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

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

# Coupling Guidance and Progressiveness in Visual Analytics

I. Pérez-Messina[1], M. Angelini[2,3], D. Ceneda[1,4], C. Tominski[5], and S. Miksch[1]

[1]TU Wien, Institute of Visual Computing and Human-Centered Technology, Austria
[2]Sapienza University of Rome, Italy, [3]Link University of Rome, Italy
[4]TU Eindhoven, The Netherlands
[5]University of Rostock, Institute for Visual & Analytic Computing, Germany

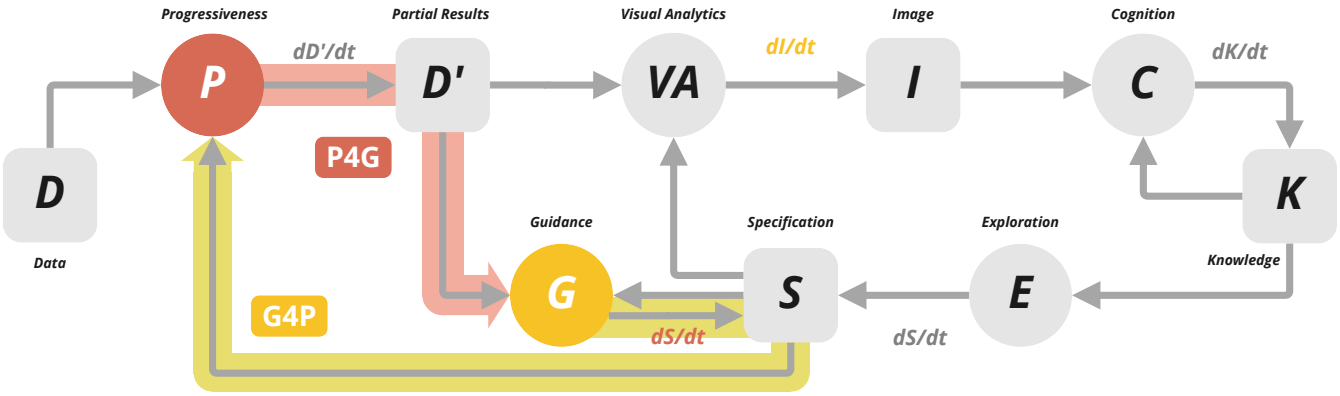


**Figure 1:** ***Guidance-enhanced and progressive VA model****—Extending van Wijk's model of visualization [VW06], we include a guidance agent G (based on the extension already proposed by Ceneda et al. [CGM*17]) and progressiveness agent P, both system-side processes that can be controlled through the specification S. P produces partial results $D'$ through a progression $dD'/dt$. There are two cases addressed in this figure: In* ***G4P*** *(yellow), G provides guidance for the steering of P, while P mediates between data D and the rest of the system, producing also the visualization progression $dI/dt$. In* ***P4G*** *(red), P only mediates between D and G. G behaves progressively in this case inducing the guidance progression $dS/dt$, while D outputs directly to VA.*

**Abstract**

*Data size and complexity in Visual Analytics (VA) pose significant challenges for VA systems and VA users. Two recent developments address these challenges: progressive VA (PVA) and guidance for VA (GVA). Both share the goal of supporting the analysis flow. PVA primarily considers the system perspective and incrementally generates partial results during long computations to avoid an unresponsive VA system. GVA is primarily concerned with the user perspective and strives to mitigate knowledge gaps during VA activities to prevent the analysis from stalling. Although PVA and GVA share the same goal, it has not yet been studied how PVA and GVA can join forces to achieve it. Our paper investigates this in detail. We structure our research around two questions: How can guidance enhance PVA and how can progressiveness enhance GVA? This leads to two main themes: Guidance for Progressiveness (G4P) and Progressiveness for Guidance (P4G). By exploring both themes, we arrive at a conceptual model of how progressiveness and guidance can work together. We illustrate the practical value of our theoretical considerations in two case studies of G4P and P4G.*

**CCS Concepts**

• ***Human-centered computing*** → ***Visualization theory, concepts and paradigms;*** *Visualization design and evaluation methods;*

## 1. Introduction

Guidance and progressiveness are two core approaches in Visual Analytics (VA). They address a common challenge: sustaining and helping users to keep the flow when solving complex, data-intensive analytical tasks. While each approach achieves this from a different perspective—progressiveness by delivering partial results during long-running computations and guidance by bridging users' knowledge gaps—they hold significant potential to complement one another in complex analysis scenarios. However, despite their common goal, the design space arising in the interstice between guidance and progressiveness remains unexplored.

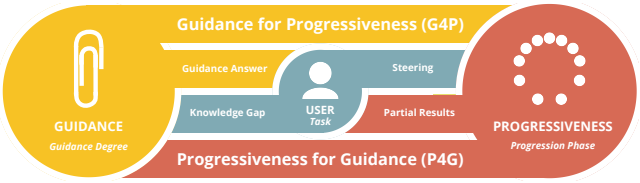


**Figure 2:** *The vision for our work is to investigate the coupling of guidance and progressiveness, and the implications for the user.*

Progressive VA (PVA) addresses the challenges posed by large datasets and complex analytical computations that can cause long waits for results to be generated, blocking any analytical progress in the meantime. PVA solves this problem by incrementally delivering meaningful intermediate results based on data and process chunking [ASSS18, FFNS19, UAF*23, FFS24]. Thanks to the incremental nature of PVA, analytical discourse is not blocked and analytical progress is possible earlier. However, as intermediate results do not show the full picture of the data, care must be taken not to steer (i.e., change the focus of the result generation) the analysis toward potentially useless results or even introduce bias into the analysis, thereby diminishing the effectiveness of PVA. Guidance seems to be a natural fit to mitigate these PVA problems by assisting users in steering the progressive analysis.

Guidance-enhanced VA (GVA), on the other hand, addresses the challenges posed by complex analysis tasks and user *knowledge gaps*, which may cause the analysis to stall. GVA aims to assist in such situations by providing user-, data-, and task-aware answers that enhance user understanding, redirect user actions with interactive methods, or automatically enact analytical pathways [CGM*17]. Being "timely" is a key characteristic that good guidance must possess [CAA*20, CCEA*23]. However, this quality is at risk when GVA faces large datasets or requires extensive processing. Interestingly, such considerations are exactly the ones that motivated progressive approaches in the first place, and it seems logical to use progressiveness to make GVA timely and relevant.

These considerations led us to explore the interplay between PVA and GVA, involving questions such as: How can guidance be produced when the data are large? How can guidance be effectively offered while progressive methods generate increasingly mature yet still partial results? Which design considerations should be followed while merging the two approaches? Which drawbacks must be paid attention to?

Addressing these questions leads to two research perspectives : *progressiveness for guidance* (P4G) and *guidance for progressiveness* (G4P). Our work examines the defining characteristics of P4G and G4P, their interaction with users, and the potential for a unified design space at their intersection. Specifically, we make the following contributions:

- A conceptual model of the interplay of guidance and progressiveness in VA based on guidance and user tasks as well as guidance degrees and progression phases (Sect. 2);
- A conceptualization of guidance for progressiveness (G4P; Sect. 3), and of progressiveness for guidance (P4G; Sect. 4),
- Two case studies illustrating how to apply our conceptualizations in each scenario.

## 2. Background: User, Guidance, and Progressiveness

Before considering the interplay of PVA and GVA, we first have to introduce what is known about them individually, their characterizations, and their relations to the user. We envision progressiveness, guidance, and the user as agents with their own internal state and free interaction possibilities with the other two as shown in Fig. 2. A more systemic view is provided in Fig. 1, where we extend van Wijk's model of visualization [VW06] with the processes *G* for guidance and *P* for progressiveness. Two scenarios are distinguished in this figure: G4P, representing a guidance-enhanced PVA system, and P4G, representing a VA system enhanced with progressive guidance.

### 2.1. User

Users take a central role in VA. They can be characterized according to their domain, their expertise in their domain, their expertise in VA, their physical environment, and their skills and (dis)abilities, and various factors can affect their performance in VA. Such characterizations are state-oriented, proposing the user as a kind of receptacle with certain knowledge, which is useful for arriving at specific designs [MA14]. There is also a process-oriented view of the user, where the human plays an active role as a knowledge-producing reasoning agent [VW06, SSS*14]. We adopt this process-oriented view of the user, as we are concerned with the interactions with the other entities in our framework. An abstract task-based model provides the grounding for our conceptualization of user-side interactions.

**User tasks.** Tasks are abstractions for classifying user analytical behavior or the user intent. Several task taxonomies and typologies have been proposed for VA. Brehmer and Munzner's multi-level typology of abstract visualization tasks bridges the knowledgeability of users with their intent and allows creating modular task diagrams to represent complex task workflows [BM13]. Within their *why* dimension, the *search* level encapsulates four different tasks—explore, locate, browse, and lookup—whose scope encompasses all the possibilities of user knowledge regarding the task's intended target and its location. This task abstraction is particularly useful because it provides us with a systematic understanding of the user's **knowledge gap**, which in turn can inform the guidance [PMCEA*22]. Gotz & Wen [GW09], for example, identified four action patterns (repeating series of interactions) which can be used to provide suggestions of visualizations that facilitate user tasks based on their analytical behavior. While the influence of progressiveness on user tasks has been suggested [MSA*19], no tight connection between the tasks and the user's **steering** in a progression has been described so far. We expand on these concepts in the following and investigate this connection in Sect. 3.

### 2.2. Guidance

The basic function of guidance is to provide **answers** to user **knowledge gaps** during a VA session [SSMT13, CGM*17]. Guidance can deliver its answers in different ways, and their main characterization is by "strength" or **guidance degree**, which we consider here as the internal free variable of the guidance. Broader definitions of guidance include adaptive capabilities [SJB*21] and exclude onboarding [CGM*18]. Here we focus only on the aforementioned basic concepts.

**Guidance Degree.** There are three guidance degrees, from the least

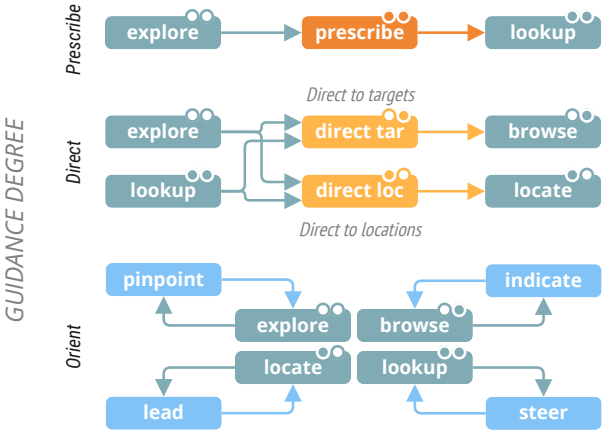


**Figure 3:** ***Guidance tasks and user tasks****—Guidance tasks and their relation with user tasks (gray-like). Orienting guidance tasks (light blue) do not change the current user task but provide cues and receive feedback from it. Task-transitioning guidance tasks (orange) change the user task to a different one by closing a location-(●|○) and/or target-related (○|●) knowledge gap. Adapted from Pérez-Messina et al. [PMCEA*22]*

to the most constraining to user freedom: orienting, directing, and prescribing [CGM*17]. While orienting only provides cues through an extra information layer, directing and prescribing offer and undertake alternative tasks and propose new targets or paths for analysis, leading to the distinction of task-preserving and task-transitioning guidance [PMCM24]. The decision what degree of guidance to provide when a knowledge gap is identified is embedded in the design of the guidance.

**Knowledge Gap.** Guidance identifies and addresses (through system initiative) user knowledge gaps. Knowledge gaps have two meanings in the literature: They either refer to the unanswered questions that lead to hypotheses driving the analysis [SSK*15], or to any kind of lack of knowledge that may hinder analytical progress [CGM*17]. Driven by the second (negative) meaning of knowledge gap, guidance aims to reduce knowledge gaps so that the analysis can continue. The knowledge gap is thus the main information that is conveyed to the guidance about the state of the user, triggering guidance behavior resulting in a guidance answer being in turn conveyed to the user.

**Guidance answer.** Based on a knowledge gap and a specified guidance degree, a guidance answer is produced. Analogous to user tasks, guidance behavior can also be abstracted as tasks that interact with the user and system. A typology of guidance tasks [PMCEA*22,PMCM24] distilled this idea into seven knowledge gap- and guidance degree-dependent *guidance tasks*, following the conventions of Brehmer and Munzner's framework. Fig. 3 shows the relations between the relevant user tasks (explore, browse, locate, lookup) and guidance tasks (pinpoint, indicate, lead, steer, direct to targets/locations, prescribe).

### 2.3. Progressiveness

In contrast to so-called classic, monolithic, or blocking VA, PVA progressively loads the data into view or processes them in small enough chunks so that meaningful (partial) results can be observed from the start and the system remains interactive at all times [PTMB09,FP16]. Progressiveness is thus defined by this chunked processing (previously studied by Mühlbacher et al. [MPG*14]) producing partial results and inducing **progression phases**, and the allowances it gives to the user for **steering**.

**Progression phase.** In PVA, the data analysis unfolds as a series of *progressions*, in which data are progressively loaded, processed, and visualized. Usually, there is one progression at a time that the analyst pays attention to, but, there can also be many running in parallel (as there can be several guidance tasks active at the same time). It is normal that interaction during progressive analysis is fast-paced, and a progression might not even reach its completion before the user draws insights, makes decisions, or goes on to a different task.

A progression is divided into three phases [ASSS18]. ***Phase I: Early Partial Results*** represents an initial, highly uncertain stage where data estimates improve rapidly, allowing analysts to detect potential errors, adjust parameters, or restart their analysis as they assess the processing strategy's suitability. Insights remain tentative here, as this phase mainly serves for calibration rather than decision-making. In ***Phase II: Mature Partial Results***, clearer patterns emerge, and analysts begin to develop early insights with increasing confidence. Yet, there is a trade-off between waiting for greater certainty and acting on preliminary findings. Here, the system plays a critical role in balancing time with insight value to suit the analyst's needs. Finally, in ***Phase III: Definitive Partial Results***, the analysis process converges, as uncertainty stabilizes at its minimum, and further incremental changes become negligible. By this point, the analyst has sufficient understanding to make informed decisions, and the remaining progression adds limited value beyond final confirmation [ASSS18,FFNS19,SSK*15].

Transitions between these phases do not take place at an exact point in time unless defined in such a way, but have a semantic dimension that depends on the behavior of the progression and how effectively (according to the task) the data are being sampled. Nonetheless, constraints such as working memory and visual display resolution impose limits on the amount of data that can be effectively presented, making it impossible to fully eliminate uncertainty. To address this, transient VA has been proposed as a more general form of PVA, where data are not only progressively added but also concurrently removed [SW24].

**Steering.** Computational steering is the interactive control of the progression during its execution. Techniques such as Sherpa [CKBE19] and steering-by-example [HASS22] have been developed to make steering more intuitive and effective. Quality indicators are relevant for informed steering [AMSS19], and managing progressive sampling for big data is still an open technical problem [HS23].

### 2.4. Summary

Although GVA and PVA would mutually benefit each other, the state of the art does not show any intersection between them. This lack of integration means that both approaches operate with inherent limitations that could be alleviated through their combination. On the one hand, PVA ensures interactivity by incrementally refining results, but it lacks explicit mechanisms to help users navigate the evolving information space, often requiring them to interpret and react to partially available data on their own. On the other hand, GVA provides structured support for decision-making and analysis but assumes a stable dataset or analytical state, making it less suitable for dynamic or evolving scenarios where information arrives progressively.

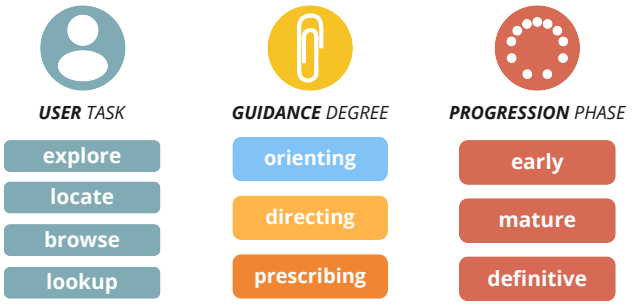


**Figure 4:** *Dimensions of the agents in our framework.*

Our approach to the problem of coupling progressiveness and guidance is inspired by an agent-based framework for VA [MGG*23] that models human and machine processes as interacting agents with distinct reasoning capabilities, as concretely realized in Sperrle et al. [SSKEA21]. We extend this perspective to incorporate progressiveness and guidance as active agents within VA. We conceptualize Progressive Visual Analytics (PVA) as P+VA, where a progressiveness agent *P* incrementally refines computational results, and Guided Visual Analytics (GVA) as G+VA, where a guidance agent *G* provides support in navigating the analysis process.

Fig. 1 shows how *P* functions as an intermediary between the data and the imaging process, allowing the user to steer the progressive refinement of results. Meanwhile, *G* mediates between the data and the visualization specification, shaping how information is surfaced and directing attention to relevant elements. Modeling *P* and *G* as agents highlights their ability to take analytical actions within the system—actions that inherently carry computational or cognitive costs. The integration of *P* and *G* allows these costs to be redistributed more efficiently: guidance can help steer progression efforts and assist in understanding partial results (G4P), and progressiveness can help generate guidance early on and iteratively refine its suggestions (P4G).

Next, we propose a design space (following [KK17]) to explore the interaction between *P* and *G*. This design space defines how guidance contributes to progressiveness (G4P) and vice versa (P4G) through three primary dimensions: (i) user task, (ii) guidance degree, and (iii) progression phase. These dimensions and their possible values are illustrated in Fig. 4.

## 3. G4P: Guidance for Progressiveness

This section discusses how guidance can be added to PVA. We describe the dimensions *user task* (what task is to be supported with guidance), *guidance degree* (how strong is the guidance provided), and *progression phase* (in which moment of the progression the task is performed). Going in the G4P direction, we first characterize user tasks inside the progression and then present how guidance can enhance and enrich them.

### 3.1. User tasks in the progression

User tasks (see Fig. 3) have so far only been defined for the non-progressive case [BM13, PMCEA*22]. Here, we expand their definition to progressive environments. Our definition is based on the notions of *target*, *location*, and *path*. Similar to a computer file system, a location would be a folder, a target would be a file, and a path would be a concatenation of locations plus targets.

Depending on a given path, users may perform different tasks. According to [BM13], user tasks are defined by the knowledge a user has about paths, in particular, whether the target is known −|● or unknown −|○ and whether the location of the target is known ●|− or unknown ○|−. The four possible combinations lead to four user (search) tasks: explore ○|○, browse ●|○, locate ○|●, and lookup ●|●. To perform a task means to operate on known targets and locations and output the unknown parts, which then serve as input for a subsequent task. Note that for lookup, the path is fully known, but the user may still need to see what is "inside" the target.

In a progressive environment, target and location have a different realization because the progression may not have produced them at a given time. For example, a user may browse a location where targets have not yet been loaded completely, and it is unknown whether or not a target will eventually appear until the progression ends. This reveals two interesting aspects of user tasks in progressive environments: task variability, and blocking and steerable tasks, on which we expand next. In this context, we use the term *immaturity* to refer to the fact that information is still missing due to an unfinished progression, and *maturity* as its opposite.

**Task variability** A PVA system produces partial results with different degrees of maturity (early, mature, and definitive) in three phases. The question now is how tasks are affected by these phases. It is commonly accepted that tasks are constantly evolving during data exploration as users incrementally develop understanding. It is only natural that PVA will have even more fluctuating tasks, because new data may constantly arrive. Moreover, the tasks' input needs to be updated with each new phase, meaning that phase transitions also entail task transitions. In contrast to the non-progressive case, where task transitions occur only when a task is solved, the progressive phase transitions do affect task transitions as the context for subsequent tasks changes.

**Blocking and steerable tasks** An interesting aspect of tasks is whether they are *blocking*, meaning they cannot be resolved until the progression reaches a certain phase or maturity. Although one of the benefits of PVA is to provide something to work with from the start, this does not satisfy every task from the beginning. The more a-priori knowledge users have about the target, the more mature the progression needs to be to satisfy the demands of the task. As a result, tasks with a known target (−|●) are *blocking*, even in progressive environments, as the necessary input for the task needs to be produced before it can be solved. On the other hand, tasks with more a-priori knowledge about the location (●|−) can leverage steering to benefit from progressive environments. We refer to such tasks as *steerable*. By steering the progression towards a more localized context, users can direct the system toward the desired region, reaching maturity faster.

Next, we discuss the characteristics of the four user tasks in progressive environments in more detail.

**Explore** ○|○ tasks are usually the first to be performed to generate hypotheses on previously unseen data. In PVA, users tend not to wait until all computations are complete before drawing insights and making their next move, even when completion time is in the order of

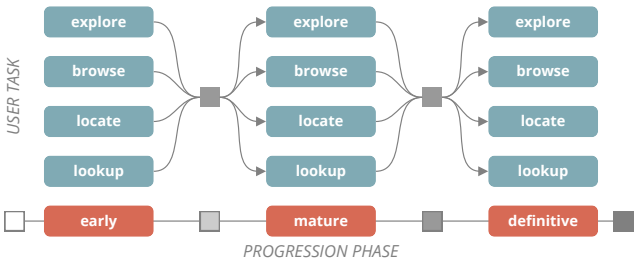


**Figure 5:** ***User tasks in progressive environments***— *Users can perform any task at any moment and are free to move between them (represented by the connector lines), but the progression (bottom axis) forces a task (as a function of their input) to be "updated" in each stage of the progression due to the change of context.*

10s [ZGC*17]. This suggests that exploratory tasks are less affected by immaturity, because users have little knowledge and assumptions about how to solve a task or what path to take anyway. Although overtrust in incomplete results can lead to misinterpretation [SSK*15], it is clear that exploration tasks are not blocking and can be performed over partial results. They are also not steerable because there is too little knowledge to decide where to steer.

**Browse** ●|○ tasks occur when users inspect different data elements to find one that best suits their needs even if their needs are not defined a priori (i.e., only the prior part of the path is known). As the target of the task is not known to the user, browsing is not blocking and can be performed from the start of the progression as long as the desired locations (i.e., the data portions containing the desired characteristics) have been loaded. Browsing is also a steerable task because users know the locations where to steer.

**Locate** ○|● tasks occur when the user needs to search for a desired target, while its location remains unknown (i.e., only the latter part of the path is known). It has been shown that locate tasks are complex enough so that user personality and completion time can be predicted from interaction logs [BOZ*14]. In progressive scenarios, however, the sought element may not have been loaded at the time the user wants to locate it. In such cases, locate is a blocking task, as a certain maturity of the progression is necessary for resolving the task. Again, as there is no knowledge about the location, locate tasks are not steerable.

**Lookup** ●|● tasks are the easiest for users to perform as they act on complete knowledge of their target and location. For lookup to be effectively performed, we assume that the visualization needs to be crisp, which only happens in the later stages of the progression, which makes lookup a blocking task. However, as the user knows the location, the lookup is steerable. The chunking order, which is mainly determined by the sampling strategy, can play an important role in speeding up lookup times.

Note that the state of maturity of the progression does not need to be global. The more steerable and blocking a task is, the more effort the progression can and should spend on producing and refining results locally to save time and resources. Guidance, in particular task-transitioning guidance, should therefore effectively guide users toward local contexts.

### 3.2. Guidance in the progression

Supporting user tasks in PVA systems with guidance is not straightforward, as PVA comes with its own challenges and introduces unique knowledge gaps to the user. Firstly, the guidance agent can operate only on a partial view of the data during early and partial result stages of the PVA workflow (as per Fig. 1 G4P). Secondly, the user is faced with some tasks that are blocking while others are steerable. Although both the guidance and the user are subject to the same restriction (i.e., the immaturity), they have different capabilities, so the guidance can still be able to provide answers that the user may need. However, if applied in a naive way without considering the progressive nature of PVA, the answers provided by the guidance will change as the progression advances and confuse the user while slowing down the progression. A unique opportunity comes from guidance supporting the user in *steering* the PVA process, i.e., providing guidance during and regarding the progression itself.

Guidance can accelerate the development of the progression by helping users steer it, e.g., by suggesting focus areas and actions that users would otherwise have to discover and input to the system manually. Thus, the guidance does not make the progression faster. Instead, the pace of analysis gets more efficient by allowing the user and the system to focus attention and thus jump to the next phase more quickly. That is, guidance supports the semantic dimension of the progression.

The three degrees of guidance (i.e., orienting, directing, prescribing [CGM*17]) instantiate increasing levels of constraint on user freedom while providing increasingly precise answers to a location/target knowledge gap. We propose here that task-preserving and task-transitioning guidance play different roles in progressive environments. While task-preserving guidance can assist the user throughout and along the progression until its completion (e.g., help make a better-informed decision for early termination), task-transitioning guidance accelerates the progression by "deepening" (i.e., taking the user to a task that can be resolved in a more constrained space). Following this idea, Fig. 6 expands Fig. 5 by adding guidance tasks from Fig. 3 along and within the progression. In the following, we explain Fig. 6 and describe how each guidance task can help in progressive environments.

**Task-preserving guidance** There are four task-preserving guidance tasks (steer, lead, indicate, pinpoint), which correspond to the manifestation of orienting guidance when supporting user tasks [PMCEA*22]. As orienting guidance only provides cues of supportive information (a partial answer to a knowledge gap, but not a determined location or target) it is said to be task-preserving (i.e., it does not constrain user freedom). In the context of PVA, this means that orienting guidance can support user tasks continuously during the progression, without explicit breaks or branching the analysis path. In Fig. 6, this is illustrated by placing all guidance tasks corresponding to orienting guidance at the very start of the progression and extending their support for the individual user tasks throughout the progression.

**Steer** supports data exploration by highlighting promising areas in partial results. This helps users generate hypotheses and stay oriented within an evolving data space. For example, it might cue clusters of outliers or highlight regions exhibiting distinct evolution patterns, enabling users to focus their exploration more effectively.

**Indicate** assists browsing by cueing users about relevant or newly appearing elements in smaller regions. This helps users stay aware of emerging details within the dataset. For example, the system

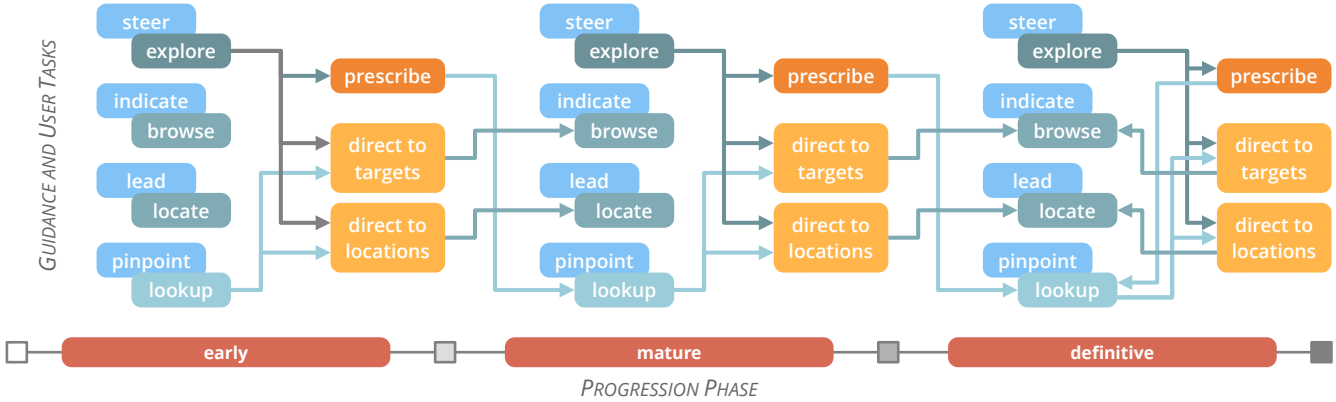


**Figure 6:** ***Guidance-enhanced progression****—While task-preserving guidance tasks (blue) follow along the progression and are continuously updated, task-transitioning guidance tasks (orange) are updated at specific moments and accelerate transition phases by taking the analysis to more local contexts.*

might notify users of similar data points appearing in a newly loaded segment, ensuring they don't miss crucial updates.

**Lead** facilitates locating tasks by directing user attention toward likely targets or patterns within evolving datasets. This minimizes idle time by making the search process more efficient. For example, the system might cue regions that are most likely to contain a specific target based on the available data.

**Pinpoint** enhances focus and accuracy by highlighting user-identified paths. It helps in lookup tasks by providing context for a hovered item or highlighting a cluster. For example, the system might cue the relevant context for a particular data point or emphasize a specific pattern within the dataset.

**Task-transitioning guidance** refers to directing and prescribing guidance degrees. They assist users by suggesting actionable paths (directing) or enacting them (prescribing), effectively supporting the steering of the progressive data analysis. As shown in Fig. 6, this can happen in three different ways: *direct to targets, direct to locations, and prescribe*.

**Direct to targets** •|○ generates the prior part of a path as the guidance answer, i.e., a range of data characteristics that are deemed interesting. This translates into a set of partial results (targets) that the user can browse. By browsing through the suggested partial results, the user's search is constrained to a smaller section of the data space, on which resources can be focused. Note that a directing guidance task provided to a lookup is considered *diverging* guidance [PMCEA*22], which "overrides" the user's known path in order to show alternatives, e.g., take them to more mature areas of the progression.

**Direct to locations** ○|• is the complement of direct to targets, generating the latter part of a path as an answer. We interpret a target without a location as a set of independent *actions* leading to a new state in the visualization where users can locate what they are searching for. As an action within a progression involves a change of parameters of the view or running algorithm (e.g., the sampling strategy), this information is directly or indirectly linked to a constraint in subsequent progressive steps. Depending on the affordances of the system, this could also mean an early termination of the progression.

**Prescribe** •|• provides a full path as an answer and so needs no user involvement. It unites the effect of both directing tasks and can propel a user task to completion along with the progression. In other words, once the given user task is resolved, the remaining progression becomes irrelevant. Guidance in this case is automatically steering the progressive agent, as it can communicate directly to it through its access to the specification (see G, S, and P in Fig. 1).

These guidance tasks require data to be present in the current state of the progression for guidance to make an informed suggestion. Task-transitioning guidance in a progression must thus come second to orienting guidance, which is task-preserving. Fig. 6 makes this clear by placing task-transitioning guidance in between early and mature partial result stages of the progression, producing a "breach" in the progression line. This breach represents the work that is performed by guidance in advancing the progression to a mature state by switching the user task to one for which the locally available data are in a mature stage. This way, guidance can shorten the progression and lower costs. This is repeated between mature and definitive partial results stages. However, the guidance answers provided are not the same because the context has already changed and the maturity of the analysis is higher. Finally, if the progression ends (which may not be the case due to early termination or infeasibility), guidance can resume work as it would for a non-progressive system.

**In summary**, the benefit of guidance in PVA is twofold. First, it can allow the system to enable reliable higher-level user tasks at a fixed quality phase of the progression (e.g., early partial results). By guiding the user toward the local context of the data space where these higher-level user tasks are reliable, guidance makes the progression more effective. Second, guidance can accelerate the transition between global quality progression phases by supporting computational steering where available. Through the use of directing and prescribing guidance, the user can make the progression more efficient by reducing the duration of a progression phase, avoiding the generation of unnecessary data, and prioritizing needed ones. The guidance effectively serves as a means for steering. Yet, the majority of today's PVA systems only provide the steering mechanism but rarely the information to target it effectively. Future PVA systems can close this gap by considering our G4P ideas.

### 3.3. Case Study: MDSteer

MDSteer [WM04] is a steerable system for a MultiDimensional Scaling (MDS) algorithm. It supports exploration by progressively visualizing high-dimensional data in a 2D scatterplot where users can select

regions at runtime to focus on interesting locations and arrive much faster at the desired analysis targets. Here, we apply G4P to arrive at a guidance-enhanced PVA solution.

**Overview** MDSteer relies on both data chunking (by sampling only a few new points at each iteration) and process chunking (by iteratively adjusting positions by a small amount to reduce a stress function). It allows users not only to get a quick partial overview of the structure of a dataset with thousands of points and hundreds of dimensions from the very start, but also to focus the computations on automatically defined subregions of the space (bins), thus making it steerable.

**VA progression** In MDSteer, the progression starts at the beginning of the session by randomly sampling a small amount of data points and iteratively running MDS until their positions stabilize, then doing a rebinning of the space if necessary. Without steering, the system will repeat this process for the full dataset and complete the progression. If, however, at any point of the progression, the user selects one or more bins, the rest will be turned inactive, and the progression will only run on these subregions, thus maturing much faster on a local scale. If we take the ratio of points visualized as a rough measure of maturity, we must consider that with steering, maturity will vary across regions until the global progression is complete.

**Applying the framework** Guidance or visual indicators for progression quality (see Angelini et al. [AMSS19]) are not included in MDSteer. We propose that such a progressive approach for exploratory analysis would benefit from considering guidance-enhanced steering in the following way. We focus on guidance for exploration tasks, as this is the main task supported by the system. That is, we assume that the analysis goal is to get a rich overview of the dataset to foster hypothesis generation.

**Orienting** According to our framework, orienting should support exploration throughout the progression, steering the user to the most interesting regions without affecting interaction. Assuming that there is an indicator of interest, such as tension (difference between high- and low-dimensional distance), relative stability (rate of change per iteration), or other quality indicators, highlighting locations (bins) or targets (elements) can provide the user with cues that translate into the different task-preserving guidance tasks. Ultimately, orienting can show the relative maturity of each bin to help the user keep track of which areas have been more explored (an indicator that would become irrelevant at the end of the progression). It may also suggest initial areas for applying the steering mechanism in a more objective way, taking into account these indicators and helping answer the question "Which area should be explored first through steering?" making the exploration more efficient.

**Directing** As a middle-ground between orienting and prescribing, guidance can **direct to targets** that show outlier qualities in certain dimensions (using information from the raw data) and **direct to locations** (bins) that contain particular values in average. As the data can only be observed from partial results, it is reasonable that directing is calculated after phase I, when an important part of the data is already there. This can happen at a global level (according to classic progressiveness) or by having used orienting guidance for local areas. In this scenario, directing guidance may be exploited for higher-level user tasks, like browsing or locating, enabling more complex analyses at a fixed phase of the progression. This step can help answer questions like "Is this local area useful for answering my analysis task?" and "Can I terminate the progression early because the desired analysis results or the progression run are not useful for solving the analysis task?" at an earlier progression phase, saving time and making the workflow more effective.

**Prescribing** The authors of MDSteer suggest future work on a "self-driving" mode where the system automatically selects new areas for development when the work on the current ones is finished. This fits the description of prescriptive guidance for exploratory purposes. If implemented, it may be tailored considering, for example, the final goal of the analysis. Depending on the user's current task, the system may prescribe the next area to explore, being triggered whenever the milestone of a local progression ends. Which bin is automatically chosen next could also be based on the quality indicators shown via orienting and directing guidance, and previous user-selected areas. This step can reduce the time needed for progressive phase III, helping in quickly converging to stable results (avoiding additional costly steering). It can also support confirmation through quick what-if analyses, in which the prescribing guidance provides the top-n areas worth focusing on.

We see that G4P can make MDSteer more effective by constantly enhancing the user's mental map, steering analysis towards potentially fruitful places, and reducing idle time.

## 4. P4G: Progressiveness for Guidance

In this section, we examine how progressiveness can support guidance (P4G) and strive to arrive at a conceptualization of *progressive guidance*. While the purpose of guidance is to support users in navigating analytical processes and bridging knowledge gaps, P4G ensures that the guidance can operate even when the data are large, processing is extensive, and computations are incomplete. First, we will put guidance in the context of progressive approaches, and then discuss how progressive guidance can be delivered at different guidance degrees and at different levels of maturity.

### 4.1. Progressive Chunking for Guidance

We first need to differentiate between *chunking for VA* and *chunking for guidance* (each corresponding to a case depicted in Figs. 1 and 2, namely, G4P and P4G):

**Chunking for VA** refers to the already known (data, process, custom) chunking in PVA systems. We have already seen how guidance can be framed inside a PVA environment, where the user tasks are the main addressee of the progression. In this case, guidance tasks behave classically, being calculated in one step only when they are called and with the currently available data. This scenario, although viable in theory, may still pose practical problems for guidance (such as the sheer amount of data to be analyzed), and progressiveness for guidance would also need to be applied.

Depending on if the progression occurs as the loading of the data, or the data are there from the beginning but need to be processed, it is said that PVA focuses on the *data space* or the *algorithmic space* [MSA*19]. These two approaches, named data chunking and process chunking, have been extensively surveyed [UAF*23]. However, data chunking and process chunking only represent the most simple cases. There are

situations in which it is necessary to take a more holistic approach to progressive design, in which algorithm and data space are both chunked concurrently, as in a "custom chunking" approach. Custom chunking has not yet been clearly characterized and defined. Thus, three types of progressive processing (chunking) are distinguished in PVA: *data chunking* (data is progressively added), *process chunking* (data is progressively processed), and *custom chunking* (combining data and process chunking in novel ways) [ASSS18, UAF*23].

**Chunking for guidance** refers to the scenario we analyze in this section, where the partial results are only provided for the guidance (see P4G in Fig. 1), while the VA system uses a classical non-progressive visualization. This leads to the question of if and how existing knowledge on chunking for visualization can be applied to guidance.

Although the VA chunking categories do not consider guidance, guidance is defined as a *process* [CGM*17], entailing that, just like the visualization process, it can be subject to different types of chunking. This is readily visible in Fig. 1 P4G, where $G$ can obtain the partial results it needs to provide answers from $P$ by controlling it through the specification, while the full dataset is still conveyed directly to the visualization. From this model it can be inferred that if the data is small enough such that non-progressive VA can handle it, then data chunking is most probably not needed for progressive guidance. Process chunking is then most relevant in P4G.

Mühlbacher et al. [MPG*14] define four strategies for algorithm chunking: (S1) *data subsetting*, making passes over an increasing subset of the data; (S2) *complexity selection*, increasing the number of processed dimensions in each pass; (S3) *divide and combine*, chunking the data and then combining partial results; and (S4) *dependent subdivision*, sequentially dependent steps where the output of one iteration is the input for the next. Here, S1 and S3 correspond to data chunking and S2 and S4 to process chunking.

Which chunking strategy is better suited for achieving progressive guidance depends on the data and task to be enhanced. Data subsetting has the problem that, as it progresses, it passes over bigger and bigger chunks of data until it passes over the complete dataset, which might be unfeasible given large datasets, and it is better suited when providing guidance in a smaller, localized context, thus making it better suited for an indicate task. Complexity selection works by progressively adding dimensions of the data to the answer, which can also be subject to the constraint mentioned before, but in the case the guidance model knows something about what the user is looking for this strategy can focus on some dimensions of the data and be well-suited for a lead task. Both divide and combine and dependent subdivision strategies divide the workload into simpler and independent steps, which makes them very powerful. However, when dealing with highly-connected data, divide and conquer can miss part of the relevant information. Dependent subdivision, on the other hand, considers at each step the input of the user, making it the most flexible for iterative refinement of a solution.

### 4.2. Guidance Degrees in the Context of Progression

Having a progressive process delivering partial results to the guidance agent through an iterative algorithm is only a first ingredient for defining progressive guidance. Feeding the guidance partial results means that the guidance answers themselves must be generated incrementally and

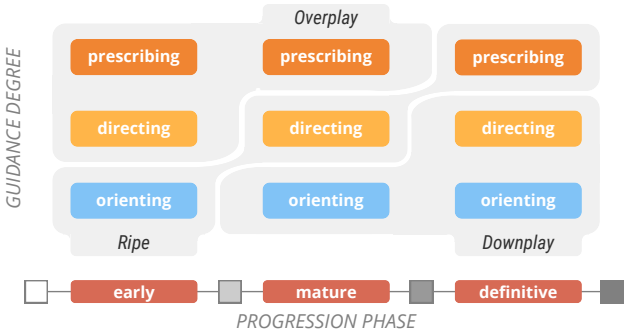


**Figure 7:** ***Guidance degrees in relation to guidance progression phases****—Due to the maturity requirements of each guidance degree, they can be either ripe for provision, or represent a downplay or an overplay at different progression phases. This does not rule out any possibility, but highlights different roles for guidance.*

go through phases of increasing maturity (early, mature, and definitive) as more and more data is processed. The increasing maturity in a progression naturally aligns with the information requirements of the guidance degrees. However, this does not mean there is only one way to provide guidance, and that guidance degrees cannot fulfill different roles at different stages.

We introduce here the concept of guidance *ripeness*, *downplay*, and *overplay*. Fig. 7 provides the mapping of these concepts to guidance degrees and guidance progression phases.

**Ripeness** refers to the alignment of the maturity of partial results with the information requirements for providing guidance at a particular degree. In this context, a degree of guidance is considered "ripe" if it matches the level of maturity of the progression. For example, prescribing guidance is only ripe when partial results have matured enough to provide a high-confidence actionable recommendation.

This concept provides a useful metric for determining when certain types of guidance should be deployed. However, it assumes a clear, linear relationship between progression maturity and guidance degrees, which might oversimplify complex real-world scenarios. In practice, the maturity of partial results might vary across different aspects of the data, leading to challenges in determining ripeness universally across all guidance degrees.

**Downplay** occurs when guidance is intentionally offered at a lower degree than its ripeness would suggest. The rationale for this approach is to give users more freedom to explore, fostering user-driven insights and waiting for a timely moment to provide potentially task-transitioning guidance. Downplaying can also help gather more information about the user (e.g., their tasks, needs, or preferences), before constraining exploratory activities with more definitive guidance.

While downplay promotes user agency, it may risk leaving users without adequate support during critical stages of the analysis. For inexperienced users or tasks requiring high precision, offering guidance below its ripeness could lead to frustration, inefficiency, or misinterpretation of the data.

**Overplay** involves providing guidance at a higher degree than ripeness warrants. This strategy is typically used to accelerate feedback from users or to expose them to areas of the data they might otherwise overlook (e.g., Ip & Varshney [IV11]). Overplay might also serve as a way to "show the user around" during the early stages of progression when partial results are not yet fully mature.

Although overplay can be helpful for orienting users quickly, it risks undermining user trust in the system if the provided guidance is perceived as premature or unreliable. Users may disregard guidance that is overly prescriptive early on, particularly if it conflicts with their expectations or observations from the data.

An important takeaway from this discussion of ripeness, downplay, and overplay is that they all have both advantages and disadvantages, and therefore, require a careful design. This should also include some way of communicating to users how ripe the guidance actually is.

### 4.3. Progressive Guidance Tasks

P4G also involves making guidance tasks (see Fig. 3) progressive, that is, give them the ability to dynamically adjust the degree of guidance. This ability is based on two key requirements. First, guidance must be progressively generated through chunking, ensuring the production of partial results and in turn partial guidance answers. Second, guidance tasks must collectively form a system that delivers comprehensive support addressing all path-related knowledge gaps for a given target across all guidance degrees [PMCM24]. Note that if a guidance system supports the degree of prescribing guidance, offering complete answers by resolving both target and location gaps, it inherently can also deliver all lower degrees of guidance for that target. In essence, a prescribed answer can be decomposed into the answers corresponding to all subordinate guidance tasks, just as a complete path integrates its location and target components. Next, we describe the tasks and how they benefit from partial results.

**Prescribing** provides a full actionable path that is automatically enacted. As such, the answer itself cannot be delivered in chunks because it represents a decision that is taken. Hence this guidance task reaches its ripeness only with definitive partial results.

**Directing** provides either location or target, and hence incomplete paths. They open up a set of ordered alternatives for the user to choose from. It can be delivered in chunks as long as these alternatives are allowed to change as the progression matures. However, providing them from the start of a progression would make the answer too unstable in the beginning, and for that reason these tasks reach ripeness during mature partial results.

**Orienting** provides answers that are only partial because they are vague. Similar to prescribing and directing tasks, orienting tasks provide parts of a path. Yet, orienting tasks are not conclusive with respect to a specific target or location, potentially highlighting a multitude of targets or locations, usually with varying degrees of intensity. Orienting tasks lend themselves to chunking and are ripe as soon as early partial results are available because they do not offer decisions but evaluations (i.e., values produced by the guidance model).

As orienting represents the evaluations of the guidance model towards a path, they show the inner workings (the partial results) of its decision-making process. When partial results are mature enough, a certain target or location can be fixed with enough certainty to provide directions. Again, a prescription is ripe only when both target and location can be fixed with enough certainty. The implicit hierarchy of guidance degrees and their information needs states that, if a system is not able to progressively generate an answer for orienting or directing guidance, it cannot provide prescribing guidance, because the partial results leading to a prescription are what constitutes the basis for the subordinate guidance tasks.

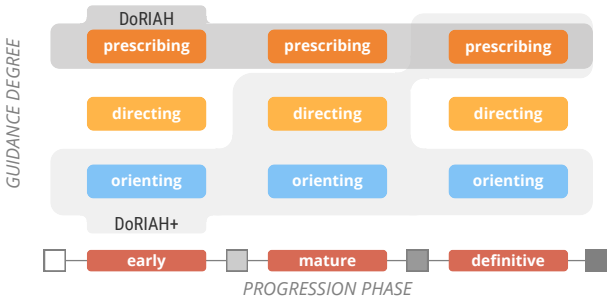


**Figure 8:** ***DoRIAH Case Study****—The original system (DoRIAH) provides prescribing guidance irrespective of the progression phase. Applying our framework, we arrive at a progressive guidance behavior (DoRIAH+) including all guidance degrees when they are ripe in addition to a progressive orienting guidance.*

### 4.4. Case Study: DoRIAH

In this section, we apply P4G to analyze the design and functionality of DoRIAH [PMCM23], a guidance-enhanced VA system with features of progressiveness. Specifically, we define the guidance progression, identify its phases, and apply the ripeness criteria to establish guidance roles.

**Overview** DoRIAH is a VA system for experts to curate selections of archival images for unexploded ordnance detection. The user's objective is to define image sets that cover a specific area over an extended time period while optimizing limited resources. DoRIAH supports explore, locate, browse, and lookup tasks over these images through guidance, mainly at two degrees, which are powered by a heuristic model that assigns an interest value to each image. **Orienting guidance** hints at promising images within a temporal vicinity by visualizing the interest values, and **prescribing guidance** pre-defines sets of aerial images for further human review and refinement. The prescription process leverages a greedy algorithm that updates the selection by adding one new image at each iteration. This incremental approach ensures system interactivity and enables users to observe the selection's construction in real time. While the algorithm focuses on a single optimization path, user interaction, such as selecting or dismissing images, can introduce constraints that dynamically alter the final solution. This strategy showcases some benefits of progressiveness, including interactivity and observability, but does not explicitly account for progression maturity or explore the full potential of progressive guidance roles as outlined in our framework. Fig. 8 summarizes the original guidance and progressive state of the system (DoRIAH) and the improved design we propose next (DoRIAH+).

**Guidance progression** The guidance progression in DoRIAH begins with the first round of evaluation of images by the model and concludes when the final image is added to the prescribed set. The progression

phases are defined as follows. **Phase I** lasts until about half of the selection has been prescribed (early results), **phase II** until most of the answer is delivered (mature results), and the **phase III** until the prescription is finished (definitive results). If the user rejects a prescribed image, the progression is temporarily set back by one step and resumed with the updated constraints. There is not orienting guidance during the progression, as the interest values assigned in the successive evaluations by the heuristic model remain constant (the guidance produces answers directly from the data and not from partial results).

**Applying the framework** In Fig. 8 we can see that DoRIAH is performing overplay by providing prescribing guidance already during the early progression phases. This is to elicit quick user feedback on preliminary suggestions to steer the algorithm toward more relevant results, and to expose the user to the overall structure of the guidance model and its evolving recommendations for transparency. However, the guidance is not correctly leveraged, as users cannot focus their attention to giving feedback and observing the progression at the same time, potentially leading to feelings of overwhelming and frustration [CAGM21], trust miscalibration issues [LMDT23], and insufficient feedback collection. The ripeness criterion does not impose a particular way of delivering progressive guidance. It rather spans a design space by aligning the degree of guidance with the maturity of the progression. This allows us to derive a straightforward alternative for how DoRIAH can improve its progressive guidance. Next, we describe DoRIAH+.

**Phase I + Orienting** The progressive guidance begins with *orienting* the user by highlighting images based on a fuzzy interest measure, e.g., by progressively aggregating partial results, which represent evaluations of the heuristic model, for each image. This allows users to gain an understanding of the model's inner workings while retaining complete freedom to explore. Since orienting guidance stabilizes quickly due to its aggregated nature, it can continue functioning effectively in later phases.

**Phase II + Directing** In the intermediate phase, the heuristic model arrives at a mature partial solution without yet incorporating user input. At this stage, the guidance task *direct to targets* can compile a list of candidate images to add to the selection. This approach encourages engagement and collects feedback to refine the solution while also fostering trust by involving users in the decision-making process.

**Phase III + Prescribing** Building on the user's interactions and feedback from the previous phases, the guidance during the final stage can *prescribe* a complete image set. This guidance answer reflects a progressively constructed, mixed-initiative result. Depending on how much user refinement is required, guidance in this phase might need to be downplayed (to *directing*) until a certain confidence is reached.

By applying the ripeness criteria and harnessing the power of guidance chunking, this case study shows a more elaborate guidance strategy for the examined system, and also outlines other possibilities for guidance design. Note that to ensure controllability [CAA*20], a system should allow users to switch between overplay and downplay, depending on their perceived knowledge gaps and interaction needs.

## 5. Discussion

As VA has evolved as a field, its inherent challenges have led to specialized subfields (e.g., explainable VA, knowledge-assisted VA, progressive VA, guidance-enhanced VA), each addressing specific, often conflicting aspects of VA. Although these subfields occasionally overlap, they have rarely been consciously integrated to take advantage of their combined strengths. Here, we have taken on the task of coupling GVA and PVA along the lines of G4P and P4G.

**Guidance for progressiveness (G4P)** Our framework highlights the potential roles of guidance within a system progression, emphasizing how guidance can align with or enhance progressiveness. However, another unexplored area lies beyond individual progressions: the *inter-progression space*. Similar to the notion of *inner* and *outer* result control in PVA [MPG*14], inner guidance supports users within a single progression, while outer guidance could oversee multiple progressions, steering users across a broader analytical landscape. Investigating this multi-progression space offers exciting opportunities for future research.

**Progressiveness for guidance (P4G)** While the concepts of ripeness, downplay, and overplay offer a flexible framework for tailoring guidance to different stages of progression, they require careful calibration. A rigid adherence to ripeness might reduce the adaptability of guidance systems, while frequent overplay or downplay could lead to inconsistent user experiences. Additionally, these concepts implicitly assume that users can distinguish and interpret the varying degrees of guidance correctly, which may not always be the case, especially for non-expert users. To address these limitations, future work should explore strategies for dynamically adapting guidance based on both progression maturity and user interaction patterns. Experiments are needed to determine how users perceive and respond to downplay and overplay in different contexts, and how these strategies impact task performance and user satisfaction. Ultimately, the success of these concepts depends on their integration into a cohesive, user-centered design framework.

**Guidance $\times$ progressiveness (GxP).** This work has explored the integration of guidance and progressiveness in two directions: G4P and P4G. Dividing the problem into these two simpler dimensions enabled us to propose a foundational conceptualization for their synergy. However, this represents only the beginning of this integration. Future research should investigate GxP, the space where both the VA part and the guidance are progressive. GxP introduces additional challenges in terms of aligning the partial results coming from the VA progressiveness and the progressive guidance and correctly quantifying and managing the resulting uncertainty introduced in this scenario.

## 6. Conclusion

Motivated by the fact that PVA and GVA share the common goal of keeping the data analysis flow in VA going, we investigated their complementary nature and the potential benefits of integrating their strengths. We highlighted how guidance can play a crucial role in steering a progression thus making it more efficient, while progressiveness can provide partial results to guidance, making the guidance more fluid and able to play different roles. Together, progressiveness and guidance can address scenarios where neither approach alone would be sufficient. Exploring this synergy, we presented a conceptualization of their combination, investigating their relation in the directions of G4P and P4G, leaving GxP for future work. Our results contribute to the theoretical foundations of a fully integrated approach and can inform designers in correctly managing the interplay and utilizing the advantages of progressiveness and guidance.

## Acknowledgments

This work was funded by Vienna Science and Technology Fund (WWTF) under grant [10.47379/ICT19047], by the Austrian Science Fund (FWF) with the grant P31419-N31 (KnoVA), by DoRIAH (FFG Grant #880883), by the MUR PRIN 2022 Project No. 202248FWFS "Discount quality for responsible data science: Human-in-the-Loop for quality data" within the NextGenerationEU Programme - M4C2.1.1.

## References

[AMSS19] ANGELINI M., MAY T., SANTUCCI G., SCHULZ H.-J.: On quality indicators for progressive visual analytics. In *EuroVA@ EuroVis* (2019), pp. 25–29. doi:10.2312/eurova.20191120. 3, 7

[ASSS18] ANGELINI M., SANTUCCI G., SCHUMANN H., SCHULZ H.-J.: A review and characterization of progressive visual analytics. *Informatics 5*, 3 (2018), 31. doi:10.3390/informatics5030031. 2, 3, 8

[BM13] BREHMER M., MUNZNER T.: A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2376–2385. doi:10.1109/TVCG.2013.124. 2, 4

[BOZ*14] BROWN E. T., OTTLEY A., ZHAO H., LIN Q., SOUVENIR R., ENDERT A., CHANG R.: Finding waldo: Learning about users from their interactions. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1663–1672. doi:10.1109/TVCG.2014.2346575. 5

[CAA*20] CENEDA D., ANDRIENKO N., ANDRIENKO G., GSCHWANDTNER T., MIKSCH S., PICCOLOTTO N., SCHRECK T., STREIT M., SUSCHNIGG J., TOMINSKI C.: Guide me in analysis: A framework for guidance designers. *Computer Graphics Forum 39*, 6 (2020), 269–288. doi:10.1111/cgf.14017. 2, 10

[CAGM21] CENEDA D., ARLEO A., GSCHWANDTNER T., MIKSCH S.: Show me your face: towards an automated method to provide timely guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2021), 4570–4581. doi:10.1109/TVCG.2021.3094870. 10

[CCEA*23] CENEDA D., COLLINS C., EL-ASSADY M., MIKSCH S., TOMINSKI C., ARLEO A.: A heuristic approach for dual expert/end-user evaluation of guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics 30*, 1 (2023). doi:10.1109/TVCG.2023.3327152. 2

[CGM*17] CENEDA D., GSCHWANDTNER T., MAY T., MIKSCH S., SCHULZ H.-J., STREIT M., TOMINSKI C.: Characterizing Guidance in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 111–120. doi:10.1109/TVCG.2016.2598468. 1, 2, 3, 5, 8

[CGM*18] CENEDA D., GSCHWANDTNER T., MAY T., MIKSCH S., STREIT M., TOMINSKI C.: Guidance or no guidance? a decision tree can help. In *EuroVA@ EuroVis* (2018), pp. 19–23. doi:10.2312/eurova.20181107. 2

[CKBE19] CUI Z., KANCHERLA J., BRAVO H. C., ELMQVIST N.: Sherpa: Leveraging user attention for computational steering in visual analytics. In *IEEE Visualization in Data Science (VDS)* (2019), IEEE, pp. 48–57. doi:10.1109/VDS48975.2019.8973384. 3

[FFNS19] FEKETE J.-D., FISHER D., NANDI A., SEDLMAIR M.: Progressive Data Analysis and Visualization (Dagstuhl Seminar 18411). *Dagstuhl Reports 8*, 10 (2019), 1–40. doi:10.4230/DagRep.8.10.1. 2, 3

[FFS24] FEKETE J.-D., FISHER D., SEDLMAIR M. (Eds.): *Progressive Data Analysis*. Eurographics Association, 2024. doi:10.2312/pda.20242707. 2

[FP16] FEKETE J.-D., PRIMET R.: Progressive analytics: A computation paradigm for exploratory data analysis. *arXiv preprint arXiv:1607.05162* (2016). doi:10.48550/arXiv.1607.05162. 3

[GW09] GOTZ D., WEN Z.: Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces* (2009), ACM, pp. 315–324. doi:10.1145/1502650.1502695. 2

[HASS22] HOGRÄFER M., ANGELINI M., SANTUCCI G., SCHULZ H.-J.: Steering-by-example for progressive visual analytics. *ACM Transactions on Intelligent Systems and Technology (TIST) 13*, 6 (2022), 1–26. doi:10.1145/3531229. 3

[HS23] HOGRÄFER M., SCHULZ H.-J.: Tailorable sampling for progressive visual analytics. *IEEE Transactions on Visualization and Computer Graphics 30*, 8 (2023), 4809–4824. doi:10.1109/TVCG.2023.3278084. 3

[IV11] IP C. Y., VARSHNEY A.: Saliency-assisted navigation of very large landscape images. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1737–1746. doi:10.1109/TVCG.2011.231. 9

[KK17] KERRACHER N., KENNEDY J.: Constructing and evaluating visualisation task classifications: Process and considerations. *Computer Graphics Forum 36*, 3 (2017), 47–59. doi:10.1111/cgf.13167. 4

[LMDT23] LIU J., MARRIOTT K., DWYER T., TACK G.: Increasing user trust in optimisation through feedback and interaction. *ACM Transactions on Computer-Human Interaction 29*, 5 (2023), 1–34. doi:10.1145/3503461. 10

[MA14] MIKSCH S., AIGNER W.: A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics 38* (2014), 286–290. doi:10.1016/j.cag.2013.11.002. 2

[MGG*23] MONADJEMI S., GUO M., GOTZ D., GARNETT R., OTTLEY A.: Human–computer collaboration for visual analytics: an agent-based framework. *Computer Graphics Forum 42*, 3 (2023), 199–210. doi:10.1111/cgf.14823. 4

[MPG*14] MÜHLBACHER T., PIRINGER H., GRATZL S., SEDLMAIR M., STREIT M.: Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1643–1652. doi:10.1109/TVCG.2014.2346578. 3, 8, 10

[MSA*19] MICALLEF L., SCHULZ H.-J., ANGELINI M., AUPETIT M., CHANG R., KOHLHAMMER J., PERER A., SANTUCCI G.: The human user in progressive visual analytics. In *EuroVis Short Papers* (2019), pp. 19–23. doi:10.2312/evs.20191164. 2, 7

[PMCEA*22] PÉREZ-MESSINA I., CENEDA D., EL-ASSADY M., MIKSCH S., SPERRLE F.: A typology of guidance tasks in mixed-initiative visual analytics environments. *Computer Graphics Forum 41*, 3 (2022), 465–476. doi:10.1111/cgf.14555. 2, 3, 4, 5, 6

[PMCM23] PÉREZ-MESSINA I., CENEDA D., MIKSCH S.: Guided visual analytics for image selection in time and space. *IEEE Transactions on Visualization and Computer Graphics 30*, 1 (2023). doi:10.1109/TVCG.2023.3326572. 9

[PMCM24] PÉREZ-MESSINA I., CENEDA D., MIKSCH S.: Enhancing visual analytics systems with guidance: A task-driven methodology. *Computers & Graphics 125* (2024), 104121. doi:10.1016/j.cag.2024.104121. 3, 9

[PTMB09] PIRINGER H., TOMINSKI C., MUIGG P., BERGER W.: A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1113–1120. doi:10.1109/TVCG.2009.110. 3

[SJB*21] SPERRLE F., JEITLER A., BERNARD J., KEIM D., EL-ASSADY M.: Co-adaptive visual data analysis and guidance processes. *Computers & Graphics 100* (2021), 93–105. doi:10.1016/j.cag.2021.06.016. 2

[SSK*15] SACHA D., SENARATNE H., KWON B. C., ELLIS G., KEIM D. A.: The role of uncertainty, awareness, and trust in visual analytics. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 240–249. doi:10.1109/TVCG.2015.2467591. 3, 5

[SSKEA21] SPERRLE F., SCHÄFER H., KEIM D., EL-ASSADY M.: Learning contextualized user preferences for co-adaptive guidance in mixed-initiative topic model refinement. *Computer Graphics Forum 40*, 3 (2021), 215–226. doi:10.1111/cgf.14301. 4

[SSMT13] SCHULZ H.-J., STREIT M., MAY T., TOMINSKI C.: Towards a Characterization of Guidance in Visualization. Poster at IEEE Conference on Information Visualization (InfoVis), 2013. 2

[SSS*14] SACHA D., STOFFEL A., STOFFEL F., KWON B. C., ELLIS G., KEIM D. A.: Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1604–1613. doi:10.1109/TVCG.2014.2346481. 2

[SW24] SCHULZ H.-J., WEAVER C.: Transient visual analytics. In *Proceedings of the 15th International EuroVis Workshop on Visual Analytics* (2024), The Eurographics Association. doi:10.2312/eurova.20241108. 3

[UAF*23] ULMER A., ANGELINI M., FEKETE J.-D., KOHLHAMMER J., MAY T.: A survey on progressive visualization. *IEEE Transactions on Visualization and Computer Graphics 30*, 9 (2023), 6447–6467. doi:10.1109/TVCG.2023.3346641. 2, 7, 8

[VW06] VAN WIJK J. J.: Views on visualization. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 421–432. doi:10.1109/TVCG.2006.80. 1, 2

[WM04] WILLIAMS M., MUNZNER T.: Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization* (2004), IEEE, pp. 57–64. doi:10.1109/INFVIS.2004.60. 6

[ZGC*17] ZGRAGGEN E., GALAKATOS A., CROTTY A., FEKETE J.-D., KRASKA T.: How progressive visualizations affect exploratory analysis. *IEEE Transactions on Visualization and Computer Graphics 23*, 8 (2017), 1977–1987. doi:10.1109/TVCG.2016.2607714. 5