Eurographics Conference on Visualization (EuroVis) 2025
W. Aigner, N. Andrienko, and B. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 3

Voronoi Cell Interface-Based Parameter Sensitivity Analysis for Labeled Samples

R. Bauer[1] M. Evers[1] Q. Q. Ngo[1] G. Reina[1] S. Frey[2] and M. Sedlmair[1]

[1]Visualization Research Center (VISUS), University of Stuttgart, Germany
[2]University of Groningen, The Netherlands

Abstract

Varying the input parameters of simulations or experiments often leads to different classes of results. Parameter sensitivity analysis in this context includes estimating the sensitivity to the individual parameters, that is, to understand which parameters contribute most to changes in output classifications and for which parameter ranges these occur. We propose a novel visual parameter sensitivity analysis approach based on Voronoi cell interfaces between the sample points in the parameter space to tackle the problem. The Voronoi diagram of the sample points in the parameter space is first calculated. We then extract Voronoi cell interfaces which we use to quantify the sensitivity to parameters, considering the class label information of each sample's corresponding output. Multiple visual encodings are then utilized to represent the cell interface transitions and class label distribution, including stacked graphs for local parameter sensitivity. We evaluate the approach's expressiveness and usefulness with case studies for synthetic and real-world datasets.

CCS Concepts

• ***Human-centered computing*** → ***Information visualization; Visual analytics;***

1. Introduction

Studying the influence of parameters on the result is a common task in many areas [SHB*14], including the study of input parameter spaces of simulation ensembles or measurement results. The output data can often be divided into classes of similar behavior or labeled based on domain knowledge. One example is the study of droplets, where different parameter configurations lead to different droplet shapes [TFE22]. Transferring these labels to the parameter spaces induces a unique label for each parameter sample point. In many applications, the transition regions in the parameter space are the most interesting in the analysis process [FFRE19]. This allows answering questions such as which parameter to change to transition from one class to another and where the results are the most stable to small parameter changes. Therefore, a common goal is the quantification of the relevance of parameters. A range of local and global sensitivity analysis techniques [SRA*08], such as Sobol indices [Sob01], has been proposed, but they commonly require a differentiable outcome in parameter space. However, discrete labels that, for example, describe different shapes of droplets are not differentiable and do not have an inherent order. Thus, existing sensitivity analysis techniques cannot be applied.

In this paper, we propose a visual parameter space analysis approach that supports the sensitivity analysis of discrete labeled data. Inspired by the work of Fernandes et al. [FFRE19], we focus our study on the transitions between regions of different behaviors in the parameter space as a sensitivity measurement. Thus, the sensitivity is higher if changing the value of one parameter leads to more transitions between regions. However, we do not assume that the parameter space is sampled on a regular grid. In many applications [BJP*21, ARR23], regular sampling can be costly or even impossible. Given our input samples' discreteness (category) and irregularity, we propose to compute a Voronoi diagram to obtain a partitioning of the parameter space. The Voronoi cells approximate the regions in the parameter space and the Voronoi cell interfaces approximate the boundaries between them at which transitions occur. We then use the interfaces' sizes to determine the amount of transitions that occur when changing a single parameter. A detailed analysis of the influence of the input parameters is supported by looking at the labels between which the transitions occur. Considering the transitions over the total range covered by a parameter provides a quantitative measure of the parameter's sensitivity.

We visualize the different sensitivity measures to analyze the sensitivity in labeled parameter spaces on different levels of detail, going from local to global analysis. While the aggregated sensitivities for individual parameters can be easily visualized using common statistics visual representations, encoding the pairwise label-to-label transitions over the parameter change is more challenging as the amount of sensitivity values grows, in the worst case, quadratically with the number of labels. We propose using stacked graphs [BW08] combined with carefully designed color coding and patterns to show the quantified sensitivity values and encode the class labels between

which transitions occur. We verify our approach by applying it to synthetic data and evaluating its deviation from the analytical result. The usefulness of our approach is shown by applying it to different real-world datasets.

In summary, our main contributions are:

- A novel Voronoi-based parameter sensitivity quantification approach given labeled samples
- A corresponding parameter sensitivity visualization supporting local and global sensitivity analysis

2. Related Work

For a general overview of the area of visual parameter space analysis, we refer to the work by Sedlmair et al. [SHB*14] and the more recent survey by Piccolotto et al. [PBM23]. Several different aspects of parameter space analysis have been explored, often focusing on parameter optimization [BM10, TWSM*11, USKD12, PBCR11, BPFG11]. This work focuses on the parameter sensitivity analysis of given labeled samples. These labels come from either domain experts or via certain similarity measures for clustering or classification models on the output. Several approaches have been proposed to generate and visualize a higher-dimensional partitioning of parameter spaces [EL22, vWvL93, TWMSK18, OBJ16, WLSL17, BSM*13, EST20]. While we do not support the classification of sample points within our approach and any classification of input samples obtained with one of the approaches above could be used as input, we partition the parameter space using Voronoi to yield a geometric representation of labeled regions and their boundaries in that space.

2.1. Parameter Sensitivity Quantification

Sensitivity analysis methods can usually be divided into local and global methods [SRA*08]. Local sensitivity analysis studies the influence of small changes in one parameter, while global methods consider the entire parameter space. In this paper, we propose sensitivity analysis methods on both levels of detail. Pianosi et al. [PBF*16] present an overview of a range of sensitivity analysis techniques. Eichner et al. [EST20] use correlation to quantify the dependency on the input parameters. However, these approaches target continuous data and cannot be applied directly to labeled samples. While some methods can be generalized to ordinal categorical data, our labels generally cannot be ordered.

2.2. Visual Parameter Sensitivity Analysis

Different approaches allow for visually analyzing the impact of local variations of parameters [BPFG11, BVPtHR09, CHAS18, HWG*20, FML16, HEG*22] but they do not provide a more global view on the impact of the parameters. HyperMoVal [PBK10] visually supports sensitivity analysis by using isolines in the output space. The sensitivity of the simulation output to the input parameters can also be assessed visually [LRHS14]. Booshehrian et al. [BMPM12] proposed Vismon with a contour plot matrix to analyze the sensitivity of models with respect to two dimensions of input parameters. However, these approaches neither derive quantitative sensitivity measures nor provide an overview of the sensitivity to all parameters.

Only a few approaches for visualizing the quantified sensitivity have been proposed. Fanovagraph [FRM13] provides a visualization of the interaction strength between different parameters. It has been extended to a computationally effective tool for visually analyzing Sobol indices [YBRP21] but does not support sensitivity analysis on different levels of detail. A recent approach visualizes the spatial variation of parameter sensitivity in the simulation output [ELRL24] but only uses global sensitivity indices. Most of the aforementioned approaches focus on differentiable parameter spaces.

Evers and Linsen [EL25] consider transitions between partitions in parameter space and also take sizes of neighboring segments into account. While their visualization provides an overview of the different partitions and their connectivity, they do not relate the information to the input parameters and, thus, do not support a sensitivity analysis. Fernandes et al. [FFRE19] proposed a glyph-based approach to analyze transition patterns among partitions of labeled parameter space. The approach has scalability issues with respect to the number of partitions in the case of high-frequency partitioned parameter space. Both approaches require a regularly sampled grid or a resampling of the parameter space, which we can avoid by using the Voronoi diagram of the sample points.

In summary, none of the discussed approaches supports a quantitative sensitivity analysis of discretely labeled parameter spaces on different levels of detail.

3. Voronoi Cell Interface-Based Parameter Sensitivity Quantification

This section provides instructions on quantifying the parameter sensitivity given the discretely (category) labeled samples in the parameter space via the geometric representations from the Voronoi diagram. We show how the Voronoi cells and interfaces can be used to quantify the sensitivities on a local or global scale, for considering label-to-label or per-label sensitivities, approximating the multidimensional regions' volume (the label distribution), and also for determining non-axis aligned primary changes in labeling.

3.1. Sensitivity Quantification and the Concept of Flux

To explain our approach of sensitivity quantification for a labeled parameter space, we first consider an example without constructing a Voronoi diagram. We assume that there is some mapping from the parameter space to a set of discrete labels, which results in a set of continuous regions in the parameter space where all points in a single region are mapped to the same discrete label. For instance, in Figure 1, there is a two-dimensional parameter space where every point is mapped to one of the discrete labels "A," "B," or "C," forming three continuous regions. We note that there could be multiple disconnected regions of the same discrete labels, which is not the case in Figure 1. Our goal is to quantify the sensitivity of the labeling to a specific parameter, i.e., measure how many changes in labeling (transitions over region boundaries) occur for varying that parameter in a certain range, which is usually some quantity rather than a discrete number. Our approach for measuring the amount of transitions is inspired by the concept of flux [SLS09], a mathematical concept describing the resulting flow of a vector field through surfaces. In our case, the surfaces are the boundaries between different regions in the parameter space, and the sensitivity to a parameter is the flux

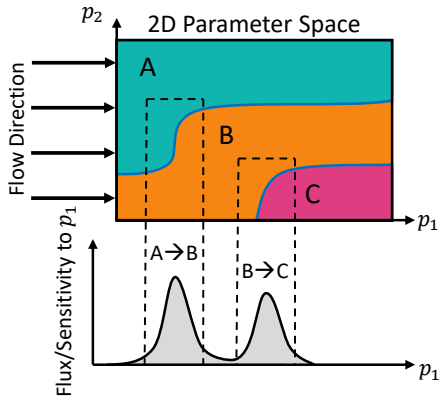

Figure 1: *Schematic parameter space segmentation with three continuous regions A, B, and C; a theoretical constant flow parallel to parameter axis p_1, and the flux at the region boundaries along the p_1-axis. Where the boundaries are perpendicular to the flow direction, more flux occurs (see the two striped boxes). The line chart represents the flux for infinitesimal ranges. However, when computing the local sensitivity, we would consider the flux that occurs in certain small but not infinitesimal ranges, which would correspond to integrating the area under the curve per range.*

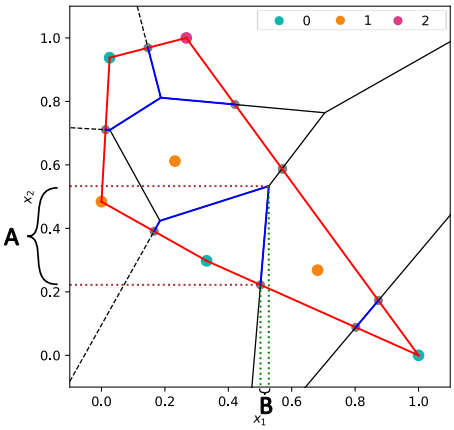

Figure 2: *A Voronoi diagram clipped with the convex hull (red) of the samples (colored by label $0, 1, 2$). The interfaces between differently labeled cells are colored blue and the clipping points are marked gray. Dashed lines represent originally infinitely large interfaces. The colored dotted lines show the length that is perpendicular to each parameters' axis for one example interface: dotted brown lines (**A**) for parameter axis x_1, and dotted green lines (**B**) for parameter axis x_2.*

that occurs at these boundaries for a constant vector field with a flow direction that is parallel to that parameter's axis. In Figure 1, the arrows indicate the direction of such flow field with vectors parallel to parameter axis p_1. The boundaries between the regions A, B, and C are the surfaces where flux occurs. Boundary segments that are perpendicular to the flow direction result in large flux, whereas segments that are rather parallel to the flow axis result in small flux due to less surface area that is exposed to the flow. The flux, therefore, corresponds to the projected size of the surfaces onto the plane (or in 2D, the axis) defined by a normal vector parallel to the flow direction, i.e., the p_2-axis in Figure 1.

The above model of parameter sensitivity provides flexibility for considering different levels of detail:

Local sensitivity analysis: By choosing small ranges for the parameters in which we consider the flux we can assess local parameter sensitivities that show fluctuations that would be not visible for more extensive ranges.

Global sensitivity analysis: By choosing a range that spans the whole available domain for a parameter, the parameter's sensitivity on a global scale can be assessed.

Label-to-label or per-label sensitivity analysis: By considering only the flux that occurs from one label to another label along the parameter axis (for an increase of the parameter (on any scale)), a more fine-grained sensitivity analysis is possible. A label-to-label analysis would show between which labels the flux occurs. A per-label analysis could answer questions like: "How does changing the parameter affect a specific label."

3.2. Voronoi-based Approximation of Regions and Boundaries

In practice, it is challenging to compute the actual flux that occurs at the region boundaries. We usually only know a set of labeled sample points in the parameter space without a definition of regions or their boundaries. Therefore, we propose to approximate them using the nearest-neighbor scheme with the multi-dimensional Voronoi diagram. Each Voronoi cell then corresponds to a sample point and label in the parameter space. Connected Voronoi cells with the same label approximate the multidimensional volume of a region, and their cell interfaces that connect differently labeled cells approximate the surfaces corresponding to region boundaries.

From now on, we refer to the sensitivity in a specific range as the total amount of transitions, instead of "flux," whereas transitions always have a direction and occur at an interface from one cell to another cell with different labels. If we refer to transitions "from" a specific cell "to" another cell, we always assume a positive transition direction for the considered parameter.

3.3. Clipping Voronoi-Cells and Interfaces

Constructing a Voronoi diagram results in a set of unbounded cells at the diagram's outside with infinitely large interfaces (for instance, the dashed black lines in Figure 2). These unbounded cells and infinitely large interfaces introduce artifacts to our later computations. Similar artifacts occur if cells and interfaces at the outside become very large, even though they might be finite. Therefore, we propose two options to clip the Voronoi diagram and reduce artifacts: *axis-aligned bounding box clipping* and *convex hull clipping*.

In the first case, the Voronoi diagram is clipped by an axis-aligned bounding box. This can be chosen as the bounding box of all samples or any bounding box containing a subspace of the parameter space,

offering a view of the transitions in that subspace. In the second case, the diagram is clipped by the convex hull of its samples to avoid the influence of very long interfaces that developed due to under-sampling. The choice between the clipping options is highly domain-dependent and can be chosen based on analysis goals.

3.4. Computing Voronoi Cell Interface Transitions

In the following, we will describe the computation of the amount of transitions for a single interface between two cells in the 1D, 2D, and then generalized 3D or higher dimensional example.

Quantifying the amount of transitions at an interface depends on the interface's size that is perpendicular to the considered parameter's axis. We describe the sizes of Voronoi cells and interfaces by their volume. A Voronoi cell in n-dimensional space contains a n-dimensional volume (short: n-volume). A cell interface of a n-dimensional Voronoi cell is of one dimension less (it is "flat" in the n-dimensional space) and lies on a $(n-1)$-dimensional hyperplane enclosing an $(n-1)$-volume.

In the one-dimensional case, the cell size is a 1-volume (a length), and the interface size is a 0-volume with a constant size of one. Here, the "amount of transitions" for a single interface between two neighboring cells with different labels is always 1, and the transition is located at the midpoint between the neighboring cells' corresponding sample points. For a specific range $r(p_0, bw) = [p_0 - \frac{bw}{2}, p_0 + \frac{bw}{2}]$, specified by an evaluation point p_0 at the center of that range and a bandwidth bw, the total amount of transitions is the number of such midpoints that fall into that range.

In 2D, a Voronoi cell size is a 2-volume (an area), and an interface between two Voronoi cells is a line segment with a 1-volume (a length) that is defined by two points. As the parameter axes are orthogonal to each other, the size of the projected line segment to the other parameter's axis is precisely the amount of transitions the interface contributes for the considered parameter. We present an example in Figure 2, where we project one of the interfaces to each parameter axis, resulting in **A**: the amount of transitions at this interface when considering parameter x_1, and vice versa, **B**: when considering parameter x_2. Due to the steeper slope of the chosen example interface for parameter x_1, the projected length **A** is larger compared to the projected length **B**, i.e., the labeling at this interface is more sensitive to parameter x_1 than to parameter x_2, for the same absolute variation in parameter values. To consider only a specific range, similar to the 1D example, the line segment (interface) has to be clipped to that range before projecting it.

We can follow the same procedure as before for three- and higher-dimensional spaces. However, in contrast to the 2D case, where each Voronoi interface is defined by exactly two points, in 3D or higher, each interface may be determined by 3 or more points (at least n points for an $(n-1)$-dimensional interface of an n-dimensional cell), resulting in an arbitrary convex polytope. To consider the amount of transitions of an interface for a specific range $r(p0, bw)$, similar to before, the interface has to be clipped to that range first. The clipping range for parameter x_i is enclosed by two n-dimensional halfspaces $(x_i = p_0 - \frac{bw}{2})$ and $(x_i = p_0 + \frac{bw}{2})$, which are two $(n-1)$-dimensional hyperplanes with normal vectors in opposite directions.

In the 3D case, a cell has a 3-volume, and cell interfaces are convex polygons with a 2-volume. Figure 3 shows an example of a single interface (colored blue) of a 3D Voronoi cell that resembles a cube. The interface is clipped by two halfspaces shown with the two gray planes and then projected to the remaining x_2x_3-plane. The 2-volume (area) of the projected clipped interface (colored in orange) is the amount of transitions this interface facilitates in the range enclosed by the two gray planes considering parameter x_1.

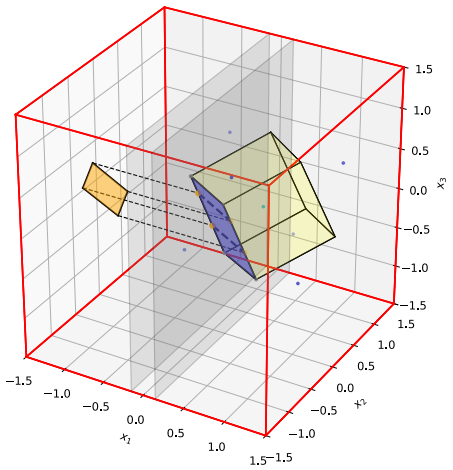

Figure 3: *A Voronoi cell in 3D in the shape of a cube. The figure illustrates the projected 2-volume of the blue interface in the x_1-range enclosed by the two gray planes. The projection of the interface onto the x_2x_3-plane after clipping to that range is shown in orange.*

3.5. Sensitivity Computation

In the previous section, we showed how we compute the amount of transitions that occur at a single interface regarding a specific parameter and range. Depending on the aggregation of transitions from individual interfaces, we can get measures for label-to-label sensitivity and per-label sensitivity on a local or global scale.

Choice of the bandwidth: Conceptually, we can compute the amount of transitions for the interfaces in any parameter range. However, by choosing non-overlapping ranges of a fixed width (the bandwidth bw) along a parameter axis, similar to the bins of a histogram, we can yield comparable local sensitivity values along that axis. Summing up all local sensitivity values for one parameter then yields the global sensitivity for that parameter and does not require computing it again for the whole range. The bandwidth is domain-dependent and may be chosen differently per parameter. It determines the amount of bins required to cover the whole available parameter range. A larger bandwidth generally requires higher computational effort but offers more details on a local scale, whereas a higher bandwidth may obscure local fluctuations of the sensitivity.

Local label-to-label sensitivity We propose to discretize each parameter dimension of the parameter space into n_i non-overlapping bins b_j, each of extent bw_i, per parameter dimension x_i, and L to be the number of labels from the input sampled data set. We then compute the amount of transitions per interface for each bin's range, as described in Section 3.4, and aggregate the individual values based

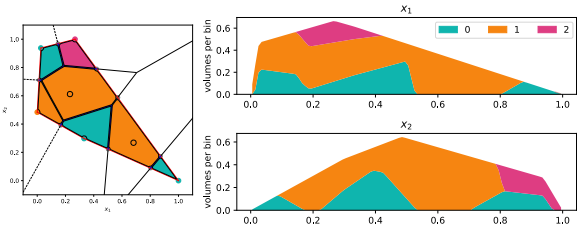

Figure 4: *A convex hull-clipped Voronoi diagram with label-colored cells (left), and its absolute label distribution plots (right).*

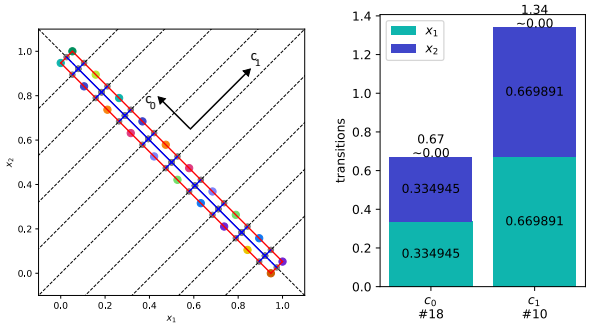

Figure 5: *Left: A Voronoi diagram clipped via the convex hull that results in two primary transition directions (indicated by the two arrows, scaled proportionally to the corresponding clusters' weights). Right: the centers of the two corresponding clusters are weighted by their overall amount of transitions in the stacked bar chart, with each bar representing a parameter dimension. A striped bar indicates a negative entry, e.g., x_1 in cluster c_0.*

on the involved labels per interface to yield (directed) label-to-label sensitivities. Our local label-to-label sensitivity computation output along parameter dimension x_i is a list of multidimensional arrays M_i, each with shape (L, L, n_i). Each entry $M_i[A,B,j] = s(j,i,A,B)$ contains the amount of transitions from label A to label B for all interfaces in the clipping range of bin b_j for parameter dimension x_i.

Local per-label sensitivity: We aggregate the above label-to-label sensitivity regarding incoming and outgoing transitions per label. Specifically, we denote $is(j,i,A) = \sum_{B_k} s(j,i,A,B_k)$ for an aggregated quantity on the amount of transitions that leave the label A. And $os(j,i,B) = \sum_{A_k} s(j,i,A_k,B)$ for an aggregation quantity on the amount of transitions that add to a label B.

The global sensitivity per parameter dimension: We propose aggregating the above sensitivity values for all bins b_j along the parameter dimension x_i; we denote $gs(i,A,B)$ to be the global label-to-label sensitivity to changes from region A to B for x_i with $gs(i,A,B) = \sum_j s(j,i,A,B)$. By comparing gs for different parameter dimensions, the most influential parameters can be identified. Similar aggregations can be done for is and os. Finally, a global measure that considers all transitions between label pairs can be computed via $\hat{gs}(i) = \sum_{A_k,B_k} gs(i,A_k,B_k)$, which is a single scalar per parameter dimension x_i.

3.6. Voronoi Cell-Based Label Distribution

Besides the parameter sensitivity computation, we also study the label distribution for each parameter dimension x_i. Mainly, it is a list of matrices D_i, each with shape (n_i, L), where n_i is the number of discretized bins per parameter dimension x_i and L is the number of labels from the input sampled data set. Each entry $D_i[j,A]$ contains the summed cell sizes (n-volumes) of all the cells with label A clipped to the clipping range of bin b_j for parameter dimension x_i.

Computing the label distribution provides an overview of the parameter space. It complements the parameter sensitivity quantification by providing the context of the overall label distribution. For instance, a high sensitivity does not necessarily mean that the overall label distribution changes since there could be many simultaneous transitions from one label to the other and vice versa. Furthermore, when convex hull clipping is applied, a change in the absolute distribution does not necessitate transitions but instead may be induced by a change in the size of the convex hull and the considered parameter space. Overall, convex hull clipping should be used carefully, or even replaced with a different strategy, when the data is distributed in a highly non-convex space.

The *absolute label distribution* shows the total sizes of clipped cells per label and clipping ranges determined by the bandwidth, as shown in Figure 4.

3.7. Interface Orientations and the Primary Transition Direction

The described sensitivity quantities focus on transitions along parameter axes. In another perspective, the orientation of interfaces could provide insights into the primary non-axis-aligned transition directions to complement the overall sensitivity analysis. For instance, transitions that majorly happen due to a joint change of parameters can only be estimated given the axis-aligned evaluation approach.

By considering the normals of the hyperplanes that each interface lies on, we can determine the direction of the maximal transitions, i.e., the highest sensitivity due to changes in multiple parameters at once. Since there might be multiple common transition directions, we first apply the K-Means clustering technique to the normals. To cluster the normal vectors, we transform them to have a positive dot product with the one-vector 1^N by reversing their orientation. Otherwise, a mean between two normal vectors with opposite directions would result in the zero vector, which is undesired. We then apply K-Means with a plane-plane distance metric that computes the angle between two planes defined by their normal vectors: $d(n_1,n_2) = \arccos(|n_1 \cdot n_2|) * \frac{2}{\pi}$. The K-Means algorithm is further slightly adjusted by adding an additional step to normalize the means when updating the centers (mean normal vectors). The supplemental material contains illustrations for the plane-plane metric.

The K-Means output provides the clustered orientations of the interfaces transformed for improved clustering as a list of *cluster means*, which represent the *primary transition direction per cluster*, and the *total amount of transitions per cluster*.

Figure 5 shows an artificial example (left) that displays two clear primary transition directions. The visualization of the clustering (at

the right) displays their orientation and magnitude (see Section 4 for details on the visual encoding of the quantity).

3.8. Implementation Details

We use the QHull (quickhull algorithm) [BDH96] Python bindings, available via scipy [VGO*20], to compute Voronoi diagrams, convex hulls, clipping of cells and interfaces via halfspace intersections, and the volumes of interfaces and Voronoi cells (represented as convex hulls). For clustering, we use the K-Means implementation of Novikov [Nov19] with the plane-plane distance metric. Our implementation is available at https://doi.org/10.18419/DARUS-4930.

4. Visual Encodings

In this section, we provide our design choice for visual encodings of the derived quantities using our quantification approach described in Section 3, which includes: (a) *Voronoi cell interface-based parameter sensitivity quantification*, including local and global amount of label-to-label and per label transitions per parameter dimension, (b) *Voronoi cell-based label distribution*, and (c) *interface orientations*. We first present analysis tasks, possible visualization designs, and the reasoning for our final design choice. We used convex hull clipping for deriving the quantities in the corresponding example figures.

4.1. Analysis Tasks

Our visual analysis approach should support different levels of details, e.g., global and local sensitivity analysis. To achieve that, we identify a list of analysis tasks:

I. **Identify the most sensitive parameters**: compare and find which parameter dimension leads to the most transitions.
II. **Discover the transition between labels**: observe and identify which transitions between which labels happen.
III. **Find sensitive parameter ranges**: present local sensitivity and compare them to find the most sensitive parameter ranges and how they affect the label distribution.
IV. **Visual present the overview of the multidimensional distribution** of labels along parameter axes.
V. **Determine the primary transition direction**: which simultaneous parameter changes lead to the most transitions.

4.2. Design Options and Choices

We aim to provide static visualizations for the quantities (a)-(c) to perform the analysis tasks (I - V).

Task I: The most sensitive parameter is the one that has the most transitions. A bar chart provides a good option for visualizing *the global amount of transitions*. However, (a) also provides additional information that we may use to answer which transitions between labels happened most frequently along each parameter dimension, i.e., the label-to-label transitions. We can encode this additional information by color-coding the bar chart bars. However, we cannot generally color-code each label-to-label transition to the quadratic number of possible pairs and thus lack differentiable colors. Therefore, we choose a different color coding. We use one bar per parameter dimension and one color per label. The bar consists of

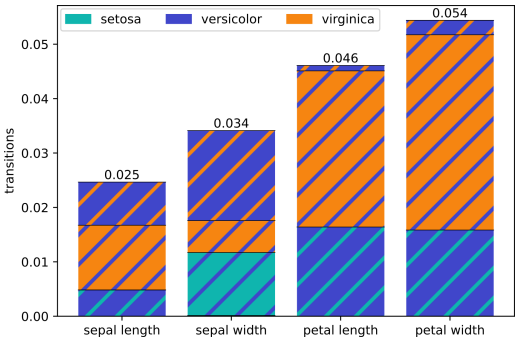

Figure 6: *The global transitions per parameter for the Iris dataset, color-coded to visualize label-to-label transitions. Rectangle colors encode target labels, while stripe colors encode origin labels.*

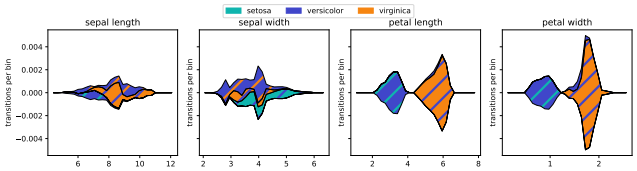

Figure 7: *The label-to-label transitions per parameter for the iris dataset as stacked graph. The areas of the stacked graph encode for transition volume. Their color encodes the target label, and the color of their stripes encodes the origin label.*

stacked rectangles, each belonging to one label-to-label transition, e.g., *setosa* → *versicolor*. The height of the rectangle encodes the amount of transitions of this pair. To differentiate between transition pairs, we color the rectangle with the target label and add stripes with the color of the origin label, as shown in Figure 6 for the Iris dataset.

Task II: We can visualize the local label-to-label transitions in (a) directly. It supports identifying the transitions between labels that happen at different parameter values. We chose a symmetric stacked graph for visualizing the local label-to-label sensitivity. Similar to the stacked bar chart, the stacked graph supports viewing the individual label-to-label transitions compared to the total sum (per bin), making it easier to assess their proportions. We encode the label-to-label transitions in the stacked graph with the same color coding we used in task I. An example visualization of label-to-label transitions for the Iris dataset is shown in Figure 7.

Task III: With the same argument above, we also visualize the local *transitions per label* for incoming and outgoing transitions in (a) with stacked graphs. We encode the relevant label by color. Figure 8 shows the visual encoding for incoming (upper row) and outgoing (lower row) transitions per label.

Task IV: We also visualize the *label distributions* (b), a list of two-dimensional matrices, using stacked graphs. However, this time, we chose a stacked graph with a zero baseline instead of a symmetric one to better assess the absolute values in the distribution. Figure 9 shows the visual encoding of our distribution quantity per label and dimension for the Iris dataset.

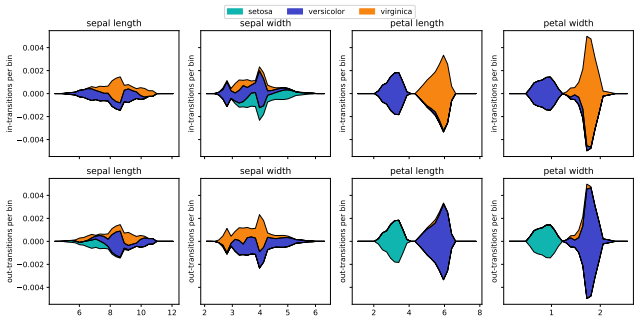

Figure 8: *Input- and output transitions per parameter for the Iris dataset as stacked graphs matrix. The upper row shows the incoming transitions for each label and the lower row shows the outgoing transitions for each label. Color encodes the corresponding labels.*

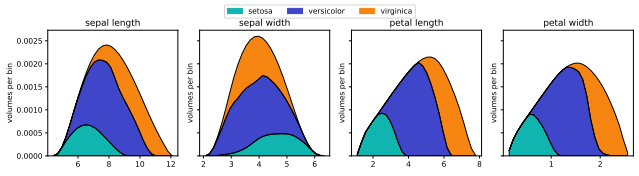

Figure 9: *The label distribution per parameter axis for the Iris dataset. It is based on the cell volumes resulting from constructing the Voronoi diagram on the labeled samples.*

Task V: Quantity (c) describes *the primary transition directions* of the parameter space. Potential options include star-glyph visualizations or flower-glyphs [vOVR23], and parallel coordinate plots (PCP) [LMW*17] for visual encoding (c). PCPs suffer from overplotting, and comparing individual parameters by their magnitude would be tedious. Flower- and star glyphs can show the absolute values per dimension but would need additional encoding to encode whether the original entry was positive or negative. Neither option supports the accurate assessment of the overall magnitude. Therefore, we choose a bar chart to visualize the vectors of (c), where each rectangle in the bar chart is color-coded according to one parameter dimension, and its size corresponds to the absolute value of the parameter's entry. We add black stripes to the rectangles of negative entries. Figure 10 shows such a visualization in the example of the Iris dataset. We clustered the interface orientations into five groups and sorted them by their total size of interfaces. We annotate the bar chart with additional information, such as the total transition size and the average distance of the interfaces to their normal vector (cluster center, $\sim x$) above, and with the cluster index (c_x) and total number of interfaces ($\#x$) below.

5. Evaluation

In this section, we provide an algorithmic evaluation of our sensitivity quantification, a brief comparison to alternative approaches, and evaluate our technique's results with three case studies. Individual parameters can span different ranges or even have different orders of magnitude. For all case studies, we scale the parameters to the range $[0, 1]$ before constructing the Voronoi diagrams but assign the original values for visualization. The Voronoi diagrams in the case studies were all clipped with the convex hull of the scaled samples.

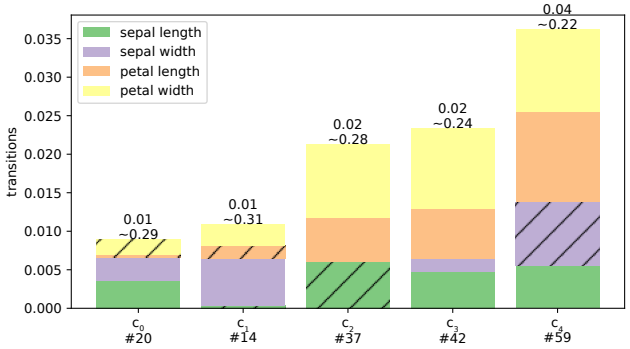

Figure 10: *Primary direction of transition vectors for the Iris dataset. The transition interfaces were clustered into five groups. The total height encodes the size of total transitions along this direction. The height of individual rectangles encodes the proportion of transitions per parameter dimension. The rectangles are color-coded by their parameter dimension.*

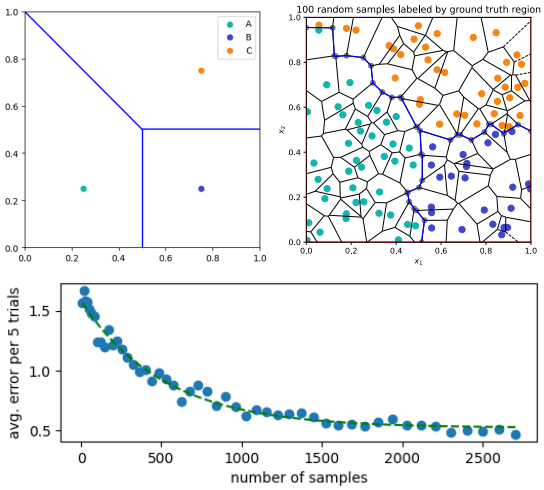

Figure 11: *Top left: ground truth segmentation (and perfect placement of three samples). Top right: 100 random samples labeled by their segment. Bottom: Average error between ground-truth transition interface sizes and Voronoi-based transition computation given n random samples and 5 trials per n.*

5.1. Algorithmic Evaluation

We evaluate the Voronoi-based sensitivity quantification provided an artificial 2D parameter space segmentation to assess how well a random sampling with correct labeling would approximate the "real" boundaries and resulting parameter sensitivity. The ground truth segmentation is shown in Figure 11 (top left). With a perfect placement of three labeled samples, our approach would precisely approxi-

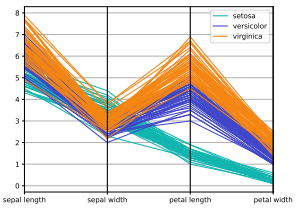

Figure 12: *Parallel coordinates plot of the Iris dataset.*

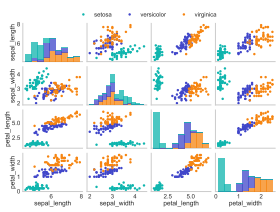

Figure 13: *Scatter plot-matrix of the Iris dataset.*

mate the same segmentation and transitions. However, in practice, many random samples will probably approximate the ground truth segmentation, such as shown in Figure 11 (top right)). Increasing the number of random samples should improve the approximation and reduce the average error between ground truth transitions and those computed with our approach. This is reflected in Figure 11 (bottom), which shows the error to the ground truth by the number of random samples. However, while the error keeps decreasing, it is still relatively high given the large number of random samples. This difference is given due to the roughness of the aggregated Voronoi interfaces (many small, slightly differently oriented interfaces), which lead to many more transitions per parameter dimension than the few long and smooth segmentation boundaries of the ground truth. However, a sensitivity analysis with our approach is still meaningful. The "approximation roughness" is introduced throughout the whole parameter space and differences at local and global scale are still comparable since they suffer from the same approximation problem, i.e., the computed sensitivity is slightly higher everywhere.

5.2. Comparison to Alternative Approaches

Typical design choices for visualizing multi-dimensional parameter spaces are parallel coordinate plots (PCP, see Figure 12) and scatter plot matrices (SPLOM, see Figure 13). Both plots visualize the sample distribution per label projected to one (PCP) or two (SPLOM) dimensions simultaneously. Neither provides a means to quantify the proportion of labels in the multidimensional space or the amount of transitions that occur. While the quantification is lacking, SPLOM and PCP may still indicate possible sensitive ranges for the parameters, e.g., ranges that have a lot of overlap from samples with different labels as well as clear gaps between two clusters of samples with the same labels. The pairwise projections in the SPLOM also support estimating the transition direction regarding two parameters at a time, though it is hard to estimate the actual transitions that happen due to the missing relation to other dimensions. Both approaches suffer from overplotting and usually require interaction with brushing and linking to make sense of the data. We provide static visualizations that fully visualize the computed quantities to solve our analysis tasks without interaction.

5.3. Case Study 1: Iris Dataset

The Iris dataset contains 150 flower samples, where each flower is either from the species *setosa*, *versicolor*, or *virginica*. The flowers have been examined by their *sepal-* and *petal length*, as well as *sepal-* and *petal width*. We consider the Iris dataset a parameter space of four dimensions ($N = 4$), with three potential output labels ($L = 3$). PCP and SPLOM visualizations of the Iris dataset are shown in Figure 12 and Figure 13.

Species distribution: One interesting aspect when inspecting the pairwise distribution of the labeled samples in the SPLOM of Figure 13 is that the label *versicolor* appears to be more distributed in the multidimensional space, e.g., regarding the *sepal length* than *virginica*. However, the plot of our distribution quantity in Figure 9 (bottom) shows that *versicolor* takes up a more significant portion of the space along *sepal length* parameter dimension. A notable difference is also visible regarding the stacked histogram of sample distribution in the SPLOM, which significantly differs from our distribution for *petal length* and *petal width*. The histogram contains peaks and gaps that are not present in our distribution.

Most sensitive parameter: The global transition plot in Figure 6 shows the aggregated amount of transitions per parameter dimension. It shows that parameter *petal width* is responsible for most of the transitions, followed by *petal length* and *sepal width*. Interestingly, we see no transitions between *setosa* and *virginica* for either parameter dimension and direction, which we could only identify by thoroughly inspecting the SPLOM or PCP.

Sensitive parameter ranges: Figure 8 shows the overall amount of incoming and outgoing transitions for different flower species along each parameter axis. It shows, for instance, that *setosa* regions lose volume along the first half of the extents for *sepal length*, *petal length*, and *petal width* while gaining volume starting with increasing *sepal width*. Furthermore, it clearly shows two major sensitive ranges for parameters *petal length* and *petal width*, displaying two clear transition peaks for each.

Transition between labels: The label-to-label transition plot in Figure 7 shows the origin and target regions for transitions. For instance, the volume lost for *setosa* in the first half of *sepal length* transitions to *versicolor*. Meanwhile, an increase of *sepal width* generally leads to transitions from *versicolor* regions to *setosa*.

Primary transition directions: The primary transition directions in Figure 10 indicate the set of directions in the parameter space with the largest amount of transitions. For the Iris dataset, we can see that the primary transition directions are not axis-aligned but generally happen due to a combination of parameter changes. Here, most transitions occur for simultaneous changes of all four parameters, weighted similarly to the global transitions per parameter.

5.4. Case Study 2: Semiconductor

This dataset contains simulation data for studying the transport properties of a semiconductor quantum wire [EL22, EL25]. The simulation output can be clustered based on similar behavior. Typical simulation outcomes that represent different classes are shown as inlays in Figure 14. In this paper, we use the same clustering as used by Evers and Linsen [EL22]. The simulations depend on four different parameters, which are the delay between laser pulses (*pulsedelay*), the energy difference to the bandgap energy (*exciteOverGap*), the spatial variance (*pumpsigmax*) and the area (*pumparea*) of the laser.

When looking at the global parameter sensitivity, it becomes clear

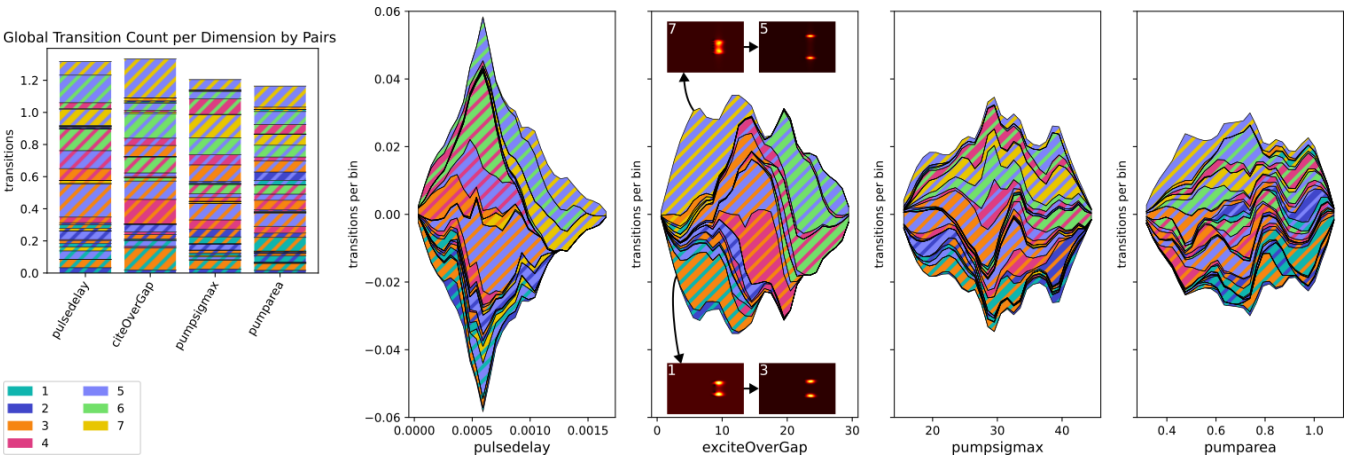

Figure 14: *Gobal (left) and local (right) sensitivity analysis results for the semiconductor quantum wire. The global sensitivity analysis shows similar values for the different parameters while the local sensitivity analysis shows variations over the parameter range (pulsedelay) and transitions from class 7 to 5 and 1 to 3 (exciteOverGap).*

that all parameters show a similar sensitivity (see Figure 14). Investigating the local sensitivity provides more detailed information. Looking at the overall distribution of the sensitivities reveals that increasing the pulse delay from 0.0005 ns to 0.0007 ns leads to the most transitions which clearly shows that this parameter region is the most sensitive to changes. The sensitivity to the other parameters varies less over the investigated range.

The label-to-label sensitivity with respect to the parameter *exciteOverGap* reveals variations with respect to the parameters between which the transitions occur. Especially notable is the high sensitivity with respect to transitions from class 7 to 5 for the lower part of the parameter range. The characteristic behavior for these classes, which is shown in the inlays in Figure 14, reveals that changing this parameter in this range leads to transitions from two connected peaks (see inlay for class 7) to two separate peaks (see inlay for class 5). In the same parameter region, a significant amount of transitions from class 1 to 3 can be observed. When observing the visualization of the simulation outcome, it also indicates a separation of peaks but with other patterns at the boundary of the peak. Thus, the sensitivity analysis does not only highlight parameter ranges in which the sensitivity is increased but also indicates in which parameter range the transition occurs, which can be used for parameter tuning.

5.5. Case Study 3: Droplet Impact Experiments

This case study investigates a labeled set of droplet impact experiments. Here, in each experiment, a droplet of one fluid is dropped onto a small film of a different fluid, resulting in different splash patterns. The experiments study the splash patterns that develop for different types of liquids, velocities, film thickness, and other parameters. The experiments were labeled by their resulting splash patterns. The labels are “crown”, “splash”, “jetting”, and “crown splash.” The type of fluids were indexed to represent them numerically. We calculated the three quantities using our approach for a subset of the experiment parameters, that are, the *dimensionless film thickness*, *Weber number*, *droplet liquid*, and *wall-film liquid*.

Figure 15 shows the global label-to-label transitions (left) and the local label-to-label transitions (right) side-by-side. The global view makes it easy to spot the most sensitive parameter, the *Weber number*, closely followed by the *dimension less film thickness*. The type of liquid appears to be overall less relevant.

The most sensitive range for *dimensionless film thickness* and *Weber number* are around 0.3 and 1000 respectively. Multiple transitions to *bubble-splash* occur for *film thickness* values around 0.2 and *Weber number* values 1000 to 1500. Most of these transitions originate from *splash* or *crown* pattern. There is a significant spike between *Hyspin* and *Hexadecan* liquid for both *droplet-* and *wall-film liquid*. For the *wall-film liquid*, exchanging *Hexadecan* with *Hyspin* shows many transitions from *splash* to *crown* patterns. For the *droplet liquid*, it results in a shift from *bubble-splash* to *crown* patterns instead. The primary transition directions appear mostly axis-aligned with one of the parameters, as shown in Figure 16. We note that the transitions resulting from changes in discrete values like liquid depend on the ordering of the values. A different ordering would show different transitions between other pairs of liquids.

5.6. Runtime Experiments

The runtime of our approach regarding the number of bins m, which relates to the bandwidth by $bw = \frac{1}{m}$, given normalized sample points, appears to be growing linearly by m. We also see a linear growth in runtime regarding the number of samples. The runtime regarding the data dimensionality appears to be growing exponentially. More results are available in the supplemental material.

6. Discussion

Our approach supports the sensitivity analysis for multi-dimensional parameter spaces. We show the plausibility of the results by applying our techniques to the well-known Iris dataset. The applicability for real-world data is shown based on the application to two real-world

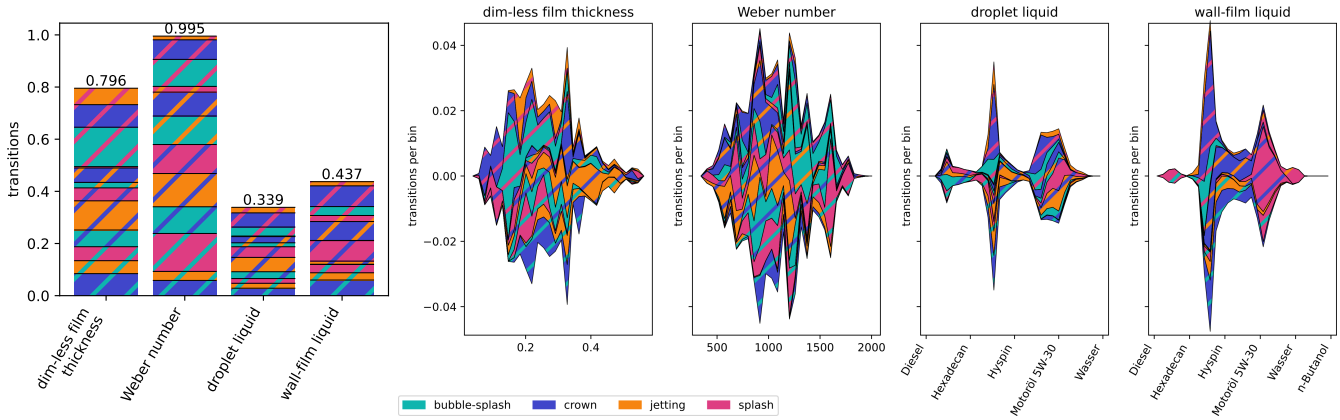

Figure 15: *The global sensitivity (left) and local sensitivity (right) for the parameters dim-less film thickness, Weber number, droplet liquid, and wall-film liquid of the droplet impact experiments, by label-to-label transitions.*

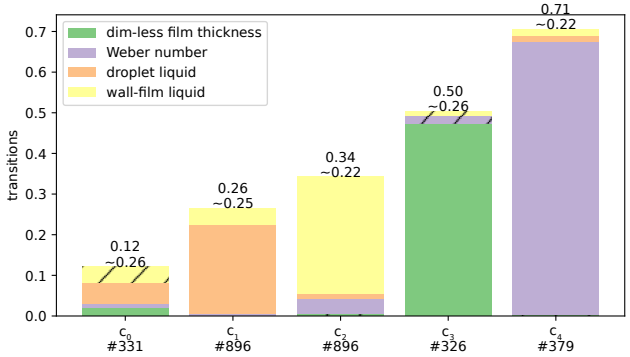

Figure 16: *The primary transition directions for the droplet impact experiments. The mean interface orientations appear mostly axis-aligned with one of the parameter axes.*

datasets. The continuous description of the boundaries between different regions does not require a resampling of the parameter space as employed by Evers and Linsen [EL22]. Further, our approach generalizes to any existing partition of a multi-dimensional space if the interfaces are provided explicitly.

Our approach also has some limitations. Computing the Voronoi diagram and clipping interfaces and cells in high-dimensional spaces is computationally expensive. As the Voronoi diagram computation can be done as a separate preprocessing step for the analysis, longer computation times do not hinder the analysis process. Still, we found that our approach is computationally feasible for up to 7 dimensions on consumer hardware such as a computer with 32 GB RAM. Our approach strongly depends on the quality of the Voronoi diagram, so numerical problems such as degenerate interfaces or a rough approximation of the original region boundaries also affect our approach. Similar to other sensitivity analysis approaches, we require a sufficiently dense sampling of the parameter space. Missing values might lead to artifacts in the Voronoi diagram, which can cause wrong estimates of the amount of transitions.

Our visualizations allow for showing label-to-label transitions and their variations over changing each parameter. While this approach provides a static overview of the parameter transitions, the amount of label-to-label transitions scales, in the worst case, quadratically with the number of output labels. This limits the visual scalability if many transitions between different segments occur. While our approach depends on a set of hyperparameters like bandwidth, the clipping method, and the choice of parameter range normalization, little dedicated hyperparameter tuning is required. Except for the bandwidth, all hyperparameters can be directly chosen based on analysis tasks and domain expertise.

7. Conclusion

We quantify sensitivity based on transitions between differently labeled regions. Our sensitivity measures allow for detecting parameter configurations in regions of the parameter space that are more or less likely to change the output behavior of a simulation when slightly varying some of their parameters.

In the future, we plan to explore other applications or improvements of the Voronoi-based parameter sensitivity model. For instance, improving the approximation error of region boundaries by further processing the individual interfaces, or using ray tracing-inspired approaches for searching interesting patterns along a ray and identifying oscillating changes between different regions. Another future research direction includes the integration of our approach into an interactive parameter space analysis system. By combining our method with other existing methods, we see the potential for a more comprehensive investigation of the parameter space.

Acknowledgments

Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 - 390740016, Project 327154368 - SFB 1313 (D01), and Project 251654672 – TRR 161 (A01, A08). We acknowledge the support of the Stuttgart Center for Simulation Science (SimTech).

References

[ARR23] AHMED A. A., RAHIMIAN M. A., ROBERTS M. S.: Estimating treatment effects using costly simulation samples from a population-scale model of opioid use disorder. In *2023 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)* (2023), pp. 1–4. doi:10.1109/BHI58575.2023.10313496. 1

[BDH96] BARBER C. B., DOBKIN D. P., HUHDANPAA H.: The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw. 22*, 4 (1996), 469–483. doi:10.1145/235815.235821. 6

[BJP*21] BAE J. H., JO W., PARK J. H., VOYLES R. M., MCMILLAN S. K., MIN B.-C.: Evaluation of sampling methods for robotic sediment sampling systems. *IEEE Journal of Oceanic Engineering 46*, 2 (2021), 542–554. doi:10.1109/JOE.2020.3005576. 1

[BM10] BRUCKNER S., MÖLLER T.: Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1468–1476. doi:10.1109/TVCG.2010.190. 2

[BMPM12] BOOSHEHRIAN M., MÖLLER T., PETERMAN R. M., MUNZNER T.: Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum 31*, 3pt3 (2012), 1235–1244. doi:10.1111/j.1467-8659.2012.03116.x. 2

[BPFG11] BERGER W., PIRINGER H., FILZMOSER P., GRÖLLER E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum 30*, 3 (2011), 911–920. doi:10.1111/j.1467-8659.2011.01940.x. 2

[BSM*13] BERGNER S., SEDLMAIR M., MÖLLER T., ABDOLYOUSEFI S. N., SAAD A.: Paraglide: Interactive parameter space partitioning for computer simulations. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1499–1512. doi:10.1109/tvcg.2013.61. 2

[BVPtHR09] BRECHEISEN R., VILANOVA A., PLATEL B., TER HAAR ROMENY B.: Parameter sensitivity visualization for dti fiber tracking. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1441–1448. doi:10.1109/TVCG.2009.170. 2

[BW08] BYRON L., WATTENBERG M.: Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1245–1252. doi:10.1109/TVCG.2008.166. 1

[CHAS18] CUTURA R., HOLZER S., AUPETIT M., SEDLMAIR M.: VisCoDeR: A Tool for Visually Comparing Dimensionality Reduction Algorithms. In *Euro. Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (2018), pp. 641–646. URL: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-74.pdf. 2

[EL22] EVERS M., LINSEN L.: Multi-dimensional parameter-space partitioning of spatio-temporal simulation ensembles. *Computers & Graphics 104* (2022), 140–151. doi:10.1016/j.cag.2022.04.005. 2, 8, 10

[EL25] EVERS M., LINSEN L.: 2D embeddings of multi-dimensional partitionings. *IEEE Transactions on Visualization and Computer Graphics 31*, 1 (2025), 218–228. doi:10.1109/TVCG.2024.3456394. 2, 8

[ELRL24] EVERS M., LEISTIKOW S., RAVE H., LINSEN L.: Interactive visual analysis of spatial sensitivities. *IEEE Transactions on Visualization and Computer Graphics* (2024). doi:10.1109/TVCG.2024.3433001. 2

[EST20] EICHNER C., SCHUMANN H., TOMINSKI C.: Making parameter dependencies of time-series segmentation visually understandable. *Computer Graphics Forum 39*, 1 (2020), 607–622. doi:10.1111/cgf.13894. 2

[FFRE19] FERNANDES O., FREY S., REINA G., ERTL T.: Visual Representation of Region Transitions in Multi-dimensional Parameter Spaces. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference* (2019), Agus M., Corsini M., Pintus R., (Eds.), The Eurographics Association. doi:10.2312/stag.20191367. 1, 2

[FML16] FOFONOV A., MOLCHANOV V., LINSEN L.: Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE Transactions on Visualization and Computer Graphics 22*, 8 (2016), 2037–2050. doi:10.1109/TVCG.2015.2498554. 2

[FRM13] FRUTH J., ROUSTANT O., MUEHLENSTAEDT T.: The fanovagraph package: Visualization of interaction structures and construction of block-additive kriging models. *HAL preprint 00795229* (2013). URL: https://hal.archives-ouvertes.fr/hal-00795229. 2

[HEG*22] HEIMES K., EVERS M., GERRITS T., GYAWALI S., SINDEN D., PREUSSER T., LINSEN L.: Studying the Effect of Tissue Properties on Radiofrequency Ablation by Visual Simulation Ensemble Analysis. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (2022), Raidou R. G., Sommer B., Kuhlen T. W., Krone M., Schultz T., Wu H.-Y., (Eds.), The Eurographics Association. doi:10.2312/vcbm.20221187. 2

[HWG*20] HE W., WANG J., GUO H., WANG K.-C., SHEN H.-W., RAJ M., NASHED Y. S. G., PETERKA T.: InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations . *IEEE Transactions on Visualization & Computer Graphics 26*, 01 (2020), 23–33. doi:10.1109/TVCG.2019.2934312. 2

[LMW*17] LIU S., MALJOVEC D., WANG B., BREMER P.-T., PASCUCCI V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics 23*, 3 (2017), 1249–1268. doi:10.1109/TVCG.2016.2640960. 7

[LRHS14] LUBOSCHIK M., RYBACKI S., HAACK F., SCHULZ H.-J.: Supporting the integrated visual analysis of input parameters and simulation trajectories. *Computers & Graphics 39* (2014), 37–47. doi:10.1016/j.cag.2013.09.004. 2

[Nov19] NOVIKOV A.: PyClustering: Data mining library. *Journal of Open Source Software 4*, 36 (2019), 1230. doi:10.21105/joss.01230. 6

[OBJ16] OBERMAIER H., BENSEMA K., JOY K. I.: Visual trends analysis in time-varying ensembles. *IEEE Transactions on Visualization and Computer Graphics 22*, 10 (2016), 2331–2342. doi:10.1109/tvcg.2015.2507592. 2

[PBCR11] PRETORIUS A. J., BRAY M.-A., CARPENTER A. E., RUDDLE R. A.: Visualization of parameter space for image analysis. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2402–2411. doi:10.1109/tvcg.2011.253. 2

[PBF*16] PIANOSI F., BEVEN K., FREER J., HALL J. W., ROUGIER J., STEPHENSON D. B., WAGENER T.: Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling & Software 79* (2016), 214–232. doi:10.1016/j.envsoft.2016.02.008. 2

[PBK10] PIRINGER H., BERGER W., KRASSER J.: HyperMoVal: Interactive visual validation of regression models for real-time simulation. *Computer Graphics Forum 29*, 3 (2010), 983–992. doi:10.1111/j.1467-8659.2009.01684.x. 2

[PBM23] PICCOLOTTO N., BÖGL M., MIKSCH S.: Visual parameter space exploration in time and space. *Computer Graphics Forum 42*, 6 (2023), e14785. doi:10.1111/cgf.14785. 2

[SHB*14] SEDLMAIR M., HEINZL C., BRUCKNER S., PIRINGER H., MÖLLER T.: Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2161–2170. doi:10.1109/TVCG.2014.2346321. 1, 2

[SLS09] SPIEGEL M. R., LIPSCHUTZ S., SPELLMAN D.: *Vector Analysis*, 2nd ed. Schaum's Outlines. McGraw Hill, 2009. 2

[Sob01] SOBOL I.: Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation 55*, 1 (2001), 271–280. The Second IMACS Seminar on Monte Carlo Methods. doi:10.1016/S0378-4754(00)00270-6. 1

[SRA*08] SALTELLI A., RATTO M., ANDRES T., CAMPOLONGO F., CARIBONI J., GATELLI D., SAISANA M., TARANTOLA S.: *Global Sensitivity Analysis. The Primer*. John Wiley & Sons, Ltd, 2008. 1, 2

[TFE22] TKACHEV G., FREY S., ERTL T.: S4: Self-Supervised learning of Spatiotemporal Similarity. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2022), 4713–4727. doi:10.1109/TVCG.2021.3101418. 1

[TWMSK18] TORSNEY-WEIR T., MÖLLER T., SEDLMAIR M., KIRBY R. M.: Hypersliceplorer: Interactive visualization of shapes in multiple dimensions. *Computer Graphics Forum 37*, 3 (2018), 229–240. doi:10.1111/cgf.13415. 2

[TWSM*11] TORSNEY-WEIR T., SAAD A., MÖLLER T., HEGE H., WEBER B., VERBAVATZ J., BERGNER S.: Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1892–1901. doi:10.1109/tvcg.2011.248. 2

[USKD12] UNGER A., SCHULTE S., KLEMANN V., DRANSCH D.: A visual analysis concept for the validation of geoscientific simulation models. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2216–2225. doi:10.1109/TVCG.2012.190. 2

[VGO*20] VIRTANEN P., GOMMERS R., OLIPHANT T. E., HABERLAND M., REDDY T., COURNAPEAU D., BUROVSKI E., PETERSON P., WECKESSER W., BRIGHT J., VAN DER WALT S. J., BRETT M., WILSON J., MILLMAN K. J., MAYOROV N., NELSON A. R. J., JONES E., KERN R., LARSON E., CAREY C. J., POLAT İ., FENG Y., MOORE E. W., VANDERPLAS J., LAXALDE D., PERKTOLD J., CIMRMAN R., HENRIKSEN I., QUINTERO E. A., HARRIS C. R., ARCHIBALD A. M., RIBEIRO A. H., PEDREGOSA F., VAN MULBREGT P., SCIPY 1.0 CONTRIBUTORS: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272. doi:10.1038/s41592-019-0686-2. 6

[vOVR23] VAN ONZENOODT C., VÁZQUEZ P.-P., ROPINSKI T.: Out of the plane: Flower versus star glyphs to support high-dimensional exploration in two-dimensional embeddings. *IEEE Transactions on Visualization and Computer Graphics 29*, 12 (2023), 5468–5482. doi:10.1109/TVCG.2022.3216919. 7

[vWvL93] VAN WIJK J., VAN LIERE R.: Hyperslice. In *Proceedings Visualization '93* (1993), pp. 119–125. doi:10.1109/VISUAL.1993.398859. 2

[WLSL17] WANG J., LIU X., SHEN H.-W., LIN G.: Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 81–90. doi:10.1109/tvcg.2016.2598830. 2

[YBRP21] YANG H., BALLESTER-RIPOLL R., PAJAROLA R.: SenVis: Interactive tensor-based sensitivity visualization. *Computer Graphics Forum 40*, 3 (2021), 275–286. doi:10.1111/cgf.14306. 2