

BiasBench: A reproducible benchmark for tuning the biases of event cameras

Supplementary Material

7. Access to the Dataset

The dataset, the presented metrics as well as some additional information is accessible on our project website: <https://cogsys-tuebingen.github.io/biasbench/>

8. Datasets

Here, we provide some additional information about the three setups of our dataset. The spinning dot in Sec. 8.1, the blinking LED board in Sec. 8.2, and the Visual Odometry (VO) in Sec. 8.3.

8.1. Spinning Dot

The setup of the spinning dot is shown in Fig. 11. A selec-

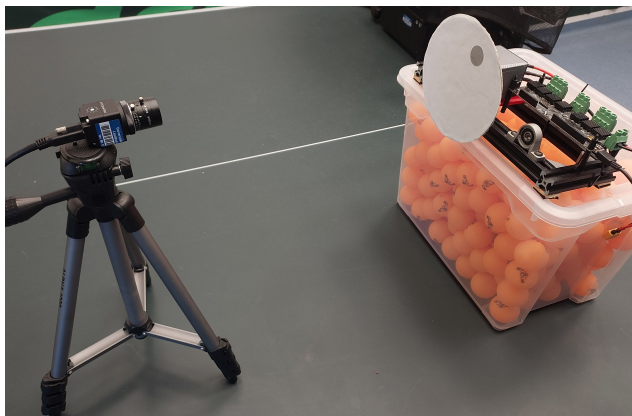


Figure 11. The setup of how we recorded the spinning dot consists of an EVKv4 event-based camera from Prophesee with a resolution of 1280×720 pixels and a white disk with a grey dot connected to a motor. The distance between the motor and the spinning disk is 40 centimeters.

tion of screen captures from the recordings of the grey dot with their bias settings is visualized in Fig. 12. And for the black dot with their bias settings in Fig. 13.

8.2. Blinking LED Board

A blinking LED board, shown in Fig. 14, was specifically designed as an evaluation tool for event cameras in a scientific study. Its purpose is to enable the simultaneous observation of varying frequencies and luminosity gradients, offering a robust testing environment for evaluating event camera performance under diverse conditions.

The board consists of a total of 16 LEDs, categorized as follows:

- **High-Frequency Blinking LEDs (4 LEDs):** These LEDs blink at high frequencies that are not perceivable by the human eye (>50 Hz). They are particularly sensitive to the *bias_hpf* bias setting, as higher cutoff frequencies in the high-pass filter will predominantly retain events from these high-frequency LEDs while filtering out lower-frequency signals.
- **Sinusoidal Luminosity LEDs (6 LEDs):** These LEDs modulate their brightness following a sinusoidal wave at different frequencies. The smooth, continuous change in luminosity creates a distinct time gradient in the events generated by the camera.
- **Triangular Wave Luminosity LEDs (6 LEDs):** These LEDs also change their brightness at the same set of frequencies as the sinusoidal LEDs, but with a triangular wave pattern. The sharper transitions of the triangular wave offer a different type of temporal gradient for evaluation.

The difference in frequencies across the sinusoidal and triangular LEDs results in varying temporal gradients, allowing for a nuanced assessment of how the event camera responds to different rates of change in light intensity.

The board’s design also allows for testing bias tuning algorithms:

- Adjusting the *bias_diff_on* and *bias_diff_off* parameters can suppress the events generated by low-frequency LEDs, effectively isolating high-frequency activity.
- The *bias_fo* setting provides additional filtering by attenuating signals from lower frequency LEDs, enabling more precise control over the frequency response of the camera.

The LED board is controlled by a microcontroller, allowing easy adjustment of LED frequencies. While Pulse Width Modulation (PWM) is commonly used for LED intensity control, the event camera would detect the rapid on-off transitions. To avoid this, capacitors were added in parallel to the sinusoidal and triangular LEDs, acting as integrators to smooth the PWM signal into a continuous analog waveform. This ensures gradual brightness changes and prevents unwanted high-frequency noise.

We recorded 30,976 sequences, covering the following biases:

- *bias_diff_on*: -80, -60, -40, -20, 0, 20, 40, 60, 80, 100, 120
- *bias_diff_off*: -30, -10, 10, 30, 50, 70, 90, 110, 130, 150, 170
- *refr_time*: 4, 20, 36, 52, 68, 84, 100, 116
- *refr_fo*: -29, -18, -7, 4, 15, 26, 37, 48
- *refr_hpf*: -15, 65, 145, 225

Figure 15 shows the average ER over the whole dataset

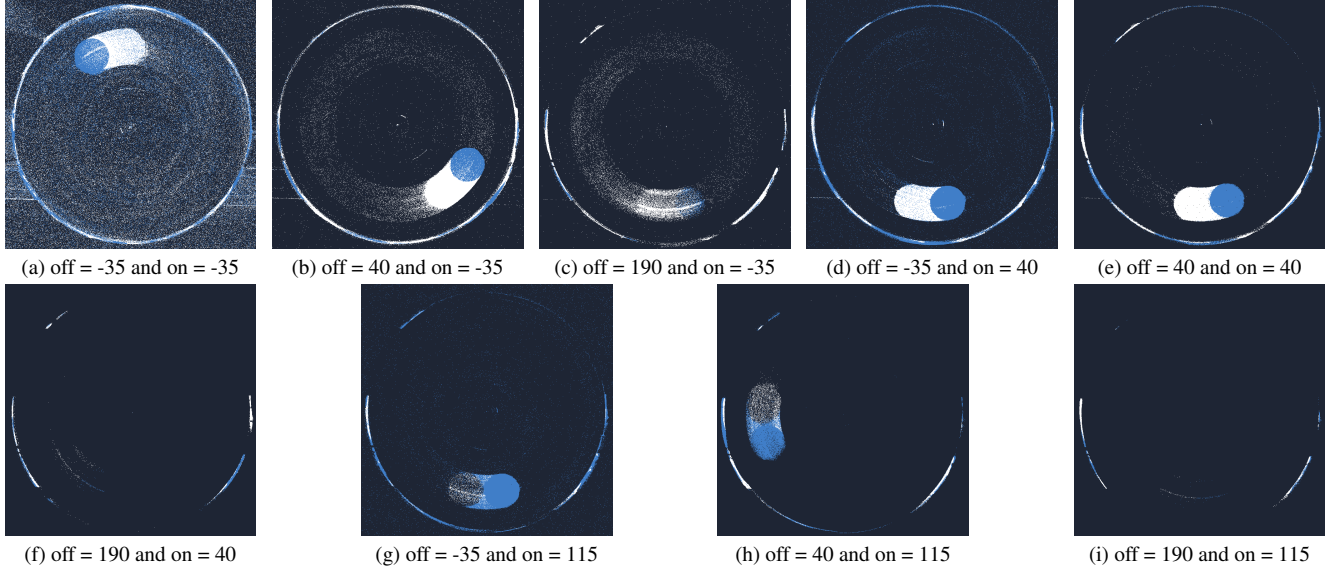


Figure 12. A selection of screen captures of Metavision Studio from 100 recordings of a **grey dot** with their bias settings, where off refers to *bias_diff_off* and on refers to *bias_diff_on*.

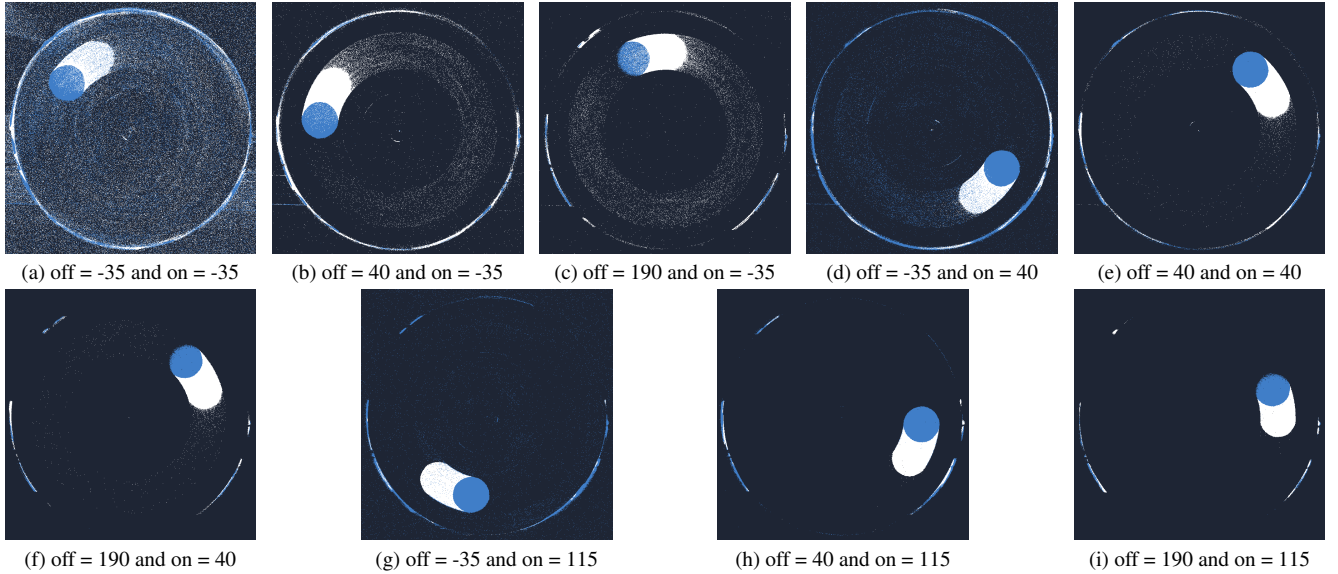


Figure 13. A selection of screen captures of Metavision Studio from 100 recordings of a **black dot** with their bias settings, where off refers to *bias_diff_off* and on refers to *bias_diff_on*.

for each bias setting. The bias combinations with the lowest value for one of the two thresholds contain around 50% of the total event count. Additionally, a very high high-pass filter can be used to get rid of many events. A decrease in the refractory period also decreases the average event rate as displayed in Figure 16.

The slow-blinking LEDs start to vanish for high bias thresholds. Resulting in an increased RFU in these regions. This can be seen in Fig. 17. The increase of noise in low

bias setting also hinders the frequency detection, therefore, an optimal around $bias_diff_off = -20$, $bias_diff_on = 20$ results in the highest average RFU. A strong high-pass filter in combination with a strong low-pass filter also decreases the sensitivity for the detection of some LEDs, therefore, the RFU also increases in these regions significantly.

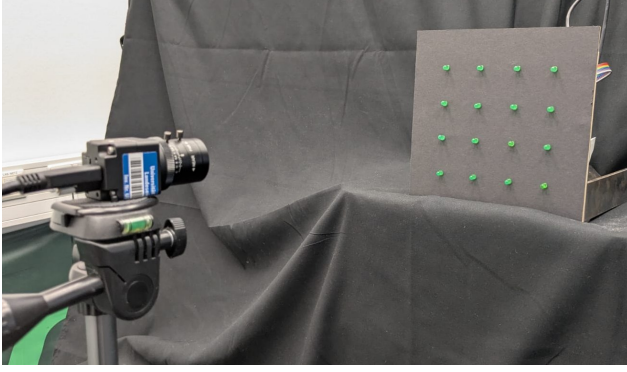


Figure 14. A blinking LED board consisting of LEDs with different frequencies and waveforms.

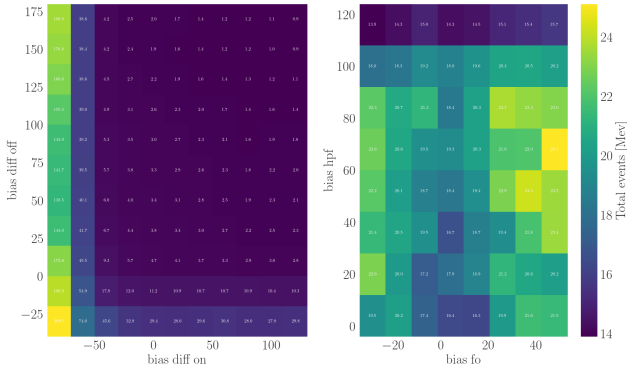


Figure 15. The average event Rate for different bias combinations. *bias_diff_off* vs *bias_diff_on* on the left side, *bias_hpf* vs *bias_fo* on the right side. The lowest threshold bins contain roughly 50% of all events produced.

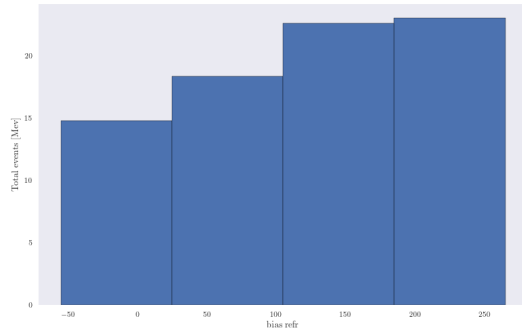


Figure 16. The average event Rate for different refractory periods.

8.3. Visual Odometry (VO)

To get a repeatable motion and an accurate ground truth trajectory, we mounted an event camera to the end-effector of a Pandas robot arm, shown in Fig. 18, to move the event camera along the same trajectory with different biases. The

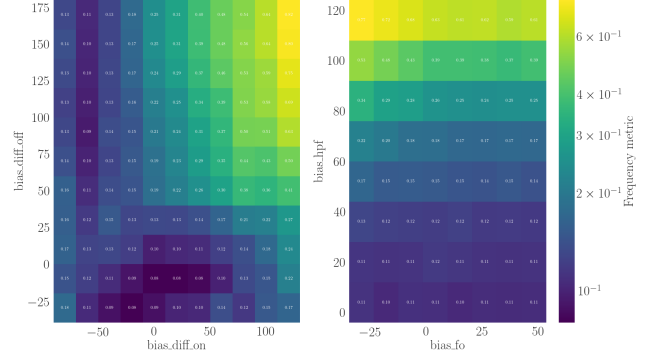


Figure 17. The average RFU for different bias combinations *bias_diff_off* vs *bias_diff_on* on the left side, *bias_hpf* vs *bias_fo* on the right side. The most efficient frequency estimation can be achieved with a *bias_diff_off* = -10, *bias_diff_on* = 20 and *bias_hpf* = 0.

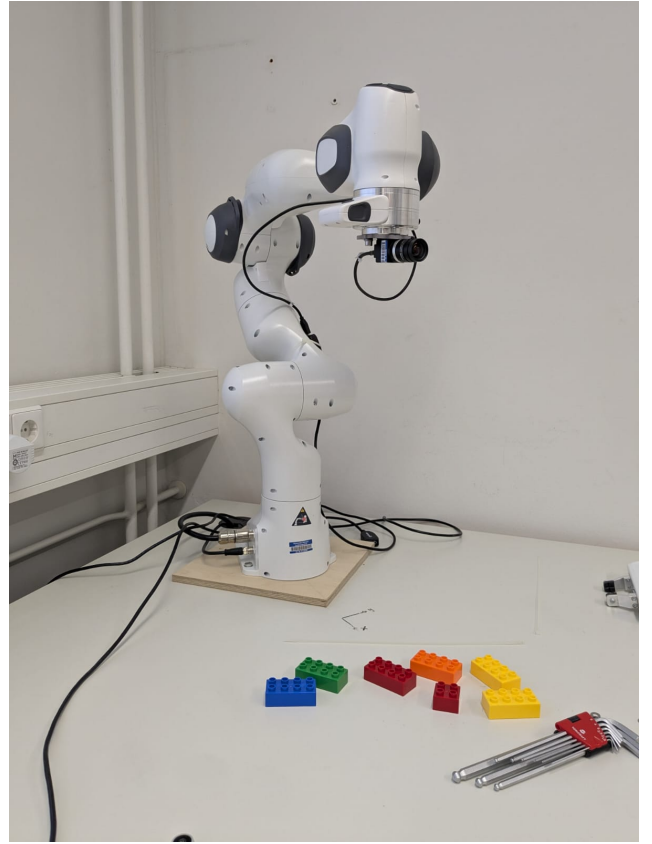


Figure 18. The setup to record the VO dataset.

scene was a static environment under constant illumination.

9. Baseline

The overall pipeline used for this learning process is illustrated in Fig. 19.

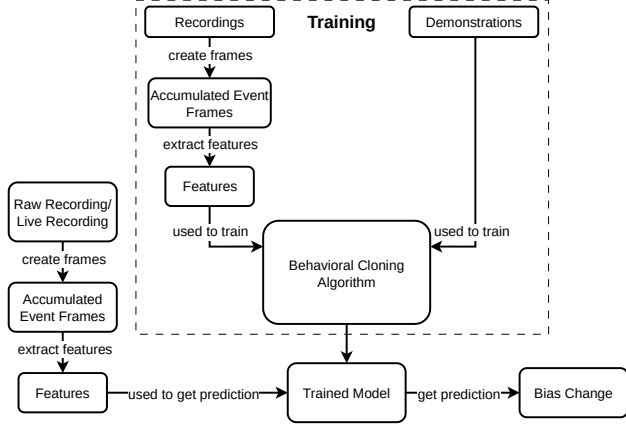


Figure 19. An overview of the Behavioral Cloning (BC) pipeline.

In order to rate the output of the event camera, the primary metric we chose was whether the dot on an accumulated event frame is visible to the human eye or not. This is the first and foremost concern because if the dot is not visible, it defeats the entire point of the recording. The secondary metric that we chose is the ER, which should be as low as possible while keeping the signal strong. This translates to low background noise. We define an optimum bias setting as any setting that fulfills the two conditions.

The average ER was calculated for each recording by dividing the total amount of events by the length of the recording. The ER is used to measure the amount of signal and noise and to try to find a bias setting with a good ratio between the two. As can be observed in Fig. 20 and in Fig. 21, for recordings where either of the two biases has a small value, the ER is larger, by order of magnitude, compared to the recordings where the biases had higher values. The recordings with the *bias_diff_on* values between -85 and -60 contain errors because the readout bus of the event camera did not have enough bandwidth to transmit all the events. This means that the ER is bigger than the maximum allowed one.

The most noticeable difference between the heatmap of the ER for the grey dot in Fig. 20 and the one for the black dot in Fig. 21 is that the average ER is higher for the black one. This is expected, as the contrast between the black dot and the white background is higher than for the grey dot.

To find the range of the optimal biases, we also look at the accumulated event frames. An accumulation period of 8 ms has been chosen so that the entire body of the dot can be seen. The first thing noticeable is that for high values of *bias_diff_off*, the dot is no longer visible, as seen in Fig. 12. The camera has such a low sensitivity for those settings that the contrast between the dot and the background is no longer enough to produce any events. As we observed from the ER, for too small values of both biases, especially

if they are negative, the noise severely affects the quality of the recording, even if the dot is visible.

For the grey dot, an optimal bias, as we defined it, is *bias_diff_off* = 40 and *bias_diff_on* = 40. For this setting, the dot is visible, and the noise is greatly reduced. We have considered that for the setting *bias_diff_off* = 40 and *bias_diff_on* = 115, the signal was too reduced for it to be considered an optimal bias, even though the dot is visible to the human eye. The observations made from viewing the recordings helped us conclude that optimal biases can be found in the range $[15, 65]$ for *bias_diff_off* and in the range $[40, 90]$ for *bias_diff_on*.

For the black dot, we can observe that the dot is visible even for the maximum values of the thresholds, while the grey one disappears. Because of this, the range of the optimal biases is larger than for the grey dot, being $[40, 165]$ for *bias_diff_off* and $[40, 115]$ for *bias_diff_on*.

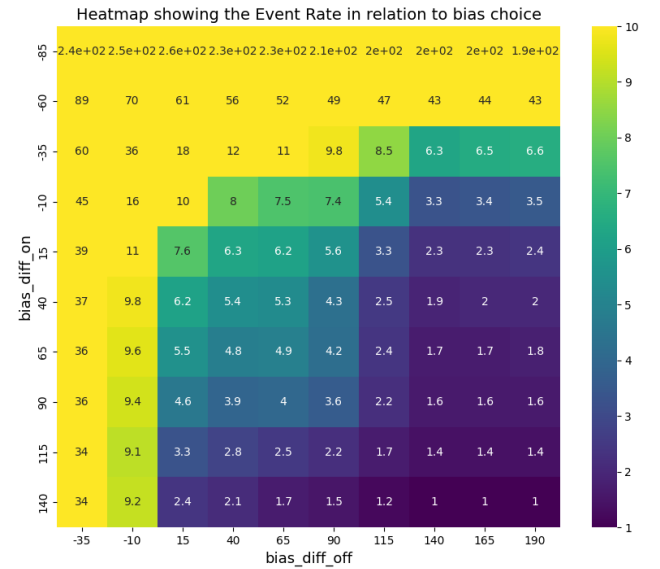


Figure 20. A heatmap showing the Event Rate (ER) for different bias choices of the spinning grey dot recording.

We have chosen to define the action in this environment as the relative bias change, expressed with an integer to be added to the current bias value, as this is the way we went about tuning the biases by looking at the output and approximating how big of a change is necessary. Thus, we have written demonstrations consisting of a tuple of actions with the form $(a, b, 0, 0, 0)$, where a represents the change of *bias_diff_off* and b represents the change of *bias_diff_on* that is needed to reach the range of optimal biases.

Different Feature Extractors As an ablation study for the feature extraction, we have adapted two models of ResNet: ResNet18, with 18 layers, and ResNet50, with 50 layers, a more complex network, capable of capturing more

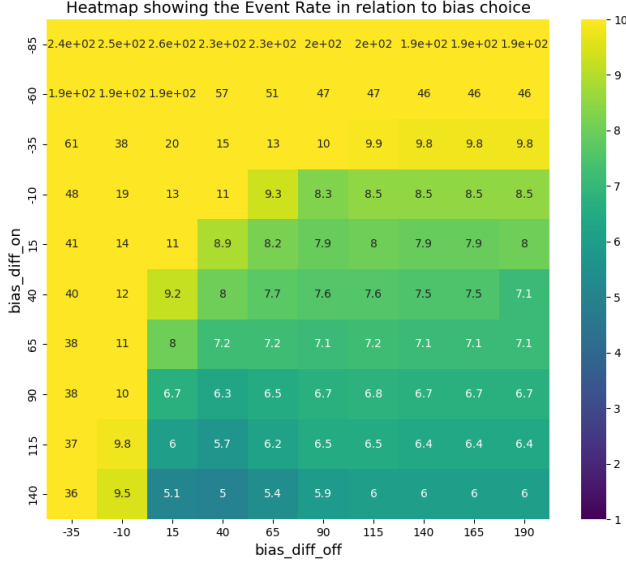


Figure 21. A heatmap showing the Event Rate (ER) for different bias choices of the spinning **black** dot recording.

information. The choice of the network has an influence on the ability of the network to learn.

ResNet50 is a more complex and deep network compared to ResNet18, and therefore, the BC model is able to finish the learning process in 1000 steps. Due to its more complex structure, ResNet50 creates a more information-dense latent space. The downside of this is a more computationally expensive training process for the model, which was already done on ImageNet and is of no concern for this work. The exact loss values after the training process finished are listed in Tab. 6.

As evident from the training and test loss in Tab. 6, ResNet50 is the better option for our purposes, and therefore, we decided to use ResNet50 as our feature extractor.

Model	Test loss
ResNet18	0.24
ResNet50	0.2

Table 6. The final loss values when the training process is finished. The loss is displayed as the Mean Squared Error of the normalized proposed action.

Different Accumulation Times As described in Sec. 4.1, we are using accumulated event frames to extract features from the event stream. In an additional ablation study, we investigated the influence of the accumulation time. Depending on the application, the duration of the accumulation time can have a significant influence on the performance.

Therefore, we investigated the influence of different accumulation times (1 ms, 2 ms, 8 ms, and 16 ms).

Accumulation Time	Test loss
1 ms	0.2
2 ms	0.22
8 ms	0.21
16 ms	0.27

Table 7. The final loss values with different accumulation times after the training process finished. The loss is given in the mean squared Error of the normalized proposed action.

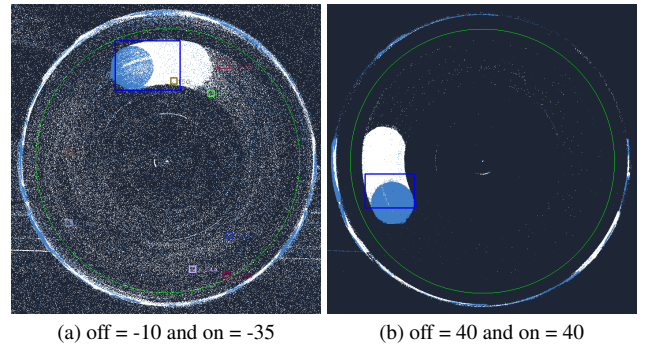


Figure 22. Demonstration of the model proposing a bias change from (a) to (b) that allows the object tracking algorithm to accurately track the dot and not see any other objects caused by noise.

From the final test losses in Tab. 7, we can see that for our data, a shorter accumulation time is favorable.