We thank the reviewers for their constructive and inspiring feedback. We will improve the paper by incorporating the following responses. As we cannot see R2 (i.e., Reviewer #2), we respond to the reviews by R1, R3, and R4 only.

[R1/R4] The correctness of autodiff systems defined in the paper could be misleading to practitioners. Any relationship between the correctness of autodiff systems and that of applications (e.g., gradient descent) built upon autodiff systems?

We agree with the reviewers' points that (i) the correctness of the applications built upon autodiff systems is as important as that of the underlying autodiff systems, but (ii) the former does not necessarily follow from the latter (especially as defined in the paper). These two notions of correctness address separate issues, and our work is mainly about the second notion (i.e., the correctness of autodiff systems). Also, we do not claim that our correctness condition is "the" gold standard. Rather we are just suggesting "a" correctness condition that can serve as a reasonable (possibly minimal) requirement for existing and future autodiff systems. We will clarify this limitation in the revised version of the paper.

Here are detailed responses to the point (ii) on the applications mentioned in the reviews.

- Gradient descent: As illustrated by R1's example of $f(x)$, our correctness condition for autodiff systems does not necessarily imply the correctness of the gradient descent based on those systems (i.e., that the gradient descent converges to Clarke critical points). This gives a partial answer to R3's question on possible drawbacks of using intensional derivatives. On the other hand, we conjecture that if gradient descent is based on intensional derivatives and starts at randomly chosen initial points, it would be correct almost surely. This is an open problem.
- Hamiltonian Monte Carlo (HMC) and variational inference (VI), possibly for probabilistic programming: We reiterate that PAP functions enjoy a nice property that they are analytic on each piece of domain, whose boundary is measure-zero. The property has been crucially used to design various methods of HMC and VI for non-differentiable densities and prove their correctness (e.g., [1, 2]). This signifies the importance of studying PAP functions. Whether the correctness claims in those works would still hold if intensional derivatives are used in place of standard ones, is another open problem. We will discuss these interesting open problems in the revised version of the paper.

Although there remain a few open problems, we strongly believe that our work would serve as an important first step towards understanding and resolving those problems. Above all, as far as we know, this is the first work that (i) raises subtleties in the well-known chain rule when applied to almost-everywhere differentiable functions, (ii) gives concrete counterexamples illuminating those subtleties, and (iii) proves some reasonable (possibly minimal) correctness of existing autodiff systems that permits non-differentiable functions, using only elementary mathematics.

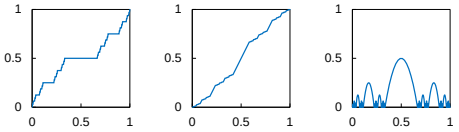
[1] Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods. Biometrika, 2020.
[2] Reparameterization Gradient for Non-Differentiable Models. In NeurIPS, 2018.

[R1] Is there any better correctness condition for autodiff systems that permits some non-differentiable functions but, at the same time, ensures nice behaviors of autodiff systems when applied to differentiable functions?

This is a good question that would lead to interesting future work. We do not have concrete results, but a possible approach would be to divide PAP representations and intensional derivatives into "good" and "bad" ones, and consider a new correctness condition that involves only those "good" ones. A desired property of "good" intensional derivatives might be that they must be identical to the standard derivative if a given function is differentiable everywhere. Under the property, R1's $f(x)$ is considered a "bad" PAP representation. A promising idea to construct "good" PAP representations that induce "good" intensional derivatives is to put additional requirements to their domain partitions, such as that each piece of a domain partition should be a half-open interval if the entire domain is $\mathbb{R}$; R1's $f(x)$ violates this requirement.

[R3] Plots of functions used in the proof of Proposition 1?

Shown right are draft plots of the 1-Cantor function, and f and g in the proof of Proposition 1. We will add them to the paper.



[R4] Implementation of relu, reciprocal_no_nan, and sqrt? Any connections to observed values of their "derivatives"?

We checked that TensorFlow and PyTorch compute the "derivatives" of the three functions f, not by applying autodiff (or symbolic differentiation) to the implementation of f, but by evaluating a separately written implementation of $\widetilde{D}f$. The implementations of f and $\widetilde{D}f$, denoted by `f` and $\widetilde{D}$`f`, are as follows.

	$\texttt{relu}(x)$	$\widetilde{D}\texttt{relu}(x)$	$\texttt{recip}(x)$	$\widetilde{D}\texttt{recip}(x)$	$\texttt{sqrt}(x)$	$\widetilde{D}\texttt{sqrt}(x)$
TensorFlow	$\texttt{\_max}(x, 0)$	$\texttt{if}\ (x > 0)\ 1\ 0$	$\texttt{\_div}(1, x)$	$\texttt{\_div}(-1, x \times x)$	$\texttt{\_sqrt}(x)$	$0.5/\texttt{\_sqrt}(x)$
PyTorch	$\texttt{if}\ (x \leq 0)\ 0\ x$	$\texttt{if}\ (x \leq 0)\ 0\ 1$	N/A	N/A	$\texttt{\_sqrt}(x)$	$1/(2 \times \texttt{\_sqrt}(x))$

Here `_max` and `_sqrt` are functions provided by a standard math library, and $\texttt{\_div}(x_1, x_2)$ is implemented as $\texttt{if}\ (x_2 = 0)\ 0\ (x_1/x_2)$. If we interpret `relu` and $\widetilde{D}$`relu` as PAP representations, one can see that $(\widetilde{D}\texttt{relu}) = D(\texttt{relu})$ may fail depending on the implementation of `_max` (e.g., consider $\texttt{\_max}(x_1, x_2) = \texttt{if}\ (x_1 \geq x_2)\ x_1\ x_2$). This suggests that "derivatives" of primitive functions f in autodiff systems could have nothing to do with the implementation of f.