A NEW APPROACH TO PROGRAMMED INSTRUCTION

Francis Mechner
June 1, 1977

The purpose of this paper is to propose some new programmed instruction techniques that appear to be superior to those used in the past.

The idea of programmed instruction was originally advanced by B. F. Skinner (Skinner, 1954), and the first practical implementation of programmed instruction to train people was achieved in 1960 by Basic Systems, Inc.

In the early 1960s, the proponents of programmed instruction, including Skinner, defined programmed instruction as using (1) active response by the learner; (2) immediate reinforcement of correct responses; and (3) successive approximations towards the knowledge to be learned, in a sequence of steps so small that the learner can take each one with little difficulty.

Skinner stated, from the start, that in order for programmed instruction to work well it is important to prevent the learner from seeing the correct answer before making his own response, and that to accomplish this, a teaching machine is necessary (Skinner, 1958). But I doubted this, and so in 1959 I developed an elementary algebra program to demonstrate that programmed instruction material presented without the use of a machine is adequate and more practical. I believed that if the steps of the program were properly designed and of the right size, the learner could be relied upon to keep the correct answer covered and out of view until he had committed himself. By 1961 this viewpoint had become widely accepted, and even Skinner agreed (Holland & Skinner, 1961; Rushton, E. 1963).

During the years 1960 to 1962, through my company Basic Systems, Inc., I continued to develop methods and techniques, and wrote guides, for creating effective programmed texts (Mechner, 1961a, 1963; Margulies, 1963). An illustration of the techniques I advocated originally is provided in Mechner, 1961b. I conducted this work in a business environment in which results had to be achieved and products delivered to customers.

Starting in 1960, Basic Systems entered into numerous contracts, with corporate and governmental customers for the production of large-scale training systems that used programmed text as the main medium. In the first year of operations, I recruited Basic Systems' technical staff and trained them in a methodology and techniques that have since become widespread. This methodology included

1. specifying the desired terminal behavior in the form of test items, before the program is written;

2. analyzing the specified terminal behavior by identifying the critical "concepts" and "chains" (chains are skills, reasoning processes, procedures, etc.) that comprise the terminal behavior;

3. when the program is constructed, teaching concepts by proceeding from the specific to the abstract, or from examples to the more general case;

4. teaching chains by the backward fading method;

5. constructing frames in such a way that a critical and relevant response is called for, in a response modality that corresponds to the desired terminal behavior;

6. ensuring that the structure of the frame enables the student to arrive at the correct response only by applying the intended knowledge or skill, rather than by providing him with a trivial prompt; and

7. subjecting the program, during its development, to several cycles of empirical testing and revision, using typical members of the intended target population as test subjects.

While there is no denying that the training systems developed by Basic Systems were very effective and highly praised (Holland, 1967), it also became increasingly evident to me that some serious problems of the programmed text medium had not yet been solved.

THE PROBLEMS

One of these was student boredom: An uncomfortably high percentage of students complained that going through a program was a tedious experience.

A second, and frequently associated problem was a breakdown of learning in the middle of the program. Frequently, when going through a program, even one of high quality, the student would, at a certain point in the program, start looking at the correct answers before committing himself. The first time he "cheated" in this manner, he might rationalize it by saying or thinking something like, "I just wanted to make sure before committing myself." Then he would start doing it with increasing frequency because it was so much less effortful. As the felt need for rationalization faded, he would finally just read the questions and answers as if the program were a textbook.

When the student "cheats" in this manner, he is not going through the thought processes that the programmer was counting on. Many of the essential concepts and chains that are necessary foundations for more advanced or complex concepts and chains are then not being learned. The student himself does not know, and cannot judge, which concepts he is or is not learning, because he is not testing himself as he goes along.

Therefore, the inevitable consequence of this type of "cheating" is that the student experiences increasing difficulty with successive items. As his foundation becomes increasingly shaky, cheating becomes ever more necessary. The end result is a breakdown of learning, and the way in which the student then goes through the program is not significantly different from the process of reading an ordinary conventional text. The practicality of trying to learn a relatively complex subject by reading a textbook need not be discussed here.

The twin problems of boredom and cheating are familiar to anyone who has tested programmed instruction courses or who has personally tried to go through a conventional programmed text from beginning to end.

I first became concerned about these problems in 1961, but remained convinced that they could be solved without surrendering to the teaching machine advocates. So, in May of 1962 I wrote a programmed instruction course that demonstrated a new set of techniques which appeared to offer an avenue for eliminating these problems. The course was entitled <u>Logical Reasoning and Inference</u> for high school students. Before discussing this new set of techniques any further, however, I will discuss some of its theoretical basis.

<u>THE FUNCTION OF PROVIDING THE CORRECT ANSWERS</u>

Research studies as well as information observation have convinced most workers in the field of programmed instruction that the student rarely gives an incorrect response while convinced that he is answering correctly. Usually, the student knows when he doesn't know, or when he isn't sure. He perceives his own uncertainty before answering incorrectly. Conversely, when the student knows the correct answer, he usually knows that he knows. Now, let us examine the implications of these facts for our teaching techniques.

When the Student Has Answered Correctly

The first question is: How much benefit does the student receive if he is given the correct answer after he has answered <u>correctly</u>? Let us assume for argument's sake, as usually seems to be the case, that the student is fairly confident that he answered correctly even before he is given the correct answer. Obviously, then, giving the student the correct answer does not tell him much he did not already know. The student's reaction in such cases depends on personality. In many cases, students don't check the correct answer when they believe they have answered correctly. Some check the correct answer anyway, not to find out whether or not they were right, but because getting confirmation can be psychologically satisfying. But in all of these cases, the important fact is that the desired learning had already occurred at the time the correct answer was given. Confirmation, if received, comes after the fact.

The reason for this is that in the learning of concepts and complex chains, such as reasoning skills, the behavior involved is usually complex (Mechner, 1963, 1968). The response may involve an elaborate series of verbal chains that interlock in various intricate ways and lead, like a computer program, from the initial input to the final behavioral output. The output is the student's overt response. Note that an important distinction is being made here between the covert thought processes that lead to the a final overt response and must precede it, and the final overt observable response itself, which may involve writing something down, saying something, or performing some other physical act. If there is a weak link or element of uncertainty in any part of the covert thought process, the student will usually perceive it, the way the driver of a car perceives that he has made a wrong turn when he finds himself on unfamiliar roads or a dead end. On the other hand, if the student's covert processes follow familiar paths and interlock with each other in the accustomed manner, the student is aware of that too, like the car driver when he is on the right road. So, the student's awareness that he is responding correctly accompanies the covert thought processes constituting his response, including the final overt element.

Where does that leave the idea of reinforcement? According to the above analysis, the student may be reinforcing his own behavior continuously, as he goes through the necessary covert thought processes leading to the final overt response. This type of self-reinforcement is difficult to conceptualize, as it consists of covert behavior. It may involve the student saying to himself the equivalent of "I understand this," "I know how to do this," "I'm doing it right," or "I got it right." But regardless of whether or how this happens, one thing seems fairly clear: that telling the student the correct answer after he has successfully carried out the complex thought processes from the starting point through the final overt response, does not tell him anything that he did not know already, and is therefore of little value to him. Once the driver of a car has reached his destination, it does not benefit him to be told that he got there.

Some people who have written about programmed instruction have put forward the idea that providing the student with the correct answer, after he has responded correctly, functions as a "reinforcement." This seems to me like pseudo-psychology - using psychological jargon, in this case the term "reinforcement," to create the impression that the technique of providing the student with the correct answer is scientifically justified and is in some way derived from learning theory. No one has ever advanced any evidence suggesting that showing the correct answer to a student who has responded correctly reinforces the correct response, nor has anyone ever attempted to explain <u>what</u> behavior is presumably reinforced. It is not the purpose of this paper to defined and analyze the concept of reinforcement, as the term is used by psychologists, except to say that the presentation of the correct answer after the student has arrived at the correct answer, in a case of complex verbal learning, is not an example of reinforcement in the technical sense of the term.

When the Student is Having Trouble

The second question to consider is: What is the value of giving the student the correct answer when he is uncertain or cannot answer? Here the problem is that the student has not learned the thought processes that could lead him from the question to the answer. A psychologist might say that he has not acquired the covert mediating behavior. To use our motorist analogy again, he does not have the driving instructions or road map, and knows it. Just as it would not help the driver to be shown a photograph of his destination in such a case, so it does not help the student to be shown the correct answer. Being shown the correct answer this time would be of very little help to him the next time he is faced with a similar question or problem. Seeing the desired end result does not provide him with the thought processes he would need in order to achieve the end result on his own.

Of course, there are exceptions to this principle in certain trivial cases. If the motorist's destination is already within view, being shown a photograph of the destination <u>does</u> help him get there. Similarly, if the correct response is easy and trivial, being shown the correct answer <u>may</u> help the student bridge the gap in his thought processes and fill in the missing links that might have enabled him to get the answer independently. He may thereby learn these thought processes, so that he would be able to apply them next time. But, as stated earlier, this is the unusual and relatively unimportant case, where the item is so trivial as not to constitute an important forward step in the student's learning program. Such trivial items should be eliminated from a well-designed program.

In summary, we can say that if the student does not know how to answer, he is not greatly helped by being shown the correct answer. He needs to learn <u>how to get to</u> the answer. He needs to learn the covert mediating behavior.

When the Student Has Answered Incorrectly

The third question to consider is: What is the effect of giving the student the correct answer when he has answered incorrectly? This question divides into two parts, according to whether or not the student is aware of the fact that he has answered incorrectly.

It is relatively unusual for a student to answer incorrectly thinking that he has answered correctly. When this happens, it is generally due to a careless error or a misreading of the question. For example, in technical programs, it is sometimes due to an arithmetic error or other minor slip. In general, errors that are not accompanied by a sense of uncertainty are those made within a skill domain foreign and extraneous to the one at which the program is directed. Such errors are of little consequence, and little is gained by correcting them. For example, if a physics student makes an arithmetic error as he is solving a problem in physics, his knowledge of physics will not be increased if his arithmetic error is brought to his attention or corrected.

In the other case where the student answers incorrectly, he has a feeling of uncertainty as he is answering, or he even knows that the answer he is giving is wrong. He answers because an answer is required of him, not because he thinks it is correct. This case reduces to the second case analyzed above, where the student feels uncertain or unable to answer. He lacks the required mediating behavior, and being shown the correct answer does not install that missing mediating behavior.

When There Are Several Alternative Correct Answers

The fourth, and final question to consider is: What is the function of providing a correct answer to the student when there are several alternative correct answers? There may be many correct or appropriate ways to answer a question or to respond. Often, there are several ways to solve a problem. In such cases, what is the effect on the student of being shown one particular correct answer, and not the others? He may wrongly conclude, when seeing the given answer, that it is the only permissible answer and that his own answer was wrong. Even though he may have given an alternative correct answer, his sense of confidence when he sees the given answer and his prior justified feeling of "knowing that he knows" may be shaken. As a result, his previously-learned correct and useful mediating behavior may be weakened. At best, he will realize that he was right anyway, understanding that the answer provided by the program was an alternative correct answer, in which case he is not helped or hindered by being given the correct answer.

It seems to me that the conclusion one must draw from the above analysis of the four possible cases is that there is little or no point to providing the student with the correct answer. In none of the four cases is there any major benefit. When the learning process is proceeding satisfactorily, showing the correct answer contributes little or nothing. When the learning process is proceeding badly and there are problems, showing the correct answer does not fix the problem. It must be borne in mind that this analysis applies only to programs that teach knowledge or skills requiring significant mediating behavior, but those kinds of programs encompasses virtually every domain for which programmed instruction has been used.

WHAT IS THE ALTERNATIVE?

For the alternative, it is helpful to look at how complex verbally-mediated concepts and skills have been taught for millennia, by good teachers and tutors. In general, depending on the subject matter, a good tutor first makes verbal statements to the student or shows the student how to do it, and then asks the student to demonstrate understanding or mastery. He does this by asking the student a question or presenting him with a problem or task.

If the student answers incorrectly or makes an error, a good tutor will not simply give the student the correct answer. Rather, he may repeat the statement or the demonstration, and ask the student to try again. He may also recapitulate the series of steps that led up to this point. Or he may give the student a hint by focusing his attention on a specific part of the information that he should use in arriving at the answer.

If the student says "I can't do it," or "I don't know," the tutor may say something like "Then let's go through it again" or "Let's take a look at a related problem we did before." He may also present the student with a diagnostic question or task, to try to find out where the missing links might be. But a good tutor will never simply tell the student the right answer. Doing so would not help the student learn, and would deprive the student of the opportunity to identify and fill in the missing knowledge.

Programmed instruction should attempt to simulate what a good tutor would do. As a medium, it has some disadvantages and some advantages over a good tutor.

The most obvious disadvantage is that a pre-programmed learning sequence cannot anticipate all of every student's questions and problems with individualized precision. Every student is different and will make contact with a given program in a different way. A second disadvantage is that a program cannot maintain a student's motivation by means of the interpersonal dynamics of the interaction with a live tutor.

But a program also has many important advantages. One is that the learning sequences are carefully designed and planned, and therefore tend to be far superior to those that even an excellent tutor would be able to improvise. They can be based on a meticulous behavioral analysis of the knowledge and skills involved (Mechner, 1968) and can be pre-tested and revised during development, with representative members of the target population serving as the test subjects.

A second important advantage, a very practical one, is that instructional material can be carried around and used any time, any place, at the student's convenience. No appointments are necessary, even assuming that good tutors are available and affordable, which they are usually not.

And finally, there is no possibility of embarrassment when the student experiences difficulty or needs to think about something for a long time. He can pause and ponder whenever he desires.

In summary, I am arguing that (1) no useful purpose is served by providing the student with the correct answer, regardless of whether the student responded correctly, incorrectly, or was unable to answer; (2) when the student cannot respond correctly, a good tutor will never give him the correct answer immediately; and (3) programmed instruction should try to simulate a good tutor as much as possible.

IMPROVED PROGRAMMED INSTRUCTION TECHNIQUES

These three points suggest that we must rethink the assumption that it is useful or desirable to provide the correct answers in programmed instruction materials. But how would programs without correct answers be constructed? Simply taking a program in which answers are provided, and eliminating them, would not work. Generally, a student going through such a program would soon get stuck and that would be the end of it. A number of studies have been conducted in the past years comparing programs with and without the answers. But all of these studies have been done on conventional programs, and programs of rather poor quality at that. Such programs generally work less well without the answers, which is what these studies have found.

Omitting the correct answers offers advantages and benefits only if the program has been designed and developed to work that way. Here are some of the principal requirements of a program constructed without the correct answers:

1. The student must be required to make all of the responses that are critical for mastery of the material. Every important building block and stepping stone must be represented in the sequence of responses that are required of the student.

2. The principles of item construction, inadvertent or irrelevant cuing, demanding critical rather than non-critical responses, etc., apply in this type of program with even greater urgency than in programs where the correct answers are shown.

3. The program must contain all the information that the student will need in order to find or figure out the correct answer to every item.

4. Every item, instead of telling the student the correct answer, should tell him <u>how to get it</u>. For example, after items the answer to which cannot obtained by examining the item itself, the student should be referred to the earlier items in which the necessary information is provided, or where the required skill has previously been taught. Or, the program might provide the student with a hint that he can use if he needs it, the way a good tutor might give a hint. The hint could suggest the type of reasoning that might be applicable, remind the student where he did or learned something similar, or it might consist of a leading question.

5. The program must be tested without the answers, during the developmental stages, and revised on the basis of the results until students are able to get through the program without major problems. After several cycles of testing and revision of such a program, the program is usually quite effective. In testing, the very fact that typical students can successfully complete a program of this type constitutes proof, in and of itself, that the program is effective.

6. The medium chosen for the program, whether print, audio-lingual, manipulative, or other, should take into account the stimuli that the student will encounter, and the responses that will be called for in the situation for which the student is being trained.

Clearly, a program designed and developed to have these features requires more care and technical skill than the conventional types of programs.

Multiple Tracks - A Form of Branching

The idea that programmed texts could offer alternative tracks for different student populations and people who have different learning styles was first described by me in early 1961 (Mechner, 1961a).

The effectiveness of multiple tracking, or "express stops," depends on self-testing by the student. The student must diagnose his own knowledge to determine on which track he belongs, and his decision should have a compelling consequence for him, that is, he should be unable to continue if he incorrectly decides that he belongs on the fast track. Implementing this type of consequence requires omitting the correct answers.

A second requirement of successful multiple tracking is the use of self-diagnostic test items. The student should be able to check his understanding of the material rigorously when skipping ahead to those items. If he can answer those items correctly, he knows that he can skip the preceding material that led up to those items. If he fails those items, he knows that he needs to go through the preceding items. Self-diagnostic test items are therefore different from instructional items within a programmed sequence. In the case of a well-designed instructional item, as was explained earlier, the student almost always knows, with a fair degree of confidence, whether or not he got it right. In the case of self-diagnostic test items, he will generally need to check his answer, and the answers must therefore be provided.

Multiple tracking can significantly reduce boredom when learning from programmed instruction materials. It is not just a matter of saving time when different students have different degrees of knowledge and are therefore capable of "skipping." It is also a matter of different students being interested in different parts of the subject matter. A student may want to skip a section because the topic doesn't interest him. To make this genuinely possible, the program must tell the student whether skipping a section or topic will deprive him of a building block he will need later in the program, or whether it's okay to skip because skipping that block will have no effect on future progress. Without such knowledge, the student can never feel secure if he decides to skip something.

Different Types of Items

Professional writers of conventional text use many techniques for holding the reader's attention. They use varied sentence structure, variable formats and typefaces, dialogue, colorful examples, illustrations, etc. These types of devices are applicable to programmed instruction materials as well. But program writers often adopt the unjustified attitude that the student is a captive audience. Accordingly, they usually feel no need to be creative in holding the student's attention or to entertain him.

There are simple and effective ways to make programs more interesting. First of all, there should always be many different types of items: Short ones and long ones; items where the student makes a mark on a diagram; multiple choice items; items where the student is asked to write a word or a phrase; items where he is instructed to just think his answer; items where he is asked to review something by saying it to himself out loud; items where he is asked to turn back and re-read an earlier item; items called "test yourself," with a self-scoring key; and the numerous and varied formats available in audio-lingual programs.

The Teacher as Narrator

Every program should have continuous, running commentary in the form of instructions and side-comments to the student, coming from the programmer, who is cast in the role of the teacher. The purpose of this narration is to let the student know, on a continuous basis, where he is headed, the significance of what he is learning in relation to the subject matter as a whole, the relative importance of the various concepts being covered, good points to take a break if desired, where he should gear himself up for a major intellectual effort, etc.

Footnotes may direct him to additional information or elaboration regarding a certain topic or for an explanation of a certain term that was used and possibly glossed over, and other comments of such types that the student would find reassuring, interesting, or orientational.

Optional Inserts

Every program should have frequent, optional textual or pictorial inserts. These could be passages of conventional (unprogrammed) text (or narration) that go into more detail on a topic that was touched upon in the program. Such material would be optional, in the sense that the student could go on with the program without reading it (or listening to it in the case of audio-lingual programs), and he would be told so every time such material appears. But the student who is particularly interested in the topic, or just curious, could stop to read the text passage or come back and read it later. Visual illustrations are always welcome. Such interspersed material could be used to explain technical terms, describe interesting practical applications, and add historical notes or anecdotes that could provide spice.

Varied Typefaces

If one examines advertisements in newspapers and magazines, one sees creative use of typefaces, layouts, and graphics. Advertisers have learned that these devices help capture and hold the attention of non-captive audiences. The front pages of newspapers, which also face that challenge, use some of the same devices.

There is no reason not to use the same graphic techniques and devices in programmed instructional materials. For example, different types of items could use different typefaces and formats. The programmer's running commentary to the student could always be in a special typeface. The self-diagnostic items could always use a particular bold type. The interspersed optional test passages could be in small type, using a newspaper format. Historical perspectives and anecdotes could be in italics or boxes along the side of the page. Very important or critical items could be a in a typeface that signals their special character. Varied typefaces can always be supplemented with associated varied margins, special spacings between lines, boxes, and even varied colors if budget permits.

The consistent use of such graphics would make programs much more interesting and palatable to students.

RESISTANCE TO CHANGE

When I trained the original cadre of programmers at Basic Systems starting in

1960, the techniques I advocated were those described in my 1961 monographs "Introduction to Programming" and its five supplements (Mechner, 1961a). Those were the techniques and formats that became identified with programmed instruction in the educational technology community.

I soon became aware of some of the shortcomings of these techniques and started to develop the new techniques described in this paper. By the beginning of 1962, I was making strenuous efforts to persuade my programming staff of dozens of highly skilled programmers, as well as my professional colleagues at Basic Systems who had Ph.D.s in psychology, of the superiority and advantages of these techniques, and of their theoretical justification. But I was unable to modify a conception of the technology and a modus operandi that had by that time become entrenched. This failure was not due to anyone disagreeing with me on the intellectual level. It was due entirely to organizational inertia and unwillingness to change well-honed practices and public positions.

Resistance to the proposed new techniques came from various sources. First, there were Basic Systems' clients, consisting of several dozen major organizations like IBM, AT&T, the U.S. Air Force, Pfizer, Merck, Schering, and others. These clients had accepted Basic Systems' proposals for the development of large-scale training systems that involved programmed instruction, and they understood programmed instruction as consisting of the conventional techniques described in the original proposals that Basic Systems had presented to them. The Basic Systems sales staff was justifiably unwilling to ask these clients, who thought they had contracted for the best available techniques, to consent to become guinea pigs for technological experimentation. Furthermore, Basic Systems had produced and distributed various technical brochures and promotional materials describing the technology of programmed instruction in the conventional manner, and had hung its public identity on that description. Basic Systems' school market publisher Appleton Century Crofts, was already publishing and marketing Basic Systems programs that used the conventional techniques and formats. All of these factors acting together had effectively frozen the state of the art of programmed instruction for Basic Systems.

However, the main and strongest resistance came from Basic Systems' programming staff. By mid-1962 Basic Systems had about eighty highly trained programmers who had made significant intellectual commitments to the techniques I had taught them, and who were beginning to regard these techniques as their profession. For them, the prospect of having to change may have been too threatening. Such change may have implied admission of past error, and perhaps also provoked fear of the unknown. Their professional self-image was locked into the specific techniques they had learned, and did not require a continuing re-examination of the theoretical foundations of these techniques.

Even the managerial control I had over the programming organization and my status as Basic Systems' technological leader did not enable me to overcome these forces of resistance to change. The physicist Niels Bohr's said that it takes a new generation to adopt a new theory, because a new theory is rarely adopted just because it is superior to the old one. The same principle may apply to technology, with the substitution of "organization" for "generation."

Not until 1973, eleven years later, were the techniques described in this paper first implemented in published programmed instruction courses. These were produced by Behavioral Science Applications, Inc. for Pfizer Laboratories (Mechner, 1973) and the American Journal of Nursing (Mechner, 1974-75) and proved extremely successful both from the standpoint of teaching effectiveness and appeal to students.

THE FUTURE

It would be desirable to conduct research comparing programs developed in this manner with programs that use more conventional techniques. But even more important than such research is the further refinement of these techniques. The present paper does no more than point to some promising directions for technological innovation. These directions should be explored and developed with different media, response modes, subject matters, and target populations.

REFERENCES

Holland, J.G. & Skinner, B.F. (1961). The analysis of behavior. New York. McGraw-Hill.

Holland, J.G. (1967). A quantitative measure for programming instruction. American educational research journal, 4(2), 87-101.

Mechner, F. (1961a). Programming for automated instruction. http://www.mechnerfoundation.org/

Mechner, F. (1961b). Behavioral contingencies. In J.G. Holland, C. Solomon, J. Doran, & D.A. Frezza (Eds.), The Analysis of Behavior in planning instruction (pp. 50-62). New York. Addison-Wesley.

Mechner, F. (1967). Behavioral analysis and instructional sequencing. In P.C. Lange, (Ed.) Programmed Instruction, 81-103. Sixty Sixth Yearbook of the National Society for the Study of Education. Chicago, IL: University of Chicago Press.

Mechner, F. (1973). Bioavailability. A Pfizer Laboratories Publication. New York: Pfizer Laboratories.

Mechner, F. (1974-75). Programmed instruction courses in patient assessment. The American Journal of Nursing, Volumes 74-75. Reprints available from the American Journal of Nursing Company, Educational Services Division. New York: 10 Columbus Circle, NY 10019.

Margulies, S. (1963). Some general rules of frame construction. New York: Basic Systems.

Mechner, F. (1968). An overview of behavioral analysis. http://www.mechnerfoundation.org/

Rushton, E. (1963). The Roanoke story. Chicago: Encyclopedia Britannica Press.

Skinner, B.F. (1954). The science of learning and the art of teaching. Harvard Educational Review, 24, 86-97.

Skinner, B.F. (1958). Teaching machines. Science, 128, 969-977.