We thank the reviewers for their thoughtful feedbacks. We are encouraged they think M-HMC is addressing an important problem (**R1**, **R2**), "is potentially going to be very influential" (**R1**) and is "a method with potential wide usage" (**R2**). We are glad all reviewers find the paper to be well-written and clearly positioned w.r.t. prior work, the theory is correct and "presented carefully" (**R1**), the experiments are "extensive" (**R1**), "well-designed" (**R2**), "very reasonable" (**R4**).

[**R2**] **...too many contents on the correctness of the proposed methods are in the appendix...I would like to increase my score if a better sketch of the proof can be made in the main paper.** Great point. The main idea is to decompose the transition probability kernel $R_T(s, B)$ into summation along countable number of deterministic "probabilistic paths", then use similar proof techniques as in RRHMC. Starting from $s \in \Omega \times \Sigma$, for a given travel time T, an M-HMC iteration (without MH correction) specifies a deterministic mapping for a fixed sequence of random discrete proposals (Y in appendix). We introduce "probabilistic path" ($\omega(s, T, Y)$ in appendix), containing all information (sequence of proposals, indices/times&accept/reject decisions for discrete updates, and evolution of the discrete/continuous states and auxiliary variables) of such deterministic trajectories for fixed Y's. There can only be a countable number of possible probabilistic paths, since T and discrete state space are finite, so traveling from s for time T gives a countable number of possible destinations s'. This implies the decomposition $R_T(s, B) = \sum_{s'} \sum_{Y} r_{T,Y}(s, s')$, where the summation is done over possible destinations s' and all valid Y's that bring s to s' through $\omega(s, T, Y)$. $r_{T,Y}(s, s')$ is the transition probability along the deterministic trajectory $\omega(s, T, Y)$. Using similar proof techniques as in RRHMC, we can prove detailed balance for these deterministic trajectories (Lemma 4 in appendix). This in turn proves detailed balance of M-HMC. ***We will expand and include the above proof sketch in camera-ready version.***

[**R4**] **The distinction between this method and MH within HMC is not sufficiently strong for this work to be considered very novel...the implementation in Appendix L262...don't quite match the description in the paper...The comparison of M-HMC to MH within HMC is questionable.** This seems a *factual* misunderstanding. This comparison is not meant to be novel. Rather, MH (with or without self-transition) within HMC isn't valid (Fig. 1c), while M-HMC is (Fig. 1a). We can't imagine a stronger comparison. Appendix L262 is MH within HMC (L188, appendix L178), so **R4** is correct, it's invalid. M-HMC is in appendix L256, which is valid and matches Algo. 1. To us, the distinction being simple in this 1 discrete variable case is a strength. This simple change naturally comes out of Algo. 1, and corrects the inherent bias in MH within HMC (Fig. 1a/1c). In fact, we specifically included this function to highlight this simple distinction, showing M-HMC is correct, more efficient, yet extremely easy to incorporate into existing HMC implementations. For novelty: **R1**, **R2** both think M-HMC solves an important problem and can potentially be very influential and see wide usage. Even **R4** mentioned "The work is clearly a novel extension of previous work".

[**R4**] **M-HMC equally suffers from this issue (long trajectories for random-walk suppression vs frequent discrete updates) because if the discrete variables are frequently updated...then the trajectory length between successive discrete random variable updates is short.** We respectively disagree. In HMC, using long trajectories gives distant proposals/random-walk suppression (due to consistent momentum within an iteration), but means infrequent discrete updates. Shorter trajectories enable more frequent discrete updates, but frequent momentum resampling between iterations increases random-walk behavior. In contrast, M-HMC can always utilize consistent momentum/kinetic energy in long trajectories, with potentially frequent discrete updates, to get distant proposals/random-walk suppression. The trajectory lengths between successive discrete updates are irrelevant, as these updates involve no momentum resampling.

[**R3**] **No theoretical insights about the speed-up...** Intuitively speed-up is from more frequent discrete updates. More detailed theoretical analysis is good future work. [**R3**] **...M-HMC will have to introduce much more continuous variables than DHMC...** Using Algo. 1, we only need 1 continuous variable ($k_i^{\mathcal{D}}$ in Algo. 1) for each discrete variable. DHMC needs 2 ($q_i^{\mathcal{D}}$ in L85, and $p_i^{\mathcal{D}}$). M-HMC also doesn't need to update all discrete variables every time, making it more efficient than DHMC (see L265-268 for comparison on BLR). [**R3**] **...performance...very dependent of...T...suggestions...?** HMC is known to be sensitive to T (and step size). So is M-HMC. Automatically picking these parameters is important future work (L325). [**R3**] **It's not clear at all...the discrete part...needs...HMC...** This mechanism enables discrete updates within HMC, which, if done naively, is invalid. The proposals Q_i's are used instead of gradients information. A random walk type implementation would be the invalid MH within HMC (Fig. 1c).

[**R2**, **R4**] **HMC-within-Gibbs (HwG) should be used instead of NwG as baseline** Additional experiments with HwG show M-HMC is 2.5 (24D GMM)/3.6 (BLR)/3.3 (CTM) times more efficient than HwG in MRESS. HwG is indeed better than NwG as **R2** suggested. ***We will include HwG as an additional baseline in camera-ready version.*** [**R2**] **...MCMC component for discrete...should be emphasised...** Will do. [**R2**] **...different MCMC components for discrete variables should be experimented.** We have started (and will include in camera-ready version) experiments with *Turing.jl* for particle Gibbs. But we share **R2**'s expectation that M-HMC would still be better. [**R2**] **How adaptation for NUTS is set up?** Dual averaging, 0.8 target acceptance. [**R2**] **The outperforming of M-HMC over NUTS may be due to NUTS stuck in some of the modes...using a larger step size manually...might be simply better...** This turns out to be exactly the case. NUTS MRESS increased to 1.37×10^{-3} with step size 5 (v.s. 4 from dual averaging), outperforming M-HMC as one expects. ***We will update the paper with these observations.***

[**R1**] We will add a concise summary of experiments. [**R2**] We visualized the kernel for 1d GMM in 4.1, showing discrete updates and distant proposals. We will include the visualization, along with suggested proof/algorithm re-organizations. [**R3**] We will keep only Fig. 4a for 1 chain in main paper to improve readability of the traceplots.