

Peer Review

Review of: "Deep Sylvester Posterior Inference for Adaptive Compressed Sensing in Ultrasound Imaging"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper addresses ultrasound imaging, where an image is built from sequentially acquired scan-lines, so acquiring fewer lines per frame frees frame rate, depth of view, and power (Section I). Earlier learned subsampling masks are fixed and ignore what was observed during the scan. Unlike van de Camp et al. [10], who used Markov Chain Monte-Carlo posterior sampling on a static scene, this work uses one encoder pass and handles video. The method (Section II.B) is a variational autoencoder whose encoder is extended with a Sylvester normalizing flow. From a partially observed frame, it gives an approximate posterior over a latent code in one network evaluation. The generative model is pre-trained on full frames and frozen, then the inference model is trained on partial observations. Lines for the next frame are chosen greedily to maximise the marginal entropy of the future observation (Eq. 4, Section II.C), with covariance sampling (Eq. 6) or a cheaper trace sampling (Eq. 7).

Evaluation (Section III.A) uses EchoNet cardiac videos (6,986 training, 500 validation, 500 test) at budgets of 6, 9, 12, and 15 scan-lines (5.4% to 13.4% observation, Table I), against uniform random, variable density, and equispaced baselines. The Abstract claims the method outperforms them, "improving mean absolute reconstruction error by 15%". Section III.B reports 0.015 seconds per step for trace sampling on an NVIDIA GeForce RTX 2080 Ti, about 66 Hz, presented as fast enough for real-time 2D imaging.

2. Whether the conclusion holds on the data

The question is practical and well posed, and several things are done carefully. All five policies share one frozen generative model, so, as Section III.B says, "the differences in performance can be attributed to the sampling strategies only". Table I gives the full-observation upper bound in its caption (L1-Loss=0.053,

SSIM=0.523, PSNR=68.33), so the reader sees how much room is left. The authors report the $l = 12$ result where their method loses. The Discussion (Section IV) names the risk that joint optimisation "may lead to collapse". Timing is measured on a named consumer GPU without JIT, pruning, or quantization (Section III.B).

The Abstract says "improving mean absolute reconstruction error by 15%". The string "15%" appears only in the Abstract, and the baseline and line budget are not stated. From Table I (L1-Loss), Trace against Uniform random gives 17.6% at $l=6$, 18.4% at $l=9$, 11.6% at $l=12$, and 10.8% at $l=15$, with a mean of 14.6%. So 15% seems to be the average gain over uniform random sampling (my reconstruction, the paper does not say), and I would name the baseline in the Abstract. Against Equispaced, the picture is different (Table I, L1-Loss): 0.070 vs 0.073 at $l=6$ (4.1%) and 0.062 vs 0.064 at $l=9$ (3.1%). At $l=12$, Trace is worse, 0.061 vs 0.060, also in SSIM (0.489 vs 0.494) and PSNR (67.60 vs 67.73). At $l=15$, both are 0.058 in L1-Loss, SSIM is slightly better for Equispaced (0.500 vs 0.502), and PSNR is slightly better for Trace (67.93 vs 67.89). Section III.B says Trace outperforms the others at $l=15$, and I would write "about equal" there.

The covariance policy, derived from the information-gain objective (Eq. 6), is behind Equispaced at every budget in Table I (L1-Loss 0.082 vs 0.073 at $l=6$; 0.071 vs 0.064 at $l=9$). Section III.B only says "Interestingly, the trace-based sampling policy outperforms the covariance sampling method", and I would add an explanation. The evidence supports "a trace heuristic with neighbour exclusion slightly beats equispaced lines at 6 and 9 lines", not yet "maximising mutual information beats static sampling". It would also help to add a run with a plain Gaussian encoder (zero flow steps) against $K=8$ (Section III.A), since the flow is in the title. A learned fixed-mask baseline such as Deep Probabilistic Subsampling [5] or its active version [8] would be a useful comparison too.

Table I gives means over 500 test videos with no standard deviation, confidence interval, or paired test, while the gaps to Equispaced are 0.003 at $l=6$ and 0.002 at $l=9$ in L1-Loss. The two random baselines depend on a random draw, and the number of repetitions is not reported. I searched the text for "deviation", "seed", and "confidence" and found nothing, and statistical significance is not discussed, so I would add these. On timing, 0.015 seconds for trace sampling gives 66.7 Hz, above the 50 Hz capture rate (Section III.A), so the real-time statement holds for trace sampling. Covariance sampling takes 0.112 seconds, or 8.9 Hz. Section I claims inference time is reduced "by several orders of magnitude" relative to [10], but no timing of [10] is reported, so that comparison is not measured in the paper.

Finally, the experiment subsamples columns of stored B-mode videos converted to polar coordinates (Section III.A), not raw scanner data. This is a fair first step, and I would state it as a limitation. In short,

the gain over uniform random sampling and the real-time claim for trace sampling are supported, while superiority over Equispaced, the benefit of the information-gain objective, and the speed-up over [10] are not yet.

3. Reproducibility

I appreciate how much is already specified in Section III.A. The dataset is the public EchoNet [14] with 10,030 videos, split into 6,986 training, 500 validation, and 500 test, with 2,044 excluded "due to artifacts or missing data" (the numbers add up). The architecture is described clearly: an encoder with 10 gated convolutional layers, stride 2, $c=64$ channels, and a 512-dimensional latent; $K=8$ flow steps with 16 orthogonal vectors; a decoder with 8 blocks of gated transpose convolutions, $c=128$. Also given are $\beta = 1 \times 10^{-4}$, $N_S=3$ posterior samples, and $S=10,000$ candidate masks per frame. Section III.B names the hardware for timing: an NVIDIA GeForce RTX 2080 Ti and PyTorch 2.2.

To make the training repeatable, I would add the optimizer, learning rate, batch size, number of epochs, training time, and random seeds. I searched the text for "learning rate", "epoch", and "Adam" and found nothing, and "batch" appears only as Batch Normalization. The number of importance samples in IWAE [18] is also not given (Section III.A). Section III.A says "The training procedure is given in Algorithm 1", but I could not find Algorithm 1 in the text of the Qeios page (the word occurs only in that sentence and in "the IWAE[18] algorithm"). If it is embedded as an image, please check that it displays. I also found no link to code or trained weights: "github" does not occur, and "code" occurs only inside words such as "encoder" and "decoder". Sharing both would help readers a lot.

For the data, it would help to list the 2,044 excluded videos and state the exclusion rule. The file lists for training, validation, and test are also needed to repeat Table I. I would also write out the polar grid size (N_r , N_γ) and the interpolation used for the Cartesian-to-polar conversion. The percentages in Table I imply 112 scan-lines ($6/112 = 5.4\%$, $15/112 = 13.4\%$), but this should be stated. Please say whether 0.02 in $n \sim N(0, 0.02)$ is the variance or the standard deviation. The variable density baseline is "a polynomial distribution centered on the middle of the image with a decay factor of 6"; the formula would make it exact. Finally, Section III.B gives one timing figure per policy, so batch size, number of timed repetitions, and spread would be welcome. The Acknowledgements mention the Snellius supercomputer, and the training compute could be quantified.

4. What specifically to fix

1. Abstract, "improving mean absolute reconstruction error by 15%": I would state the baseline and the budget next to this number. From Table I (L1-Loss), the gain of Trace over Uniform random averages 14.6% over the four budgets (17.6%, 18.4%, 11.6%, 10.8%). Over Equispaced it is 4.1% at $l=6$ $((0.073-0.070)/0.073)$, 3.1% at $l=9$ $((0.064-0.062)/0.064)$, negative at $l=12$ (0.061 vs 0.060, about -1.7%) and zero at $l=15$ (0.058 vs 0.058). I would report the gain over Equispaced next to the 15%, and reword the Abstract sentence that lists equidistant lines among the outperformed baselines. In Section III.B, the $l=15$ column is a tie on L1-Loss, with SSIM 0.500 vs 0.502 in favour of Equispaced and PSNR 67.93 vs 67.89 in favour of Trace, so "outperforms the other methods at ... $l=15$ " should be softened.
2. Eq. 5 and Eq. 6 with $N_S=3$ (Section III.A), a question: as I read Eq. 5, the covariance is a sum of N_S outer products, so its rank is at most $N_S = 3$ (at most 2 if the mean is the sample mean of the same three draws), while the observation has $M = N_r * M_{\text{gamma}}$ entries (six full scan-lines at $l=6$). If so, the determinant is zero and the log-determinant in Eq. 6 is not finite unless something is added, for example the observation noise of Eq. 1 or a diagonal loading. I did not find such a term in Eq. 5 or in the text (I searched for "regulari" and "diagonal"), but I may be missing an implementation detail. Could you state how the log-determinant is computed in practice? If the estimate is rank-limited, this may be one reason the covariance policy trails Equispaced at every budget in Table I (for example L1-Loss 0.082 vs 0.073 at $l=6$), which the text leaves at "Interestingly". Section III.A says larger N_S brought little improvement; showing those numbers, or a variant with the determinant taken over a per-line summary, would tell whether the objective or its estimate is at fault.
3. Table I: it would help to add standard deviations or confidence intervals over the 500 test videos, and a paired test for Trace vs Equispaced, since the gaps are 0.002-0.003 in L1-Loss. I would also repeat the two random baselines over several seeds.
4. Table I, PSNR column: the values are 65.53 to 67.93 dB, while the L1-Loss is 0.058 to 0.086 on images normalized to [0,1] (Section III.A). With peak value 1, PSNR 66.69 dB (Trace, $l=6$) would mean $MSE = 10^{-(6.669)} = 2.1e-7$, a root-mean-square error near 0.0005. A root-mean-square error cannot be smaller than the mean absolute error on the same pixels, and the L1-Loss here is 0.070. With peak value 255 the same PSNR gives $MSE = 65025 / 10^{6.669} = 0.0139$, root 0.118, which is in line with L1 0.070. So my guess is that PSNR was computed with peak 255 on [0,1] data; this is an inference from the numbers, not something the paper states. With peak 1 the values would be about 48 dB lower ($20 * \log_{10}(255) = 48.1$). Could you state the peak value and the PSNR formula, or recompute?

5. Eq. 2: as rendered on Qeios, the prior term enters as minus $\log p(z_K)$, with the same sign as the encoder term, while in the evidence lower bound the prior log-density normally enters with a plus sign. This may be a typesetting slip, so please check the sign. Section III.A sets $\beta = 1 \times 10^{-4}$ "with the loss function defined in (2)", but Eq. 2 as rendered contains no β . Please show where β multiplies.
6. Section III.A, architecture: 10 stride-2 layers are hard to follow if the polar input is about 112 wide like the original frames, since seven halvings already take 112 down to 1 (112, 56, 28, 14, 7, 4, 2, 1). Likewise, 8 stride-2 transpose blocks give a factor of 256. A small table of layer output sizes would clear this up.
7. I would add an ablation with zero flow steps (plain Gaussian encoder) against $K=8$, in reconstruction and in sampling quality, because the title rests on the Sylvester flow and Table I does not isolate it. It would also help to add one learned baseline from the authors' own cited work ([5] or [8]), and a measured runtime for the MCMC method [10] to back "several orders of magnitude" (Section I).
8. Section II.C assumes an identity transition between consecutive latent states; an error-versus-heart-phase plot (systole vs diastole) would show where this assumption costs most.
9. Section IV cites "LSTM[21]", but reference 21 is the xLSTM paper (Beck et al., 2024), so I would cite the original LSTM or change the wording. Algorithm 1 (Section III.A) is also something I could not find on the Qeios page.

5. Recommendation

Major revisions. The idea and the engineering are sound and worth publishing: a single-pass flow encoder that makes active scan-line selection run at about 66 Hz (0.015 seconds per step, Section III.B) is a useful result, and the comparison on a shared frozen generative model is clean. The revisions are mostly about aligning claims with Table I. The headline 15% (Abstract) matches only the average gain over uniform random sampling, by my reconstruction from Table I; the gain over Equispaced is 3-4% at $l=6$ and $l=9$ and absent at $l=12$ and $l=15$. It would help to explain the covariance policy (Eq. 5-6 with $N_S=3$), and I would add error bars, a no-flow ablation, and code. The 500 test videos are enough for all of this.

Declarations

Potential competing interests: No potential competing interests to declare.