description	cmdstan3/cmdstanX	rstan/pystan	cmdstan2	discussion
data block values (either as file or in memory object)	data	data	data file	cmdstan2 has config hierarchy, "file" can be data or output filename; flattening argument structure will require "data_file" or just "data"
number of chains to run	chains	chains	N/A	CmdstanX wrappers run multiple chains
number of cores to use (optimistically)	cores	cores	N/A	CmdStanX wrappers can specify available cores
number of iterations during warmup	iter_warmup	warmup	num_warmup	CmdStanPy currently has "warmup_iters", "sampling_iters", but will change
number of iterations during sampling	iter_sampling	N/A	num_sampling	
total number of iterations	N/A	iter	N/A	only need to implement total or sampling iters - prefer to specify sampling iters
RNG seed	seed	seed	random_seed	can be specified on per-chain basis - continue to allow or single seed plus stride?
ids - this is the stride passed into RNG	chain_ids	chain_id	id	making this plural matches "chains" "cores". the use of "id" is necessary in order to run multiple chains using the same seed. in the case where these chains are spread across nodes on a cluster, as opposed to running all chains via a single call to the "sample" method, the user needs a way to manage the offset.
name of output file in stan_csv format with sampler config, (warmup iterations), metric, sampling iterations, timing	output_dir	sample_file	output file	CmdStan3 will be outputting more than one file and will have to do what CmdStanX wrappers are already doing - file name is a combination of model, chain, and output type
name of file for extra HMC info	latent_dynamics_file	diagnostic_file	diagnostic_file	gradient, unconstrained params, momenta realizations - objection to "diagnostic_file" is that the "diagnose" utility doesn't use these diagnostics. "extra_diagnostics", however, is equally vague.
print status messages to console	refresh	refresh	refresh	report progress every X iterations
NUTS-HMC controls for models which have parameters				don't expose all the other kinds of samplers available
set of initial parameter values	init_values	init=list of per-chain named lists>	init=<filename>	cmdstan3 should allow both a list of initial parameter values as well as initial radius - for now wrapper interfaces can't have both init values and init radius because CmdStan conflates the two.
initialize all parameter values to random value within interval <radius, radius	init_radius	init="random" init_=<positive value>	init=<radius value>	
whether or not to save warmup iterations	save_warmup	save_warmup	save_warmup	
period between saved iteration	thin	thin	thin	
limits to maximum number of leapfrog algorithm steps	max_treedepth	max_treedepth	max_depth	
metric type: unit_e, diag_e, dense_e	inv_metric	metric	metric	
initial metric - diag or dense	inv_metric_file	N/A	metric_file	should be inv_metric_file in CmdStan3
initial step_size for HMC	step_size	stepsize	stepsize	"step_size" follows better naming conventions - worth changing?
step size jitter	remove from all interfaces	stepsize_jitter	stepsize_jitter	intended as a way to keep adaptation from getting stuck, but now deemed unnecessary
should sampler do adaptation?	adapt_engaged	adapt_engaged	engaged	
adaptation target acceptance statistic	adapt_delta	adapt_delta	delta	
additional controls to Nesterov dual-averaging algorithm	remove from all interfaces	adapt_gamma adapt_kappa adapt_t0	gamma kappa t0	only god can tune Nesterov algorithm; this was already proposed last summer - "get rid of jitter, kappa, gamma, and t0, as we haven't seen any use cases for these parameters."
adaptation schedule during warmup	adapt_init_phase_size adapt_term_phase_size adapt_base_window_size	adapt_init_buffer adapt_term_buffer adapt_window	init_buffer term_buffer window	try to find the typical set adapt step size given fixed inv metric the initial size for an exponentially growing number of draws used to estimate the metric.
	save_latent_dynamics			flag used by CmdStanX interfaces to save diagnostics/grad_upараm file
Stan program, no parameters specified	fixed_param	algorithm="fixed_param"	fixed_param	run a Stan program which computes a set of QOIs