Supplemental Material to Another Approximation of the First-Passage Time Densities for the Ratcliff Diffusion Decision Model

Kendal Foster

University of Warwick

Henrik Singmann

University College London

University of Warwick

Author Note

Correspondence: kendalfoster@gmail.com (Kendal Foster) or singmann@gmail.com (Henrik Singmann)

The software developed for this paper, `fddm`, is available at: https://cran.r-project.org/package=fddm

The source code of `fddm` as well as the code to recreate all results and figures is available at: https://github.com/rtdists/fddm

1 Introduction

This document provides additional analysis to supplement the analysis done in Section 5: Benchmark Testing in the `fddm` paper. In particular, this includes benchmark tests performed on alternate predefined parameter spaces and more details on the differences between the two k_s functions, k_s^{Nav} and k_s^{Gon}.

The structure of this supplemental document mirrors that of Section 5: Benchmark Testing in the `fddm` paper, however it includes a separate section for providing the parameter spaces used in Section 2. Section 3 shows complementary results to the benchmark tests done in Section 5.2 of the `fddm` paper. In Section 4, we discuss the empirical differences between the two k_s functions, k_s^{Nav} and k_s^{Gon}; Section 5.3 of the `fddm` paper refers to this additional analysis. Lastly, we provide alternate benchmark tests in Section 5 to complement those found in Section 5.3.1 in the `fddm` paper.

2 Tables of Parameter Spaces

This section includes the two parameter spaces that were used in the analysis in the `fddm` paper. Consequently, we use the same parameter spaces in this supplemental analysis. Table 1 and Table 2 in the `fddm` paper correspond to Table T-1 and Table T-2 in this supplemental document, respectively. In addition, we include Table T-3 because that is the parameter space that we used when collecting the benchmark data for k_s^{Nav} and k_s^{Gon} in Section 4 of these supplemental materials.

3 Determining the Default Behavior of our Heuristic Switching Mechanism, δ

In the `fddm` paper, we ran benchmark tests to determine the optimum value of δ, the keystone to our mechanism for switching between the SWSE "small-time" approximation and the "large-time" approximation given by Navarro and Fuss (2009). Those benchmark tests were run on the parameter space with response times ranging from 0 to 30 seconds, a rather wide range for the DDM. In Figure S-1, we present the

Table T-1

Parameter space with response times ranging from 0 to 30 seconds, used for determining δ.

Parameter	Values
t	0.001, 0.1, 1, 2, 3, 4, 5, 10, 30
a	0.25, 0.5, 1, 2.5, 5
v	-5, -2, 0, 2, 5
w	0.2, 0.5, 0.8
η	0, 0.5, 1, 1.5

Note. Parameter t_0 is fixed to 0.0001 for consistency with the other reported benchmarks.

Table T-2

The parameter space used in the benchmark tests.

Parameter	Values
t	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0
a	0.5, 1, 1.5, 2, 2.5, 3, 3.5
v	-5, -2, 0, 2, 5
w	0.3, 0.4, 0.5, 0.6, 0.7
η	0, 1, 2, 3.5

Note. The parameter t_0 is fixed to 0.0001 because this parameter must be greater than zero for the density function from the `RWiener` package.

same benchmark test as in Figure 4 from the `fddm` paper; but instead, we use the parameter space found in Table T-2, where the response times range only from 0 to 2 seconds. Still, the results favor $\delta = 1$ as the default value.

The second benchmark test to determine the optimum value of δ measured the execution time of a data fitting routine that used different values of δ. We only presented the benchmark times when using each value of δ in the `fddm` paper and not the number of function evaluations. Akin to Figure 10 and to accompany Figure 5 from the `fddm` paper, Figure S-2 presents these results. With the previously known exception

Table T-3

The parameter space used in the benchmark tests specifically for k_s^{Nav} and k_s^{Gon}.

Parameter	Start Value	Stop Value	Step Size
t	0.1	10	0.1
w	0.1	0.9	0.1
ϵ	10^{-10}	10^{-2}	10^{*}

Note. *This step size is multiplicative instead of additive.

of $\delta = 0$, there are mostly small differences in the number of function evaluations across the other values of δ. There is, however, a slight preference for $\delta = 1$, and this further confirms our default choice of $\delta = 1$.

4 k_s Calculations

In Section 5.3.1 of the `fddm` paper, we showed the results of benchmarking the implementations of the density function approximations across a predefined parameter space. One of the surprising results was that our implementation of the "small-time" approximation method from Navarro and Fuss (2009) was faster than that of Gondan et al. (2014), despite the approximation method from Navarro and Fuss (2009) being older. In Figure 7 in the `fddm` paper, we showed a plot of the differences in the k_s^{Nav} and k_s^{Gon} calculations for one value of w, the relative starting point of the DDM. In Figure S-3 in this document, we offer four similar plots, but with different values of w. In these plots, we can see a similar pattern that k_s^{Nav} is generally better for larger response times. Moreover, when varying w, k_s^{Gon} requires fewer terms than k_s^{Nav} for only very large values of w (around $w = 0.9$).

Another potential explanation for the older k_s^{Nav} outperforming the more recent k_s^{Gon} is that the calculation of k_s^{Nav} itself was simpler than that of k_s^{Gon}. In other words, it could take less time for a computer to calculate k_s^{Nav} than it would take to calculate k_s^{Gon}. We mentioned this in the `fddm` paper but obtained conflicting results depending on which computer we used, and so we were unable to include these results in the main

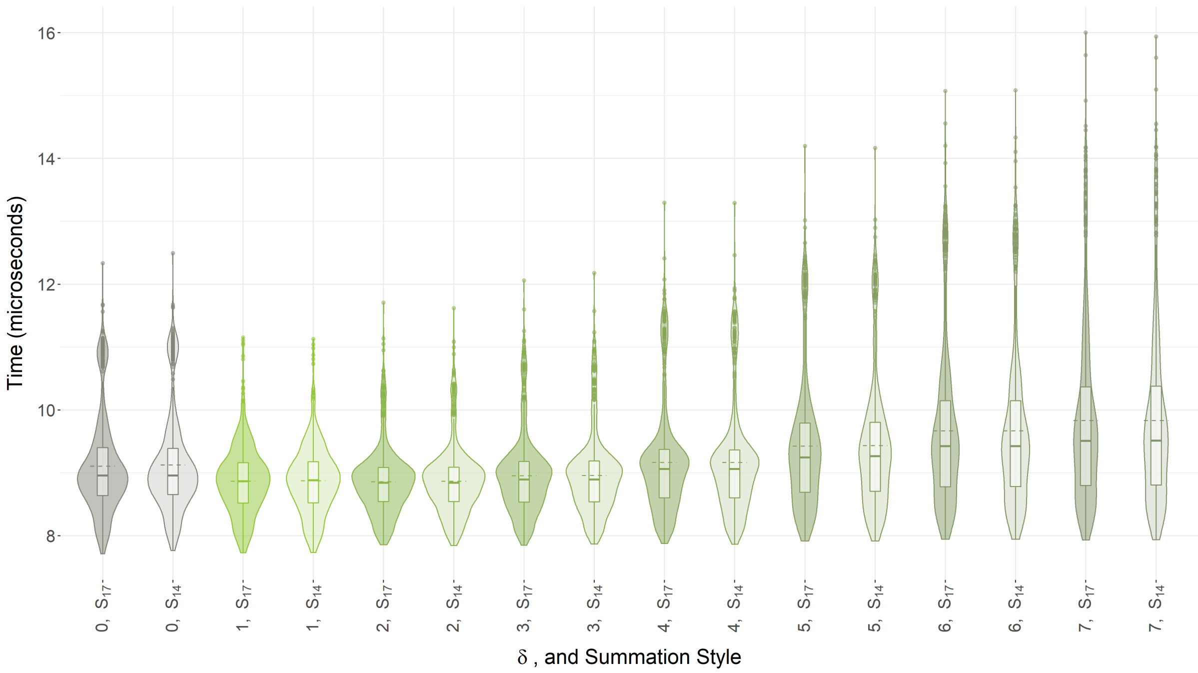

Figure S-1. Effect of δ on the benchmark times for the “combined-time” SWSE implementation and selected response times. The horizontal axis shows the combination of the chosen value of δ and the summation style used, and the vertical axis shows the benchmark time. The violin plot shows a mirrored density estimate; overlaying the violin plot, the boxplot shows the median in addition to the first and third quartiles; the horizontal dashed line shows the mean. The response times used as inputs to the implementations were between 0 and 2 seconds and input to the density function as a vector; the full parameter space can be found in Table T-2. This figure is the counterpart to Figure 4 in the `fddm` paper.

paper. However, for the sake of completeness, we include the benchmark results from running k_s^{Nav} and k_s^{Gon} on the parameter space described in Table T-3.

5 Benchmarking All Implementations

Section 5.3 in the `fddm` paper shows results of selected benchmark tests across two parameter spaces. While we believe those results included in the main paper are the most pertinent, we include the results of similar benchmark tests here.

5.1 Benchmark Tests on Predefined Parameter Spaces

First, we will present the results of running similar benchmark tests to Figure 6 in the `fddm` paper. These benchmark tests were performed with the same implementations,

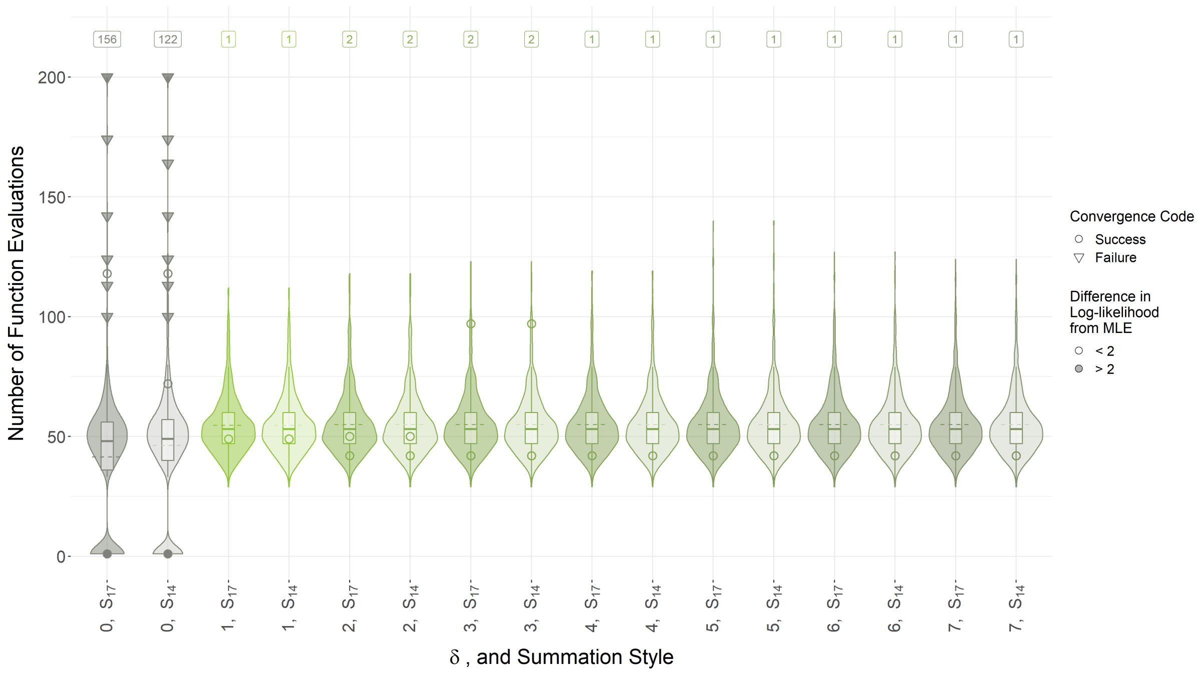

Figure S-2. Effect of δ on the number of calls to the likelihood function for the "combined-time" SWSE implementation during data fitting. Each distribution shows results from 407 fitting runs (the 37 participants of Trueblood et al., 2018 times 11 fixed sets of starting values). All visible points have a difference in log-likelihood greater than 0.0001 compared to the maximum likelihood estimate for that participant; above each violin plot is the total number of such data points. "Convergence Code" in the legend refers to the return value provided by the optimization algorithm `nlminb`, where 0 indicates successful convergence and 1 indicates failed convergence. More details on the data fitting process are included in Section 5.1.2 of the `fddm` paper, and more details on the graphical elements are given in Figure S-1. This figure is the counterpart to Figure 5 in the `fddm` paper.

but either on a different parameter space or inputting the response times differently.

Figure S-5 shows the results of benchmark tests that use the same parameter space as in Figure 6 of the `fddm` paper, but inputs the response times individually instead of inputting them as a vector; the parameter space can be found in Table T-2. Inputting the response times individually decreases the proportion of of runtime in `C++`; thus, the proportion of overhead from function calls in `R` is increased. Moreover, we recorded the median benchmark time of only 1,000 function calls instead of 10,000 function calls as we did for the vectorized inputs because the response times were input individually and the nested loops in `R` would take much longer to run. The reduced

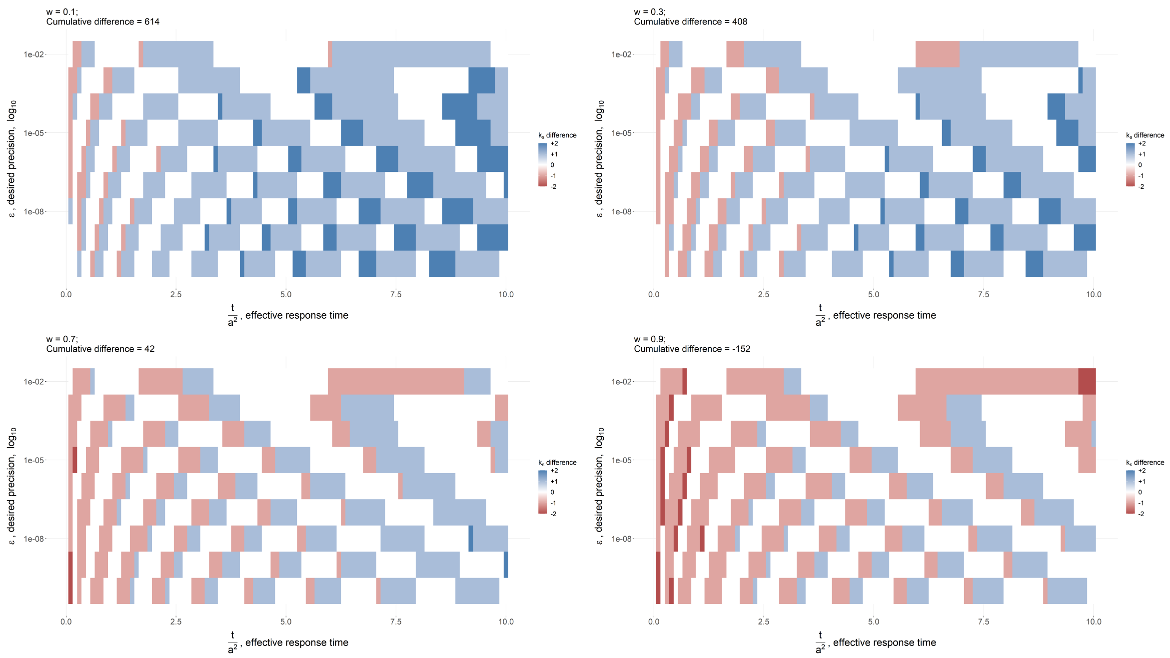

Figure S-3. Difference in number of terms for existing "small-time" approximations. The heatmap shows the differences in k_s^{Gon} and k_s^{Nav} across varying response times and error tolerances. Positive (blue) values indicate that k_s^{Nav} uses fewer terms, and negative (red) values indicate that k_s^{Gon} uses fewer terms. Note that k_s^{Gon} uses the DDM parameter w in its calculation, but k_s^{Nav} does not. In each of these plots, w varies from 0.1 to 0.9, to supplement Figure 7 in the `fddm` paper, where $w = 0.5$ in the main paper.

number of function calls results in a larger impact of outliers on the benchmark data. The proportionally increased overhead in `R` and increased impact of outliers causes the differences across the implementations to be much smaller than when using vectorized inputs. Still, the SWSE "combined-time" and "small-time" implementations marginally outperform the other implementations available in the `dfddm()` function.

Similarly in Figure S-6, we show the results of benchmark tests where the response times were input individually but the larger parameter space was used; the parameter space can be found in Table T-1. See the previous paragraph for explanations on the small differences across implementations available in the `dfddm()` function. Again, the SWSE "combined-time" and "small-time" implementations marginally outperform the others.

In Figure S-7, we show the results of benchmark tests where the response times

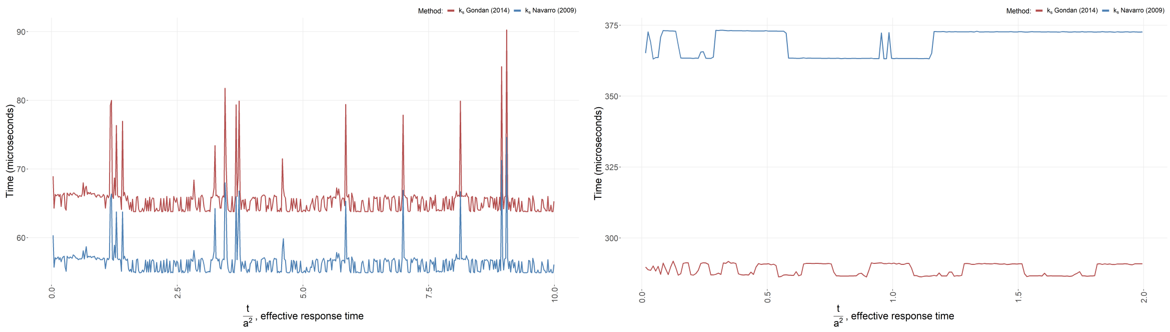

Figure S-4. Effect of effective response time on benchmark times for k_s^{Gon} and k_s^{Nav} on two different computers. On the horizontal axis is the effective response time, and the vertical axis displays the benchmark time; note that the vertical axis is not fixed across panels. The parameter space used in this plot can be found in Table T-3.

varied over a greater range and were input as a vector; the parameter space can be found in Table T-1. These results more closely mimic that of Figure 6 in the `fddm` paper, but all of the "combined-time" implementations available in the `dfddm()` function have shorter tails. The shorter tails are likely due to more response times being either "small" or "large" in the extreme and fewer response times being somewhere in between; it is in these areas where the response times are neither "small" nor "large" where the "combined-time" approximations tend to struggle most. Still, the SWSE "combined-time" implementation has virtually no tail and outperforms the others.

5.2 Benchmark Times While Varying a Model Parameter

Second, we will present the results of running benchmark tests similar to that of Figure 8 in the `fddm` paper. These benchmark tests were performed with the same implementations and parameter space, but we varied different DDM model parameters to see how they affected the execution time of various approximations to the the DDM density functions. The parameter space can be found in Table T-1. The results shown in Figure 8 in the `fddm` paper are informative by showing the relationship between the two timescales of density function approximations (i.e., "small-time" and "large-time"), but these supplementary results are largely uninteresting; nonetheless, we include them for completeness.

Figure S-8 shows the benchmark times as we vary the parameter w, the relative starting point of the DDM. The means, 10% and 90% quantiles, and the minimum and maximum benchmark times are very flat as w varies; this indicates that w has negligible effect on the exection time of the approximations to the DDM density functions.

Again in Figure S-9, the benchmark times are relatively flat across the values of v. There is a slight spike in benchmark time around $v = 0$ for the "small-time" implementations, but it is nothing significant.

Similarly, η has negligible effect on the execution time of the approximations to the DDM density functions, as shown in Figure S-10.

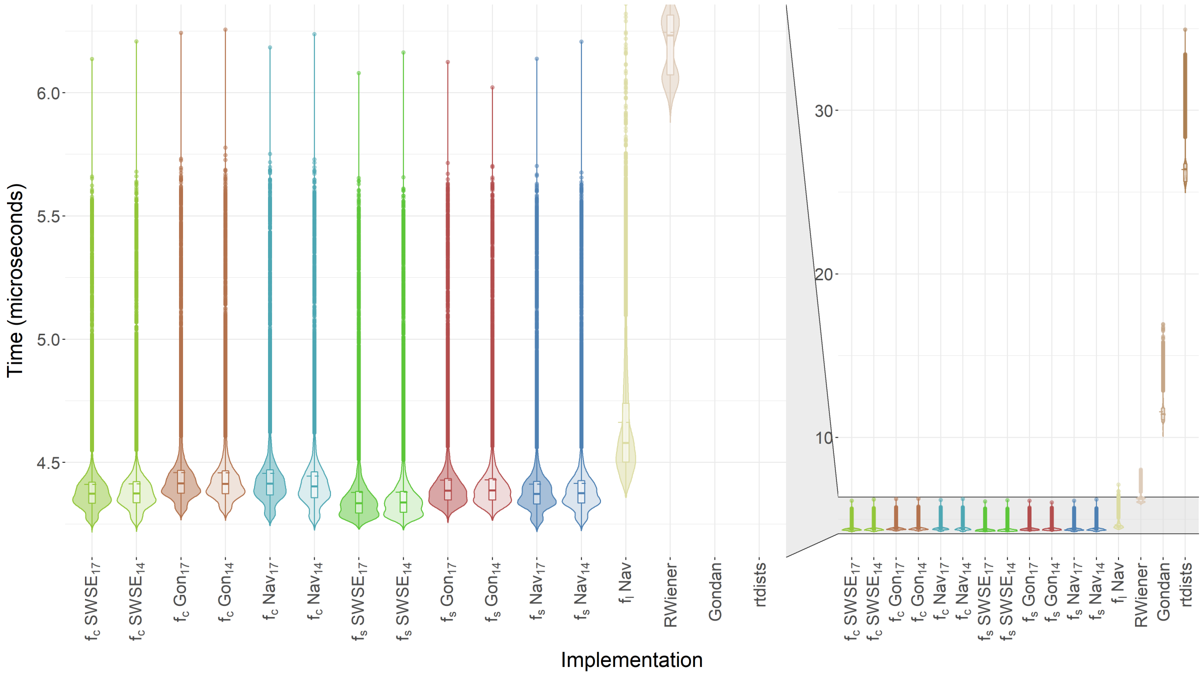

Figure S-5. Benchmark times of different density function approximation implementations for selected response times. The horizontal axis shows the implementation, and the vertical axis shows the benchmark time. The thirteen implementations on the left are options available in the `dfddm()` function, with three indicators to show: the timescale of the approximation (“**l**arge-time”, “**s**mall-time”, or “**c**ombined-time”), the paper from which the approximation was taken, and the “small-time” summation style if necessary ($S_{\mathbf{14}}$ or $S_{\mathbf{17}}$). The plot on the left is a zoomed in version of the plot on the right in order to show the more subtle differences between the benchmark data from the implementations available in `fddm`. The response times used as inputs to the implementations were between 0 and 2 seconds and input to the density function individually as opposed to input as a vector; the full parameter space can be found in Table T-2. This figure is a counterpart to Figure 6 in the `fddm` paper. For more details on the graphical elements, see Figure S-1.

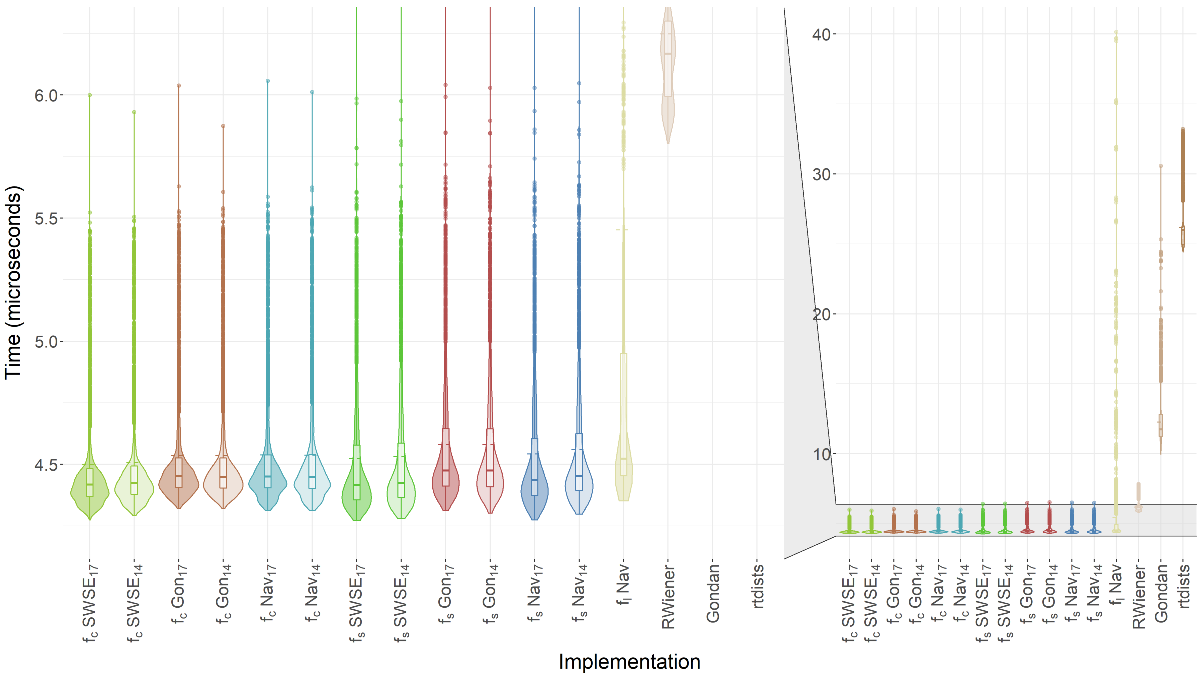

Figure S-6. Benchmark times of different density function approximation implementations for selected response times. The response times used as inputs to the implementations were between 0 and 30 seconds and input to the density function individually as opposed to input as a vector; the full parameter space can be found in Table T-1. This figure is a counterpart to Figure 6 in the `fddm` paper. For more details on the graphical elements, see Figures S-1 and S-5.

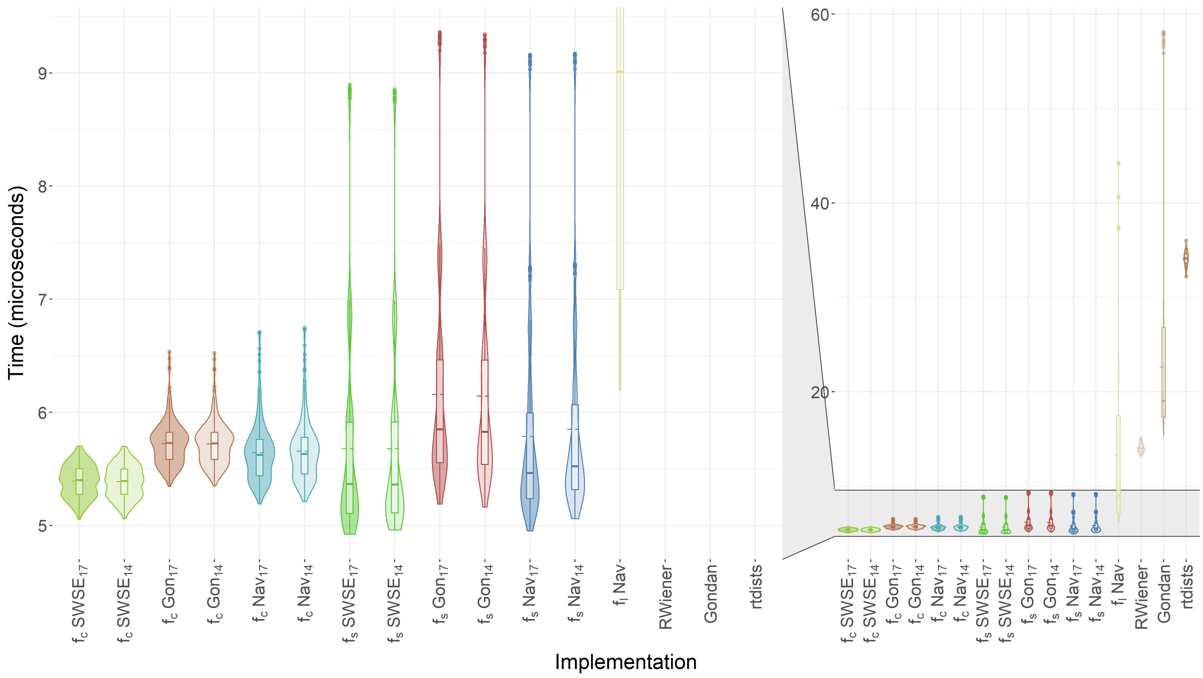

Figure S-7. Benchmark times of different density function approximation implementations for selected response times. The response times used as inputs to the implementations were between 0 and 30 seconds and input to the density function as a vector; the full parameter space can be found in Table T-1. This figure is a counterpart to Figure 6 in the `fddm` paper. For more details on the graphical elements, see Figures S-1 and S-5.

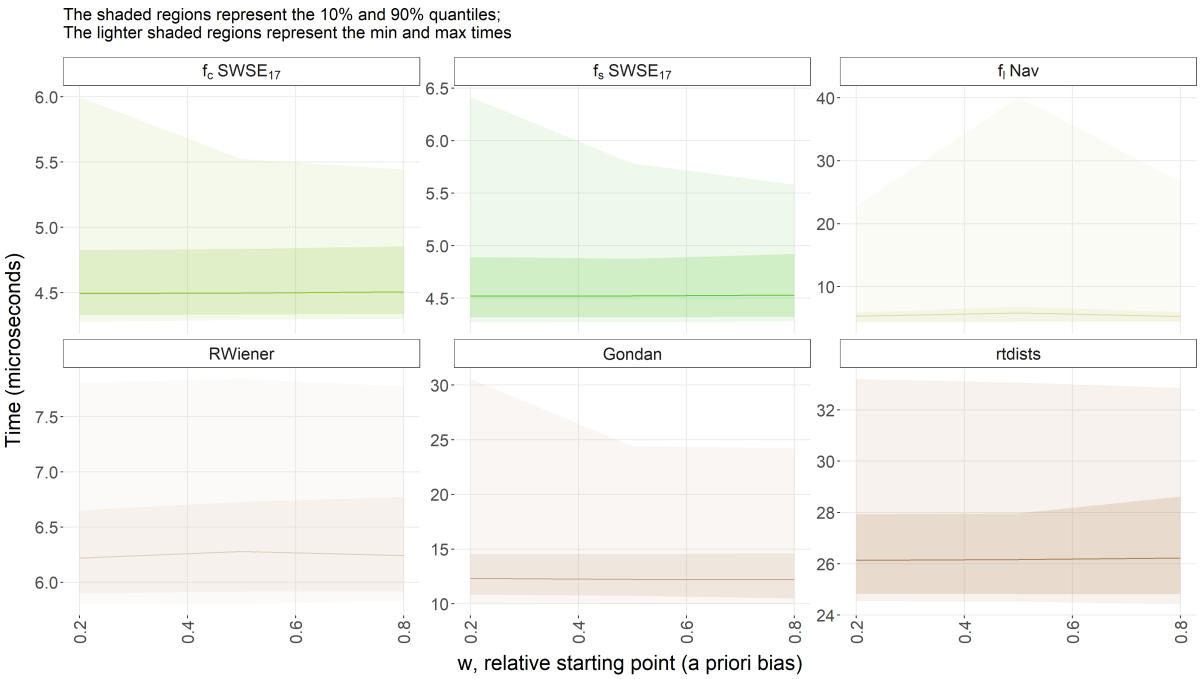

Figure S-8. Effect of w, the relative starting point of the DDM, on benchmark times for selected implementations. On the horizontal axis is the effective response time, and the vertical axis displays the benchmark time; note that the vertical axis is not fixed across panels. The dark line in each panel indicates the mean benchmark time; the darker shaded region in each panel shows the 10% and 90% quantiles; and the lightly shaded region shows the minimum and maximum benchmark times. The parameter space used in this plot can be found in Table T-1, and the response times were input individually as opposed to input as a vector. This figure is a counterpart to Figure 8 in the `fddm` paper.

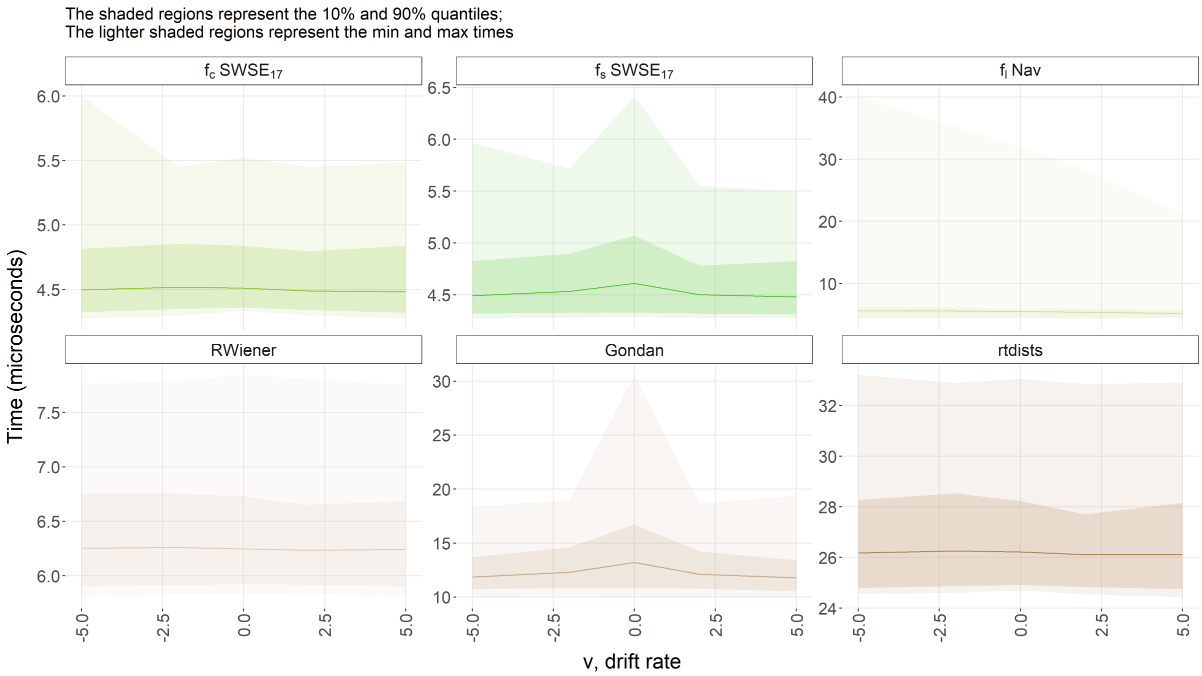

Figure S-9. Effect of v, the drift rate of the DDM, on benchmark times for selected implementations. The parameter space used in this plot can be found in Table T-1, and the response times were input individually as opposed to input as a vector. This figure is a counterpart to Figure 8 in the `fddm` paper. For more details on the graphical elements, see Figure S-8.

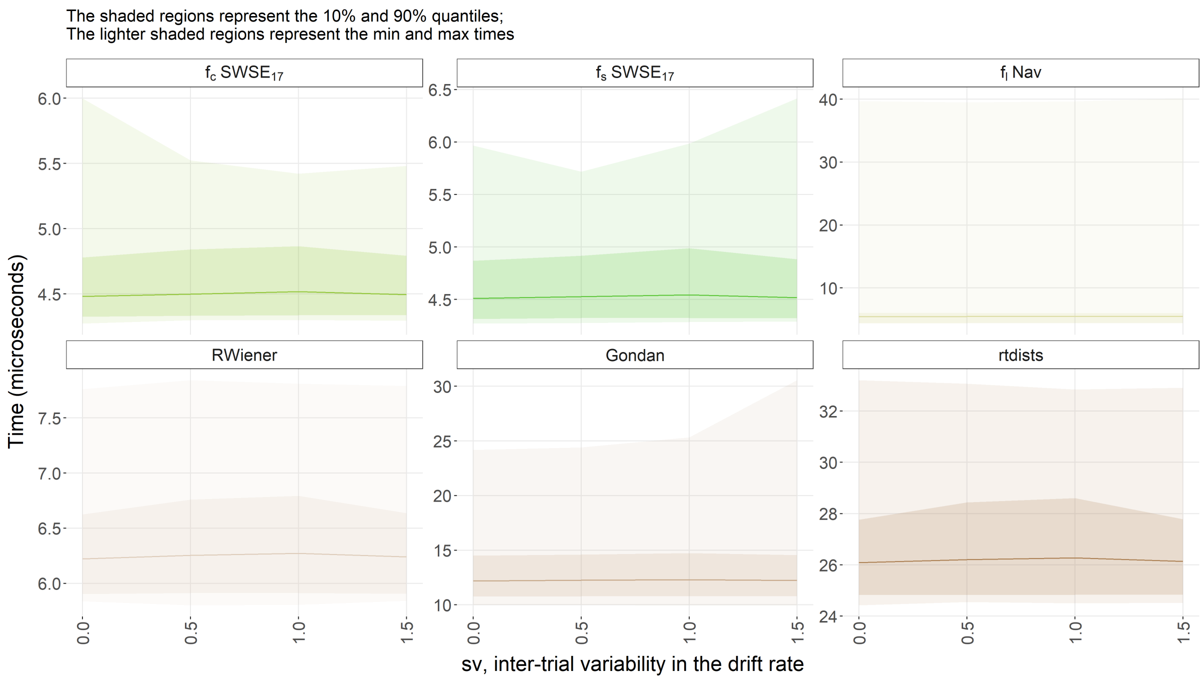

Figure S-10. Effect of η, the variability in the drift rate of the DDM, on benchmark times for selected implementations. The parameter space used in this plot can be found in Table T-1, and the response times were input individually as opposed to input as a vector. This figure is a counterpart to Figure 8 in the `fddm` paper. For more details on the graphical elements, see Figure S-8.