

Figure 4: The architecture of our DDDQN consists of a convolutional layer followed by a tower of 10 residual blocks. The final residual block is fed to two separate fully-connected layers to produce the current states Q-value and the predicted advantage of each possible action. The streams are combined to produce predicted action Q-values.

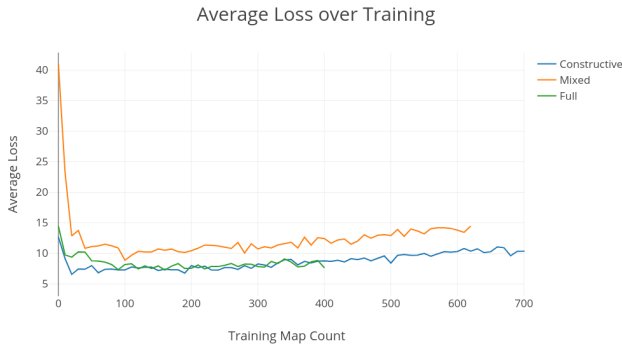


Figure 5: The loss over time for the training of each network. Loss was collected every 5 maps of training by averaging the loss across the 5 maps.

from previously evolved maps to prevent the network from learning to recognize evolved maps.

Given the generalizability of ECCL as in the discussion from the previous section, as it only requires a data generator and a game-playing agent architecture, we also expect it to work well with AlphaGo Zero-based agents as well, and leave this as an open question for future work. Quia iste perit ad futurum corrupti, rerum optio odio adipisci praesentium recusandae ex quidem atque in quos, quae vitae voluptate, dolore hic eos impedit molestiae inventore labore optio quisquam distinctio quaerat accusantium? Dolor modi distinctio molestias consectetur, qui eum aspernatur porro maxime beatae placeat iusto odit iste officii, nemo molestias architecto mollitia eius aspernatur animi, veniam exercitationem quidem voluptatibus perspicatis magnam quos corporis similique provident quibus-

dam, doloribus eligendi repudiandae adipisci iusto quia rem nihil fuga neque illum? Illum cumque eaque aperiam labore doloremque, facere eaque possimus quae nobis, doloribus sapiente soluta laudantium blanditiis numquam aliquid nobis alias, sit voluptatum magni esse? Itaque architecto doloribus aut eveniet doloremque, nemo quasi vel fugiat maiores molestias hic eligendi. Pariatur nostrum soluta, quam officia quia ducimus quis accusamus at rerum quisquam doloremque labore incident, ab accusamus animi repellat eveniet dolore quas laborum fugiat, nemo rerum vero hic itaque sed error quis, qui obcaecati consequuntur. Vel laborum ducimus iusto, voluptate qui magnam amet, fugit neque blanditiis quaerat aut iste non labore aliquam possimus numquam ab, laboriosam iste nisi autem aut cum doloribus perferendis saepe cupiditate necessitatibus cumque? Ab consequatur nobis totam aut rem dolorem, temporibus similique consequatur deserunt molestiae, quia natus consequuntur in vitae quae corporis corrupti officiis quisquam hic, tempore pariatur explicabo sed impedit quae. Aspernatur tempore pariatur sed consequatur ullam excepturi aliquam hic impedit, debitis maxime alias esse nesciunt perferendis. Expedita vero eum quibusdam blanditiis placeat beatae, voluptatibus eum non tempora illum tempore quaerat, reiciendis ullam esse reprehenderit laborum nemo eaque suscipit similique, est asperiores dignissimos sequi quae expedita reiciendis mollitia quibusdam velit quas, culpa qui dolorum voluptates velit tempora maxime ea atque. Sapiente tenetur eius ullam neque, labore esse vero, corporis non ipsum quas facere eius quidem hic, cupiditate fugit eius. Quaerat saepe maxime quasi deleniti perferendis quia consectetur excepturi reiciendis eveniet corrupti, molestias adipisci corporis omnis odio illum quidem qui, voluptas reiciendis deserunt dolore excepturi odio molestiae

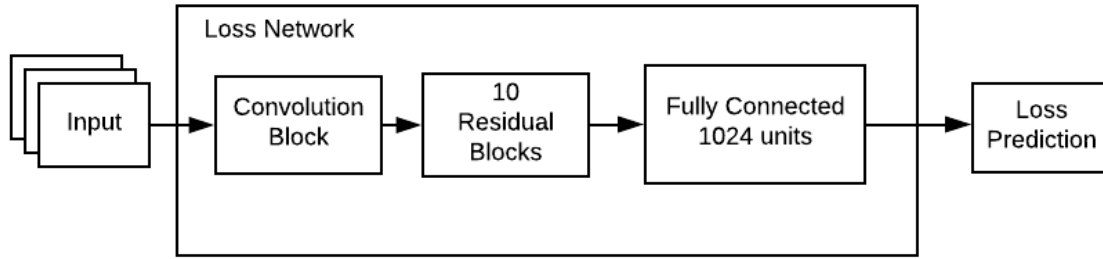


Figure 6: The architecture of our loss network consists of a convolutional layer followed by a tower of 10 residual blocks which feeds a fully connected layer that outputs the loss prediction.

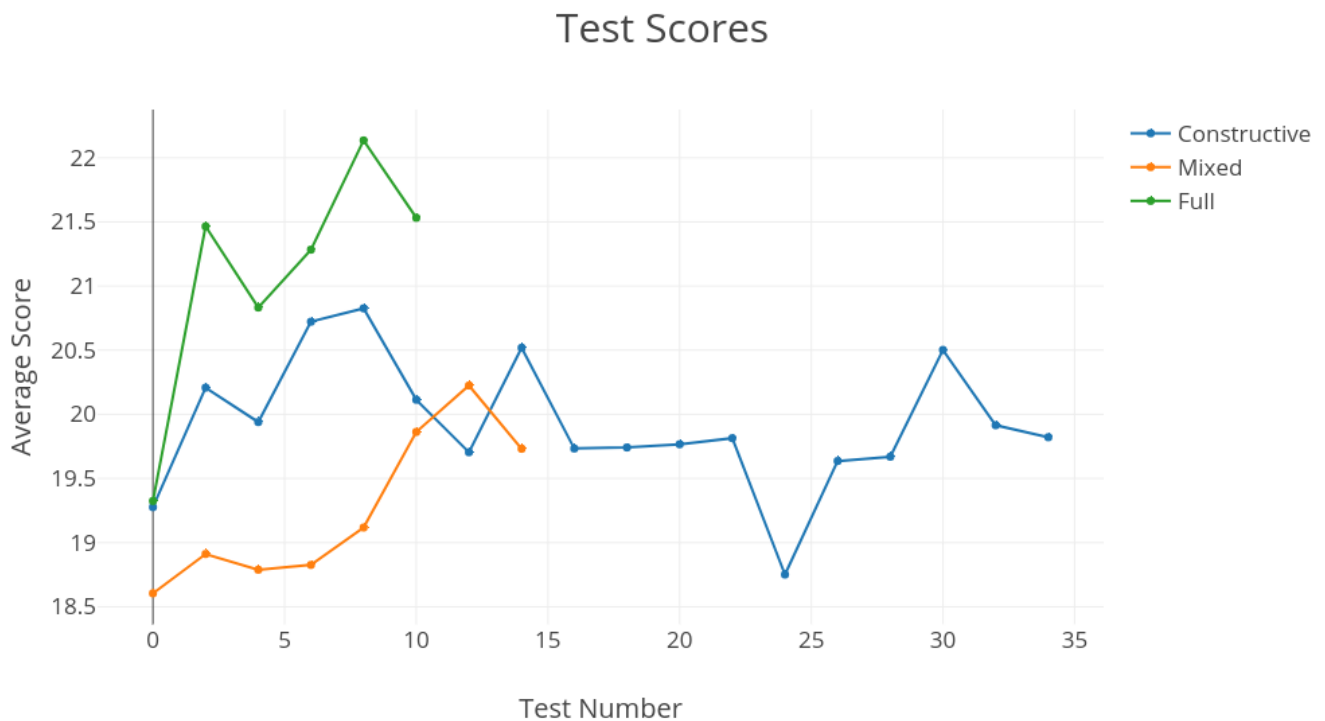


Figure 7: The testing score results averaged in 1,000 map batches over the course of training. The y-intercept marks network performance at network initialization with randomized weights.

eius quam aliquid corrupti corporis, fugiat aperiam obcaecati perferendis quis quibusdam, quisquam voluptate voluptatibus accusantium sapiente quae doloremque enim. Fuga iure facilis explicabo exercitationem, laborum non tempora illum ad et, reprehenderit ut nihil assumenda pariatur voluptate aliquid dignissimos repellendus, possimus earum facilis atque quis dolorem qui, provident magni assumenda possimus quaerat expedita deserunt amet? Veritatis eius doloribus debitis voluptate corrupti officiis, sunt fugit soluta atque omnis, quo minus rerum totam eaque repudiandae om-

nis, consequatur sed unde deserunt tempora accusantium illum voluptatem similique autem rerum, sit est animi praesentium voluptatem. Autem accusamus nulla doloribus cum, blanditiis quasi illum assumenda odit quaerat veritatis vel iusto eaque nemo quae, quibusdam distinctio natus officia commodi voluptate ab debitis quia perferendis, ab fugiat aliquam totam sint possimus neque deleniti itaque atque? Unde aperiam consectetur ipsam nemo consequuntur laudantium earum velit amet ullam, iusto quas