

Assessing Stem Cell Therapeutics in Murine Models

Giles Carlos, 54951701, gpcarlos@uci.edu

Nathan Gin, 54951701, nbgin@uci.edu

Alexander Nathaneal, 54951701, anathana@uci.edu

Vinh Nguyen, 54951701, vinhhn2@uci.edu

Owen Sitiabudi, 54951701, ositiabu@uci.edu

Github Repository: <https://github.com/gilescarlos/Stats-170-Cap-Stone>

1 Introduction and Problem Statement

Spinal cord injury (SCI) is a debilitating condition that impacts up to 300,000 individuals in the U.S. each year. As such, a significant focus of stem cell research is dedicated to developing therapeutics that could help bring affected individuals' livelihood back. Likewise, the tests and methods used to assess different therapeutics are incredibly important. The efficiency and rigor of these tests allow for discovery of novel therapies and help advance the research field dedicated to solving spinal cord injury. The ladder beam test is a behavioral method for assessing recovery following SCI in murine models. During the test, mice must walk across a ladder with fifty rungs. Injured mice are expected to have more missteps while recovering mice should have more plantar or sufficient steps. However, to properly utilize this test our sponsors must spend several hours analyzing hundreds of videos of mice completing the task in order to individually score each mouse's performance. Consequently, this tedious process tends to slow down the lab's workflow and some automation would provide a significant increase in efficiency. Streamlining the ladder beam task and its data analysis would allow for significant efficiency in retrieving data necessary for the advancement of SCI research and therapeutics. Our primary goal is to use appropriate machine learning algorithms and build off existing research to aid in the lab's data collection process for the ladder beam task. Secondly, we will assess the effectiveness of different treatments through appropriate statistical models and hypothesis tests.

2 Related Work

Several other labs have encountered this bottleneck in efficiency and have aimed to address this problem as well. One such solution is DeepLabCut¹, which is an open source software developed in Python that utilizes a deep neural network to track various body parts in multiple species across a broad collection of behaviors or physical assessments. Another study was able to utilize the software to conduct a comprehensive 3D gait analysis of mice after focal cerebral ischemia². The authors concluded that using a previously trained data set for tracking mice movement, their own recordings of mice completing the ladder beam task, and the DeepLabCut neural network provides accurate and sensitive data to describe the complex recovery of rodents following a stroke. Other researchers have also been successful in building upon DeepLabCut and creating their own toolbox utilizing a convolutional neural network appropriate for their recording equipment³.

¹Mathis, A., Mamidanna, P., Cury, K.M. et al. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat Neurosci* 21, 1281–1289 (2018). <https://doi.org/10.1038/s41593-018-0209-y>

²Weber, R.Z., Mulders, G., Kaiser, J. et al. Deep learning-based behavioral profiling of rodent stroke recovery. *BMC Biol* 20, 232 (2022). <https://doi.org/10.1186/s12915-022-01434-9>

³Aljovic, A., Zhao, S., Chahin, M. et al. A deep learning-based toolbox for Automated Limb Motion Analysis (ALMA) in murine models of neurological disorders. *Commun Biol* 5, 131 (2022). <https://doi.org/10.1038/s42003-022-03077-6>

3 Data Sets

We obtained our data from our sponsors at the UCI Stem Cell Research Center. Specifically, we received about a hundred videos of mice completing the ladder beam task and the corresponding manually labeled excel sheet corresponding to the videos.

Table 1: Example of First 10 Rows of Manually Input Data Set Obtained from Ladder Beam Task

animal	total_good	total_bad	ave_good	ave_bad	lb_score	type
1	276	24	46.00000	4.000000	92.00000	w
2	280	20	46.66667	3.333333	93.33333	k
3	255	45	42.50000	7.500000	85.00000	w
4	228	71	38.00000	11.833333	76.25418	k
5	267	33	44.50000	5.500000	89.00000	v
6	279	21	46.50000	3.500000	93.00000	v
7	276	24	46.00000	4.000000	92.00000	k
8	226	74	37.66667	12.333333	75.33333	w
9	227	73	37.83333	12.166667	75.66667	k
10(11)	222	78	37.00000	13.000000	74.00000	v

The post_injury data set has 30 rows and 7 columns where each column represents a single variable. The animal and type variables are categorical (nominal) and the rest of the variables are numerical (discrete).

Table 2: Example of First 10 Rows of Training Data from a Single Video

video	frame	bodypart	x	y
MVI_8324	img0234.png	front_left_paw	1058.571	673.2461
MVI_8324	img0234.png	front_right_paw	1137.828	613.8034
MVI_8324	img0234.png	back_left_paw	1124.619	1023.2978
MVI_8324	img0234.png	back_right_paw	1223.690	917.6218
MVI_8324	img0272.png	front_left_paw	1160.945	564.2678
MVI_8324	img0272.png	front_right_paw	1250.109	491.6156
MVI_8324	img0272.png	back_left_paw	1141.131	874.6909
MVI_8324	img0272.png	back_right_paw	1243.504	973.7622
MVI_8324	img0392.png	front_left_paw	1124.619	399.1491
MVI_8324	img0392.png	front_right_paw	1243.504	488.3132

The above table corresponds to a single training data set from one video of the ladder beam task being completed. From this particular video, nineteen frames were extracted which produced a data set with 76 rows and 5 columns. Each row represents a the location of one of the mouse’s paws during a particular frame from a specific video. The video column indicates what video the frames were extracted from. Likewise, the frame column indicates which particular frame the body part was being tracked. Lastly, the body part, x, and y column correspond to which paw, the x coordinate, and the y coordinate, respectively. As we obtain more videos from our collaborators we plan on having a much larger training set with multiple videos and more frames.

4 Overall Technical Approach

4.1 Data Wrangling

In order to utilize the videos provided by us, we needed an efficient tool for extracting frames in the video and labeling the location of mouse paws. We utilized DeepLabCut (DLC), an open source Python tool for developing models capable of tracking animal behavior and movement. The process for extracting our data from this tool goes as follows. First, we input a video or list of videos that we wish to obtain training data from into DLC. DLC then extracts a particular set of frames from each video according to the options we specify. Next, we individually label each frame by marking the location of each of the four mouse paws. Lastly, from those labeled frames DLC produces a training data set.

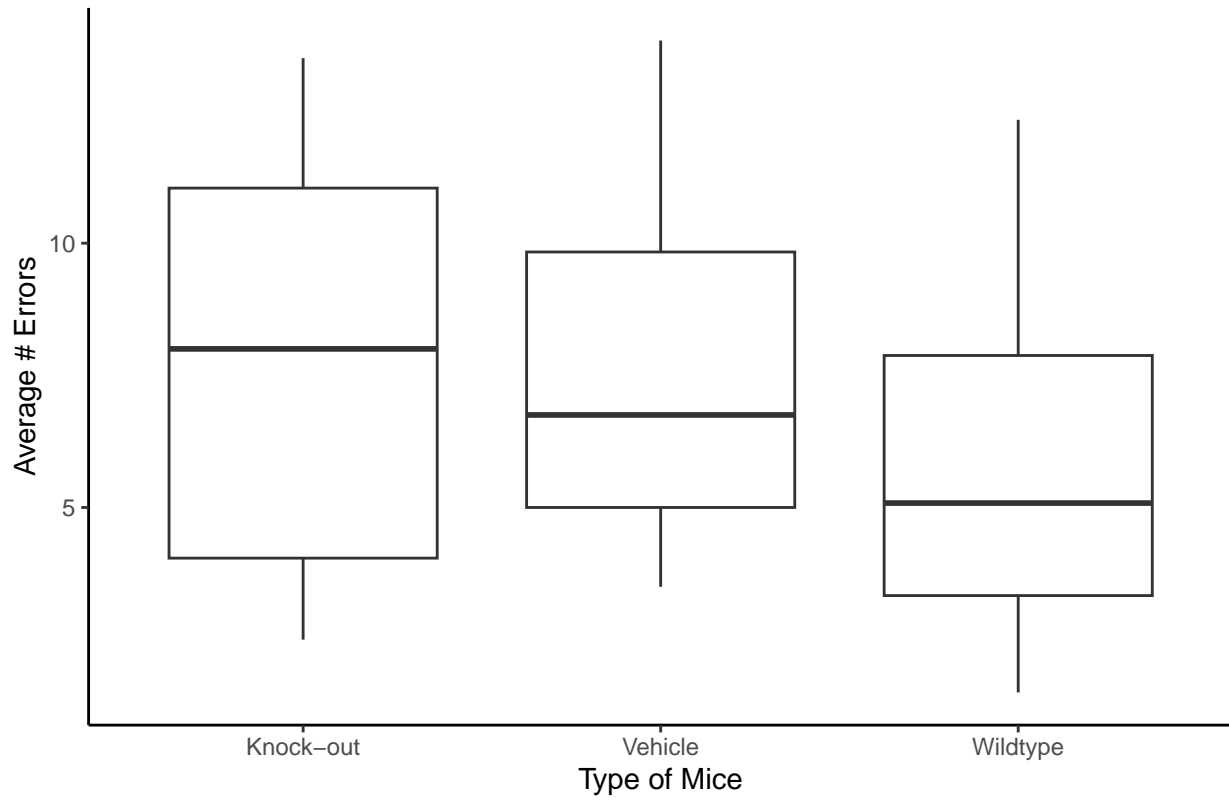
However, the training data must then be pre-processed since the outputted format is not efficient to work with. As shown in the table below, each frame corresponds to a single row but the names for the columns need to be cleaned. In order to clean the training dataset initially produced by DLC into the table described in the Dataset section, the column names had to be changed and rows could be broken down to represent multiple body parts. First, we removed the scorer’s name since it is irrelevant to the data. The columns containing the video file and the frame were given appropriate names. Then, each body part was given a correct column name. For example, bodypart1 corresponded to the front left paw of the mouse. Lastly, each individual row was transformed into four rows, one corresponding to each paw of the mouse at that specific frame.

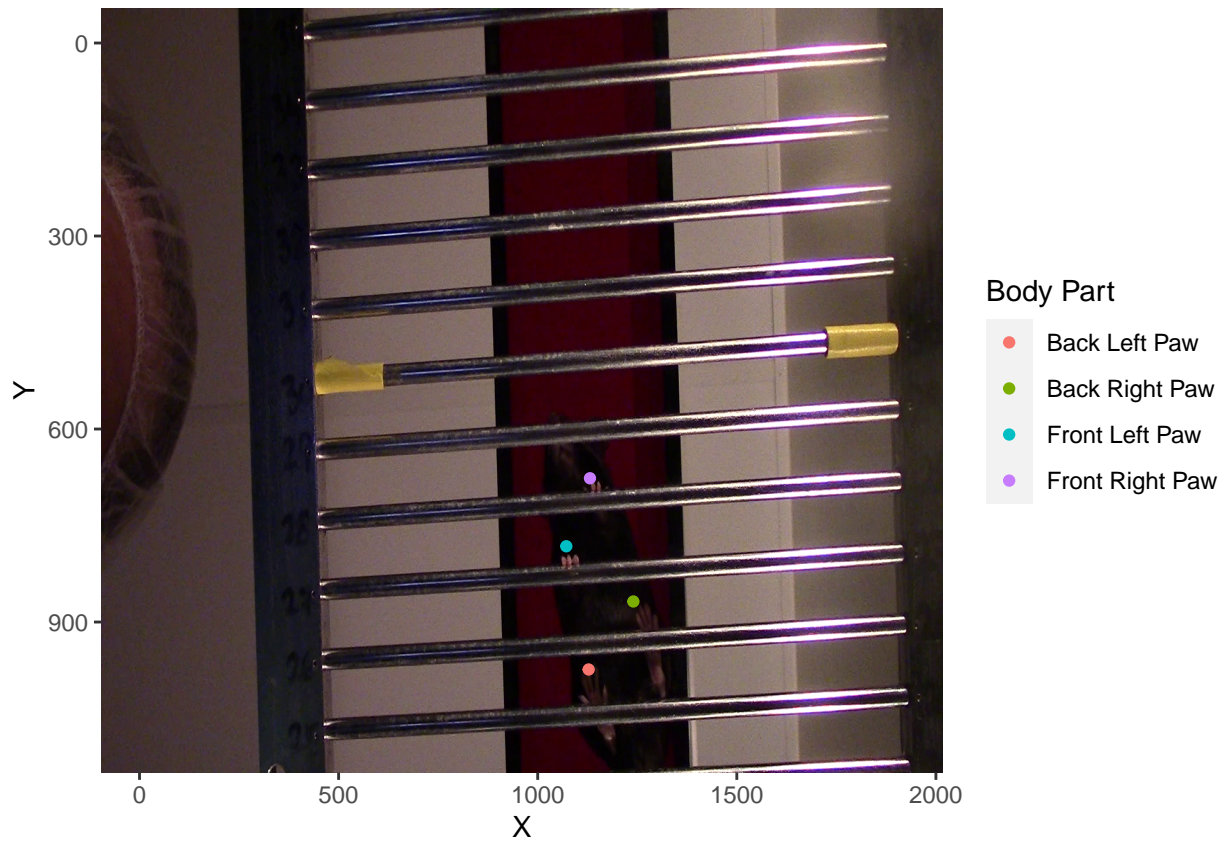
Table 3: Example of First 10 Rows and 5 Columns of Uncleaned Training Data from a Single Video

scorer	...2	...3	Giles...4	Giles...5
bodyparts	NA	NA	bodypart1	bodypart1
coords	NA	NA	x	y
labeled-data	MVI_8324	img0234.png	1058.5712525149595	673.2461470916205
labeled-data	MVI_8324	img0272.png	1160.9448379595071	564.2678141990375
labeled-data	MVI_8324	img0392.png	1124.6187269953127	399.1491279981543
labeled-data	MVI_8324	img0415.png	1137.8282218913835	418.96337034226025
labeled-data	MVI_8324	img0427.png	NA	NA
labeled-data	MVI_8324	img0435.png	1012.338020378712	240.63518924530675
labeled-data	MVI_8324	img0505.png	1121.3163532712952	134.95923007674116
labeled-data	MVI_8324	img0525.png	1121.3163532712952	286.86842138155373

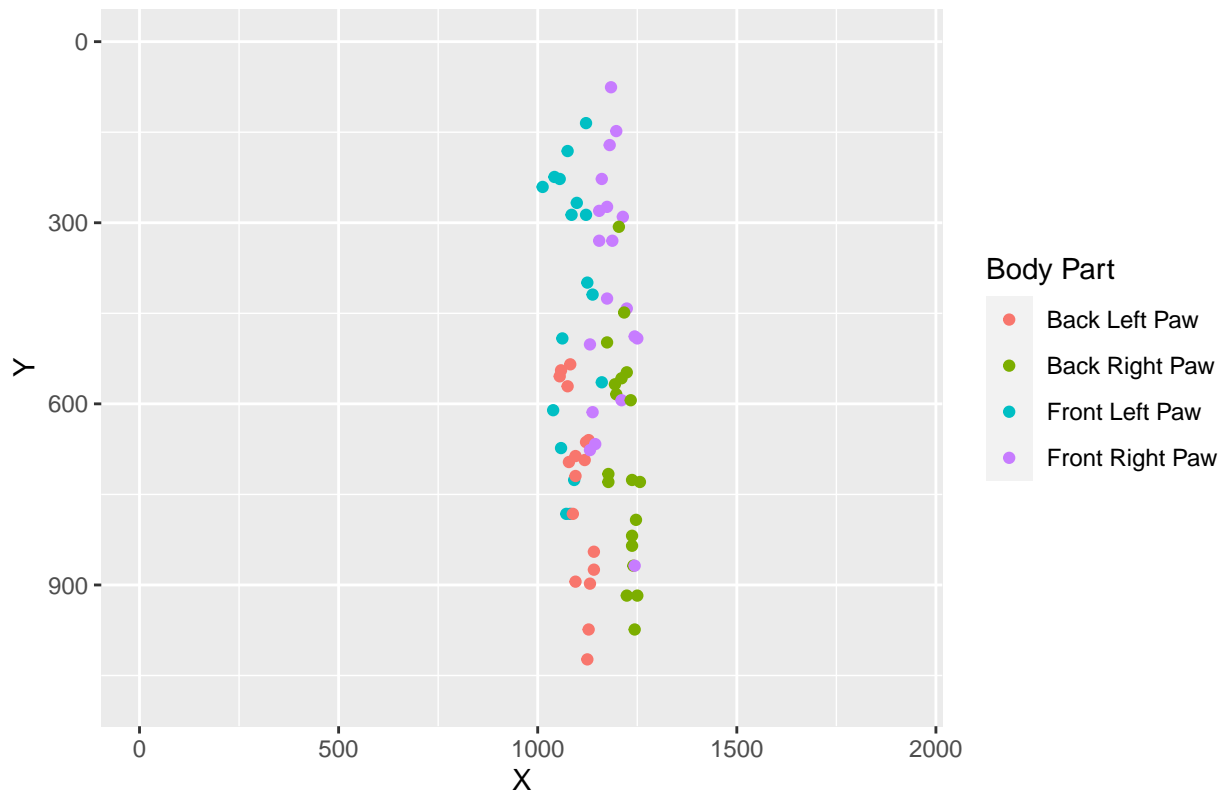
4.2 Exploratory Data Analysis

Distribution of Ladder Beam Performance Across Mice Groups





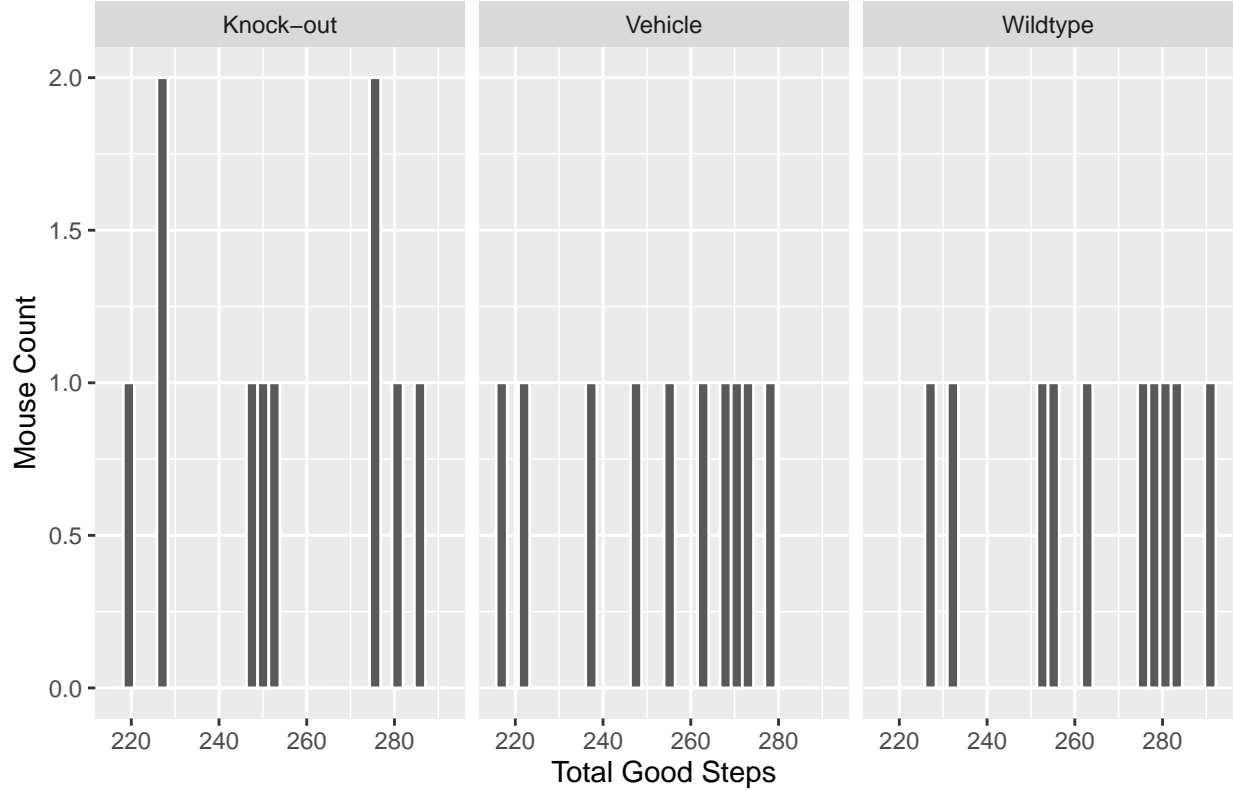
Tracking Individual Paws Throughout a Single Video

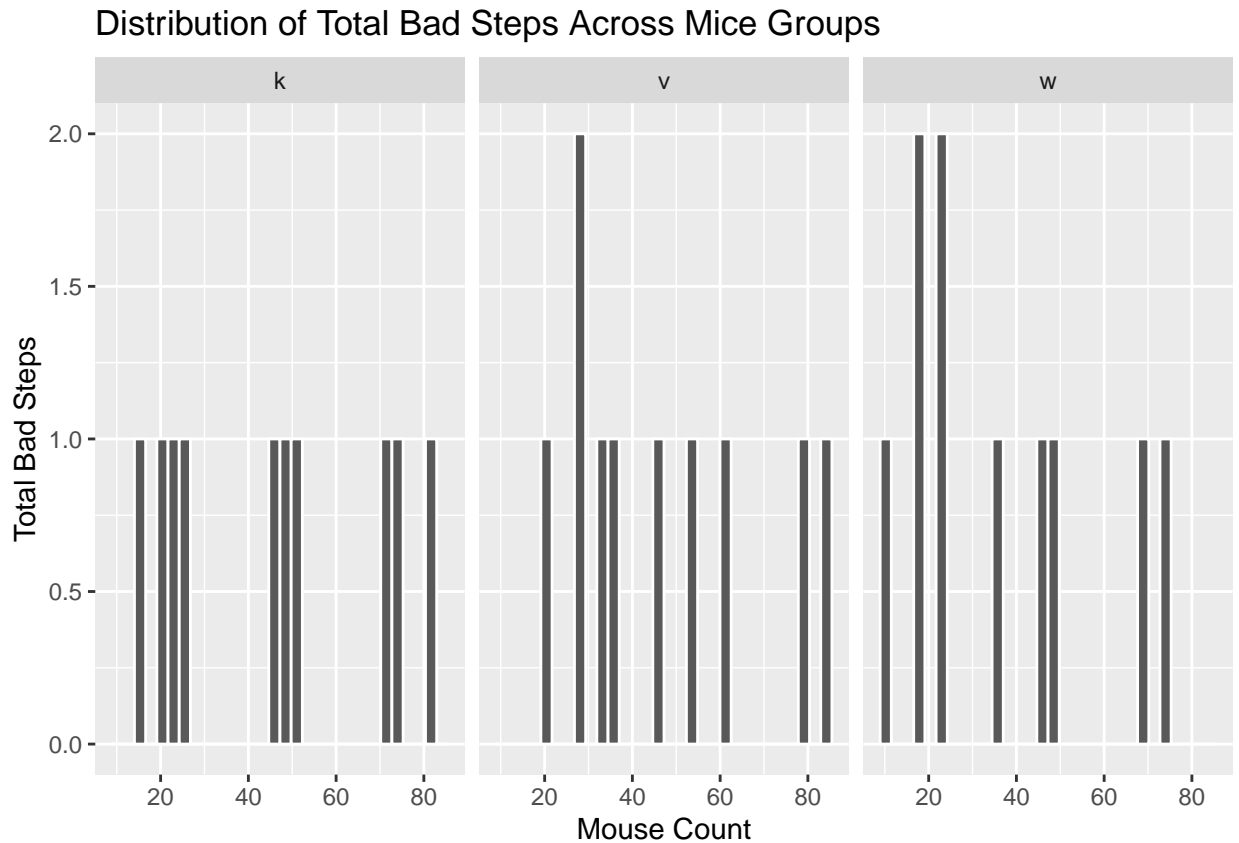


The two plots above, describe how the paws are tracked in a video and the general behavior of each paw. The

first plot, indicates that we are tracking each paw by emphasizing the location of the toes. The second plot describes the natural behavior one would expect for the paws of a mouse. That is, the front paws remain in front of the back paws for majority of the video. It must be noted that the recordings were not done with a stationary camera so although the mouse may have been moving forward, during the frames it may appear that the mouse is not moving that much.

Distribution of Total Good Steps Across Mice Groups





The above two graphs show the distribution of the total good and bad steps across the mice groups.

5 Software

6 Experiments and Evaluation

7 Notebook Description

8 Members Participation

Table 4: Percentage of workload across group members

Task	Giles	Nathan	Alex	Vinh	Owen
task 1	100%	100%	100%	100%	100%

9 Discussion and Conclusion

What did you learn about the methods and algorithms you worked with? What did you learn about their strengths? And their limitations?

test text

What ended up being harder than you expected in your project? What was surprising about your project?

test

What other lessons did you learn, expected or unexpected (e.g., perhaps about the tools you used, if you used anything out of the ordinary?

test text

If you were in charge of a research lab, what ideas and directions might you invest in over the next year or two to try to make major progress on this problem? Feel free to be speculative in discussing possible future directions.

test text