Developing Remote Delivery of Language and Cognitive Training for Use With Children With Autism: A Technological Report

Jordan Belisle¹ · Raymond Burke² · Leah Clark³ · Rebecca Jepsen⁴ · Kayla Welch^{1,3} · Lindsey Dennis¹ · Nicole McDonald¹

Accepted: 11 December 2020 / Published online: 20 April 2021
© Association for Behavior Analysis International 2021

Abstract

Remote delivery of language and cognitive training is becoming increasingly prevalent within special education settings, and the recent COVID-19 pandemic has challenged many providers to pivot to telehealth models. This technical article outlines a procedure for developing computerized discrete-trial training programs using commonly available software, as well as a description of how to adapt this strategy to teach chained tasks remotely. Within this article, we describe how to establish unidirectional and bidirectional remote interfaces to work directly with learners. Finally, we conducted a field test of these approaches with programs adapted from two standardized curricula: PEAK and PRISM. We conclude the article by discussing barriers and potential solutions that we observed while field-testing these procedures within special education settings in response to the wide-scale emigration to remote teaching due to the COVID-19 pandemic.

Keywords technology · telehealth · remote delivery · PEAK · LIFE

Among the several dimensions of applied behavior analysis are the generality and transportability of intervention technologies (Baer, Wolf, & Risley, 1968; Quigley, Blevins, Cox, Brodhead, & Kim, 2019). *Setting generality* refers to the transfer of efficacious intervention outcomes to natural settings where behavior change is most likely to result in reinforcement for the client (Rincover & Koegel, 1975). *Transportability* in intervention research can be defined as “the movement of efficacious treatments to usual-care settings” (Schoenwald & Hoagwood, 2001, p. 1192) and may be especially important in the context of treatment for children with disabilities. More recently in the evolution of behavior-analytic services, generality and transportability are augmented through telehealth models that make use of electronic

information and telecommunication technologies to deliver services from a distance (Pellegrino & DiGennaro Reed, 2020). For example, Pellegrino and DiGennaro Reed (2020) used videoconferencing to establish chained home-cooking tasks across two adults with intellectual disabilities. As noted by LeBlanc et al. (2020), although telehealth strategies have been emerging within behavior analysis over the past 2 decades, many behavior analysts have had to become rapidly familiar with these approaches due to the COVID-19 pandemic. In the current technological article, we describe a set of procedures for promoting the transportability of language and cognitive training delivered through discrete-trial instruction. Additionally, we field-tested this strategy within a standardized program (i.e., PEAK Relational Training System; see Dixon et al., 2017a; Dixon, Belisle, Munoz, Stanley, & Rowsey, 2017b) using a familiar stimulus-building platform (e.g., Microsoft PowerPoint) and delivering the content remotely using commercially available telecommunication technologies (e.g., Zoom). By building computerized programs and delivering content directly through telecommunication, behavior analysts can deliver instruction remotely in the context where behavior change may be most preferable and can quickly adapt to public health crises.

Ferguson et al. (2019) conducted a systematic review of telehealth approaches to behavior-analytic interventions with

✉ Jordan Belisle
jbelisle@missouristate.edu

¹ Missouri State University, 901 S. National Ave., Springfield, MO, USA

² Apex Regional Program, Glenwood, IA 51534, USA

³ Pender Public School, Pendragon Academy, Pender, NE 68047, USA

⁴ Tower School, ESU #1, Wayne, NE 68787, USA

individuals with autism, locating 28 studies that met the inclusion criteria and involving a total of 307 participants with autism. Results suggested that interventionist training delivered through telehealth allowed for successful interventions that addressed one or more target behaviors. The majority of strategies included functional analyses to address challenging behavior (43%) and naturalistic training of adaptive skills (36%). From our own review of the literature, few studies have attempted to deliver training directly through the telecommunication interface, instead relying on implementation by a caregiver. However, discrete-trial training strategies such as those used in PEAK or other curricular packages may allow for direct work with a remote interventionist (i.e., telehealth direct therapy). Ferguson et al. (2019) also noted a low quality within current research on telehealth within behavior-analytic interventions for individuals with autism. Using a standardized strategy for material development and content delivery through telecommunication [could allow for more high quality evaluations of behavior analytic services delivered remotely]. In addition, if used within a standardized assessment and curriculum such as PEAK as was done in this article, this could confer an additional advantage of streamlining approaches across multiple intervention targets to further augment greater structure across multiple programs.

We chose to target procedures described within PEAK specifically due to the international use and availability of this technology, as well as a direct evidence base supporting the reliability and validity of PEAK assessments and the effectiveness of PEAK interventions (Ackley et al., 2019; Dixon, Belisle, McKeel 2017a et al., 2017; Dixon et al., 2017b; Dixon, Paliliunas, Barron, Schmick, & Stanley, 2019). However, the strategies described here are not specific to any single program or curriculum. In particular, these strategies can be used to develop and deliver both expressive and receptive communication training directly through a remote training interface within a comprehensive telehealth approach. Although prior tutorials have described how to develop discrete-trial training software using tools like Microsoft PowerPoint (e.g., Cummings & Saunders, 2019), the current article extends this work by delineating between receptive and expressive targets, allowing for the recording of learner responses within receptive communication training, and delivering the content through a remote interface.

At the time of writing the current article, the spread of COVID-19 as a global pandemic has restricted the access of thousands of children with disabilities to schools or centers that traditionally provide behavior-analytic programming. Remote service delivery affords the advantage of continuing programming in the context of natural disasters that can restrict in-person delivery. Due to the COVID-19 pandemic and the shutdown of three in-person special education programs, we were able to initiate a field test of PEAK programming across two sites and adapt the strategy to teach chained

responding using another related curriculum developed by the same research team (i.e., LIFE, Dixon, 2021). Field-testing allows technology developers to test the feasibility of a new technology prior to disseminating the technology for future use and empirical evaluation. We had the opportunity to provide the tutorials for creating stimuli and initiating remote interfaces to novel implementers to ensure they could develop materials and deliver training given the instructions contained in this article. We also had the opportunity to identify potential barriers and obtain feedback on potential solutions when using these tools within a telehealth model.

Task Analyses for Building and Delivering Remote Programming

Rodriguez (2020) described a telehealth model selection matrix for behavior analysts that describes four levels of intervention: minimal modifications required, modifications to skill acquisition plan required, caregiver coaching required, and advanced problem solving required. The programs developed and described here may be most appropriate for programs that occur in the first and second levels, where the behavior technician delivers the intervention directly in a one-to-one format, and individualized adjustments to the program can be made by the behavior analyst. The programs are designed such that the involvement of a caregiver or interventionist is not required, although such individuals could aid in the delivery of the programming.

In general, language and cognitive training programs can be subdivided into speaker (expressive) and listener (receptive) targets. We have adopted the terms “expressive” and “receptive” in this article to remain consistent with a vernacular more generally accepted outside of the behavior-analytic community. Expressive programs involve the participant engaging in a vocal or motor communicative response that is evoked by specific discriminative stimuli. Our goal with this programming is to allow for the systematic delivery of discriminative stimuli in either a randomized or quasirandomized presentation and to allow for a measurable verbal response emitted by the learner. The procedures are extended from the methods described by Cummings and Saunders (2019). Receptive programs involve the participant engaging in a motor selection response that is evoked by specific discriminative stimuli. We have field-tested each of these procedures with learners receiving remote service delivery.

In addition, we adapted the procedures used for expressive programming to teach chained tasks within the field test, allowing for a consistent deployment of procedures such as those described by Pellegrino and DiGennaro Reed (2020). To fully execute program development and remote delivery, the implementer will need a computer or tablet, stable internet connection, and microphone. If the implementer additionally

has a webcam, this provides the option of allowing the learner to see the implementer. The learner also will need a computer or tablet, stable internet, microphone, and webcam so that the implementer can ensure the learner is attending and to respond to behavior in the moment. It may also be helpful for the computer or tablet to have touch screen capabilities to allow the learner to touch the stimuli in receptive programs.

Computerized Expressive Communication (Speaker) Programs

An expressive program generally requires the learner to engage in a vocal response to a discriminative stimulus. Two examples include a “yes” or “no” response (e.g., when shown a picture of a cat and asked “Is this a cat?” the learner may respond vocally with the word “yes”) or an open-ended vocal response (e.g., when shown a picture of a cat and asked “What is this?” the learner may respond with the vocal word “cat”). In echoic programs, the expressive response is formally identical or similar to the discriminative stimulus. In tact programs, the expressive response is contingent on the physical features of the stimulus and maintained by generalized reinforcement (e.g., praise). In manding programs, the response is under the control of specific motivating operations and discriminative stimuli signaling the availability of the requested item. The procedures that follow are adaptable enough to allow for the development of several forms of expressive verbal behavior training.

Develop Materials Needed to Conduct the Programs

We have developed materials for several programs using Microsoft PowerPoint, as well as Boom Cards (www.boomlearning.com). Microsoft PowerPoint can allow for greater customizability when developing new programs that may be especially useful for more advanced programs, such as

those that appear later in comprehensive curricula such as PEAK (e.g., deictic, “What do you see?” and “What do I see?”). Boom Cards allow for built-in feedback systems and automatically track the percentage of correct responding. We encourage our technicians to select the development tool that best fits the program being converted to an online platform. A progression series for each is shown in Figures 1 and 2. To develop an expressive communication program using this format, open Microsoft PowerPoint and select the blank design template. On the first slide, place a square at the top left of the screen and number the square “1.” Copy and paste the square and number the next square “2.” Continue this process until the number of squares is equal to the number of potential target stimuli. For example, if a program contains eight discriminative stimuli, then eight squares should be created and laid out on the screen with equal spacing. Once done, this screen will be used by the implementer to quasirandomize the delivery of the discriminative stimulus. On the second slide, build the first discriminative stimulus. For example, in the PEAK Direct Training (DT) program “Tacting Animals,” the first discriminative stimulus may be a picture of a dog. On the third slide, build the second discriminative stimulus. Continue this process until the number of discriminative stimuli equals the number of squares on Slide 1 (e.g., eight). On the final slide, go to Objects and select the button that resembles a return arrow (see Figure 1). Place this button in the middle of the screen. Above the button, add a text box and the text “That is correct!” and place the text above the button. Your slide distribution should resemble that shown in Figure 2. The positive-feedback slide can be adjusted to contain any visual feedback developed by the implementer.

Next, you need to build programmatic functions into the slideshow so that the slides progress as intended when the implementer presses the buttons in Slideshow mode. First, go to Slide 1, which contains the implementer selection array. Right-click the first box (Box 1) and left-click “Link to.” Then, left-click “Place in this document” and select the first

Fig. 1. Target Slide Progression for Expressive Communicative Programs on PowerPoint. *Note.* The selection array contains boxes that are linked to discriminative stimuli. The discriminative stimulus corresponding with the implementer’s selection is displayed in the center of the screen (e.g., “Dog”) and corresponds with the

implementer’s vocal instruction (e.g., “What is this word?”). Clicking or touching the stimulus progresses to the positive-feedback screen. Clicking or touching the return arrow returns the program to the selection array to initiate the next trial

Fig. 2. Developing Slide Decks Using Boom Cards Commercially Available Software. *Note.* Decks are developed initially by the implementer. Random presentation is achieved by the program. The

user selection response is recorded by touching or clicking the stimulus. Feedback is provided in the form of a green circle around the correct selection, and the implementer progresses to the next trial in the block

discriminative stimulus slide; finally, left-click “OK.” To test this function, activate the slideshow by left-clicking “Slideshow” on the top panel and left-click “From the Beginning.” In the slideshow, select the first box (Box 1); this should progress the screen to the first discriminative stimulus. To activate the other buttons on the implementer array, repeat with the remaining box–stimulus slide combinations. Once done, each slide number should activate one discriminative stimulus that is selected by the implementer at the beginning of the trial.

If you are tracking the stimulus presentation arrangement during data collection, we recommend using the number corresponding to the selection array for ease of implementation and streamlining program delivery. At this point, the program will allow for the implementer to select one stimulus; however, we do not want to restart the slideshow for each stimulus presentation. We also want the program to provide positive feedback when the learner engages in a correct response to allow for a visual component that can be combined with verbal praise delivered through a platform such as Zoom. To do this, go to Slide 2, right-click the stimulus, and left-click “Link to.” Then, left-click “Place in this document” and select the final slide (“That is correct”). Repeat this process with the remaining discriminative stimuli. Now, when the implementer clicks the stimulus in Slideshow mode, the program will provide the learner with positive feedback. Finally, link the return button on the positive-feedback slide to the implementer selection array. Doing this creates a loop function where the implementer can simply click the return button after each successful trial to return to the selection array in order to progress to the next trial in the block. Run the program to test out these functions and to troubleshoot any missed steps that may be hindering your program. At this stage, you have successfully built an expressive communication program that can be delivered using any tablet or computer with PowerPoint functionality (e.g., Microsoft Surface, iPad). Other platforms such as Zoom or Google are needed to give remote functionality, which we will progress through next. At this stage, however, this program could be used within in-person programming to reduce material costs and potentially improve learner engagement.

We began this section by describing how to build programs using Microsoft PowerPoint because of the vast array of customization options. Multiple stimulus arrays, sequences with a delay, and many other progressions can be built using this system, which are outside the scope of the current article. We describe a method for building a basic expressive program and leave it to readers to explore ways to build on this basic model. Similar functionality can be achieved using Boom Cards for setting up other basic programs. At the time of writing this article, due to the COVID-19 pandemic and school shutdowns, school personnel could develop free user accounts, and accounts can generally be created for a small user fee. Users can create their own materials or purchase and download materials developed by other users, where several card decks have already been developed. To create new decks, left-click the “Studio” icon in the top menu bar and then select “Make Decks” under “Asset Managers.” Create a “New Deck.” Then, design your template card that specifies the location of the discriminative stimulus. After the template has been developed, begin creating your deck by selecting “Card 1” and inputting the first stimulus. Repeat this process with the remaining cards. You can also add audio and video clips to the decks just as you can with PowerPoint. To title the deck, left-click “Details” at the top of the page and add a title corresponding to the PEAK program. Now, when you run the program with the learner, select the deck with the stimuli you want to use, and the software will automatically randomize the presentation of the stimuli. You also can record on the program if a correct response was registered that will provide a tally at the end of the trial block to record for PEAK data collection.

Initiate the Unidirectional Interface With the Learner

Zoom is a common teleconferencing platform that is commercially available and, under certain subscriptions, can be made compliant with ethical laws and regulations. Zoom can be downloaded onto a computer or tablet that is accessible to the learner. Begin by scheduling a meeting time with the

learner. A parent or caregiver may need to set up the connection at the beginning of each training session. At the onset of the meeting time, join the session by selecting “My Meetings,” locating the meeting, and selecting “Join.” We encourage readers to name the meetings with a convention that allows for smooth service delivery, such as denoting the implementer (e.g., JB), the learner (e.g., NC), and the session number (e.g., 5) in the meeting name (e.g., “Session JBNC5”). The learner follows these same steps to join the session. This activates a two-way visual–audio connection between the implementer and the learner. If the learner is using an iPad or other touch screen device, we recommend locking the device so that the learner cannot close the session or engage with other applications. To share the PowerPoint slideshow with the learner, left-click the green “Share” button located at the bottom middle of the screen, as shown in Figure 3. From the available share options, select the PowerPoint slideshow you developed previously. This will share your program with the learner. Then, run the slideshow. At this stage, the learner should see the implementer in the top-left corner of the screen and the implementer selection array. Because this is a unidirectional interface, only the implementer can control the program. The learner can also hear the implementer. The implementer should also see and control the selection array, as well as see the learner in the top-left corner of the screen. The implementer can also hear the learner, which is critical for expressive programs. Now, the implementer can simply run the program as designed through the Zoom interface. Audible instructions and prompts can be delivered by the implementer to the learner through this interface, and responses can be recorded manually by the implementer throughout instruction, like when conducting in-person programming. If the implementer wishes to record the session for review or to obtain interobserver agreement, this can be achieved

by selecting the “Record” option within the Share menu that is made available while the screen is being shared. The implementer will then select the location in which the recording will be saved.

Although this tutorial is specifically geared to the development of discrete-trial training programs such as those in PEAK, the unidirectional interface could also be used to allow for remote delivery of chained vocational training systems such as in the LIFE curriculum (Dixon, 2021). The LIFE curriculum includes a list of 184 vocational/life skill targets that range from simple life skills (e.g., getting dressed) to more complex vocational skills (e.g., handling currency at a retail location). Program development from within a remote medium differs minimally from in-person programming. Once a list of target programs has been identified, the implementer develops a task analysis that is used to chain the complex behavior. The task analysis can be developed by performing the skill and recording the steps, observing another learner perform the skill and recording the steps, or recruiting an expert to develop a list of steps. A major consideration within an online platform may involve the inclusion of caregivers in the development of the task analysis. Contextual factors can differ markedly across locations that may influence the steps within the task analysis. For example, if you are using a remote system to teach getting dressed, the steps required to access the clothes may differ based on the location of the clothes, as well as the desired destination for worn clothes. An advantage of this system is the ability to teach the behavior directly within the context within which the behavior is likely to occur, while minimizing travel and intrusive observation conditions that may result from in-person training within the home context. Zoom can be used for interfacing LIFE targets or other chained task targets in the same way as described previously.

Fig. 3. Screen Sharing to Develop Unidirectional User Interface Through Zoom. *Note.* Select “Share Screen” from the bottom panel (1), then click the program developed using PowerPoint or Boom Cards from the screen array (2)

Computerized Receptive (Listener) Programs

We chose to discuss expressive programs first even though these programs tend to occur later in most curricular approaches such as in PEAK, as well as in language development more generally (Dixon, 2016). This was done because additional stimuli need to be provided to allow for receptive responding, which makes developing receptive programs more complicated. In particular, when building the programs, you need to provide selection arrays so that learners can engage in the correct response. Second, you need to build a bidirectional Zoom interface so that the learner's selection is recorded and progresses the slideshow to automatically provide the positive feedback. To reduce text, we are assuming the readers have now developed an expressive program or are more advanced in their understanding of PowerPoint, and thus we focus on the differences in these programs, as well as on how to establish the bidirectional interface needed to run these programs.

Develop Materials Needed to Conduct the Programs

Once again, PowerPoint can allow for considerable customizability within programming, and streamlined software like Boom Cards can be used to build simple programs more quickly. Select the appropriate software depending on the complexity of the program and the experience of the program designers and implementers. Building the implementer selection array is identical to expressive programming; however, at least three slides are likely needed for each discriminative stimulus. Additional slides are needed to allow the implementer to vary the array presentation, so that the correct response is not located in the same location in the array on every trial. Doing so would risk confounding the correct response with a location cue. Next, the array slides should be built similarly to Figure 4 (see progression). In this example, the sample discriminative stimulus is located in the top middle of the slide, and three array options are provided at the bottom.

Build the first slide and link the correct response array stimulus to the final positive-feedback slide. Now, when the learner touches the correct stimulus, the program will automatically and immediately provide this feedback to the learner, which is a direct benefit of computerizing programming. Copy and paste the array slide two times, so that there are three of this same slide. On the second two slides, move the array stimuli to new locations. By using the copy and paste functions, the sample and array stimuli will automatically occur in the new slides. In our experience, this saves considerable time. Next, build the first array slide for the second discriminative stimulus. Repeat the link and copy-and-paste sequence. Additionally, repeat until each discriminative stimulus in your program has multiple slides with varied arrays. Finally, ensure the return button on the positive-feedback slide links to the implementer selection array. Run the slideshow to ensure all functions work as intended. At this point, the fully functioning program should allow the implementer to select the stimulus array (and record manually) and the learner to touch or click a stimulus in the array. When the response is correct, the program will provide immediate feedback, and the implementer can progress to the selection screen to initiate the next trial. If the learner selects the incorrect stimulus, the screen will not progress, allowing the implementer the opportunity to provide prompts in order to evoke the correct response. Once again, at this stage, you have developed a fully functioning receptive communication training program that can be conducted in person, reducing material costs and allowing for immediate visual feedback for the learner. If you are conducting this programming remotely, the next step is to initiate a bidirectional Zoom interface so that the learner can select stimuli on their tablet or computer.

This description will allow readers to build basic receptive programs, and we encourage readers to explore additional changes that can be made given the vast customizability afforded by PowerPoint. Receptive programs can also be developed using Boom Cards that will automatically randomize the array and provide immediate visual feedback for learner

Fig. 4. Target Slide Progression for Receptive Communication Programs. Note. The selection array contains boxes that are linked to discriminative stimuli. The discriminative stimulus corresponding with the implementer selection is displayed in the center of the screen (e.g., “Dog”) along with learner selection options. Clicking or touching the

correct array stimulus progresses to the positive-feedback screen. Clicking or touching any other stimulus will not progress the screen, allowing for prompting. Clicking or touching the return arrow returns the program to the selection array to initiate the next trial

selections. Creating a new deck is similar to receptive programs. One difference is in the development of the template program. At this stage, a location for the sample stimulus must be specified, as well as the number of array stimuli and their locations on the card. When developing the deck, specify images that will serve as discriminative stimuli, as well as images that correspond to the correct selection response. Now, when the deck is selected, the program will randomize the presentation of the discriminative stimulus, as well as the array. When the learner touches the correct stimulus, a green circle will encompass the answer, indicating that the response was correct. When the learner touches an incorrect stimulus, a red circle will encompass the answer, indicating that the response was incorrect. As with the receptive programs, a tally is provided at the end of the block that provides data for ongoing tracking and analysis.

Initiate the Bidirectional Interface With the Learner

Setting up the bidirectional interface will likely require the assistance of a parent or caregiver; however, if a receptive response is required, the program is likely an earlier program in the curriculum, so a parent or caregiver is likely needed to initiate any program at this stage. Begin the meeting in the same way as when conducting an expressive program and share the screen with the activated slideshow. This will initiate the program unidirectionally, meaning only the implementer can manipulate the program. You will also want the learner to be able to manipulate the program to allow for the selection responses. Additionally, whereas in expressive programs the

implementer can hear the learner response, seeing the learner response is difficult if they are touching the screen, such as when running this using a tablet. The bidirectional interface needs to be initiated on the learner's end. Have the caregiver hover their mouse over the slideshow. The option "Take Remote Control" should appear at the top of the screen, as shown in Figure 5. Left-click this option. The implementer and the learner can now both control the screen, allowing the implementer to control the slide progression, and the learner to progress to the positive-feedback slide by selecting the correct comparison stimulus. If both the implementer and the learner have a strong internet connection, there should be a minimal time delay between the learner's correct response and the positive-feedback screen. If prompting is needed, the implementer can hover their mouse over the correct stimulus, indicating where the learner should click or touch. Additional prompts can be built into each array slide, constrained only by the creativity of the programmer. For example, an icon may be placed over the correct array stimulus that "Appears on click," such that if the learner touches anywhere on the screen other than the correct stimulus, the icon appears as a visual prompt for the correct response. When using Boom Cards, a red box will appear circling the learner's incorrect response and allowing the implementer to then prompt the correct response vocally or with their cursor.

Field-Testing and Challenges of Remote Implementation and Solutions

We field-tested these programs across three special education sites in the midwestern United States in response to the

Fig. 5. Screen Sharing to Develop Bidirectional User Interface Through Zoom. *Note.* The learner or caregiver selects "Remote Control" from the top panel (1), then selects "Take Remote Control" from the available options

COVID-19 pandemic, a global pandemic that resulted in schoolwide shutdowns in states where these programs were tested. This test was designed to collect in-the-moment feedback on the functioning of these tools when used to augment remote instructional strategies, and to highlight any challenges experienced when using these technologies. Data were collected from unstructured interviews that were conducted with agency implementers throughout program delivery.

The first site implemented remote PEAK Expressive and Receptive programs using a combination of PowerPoint and Boom Cards initiated through the Zoom interface. The programs were developed by paraprofessionals at this site using the task analyses contained in this technical article. The second site trained a parent to conduct the PEAK Expressive and Receptive programs using PowerPoint slideshows that they developed on site and shared with the parent. Data and progress updates were shared in real time with the site to track progress and to inform program adaptations. For this site, the programs were developed by paraprofessionals using the task analyses contained in this article, and materials were given to the parents to implement. The third site implemented LIFE programs with one adolescent and one young-adult student with high degrees of individuation of program task analyses that were developed in collaboration with caregivers. Unidirectional interfacing was achieved using Google Meet. In this case, the sessions were conducted by an author of the present study who developed the procedures for initiating the unidirectional interface using Google Meet. Therefore, this final site test provides an analysis of this remote approach to delivering programs through LIFE, rather than a test of the utility of the tutorials provided in this article for achieving successful program development. For the first two sites, the field-testing of the materials served a secondary function of evaluating whether novel staff can develop materials using the procedures described here.

It is important to note that the field test was not an experimental evaluation of these tools; it serves as a feasibility analysis of these tools and documents potential barriers. We documented challenges that emerged through remote implementation over the course of field-testing, along with solutions developed by school personnel that may assist readers in developing solutions when similar problems arise. We identified four common barriers to implementation that were solved by implementers at these sites.

Establishing Receptive and Expressive Responses On Screen

Learners come to the remote training arrangement with a history of learning through in-person instruction, and many prompts that implementers use with learners require being physically present, such as physical or gestural prompts, as

well as many forms of response modeling. A challenge experienced by the first sites involved teaching learners to interact with the remote interface when a person was not immediately present. The implementers at these sites established a new PEAK program using the customized template located at the end of the PEAK-DT module. This PEAK program involved the presentation of a circle located on the screen. To obtain the reinforcer, the learner simply had to touch the circle on the screen. Each trial relocated the circle so that the learner had to track the location of the circle and touch the screen. Occasionally, caregivers were recruited in the initial stages of this training to physically prompt the touch response and to fade the physical prompt until the independent selection response was established. Subsequent programs involved replacing the circle with pictures used within programming (e.g., a dog and a frog), then introducing an auditory discriminative stimulus presented by the implementer (e.g., “Find dog”). Once the learners mastered this step, the sites were able to introduce new PEAK programs. The primary lesson that we learned was that remote delivery may require developing new programs that specifically teach learners to interact in this new medium before progressing to programs that already exist within the PEAK modules.

For PEAK Expressive programs, the learner may not readily respond expressively to discriminative stimuli that the implementer delivers remotely. To solve this barrier, we adapted a rival/model training paradigm reported by Dixon, Belisle, Munoz et al. (2017). In rival/model training, the implementer delivers the discriminative stimulus and instruction, and a trained adult model engages in the correct response and obtains reinforcement from the implementer. Through vicarious-reinforcement observational-learning processes, the learner may additionally demonstrate the response without direct training. At the first site, this procedure had been attempted in the classroom to establish vocal imitation, which is difficult to physically prompt. In a remote-delivery format, the site had the caregiver serve as the rival model. The implementer delivered the instruction and discriminative stimulus, the caregiver engaged in the response, and the caregiver received the reinforcer. Testing was then conducted to determine if the learner also engaged in the response. The caregiver response was then faded out so that the implementer could deliver the instruction and discriminative stimulus that successfully evoked the learner response.

These represent two solutions among many possible solutions that are conceptually systematic with behavior-change principles and require only small adaptations to procedures in the existing literature. The fast-paced nature of the transition to remote instruction did not allow for the systematic collection of experimental data on this phenomena, but these case descriptions can allow for additional research on the efficacy of these and other procedural adaptations required for remote program delivery.

Developing and Maintaining Learner Attention in a Remote Format

We also encountered occasional deficits in attending to remote delivery of PEAK and LIFE programs, even after expressive and receptive responding through the remote interface was established. Many implementers within an in-person format likely embed subtle social reinforcers (e.g., smiling, laughing, overt indicators of excitement) in addition to larger programmed reinforcers (e.g., contingent breaks with preferred tangible items). We observed that the removal of these subtle social reinforcers in addition to the loss of the ability to manage and deliver many programmed reinforcers could negatively impact remote delivery of our programming. The first and third sites resolved this barrier in two ways. First, because of the video interface, social reinforcers can be delivered, but they cannot be subtle. Both sites achieved success by encouraging their implementers to become more animated in the delivery of social praise and to evoke attending behavior in the learner by engaging in preferred scripts (e.g., acting like an animal such as a moose) or, for more advanced learners, to engage in preferred conversations (e.g., discussing Star Wars characters in between trials). Success was also achieved by ensuring that the social reinforcement strategies were developed with the individualized interests of the students in mind, and both sites reported a desire to use more of these strategies within in-person programming to increase learner engagement.

Second, although remote delivery removed the potential to use some programmed reinforcers, the computerized PEAK programs also allowed for embedding reinforcers directly within the programs. For example, the first site embedded preferred YouTube videos that were presented to the participants for a fixed duration contingent on a certain number of correct responses. We also experimented with the development of a token board using PowerPoint, where tokens were exchanged for minutes of access to YouTube videos that the participant could control using the bidirectional Zoom interface. At the second site, implementers used the Premack principle by introducing preferred PEAK programs contingent on the completion of LIFE tasks. The Relational Accelerator Program (RAP; Belisle & Burke, 2020) is downloadable software that allows for the gamification of PEAK programming that creates programs that resemble a game of Whac-a-Mole. For one learner, self-ratings of program enjoyment were high for the PEAK program developed using the RAP and moderate for the LIFE tasks that were deemed essential for achieving vocational goals. Progressing from the low-probability LIFE task to the high-probability PEAK task was efficacious for this learner in increasing attending and the willingness to complete both aspects of her PEAK programming remotely. A third solution used by the third site was embedding a motivative augmental (i.e., “Values”) statement at the onset of LIFE training (e.g., asking the participant “Why are we doing this

today?” to orient the participant to their previously stated value of getting a job in the community). This was effective in reducing task refusal with young-adult learners undergoing LIFE programming in this study.

Ensuring Caregiver Implementation Fidelity and Involvement

Requirements of caregivers to ensure the success of remote PEAK programming vary considerably across learners, programs, and contexts. All three sites experienced challenges related to ensuring that caregivers were implementing aspects of the programs with fidelity, as well as ensuring continued involvement of caregivers in programming. The second site involved parents directly by teaching them to conduct programs in person with their children; this presented several barriers. A first barrier was the inconsistent implementation of PEAK programs by parents. We anticipate that this may be a common barrier for readers who attempt this strategy. To encourage more consistent participation, the site staff scheduled weekly meetings with the parent implementers to discuss any barriers to implementation. Site staff believed this strategy would be effective by easing implementation barriers through joint problem solving with parents, as well as socially reinforcing the completion of PEAK programming and data collection with learners. Respondents reported that this solution was effective in promoting program compliance; however, several errors and challenges were reported by parents attempting to implement the programs. The second site made several modifications to the programming to reduce the probability of implementation errors. First, data collection was simplified so that parents only recorded whether responses were correct or incorrect (i.e., prompt levels were not recorded). Second, test trials were removed so that parents could provide praise and reinforcement following each trial. This reduced implementation errors that would occur when discriminating between training and testing items. Third, parents were not required to record the stimulus that was presented; rather, they were instructed to randomly select a new stimulus on each trial within PowerPoint. Finally, YouTube videos were delivered along with new programs modeling the correct implementation of the program as it was developed by the school personnel.

The other two sites involved caregivers less directly, where caregivers were required to initiate some of the interfaces through Zoom or Google Meet and to provide prompts and access to reinforcers when required for specific programs and learners. Ensuring compliance and attendance to the meetings was essential for the successful delivery of these programs. Greater attendance was achieved by ensuring the meeting times were consistent each week and setting reminders to occur the day before the meeting and 30 min prior to the meeting. The latter is consistent with developing programmed

prompts to ensure the successful completion of daily tasks, and we observed greater compliance once these prompts were embedded. These reminders can be set for both Zoom and Google Meet events directly within the programs and can be set as a default option by implementers.

Adapting LIFE Programming to Fit the Home Context

The third site field-tested the remote delivery of LIFE programs. This delivery format has many potential implications outside of our context of responding to the COVID-19 pandemic and school shutdowns. LIFE programs represent daily living and vocational skills where the terminal programming goal is for the skill to occur outside of the classroom context, such as in the learner's home or at an employment location. Remote delivery of LIFE programming can allow for training to occur directly within the context where the skill is expected to occur. We experimented with using Google Meet to establish a unidirectional interface with participants. We chose Google Meet because it is fairly ubiquitous and may be more accessible to families and in locations where LIFE programming is likely to occur.

One challenge that this site experienced was that, in the school setting, the context must be adapted to allow for the behavior to occur consistent with the steps indicated in the task analysis. For example, if a step in a task analysis for cleaning windows involves retrieving the window cleaner from the cleaning cabinet, school personnel must ensure that a full cleaning bottle is located in a specific cabinet within the classroom prior to delivering the instruction for the learner to complete the task. In the home setting, however, where the field test took place, caregivers may be less willing to adapt their context to fit the steps identified within the task analysis. In addition, events such as an empty bottle are not as easily predicted in real-world locations. The site staff solved this problem in two ways. First, following the development of the task analysis, the site staff met with the caregiver to discuss each step and its applicability to the home context. Items that were not applicable were adapted or removed to ensure that the steps fit within the context where the behavior was expected to occur. Second, the data collection sheet typically contained the columns "correct" and "incorrect" with the option to record the type of prompt that was used to evoke the step in the chain. The site added a third option, "contextually appropriate," to the data collection sheet to indicate that if an item could not be successfully completed (e.g., empty cleaning bottle), the learner engaged in an appropriate solution to that event (e.g., asking the caregiver to fill the bottle). This also allowed the implementer to prompt a contextually appropriate response (e.g., "Try asking your mother for help filling the container") in the moment rather than adhering strictly to the written step within the task analysis. We also experimented with embedding visual prompts directly within

the home context. For example, two learners were given programs that involved the use of three different cleaning agents to complete three cleaning tasks (cleaning windows, tables, and fridges). An antecedent intervention involved placing pictured labels on each of the cleaning agents corresponding with their use (e.g., placing the word "WINDOWS" on the window cleaning agent and "TABLE" on the table cleaning agent). The respondents reported that this was effective in promoting the successful selection of the correct cleaning agent contingent on the cleaning task. Both of these examples again involve strategies that are already supported in the behavior-analytic literature that require a degree of adaptation to ensure success within remote program delivery.

Summary

The procedures described in this technical article can be used to (a) computerize manual intervention programming (e.g., PEAK and LIFE) and (b) deliver programming remotely to learners in a home context. Computerizing programming may be desirable regardless of location to save in material costs, increase efficiency, and promote engagement in some learners as an antecedent strategy. Many single-case experimental designs can be implemented to determine whether this delivery format is efficacious in promoting greater attending or orienting to programs, engagement, and potential reductions of challenging behavior in sessions. Developing computerized programming may be necessary when in-person delivery is not desirable or possible, such as when transferring training to new locations or due to school shutdowns. We were able to field-test these programs because of statewide school shutdowns due to the COVID-19 pandemic, prompting the development of these remote-delivery systems. The procedures described here can be used to establish unidirectional and bidirectional interfaces using commercially available software. In all of the aforementioned procedures, we attempted to develop strategies that were largely customizable and adaptable.

A staple of discrete-trial training, and indeed within special education itself (Hurwitz, Perry, Cohen, & Skiba, 2020), is the individuation of programming to meet the needs of individual learners and contexts. Not only can implementers change or alter the stimuli or arrays within programming, but unique slide transitions and embedding of videos and audio files can further allow for a user experience that is distinct and more advanced than programming with the paper stimuli that are commonly associated with discrete-trial programming. Every step described in the task analyses and each solution developed within our field test of these procedures are amenable to research designed to improve computerization and remote delivery of PEAK programming. For example, new prompting procedures such as gradually fading the translucency of distractor stimuli using a PowerPoint slide could be used as an adaptation to traditional stimulus-fading procedures either in person or remotely (Cummings &

Saunders, 2019). Embedding videos to accompany LIFE task analysis step prompts could also be included, where the learner has the option to play the video to complete the step but may earn fewer points in doing so (i.e., reinforcing independent completion). These are just two examples of a myriad of research studies that are open to investigation corresponding with behavior-analytic researchers' willingness as a field to adopt new technologies in their programming. In addition, direct testing of the development and use of these tools by individuals implementing behavior-analytic training programs could extend considerably the anecdotal field report provided in this technical article. Doing so could also allow for the direct comparison of this strategy with alternative strategies that are still under development, allowing for an increasingly inductive approach to technological development. Our hope is that this article serves as a resource for implementers to start on this journey, to allow teachers and therapists to respond to this and future pandemics, and to move programming outside of the walls of the classroom or therapy room and into the world where behavior analysts seek to affect the most change in the lives of their learners.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s40617-020-00544-6>.

Funding The authors received no financial support for the research, authorship, and/or publication of this article.

Declarations

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional human subjects committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Conflict of interest The authors declare that they have no conflict of interest.

References

- Ackley, M., Subramanian, J. W., Moore, J. W., Litten, S., Lundy, M. P., & Bishop, S. K. (2019). A review of language development protocols for individuals with autism. *Journal of Behavioral Education*, 28(3), 362–388. <https://doi.org/10.1007/s10864-019-09327-8>.
- Baer, D. M., Wolf, M. M., & Risley, T. R. (1968). Some current dimensions of applied behavior analysis. *Journal of Applied Behavior Analysis*, 1(1), 91–97. <https://doi.org/10.1901/jaba.1968.1-91>.
- Belisle, J. & Burke, R. (2020). Relational accelerator program. Retrieved from www.humansunderstandingbehavior.com/resources.
- Belisle, J., & Dixon, M. R. (2020). Rational skepticism: A scientific review of Witts' (2018) criticisms of the PEAK Relational Training System. *Journal of Applied Behavior Analysis*, 53, 620–634. <https://doi.org/10.1002/jaba.654>.
- Cummings, C., & Saunders, K. J. (2019). Using PowerPoint 2016 to create individualized matching to sample sessions. *Behavior Analysis in Practice*, 12(2), 483–490. <https://doi.org/10.1007/s40617-018-0223-2>.
- Dixon, M. R. (2016). *PEAK relational training system*. Shawnee Scientific Press.
- Dixon, M. R. (2021). *LIFE Skill Emergence System*. Shawnee Scientific Press.
- Dixon, M. R., Belisle, J., McKeel, A., Whiting, S., Speelman, R., Daar, J. H., & Rowsey, K. (2017). An internal and critical review of the PEAK Relational Training System for children with autism and related intellectual disabilities: 2014–2017. *The Behavior Analyst*, 40(2), 493–521. <https://doi.org/10.1007/s40614-017-0119-4>.
- Dixon, M. R., Belisle, J., Munoz, B. E., Stanley, C. R., & Rowsey, K. E. (2017). Teaching metaphorical extensions of private events through rival-model observation to children with autism. *Journal of Applied Behavior Analysis*, 50(4), 744–749. <https://doi.org/10.1002/jaba.418>.
- Dixon, M. R., Paliliunas, D., Barron, B. F., Schmick, A. M., & Stanley, C. R. (2019). Randomized controlled trial evaluation of ABA content on IQ gains in children with autism. *Journal of Behavioral Education*, 1–23. <https://doi.org/10.1007/s10864-019-09344-7>.
- Ferguson, J., Craig, E. A., & Dounavi, K. (2019). Telehealth as a model for providing behaviour analytic interventions to individuals with autism spectrum disorder: A systematic review. *Journal of Autism and Developmental Disorders*, 49(3), 582–616. <https://doi.org/10.1007/s10803-018-3724-5>.
- Hurwitz, S., Perry, B., Cohen, E. D., & Skiba, R. (2020). Special education and individualized academic growth: A longitudinal assessment of outcomes for students with disabilities. *American Educational Research Journal*, 57(2), 576–611. <https://doi.org/10.3102/0002831219857054>.
- Pellegrino, A. J., & DiGennaro Reed, F. D. (2020). Using telehealth to teach valued skills to adults with intellectual and developmental disabilities. *Journal of Applied Behavior Analysis*, 53(3), 1276–1289. <https://doi.org/10.1002/jaba.734>.
- Quigley, S. P., Blevins, P. R., Cox, D. J., Brodhead, M. T., & Kim, S. Y. (2019). An evaluation of explicit ethical statements in telehealth research with individuals with autism spectrum disorder. *Behavior Analysis: Research and Practice*, 19(2), 123. <https://doi.org/10.1037/bar0000094>.
- Rincover, A., & Koegel, R. L. (1975). Setting generality and stimulus control in autistic children. *Journal of Applied Behavior Analysis*, 8(3), 235–246. <https://doi.org/10.1901/jaba.1975.8-235>.
- Rodriguez, K. A. (2020). Maintaining treatment integrity in the face of crisis: A treatment selection model for transitioning direct ABA services to telehealth. *Behavior Analysis in Practice*, 13(2), 291–298. <https://doi.org/10.1007/s40617-020-00429-8>.
- Schoenwald, S. K., & Hoagwood, K. (2001). Effectiveness, transportability, and dissemination of interventions: What matters when? *Psychiatric Services*, 52(9), 1190–1197. <https://doi.org/10.1176/appi.ps.52.9.1190>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.