pair, and (the complement of) crossing angle of an edge pair if the nodes or edges intersect with each other. Fig. 2 shows an increasing amount of overlap between pairs from left to right, from none, to near-minimum, some, and near-maximum overlap. Obviously, the degree of clutter corresponds to the amount of overlap. In the area case, the degree of clutter corresponds positively to  $area(v_1, v_2)$  or  $length(v, e)$ . We consider length to be the degenerate version of area, and below often use  $area$  to mean both quantities. In the angle case, the degree of clutter corresponds positively to  $c\_angle(e_1, e_2)$ , where a perpendicular angle of 90 degrees is the least cluttered and a glancing angle of 0 degree – where one line segment is on top of the other – is the most cluttered. We thus consider the complement of the crossing angle rather than the crossing angle itself. In Fig. 2, the count-based metrics are only able to distinguish between the leftmost column with no overlap and the other three columns, but not between those three.

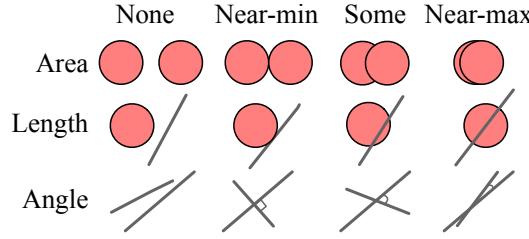


Fig. 2. Increasing amount of overlap between node-node, node-edge, and edge-edge pairs. Count-based metrics can only distinguish no overlap from the other three cases, but not between them.

### 3.3 Mapping measurements to penalties

We map the measured amount of overlap to penalties with a penalty mapping function in order to match the measurement to the degree of perceived clutter. Generally, we denote the penalty mapping functions by  $f(x)$ , where  $x$  is the overlapping measurement. Specifically, for the three different families, we use  $f^{NN}(x)$ ,  $f^{NE}(x)$ , and  $f^{EE}(x)$ . Note that for the EE family,  $x$  denotes the complementary angle so that the valence of  $f^{EE}(x)$  matches the others (since an original angle of 0 is the worst case).

We avoid the trivial instantiation of  $f(x)$  where geometric measurements are used directly as penalties (i.e.  $f(x) = x$ ), because the measured area / length / angle

---

### Algorithm 1: Computation of node-node area-aware metrics.

---

```

Input :  $G = (V, E)$ 
Output: total penalty  $A^{NN}(G)$  and count  $C^{NN}(G)$ 

1 totalPenalty  $\leftarrow 0$ 
2 count  $\leftarrow 0$ 
3 for  $v_1 \in V$  do
4   for  $v_2 \in V$  do
5     if  $v_1 \neq v_2 \&\& !IsAncOrDesc(v_1, v_2)$  then
6       if CheckIntersection( $v_1, v_2$ ) then
7          $x \leftarrow ComputeOverlapArea(v_1, v_2)$ 
8         penalty  $\leftarrow PenaltyFunc(x, v_1, v_2)$ 
9         totalPenalty  $\leftarrow totalPenalty + penalty$ 
10        count  $\leftarrow count + 1$ 
11 return totalPenalty, count

```

---

does not in general equal the degree of visual clutter. In particular, the distinction between the no overlap and touching cases is crucial (first and second columns in Fig. 2), but  $f(0) = 0$  would fail. The penalty mapping function is discussed further in Sect. 4.

### 3.4 Summing penalties

We directly sum up penalties of all overlapping pairs of node-node, node-edge, and edge-edge in a graph layout respectively, and refer to the three resulting total penalties as area-aware metrics, denoted by  $A^{NN}$ ,  $A^{NE}$ , and  $A^{EE}$ , while the count-based metrics are denoted by  $C^{NN}$ ,  $C^{NE}$ , and  $C^{EE}$ .

These area-aware metrics address the integer-count problem with a more precise indication of the degree of clutter. They also address the single-level problem for NN and NE, because the area awareness automatically captures the hierarchical information in multi-level graph layouts. Metanodes increase in area and diameter at ascending levels of the hierarchy: higher-level metanodes are larger than their constituent metanodes at lower levels, and than their leaf nodes at the lowest level. Hence, summing area or length penalties is analogous to a weighted sum of binary counts, where larger overlaps are weighted more heavily than smaller ones.

We show the pseudocode to compute the node-node area-aware metric in Alg. 1; the similar analogs for NE and EE are in Sect. S2 of supplemental materials. The condition statement in line 1 checks for validity of the node pair, where the function  $IsAncOrDesc(v_1, v_2)$  checks if one node is the ancestor / descendant of the other in the node hierarchy. Function  $CheckIntersection(v_1, v_2)$  returns true if  $v_1$  and  $v_2$  intersect each other.

### 3.5 Combining clutter metrics with sprawl

While clutter indicates overlaps and occlusions of nodes and edges, sprawl indicates geometric sparsity of graph elements on a drawing canvas, as illustrated in Fig. 1. The relevant factors contributing to sprawl include total drawing area (typically the axis-aligned minimum bounding box of

the entire layout), node areas, edge lengths, and the number of nodes and edges. Spreading out nodes of a fixed size on a larger canvas would increase the sprawl, for example.

We quantify sprawl by dividing the total drawing area of a graph layout  $G$  by the area occupied by all nodes (excluding overlaps). It is the reciprocal of the compactness from Kieffer et al. [10] or the visualization coverage from Dunne et al. [11]. Formally:

$$S(G) = \text{area}(G)/\text{area}(\bigcup_{v \in V} v) \quad (1)$$

$S(G)$  denotes the sprawl, and  $\text{area}(G)$  the total drawing area of the graph layout  $G$ . The area of union of all nodes captures the intuition behind sprawl well by accounting for variation in node sizes. We also considered other alternatives. If we use the number of nodes,  $|V|$ , as the denominator, it would not account for absolute node sizes: for example, the sprawl of layout (b) and layout (c) in Fig. 1 would be the same. If we use the smallest node area,  $\min\{\text{area}(v) | \forall v \in V\}$ , the resulting ratio would not gracefully handle relative node size variation: for example, a graph with 1000 large nodes and one tiny node would be over-penalized.

There is a direct trade-off between clutter and sprawl: for a fixed node size, the more spread-out the nodes are, the less clutter there would be and the more sprawl there would be. However, the interplay between clutter and sprawl is not strictly a zero-sum game: it is very possible to keep clutter low while having perceivable node sizes (or edge lengths) and reasonable total drawing area. It is common to combine multiple values by computing a mean of them. In mathematics, there exists many different versions of means [38] for different purposes. We choose the geometric mean (i.e. square root of the product of two positive values) because it is capable of normalizing different ranges of the constituent values: e.g. a 10% change in sprawl has the same effect as a 10% change in clutter. It is more suitable than the commonly-used alternatives of the harmonic mean, arithmetic mean, and quadratic mean. The harmonic mean is usually dominated by the minimum of the two values; a typical usage is to combine precision and recall for machine learning models. The arithmetic mean requires that the values have identical range to avoid unfair averaging. The quadratic mean is often used to measure errors between estimations and ground truths.

The sprawlter metric (denoted by  $T$ ) for NN is stated as follows, and those of NE and EE have the same form.

$$T^{NN}(G) = \sqrt{S(G) \cdot \max\{A^{NN}(G), 1\}} \quad (2)$$

We impose a lower bound for the area-aware component: when the area-aware metric, or generally, the clutter measurement, is very small, the sprawlter metric should retain the sprawl-aware information; otherwise sprawl has little influence on the sprawlter metric in such situations. In contrast, we do not need to impose any bound for the sprawl as it is always greater than or equal to 1.

We note that the general idea of sprawlter is completely independent of area-awareness: an instantiation could be designed for purely count-based metrics.

## 4 PENALTY MAPPING FUNCTION

The previous section provides an overview of the sprawlter metrics computation, which has the penalty mapping function at its core. We now discuss this crucial and challenging component in detail.

The penalty mapping function,  $f(x)$ , takes a geometric measurement of a single overlap event (area or length or complementary angle), and emits a number to indicate the degree of clutter for this event. In short, it is a mapping between overlap and clutter.

### 4.1 General requirements

We identify four important general requirements for the penalty mapping functions, illustrated in Fig. 3.

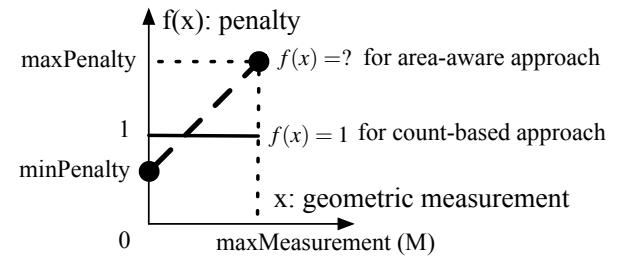


Fig. 3. General requirements for area-aware penalty mapping function, vs. a constant function  $f(x) = 1$  for the count-based approach.

- R1) **Increasing penalty.**  $\forall x_1 < x_2, f(x_1) < f(x_2)$ . The penalty should increase as the overlap increases. In Fig. 3,  $f(x)$  for the area-aware approach increases from a minimum to a maximum penalty, as opposed to a flat line for count-based approach. This requirement is necessary to distinguish between minimum, some, and maximum degree of clutter.
- R2) **Substantial touching penalty.**  $f(0) >> 0$ . In order to clearly distinguish the no-overlap case from a very small overlap, we require a substantial minimum penalty that is incurred as soon as two elements touch. That is,  $f(0)$  should be substantially greater than zero.
- R3) **Count calibration.**  $\text{minPenalty} < 1 < \text{maxPenalty}$ . The sprawlter metrics are intended to be a drop-in replacement to count-based metrics that provide richer information, and thus the penalties need to bracket, and be directly comparable to, the count-based value of 1. The touching case creates less clutter than what is captured by a simple count; the full overlap case demands a higher penalty.
- R4) **Local function.**  $f_{v_1, v_2}^{NN}(x)$  instead of  $f_G^{NN}(x)$ . We need to have a function that is calculated separately for each pair of nodes that can incorporate local knowledge of their individual sizes, rather than an identical global function that applies to all node pairs independently of their size. For the EE family, since all crossing angles share a universal unit (radians) and are bounded within the range  $[0, \pi/2]$ , a local function is equivalent to a global one. In the

notation, we use subscripts to indicate whether the function is local or global.

In order to calibrate penalties to counts (R3), it is necessary to normalize the absolute node areas (and edge lengths) to relative areas. We divide the area of each node by the smallest node area in the layout such that the normalized area of the smallest nodes is 1. If all nodes have the same size, the normalized area of every node is 1; if node sizes vary, the normalized areas of larger nodes are greater than 1. Similarly, for lengths, we divide them by the diameter of the smallest node. There is no need to normalize angles as they are all in radians. We use normalized areas and lengths below, unless otherwise specified.

## 4.2 Specific choices

We make specific choices to fulfill the general requirements of the penalty mapping function, while there exist other alternatives. Our goal is to construct a specific function that is simple and easy to understand. Fig. 4 shows the NN penalty mapping function for the smallest node pair.

C1) We choose a **power function**:  $f(x)$  is linear to  $x^\gamma$ , where  $\gamma$  denotes the exponent in the power function and dictates the curve shape of  $f(x)$ . In Fig. 4, we show the curves for three different  $\gamma$  values. We chose  $\gamma$  based on previous empirical work [21], [39]. The longstanding psychophysical power law proposed by Stevens [39], which is a mapping between physical stimulus and perceived sensation, inspired our choice of  $\gamma = 0.7$  for node-node overlap areas and  $\gamma = 1.0$  for node-edge overlap lengths. More recently, Huang et al. [21] suggest that human performance is correlated to the crossing angle with a quadratic term in their preliminary user evaluation, so we use a quadratic function ( $\gamma = 2.0$ ) for edge-edge crossing angles. To check the impact of this choice, we conducted theoretical and computational analysis comparing a linear EE penalty mapping function and a quadratic one. We found that difference between the two is very small except when there are many glancing crossing angles. The details of this comparison are in Supp. Sect. 1.2.

C2) We choose a **minimum penalty** proportional to the node sizes, specifically a fraction of the maximum possible clutter measurement  $M$  that would be incurred if they completely overlapped:  $f(0) = \alpha M^\gamma$ , where  $\alpha$  denotes the fraction, and  $0 < \alpha < 1$ . For example, for the smallest node pair  $v_1, v_2$ , the maximum overlap  $M_{v_1, v_2} = 1$ , and thus  $0 < f_{v_1, v_2}^{NN}(0) = \alpha < 1$ , which is compatible with the requirements of a substantial touching penalty (R2) and calibration to count (R3). In a layout with variable node sizes,  $M$  would be different for node pairs of different sizes, resulting in a larger touching penalty for large pairs and a smaller touching penalty for small pairs, so the local function requirement is also met (R4). The touching penalty could be greater than one for sufficiently large nodes. Similarly, the same amount of overlap area or length ( $x$ ) will have a bigger

impact ( $f(x)$ ) if it happens on bigger nodes than smaller nodes.

C3) We choose a **maximum penalty** proportional to the node sizes, with  $M$  as above:  $f(M) = \beta M^\gamma$ , where  $\beta$  denotes the proportion, and  $\beta > 1$ . For the same example as above where  $M = 1$ , R3 and R4 compatibility also holds, and  $f(M) = \beta > 1$ . Meanwhile, as  $\beta > 1 > \alpha$ , the maximum penalty is guaranteed to be greater than the minimum penalty; combined with the choice of a power function (C1),  $f(x)$  is an increasing penalty (R1).

C4) We choose to set the **halfway point** of measured clutter for the smallest pair to a penalty of 1:  $f_{v_1, v_2}^{NN}(0.5) = 1, f_{v, e}^{NE}(0.5) = 1, f_G^{EE}(\pi/4) = 1$ ; where  $v_1, v_2$  is the smallest node pair so  $M_{v_1, v_2} = \min\{\text{area}(v_1), \text{area}(v_2)\} = 1$ , and  $v, e$  is the smallest node-edge pair so  $M_{v, e} = 1$ . Note that the flat line for count-based metrics lies exactly in the middle of minimum and maximum penalty when  $\gamma = 1$ , as shown in the left function in Fig. 4, but this statement does not hold when  $\gamma \neq 1$ . This choice is a specific way of calibrating penalties to counts (R3). It also constrains the relationship between  $\alpha$  and  $\beta$  so that only one parameter needs to be defined, improving the simplicity of the function.

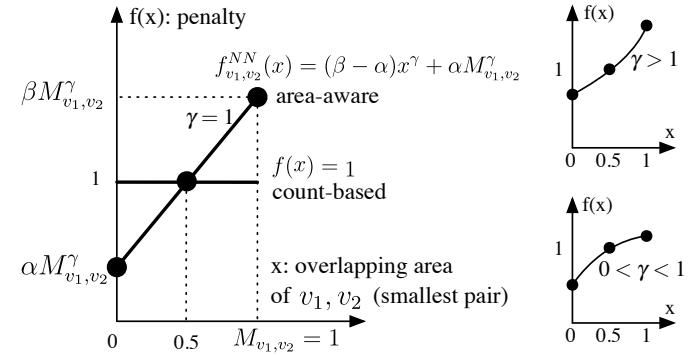


Fig. 4. Illustration of the penalty mapping function for node-node overlap of a pair of the smallest nodes ( $M_{v_1, v_2} = 1$ ) in the graph layout. The three functions show three different choices of curve shape ( $\gamma$ ).

## 4.3 Formal function instantiations

Combining the general requirements and our specific choices, we can define the penalty mapping function in a general form:

$$f(x) = (\beta - \alpha)x^\gamma + \alpha M^\gamma \quad (0 \leq x \leq M) \quad (3)$$

This general form comes with three parameters,  $\alpha$ ,  $\beta$ , and  $\gamma$ , which are responsible for minimum penalty, maximum penalty, and curve shape respectively. Our choice of  $\gamma$  values was guided by empirical results from previous work [21], [39], as discussed in C1 above. With the halfway point choice (C4), i.e.  $f(\text{half}) = 1$ , we are able to remove  $\beta$  from the general form. We can also constrain  $\alpha$  using the choices for minimum (C2) and maximum (C3).

$$(NN) \quad \beta = (1 - \frac{1}{.5^7})\alpha + \frac{1}{.5^7} \approx 1.625 - .625\alpha \quad (0 < \alpha < 1)$$

$$(NE) \quad \beta = 2 - \alpha \quad (0 < \alpha < 1) \quad (4)$$

$$(EE) \quad \beta = \frac{16}{\pi^2} - 3\alpha \quad (0 < \alpha < \frac{16-2\pi}{3\pi^2} \approx .328)$$

Inserting the known parameters  $\beta$  and  $\gamma$ , we obtain the specific instantiations of penalty mapping function for NN, NE, and EE.

$$f_{v_1, v_2}^{NN}(x) = (1 - \alpha)(2x)^{0.7} + \alpha M_{v_1, v_2}^{0.7} \quad (0 \leq x \leq M_{v_1, v_2}) \quad (5)$$

$$f_{v, e}^{NE}(x) = 2(1 - \alpha)x + \alpha M_{v, e} \quad (0 \leq x \leq M_{v, e}) \quad (6)$$

$$f_G^{EE}(x) = (\frac{16}{\pi^2} - 4\alpha)x^2 + \alpha \frac{\pi^2}{4} \quad (0 \leq x \leq \frac{\pi}{2}) \quad (7)$$

#### 4.4 Parameter space analysis

We conducted a mathematical analysis to understand the influence of the unspecified parameter in the formal statement above, namely, the minimum penalty fraction  $\alpha$ . In addition to the theoretical analysis presented in this section, in Supp. Sect. S1.1.2 we report on a computational analysis that allows us to suggest practical choices for values.

We only report on the analysis of  $\alpha$  in the NN penalty mapping function, since  $\alpha$  for NE and EE functions is very similar. Fig. 5 shows the NN penalty mapping function for the smallest node pair ( $M_{v_1, v_2} = 1$ ), using 6 different  $\alpha$  values within the range  $(0, 1)$ : 0.01, 0.2, 0.4, 0.6, 0.8, 0.99. We also include the function plot for a node pair that is 10x larger ( $M_{v_1, v_2} = 10$ ) in Supp. Fig. S1, where the shapes and relationships between curves remain the same except that the x and y values are 10x larger.

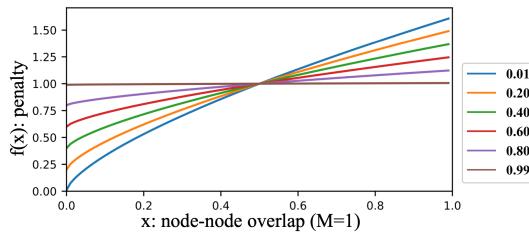


Fig. 5. The node-node overlap penalty mapping function of the smallest node pair ( $M = 1$ ), with different minimum penalty fraction ( $\alpha$ ) values. The  $\alpha$  controls the minimum penalty (y-intercept) as well as the difference between maximum and minimum (function range).

First, we observe that as  $\alpha$  increases from 0 to 1, the range of  $f(x)$  decreases, and when  $\alpha$  takes the extreme value 0.99,  $f(x)$  is approximately a flat line, degenerating to the count-based metrics case. Mathematically, the range of the penalty mapping function is  $f(M) - f(0) = (\beta - \alpha)M^\gamma = (1 - \alpha)(2M)^{0.7}$ . Semantically, the range represents its ability to distinguish between minimum, some, and maximum overlap. This ability has negative correlation with  $\alpha$  according to both our observation in Fig. 5 and the mathematical representation above. Second, we observe that as  $\alpha$  increases from 0 to 1, the touching penalty  $f(0)$  (i.e. the y-intercept of the curves) also increases from 0 to 1 in Fig. 5. Mathematically, the touching penalty is  $f(0) = \alpha M^\gamma = \alpha M^{0.7}$ . Semantically, the touching penalty represents its ability to

distinguish no overlap from minimum overlap, which has positive correlation with  $\alpha$ . Third, the changes of the range and minimum of the function is linear to the change of  $\alpha$ .

Therefore,  $\alpha$  is the trade-off between the ability to distinguish different amounts of overlap and the ability to distinguish no overlap from touching.

## 5 DATA GENERATION

To validate the sprawlter metrics, we created small synthetic graphs with both manual and force-directed layouts, and applied four different layout algorithms to large real-world graphs, resulting in 56 layouts in total: 38 synthetic and 18 real-world layouts.

We created 9 small synthetic graphs, with under 50 nodes and 150 edges. We either manually positioned the nodes or applied off-the-shelf force-directed layout algorithms in Tulip [40] to make 38 layouts for five targeted purposes: problem demonstration, debugging and sanity check, progression of clutter, variable sprawl, and limitations of our metrics. We varied the amount of clutter between NN, NE, and EE pairs, and the overall layout sparsity by pushing nodes away from each other or shrinking node sizes.

To confirm that the sprawlter metrics work with large real-world graph layouts, we used 4 large real-world graphs with 100-5K nodes and 500-10K edges. These are an academic coauthor network benchmarked in the Koala layout algorithm paper [41], a different coauthor network from the InfoVis 2004 contest [42] benchmarked in the GrouseFlocks layout paper [43], an email network provided by Leskovec et al. [44], and the add32 benchmark graph from the graph partitioning archive [45]. Supp. Table S3 provides a complete list. We applied the following layout algorithms to the real-world graphs: two single-level layouts available within Tulip [40], namely GEM [29] and FME [46] (the single-level version of FM3 [47]), the two-level Koala layout designed to emphasize cluster structure [41], and the GrouseFlocks multi-level layout [43]. We chose these based on implementation availability, computation speed, and ability to handle multi-level structure.

Our full computational pipeline is documented in detail in Supp. Sect. S5. In brief, we convert a variety of input graphs into the Tulip format, apply layout algorithms, extract geometries and the node hierarchy, compute the metrics, and finally display and analyze results for comparison. Except for applying the layout algorithms, our code is written in Python. In the supplemental materials, we include all input files, output files, implementation code, and Python scripts for results analysis. We will release these materials as open source upon acceptance.

## 6 RESULTS

We conduct a comparative analysis between the sprawlter metrics, count-based metrics, and two from Dunne et al. [11], with respect to qualitative assessment of layout pictures. We report on the analysis approach, and then discuss instructive examples across all three of the NN, NE, and EE families. For computational time, we show that our

approach only costs a constant additional time to the count-based approach. Finally, we present the computational parameter analysis for the penalty mapping functions, guided by the theoretical analysis in Sect. 4.4.

A large table with pictures and metric values for all 56 layouts and a table of running times is available in the supplemental materials.

## 6.1 Analysis approach

We compute the area-aware clutter penalties ( $A$ ), sprawl ( $S$ ), sprawlter ( $T$ ), and counts ( $C$ ) for each graph layout, in order to validate them separately. We also derive the *average penalty*  $A/C$ , the ratio between the penalties and counts, to help us find representative cases. When the average penalty is below one ( $A/C < 1$ ), the overlap areas, lengths or crossing angles are less cluttered than the count suggests, which is typically caused by overlapping between small nodes and edges, near-touching overlaps, or near-orthogonal crossings. When the average penalty is above one ( $A/C > 1$ ), the overlaps are more cluttered than the count suggests, which is typically caused by overlapping between big nodes and edges, nearly full overlaps, or glancing angle crossings.

We also compute the two most relevant metrics proposed by Dunne et al. [11] (denoted by  $D$ ), the global node-node overlap and the global crossing angle. The node-node overlap metric is the ratio of the area of the union of the node representations over their total area if drawn independently (perceived readability over best readability). A point not discussed explicitly in their paper is that this union formula assumes nodes are at the same level. It is not obvious how to extend it to multi-level layouts, since the union of parent metanodes and their constituent children would be an over-count. We thus compute it only on leaf nodes. The edge-edge crossing angle metric computes deviations from a specific optimal angle, a parameter that they set to  $70^\circ$  according to the empirical study from Huang et al. [21]. Although Dunne et al. also propose a group overlap metric that involves metanodes, it only counts the crossings between metanodes and leaf nodes, but not between metanodes, so we do not compute it.

We compare the five computational results ( $A, S, T, C, D$ ) to the layout pictures, and make a subjective qualitative assessment of correspondence between the metrics and quality of layout. For the first four metrics, low numbers are good and high numbers are bad. For  $D$ , 0 means the worst and 1 the best.

## 6.2 Comparative analysis

We present several representative cases in different types of graphs and layout algorithms, and discuss how the tested metrics reflect the readability of these cases. (We use  $\alpha = 0.20$  for NN, NE, and EE in this section.)

### 6.2.1 Node-node overlap

**Increasing metanode overlaps (synthetic).** Fig. 6 shows a small synthetic graph layout with increasing overlap between the red and blue metanodes (from left to right). The area-aware metrics for NN ( $A^{NN}$ ) also increase ( $4.27 < 14.36 < 30.62$ ), but the counts ( $C^{NN}$ ) do not ( $1 = 1 < 12$ ) and neither does Dunne's NN metric ( $1.00 = 1.00 > 0.90$ ).

Looking closer at the breakdown by metanodes and nodes, the penalties of metanodes (4.27, 14.36, 22.05), compared to penalties of leaf nodes (0, 0, 11), are the major contribution towards the totals, which matches our intuition that big metanode overlap should be penalized more heavily than that of leaf nodes, whereas the breakdown of counts does not match this intuition.

Node-node overlap						
	Near-min		Some		Near-max	
A	4.27+0=	4.27	14.36+0=	14.36	22.05+8.57=	30.62
C	1+0=	1	1+0=	1	1+11=	12
D		1.00		1.00		0.90

Fig. 6. Increasing overlap between red and blue metanodes in a small 2-level synthetic graph. Only closeups for the overlaps are shown.  $A$  and  $C$  are broken down by penalties / count of metanodes and leaf nodes. The area-aware approach ( $A$ ) matches with the visible configuration better than count-based ( $C$ ) and Dunne's ( $D$ ) approach.

**Force-directed layouts for variable-size nodes (synthetic)** Fig. 7 shows a single-level graph using two force directed layout algorithms, namely Davidson-Harel [48] and stress majorization [30]. The nodes have variable node sizes, as can occur either where node size is mapped to some quantitative attribute, or where variable-length labels is drawn within the node extent. We can see that there are no overlaps in the right layout and a few overlaps between large nodes in the left layout ( $16.05 > 0$ ), but the right one is much more sparse ( $1605.28 \gg 6.09$ ). Overall, the left has better readability ( $9.95 < 40.07$ ), because its partial overlaps are less problematic than the tiny size of the nodes on the right, with uninterpretable size coding. The difference of Dunne's metrics between left and right do not suffice to signal the large node overlaps in the left, nor they address the sparsity problem in the left.

**Many touching leaf node pairs and high sprawl (real-world).** Fig. 8 shows a large real-world graph from the partition benchmark dataset, with 4960 nodes and 9462 edges, using FM3 layout (single level). The most noticeable feature about this layout is that the nodes are too tiny to be seen and there is huge wasted space on the upper left and bottom right, resulting in a large sprawl (319.28). Despite the many touching overlaps between the tiny leaf nodes, as shown in the inset on the upper left, the area-aware penalty is still less than count ( $213.28 < 670$ ), showing that the penalties for these small overlaps sum to an appropriate amount that is non-zero but not disproportionate. Dunne's metric (almost 1.00) does not reflect the wasted space or the touching overlaps.

**High vs low sprawl (real-world).** Fig. 9 shows an email network layout by GrouseFlocks, with only a few metanodes open and others closed. There are only 61 nodes and

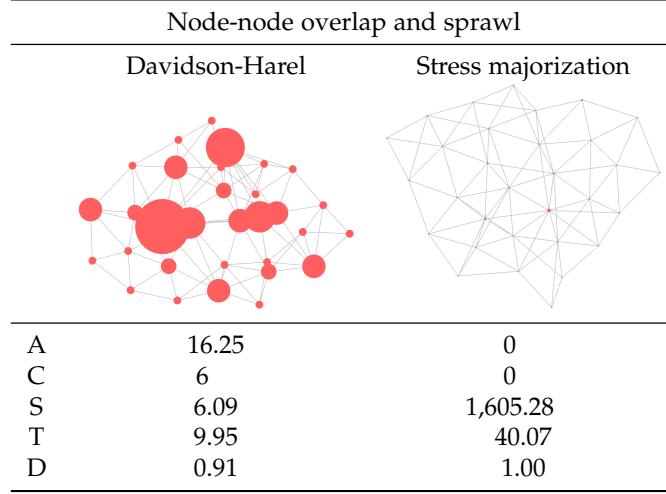


Fig. 7. A single-level synthetic graph with variable node sizes. The left layout has a lower sprawlter metric than the right layout although there are substantial overlaps between nodes; we argue that qualitatively the readability is indeed better.

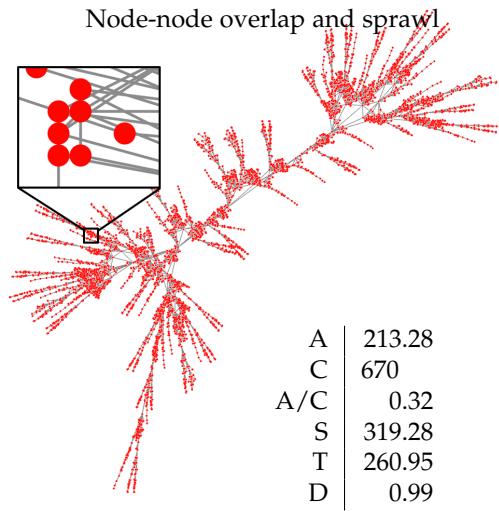


Fig. 8. Large real-world graph layout by FM3 in Tulip (single-level), with high sprawl and many touching events. The average penalty ( $A/C$ ) indicates penalties for overlaps are small but non-zero, and the high sprawlter value ( $T$ ) indicates that this layout is bad in terms of sprawl ( $S$ ) and clutter ( $A$ ).

more than 1000 edges in the left layout, and 88 nodes and more than 2000 edges in the right layout. As the overlaps only happen between metanodes, Dunne's metric (1.00) is not helpful. The left layout is less cluttered than the right one in terms of both area-aware penalties ( $133.52 < 184.13$ ) and counts ( $16 < 40$ ), but the left is less efficient in space usage than the right (i.e. higher sprawl,  $947.59 > 219.59$ ), resulting in a higher sprawlter ( $335.71 > 201.08$ ). This example and the previous one suggest that both clutter and sprawl are important aspects of measuring graph layout quality, and the trade-off between the two should be considered.

### 6.2.2 Node-edge overlap

**Increasing metanode and edge overlaps (synthetic).** Fig. 10 shows a progression of node-edge overlap on a small synthetic layout. Both the area-aware metrics ( $12.11 < 54.46 < 136.72$ ) and counts ( $6 < 15 < 36$ ) are increasing to in-

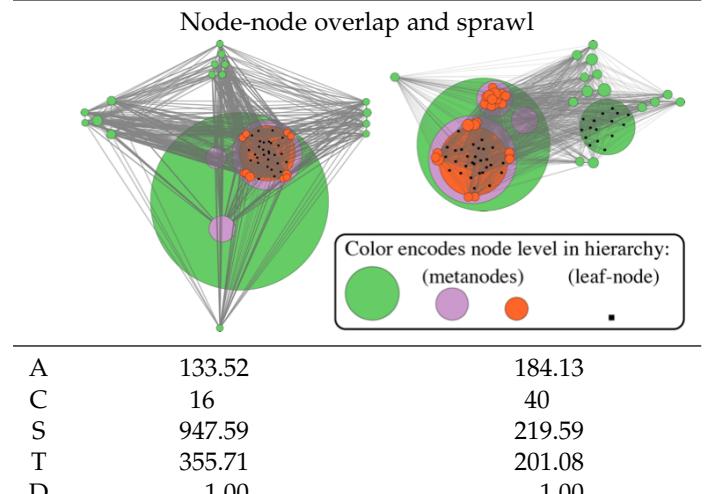


Fig. 9. Two large multi-level real-world graph layouts by GrouseFlocks (4-level). The left layout has higher sprawl but less overlap than the right one, captured by the sprawl ( $S$ ), and clutter with area-aware approach ( $A$ ) and count-based one ( $C$ ).

dicate the increasing overlap between the blue metanode and edges, and also the constituent leaf nodes and edges, but the increasing average penalty ( $2.02 < 3.63 < 3.80$ ) indicates that the penalty increases at a faster pace. As in the analysis of Fig. 6, the breakdown of penalties by metanodes and leaf nodes shows that overlap of metanodes and edges contributes more than that of the leaf nodes. The area-aware metric better aligns with our judgement of the layout pictures, because the clutter between nodes and edges deteriorates as the blue metanode moves upwards.

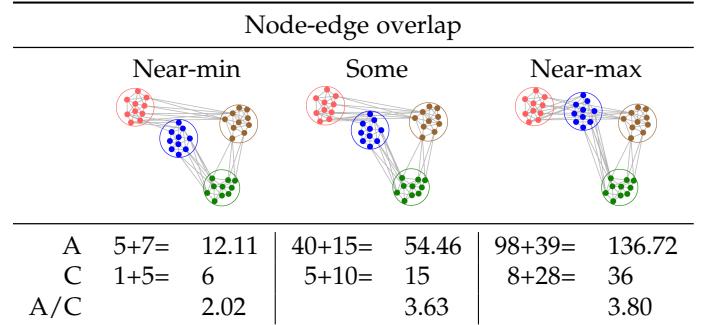


Fig. 10. Increasing overlap between blue metanodes and edges in a small 2-level synthetic graph. Although counts ( $C$ ) are increasing, the area-aware approach ( $A$ ) captures the deteriorating situation in the picture more precisely, visible via the increasing ratio of average penalty ( $A/C$ ).

### Increasing metanode and edge overlaps (real-world).

Fig. 11 shows the comparison between area-aware metrics and count-based metrics on a real-world example. On the left layout, all leaf nodes are hidden; on the right layout, we open up two of the orange metanodes, making it more cluttered. For the node-edge overlap, there are significantly fewer overlaps on the left layout, which is reflected by both metrics ( $4482.20 < 32618.75$ ,  $539 < 1650$ ), and by the average penalty ( $8.32 < 19.77$ ). Notice that there are many edges that fully cross the biggest green metanode on the right layout, resulting in an extremely high average penalty.

Also, the fact that  $A \gg C$  for both layouts matches our intuition that overlaps related to big metanodes are visually salient and should be penalized heavily.

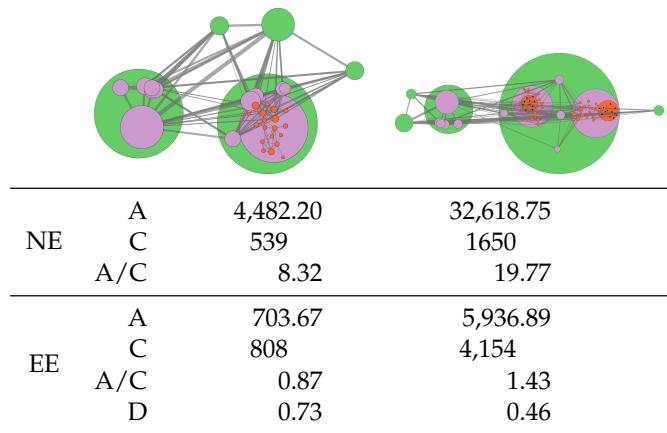


Fig. 11. Two large multi-level real-world graph layouts by GrouseFlocks (4-level). The area-aware NE metric ( $A^{NE}$ ) captures the property that there is less node-edge overlap on the left than the right. The area-aware EE metric ( $A^{EE}$ ) captures that on the left the crossing angles between edges are closer to orthogonal angles, while on the right they are closer to glancing angles.

### 6.2.3 Edge-edge crossing

**Decreasing crossing angles (synthetic).** Fig. 12 shows a progression of decreasing angles between crossing edges on a small single-level synthetic graph, from orthogonal to near-glancing angles. There are exactly two edge-edge crossings on each layout, but the area-aware metric is increasing ( $0.99 < 1.46 < 3.45 < 4.42$ ). In two layouts on the right, the edges are almost on top of each other, which looks a lot worse than the two layouts on the left. The count does not indicate such clutter on this naive example, whereas the area-aware metric is a better match to the degree of clutter of the edge-edge crossings. Dunne’s metric also goes down as clutter increases ( $0.73, 0.87 > 0.28 > 0.10$ ) (least clutter is reached at  $70^\circ$ ), but reports near-maximum EE readability in the orthogonal case which ignores the existence of remaining edge-edge crossings.

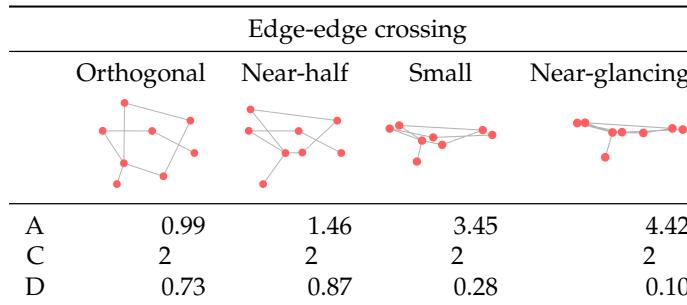


Fig. 12. Decreasing crossing angles between edges in a small single-level synthetic graph, captured by the area-aware EE metric ( $A$ ).

**Orthogonal vs glancing angles (real-world).** Fig. 11 shows similar results on real-world examples. From the layout pictures, we can see that the edge-edge crossings on the left layout are mostly near-orthogonal, while those on the right layout are mostly small and even near-glancing

angles. Not only are there fewer edge-edge crossings on the left ( $808 < 4154$ ), but the clutter caused by crossing angles is also less, which is reflected by the average penalties for EE ( $0.87 < 1.43$ ). In this case, Dunne’s metric ( $0.73 > 0.46$ ) is also successful in differentiating the two layouts.

### 6.3 Computational time

As we see from Alg. 1, the sprawlter metrics are computed along with the counts with a computational complexity linear to the number of NN, NE, and EE pairs. Therefore, we expect that the computation of the sprawlter metrics would be in the same scale as that of the count-based metrics, with a constant factor slowdown.

For every graph layout measured, we timed the full computation of the sprawlter metrics, and also of the count-based metrics by skipping the code only relevant to sprawlter metrics; that is, operations within the inner condition in Alg. 1 are all skipped except the one that increments count (line #10). We used a 2012 MacBook Pro with a 2.5 GHz Intel Core i5 CPU and 8GB RAM to run all computations, and report the average computational times over 4 runs. The absolute running time is related to the number of nodes and edges, and also the number of crossings. For example, for a large real-world graph with about 5k nodes and 10k edges (Fig. 8), it takes 128s, 505s, and 510s to compute the NN, NE, and EE sprawlter metrics respectively, while it takes 92s, 358s, and 372s to compute the counts.

We then compared the two execution times by deriving the ratio between running time of sprawlter metrics and counts, which is the equivalent of the slowdown factor. We excluded execution times under 1 second as unreliable estimates. The results align with our expectation. The average ratio over all included layouts is 1.38 for NN, 1.50 for NE, and 1.85 for EE, with relatively small standard deviations. Our findings suggest that the slowdown factor is a constant, which ranges from roughly 1 to 3.

Our Python implementation of the sprawlter metrics serves adequately as a proof-of-concept, but there are obvious directions for optimization. Our naive algorithm simply iterates every pair of nodes (and edges in NE and EE case). It would be possible to apply space decomposition algorithms to speed up the computation, such as the Bentley-Ottmann line sweeping algorithm [49] to compute all crossing edges.

## 7 DISCUSSION AND FUTURE WORK

Overall, the area-aware approach clearly outperforms the simple count-based approach in the NN, NE, and EE cases, and also the Dunne approach for the NN case. Our results are somewhat better than Dunne’s EE case: more edges does imply more crossings, but not necessarily worse crossing angles. The area-aware approach is more complicated than the others due to a parameterized penalty mapping function; we argue that benefit of precision is worth this cost.

Our approach does not impose an upper bound for the metrics, in contrast to Dunne’s metrics that are within  $[0, 1]$ . Their metrics are essentially ratios of the current readability and the best achievable readability, where 1 corresponds to the most readable and 0 to the least readable. In particular, they propose normalizing the EE metric by the  $O(|V|^2)$  term

of all possible edge crossings in a complete graph, but this number is misleading because the density levels of node-link graph drawings must be far below that ceiling in order to be comprehensible; Melançon [50] argues that typically  $|E| < 4|V|$ . While having a fixed bound is beneficial in comparing the same graph with different layouts, it is not for comparing different graphs because their metrics is a ratio relative to the graph itself, comparing the perceived readability against the best it could have; therefore the same number would have a very different meaning for a small and simple graph than for a large and complex graph.

We argue for the necessity of clutter metrics despite the existence of clutter removal algorithms [51]. The entire point of our area-aware clutter metrics is to more precisely quantify the clutter that results from overlap. It could be used algorithmically to determine which layout algorithm to use for a particular graph; for example, whether the computational cost of overlap removal is worth the improvement in the layout quality. Moreover, overlap removal sometimes results in an increase of sprawl (especially for large graphs), so the trade-off between sprawl and clutter can help determine to what extent we want to remove overlap completely, partially, or just leave it. Moreover, many layout algorithms used in practice do not guarantee the total elimination of all overlaps; their approach to accelerating computation may leave some in place. We would like to see these new metrics incorporated into layout algorithms, which would be easy to do with optimization-based approaches.

One limitation of the sprawlter metric is that only clutter and sprawl are captured. Other readability factors that affect the overall quality of a layout include its ability to show cluster structure; using spatial proximity to show groupings may incur clutter costs as a direct trade-off, as in the Koala layouts (shown in the supplemental table of full results).

Our general area-aware approach could be extended to cover a wider range of metric families as future work. Another interesting direction of future work would be a resolution-aware approach. In the mathematical formulas, pixels can be divided into infinitely small sub-pixels. If two nodes just touch each other, the overlapping area is exactly zero. However, in the practical world, pixels on a screen are discrete. An explicit awareness of the display resolution would provide the precision required to exactly assess when there is no longer visible space between two items because they occupy neighboring pixels.

A very rich direction for future work would be to incorporate more aspects of multi-level structure by considering metaedges. Progress may not occur until more layout algorithms support them and more psychophysical experimentation has taken place to untangle the factors that affect edge-edge crossing clutter beyond the crossing angle.

## 8 CONCLUSION

We contribute the area-aware sprawlter metric for graph layout to capture important properties beyond integer counts of crossings on the leaf level by accounting for the amount of overlap and beyond clutter by accounting for the geometric sparseness of layout. We instantiate and implement the sprawlter metric, and present the technical details for the crucial component, the penalty mapping

function. Through our validation with synthetic and real-world layouts, we show that the new metric succeeds at providing a better quality measure than previous work by taking into account both clutter and sprawl.

## ACKNOWLEDGMENTS

This project is supported by NSERC Discovery #RGPIN-2014-06309. We thank Daniel Archambault and David Auber for inspiring early discussions on multi-level metrics. We thank the UBC infovis group for feedback on this paper.

## REFERENCES

- [1] M. Behrisch *et al.*, "Quality metrics for information visualization," in *Computer Graphics Forum*, vol. 37, no. 3. Wiley Online Library, 2018, pp. 625–662.
- [2] S. Hachul and M. Jünger, "An experimental comparison of fast algorithms for drawing general large graphs," in *Intl. Symp. Graph Drawing*. Springer, 2005, pp. 235–250.
- [3] H. Gibson, J. Faith, and P. Vickers, "A survey of two-dimensional graph layout techniques for information visualisation," *Information Visualization*, vol. 12, no. 3-4, pp. 324–357, 2013.
- [4] J. Q. Dawson, T. Munzner, and J. McGrenere, "A search-set model of path tracing in graphs," *Information Visualization*, vol. 14, no. 4, pp. 308–338, 2015.
- [5] H. C. Purchase, D. Carrington, and J.-A. Allder, "Empirical evaluation of aesthetics-based graph layout," *Empirical Software Engineering*, vol. 7, no. 3, pp. 233–255, 2002.
- [6] F. McGee, M. Ghoniem, G. Melançon, B. Otjacques, and B. Pinaud, "The state of the art in multilayer network visualization," in *Computer Graphics Forum*. Wiley Online Library, 2019.
- [7] T. Munzner, *Visualization Analysis and Design*. CRC Press, 2014.
- [8] E. R. Tufte, *The visual display of quantitative information*. Graphics Press, 2001.
- [9] M. J. McGuffin and J.-M. Robert, "Quantifying the space-efficiency of 2D graphical representations of trees," *Information Visualization*, vol. 9, no. 2, pp. 115–140, 2010.
- [10] S. Kieffer, T. Dwyer, K. Marriott, and M. Wybrow, "HOLA: Human-like orthogonal network layout," *IEEE Trans. Visualization & Computer Graphics*, vol. 22, no. 1, pp. 349–358, 2015.
- [11] C. Dunne, S. I. Ross, B. Shneiderman, and M. Martino, "Readability metric feedback for aiding node-link visualization designers," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 14–1, 2015.
- [12] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.
- [13] P. Eades, S.-H. Hong, K. Klein, and A. Nguyen, "Shape-based quality metrics for large graph visualization," in *Intl. Symp. Graph Drawing and Network Visualization*. Springer, 2015, pp. 502–514.
- [14] H. Purchase, "Which aesthetic has the greatest effect on human understanding?" in *Intl. Symp. Graph Drawing*. Springer, 1997, pp. 248–261.
- [15] T. Dwyer, B. Lee, D. Fisher, K. I. Quinn, P. Isenberg, G. Robertson, and C. North, "A comparison of user-generated and automatic graph layouts," *IEEE Trans. Visualization & Computer Graphics*, vol. 15, no. 6, pp. 961–968, 2009.
- [16] C. Ware, H. Purchase, L. Colpoys, and M. McGill, "Cognitive measurements of graph aesthetics," *Information Visualization*, vol. 1, no. 2, pp. 103–110, 2002.
- [17] M. Klammel, T. Mchedlidze, and A. Pak, "Aesthetic discrimination of graph layouts," in *Intl. Symp. Graph Drawing and Network Visualization*. Springer, 2018, pp. 169–184.
- [18] H. C. Purchase, "Metrics for graph drawing aesthetics," *Journal of Visual Languages & Computing*, vol. 13, no. 5, pp. 501–516, 2002.
- [19] C. Dunne and B. Shneiderman, "Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts," *University of Maryland, HCIL Tech Report HCIL-2009-13*, 2009.
- [20] E. N. Argyriou, M. A. Bekos, and A. Symvonis, "Maximizing the total resolution of graphs," *The Computer Journal*, vol. 56, no. 7, pp. 887–900, 2013.
- [21] W. Huang, S.-H. Hong, and P. Eades, "Effects of crossing angles," in *Pacific Visualization Symp. (PacificVIS)*. IEEE, 2008, pp. 41–46.

- [22] A. Noack, "Energy models for graph clustering," *J. Graph Algorithms Appl.*, vol. 11, no. 2, pp. 453–480, 2007.
- [23] F. van Ham and B. Rogowitz, "Perceptual organization in user-generated graph layouts," *IEEE Trans. Visualization & Computer Graphics*, no. 6, pp. 1333–1339, 2008.
- [24] H. Almeida, D. Guedes, W. Meira, and M. J. Zaki, "Is there a best quality metric for graph clusters?" in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 44–59.
- [25] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, "Analysis of network clustering algorithms and cluster quality metrics at scale," *PLoS one*, vol. 11, no. 7, p. e0159161, 2016.
- [26] Y. Wang, Q. Shen, D. Archambault, Z. Zhou, M. Zhu, S. Yang, and H. Qu, "Ambiguityvis: Visualization of ambiguity in graph layouts," *IEEE Trans. Visualization & Computer Graphics*, vol. 22, no. 1, pp. 359–368, 2016.
- [27] H. Haleem, Y. Wang, A. Puri, S. Wadhwa, and H. Qu, "Evaluating the readability of force directed graph layouts: A deep learning approach," *IEEE Computer Graphics & Applications*, vol. 39, no. 4, pp. 40–53, 2019.
- [28] Y. Wang, Z. Jin, Q. Wang, W. Cui, T. Ma, and H. Qu, "Deepdrawing: A deep learning approach to graph drawing," *IEEE Trans. Visualization & Computer Graphics*, 2019.
- [29] A. Frick, A. Ludwig, and H. Mehldau, "A fast adaptive layout algorithm for undirected graphs (extended abstract and system demonstration)," in *Intl. Symp. Graph Drawing*. Springer, 1994, pp. 388–403.
- [30] E. R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *Intl. Symp. Graph Drawing*. Springer, 2004, pp. 239–250.
- [31] W. Huang, P. Eades, S.-H. Hong, and C.-C. Lin, "Improving multiple aesthetics produces better graph drawings," *Journal of Visual Languages & Computing*, vol. 24, no. 4, pp. 262–272, 2013.
- [32] W. Dong, X. Fu, G. Xu, and Y. Huang, "An improved force-directed graph layout algorithm based on aesthetic criteria," *Computing and Visualization in Science*, vol. 16, no. 3, pp. 139–149, 2013.
- [33] W. Huang, M. L. Huang, and C.-C. Lin, "Evaluating overall quality of graph visualizations based on aesthetics aggregation," *Information Sciences*, vol. 330, pp. 444–454, 2016.
- [34] H. C. Purchase, R. F. Cohen, and M. James, "Validating graph drawing aesthetics," in *Intl. Symp. Graph Drawing*. Springer, 1995, pp. 435–446.
- [35] H. C. Purchase, B. Plimmer, R. Baker, and C. Pilcher, "Graph drawing aesthetics in user-sketched graph layouts," in *Proceedings of the Eleventh Australasian Conference on User Interface-Volume 106*. Australian Computer Society, Inc., 2010, pp. 80–88.
- [36] W. Huang, S.-H. Hong, and P. Eades, "Layout effects on sociogram perception," in *Intl. Symp. Graph Drawing*. Springer, 2005, pp. 262–273.
- [37] S. G. Kobourov, S. Pupyrev, and B. Saket, "Are crossings important for drawing large graphs?" in *Intl. Symp. Graph Drawing*. Springer, 2014, pp. 234–245.
- [38] P. S. Bullen, *Handbook of means and their inequalities*. Springer Science & Business Media, 2013, vol. 560.
- [39] S. S. Stevens, "On the psychophysical law," *Psychological review*, vol. 64, no. 3, p. 153, 1957.
- [40] D. Auber et al., "TULIP 5," in *Encyclopedia of Social Network Analysis and Mining*, R. Alhajj and J. Rokne, Eds. Springer, Aug. 2017, pp. 1–28.
- [41] T. Itoh and K. Klein, "Key-node-separated graph clustering and layouts for human relationship graph visualization," *IEEE Computer Graphics and Applications*, vol. 35, no. 6, pp. 30–40, 2015.
- [42] J. D. Fekete, G. Grinstein, and C. Plaisant, "IEEE InfoVis 2004 contest: The history of InfoVis," [www.cs.umd.edu/hcil/iv04contest](http://www.cs.umd.edu/hcil/iv04contest), 2004.
- [43] D. Archambault, T. Munzner, and D. Auber, "GrouseFlocks: Steerable exploration of graph hierarchy space," *IEEE Trans. Visualization & Computer Graphics*, vol. 14, no. 4, pp. 900–913, 2008.
- [44] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.
- [45] A. J. Soper, C. Walshaw, and M. Cross, "A combined evolutionary search and multilevel optimisation approach to graph-partitioning," *Journal of Global Optimization*, vol. 29, no. 2, pp. 225–241, 2004.
- [46] M. Gronemann, "Engineering the Fast-Multipole-Multilevel Method for multicore and SIMD architectures," Master's thesis, Lehrstuhl für Algorithm Engineering, Fakultät für Informatik, Technische Universität Dortmund, 2009.
- [47] S. Hachul and M. Jünger, "Drawing large graphs with a potential-field-based multilevel algorithm," in *Intl. Symp. Graph Drawing (GD 2004)*. Springer, 2005, pp. 285–295.
- [48] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing," *ACM Trans. Graphics (TOG)*, vol. 15, no. 4, pp. 301–331, 1996.
- [49] J. L. Bentley and T. A. Ottmann, "Algorithms for reporting and counting geometric intersections," *IEEE Trans. Computers*, no. 9, pp. 643–647, 1979.
- [50] G. Melançon, "Just how dense are dense graphs in the real world?: A methodological note," in *Proceedings of the AVI Workshop BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2006.
- [51] T. Dwyer, K. Marriott, and P. J. Stuckey, "Fast node overlap removal," in *Intl. Symp. Graph Drawing*. Springer, 2005, pp. 153–164.



**Zipeng Liu** is a PhD student in the Department of Computer Science at the University of British Columbia, in the InfoVis group. He received his BS degree from Peking University in 2015. His research involves visualization of multi-level structures in trees, graphs, logs and models.



**Takayuki Itoh** is a full professor of the Department of Information Sciences of Ochanomizu University, Japan. He has received his Ph.D. degree from Waseda University, Japan in 1997. He has been a researcher at Tokyo Research Laboratory of IBM Japan from 1992 to 2005. He has been an associate professor in Ochanomizu University since 2005 and a full professor since 2011. His research includes graph, tree and multi-dimensional visualization.



**Jessica Q. Dawson** is pursuing her MD at the University of British Columbia. She is a former lecturer and science teaching and learning fellow in Computer Science at the University of British Columbia. Her current research interests include post-secondary computer science and medical education, patient information quality, and patient education.



**Tamara Munzner** is a professor at the University of British Columbia, and holds a PhD from Stanford from 2000. She has co-chaired InfoVis and EuroVis, her book *Visualization Analysis and Design* appeared in 2014, and she received the IEEE VGTC Visualization Technical Achievement Award in 2015. Her research interests include the development, evaluation, and characterization of visualization systems and techniques from both user-driven and technique-driven perspectives.