We thank the reviewers for their thoughtful comments and insights. We are encouraged that reviewers found our work to be tackling a significantly novel (**R1**) task, that is both ambitious (**R3**) and more complicated (**R4**) than its rasterized counterpart. We are glad they found our method to be novel (**R1**), thoughtful (**R3**) and intuitive (**R4**). The release of our collected dataset is appreciated by the reviewers (**R3**) and qualified as interesting for the research community (**R2**). We are also pleased that reviewers liked the figures and interactive tool provided in the supp. material of our work (**R3**).
Code and dataset release: Since our submission, we have indeed released all code and the Icons-8 dataset.

R1-W1. More complex SVGs: Our work is intended as a step towards the ability to process more and more complex vector graphics. Note that the icons data considered in this paper represents a significant increase in complexity compared to the sketch and font datasets used in prior work. Moreover, unlike [4, 10], our model learns representations of *arbitrary* icons, without being conditioned on auxiliary class labels (e.g. glyph unicode or sketch category).
R1-W2. Readability: We thank **R1** for the suggestions, and will revise the specified parts to increase readability.
R1-C1, R4-W4. SVG-VAE: We aim to design a transformer-based architecture that avoids the extra step of rasterization in the encoder/decoder pipeline. In this regard, SVG-VAE is not our baseline; in fact, our experiments on the fonts dataset (see Sec. F in the supp.) suggest that our baseline performs better than SVG-VAE.
R1-C2,C3, R4-W2. Baseline: We regret that the description of our baseline (L192-197) is short due to space limitations. In the final version, we will provide a detailed description of all methods in Tab. 2. As described in Sec. 4.1, our baseline predicts commands autoregressively, while the 'one-stage feed-forward' does it in one forward pass (L127-138).

R2-W1a. Terminology: We regard *animation* to be a task and investigate *interpolation* as *one* approach to perform it. We will clarify this further in the revised paper. We believe our generative model could be used as an interactive tool to assist 2D animators in shape morphing between two keyframes. This process can be repeated iteratively – adding a hand-drawn keyframe at every step – until a satisfying result is achieved. **R2-W1b. Generation results:** We provide generative examples, sampled from the learnt distribution of both fonts and icons in Sec. F and G of our supp. material.

R2-W2. Motivation: A major purpose of our work is, as for prior VAE designs [10], to learn deep *representations* of SVGs. As also mentioned by **R1**, we believe that this ability has many applications, including image vectorisation, style transfer (investigated by the *squarify* op. in Sec. 4.3), classification, animations, or indeed text-conditional generation.

R2-W3. Experiments: Since SketchRNN is only able to process polyline drawings, the only possible direct comparison with prior work is SVG-VAE [10], which we present in Sec. F of the supp. material. We will move it to the main paper.

R3-W1, R4-W5. Quantitative measures: We are unaware of quantitative metrics for vector image generation. We therefore here propose two metrics. We first define the Chamfer distance between two SVGs: $d_{\text{Chfr}}(V, \hat{V}) = \frac{1}{N_P}\sum_{i=1}^{N_P} \min_j \int_t \min_\tau \|P_i(t) - \hat{P}_j(\tau)\|_2 dt$, where $P_i \in V$ is a path (L91). The *Reconstruction Error* (RE) is $d_{\text{Chfr}}(V, \hat{V})$ where V and $\hat{V}$ are the GT and reconstruction. The *Interpolation Smoothness* (IS) is defined as $\sum_{k=1}^{M} d_{\text{Chfr}}(V^{\alpha_{k-1}}, V^{\alpha_k})$, where M is the number of frames, $\alpha_k = k/M$ and V^α is the predicted SVG interpolation parametrized by $\alpha \in [0, 1]$. Results are shown in Tab. 1. Compared the the One-Stage method, our approach achieves improved RE on the test set and significantly better interpolation quality (IS). We will add this experiment along with a discussion and additional details in the final version.

Table 1: Metrics on the train / test sets.

	RE	IS
Baseline	0.10 / 0.17	0.25 / 0.36
One-stage feed-forward	**0.007** / 0.014	0.12 / 0.17
Ours (Ordered)	**0.007 / 0.012**	**0.08 / 0.12**

R3-W2, R4-W5. More complicated SVGs and generalization: As mentioned in **R1**-W1 above, please recall that the setting we tackle in this work is a significant *increase* in complexity compared to prior works [4,10]. Out of domain representations can be processed and interpolated, as shown in Sec. 4.2, where user-drawn images do not appear in the training set. The maximum number of paths, N_P, is simply a parameter that can be adjusted based on the statistics of the dataset. In the extreme case where the number of paths exceeds the specified N_P, the SVGs can still be processed by partitioning it into smaller parts, analogous to how CNN-based image generation networks are applied to HD images.

R4-W1. Eq. 2: The mentioned equation is intended to clarify that we predict paths and commands in a purely feed-forward manner – as opposed to autoregressive models, whose predictions are conditioned on previous outputs. It covers both the hierarchical and one-stage architectures. In the former setting, as illustrated in Fig. 4, $(\hat{c}_i^j, \hat{X}_i^j))_{j=1}^{N_C} = D^{(1)}(\hat{u}_i)$ is output by $D^{(1)}$, where the path encoding $\hat{u}_i$ is itself predicted as $\{\hat{u}_i\}_{i=1}^{N_P} = D^{(2)}(z)$ based on latent vector z. Thus, in the hierarchical case, we can write $p(\hat{X}_i^j|z, \theta) = p(\hat{X}_i^j|\hat{u}_i(z, \theta), \theta)$, and similarly for the other factors in Eq. (2).

R4-W3. Interpolation quality: To quantitatively evaluate interpolation quality, we introduce the interpolation smoothness (IS) metric (see **R3**-W1), reported in Tab. 1. The hierarchical model achieves superior performance compared to the one-stage architecture. In particular in the first two examples in Fig. 5 (onion-tree, footstep-shovel), the one-stage model's interpolations suffer from instability, as reflected in the color changes. This results in severe flickering, which is captured by our IS metric (Tab. 1) but is difficult to visualize in printed form. For the human study, the raters were therefore shown looped animations, as included in our interactive supplementary material.